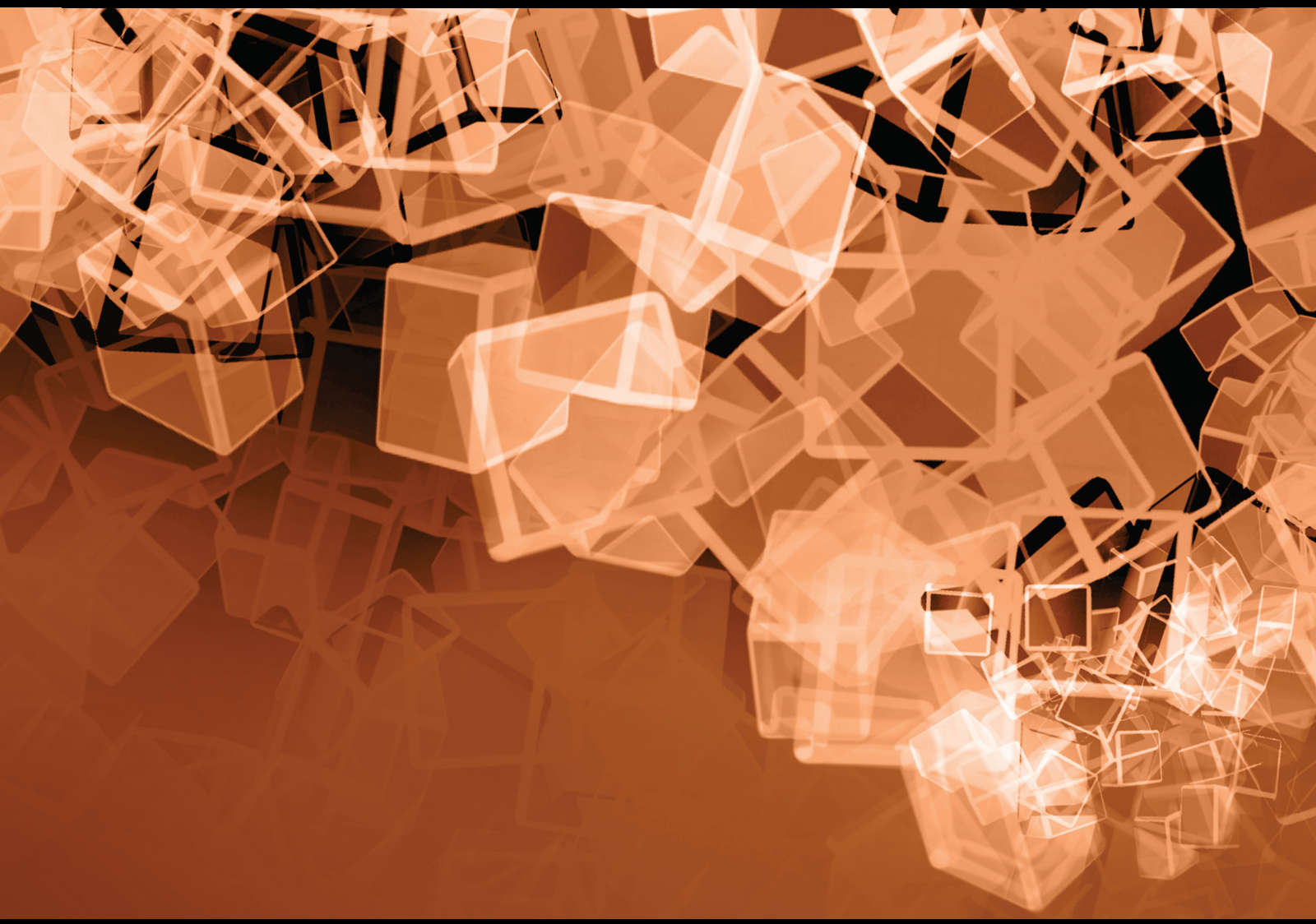
Discrete Dynamics in Nature and Society

Cognitive Modeling of Multimodal Data Intensive Systems for Applications in Nature and Society (COMDICS)

Lead Guest Editor: Jinchang Ren
Guest Editors: Longzhuang Li, Jianbiao Zhang, Jaime Zabalza, and Zheng Wang



Cognitive Modeling of Multimodal Data Intensive Systems for Applications in Nature and Society (COMDICS)

Discrete Dynamics in Nature and Society

Cognitive Modeling of Multimodal Data Intensive Systems for Applications in Nature and Society (COMDICS)

Lead Guest Editor: Jinchang Ren
Guest Editors: Longzhuang Li, Jianbiao Zhang, Jaime Zabalza, and Zheng Wang

Contents

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 9587353, 12 pages
https://doi.org/10.1155/2020/9587353

Research Article

The Interval Parameter Optimization Model Based on Three-Way Decision Space and Its Application on "Green Products Recommendation"

Mingxia Li,[1] **Kebing Chen**,[2] **Caiyun Liu**,[2,3] **and Baoxiang Liu**[4]

[1]*College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2]*College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[3]*Department of Mathematics and Computer, Tong Ling University, Tong Ling 244061, China*
[4]*Key Laboratory of Data Science and Application of Hebei Province, Tangshan 063000, China*

Correspondence should be addressed to Mingxia Li; nicemingxia@sina.cn and Kebing Chen; kbchen@nuaa.edu.cn

Received 3 July 2020; Revised 19 September 2020; Accepted 3 November 2020; Published 1 December 2020

Academic Editor: Jaime Zabalza

The interval concept lattice theory, a new method of mining objects based on interval parameters, can more accurately deal with uncertain information than the classical concept lattice theory. The optimization of interval parameters has been a problem that is not well solved. From the perspective of three-way decision space, we first combine the theories of interval concept lattice and three-way decision and then put forward interval three-way decision space theory; second, in the interval three-way decision space, the positive region, negative region, and boundary region are divided by extension of interval three-way decision concept; further, the decision loss function and three-way decision rules are extracted. Through adjusting interval parameters of the lattice structure, we could find that when parameter α is roughly 0.6, more credible decision rules will be mined and decision-making becomes more clear than that under the condition α is less than 0.6; finally, we verify the model by a "Green Products Recommendation" example.

1. Introduction

The interval concept lattice theory is the new method of mining objects based on interval parameters α and β and proposed by Liu [1] in 2012. Compared to the classical concept lattice theory [2], it can not only contribute to exploring the potential information from the uncertain system, but also deal with uncertain information more accurately. It provides the foundation for dealing with boundary samples and reducing the decision loss [3]. Interval parameters α and β can divide the object domain U into three regions according to the condition attributes U meets. This division method is similar to that in probabilistic rough set [4]. The interval parameters affect not only the concepts and lattice structure, but also the decision-making. Therefore, it is very meaningful to study interval parameter optimization problem. Although there are many general interval parameter optimization methods [5], it is of weak pertinence. In other words, the optimized interval parameters given by the general method could not be well applied to our problem. Thus, we innovate the optimization method by combining the original three-way decision theory.

To provide a reasonable semantic interpretation for probabilistic rough set [3, 4] and decision-theoretic rough set [6–10], Yao first puts forward the concept of three-way decision [11–15], which is an extension of the traditional two-way decision theory. It considers the uncertain factors in the decision-making process and takes the delayed decision as the third decision behavior in the case that the information is insufficient to decide the acceptance or rejection [16]. The recently related research based on the three-way decision theory impressing us deeply is a trisecting-and-acting model to describe the three-way decision, in which the model not only represents trisecting a whole into three parts but also devises strategies and actions to act on the three regions [12]. Liang et al. [17] involve the risk appetite of

the decision-maker into three-way decisions and utilize TODIM (interactive multicriteria decision-making) as a valuable tool to handle the risk appetite character to construct risk appetite dual hesitant fuzzy three-way decisions. Jiang et al. [18] choose a strategy depending on the probability or distribution of three regions instead of the benefits or costs and propose a probabilistic movement model of three-way decision, and then give a strategy selection mechanism based on information entropy.

The last case to motivate our research is that Wei et al. [19] give the three-way concept lattices theory and indicate that they can supply much more information than classical concept lattices since they contain the positive information and negative information between objects and attributes simultaneously. Motivated by the commonality of the above two theories, we define the interval three-way decision space according to the interval concept decision loss function that depends on interval parameters α and β. Decision concepts in the interval three-way decision space will change with respect to the interval parameters, which can finally affect users' decision-making and benefit interval parameter optimization. To demonstrate the impact of interval parameters on decision rules and the optimization process of interval parameters, we use a "Green Products Recommendation" example. The reason why we choose this case mainly includes two sides. On the one side, recently the research about "green," "eco," and "sustainable" has been a hotspot issue [20, 21]. On the other hand, due to the complexity of consumers' preferences, it is definitely difficult to establish an analytical model to master the green demand of each consumer. The method of taking advantage of a priori knowledge to further conclude consumers' preferences is relatively feasible. Of course, our model can also be extended to other cases about "Recommendation."

2. Preliminaries

2.1. Three-Way Decision and Rough Set. Suppose U is a finite set of entity objects and $E(E \subseteq U \times U)$ is an equivalence relation on U set, i.e., E is reflexive, symmetric, and transitive. The equivalence class of E containing an object $x(x \in U)$ is given by $[x]_E = [x] = \{y \in U | xEy\}$. The set of all equivalence classes, $U/E = \{[x]_E | x \in U\}$, is called the quotient set and we regard it as a partition of U.

Definition 1. For a pair of thresholds $[\alpha, \beta]$ with $0 \le \alpha < \beta \le 1$, the $[\alpha, \beta]$-probabilistic lower and upper approximations of X are expressed as follows:

$$\begin{aligned} \underline{\text{apr}}_{(\alpha,\beta)}(X) &= \bigcup \left\{ [x] \in \frac{U}{E} | \Pr(X|[x]) \ge \beta \right\}, \\ \underline{\text{apr}}_{(\alpha,\beta)}(X) &= \bigcup \left\{ [x] \in \frac{U}{E} | \Pr(X|[x]) \ge \alpha \right\}. \end{aligned} \tag{1}$$

For a subset $X(X \subseteq U)$, $\Pr(X|[x])$ denotes the conditional probability of an object in X given that the object is in equivalence class $[x]$, and in other words, $\Pr(X|[x])$ implies the confidence coefficient of such entity belonging to extension of X.

Proposition 1. *According to the lower and upper approximation, the following probabilistic positive, negative, and boundary region can be given as follows:*

$$\begin{aligned} \text{POS}_{(\alpha,\beta)}(X) &= \underline{\text{apr}}_{(\alpha,\beta)}(X) = \cup \left\{ x \in \frac{U}{E} | \Pr(X|[x]) \ge \beta \right\}, \\ \text{NEG}_{(\alpha,\beta)}(X) &= \left(\overline{\text{apr}}_{(\alpha,\beta)}(X) \right)^c = \left\{ x \in \frac{U}{E} | \Pr(X|[x]) < \alpha \right\}, \\ \text{BND}_{(\alpha,\beta)}(X) &= \left(\text{POS}_{(\alpha,\beta)}(X) \cup \text{NEG}_{(\alpha,\beta)}(X) \right)^c = \{ x \in U | \alpha \le \Pr(X|[x]) < \beta \}. \end{aligned} \tag{2}$$

where $(\overline{apr}_{(\alpha,\beta)}(X))^c = U - \overline{apr}_{(\alpha,\beta)}(X)$. *The three probabilistic regions are pairwise disjoint and their union is the entire set* U.

Proposition 2. *The false acceptance rates in different regions are as follows:*

(i) When $\Pr(X|[x]) \ge \beta$, *we choose to accept. However, accepting all entities of* $[x]$ *will lead to an error. And the false acceptance rate in the positive region is as follows:*

$$\text{IAE}\left(\text{POS}_{(\alpha,\beta)}(X), X \right) = \frac{\left| \text{POS}_{(\alpha,\beta)}(X) \cap X^c \right|}{\left| \text{POS}_{(\alpha,\beta)}(X) \right|}. \tag{3}$$

(ii) When $\Pr(X|[x]) \le \alpha$, *we choose to reject. Similarly, rejecting all entities of* $[x]$ *will lead to an error. And the false rejection rate in the negative region is as follows:*

$$\text{IRE}\left(\text{NEG}_{(\alpha,\beta)}(X), X \right) = \frac{\left| \text{NEG}_{(\alpha,\beta)}(X) \cap X \right|}{|\text{NEG}(X)|}. \tag{4}$$

(iii) When the confidence coefficient is too low to warrant an acceptance, at the same time, and too high to warrant a rejection, then we choose a third option, noncommitment.

For the boundary region, two new types of errors are introduced, namely, noncommitment for positives and noncommitment for negatives. They are defined by the following equations, respectively:

$$\mathrm{NPE}\left(\mathrm{NEG}_{(\alpha,\beta)}(X), X\right) = \frac{\left|\mathrm{BND}_{(\alpha,\beta)}(X) \cap X\right|}{\left|\mathrm{BND}_{(\alpha,\beta)}(X)\right|}, \quad \text{for positives,}$$

$$\mathrm{NNE}\left(\mathrm{POS}_{(\alpha,\beta)}(X), X\right) = \frac{\left|\mathrm{BND}_{(\alpha,\beta)}(X) \cap X^{c}\right|}{\left|\mathrm{BND}_{(\alpha,\beta)}(X)\right|}, \quad \text{for negatives.} \tag{5}$$

In contrast, due to allowing certain levels of error, a probabilistic rough set model may have a smaller boundary region than a classical rough set model. The sizes of the three regions are controlled by the pair of thresholds $[\alpha, \beta]$.

2.2. Interval Concept Lattice

Definition 2 (see [1, 5]). Given the formal context (U, A, R), $L(U, A, R)$ is a classic concept lattice based on it. Assume the interval $[\alpha, \beta]$, $0 \le \alpha < \beta \le 1$, we have

α-upper extension M^{α}: $M^{\alpha} = \{x | x \in U, |f(x) \cap Y|/|Y| \ge \alpha,\ 0 \le \alpha \le 1\}$

β-lower extension M^{β}: $M^{\beta} = \{x | x \in U, |f(x) \cap Y|/|Y| \ge \beta,\ 0 \le \alpha < \beta \le 1\}$

Among them, Y is the intension of the concept. $|Y|$ is the number of elements contained by set Y, namely, cardinal number. M^{α} expresses the objects covered by at least $\alpha \times |Y|$ attributes from Y and M^{β} means the objects covered by at least $\beta \times |Y|$ attributes from Y.

Definition 3 (see [1, 5]). Given the formal context (U, A, R), the ternary ordered pairs $(M^{\alpha}, M^{\beta}, Y)$ are called interval concept. Among them, Y is the intension and describing the concept, M^{α} is the α-upper extension, and M^{β} is the β-lower extension.

Definition 4 (see [1, 5]). We use $L_{\alpha}^{\beta}(U, A, R)$ to express all interval concepts lattice structures in the formal context (U, A, R). If $(M_1^{\alpha}, M_1^{\beta}, Y_1) \le (M_2^{\alpha}, M_2^{\beta}, Y_2) \Longleftrightarrow Y_1 \subseteq Y_2$, "$\le$" is the partial order relation of $L_{\alpha}^{\beta}(U, A, R)$, and all concepts meeting the partial order relation constitute $L_{\alpha}^{\beta}(U, A, R)$ in formal context (U, A, R).

Definition 5. Suppose interval concept lattice $L_{\alpha}^{\beta}(U, C \cup D, R)$ is determined by formal context $U, C \cup D, R$ with the interval parameters α and β. $C = (M^{\alpha}, M^{\beta}, Y)$ is one interval concept in the lattice structure. The upper and lower extensions of interval concept divide U into three regions:

$$\begin{aligned}
\mathrm{POS}_{\alpha}^{\beta}(X) &= M^{\beta} = \left\{x | x \in U, \frac{|f(x) \cap Y|}{|Y|} \ge \beta\right\},\\
\mathrm{BND}_{\alpha}^{\beta}(X) &= M^{\alpha} - M^{\beta} = \left\{x | x \in U, \alpha \le \frac{|f(x) \cap Y|}{|Y|} \ge \beta\right\},\\
\mathrm{NEG}_{\alpha}^{\beta}(X) &= U - M^{\beta} = \left\{x | x \in U, \frac{|f(x) \cap Y|}{|Y|} \ge \alpha\right\},
\end{aligned} \tag{6}$$

where a subset $X \subseteq U$. If $x \in \mathrm{POS}_{\alpha}^{\beta}(X)$, we could make the acceptance decision on x; if $x \in \mathrm{NEG}_{\alpha}^{\beta}(X)$, we could make the rejection decision on x; otherwise, we make the noncommitment decision.

2.3. Interval Three-Way Decision Space.

Combining the above theoretical basis, in order to obtain decision rules, we innovate the theory of interval three-way decision space, and the following definitions are presented.

Definition 6. When object x belongs to X $(x \in X)$, we make λ_{PP}, λ_{NP}, and λ_{BP} express the cost function of dividing an object into $\mathrm{POS}_{\alpha}^{\beta}(X)$, $\mathrm{NEG}_{\alpha}^{\beta}(X)$, and $\mathrm{BND}_{\alpha}^{\beta}(X)$, respectively. And when $x \notin X$, we make λ_{PP}, λ_{NP}, and λ_{BP} express the cost function of dividing an object into $\mathrm{POS}_{\alpha}^{\beta}(\overline{X})$, $\mathrm{NEG}_{\alpha}^{\beta}(\overline{X})$, and $\mathrm{BND}_{\alpha}^{\beta}(\overline{X})$, respectively.

We suppose $\zeta = \{a_P, a_B, a_N\}$, in which a_P, a_B, and a_N, respectively, express the possible states that the current object belongs to a particular attribute set of three regions. Under different states, the risk cost [22–24] of object x taking different division plan is shown in Table 1.

Generally, the cost function meeting the conditions of $\lambda_{\mathrm{PP}} \le \lambda_{\mathrm{BP}} < \lambda_{\mathrm{NP}}$ and $\lambda_{\mathrm{NN}} \le \lambda_{\mathrm{BN}} < \lambda_{\mathrm{PN}}$, is explained as follows: for an object x belonging to X, the risk cost of dividing it into $\mathrm{POS}_{\alpha}^{\beta}(X)$ is not more than that of dividing it into $\mathrm{BND}_{\alpha}^{\beta}(X)$, and at the same time, the risk costs of the both are less than that of dividing it into $\mathrm{NEG}_{\alpha}^{\beta}(X)$. Similarly, for an object x not belonging to X, the risk cost of dividing it into $\mathrm{NEG}_{\alpha}^{\beta}(\overline{X})$ is not more than that of dividing it into $\mathrm{BND}_{\alpha}^{\beta}(\overline{X})$, and simultaneously, the risk costs of the both are less than that of dividing it into $\mathrm{POS}_{\alpha}^{\beta}(\overline{X})$.

Definition 7. The expectation loss functions [25, 26] of taking a_P, a_B, a_N decision action are expressed by the following equations, respectively:

$$\begin{aligned}
R(a_P) &= \lambda_{\mathrm{PP}}\left(\frac{\left|X \cap M^{\beta}\right|}{\left|M^{\beta}\right|}\right) + \lambda_{\mathrm{PN}}\left(\frac{\left|\overline{X} \cap M^{\beta}\right|}{\left|M^{\beta}\right|}\right),\\
R(a_B) &= \lambda_{\mathrm{BP}}\left(\frac{\left|X \cap \left(M^{\alpha} - M^{\beta}\right)\right|}{\left|M^{\alpha} - M^{\beta}\right|}\right) + \lambda_{\mathrm{BN}}\left(\frac{\left|\overline{X} \cap \left(M^{\alpha} - M^{\beta}\right)\right|}{\left|M^{\alpha} - M^{\beta}\right|}\right),\\
R(a_N) &= \lambda_{\mathrm{NP}}\left(\frac{\left|X \cap \left(U - M^{\alpha}\right)\right|}{\left|U - M^{\alpha}\right|}\right) + \lambda_{\mathrm{BN}}\left(\frac{\left|\overline{X} \cap \left(U - M^{\alpha}\right)\right|}{\left|U - M^{\alpha}\right|}\right).
\end{aligned} \tag{7}$$

Table 1: Risk cost in different decision-making plans and states.

	a_P	a_B	a_N
X	λ_{PP}	λ_{BP}	λ_{NP}
X^c	λ_{PN}	λ_{BN}	λ_{NN}

Definition 8. Suppose interval concept lattice $L_\alpha^\beta(U, C \cup D, R)$ is determined by the formal context $(U, C \cup D, R)$. $C = (M^\alpha, M^\beta, Y)$ is one interval concept in the lattice structure. We call $\widetilde{C} = (M^\alpha, M^\beta, Y; R(a_P), R(a_B), R(a_N))$ interval three-way decision concept.

Definition 9. Given the formal context $(U, C \cup D, R)$, $\widetilde{L}_\alpha^\beta(U, C \cup D, R)$ is an interval three-way decision space composed by interval three-way decision concepts and parent-children relationships between concepts.

For a new object x, the decision rules obtained by interval three-way decision concept are composed by decision-making action a_i (i can express acceptance action P, noncommitment action B, or rejection action N) and corresponding decision loss R. We denote the decision rules as $J = (a_i, R(a_i))$.

The process of generating interval concept lattice by the formal context is essentially a process of clustering concepts. Moreover, the inclusion relation between the intension of interval concepts determines the father-children relationship in the lattice structure. In our paper, the interval concept lattice generated by the prior formal context constitutes the interval three-way decision space. The divided three decision regions by interval concept in the lattice can be regarded as decision-making rules of a new object, and the parent-children relationships in decision space can decide the next action to reduce the loss of decision-making.

3. Interval Parameter Optimization under Three-Way Decision Space

3.1. Decision Optimization Algorithm under Given Interval Parameters. In the decision-making space, an object x can make different decisions according to multiple interval three-way decision concepts. If we obtain the noncommitment decision rules J, for adventurists, they are more likely to make acceptance or rejection decision J' even though which has a relatively small loss. Here, we suppose the corresponding interval three-way decision concept of J' is $\widetilde{C}'$. When decision-makers take acceptance or rejection decision which has relatively small loss, we can reduce the loss of acceptance or rejection decision according to decision regions divided by the subconcepts of $\widetilde{C}'$. First, we give a decision optimization algorithm [27] as follows (Algorithm 1).

3.2. Three-Way Decision Space Updating with Changing Interval Parameters. The algorithm of finding decision rules of object x based on the fixed interval parameters is given in the previous section, and the updating algorithm of three-way decision space with changing parameters is given in this section. Considering that the interval parameters α and β can change from $[\alpha_0, \beta_0]$ to $[\alpha_1, \beta_1]$ and the relationship between $\alpha_0(\beta_0)$ and $\alpha_1(\beta_1)$ is indeterminate, therefore there are four kinds of cases to explain the problem of updating the interval three-way decision space. Moreover, the change of interval parameters α and β can first lead to the change of extension of interval three-way decision concepts.

Proposition 3. *When* $\alpha_1 < \alpha_0$, $\beta_1 < \beta_0$, $M^{\alpha_1} \supseteq M^{\alpha_0}$ *and* $M^{\beta_1} \supseteq M^{\beta_0}$.

Proof. Given $M^{\alpha_0} = \{x | x \in M, (|f(x) \cap Y|/|Y|) \geq \alpha_0 > \alpha_1\}$, $M^{\beta_0} = \{x | x \in M (|f(x) \cap Y|/|Y|) \geq \beta_0 > \beta_1\}$, $M^{\alpha_1} = M^{\alpha_0} \cup \{x_1\}$, where $\{x_1 | \alpha \leq (|f(x_1) \cap Y|/|Y|) \leq \alpha_1\}$. Similarly, $M^{\beta_1} = M^{\beta_0} \cup \{x_{11}\}$, where $\{x_{11} | \beta_1 \leq (|f(x_{11}) \cap Y|/|Y|) \leq \beta_0\}$; obviously, $M^{\alpha_1} \supseteq M^{\alpha_0}$ and $M^{\beta_1} \supseteq M^{\beta_0}$. □

Proposition 4. *When* $\alpha_1 > \alpha_0$ *and* $\beta_1 > \beta_0$, $M^{\alpha_1} \subseteq M^{\alpha_0}$ *and* $M^{\beta_1} \subseteq M^{\beta_0}$.

Proof. Given $M^{\alpha_0} = \{x | x \in M, \alpha_1 > (|f(x) \cap Y|/|Y|) \geq \alpha_0\}$, $M^{\beta_0} = \{x | x \in M, \beta_1 > (|f(x) \cap Y|/|Y|) \geq \beta_0\}$, $M^{\alpha_1} = M^{\alpha_0} - \{x_1\}$, where $\{x_1 | \alpha \leq (|f(x_1) \cap Y|/|Y|) \leq \alpha_1\}$. Similarly, $M^{\beta_1} = M^{\beta_0} - \{x_{11}\}$, where $\{x_{11} | \beta_0 \leq (|f(x_{11}) \cap Y|/|Y|) \leq \beta_1\}$; obviously, $M^{\alpha_1} \subseteq M^{\alpha_0}$ and $M^{\beta_1} \subseteq M^{\beta_0}$.

We assume interval parameters change into $[\alpha_1, \beta_1]$ from $[\alpha_0, \beta_0]$ and there are four kinds of cases: (i) $\alpha_1 > \alpha_0$, (ii) $\alpha_1 > \alpha_0$, (iii) $\beta_1 > \beta_0$, and (iv) $\beta_1 > \beta_0$. The first two cases imply to update the upper extension of the interval three-way decision concepts, namely, $M^{\alpha_0} \longrightarrow M^{\alpha_1}$; the other two cases mean to update the lower extension of the interval three-way decision concepts, namely, $M^{\beta_0} \longrightarrow M^{\beta_1}$. Therefore, the following four functions are given, respectively, to update the interval three-way decision concepts:

(i) Function: DCL1 $(\widetilde{C}, \alpha_0, \alpha_1)//\widetilde{C}$ is any node in interval three-way decision concept lattice, and $\alpha_1 > \alpha_0$

```
DCL1 (C̃, α0, α1)
{
    Ma = {φ}
For each x in M^α0 of C̃:
    If (|f(x) ∩ Y|/|Y|) ≥ α1, then
        Ma = Ma ∪ x
    M^α1 = Ma
The corresponding decision loss functions R^β0_α0(a_B)
and R^β0_α0(a_N) are changed into R^β0_α1(a_B) and
R^β0_α1(a_N)
}
```

(ii) Function: DCL2 $(\widetilde{C}, \alpha_0, \alpha_1)//\widetilde{C}$ is any node in interval three-way decision concept lattice, and $\alpha_1 > \alpha_0$

```
DCL2 (C̃, α0, α1)
{
    Ma = M^α0
    For the upper extension Maf of any father
node C̃F in C̃:
{
Make maf1 = Maf − M^α0
For ∀x ∈ maf1:
```

Input: decision formal context $(U, C \cup D, R)$;
Interval parameters $[\alpha_0, \beta_0]$;
Object x and its condition attribute set $A(A \subseteq C)$;
Output: the decision rules of object x.
Step 1: according to the given formal context $(U, C \cup D, R)$, interval three-way decision space $\widetilde{L_{\alpha_0}^{\beta_0}}(U, C \cup D, R)$ will be built by interval three-way decision concepts:

$\widetilde{C} = (M^{\alpha_0}, M^{\beta_0}, Y; R(a_B), R(a_N))$ and parent-children relationship (Definitions 7–9);

Step 2: find intension Y in interval three-way decision concepts $\widetilde{C}$ if $Y - (Y \cap D) = A$, turn to Step 3; else $Y - (Y \cap D) \neq A$ and $(Y - (Y \cap D)) \cap A \neq \phi$, and turn to Step 6;
Step 3: if there are n concepts like $\widetilde{C} = (M^{\alpha_0}, M^{\beta_0}, A; R(a_P), R(a_B), R(a_N))$, n decision rules will be obtained, namely, $J_1, J_2, J_3, \ldots, J_n$, and they can constitute n-dimensional decision space, $JS = \{J_1, J_2, J_3, \ldots, J_n\}$;
Step 4: if there is only J_k meeting $R_{Pk} = \min(R_{P1}, R_{P2}, \ldots, R_{Pn})$, J_k will become the final decision rules of x, namely, making the accept or reject decision of the smallest loss as the final decision;
Step 5: if there are some J_i, J_m meeting $R_{Pi} = R_{Pm} = \min(R_{P1}, R_{P2}, \ldots, R_{Pn})$, search the subconcepts of $\widetilde{C}_i$ and $\widetilde{C}_m$. And on the basis of intension of those subconcepts, add related attribute of x, until get the only decision meeting condition;
Step 6: according to the parent-children relationship in the three-way decision space, search concepts whose intension is Y'. Y' meets the conditions of $A \subseteq Y'$ and $|Y' - A| = 1$, and turn to Step 3; else turn to Step 7;
Step 7: search the concepts whose intension is Y'', Y'' meets the conditions of $A \subseteq Y''$ and $|Y'' - A| = 2$ and turn to Step 3 to continue making decision, until the end node whose intension is $\varnothing$.
End.

Algorithm 1: Decision optimization algorithm under given interval parameters (GPOA).

If $(|f(x) \cap Y|/|Y|) \geq \alpha_1$, then// Y is the intension set of $\widetilde{C}$
Ma = MaUx
}
M^{α_1} = Ma
The corresponding decision loss functions $R_{\alpha_0}^{\beta_0}(\alpha_B)$ and $R_{\alpha_0}^{\beta_0}(\alpha_N)$ are changed into $R_{\alpha_0}^{\beta_0}(\alpha_B)$ and $R_{\alpha_0}^{\beta_0}(\alpha_N)$
}

(iii) Function: DCL3 $(\widetilde{C}, \beta_0, \beta_1)//\widetilde{C}$ is any node in interval three-way decision concept lattice, and $\beta_1 > \beta_0$

DCL3 $(\widetilde{C}, \beta_0, \beta_1)$
{
Mb = $\{\varnothing\}$
For each x in M^{β_0} of $\widetilde{C}$:
If $(|f(x) \cap Y|/|Y|) \geq \beta_1$, then
Mb = MbUx
M^{β_1} = Mb
The corresponding decision loss functions $R_{\alpha_0}^{\beta_0}(\alpha_P)$ and $R_{\alpha_0}^{\beta_0}(\alpha_B)$ are changed into $R_{\alpha_0}^{\beta_0}(\alpha_P)$ and $R_{\alpha_0}^{\beta_0}(\alpha_B)$
}

(iv) Function: DCL4 $(\widetilde{C}, \beta_0, \beta_1)//\widetilde{C}$ is any node in interval three-way decision concept lattice, and $\beta_0 < \beta_1$

DCL4 $(\widetilde{C}, \beta_0, \beta_1)$
{
Mb = M^{β_0}
For the upper extension Mbf of any father node $\widetilde{C}F$ in $\widetilde{C}$:
{
Make mbf1 = Mbf − M^{β_0}
For $\forall x \in$ mbf1:
If $(|f(x) \cap Y|/|Y|) \geq \beta_1$, then// Y is the intension set of $\widetilde{C}$
Mb = MbUx
}
M^{β_1} = Mb
The corresponding decision loss functions $R_{\alpha_0}^{\beta_0}(\alpha_P)$ and $R_{\alpha_0}^{\beta_0}(\alpha_B)$ are changed into $R_{\alpha_0}^{\beta_0}(\alpha_P)$ and $R_{\alpha_0}^{\beta_0}(\alpha_B)$
}

Based on the four functions, when the interval parameters change, we use the method of breadth-first to visit and judge each node from the root node in the interval three-way decision space. According to the four different cases, we can update and adjust the nodes; meanwhile, delete the redundancy concepts and empty concepts from the space structure (Algorithm 2).

On the basis of the original interval three-way decision space, when interval parameters change, the extension and decision loss function of local nodes will correspondingly change in the space. The updating algorithm can help keep or update the extension and decision loss function of each node in the original space. Finally, the new interval three-way decision space is obtained. Compared with reconstruction, the updating algorithm is superior to reconstruction in the aspects of time complexity. □

3.3. Interval Parameter Optimization in Three-Way Decision Space. Through the introduction from the previous two sections, we have mastered the interval three-way decision space updating algorithm. However, the problem of interval parameters (α and β) choice has not been solved yet. It is also an important role to play on decision-making, and the optimal parameters can bring more potential information. Therefore, we will introduce the process of interval parameter optimization as follows.

Input: decision formal context $(U, C \cup D, R)\Delta$;
$\widetilde{L_{\alpha_0}^{\beta_0}}(U, C \cup D, R)$
Interval parameters (α_1, β_1);
Output: $\widetilde{L_{\alpha_1}^{\beta_1}}(U, C \cup D, R)$.
Step 1: $\widetilde{C}_1 = (M^{\alpha_0}, M^{\beta_0}, Y, R_{\alpha_0}^{\beta_0}(a_P), R_{\alpha_0}^{\beta_0}(a_B), R_{\alpha_0}^{\beta_0}(a_N))$ is the root node of $\widetilde{L_{\alpha_0}^{\beta_0}}(U, C \cup D, R)$. If $Y = \varnothing, \widetilde{C}_1$ does not change; if $Y = \varnothing$ and $\alpha_1 > \alpha_0$, call function: DCL1 $(\widetilde{C}, \alpha_0, \alpha_1)$, else call function: DCL2 $(\widetilde{C}, \alpha_0, \alpha_1)$, then update M^{α_0} to M^{α_1}, $R_{\alpha_0}^{\beta_0}(a_B)$ to $R_{\alpha_1}^{\beta_0}(a_B)$, and $R_{\alpha_0}^{\beta_0}(a_N)$ to $R_{\alpha_0}^{\beta_0}(a_N)$; as the same, if $\beta_1 > \beta_0$, call function: DCL3 $(\widetilde{C}, \beta_0, \beta_1)$, else call function: DCL4 $(\widetilde{C}, \beta_0, \beta_1)$, then update M^{β_0} to M^{β_1}, $R_{\alpha_0}^{\beta_0}(a_P)$ to $R_{\alpha_0}^{\beta_1}(a_P)$, and $R_{\alpha_0}^{\beta_0}(a_B)$ to $R_{\alpha_0}^{\beta_1}(a_B)$. And $\widetilde{C_1}$ is totally updated to $(M^{\alpha_1}, M^{\beta_1}, Y, R_{\alpha_1}^{\beta_1}(a_P), R_{\alpha_1}^{\beta_1}(a_B), R_{\alpha_1}^{\beta_1}(a_N))$;
Step 2: visit each children nodes $\widetilde{C}_i$ in $\widetilde{C}_1$;
Step 3: suppose$\widetilde{C}_i = (M_i^{\alpha_0}, M_i^{\beta_0}, Y_i R_{\alpha_0}^{'\beta_0}(a_P), R_{\alpha_0}^{'\beta_0}(a_B), R_{\alpha_0}^{'\beta_0}(a_N))$. If $\alpha_1 > \alpha_2$, call function: DCL1 $(\widetilde{C}, \alpha_0, \alpha_1)$, else call function: DCL2 $(\widetilde{C}, \alpha_0, \alpha_1)$, then update $M_i^{\alpha_0}$ to $M_i^{\alpha_1}$, $R_{\alpha_0}^{'\beta_0}(a_B)$ to $R_{\alpha_1}^{'\beta_0}(a_B)$, and $R_{\alpha_1}^{'\beta_0}(a_B)$ to $R_{\alpha_1}^{'\beta_0}(a_N)$; if $M_i^{\alpha_1} = \varnothing$, delete node $\widetilde{C}_i$; otherwise, continue updating the lower extension: if $\beta_1 > \beta_0$, call function: DCL3 $(\widetilde{C}, \beta_0, \beta_1)$, else call function: DCL4 $(\widetilde{C}, \beta_0, \beta_1)$, then update $M_i^{\beta_0}$ to $M_i^{\beta_1}$, $R_{\alpha_0}^{'\beta_0}(a_P)$ to $R_{\alpha_0}^{'\beta_1}(a_P)$, and $R_{\alpha_0}^{'\beta_0}(a_B)$ to $R_{\alpha_0}^{'\beta_1}(a_B)$, and $\widetilde{C}_i$ is totally updated to $(M_i^{\alpha_1}, M_i^{\beta_1}, Y_i, R_{\alpha_1}^{'\beta_1}(a_P), R_{\alpha_1}^{'\beta_1}(a_B), R_{\alpha_1}^{'\beta_1}(a_N))$;
Step 4: for each father node $\widetilde{C}_i' = \widetilde{C}_i \longrightarrow$ Parent in $\widetilde{C}_i$, and $\widetilde{C}_i'(M_i^{\alpha_1}, M_i^{\beta_1}, Y_i, R_{\alpha_1}^{'\beta_1}(a_P), R_{\alpha_1}^{'\beta_1}(a_B), R_{\alpha_1}^{'\beta_1}(a_N))$, if $M_i^{'\alpha_1} = M_i^{\alpha_1}, M_i^{'\beta_1} = M_i^{\beta_1}$, $R_{\alpha_1}^{'\beta_1}(a_P) = R_{\alpha_1}^{'\beta_1}(a_P)$, $R_{\alpha_1}^{'\beta_1}(a_B) = R_{\alpha_1}^{'\beta_1}(a_B)$, and $R_{\alpha_1}^{'\beta_1}(a_N) = R_{\alpha_1}^{'\beta_1}(a_N)$, $\widetilde{C}_i \longrightarrow$ Parent $= \widetilde{C}_i' \longrightarrow$ parent, namely delete $\widetilde{C}_i'$;
Step 5: for each children node of $\widetilde{C}_i, \widetilde{C}_i = \widetilde{C}_i \longrightarrow$ Childrenren, turn to Step3, until visiting the final node in $\widetilde{L_{\alpha_0}^{\beta_0}}(U, C \cup D, R)$;
Step 6: output $\widetilde{L_{\alpha_1}^{\beta_1}}(U, C \cup D, R)$;
End.

Algorithm 2: Interval three-way decision space updating algorithm based on changing parameters (SPDA).

3.3.1. The Basic Idea. According to the given decision formal context, decision rules of various parameters will be obtained through this algorithm. First, we determine the number n of attributes in the formal context and divide α and β, respectively, by the equal step length. Here, we suppose the step length $\lambda = 1/n$, and then α_i is i/n, $(i = 1, 2, 3, \ldots, n)$. Due to having preliminarily researched [3] the values of α, we found that when α is in the median (roughly 0.5), the stability of the lattice structure can be ensured. So first we initialize $\alpha_0 = 1/2$ and $\beta_0 = 1$, build the interval three-way decision space, and then further mine decision rules. When interval parameters change by the equal step, we update the original space and obtain the new concepts, lattice structure, and decision rules. Finally, the optimal decision rules of object x and the best interval parameters are found under those decision rules.

3.3.2. Algorithm Design. The original three-way decision space obtains constantly updating with the respect of interval parameters. Both concept and lattice structure will change, and further, the decision rules of object x will be influenced. Remarkably, the changing of these decision rules is not sudden, but gradually guides to make clear decisions of acceptance or rejection. Therefore, adventurist can make a clear decision, but not noncommitment, which not only saves the cost of time, but also rationally improves the decision-making efficiency. When users make relatively accurate or pleasant decisions, at this time, the values of interval parameters can be considered to be the best parameters in this formal context.

4. Example Analysis

For the ease of understanding and exposition, we set parameter β value to 1 and explore the effect of α on decision rules. We choose ten objects to demonstrate the above algorithms and receive the meaningful decision rules. The following example is about "Green Products Recommendation." Some attributes describing the feature of green products, such as condition attribute set $\{a, b, c, d\}$, where "a" expresses "green packing," "b" expresses "green technology," "c" expresses "green raw materials," and "d" expresses "environmental certification of manufacturer"; decision attributes are "e" and "f," which express that some green products can be recommended to "consumer e" and "consumer f." When a green product has a green package, but without green technology during the production, made of green raw materials and by environmental certification holder, its corresponding condition attribute set is $\{1, 0, 1, 1\}$. Whereafter, the decision formal context is shown in Table 2.

Through simply preprocessing this formal context, we unify the representation of decision attributes and condition attributes. For example, if a green product is of the condition attributes of $\{1, 0, 1, 1\}$, consumer e will not consider purchasing this green product, while consumer f would like to purchase it as object 1 showing. The converted form context is shown in Table 3.

4.1. Model Verification. According to the formal context (U, A, R) as shown in Table 3, there are 6 attributes (A), including 4 condition attributes and 2 decision attributes, and 10 objects (U). We assume the new object x is a "green packing" product, and we aim at recommending the new

Table 2: Form context with decision.

Object	a	b	c	d	Decision
1	1	0	1	1	F
2	0	1	0	0	Ef
3	0	0	1	0	NA
4	0	1	0	1	Ef
5	1	1	1	0	Ef
6	1	0	0	1	ef
7	1	0	1	1	ef
8	0	0	1	1	f
9	1	1	1	0	ef
10	0	1	0	1	f

Table 3: Converted form context.

Object	a	b	c	d	e	f
1	1	0	1	1	0	1
2	0	1	0	0	1	1
3	0	0	1	0	0	0
4	0	1	0	1	1	1
5	1	1	1	0	1	1
6	1	0	0	1	1	1
7	1	0	1	1	1	1
8	0	0	1	1	0	1
9	1	1	1	0	1	1
10	0	1	0	1	0	1

Input: decision formal context $(U, C \cup D, R)$;
Object x and condition attribute set $A(A \subseteq C)$;
Output: the decision rules of x;
The interval parameters under optimal decision.
Step 1: determine the number n of attributes in the formal context, and set the step length $\lambda = 1/n$;
Step 2: initialize $\alpha(\alpha_1, \beta)(\alpha_1, \beta)$, $\beta = 1$, and build the interval three-way decision space $\widetilde{L}_{\alpha}^{\beta}(U, C \cup D, R)$;
Step 3: put $\widetilde{L}_{\alpha}^{\beta}(U, C \cup D, R)$ into Algorithm 1 (GPOA);
Step 4: in the output of Algorithm 1 (GPOA), if the accept loss is 0 and the reject loss is 0, turn to Step 5;
Step 5: make $\alpha = \alpha + \lambda\beta = \beta - \lambda$, and update three-way decision space according to Algorithm 2 (SPDA);
Step 6: turn to Steps 3 and 4;
Step 7: compare these decision rules of x, and output the optimal decision and interval parameters.
End.

Algorithm 3: Interval parameter optimization algorithm in three-way decision space (IPOA).

Table 4: Three-way decision concept by $\alpha = 3/6$ and $\beta = 1$

Concept $\widetilde{C}$	Upper extension M^{α}	Lower extension M^{β}	Intension Y	Accept loss $R(a_P)$	Noncommitment loss $R(a_B)$	Reject loss $R(a_N)$
$\widetilde{C}_1$	{12345679}	{5679}	Ae	0	5.5	0
$\widetilde{C}_2$	U	{15679}	Af	0	7.6	0
$\widetilde{C}_3$	{245679}	{59}	abe	0	9	0
$\widetilde{C}_4$	{135679}	{579}	ace	0	4.33	7.5
$\widetilde{C}_5$	{1356789}	{1579}	acf	0	6.67	15
$\widetilde{C}_6$	{1345679}	{67}	ade	0	6.2	5
$\widetilde{C}_7$	{1345678910}	{167}	adf	0	7.67	15
$\widetilde{C}_8$	{123456789}	{59}	$abce$	0	6	0
$\widetilde{C}_9$	U	{59}	$abcf$	0	8.125	0
$\widetilde{C}_{10}$	{1234567910}	ϕ	$abde$	0	6.67	0
$\widetilde{C}_{11}$	{1345678910}	{17}	$acdf$	0	8	15
$\widetilde{C}_{12}$	{13456789}	ϕ	$abcde$	0	6.375	7.5
$\widetilde{C}_{13}$	{1345678910}	ϕ	$abcdf$	0	8.22	15

product to the potential consumers (e and f). On the one hand, our model avoids the waste of information resources by pushing the product information to partial consumers instead of all consumers, and on the other hand, it causes less customer churn than the classical model which is only for precise consumers. Set $\lambda_{PP} = 0, \lambda_{BP} = 9, \lambda_{NP} = 15, \lambda_{PN} = 17, \lambda_{BN} = 2$, and $\lambda_{NN} = 0$. According to the previous three steps of Algorithm 3, we can obtain the initial interval three-way decision concepts as shown in Table 4.

Due to the new object x with "a" condition attribute, we only need observing the concepts that contain condition attribute "a" and the rest of interval three-way decision concepts will be omitted on account of length limits.

First, we build the lattice structure as Figure 1 shows based on three-way decision concepts in Table 4, where $\widetilde{C}_{\{a,b,c,d,e,f\}}$ refers to the concept with $\{a, b, c, d, e, f\}$ intension and $\varnothing$ extension, and $\widetilde{C}_{\varnothing}$ represents the concept with $\varnothing$ intension and {12345678910} extension. Through the lattice structure, we can obviously find the parent-child relationships between concepts. For example, $\widetilde{C}_{12} \longrightarrow \widetilde{C}_8 \longrightarrow \widetilde{C}_3$; their intensions have the relationship of inclusion. The smaller the intension is (e.g., $\widetilde{C}_3$), the higher the possibility of loss of noncommitment is. Not until we pointed out $\widetilde{C}_{12}$ did the acceptance decision obtained.

From the perspective of decision-making, when $\alpha = 3/6$ and $\beta = 1$, according to the decision loss of $\widetilde{C}_1$ and $\widetilde{C}_2$, we can see that noncommitment will bring about a certain loss. However, it will be a loss to make the decision of acceptance or rejection, corresponding to step 4 of Algorithm 3. Actually, these decision rules in this case are not meaningful because they still do not send a clear message of acceptance or rejection. For this new "green packing" product, we could not recommend it to consumer e or f yet. Next, we run step 5 of Algorithm 3 and will obtain the result as Table 5 shows.

In addition, to obtain the helpful decision-making in the case $\alpha = 3/6$ and $\beta = 1$, we could add condition attributes to the object x which may be given over a period of observing. For example, when adding condition attribute "c" to the object x, according to the decision loss of $\widetilde{C}_4$ and $\widetilde{C}_5$ we can draw a conclusion: the loss value of rejecting to recommend the object x for "consumer e" is 7.5; the loss value of rejecting to recommend the object x for "consumer f" is 15. As a consequence, the object x who includes condition attributes $\{ac\}$ should be recommended to "consumer f." Similarly, after adding condition attribute "d," the object x should be still recommended to "consumer f." And thus when we face with a new object x which is "green packing" product, it is hard to immediately make a clear decision on "product recommendation." Instead, we need to spend some time on discovering more product information, which is beneficial to make a clear decision. It is definitely the implication of noncommitment decision.

When $\alpha = 4/6$ and $\beta = 1$, it is easy to see the number of concepts is more than that under $\alpha = 3/6$ and $\beta = 1$. The parent-child relationships between concepts are obtained in Figure 2. Similarly, we find $\widetilde{C}_{14} \longrightarrow \widetilde{C}_9 \longrightarrow \widetilde{C}_3$ and make the acceptance decision from $\widetilde{C}_9$. Compared to the case $\alpha = 3/6$ and $\beta = 1$, the efficiency of decision-making is obviously improved.

According to the decision loss of $\widetilde{C}_1$, we can find that "consumer e" accepts or rejects the new object x ("green packing" product) will not bring about any loss. Nevertheless, we make decision of noncommitment on "green packing" product will bring about a certain loss. Therefore, from $\widetilde{C}_1$, this decision rule is of no guiding significance. From $\widetilde{C}_2$, it is obviously willing for "consumer f" to accept the "green packing" product. In conclusion, when $\alpha = 4/6$ and $\beta = 1$, there is no need to add condition attribute to promote decision-making, and some practical significance decisions can be directly obtained by the three-way decision concepts. Interestingly, compared to the results under the condition $\alpha = 3/6$ and $\beta = 1$, we indicate that in the case $\alpha = 3/6$ and $\beta = 1$, the new object will be more efficiently recommended to "consumer f." Therefore, our algorithm will end. To highlight the validity of our model, we will give the case $\alpha = 5/6$ and $\beta = 1$ and $\alpha = 6/6$ and $\beta = 1$ as Tables 6 and 7 show.

When $\alpha = 5/6$ and $\beta = 1$, although the three-way decision concept is more clear than before, some necessary decision concepts like whose intension is $\{abdf\}$ are missing. It is likely to cause the customer churn because the recommendation is too accurate. And the parent-child relationships between concepts are shown in Figure 3. There are four concepts mattering the condition attribute "a," namely, $\widetilde{C}_1, \widetilde{C}_2, \widetilde{C}_3$, and $\widetilde{C}_4$. And the number of concepts will start to decrease with respect to parameter α. From the perspective of optimizing parameters, we will consider $\alpha = 4/6$ as the optimal parameter of interval concept lattice. According to $\widetilde{C}_1$, we easily draw the conclusion that the "green packing" product should be recommended to "consumer f." Furthermore, when the green product is with attribute "c" or "d," the result is the same (being recommended to "consumer f"). Even although recommending the green product which is with attributes "ac" to "consumer e" will not result in loss, rejecting to recommend to "consumer f" could bring about more loss ($12.5 > 6.43$).

Similarly, when $\alpha = 6/6$ and $\beta = 1$, we can obtain the lattice structure (three-way decision space) as Figure 4 shows. There are obvious parent-child relationships, $\widetilde{C}_4 \longrightarrow \widetilde{C}_2 \longrightarrow \widetilde{C}_1$ and $\widetilde{C}_4 \longrightarrow \widetilde{C}_3 \longrightarrow \widetilde{C}_1$; meanwhile, these decision concepts all imply that the "green packing" product is definitely recommended to "consumer f." The case is the same as the classical model where the objects should completely meet the condition attributes from Y. Therefore, some potential consumers may be ignored.

4.2. Model Comparison and Instruction. To further illustrate the model, the loss value of each decision under variable parameters is given in below trend charts. Here, we assume that the new object x only is of condition attribute "a." When parameter α changes by the equal step, the trend chart of making "e-decision" (acceptance, noncommitment, or rejection of "consumer e") on the new object x is shown in Figure 5(a), where the horizontal axis shows the loss value and the vertical axis expresses the value of parameter α.

From Figure 5(a), we find that the loss value of accepting or rejecting the "green product" for "consumer e" is 0 and that of noncommitment decision is 5.5, based on the three-way decision concept whose intension set is $\{ae\}$ in the condition from $\alpha = 1/6$ to $\alpha = 4/6$. These decision rules are obviously of no practical significance. From the perspective

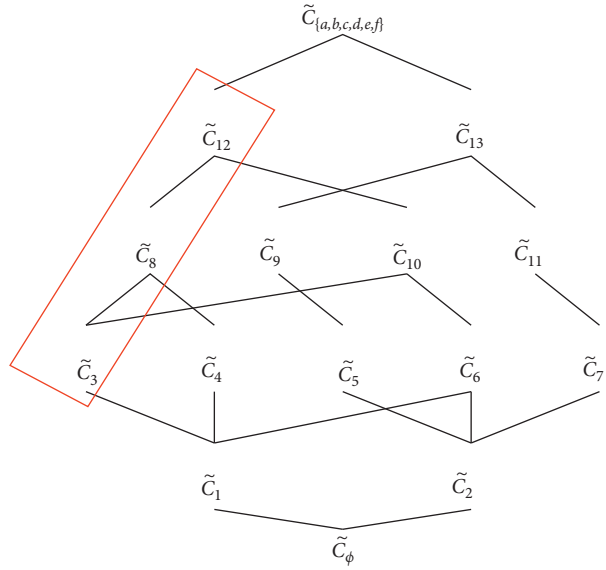


Figure 1: Lattice structure of $\alpha = 3/6$ and $\beta = 1$.

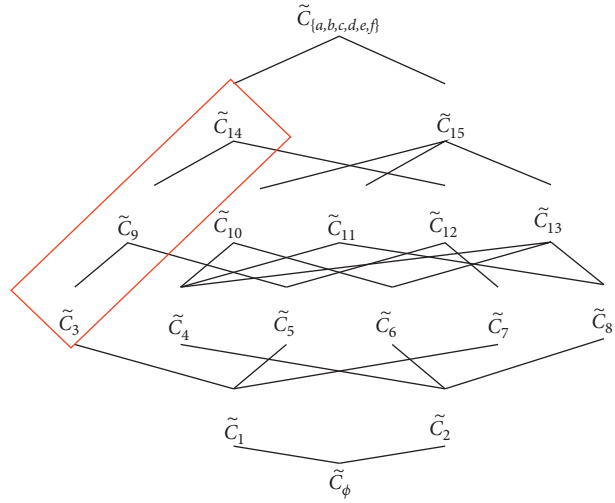


Figure 2: Lattice structure of $\alpha = 4/6$ and $\beta = 1$.

Table 5: Three-way decision concept by $\alpha = 4/6$ and $\beta = 1$.

Concept $\tilde{C}$	Upper extension M^α	Lower extension M^β	Intension Y	Accept loss $R(a_P)$	Noncommitment loss $R(a_B)$	Reject loss $R(a_N)$
$\tilde{C}_1$	{12345679}	{5679}	*ae*	0	5.5	0
$\tilde{C}_2$	{15679}	{15679}	*af*	0	0	12
$\tilde{C}_3$	{245679}	{59}	*abe*	0	9	0
$\tilde{C}_4$	{124567910}	{58}	*abf*	0	9	7.5
$\tilde{C}_5$	{135679}	{579}	*ace*	0	4.33	7.5
$\tilde{C}_6$	{1356789}	{1579}	*acf*	0	6.67	15
$\tilde{C}_7$	{1345679}	{67}	*ade*	0	6.2	5
$\tilde{C}_8$	{1345678910}	{167}	*adf*	0	7.83	15
$\tilde{C}_9$	{579}	{5}	*abce*	0	5	6.43
$\tilde{C}_{10}$	{1579}	{59}	*abcf*	0	7.83	12.5
$\tilde{C}_{11}$	{14567910}	ϕ	*abdf*	0	9	10
$\tilde{C}_{12}$	{135679}	{7}	*acde*	0	6.2	7.5
$\tilde{C}_{13}$	{1356789}	{17}	*acdf*	0	7.6	15
$\tilde{C}_{14}$	{579}	ϕ	*abcde*	0	5	6.43
$\tilde{C}_{15}$	{1579}	ϕ	*abcdf*	0	7.83	12.5

Table 6: Three-way decision concept by $\alpha = 5/6$ and $\beta = 1$.

Concept $\widetilde{C}$	Upper extension M^{α}	Lower extension M^{β}	Intension Y	Accept loss $R(a_P)$	Noncommitment loss $R(a_B)$	Reject loss $R(a_N)$
$\widetilde{C}_1$	{15679}	{15679}	*Af*	0	0	12
$\widetilde{C}_2$	{579}	{579}	*Ace*	0	0	6.43
$\widetilde{C}_3$	{1579}	{1579}	*acf*	0	0	12.5
$\widetilde{C}_4$	{167}	{167}	*adf*	0	0	12.86

Table 7: Three-way decision concept by $\alpha = 6/6$ and $\beta = 1$.

Concept $\widetilde{C}$	Upper extension M^{α}	Lower extension M^{β}	Intension Y	Accept loss $R(a_P)$	Noncommitment loss $R(a_B)$	Reject loss $R(a_N)$
$\widetilde{C}_1$	{15679}	{15679}	*af*	0	0	12
$\widetilde{C}_2$	{1579}	{1579}	*acf*	0	0	12.5
$\widetilde{C}_3$	{167}	{167}	*adf*	0	0	12.86
$\widetilde{C}_4$	{17}	{17}	*acdf*	0	0	13.125

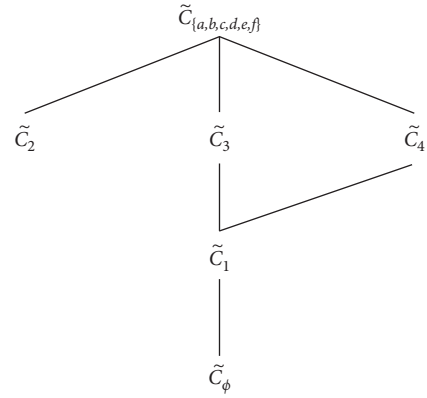


Figure 3: Lattice structure of $\alpha = 5/6$ and $\beta = 1$.

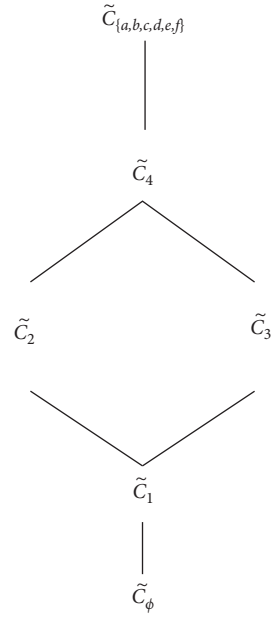


Figure 4: Lattice structure of $\alpha = 6/6$ and $\beta = 1$.

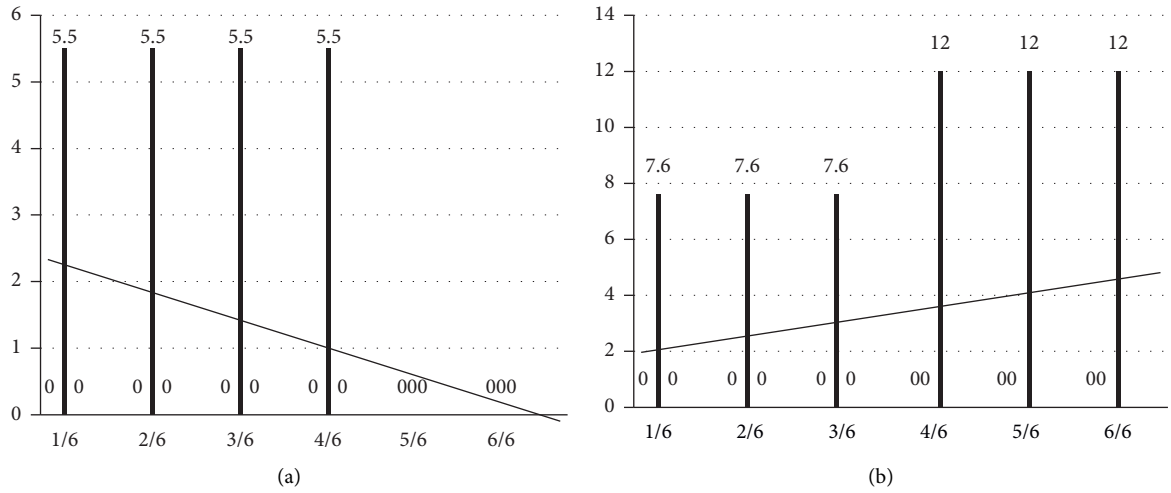


Figure 5: Trend chart of decision with changing α: (a) e-decision and (b) f-decision.

of the parameter α, when α is less than the value (roughly 0.6), it results in the slight effect of uncertain information on the decision-making. In other words, the information obtained from the condition of small α is too vague to make a clear decision of acceptance or rejection. On the intension set $\{ae\}$ side, it is not advisable for decision-maker to seek the three-way decision concept whose intension set is $\{ae\}$. According to the original form context, recommending the object x with condition attribute "a" to "consumer e" itself is ambiguous. Therefore, from Figure 5(a) we cannot obtain the meaningful implication of decision-making and the trend of line is descending.

Meanwhile, the trend chart of making "f-decision" (acceptance, noncommitment, or rejection of "consumer f") on the new object x is shown in Figure 5(b). According to Figure 5(b), when α is from 1/6 to $\alpha = 3/6$, "consumer f" still cannot make a clear decision of accepting or rejecting the object x. Nevertheless, when $\alpha = 4/6$, the decision starts to be clear and we would like to recommend the object x to "consumer f" because the loss value of rejection is 12 and acceptance is 0. When α continues increasing, we find the loss value of rejection is still 12. Obviously, we should choose "acceptance" (recommending the object x to "consumer f"). The trend of line is ascending and it means that the greater the parameter α is, the clearer the decision-making is.

From both figures, it is easy to see that the loss function of three-way decision concept has a shift at the condition of roughly $\alpha = 4/6$. In other words from this condition, the decision-making starts to be explicit, and from the perspective of parameter optimization, we could consider the parameter α at its most optimal. By the way, when we further consider adding the condition attribute of the new object x, it also contributes to the efficiency of decision-making.

5. Conclusion

The interval parameters $[\alpha, \beta]$ in the concept lattice affect the concepts and decision space generated by decision formal context. The article is mainly taking the decision loss function values as the rules of decision-making based on three-way decision space. With the change of interval parameters, different three-way decision spaces are obtained. In addition, the decision rules will be explored from the three-way decision concept in this space. It is obvious that the decision rules of the same object x are different under different interval parameters. However, there definitely exist the optimal interval parameters to make the decision rules more sufficient and clear. Until making the decision of acceptance or rejection, according to the example, we can consider that the best interval parameters (roughly more than 0.6) are obtained. The conclusion of the optimal parameters provided by our paper is the same as the best parameters previously obtained by the parameter optimization model [5] of interval concept lattice. Although there are different ideas for dealing with the problem of interval parameter optimization, finally roughly similar conclusions were drawn, which has provided a reliable basis for selecting parameter problem of interval concept lattice application. Subsequently, we will continue the study of optimization problem of interval parameters under the setting of green supply chain investment.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (nos. 71971113 and 71571100), Anhui Provincial Colleges and Universities Outstanding Talents

Cultivation Project (no. gxgnfx2019042), and Natural Science Research Project of Colleges and Universities in Anhui Province (no. KJ2019A0701).

References

[1] B. X. Liu and C. Y. Zhang, "A new concept lattice structure: interval concept lattice," *Computer Science*, vol. 39, no. 8, pp. 273–277, 2012.

[2] R. Wille, "Restructuring lattice theory: an approach based on hierarchies of concepts„" in *Ordered Sets*, I. Rival, Ed., pp. 445–470, Reidel, Dordrecht-Boston, UK, 1982.

[3] J. Yao, Y. Yao, and W. Ziarko, "Probabilistic rough sets: approximations, decision-makings, and applications," *International Journal of Approximate Reasoning*, vol. 49, no. 2, pp. 253-254, 2008.

[4] Y. Yao, "Probabilistic rough set approximations," *International Journal of Approximate Reasoning*, vol. 49, no. 2, pp. 255–271, 2008.

[5] M. X. Li, C. Y. Zhang, L. Y. Wang, and B. X. Liu, "Parameters optimization and interval concept lattice update with change of parameters," *ICIC Express Letters*, vol. 10, no. 2, pp. 339–434, 2016.

[6] Y. Y. Yao and S. K. M. Wong, "A decision theoretic framework for approximating concepts," *International Journal of Man-Machine Studies*, vol. 37, no. 6, pp. 793–809, 1992.

[7] Y. Y. Yao, "Decision-theoretical Rough set models," in *Proceedings of Rough Sets Knowledge Technology (RSKT'07)*, pp. 1–12, Toronto, Canada, May 2007.

[8] D. Liu, T. Li, and D. Ruan, "Probabilistic model criteria with decision-theoretic rough sets," *Information Sciences*, vol. 181, no. 17, pp. 3709–3722, 2011.

[9] G. L. Tang, F. Chiclana, and P. Liu, "A decision-theoretic rough set model with q-rung orthopair fuzzy information and its application in stock investment evaluation," *Applied Soft Computing Journal*, vol. 91, pp. 2–15, 2020.

[10] C. Zhang, D. Li, and J. Liang, "Multi-granularity three-way decisions with adjustable hesitant fuzzy linguistic multigranulation decision-theoretic rough sets over two universes," *Information Sciences*, vol. 507, pp. 665–683, 2020.

[11] Y. Y. Yao, "Three-way decision: an interpretation of rules in rough set theory," in *Proceeding of 4th International Conference on Rough Sets and Knowledge Technology*, pp. 642–649, Gold Coast, Australia, July 2009.

[12] Y. Y. Yao, "An outline of a theory of three-way decisions," in *Proceedings of the 8th International RSCTC Conference, LNCS (LNAI)*, pp. 1–17, Chengdu, China, August 2012.

[13] Y. Yao, "Three-way decisions with probabilistic rough sets," *Information Sciences*, vol. 180, no. 3, pp. 341–353, 2010.

[14] Y. Yao, "The superiority of three-way decisions in probabilistic rough set models," *Information Sciences*, vol. 181, no. 6, pp. 1080–1096, 2011.

[15] Y. Yao, "Three-way decisions and cognitive computing," *Cognitive Computation*, vol. 8, no. 4, pp. 543–554, 2016.

[16] T. X. Wang, H. X. Li, X. Z. Zhou, B. Huang, and H. B. Zhu, "A prospect theory-based three-way decision model," *Knowledge-Based Systems*, vol. 203, pp. 106–129, 2020.

[17] D. Liang, M. Wang, Z. Xu, and D. Liu, "Risk appetite dual hesitant fuzzy three-way decisions with TODIM," *Information Sciences*, vol. 507, pp. 585–605, 2020.

[18] C. Jiang, D. Guo, Y. Duan, and Y. Liu, "Strategy selection under entropy measures in movement-based three-way decision," *International Journal of Approximate Reasoning*, vol. 119, pp. 280–291, 2020.

[19] L. Wei, L. Liu, J. Qi, and T. Qian, "Rules acquisition of formal decision contexts based on three-way concept lattices," *Information Sciences*, vol. 516, pp. 529–544, 2020.

[20] Z. Y. Zhang, D. X. Fu, and Q. Zhou, "Optimal decisions of a green supply chain under the joint action of fairness preference and subsidy to the manufacturer," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 9610503, 18 pages, 2020.

[21] V. Albino, A. Balice, and R. M. Dangelico, "Environmental strategies and green product development: an overview on sustainability-driven companies," *Business Strategy and the Environment*, vol. 18, no. 2, pp. 83–96, 2009.

[22] H. Li, L. Zhang, X. Zhou, and B. Huang, "Cost-sensitive sequential three-way decision modeling using a deep neural network," *International Journal of Approximate Reasoning*, vol. 85, pp. 68–78, 2017.

[23] H. Li and X. Zhou, "Risk decision making based on decision-theoretic rough set: a three-way view decision model," *International Journal of Computational Intelligence Systems*, vol. 4, no. 1, pp. 1–11, 2011.

[24] H. Li, L. Zhang, B. Huang, and X. Zhou, "Sequential three-way decision and granulation for cost-sensitive face recognition," *Knowledge-Based Systems*, vol. 91, pp. 241–251, 2016.

[25] X. Y. Jia, W. Li, L. Shang et al., "An adaptive learning parameters algorithm in three-way decision-theoretic rough set model," *Acta Electronica Sinica*, vol. 39, no. 11, pp. 2520–2525, 2011.

[26] X. Y. Jia, W. Li, and L. Shang, "A simulated annealing algorithm for learning thresholds in three-way decision-theoretic rough set model," *Journal of Chinese Computer System*, vol. 34, no. 11, pp. 2604–2606, 2013.

[27] L. Y. Wang, C. Y. Zhang, and B. X. Liu, "Dynamic strategy regulation model of three-way decisions based on interval concept lattice and its application," *Computer Engineering and Applications*, vol. 52, no. 24, pp. 80–84, 2016.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 6452536, 10 pages
https://doi.org/10.1155/2020/6452536

Research Article

Text to Realistic Image Generation with Attentional Concatenation Generative Adversarial Networks

Linyan Li,[1] Yu Sun,[2] Fuyuan Hu,[2,3] Tao Zhou,[4] Xuefeng Xi,[2,5] and Jinchang Ren[6]

[1]*College of Information Technology, Suzhou Institute of Trade & Commerce, Suzhou 215009, China*
[2]*College of Electronic & Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China*
[3]*Suzhou Key Laboratory for Big Data and Information Service, Suzhou 215009, China*
[4]*School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China*
[5]*Virtual Reality Key Laboratory of Intelligent Interaction and Application Technology of Suzhou, Suzhou 215009, China*
[6]*University of Strathclyde, Glasgow, UK*

Correspondence should be addressed to Fuyuan Hu; fuyuanhu@mail.usts.edu.cn

Received 2 July 2020; Revised 25 September 2020; Accepted 6 October 2020; Published 28 October 2020

Academic Editor: Longzhuang Li

In this paper, we propose an Attentional Concatenation Generative Adversarial Network (ACGAN) aiming at generating 1024×1024 high-resolution images. First, we propose a multilevel cascade structure, for text-to-image synthesis. During training progress, we gradually add new layers and, at the same time, use the results and word vectors from the previous layer as inputs to the next layer to generate high-resolution images with photo-realistic details. Second, the deep attentional multimodal similarity model is introduced into the network, and we match word vectors with images in a common semantic space to compute a fine-grained matching loss for training the generator. In this way, we can pay attention to the fine-grained information of the word level in the semantics. Finally, the measure of diversity is added to the discriminator, which enables the generator to obtain more diverse gradient directions and improve the diversity of generated samples. The experimental results show that the inception scores of the proposed model on the CUB and Oxford-102 datasets have reached 4.48 and 4.16, improved by 2.75% and 6.42% compared to Attentional Generative Adversarial Networks (AttenGAN). The ACGAN model has a better effect on text-generated images, and the resulting image is closer to the real image.

1. Introduction

In recent years, with the rise of artificial intelligence and deep learning, natural language processing and computer vision have become the hot research fields. The text to image as a basic problem in the field has also attracted the attention and research of many scholars. Text to image is the generation of a realistic image that matches a given text description, requiring processing fuzzy and incomplete information in natural language descriptions. Text to image drives the development of multimodal learning and cross-modal generation and shows great potential in applications such as cross-modal information retrieval, photo editing, and computer-aided design.

Since Goodfellow et al. [1] proposed Generative Adversarial Networks (GANs) in 2014; the network model has received extensive attention from academia and industry. With the continuous development of GAN, it has been widely used to generate realistic high-quality images based on text descriptions. The commonly used method [2–5] encodes the entire text description into a global sentence vector, which is input to the generator as a condition variable of GAN to generate an image. However, due to the large structural differences between text and images, the use of only word-level attention does not ensure the consistency of global semantics, while it is difficult to generate complex scenes; moreover, fine-grained word information is still not explicitly used for generating images.

Therefore, the generated image does not contain enough details and is still significantly different from the real image.

To address this issue, this paper proposes Attention Cascade Generative Adversarial Networks (ACGAN). The network adopts multilevel cascade structure, the generator and discriminator in each layer are composed of convolution units, and a new network layer is added layer by layer during the training process, and the generator and discriminator are added for processing the details of the higher resolution image. At the same time, the deep attentional multimodal similarity model is introduced into the network, focusing on the fine-grained information of the word level in the semantics. The word vector is used as the input of the generator, and through the constraint of the word vector, in the case of ensuring that the overall shape of the image is unchanged, the details of the generated image are emphasized, the consistency of the image and the semantic cross-modality is maintained, and the generation process is smooth. Finally, a measure of diversity is added to a layer of the discriminator to influence the discriminator's discriminant, so that the generator can obtain more diverse gradient directions, increase the diversity of generated samples, and improve the quality of generated samples.

The contribution of our method is threefold as follows:

(i) A multilevel cascade structure is proposed, which improves the resolution of the generated image, and can generate a high-resolution image of up to 1024×1024.

(ii) Introduce the attention mechanism model into the network, and make the details of the generated image richer by paying attention to the fine-grained information of the word level in the semantics.

(iii) Add the measure of diversity to the discriminator, increase the diversity of the generated samples, and improve the quality of the generated samples.

2. Related Works

Generative image modeling is a fundamental problem in computer vision. There has been remarkable progress in this direction with the emergence of deep learning techniques. Variational Auto Encoders (VAEs) [6, 7] is aimed to maximize the lower bound of the data likelihood. Autoregressive models (e.g., PixelRNN) [8] that utilized neural networks to model the conditional distribution of the pixel space have also generated appealing synthetic images. Recently, Generative Adversarial Networks (GANs) have shown promising performance for generating sharper images and video [9–11]. For example, Eghbal-zadeh et al. [12] proposed a Mixture Density Generative Adversarial Networks to improve the clarity and quality of generated images. Gecer et al. [13] combined the generated confrontation network with a deep convolutional neural network to reconstruct a 3D facial structure from a single face image. But training instability makes it hard for GAN models to generate high-resolution images. A lot of work has been proposed to stabilize the training and improve the image quality [14–19].

Generating high-resolution images from text descriptions, though very challenging, is important for many practical applications such as art generation and computer-aided design. Lyu et al. [9] learn joint embedding to establish the relationship between natural language and real images, and then train GAN to generate 64×64 images that are conditional on text descriptions. Cao et al. [10] proposed a Stacked Generative Adversarial Networks, which decompose the complex problem of generating high-quality images into some subproblems with better control and generate 256×256 high-resolution images.

Recently, attention models have been widely used in computer vision and natural language processing, for example, object detection [20, 21], video subtitle [22], and visual question answer [23, 24]. Xu et al. [25] introduced the attention mechanism into the GAN network and proposed Attentional Generative Adversarial Networks, which instruct the generator to focus on different word-level fine-grained information when generating different image subregions. Qiao et al. [26] proposed a global-to-local collaborative attention module that uses word attention and global sentence attention to enhance the consistency of generated images and semantics.

2.1. The Proposed Model. The Attentional Concatenation Generative Adversarial Networks model proposed in this paper consists of two parts: attentional concatenation generative adversarial networks and deep attentional multimodal similarity model. As shown in Figure 1, the Attentional Concatenation Generative Adversarial Networks model is divided into multiple levels; each layer contains a generator G and a discriminator D, using a multilevel cascade structure, increasing generators and discriminators layer by layer, and continuously adds a new residual network layer during the training process, corresponding to the generation from low-resolution to high-resolution images. The Deep Attentional Multimodal Similarity Model contains a common semantic space, mapping the subregions of the image and the word vector of the sentence into one of the semantic spaces, and measuring the word-level image and text similarity. Instead of adopting a one-step approach, the entire model's training process tries to generate low-resolution images, then continuously increase the resolution, and finally generate high-resolution and high-quality images.

2.2. Concatenation Generative Adversarial Networks. The generative network has k generators $(G_0, G_1, \ldots, G_{k-1})$, which take the hidden states $(h_0, h_1, \ldots, h_{k-1})$ as input to the generator $(G_0, G_1, \ldots, G_{k-1})$, generating images of different resolutions.

Specifically,

$$\begin{aligned} h_0 &= F_0\left(z, F^{ca}\left(c_s\right)\right), \\ h_i &= F_i\left(h_{i-1}, F_i^{\text{atten}}\left(c_w, h_{i-1}\right)\right), \quad i = 1, 2, \ldots, k-1, \\ \widehat{x}_i &= G_i\left(h_i\right). \end{aligned} \tag{1}$$

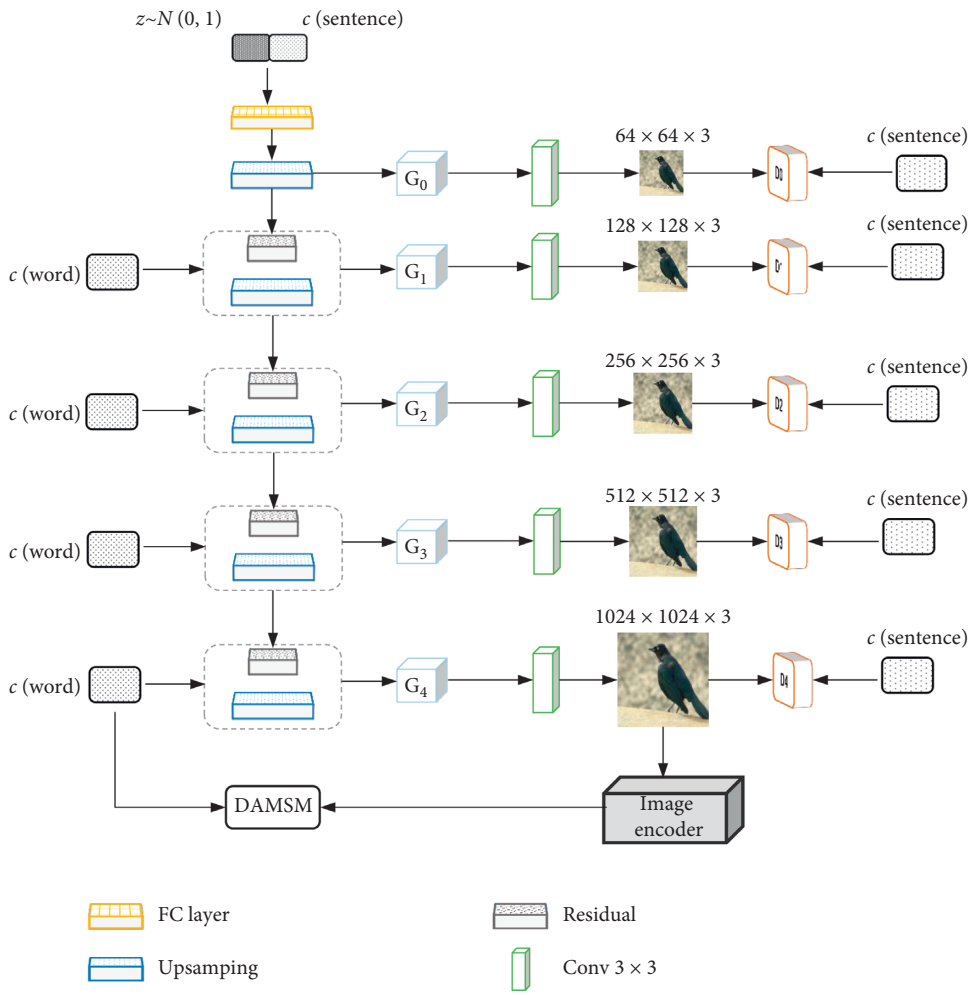


FIGURE 1: Attentional Concatenation Generative Adversarial Networks Model. Our training starts with both the generator and discriminator having a low spatial resolution of 64 × 64 pixels. As the training advances, we incrementally add layers to G and (D), thus increasing the spatial resolution of the generated images.

Here, z is a noise vector usually sampled from a standard normal distribution. c_s is a global sentence vector, and c_w is a word vector. F^{ca} represents the Conditioning Augmentation [10] that converts the sentence vector c_s to the conditioning vector. F_i^{atten} is the proposed attention model at the i^{th} stage of the attention model. The attention model $F^{\text{atten}}(c, h)$ has two inputs: the word features $c \in \mathbb{R}^{D \times T}$ and the image features $h \in \mathbb{R}^{\widehat{D} \times N}$ from the previous hidden layer.

Training starts with both the generator G and discriminator D having a low spatial resolution of 64 × 64 pixels. As the training advances, we incrementally add layers to G and D, and all existing layers remain trainable throughout the process. When doubling the resolution of the generator G and discriminator D we fade in the new layers smoothly. During the transition, we treat the layers that operate on the higher resolution like a residual block, whose weight increases linearly from 0 to 1.

Then we add a new residual layer and transform word features into semantic space of image features. Based on the hidden feature h of the image, a word context vector is calculated for each subregion of the image.

Finally, the image features and corresponding word context features are combined to generate an image in the next stage. In order to generate a real image with multiple levels (sentence level and word level) of conditions, the final objective function of the attention generation network is defined as

$$\begin{aligned} \mathscr{L} &= \mathscr{L}_G + \lambda \mathscr{L}_{\text{DAMSM}}, \\ \mathscr{L}_G &= \sum_{i=0}^{k-1} \mathscr{L}_{G_i}. \end{aligned} \tag{2}$$

Here, λ is a hyperparameter to balance the two terms of equation (2). The first term is the GAN loss that jointly approximates conditional and unconditional distributions. At the i^{th} stage of the ACGAN, the adversarial loss for is defined as

$$\mathscr{L}_{G_i} = -\frac{1}{2} E_{\widehat{x}_i \sim P_{Gi}} [\log (D_i (\widehat{x}_i))] - \frac{1}{2} E_{\widehat{x}_i \sim P_{Gi}} [\log (D_i (\widehat{x}_i, c_s))], \tag{3}$$

where the unconditional loss determines whether the image is real or fake, while the conditional loss determines whether the image and the sentence match or not.

As shown in Figure 2, for unconditional image generation, the discriminator is trained to distinguish between real images and forged images. For conditional image generation, images and variables are input to the discriminator to determine if the image matches the condition, and the bootstrap generator approximates the conditional image distribution. Discriminator D_i is trained to classify the input into the class of real or fake by minimizing the cross-entropy loss defined by

$$\begin{aligned}\mathscr{L}_{D_i} = &-\frac{1}{2}\mathbb{E}_{x_i\sim P_{\text{datai}}}[\log(D_i(x_i))]\\ &-\frac{1}{2}\mathbb{E}_{\widehat{x}_i\sim P_{Gi}}[\log(1-D_i(\widehat{x}_i))]+\\ &-\frac{1}{2}\mathbb{E}_{x_i\sim P_{\text{datai}}}[\log(D_i(x_i,c_s))]\\ &-\frac{1}{2}\mathbb{E}_{\widehat{x}_i\sim P_{Gi}}[\log(1-D_i(\widehat{x}_i,c_s))],\end{aligned} \tag{4}$$

where x_i is from the true image distribution p_{datai} at the i^{th} scale, and $\widehat{x}_i$ is from the model distribution P_{G_i} at the same scale. Discriminators D_i of the ACGAN are structurally disjoint, so they can be trained in parallel and each of them focuses on a single image scale.

2.3. Deep Attentional Multimodal Similarity Model. The Deep Attentional Multimodal Similarity Model [25] learns two neural networks that map subregions of the image and words of the sentence to a common semantic space, thus measuring the image-text similarity at the word level to compute a fine-grained loss for image generation.

This paper first uses a standard convolutional neural network to transform an image into a set of feature maps. Each feature map represents some subregions of the image. The dimension of the feature map is equal to the dimension of the word vector, and they are treated as equivalent entities. Next, based on each token in the text, attention is applied to the feature map and their weighted averages are calculated. Finally, the DAMSM is trained to minimize the difference between the attention vector and the word vector described above.

$$\mathscr{L}_1^w = -\sum_{i=1}^{k}\log P(S_i|M_i), \tag{5}$$

where "w" stands for "word".

Symmetrically, we also minimize

$$\mathscr{L}_2^w = -\sum_{i=1}^{k}\log P(M_i|S_i), \tag{6}$$

where P is the posterior probability that sentence S_i is matched with its corresponding image M_i.

Finally, the DAMSM loss is defined as

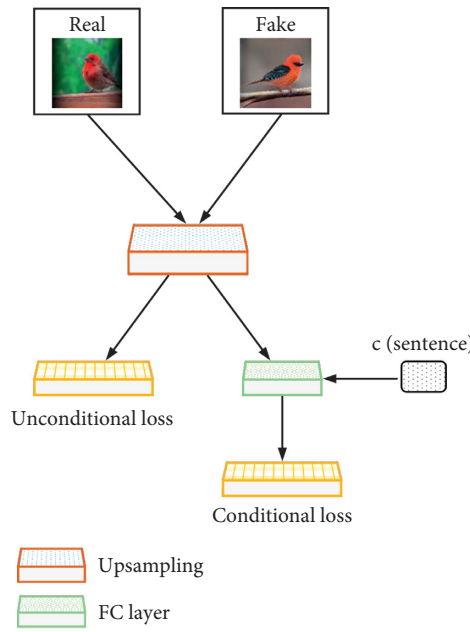


Figure 2: Discriminator model.

$$\mathscr{L}_{\text{DAMSM}} = \mathscr{L}_1^w + \mathscr{L}_2^w. \tag{7}$$

Using attention mechanism, the DAMSM is able to compute the fine-grained text-image matching loss $\mathscr{L}_{\text{DAMSM}}$. And $\mathscr{L}_{\text{DAMSM}}$ is only applied to the output of the last generator, because the ultimate goal of this paper is to generate high-resolution images through the last generator. If $\mathscr{L}_{\text{DAMSM}}$ is applied to the images generated by all generators $(G_0, G_1, ..., G_{k-1})$, the computational cost will increase greatly and the performance will not improve.

2.4. Standard Deviation of Measuring Diversity. GAN usually tends to capture only the changes found in the training data. In order to obtain more training data, this paper has greatly simplified this approach and has also improved the change based on "minibatch discrimination". Not only can feature statistics be calculated from a single image, but they can also calculate feature statistics for the entire small batch, thereby encouraging the generation of images and training images to display similar statistics. By adding a small batch layer at the end of the discriminator, the layer learns a large tensor and converts the input into a set of statistics. Finally, each instance is generated with a separate set of statistics and connected to the output of the layer so that the discriminator can use the statistics internally.

3. Experiments and Evaluation

3.1. Experimental Environment and Data. The algorithm uses the deep learning framework Tensorflow [27], and the experimental environment is Ubuntu 14.04 operating system, using four NVIDIA 1080Ti graphics processing unit (GPU) to accelerate the operation. At the same time, all models were trained on the CUB [28] and Oxford [29] datasets. As shown in Table 1, the CUB data set contains 200

Table 1: Datasets of experiment.

Dataset	CUB	CUB	Oxford	Oxford
—	Train	Test	Train	Test
Sample	8855	2933	7034	1155

Table 2: Inception scores and human rank scores for the five GAN models on the CUB and Oxford datasets.

Metric	Dataset	GAN-INT-CLS	GAWWN	StackGAN++	AttnGAN	ACGAN
Inception score	CUB	2.88 ± 0.04	3.62 ± 0.07	4.05 ± 0.05	4.36 ± 0.03	4.48 ± 0.05
	Oxford	2.66 ± 0.03	—	3.74 ± 0.03	—	3.98 ± 0.05
Human rank	CUB	2.81 ± 0.03	1.99 ± 0.04	1.37 ± 0.02	1.25 ± 0.03	1.17 ± 0.02
	Oxford	1.87 ± 0.03	—	1.13 ± 0.03	—	1.06 ± 0.02

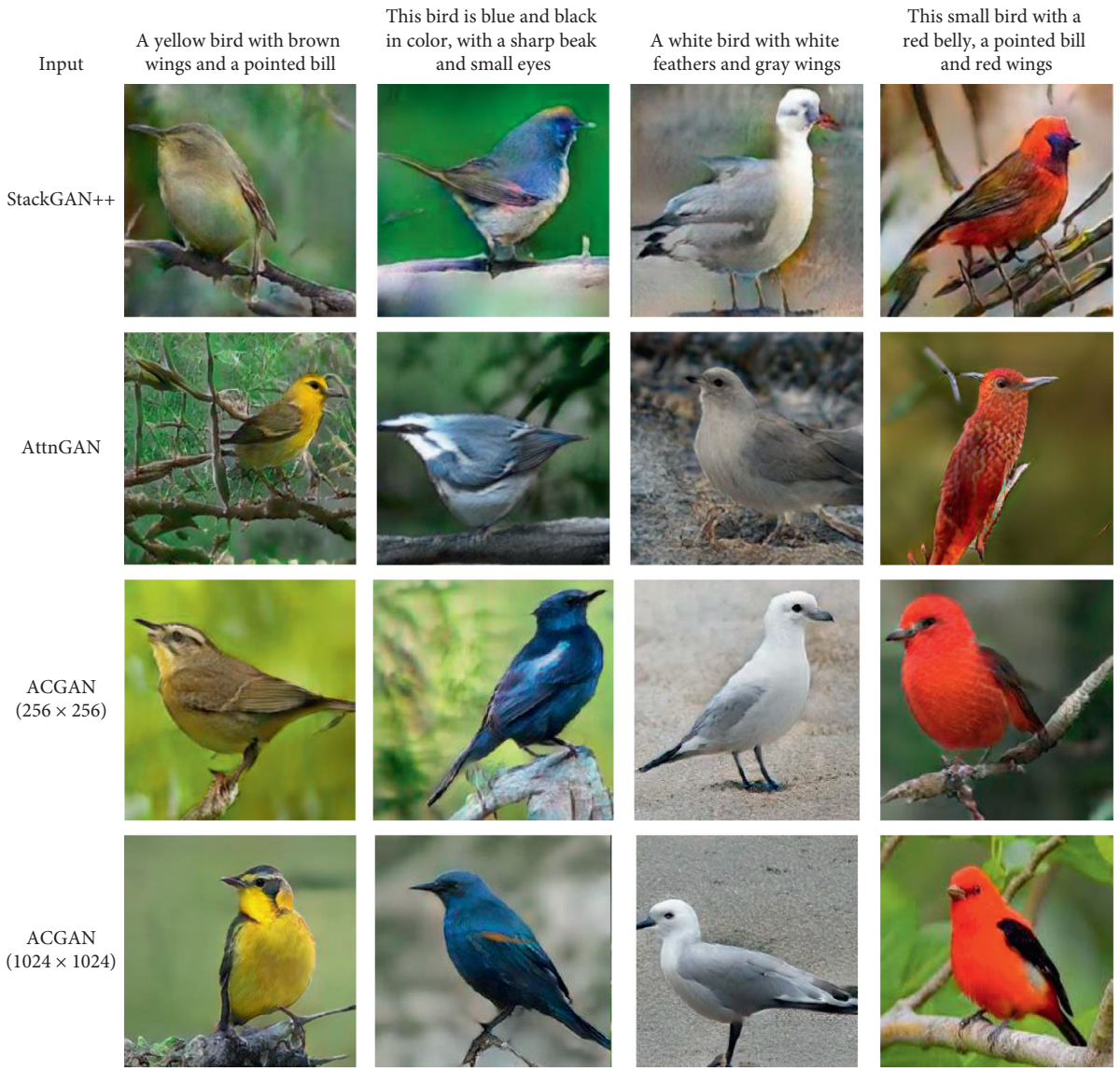


Figure 3: Images generated from descriptions using three GAN models trained on CUB test set.

species of birds with a total of 11,788 images. In this paper, 8855 images are used as training datasets and 2933 images as test datasets. Since the target area of 80% of the bird images in the dataset is less than 0.5 [28], we preprocess all images before training to ensure that the proportion of the bird target area is greater than 0.75 of the image size. The Oxford

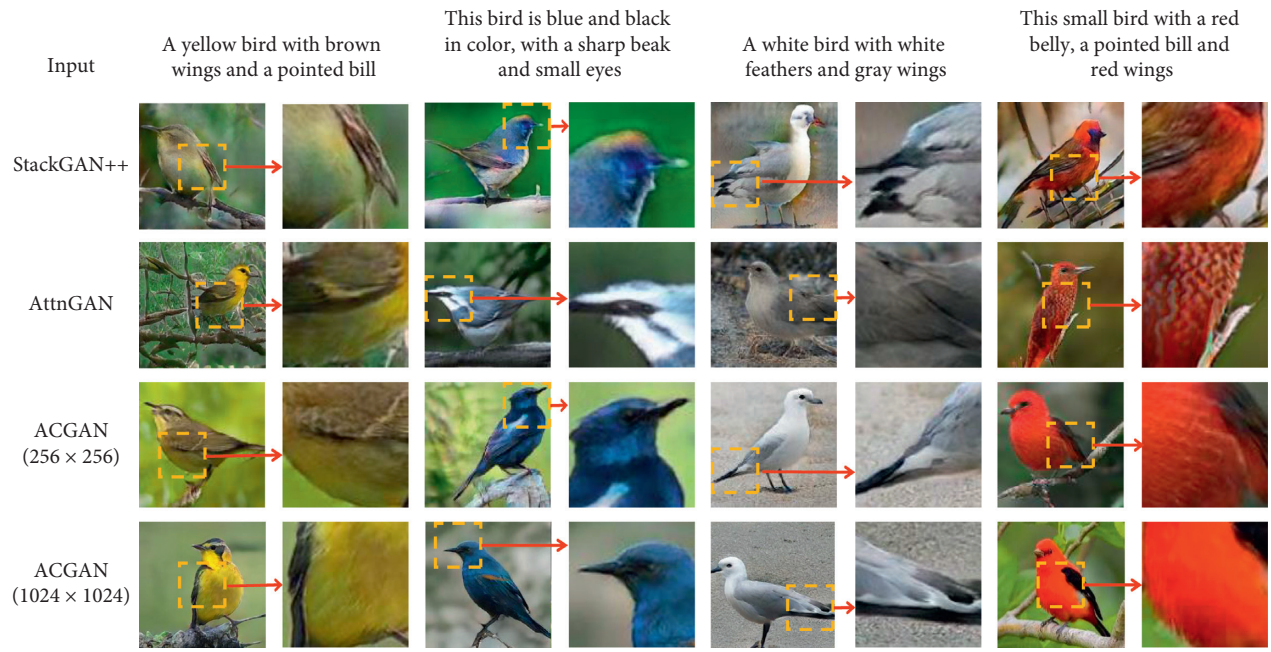


Figure 4: Details (beak, wings) comparison of the images generated from descriptions using three GAN models trained on CUB test set.

dataset contains 102 flower categories with a total of 8189 images. This article uses 7034 pictures as the training data set and 1155 pictures as the test data set.

3.2. Evaluation Metrics. For the evaluation of the GAN model, qualitative evaluation is usually used; that is, the visual fidelity of the image generated by manual inspection is required. This method is time-consuming and subjective and is somewhat misleading. Therefore, this paper mainly uses two evaluation criteria to evaluate the quality and diversity of generated images.

3.2.1. Inception Score. We choose numerical assessment approach "inception score" [16] for quantitative evaluation,

$$I = \exp\left(E_x D_{KL}\left(p\left(y \mid x\right) \| p\left(y\right)\right)\right), \tag{8}$$

where x denotes one generated sample, and y is the text label corresponding to the sample, $p(y)$ is the marginal distribution, and $p(y|x)$ is the conditional distribution. The KL divergence between the marginal distribution $p(y)$ and the conditional distribution $p(y|x)$ should be large, so that a variety of high-quality images can be generated. In the experiments, an inception model was given to the CUB data sets, and samples of each model were evaluated.

3.2.2. Human Rank. Human rank for qualitative assessment 50 text descriptions was randomly selected in the CUB and Oxford test sets, and for each sentence, the generated model generated 5 images. The five images and corresponding texts are described to different people to rank the image quality in different ways, and finally the average ranking is calculated to evaluate the quality and diversity of the generated images.

4. Experimental Result

The comparisons between the inception score and human rank results of various models on the CUB and Oxford datasets are presented in Table 2. As can be seen from the table, compared to the inception score of the AttnGAN model, the inception score of the ACGAN model on the CUB dataset has increased by 2.75% (Inception score increased from 4.36 to 4.48). Through the analysis of experimental results, ACGAN scores higher in Inception score than other GAN models; from an intuitive visual point, Human rank score is lower than other GAN models. It shows that the quality and diversity of the sample images generated by the model in this paper have been enhanced, and it is closer to the real image.

Subjective visual comparisons between the three models of StackGAN++, AttnGAN, and ACGAN on the CUB dataset are presented in Figure 3. It can be seen that the image details generated by StackGAN++ and AttnGAN are lost, colors are inconsistent with the text descriptions (1st and 2nd row), and the shape looks strange (2nd and 3rd column) for some examples. ACGAN achieved better results with more details and consistent colors and shapes compared to AttnGAN. For example, the wings are vivid in the 3rd and 4th row. By comparing ACGAN with AttnGAN, we can see that ACGAN contributes to producing fine-grained images with more details and better semantic consistency. For example, the color of the bird in the 2nd column was corrected to black. By comparing ACGAN (256 × 256) with ACGAN (1024 × 1024), we can see that the images generated by ACGAN (1024 × 1024) have higher definition, more vivid colors, and more lifelike details. Generally, content in the CUB dataset is less; therefore, it is easier to generate visually realistic and semantically consistent results on CUB. These results confirm that ACGAN

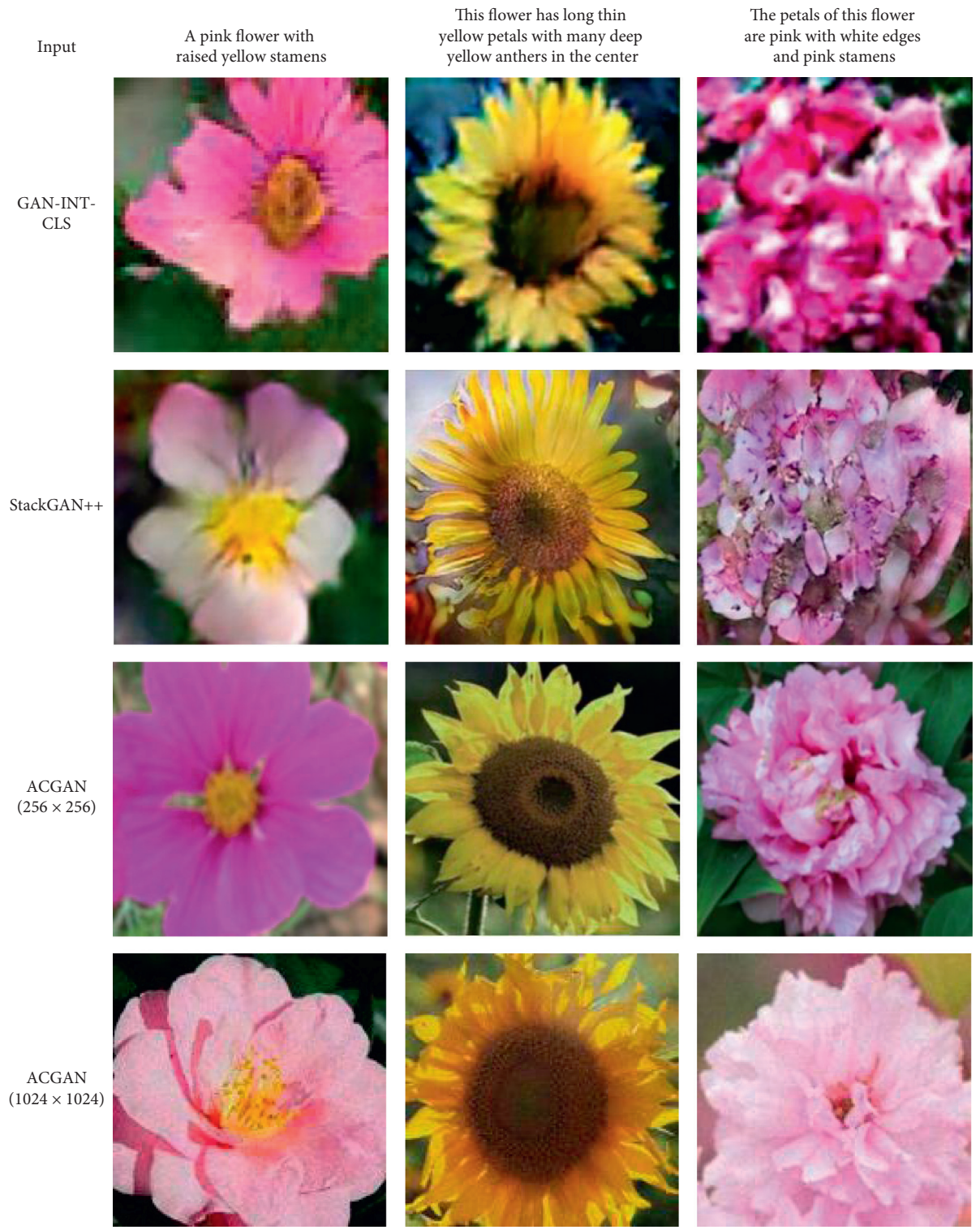


Figure 5: Images generated from descriptions using three GAN models trained on Oxford test set.

is better than AttnGAN, and the generated image is closer to the real image.

Detailed (beak, wings) comparisons of the results between the three models of StackGAN++, AttnGAN, and ACGAN on the CUB dataset are presented in Figure 4. It can be seen that the beak, wings, and feet of the bird are clearer, and the edges and details are more realistic in the images generated by the ACGAN in this paper. For example, the beak of a bird is more vivid and conforms to the text description in the 4th column. Compared with StackGAN++ and AttnGAN, it has achieved better results.

Figure 6: Details (petals) comparison of the images generated from descriptions using three GAN models trained on Oxford test set.

Subjective visual comparisons between the three models of GAN-INT-CLS, StackGAN++ and ACGAN on the Oxford dataset are presented in Figure 5. Details (petals) comparison of the results are presented in Figure 6. It can be seen that the image details generated by GAN-INT-CLS and StackGAN++ are lost, and the shape looks strange (1st and 2nd row) for some examples. ACGAN achieved better results with more details and consistent colors and shapes compared to StackGAN++. For example, the overall shape of the flowers is clearer, and the details of the petals are more obvious in the 4th row. These results confirm that ACGAN is better than StackGAN++, and the generated image is closer to the real image.

5. Conclusions

This paper adds attention mechanism and multilevel cascade structure to generate adversarial network, uses attention mechanism to pay attention to the fine-grained information of word level in semantics, enriches the details of generated images, and generates through cascade structure Higher resolution images. Experiments have shown that, on the same data set, the Attentional Concatenation Generative Adversarial Networks have clearer edge details and local textures against the image generated by the network, making the generated image closer to the real image. Although this method has achieved good results in generating images, it is still difficult to model complex scenes in life. How to deal with this problem needs further study. At the same time, the generated image is similar to the training data, lacking diversity. Therefore, it is intended to combine the zero shot learning and the generative adversarial networks to synthesize the new category image, which will be the focus of the next step.

Data Availability

The basic data used in this article was downloaded from the Internet. There are two-part datasets: (1) the CUB is a public dataset that can be downloaded from http://www.vision.caltech.edu/visipedia/CUB-200-2011.html. (2) The Oxford is a public dataset that can be downloaded from http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Linyan Li and Yu Sun contributed equally to this work.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (61876121, 62002254, 61801323, and 62062003), Natural Science Foundation of the Jiangsu Higher Education Institutions of China (19KJB520054), Research Fund

of Suzhou Institute of Trade and Commerce (KY-ZRA1805) , Primary Research and Development Plan of Jiangsu Province (BE2017663), Foundation of Key Laboratory in Science and Technology Development Project of Suzhou (SZS201609), and Graduate Research and Innovation Plan of Jiangsu Province (KYCX18_2549).

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the Advanced Neural Information Processing Systems (NIPS)*, pp. 2672–2680, Montreal, Canada, December 2014.

[2] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proceedings of the Advanced Neural Information Processing Systems (NIPS)*, pp. 217–225, Barcelona, Spain, October 2016.

[3] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text-to-image synthesis," in *Proceedings of the Advanced International Conference on Machine Learning (ICML)*, pp. 1060–1069, New York, NY, USA, June 2016.

[4] H. Zhang, T. Xu, H. Li et al., "StackGAN++: realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1947–1962, 2018.

[5] H. Zhang, T. Xu, H. Li et al., "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the Advanced International Conference on Computer Vision (ICCV)*, pp. 5907–5915, Venice, Italy, August 2017.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the Advanced International Conference on Learning Representations (ICLR)*, Banff, Canada, May 2014.

[7] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the Advanced International Conference on Machine Learning (ICML)*, pp. 1278–1286, Beijing, China, May 2014.

[8] A. Oord, N. Kalchbrenner, and N. Kavukcuoglu, "Pixel recurrent neural networks," in *Proceedings of the Advanced International Conference on Machine Learning (ICML)*, pp. 1747–1756, New York, NY, USA, August 2016.

[9] F. Lyu, Q. Wu, F. Hu, Q. Wu, and M. Tan, "Attend and imagine: multi-label image classification with visual attention and recurrent neural networks," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 1971–1981, 2019.

[10] Y.-J. Cao, L.-L. Jia, Y.-X. Chen et al., "Recent advances of generative adversarial networks in computer vision," *IEEE Access*, vol. 7, pp. 14985–15006, 2018.

[11] S. Y. Zhao and J. W. Li, "Generative adversarial network for generating low-rank images," *Journal of Acta Automatica Sinica*, vol. 44, no. 5, pp. 829–839, 2019.

[12] H. Eghbal-zadeh, W. Zellinger, and G. Widmer, "Mixture density generative adversarial networks," in *Proceedings of the Advanced Computer Vision and Pattern Recognition (CVPR)*, pp. 5820–5829, Long Beach, CA, USA, November 2019.

[13] B. Gecer, S. Ploumpis, I. Kotsia, and S. Zafeiriou, "GANFIT: generative adversarial network fitting for high fidelity 3D face reconstruction," in *Proceedings of the Advanced Computer Vision and Pattern Recognition (CVPR)*, pp. 1155–1164, Long Beach, California, USA, April 2019.

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of the Advanced International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, January 2016.

[15] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proceedings of the Advanced Neural Information Processing Systems (NIPS)*, pp. 2234–2242, Barcelona, Spain, June 2016.

[16] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Proceedings of the Advanced International Conference on Learning Representations (ICLR)*, Toulon, France, May 2017.

[17] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," in *Proceedings of the Advanced International Conference on Learning Representations (ICLR)*, pp. 1–13, Toulon, France, March 2017.

[18] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug& play generative networks: conditional iterative generation of images in latent space," in *Proceedings of the Advanced Computer Vision and Pattern Recognition (CVPR)*, pp. 4467–4477, Honolulu, Hawaii, April 2017.

[19] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," in *Proceedings of the Advanced International Conference on Learning Representations (ICLR)*, Toulon, France, May 2017.

[20] J. Liu, C. Q. Gao, D. Y. Meng, and A. G. Hauptmann, "Decidenet: counting varying density crowds through attention guided detection and density estimation," in *Proceedings of the Advanced International Conference on Computer Vision (ICCV)*, pp. 5197–5206, Santa Rosa, CA, USA, March 2018.

[21] X. N. Zhang, T. T. Wang, J. Q. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proceedings of the Advanced International Conference on Computer Vision (ICCV)*, pp. 714–722, Salt Lake City, Utah, June 2018.

[22] J. W. Wang, W. H. Jiang, L. Ma, W. Liu, and Y. Xu, "Bidirectional attentive fusion with context gating for dense video captioning," in *Proceedings of the Advanced International Conference on Computer Vision (ICCV)*, pp. 7190–7198, Salt Lake City, Utah, April 2018.

[23] P. Anderson, X. D. He, C. Buehler et al., "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the Advanced International Conference on Computer Vision (ICCV)*, pp. 6077–6086, Salt Lake City, Utah, July 2018.

[24] H. J. Xu and K. Saenko, "Ask, attend and answer: exploring question-guided spatial attention for visual question answering," in *Proceedings of the Advanced European Conference on Computer Vision (ECCV)*, pp. 451–466, Amsterdam, Netherlands, March 2016.

[25] T. Xu, P. C. Zhang, Q. Y. Huang et al., "AttnGAN: fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the Advanced Computer Vision and Pattern Recognition (CVPR)*, pp. 1316–1324, Honolulu, Hawaii, November 2017.

[26] T. T. Qiao, J. Zhang, D. Q. Xu, and D. Tao, "MirrorGAN: learning text-to-image generation by redescription," in *Proceedings of the Advanced Computer Vision and Pattern Recognition (CVPR)*, pp. 1505–1514, Los Angeles, California, USA, March 2019.

[27] M. Abadi, P. Barham, J. Chen et al., "TensorFlow: a system for large-scale machine learning," in *Proceedings of the Advanced Operating Systems Design and Implementation (OSDI)*, pp. 265–283, Savannah, GA, USA, November 2016.

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, *The Caltech-UCSD Birds-200-2011 Dataset: Computation & Neural Systems Technical Report*, California Institute of Technology, Pasadena, CA, USA, 2011.

[29] M. E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proceedings of the Advanced Computer Vision, Graphics & Image Processing (CVGIP)*, pp. 722–729, Marseille, France, December 2008.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 5916205, 14 pages
https://doi.org/10.1155/2020/5916205

Research Article

3D Semantic VSLAM of Indoor Environment Based on Mask Scoring RCNN

Chongben Tao,[1,2] **Yufeng Jin**,[1] **Feng Cao**,[3] **Zufeng Zhang**,[4,5] **Chunguang Li**,[6] **and Hanwen Gao**[1]

[1]*School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, Jiangsu, China*
[2]*Suzhou Automobile Research Institute, Tsinghua University, Suzhou 215134, Jiangsu, China*
[3]*School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China*
[4]*Department of Automation, Tsinghua University, Beijing 100084, China*
[5]*Wuhan Electronic Information Institute, Wuhan 430019, Hubei, China*
[6]*School of Computer Information and Engineering, Changzhou Institute of Technology, Changzhou 213002, Jiangsu, China*

Correspondence should be addressed to Zufeng Zhang; yafflestudio@126.com

Received 30 June 2020; Revised 12 August 2020; Accepted 9 September 2020; Published 20 October 2020

Academic Editor: Jaime Zabalza

In view of existing Visual SLAM (VSLAM) algorithms when constructing semantic map of indoor environment, there are problems with low accuracy and low label classification accuracy when feature points are sparse. This paper proposed a 3D semantic VSLAM algorithm called BMASK-RCNN based on Mask Scoring RCNN. Firstly, feature points of images are extracted by Binary Robust Invariant Scalable Keypoints (BRISK) algorithm. Secondly, map points of reference key frame are projected to current frame for feature matching and pose estimation, and an inverse depth filter is used to estimate scene depth of created key frame to obtain camera pose changes. In order to achieve object detection and semantic segmentation for both static objects and dynamic objects in indoor environments and then construct dense 3D semantic map with VSLAM algorithm, a Mask Scoring RCNN is used to adjust its structure partially, where a TUM RGB-D SLAM dataset for transfer learning is employed. Semantic information of independent targets in scenes provides semantic information including categories, which not only provides high accuracy of localization but also realizes the probability update of semantic estimation by marking movable objects, thereby reducing the impact of moving objects on real-time mapping. Through simulation and actual experimental comparison with other three algorithms, results show the proposed algorithm has better robustness, and semantic information used in 3D semantic mapping can be accurately obtained.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is a technology which enables robots or UAVs to realize autonomous positioning in an unknown environment and autonomous mapping. The robot can get rich information through sensors, which brings more conveniences to solve the problem of localization and mapping. Therefore, SLAM technology is undoubtedly a priority for robot autonomy. Compared with traditional SLAM based on laser sensor, SLAM based on camera vision can make full use of rich texture information on pictures taken by the camera, which provides a huge advantage in relocation and classification of scene semantic information. In recent years, intelligent robots have been widely used in various industries, especially for rapid development of Visual SLAM (VSLAM). Image feature extraction methods represented by deep learning technology have appeared in VSLAM. Meanwhile, deep learning also associates images with semantics and combines with VSLAM methods to build a semantic map and semantic knowledge base of environment. Salehi et al. [1] focused on the real-time fusion of monocular Vision SLAM and GPS data, where a hybrid method of constrained BA/position map is put forward to obtain the

attitude estimation and reconstruction of city scale and geographical parameters. Liu et al. [2] proposed a SSD algorithm based on YOLO and Faster RCNN, adding multiple convolution layers of different scales to maintain the accuracy of Faster RCNN, while a faster speed than YOLO is obtained. Zhang et al. [3] used collinear relationship of points to optimize the existing VSLAM algorithm based on points, and a practical line matching algorithm was given, where compensating computation assisted by straight beam was utilized and the perspective of n-point algorithm was improved. The proposed method is evaluated on indoor sequences of different ranges in the dataset of TUM and also compared with point-based and line-based methods. The results show that the designed algorithm has faster computing speed in comparison with VSLAM system based on point line. Gao et al. [4] proposed an improved method of augmented reality registration based on VSLAM to solve the problem of unstable registration and low registration accuracy of unmarked augmented reality of standard homographic matrix. The VSLAM algorithm generates a 3D scene map in the process of dynamic camera tracking, and then AR based on VSLAM uses 3D map of scene reconstruction to calculate the position of virtual object, which enables and enhances the stability and accuracy of AR registration.

Recently, robustness and availability of VSLAM technology have been strengthened, which tends to be mature [5]. However, sparse image features can provide limited environmental semantic information in dealing with dynamic target motion, lack of texture, or single texture environment. For these problems, hierarchical image feature extraction methods represented by deep learning have appeared in the field of VSLAM in recent years, providing ideas for solving such problems. By modeling bounding box of the most representative first-level detector YOLOv3 in accordance with Gaussian parameters and redesigning loss function, Choi et al. [6] proposed a method to improve detection accuracy and support real-time operation. Li et al. [7] put forward a multitarget detection framework integrating RCNN and DPM, which can precisely detect each single object among all objects in the image. Especially better performance was shown when objects are close to each other. Cai and Vasconcelos [8] developed a multilevel target detection structure, namely, Cascade RCNN, which includes a series of detectors trained by increasing IOU threshold, and higher selectivity for approaching misinformation is obtained. Ren et al. [9] proposed an improved anchoring scheme, where high resolution characterized mapping of small targets for improvement of its performance was used. Eggert et al. [10] introduced an improved scheme for generating anchor proposals and proposed a modification to Faster RCNN which leverages higher resolution feature maps for small objects. A novel multiscale location perception kernel representation (MLKP) method was presented by Wang et al. [11] to obtain the high-order statistics of depth features, and it combined discriminated high-order statistics into representation of object proposals for effective detection for objects. Note that this method can be applied to target detection flexibly. Li et al. [12] put forward a SOR Faster RCNN algorithm, which was used to search same target in different scenes with less training samples. A new robust Faster RCNN method was developed by Zhou et al. [13] to detect targets in multitag images. Unlike Fast RCNN, this design method has stronger robustness. Tao et al. [14] proposed a method of 3D environment semantic mapping based on Mask RCNN algorithm. The input image sequence was filtered by ORB-SLAM for key frame and then image semantic segmentation was combined with SLAM technology to build a 3D semantic map of the environment. Schorghuber et al. [15] fused a robust static weighting strategy based on corresponding distance of depth edge into intensity assisted ICP and thus proposed a real-time RGB-D visual range measurement method. Laidlo and Leutenegger [16] proposed a 3D reconstruction system called DeepFusion which leverages the output of a convolutional neural network (CNN) in DeepLab-v2 [17] to produce fully dense depth maps for key frames that include metric scale. DeepFusion fuses the output of a semidense multiview stereo algorithm with the depth and gradient predictions of a CNN in a probabilistic fashion, using learned uncertainties produced by the network. McCormac et al. [18] proposed an improved Elastic Fusion SLAM [19] method based on convolution neural network to build a dense 3D semantic map, which relies on Elastic Fusion SLAM algorithm to provide estimation for interframe pose of indoor RGB-D video, uses convolution neural network to predict classes and labels of pixel-level object, and finally combines Bayesian upgrading strategy and conditional random field model to realize probability upgradation of predicted CNN value from different perspectives so as to generate a dense 3D semantic map. Mur-Artal et al. [20, 21] proposed a ORB-SLAM2 method, which uses depth information to synthesize a three-dimensional coordinate, and the information of an image can be accurately extracted. The backend uses BA algorithm to build a global sparse map reconstruction. Therefore, this method is more lightweight and can be used in semantic mapping [22]. However, these aforementioned methods have some drawbacks in correctness of classification in the case of sparse feature points.

Motivated by the aforementioned existing problems, this paper proposed an ingenious semantic VSLAM algorithm combining BRISK feature [23] with a VSLAM algorithm based on Mask Scoring RCNN [24]. Semantic information of independent targets in scenes provides semantic information including categories. Meanwhile, the impact of moving objects during semantic mapping is reduced by the probability update of semantic estimation by marking movable objects.

2. Three-Dimensional Map Generation

2.1. System Overview. The overall architecture of the algorithm has two parts including front-end processing and back-end processing. A BRISK algorithm is used in front-end processing to extract features as well as key points. A Mask Scoring RCNN method is used in back-end processing including segmentation, semantic association, and semantic mapping as shown in Figure 1.

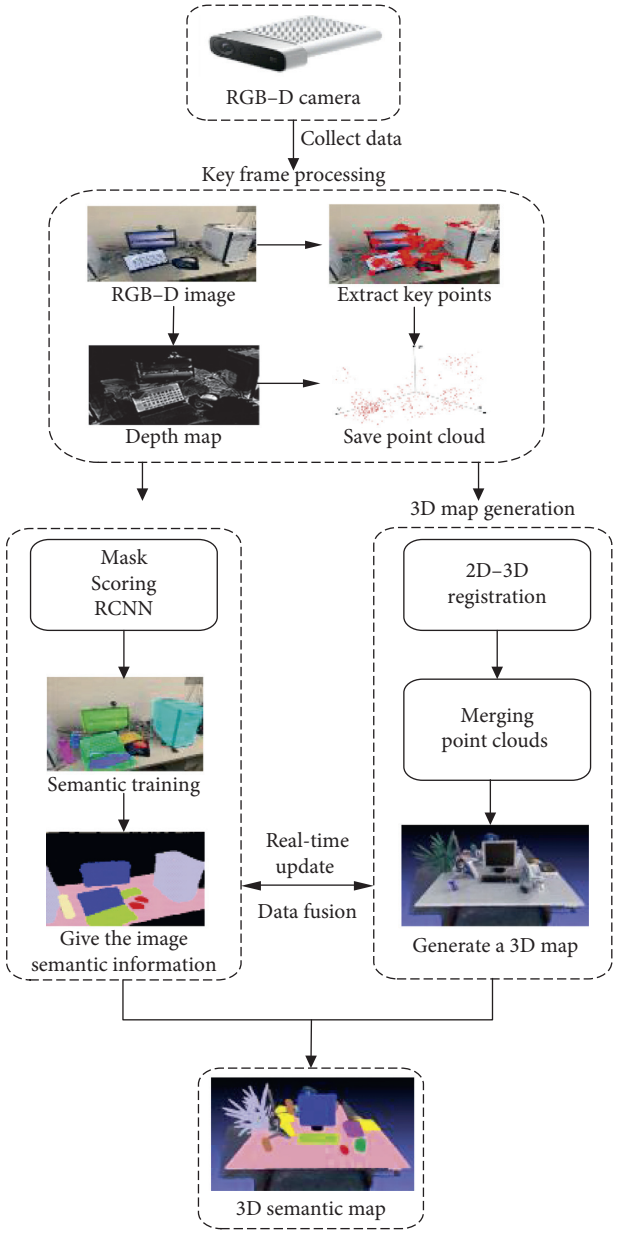


Figure 1: The entire framework of the proposed algorithm.

2.2. Dense SLAM Algorithm Based on BRISK Feature Extraction. Binary Robust Invariant Scalable Keypoints (BRISK) algorithm is similar to SIFT (scale-invariant feature transform), SURF (speeded-up robust feature), and ORB (oriented FAST and rotated BRIEF) [23], which is a feature point matching algorithm, but calculation speed is faster than other two algorithms. BRISK algorithm constructs image pyramid for multiscale expression, so it has good rotation invariance, scale invariance, good robustness, and so on. In particular, BRISK algorithm performs the best for image registration with large blurs. BRSIK algorithm consists of two parts: detection of key points and description of key points.

The detection for key points of BRISK is based on scale space composed of image pyramid. FAST is used to detect candidate key points in all layers of pyramid image, and candidate key points suppressed by nonmaximum are taken as final key points. After all the key points in image are obtained, key points need to be described. Different from descriptors constructed by SURF, SIFT, and other algorithms, BRISK algorithm applies binary string to describe key points so as to use Hamming distance to calculate matching degree and enable it to have a calculation speed faster than Euclidean distance. BRISK describes features in the mode of neighborhood sampling. The algorithm constructs multiple Bresenham concentric circles with key points as a center and takes N points evenly distributed to calculate feature direction and binary descriptors, respectively, in accordance with its long distance sampling points and short distance sampling points. Finally, Hamming distance is used to match above binary feature description so as to obtain global motion estimation of image.

In order to avoid the problem of sparse point cloud map caused by strict screening strategy to avoid gross error, this paper proposed a 3D mapping method of inverse depth filtering based on Visual SLAM. The task of inverse depth filter is used to estimate scene depth of created key frame and only build matching cost within depth range, which greatly reduces stereo matching time [25]. Based on the principle of depth similarity between adjacent pixels, after initial depth map is obtained, smoothing of intraframe and elimination of outer point are carried out, which increased density of depth map and eliminated possible isolated matching points. And Gaussian fusion is carried out for each candidate inverse depth hypothesis through an inverse depth fusion method of multikey frame to optimize current depth value of key frame. The specific algorithm steps are as follows:

Step 1: measuring for scene depth. Each map point observed by key frame at any time is projected into key frame image to calculate the depth value of the map point in the key frame coordinate system. Maximum depth and minimum depth are selected to set inverse-depth search range of scene.

$$p_i = (x_i, y_i, z_i)^T, \tag{1}$$

$$p_i^k = T_{k,w} p_i = \left(x_i^k, y_i^k, z_i^k\right)^T, \tag{2}$$

$$\begin{aligned} p_{\min} &= \min\left(z_i^{-k}\right), \\ p_{\max} &= \max\left(z_i^{-k}\right), \\ i &\in (0, n), \end{aligned} \tag{3}$$

where p_i is the homogeneous representation of 3D coordinates of map points in the world coordinate system; $T_{k,w} p_i$ is the pose transformation between the camera coordinate system and world coordinate system at time k; $p_i^k = T_{k,w} p_i$ is the homogeneous representation of 3D coordinates of map points in the camera coordinate system at time k; and N is the number of map points that can be observed in the key frame at time k.

Step 2: stereo matching. Pixel depth is calculated by using aggregate stereo matching algorithm of variable weight cost [26]. Based on layers of cost volume in the

limited stereo matching of scene depth value calculated in Step 1, it is only searched in the range of parallax opposite to inverse depth $(p_{\min}, p_{\max})$ so as to reduce the amount of calculation. Post-processing step of parallax deletion in stereo matching is eliminated at the same time, only retaining inverse depth of pixels with the same parallax in the left and right consistency matching.

Step 3: elimination of isolated outer point. It is assumed that parallax obtained by stereo matching follows the Gaussian distribution of variance 1, i.e., d: $N(d_0, 1)$:

$$p = z^{-1} = d(fb)^{-1}, \tag{4}$$

where d_0 is the parallax value calculated by stereo matching, f is the focal length of the camera, b is the baseline, z is the depth value of the pixel, and p is the inverse depth. The inverse depth distribution after transformation is as follows:

$$p\text{: } N\left(\frac{d_0}{fb}, \frac{1}{fb}\right). \tag{5}$$

The inverse depth map obtained in stereo matching stage is filled and isolated outliers are eliminated. The specific steps are as follows:

(1) For each pixel with inverse depth distribution, the number of pixels whose inverse depth distribution meets χ distribution of less than 5.99 is calculated. As shown in formula (6), inverse depth is eliminated in case of number less than 2. When the number is greater than 2, formula (7) is used to fuse the inverse depth that meets the requirements of χ distribution. After fusion, inverse depth of the pixel is p_p, while variance $\sigma_{p_p}^2$ is the minimum variance of inverse depth before fusion.

$$\frac{(p_x - p_y)^2}{\sigma_x^2} + \frac{(p_x - p_y)^2}{\sigma_y^2} < 5.9, \tag{6}$$

$$p_p = \frac{\sum_n \left(1/\sigma_{p_j}^2\right)}{\sum_n \left(1/\sigma_{p_j}^2\right)}, \tag{7}$$

$$\begin{aligned} p_p &= \frac{\sum_n \left(1/\sigma_{p_j}^2\right) p_j}{\sum_n \left(1/\sigma_{p_j}^2\right)}, \\ \sigma_p^2 &= \frac{1}{\sum_n \left(1/\sigma_{p_j}^2\right)}, \end{aligned} \tag{8}$$

where x and y are eight surrounding pixels around current pixel and n is a number which satisfies χ distribution.

(2) For each pixel that does not have an inverse depth distribution, check whether the inverse depth distribution between the eight surrounding pixels meets the chi-square distribution. When the number which satisfies χ distribution is greater than 2, formula (2) is used for inverse depth fusion, and homomorphic variance is the minimum variance of inverse depth before fusion.

Step 4: fusion of inverse depth. After position and pose of key frame are calculated by tracking thread, current depth information of key frame is optimized through following six inverse depth maps of the key frame. The specific steps are as follows:

(1) Project map point corresponding to inverse depth map of current key frame to adjacent key frame and read the inverse depth p_0 of projection point and inverse depth variance of σ_0^2.

(2) Map points whose inverse depth is $p_0 + \sigma_0$, p_0 and $p_0 - \sigma_0$ in the adjacent frame to current frame, and the reverse depth of p_1, p_2, and p_3 is retained.

(3) Construct candidate inverse depth of fusion, assuming ρ: $N(p_2, [\max(|p_1 - p_2|, |p_3 - p_2)]^2)$.

(4) Cycle above steps to obtain 6 candidate hypotheses of fusion inverse depth and select inverse depth hypothesis to be fused by using χ distribution less than 5.99. After fusion, inverse depth p_p and variance σ_p^2 are

where p represents pixels of current frame and n represents numbers of inverse depth to be fused.

Step 5: re-elimination of isolated outer point. Based on assumption that depth of adjacent areas in scene is similar, inverse depth map obtained by inverse depth fusion is smoothed in frame and removed from outer points so as to improve accuracy of output map points and increase density of point cloud. The specific steps are reverse depth filling and removal in Step 2.

Step 6: get cloud point map. All points in the processed depth graph are transformed to the global coordinate system, and point cloud map is obtained to construct current environment map. However, if point cloud data of each frame are integrated into map, a lot of computing resources will be occupied, thus reducing real-time performance of the system. Therefore, this paper uses point cloud map based on key frame to build dense environment map by dividing an entire map into several submaps with specific key frames to reduce memory consumption. The extracted key frame is optimized and saved to global map, and the dense global map is finally output, as shown in Figure 2.

For a key frame, the RGB-D camera provides color image and depth image. The formula of 3D point cloud in accordance with camera internal parameters is as follows:

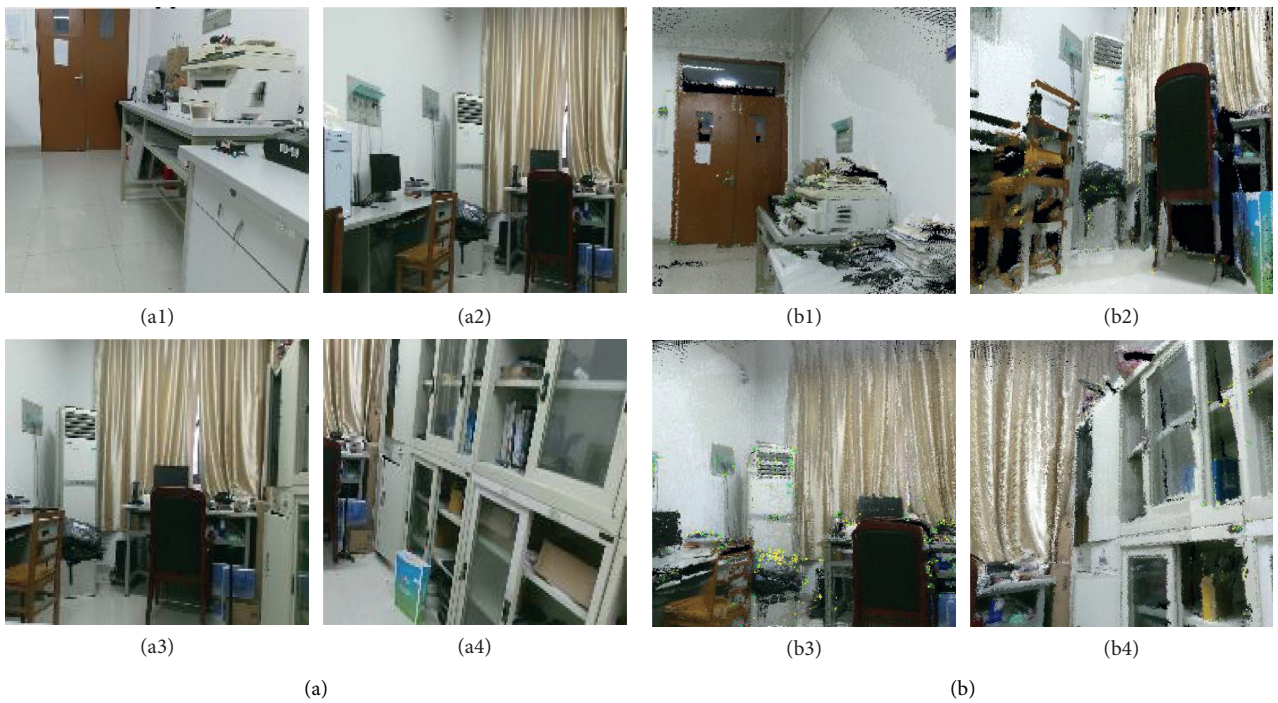


Figure 2: Key frame reconstruction in submaps. (a) Local image of four segments of experiments. (b) Local image of four segments of experiments after being constructed.

$$\begin{cases} z = \dfrac{d}{s}, \\ x = (u - c_x) \cdot \left(\dfrac{z}{f_x}\right), \\ y = (v - c_y) \cdot \left(\dfrac{z}{f_y}\right), \end{cases} \tag{9}$$

where f_x, f_y, c_x, c_y are internal parameters of the camera; (u, v) is the image coordinate; (x, y, z) is the image coordinate system; d is the distance of pixel point measured by the depth camera, with unit of mm; and s is the scale coefficient of actual distance and measured distance d. In this method, one of the advantages of point cloud is that it can be generated directly from RGB-D image efficiently without additional processing, with very intuitive operation of filtering (Algorithm 1).

3. Semantic Information Acquisition

The task of target detection includes classification and positioning, which not only gives the category information of an object to be detected but also determines position and size of the object in an image and surrounds it with a smallest rectangular frame. The main steps of target detection include preprocessing of input image and filtering of candidate areas of the image by a sliding window. Then, one kind of feature extraction algorithm is used including SIFT, HOG, or DPM to extract features for candidate areas, and finally a classification algorithm is used to classify extracted features. However, some defects such as unstable matching, weak antinoise ability, slow detection speed, and poor extraction effect for fuzzy and smooth edges coexist in the traditional object detection model. Compared with the traditional object detection model, the object detection model based on deep learning has more powerful feature expression ability, strong generalization ability, and good robustness.

A BMASK-RCNN network is designed in this paper which refers to Mask Scoring RCNN based on deep neural networks. Mask Scoring RCNN evolves from Mask RCNN, whose network framework is shown in Figure 3. The traditional Mask RCNN consists of two stages. The first stage is realized by convolution of RPN. Regardless of the object category, bounding box of a candidate object will be proposed. The second stage is called RCNN stage, which uses RoIAlign to extract features for each candidate where a bilinear interpolation is used to complete pixel-level alignment and finally generate candidate classification, bounding box regression, and mask prediction.

The loss function of Mask RCNN consists of three parts, namely, classification error, detection error, and segmentation error. The expression is as follows:

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}, \tag{10}$$

where L_{cls} and L_{box} are the same with Faster RCNN; mask branch has dimensions of km^2 for each ROI, which indicates the solution is k binary masks with the solution of $m \times m$; K represents numbers of category, conducting sigmoid for each pixel; and L_{mask} is defined as average entropy loss of binary cross.

(1) **Input:** map point data x
(2) **Output:** point cloud map y
(3) The search range of scene depth measurement $(p_{\min}, p_{\max})$ is defined as $p_i = (x_i, y_i, z_i)^T$
(4) The scene depth value limits the number of layers of matching cost in stereo matching
(5) Isolated outlier culling while $p = z^{-1} = d(fb)^{-1}d$: $N(d_0, 1)$ then The inverse depth distribution after transformation is p: $N((d_0/fb), (1/fb))$
(6) Inverse deep fusion
(7) Isolated outlier secondary culling
(8) Get point cloud map

Algorithm 1: 3D mapping method for inverse depth filtering.

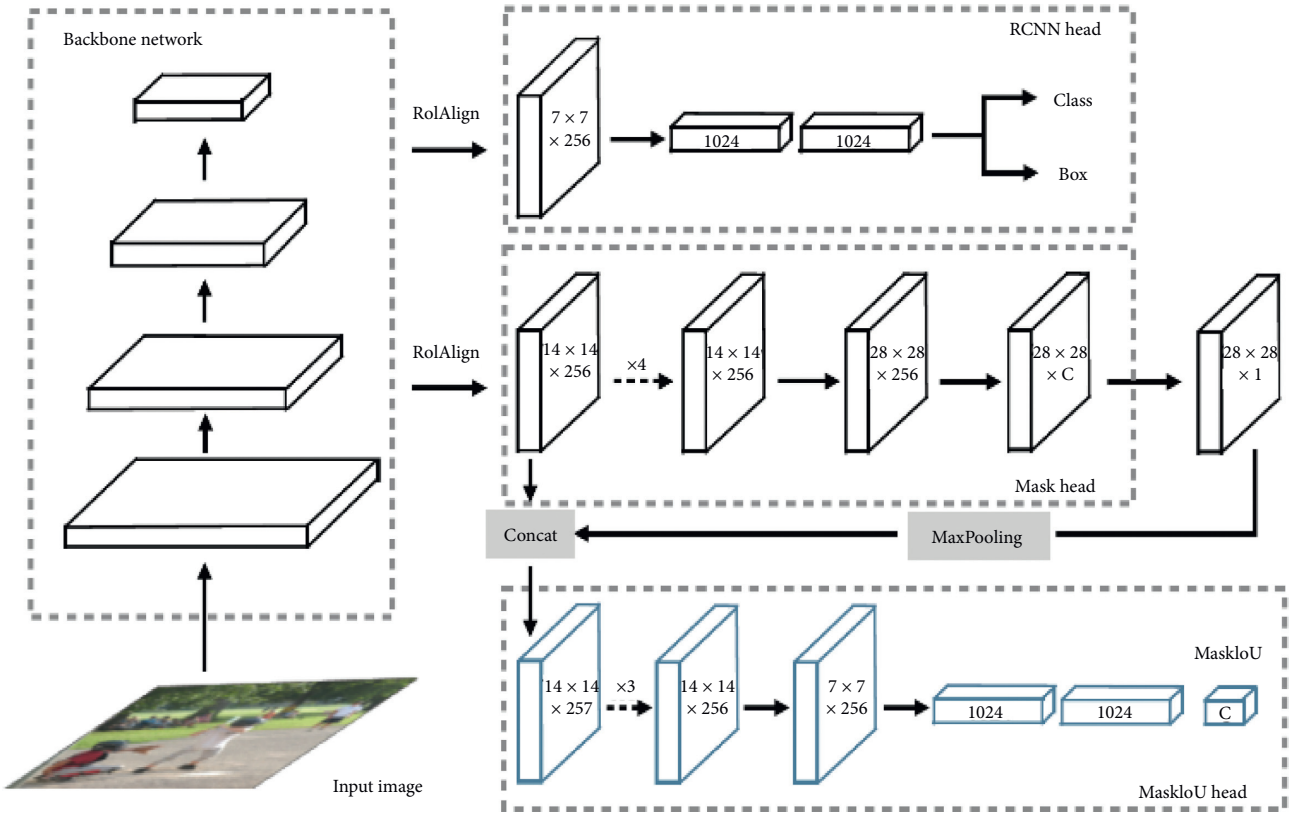


Figure 3: The network framework of Mask Scoring RCNN [24].

$$L_{\text{mask}} = -\frac{1}{m^2}\sum_{1\le i,j\le m}\left[y_{ij}\log\hat{y}_{ij}^{k} + \left(1-y_{ij}\right)\log\left(1-\hat{y}_{ij}^{k}\right)\right], \tag{11}$$

where y_{ij} is the label of cell (i, j) in the real mask within region of $m \times m$ and $\hat{y}_{ij}^{k}$ is the predicted value of the same cell in the k learning mask of ground truth value class.

However, the score for detecting (instance segmentation) hypotheses is determined by the largest element in its classification score in the current Mask RCNN framework. Due to clutter background, occlusion, and other problems, the score for classification may be high but mask quality is low. To overcome this problem, on the premise of generality of Mask Scoring RCNN, MaskIoU head module is added to enable the improved Mask RCNN for obtaining higher mask scores.

MaskIoU head is used to regress the IoU between predicted mask and its true label mask. For this purpose, feature concatenation and predicted mask of RoIAlign layer are used as input of MaskIoU head. A maximum pooling layer with a kernel size of 2 and a step size of 2 is used to make the predicted mask have the same space size as the RoI feature, and only MaskIoU is chosen to return to real label class. The MaskIoU head consists of four roll up layers and three fully connected layers. The four roll up layers follow mask head and set the kernel size and filter number of all the convolution layers to 3 and 256, respectively. Three fully connected (FC) layers follow RCNN head and set output of

first two FC layers to 1024 and the final FC output to the number of classes.

Table 1 is a comparison of results of MS RCNN algorithm and other algorithms on COCO test set, which shows that the MS RCNN algorithm has obvious advantages over other algorithms.

4. 3D Semantic Mapping Method

In the process of semantic mapping, VSLAM not only obtains geometric information in the environment but also recognizes independent individuals and obtains semantic information such as their position, posture, and functional attributes. The key of semantic VSLAM is to accurately recognize objects in the environment. Extracted features from target frame correspond to stored target object and map data, respectively, and then mapping relationship between the image data and the target object is established. The core idea of this paper can be expressed as follows: semantic information is extracted from key frames during the process of 3D mapping, and then the semantic information is fused into the constructed 3D map to create a new 3D semantic map. The flowchart of 3D semantic mapping is shown in Figure 4.

Firstly, Mask Scoring RCNN is used to train semantic database and then determine whether the current frame is a key frame. After key frame is determined, objects contained in semantic database in frame are detected and segmented, and then 2D image in the current key frame is semantically labeled. Finally, points containing semantic information in 2D image are mapped to 3D point cloud. It is regarded as the same object if there is the same semantic information.

For a system, system resources will be greatly consumed if all the image frames acquired by the camera are processed, so image key frame is usually selected for processing, and front-end tracking module of SLAM algorithm determines whether to select a current image frame as the key frame. Rules of key frame selection are as follows:

(1) There must be a sequence interval between the current frame and the previous key frame

(2) The thread of the local map is idle

(3) The current frame and previous key frame share a build area below a certain range

(4) The current frame has enough feature points to match, as shown in Algorithm 2

For each key frame, semantic information $X_t = \{X_k\}^N$ can be obtained through instance segmentation algorithm of Mask Scoring RCNN to obtain semantic information $X_t = \{X_k\}^N$, where $X_k = (x_k^a, x_k^b, x_k^c)$; x_k^a represents category of instance object; x_k^b represents outline of instance object; and x_k^c represents confidence level of instance object. The result of semantic acquisition for a key frame is shown in Figure 5.

The flowchart of semantic mapping is shown in Figure 4. After a key frame is selected, the key frame will be processed by two threads simultaneously: one is VSLAM algorithm, which runs according to original VSLAM system; the other is mainly the association and fusion process of object semantic. The obtained semantic information is processed in two aspects: on one aspect, feature points with dynamic category are marked as unavailable to reduce the impact of object movement on the mapping. On the other aspect, 2D image with semantic annotation information in the key frame is mapped to 3D point cloud so as to find mapping relationship between map points through finding feature points of object frame and semantic information. Algorithm 3 is used for data fusion.

5. Experiments and Analysis

5.1. Introduction to Experiment Platform. This experiment uses a self-built experimental platform, as shown in Figure 6, which is equipped with Microsoft Kinect 3.0 depth camera. The main body is composed of a main control unit, bracket, driving wheel, and chassis. The operating system adopts ROS (Robot Operating System) [30]. ROS is a robot-oriented open source operating system, which provides services including hardware abstraction, low-level device control, implementation of commonly functions, interprocess messaging, and package management. Operating frame is a processing architecture where ROS communication module is used to realize network connection of loose coupling between modules. It performs various types of communication, including service-based synchronous RPC (Remote Procedure Call) communication, topic-based communication of data flow, and data storage on parameter server. The mobile robot independently designed in this paper is a comprehensive experimental platform integrating environment perception, dynamic decision making and planning, behavior control, and execution. Deep learning and training are carried out in Ubuntu 18.04 system environment, with processor model of Intel i9-9900k and memory of 64 GB. In order to get higher training and testing speed, this paper uses GTX 2080Ti graphics card to accelerate training.

5.2. Verification Experiments. In order to prevent irrelevant semantic information from interfering with map construction, the network structure of Mask Scoring RCNN is adjusted. This experiment uses a TUM RGB-D SLAM dataset, where 24 types of objects are selected as shown in Table 2.

Since the onboard computer of the robot is not equipped with a GPU processor, the target detection algorithm of this paper is completed by a graphics workstation which uses TensorFlow as the framework. ROS is used for communication between the workstation and the robot. The graphics workstation is equipped with a GTX2080Ti graphics card for computing acceleration. After the key frame is detected, the semantic information of the target point cloud can be obtained according to the coordinate correspondence. The image of target detection and recognition effect and semantic map of dense point cloud are shown in Figure 7.

Comparisons of loss iteration curves for four algorithms are shown in Figure 8. The red line in Figure 8 represents the loss value of the proposed BMASK-RCNN method, and its

Table 1: Comparative results of MS RCNN algorithm and other instance segment algorithms on COCO testing set.

Method	Backbone	AP	AP@0.5	AP@0.75	APS	APM	APL
MNC [27]	ResNet-101	23.2	43.2	25.1	4.5	24.8	44.3
FCIS [28]	ResNet-101	28.9	48.7	—	—	—	—
FCIS+++ [28]	ResNet-101	34.2	53.7	—	—	—	—
Mask RCNN [14]	ResNeXt-101 FPN	36.9	61.2	38.6	17.1	38.7	52.4
MaskLab+ [29]	ResNet-101(JET)	37.8	62.4	41.0	18.2	40.9	50.7
Mask RCNN	ResNet-101	33.3	55.0	36.6	13.2	36.4	52.3
MS RCNN		35.4	54.9	38.1	13.7	37.6	53.3
Mask RCNN	ResNet-101 FPN	37.0	59.2	39.5	17.1	39.3	52.9
MS RCNN		38.3	58.8	41.5	17.8	40.4	54.4
Mask RCNN	ResNet-101-DCN+FPN	38.4	61.2	41.2	18.0	40.5	55.2
MS RCNN		39.6	60.7	43.1	18.8	41.5	56.2

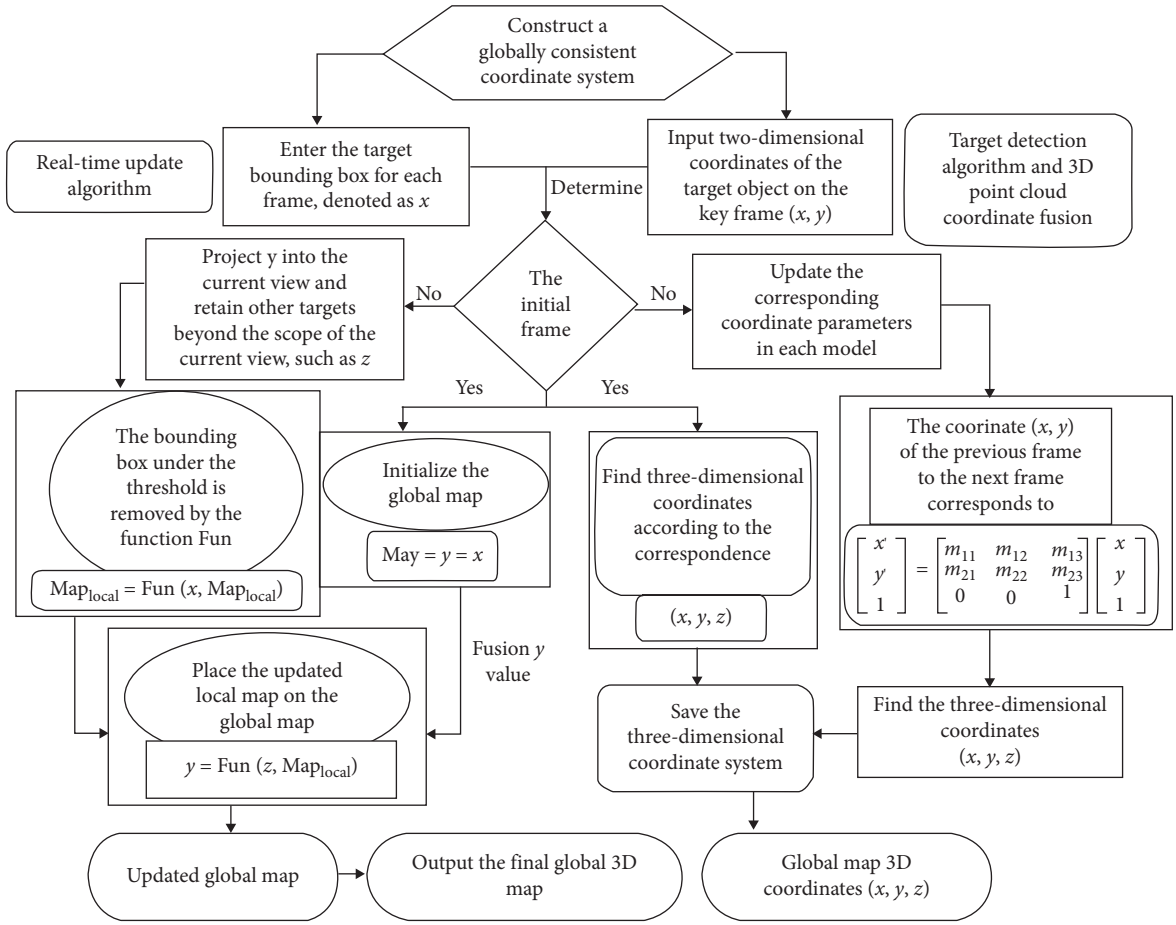


Figure 4: Flowchart for construction of semantic map.

loss value is always smaller than Fast RCNN and Faster RCNN. Although between 0.5×10^4 iterations and 1.5×10^4 iterations, the loss value of BMASK-RCNN is comparable to Mask RCNN, but after 15000 iterations, the curve of BMASK-RCNN stabilized below Mask RCNN. After 1.5×10^4 iterations, it can be seen that the proposed BMASK-RCNN method is more accurate than three methods.

Comparisons of precision-recall curves for four algorithms are shown in Figure 9, where the ordinate value represents detection accuracy of a measured target. The value of abscissa represents recall rate, namely, the total number of correctly detected targets divided by the total number of targets. Obviously, when the area under the curve is larger, the performance of the algorithm is better, and the detection effect is more accurate and complete. It can be seen from Figure 9 that the area under the precise recall curve of this algorithm is significantly larger than other three methods. Simulation results show that the proposed

Input: last key frame;
Output: new key frames;
(1) **if**
(2) (1) The interval between the current and previous key frame sequence is 30 frames;
(3) (2) Local map thread is idle;
(4) (3) The current frame and previous key frame share a build area threshold of less than 90%;
(5) (4) The number of matching point pairs is at least 100;
(6) Select as key frame;
(7) **else**
(8) discarded;
(9) **end if**

Algorithm 2: Key frame selection algorithm.

Figure 5: Acquisition of semantic information.

Input: feature points and semantic features on the current key frame ;
Output: 3D global semantic map coordinates;
(1) Coordinate system;
(2) Mark unusable points on map;
(3) Determine current frame;
(4) **if** initial frames then
(5) (1) Find map point coordinates corresponding to the target feature points;
(6) (2) Get semantic information about the target $p_k = (p_1, p_2, p_3)$, where p_1 is the category, p_2 is the confidence of the detection result, and p_3 is the target contour;
(7) (3) Semantic information is associated with geometric feature points through mapping relation so that feature points have both geometric and semantic information;
(8) (4) The relative motion of the camera is calculated according to feature matching, and the coordinates of the 3D map corresponding to the target feature points are found;
(9) **else**
(10) (5) The new parameters are substituted into the built model;
(11) (6) Insert a new key frame;
(12) (7) Repeat step (1), step (2), step (3), and step (4);
(13) (8) Save coordinate data;
(14) **end if**

Algorithm 3: Data association and fusion processes.

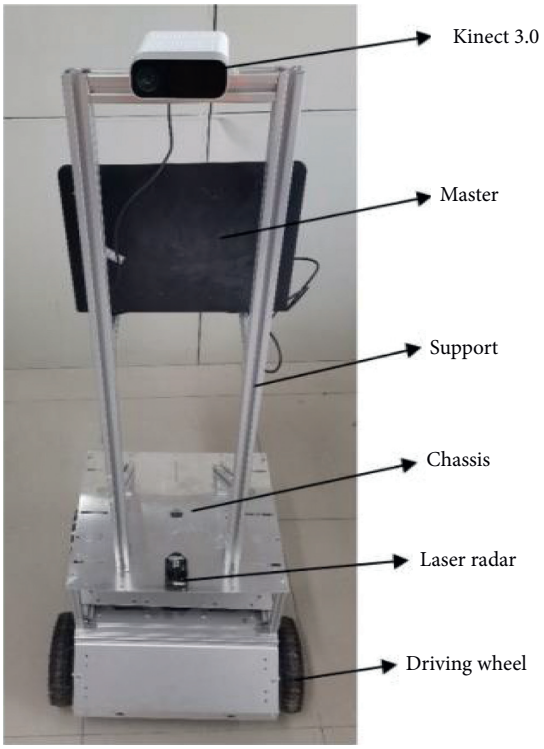


Figure 6: Experimental platform of robots.

Table 2: Selection for 24 types of objects.

Chair	Air conditioner	Screen	Robot	Desk	Bookcase
Door	Keyboard	Mouse	Drone	Cup	Person
Trophy	Switchbox	Bottle	Desk	Flower pot	Book
TV	Jackboard	Cell phone	Potted plant	Suitcase	Umbrella

(a) (b)

Figure 7: Effect image of target detection and recognition (a) and semantic map of dense point cloud (b).

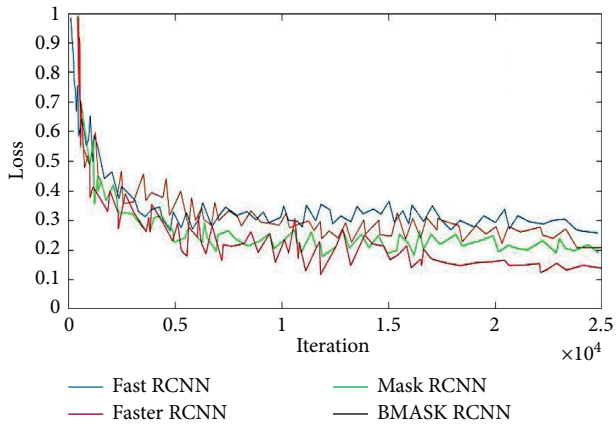


Figure 8: Curve comparative chart of loss iteration.

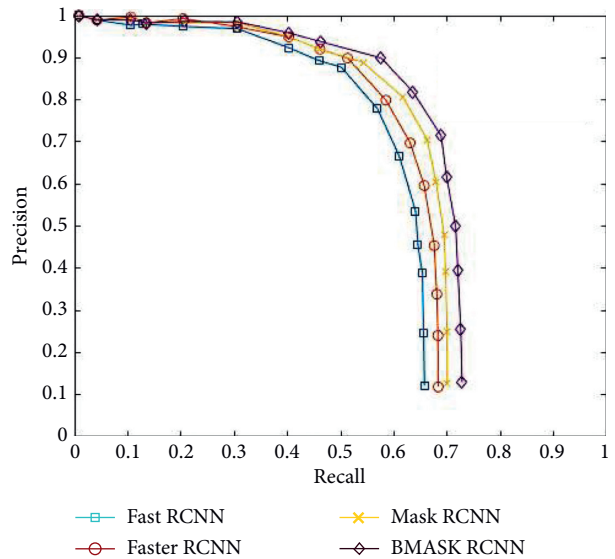


Figure 9: Curve comparative chart of precision and recall.

BMASK-RCNN method has higher accuracy than other three methods.

Figures 10 and 11 show error analysis graph generated using TUM dataset of freiburg2_large_with_Loop and freiburg1_XYZ to run the proposed VSLAM algorithm. freiburg1_xyz is a common small scenario dataset of TUM dataset, and freiburg2_large_with_loop is a large scene dataset from TUM. It can be seen from Figures 10 and 11 that overall effect of the proposed VSLAM algorithm is better than RGB-D SLAM. In the small environment, two systems have better stability; however, compared with RGB-D SLAM, the red lines representing errors in the absolute trajectory error diagram of the VSLAM algorithm are significantly reduced. In large scenarios with closed loops, under the influence of complex environment, RGB-D SLAM errors are relatively high and prone to drift. However, through the semantic information in the scene, the VSLAM algorithm can improve accuracy of mapping and localization, and thus the peak value of blue broken line in the relative pose error is small. In the same period of time, the peak value of broken lines is kept within 0.3 m, while the peak value of RGB-D SLAM lines reaches 0.8 m at most. The attitude error of the proposed VSLAM algorithm is closer to the same range, and the error is relatively low. In large scenarios, the performance of the proposed VSLAM algorithm is obviously better than RGB-D SLAM.

In order to obtain more accurate experimental results, the TUM RGB-D SLAM dataset is used which provides RGB-D images at a frame rate of 30 Hz, with a resolution of 640×480, as shown in Figure 12, for the operation effect.

The front-end part of the algorithm is the SLAM pose estimation and synchronous positioning module, which performs target detection tasks at the same time. Then, key frame pictures as well as all the useful data of key frame images including corresponding map points, semantic information, and position information are saved. Finally, data are transmitted to the server for data fusion calculation.

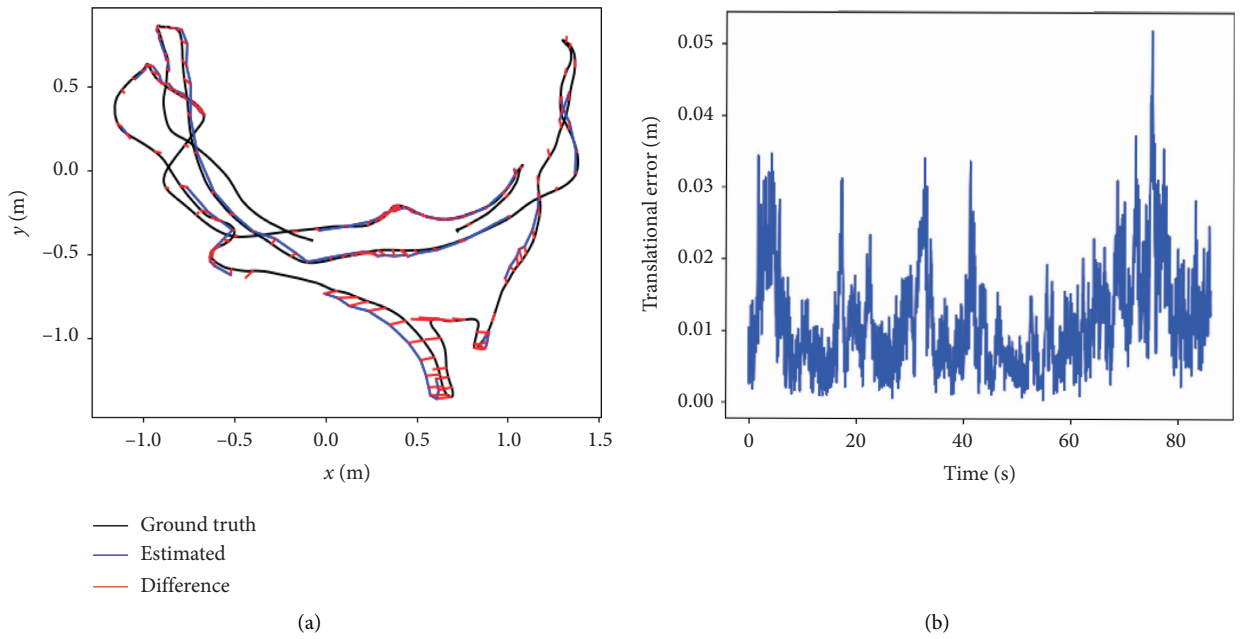


Figure 10: Error analysis of dataset freiburg1_xyz. (a) Absolute track error for construction of map. (b) Relative pose error for construction of map.

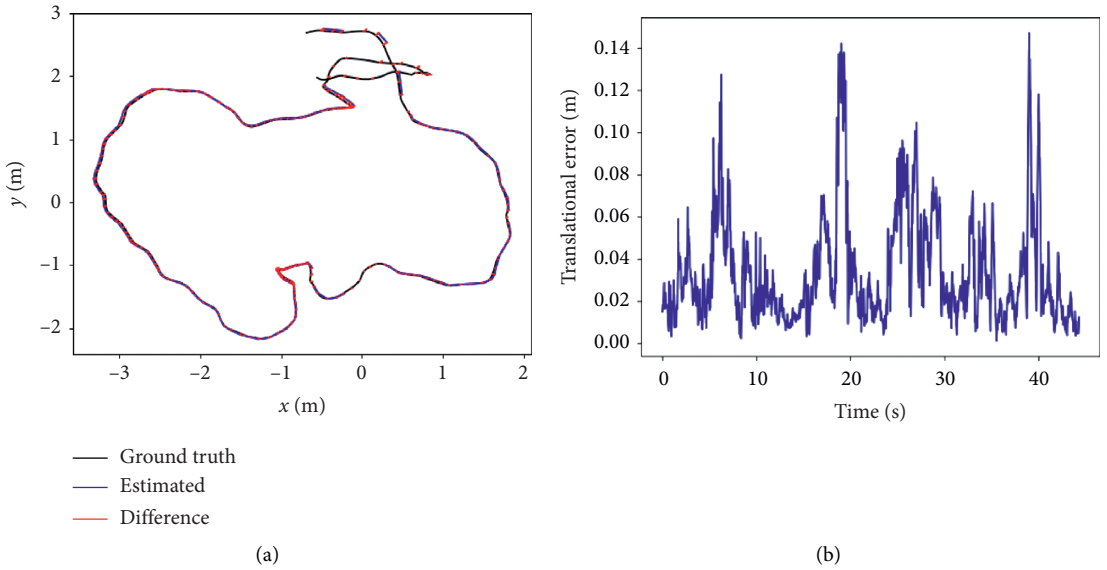


Figure 11: Error analysis of dataset freiburg2_large_with_loop. (a) Absolute track error for construction of map. (b) Relative pose error for construction of map.

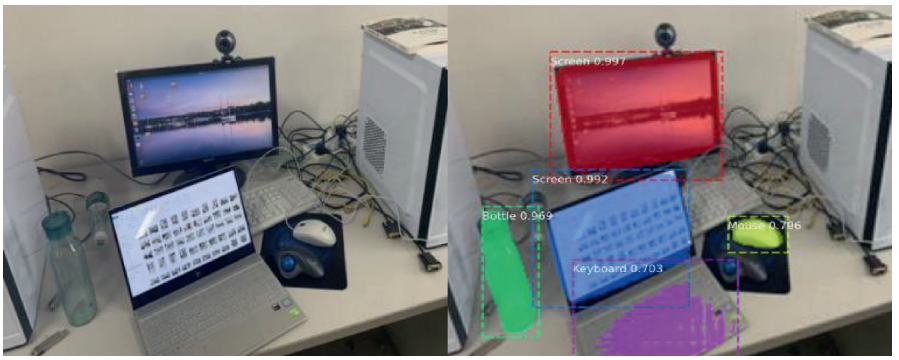


Figure 12: Operation images of the dataset.

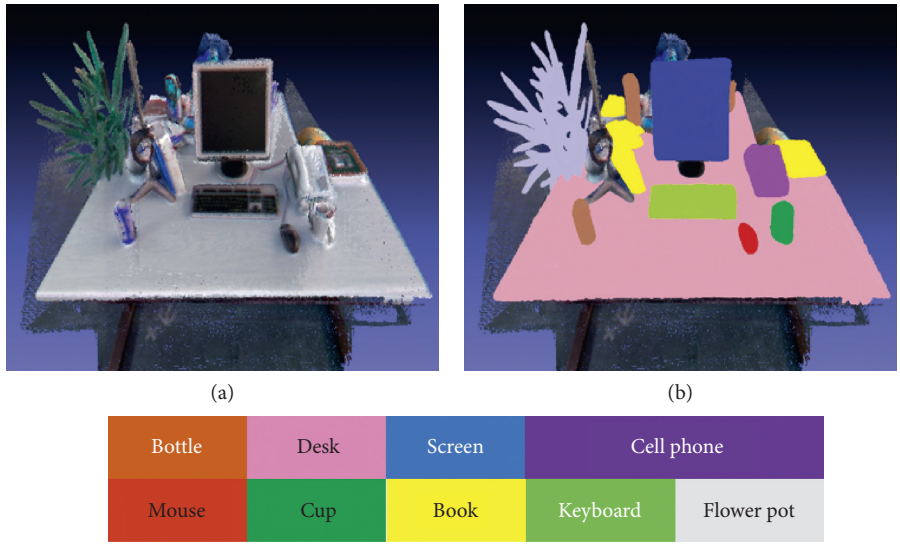


Figure 13: 3D map (a) and 3D semantic map (b).

Additionally, a semantic map is built in the robot in real time as shown in Figure 13.

6. Conclusion

This paper firstly uses a BRISK algorithm to extract feature points, then a Mask Scoring RCNN algorithm is used to detect targets and obtain semantic information of key targets in the environment, and the relative position relationship between target detection results is established. Then, targets are matched, and the similarity is calculated between key frames. Finally, the Mask Scoring RCNN algorithm is used to complete segmentation of targets, and a dense 3D semantic map surrounding the robot is constructed. The proposed method in this paper has achieved good results on the TUM RGB-D SLAM dataset and has verified the feasibility of the application of semantic information in Visual SLAM mapping. There is still room for improvement in this research. For example, the relationship between line and surface features in the target detection frame and the category of the corresponding object can be established to achieve stronger robustness and structure a semantic VSLAM system with better performance.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported in part by the National Natural Science Foundation of China (grant no. 61801323), the Science and Technology Project Fund of Suzhou (grant nos. SYG201708 and SS2019029), and the Construction System Science and Technology Fund of Jiangsu Province (grant no. 2017ZD066).

References

[1] A. Salehi, V. Gay-bellile, S. Bourgeois, and F. Chausse, "A hybrid bundle adjustment/pose-graph approach to VSLAM/GPS fusion for low-capacity platforms," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1728–1735, Los Angeles, CA, USA, June 2017.

[2] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the 2016 European Conference on Computer Vision*, Amsterdam, Netherlands, 2016.

[3] F. Zhang, T. Rui, C. Yang, and J. Shi, "Lap-slam: a line-assisted point-based monocular vslam," *Electronics*, vol. 8, no. 2, p. 243, 2019.

[4] Q. H. Gao, T. R. Wan, W. Tang, L. Chen, and K. B. Zhang, "An improved augmented reality registration method based on visual slam," *E-learning and Games*, vol. 10345, pp. 11–19, 2017.

[5] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[6] J. Choi, D. Chun, H. Kim et al., "Gaussian yolov3: an accurate and fast object detector using localization uncertainty for autonomous driving," in *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 502–511, Seoul, Republic of Korea, November 2019.

[7] J. Li, H.-C. Wong, S.-L. Lo, and Y. Xin, "Multiple object detection by a deformable part-based model and an R-CNN," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 288–292, 2018.

[8] Z. Cai and N. Vasconcelos, "Cascade RCNN: delving into high quality object detection," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, June 2018.

[9] S. Ren, K. He, R. Girshick et al., "Faster RCNN: towards real-time object detection with region proposal networks,"

Advances in Neural Information Processing Systems, vol. 39, pp. 91–99, 2015.

[10] C. Eggert, D. Zecha, S. Brehm et al., "Improving small object proposals for company logo detection," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pp. 167–174, Bucharest, Romania, June 2017.

[11] H. Wang, Q. Wang, M. Gao et al., "Multi-scale location-aware kernel representation for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1248–1257, Salt Lake City, UT, USA, June 2018.

[12] H. Li, Y. Huang, and Z. Zhang, "An improved faster R-CNN for same object retrieval," *IEEE Access*, vol. 5, pp. 13665–13676, 2017.

[13] T. Zhou, Z. Li, and C. Zhang, "Enhance the recognition ability to occlusions and small objects with Robust faster R-CNN," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 11, pp. 3155–3166, 2019.

[14] C. Tao, Z. Gao, J. Yan, C. Li, and G. Cui, "Indoor 3D semantic robot VSLAM based on mask regional convolutional neural network," *IEEE Access*, vol. 8, pp. 52906–52916, 2020.

[15] M. Schorghuber, D. Steininger, Y. Cabon et al., "SLAMANTIC-leveraging semantics to improve VSLAM in dynamic environments," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 3759–3768, Seoul, Republic of Korea, October 2019.

[16] T. Laidlo and C. S. Leutenegger, "DeepFusion: real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient predictions," in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, pp. 1231–1240, 2019.

[17] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2263–2270, 2017.

[18] J. McCormac, A. Handa, A. Davison et al., "Semanticfusion: dense 3D semantic mapping with convolutional neural networks," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.

[19] T. Whelan, R. F. Salas-Moreno, B. Glocker et al., "ElasticFusion: real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2019.

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[21] R. Mur-Artal and J. D. Tardos, "Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[22] L. Li, Z. Liu, Ü. Özgüner et al., "Dense 3D semantic SLAM of traffic environment based on stereo vision," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 965–970, Dearborn, MI, USA, 2018.

[23] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2548–2555, Barcelona, Spain, November 2011.

[24] Z. Huang, L. Huang, Y. Gong et al., "Mask scoring RCNN," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019.

[25] C. Forster, Z. Zhang, M. Gassner et al., "SVO: semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[26] P. Jianjian and B. Ruilin, "Variable weight cost aggregation algorithm for stereo matching based on horizontal tree structure," *Acta Optica Sinica*, vol. 38, no. 1, pp. 115–125, 2018.

[27] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3150–3158, Las Vegas, NV, USA, June 2016.

[28] Y. Li, H. Qi, J. Dai et al., "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359–2367, Honolulu, HI, USA, July 2017.

[29] L. C. Chen, A. Hermans, G. Papandreou et al., "Masklab: instance segmentation by refining object detection with semantic and direction features," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4013–4022, Salt Lake City, UT, USA, June 2018.

[30] M. Quigley, K. Conley, B. Gerkey et al., "ROS: an open-source robot operating system," *ICRA Workshop on Open Source Software*, vol. 3, no. 2, pp. 1–5, 2009.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 2479172, 11 pages
https://doi.org/10.1155/2020/2479172

Research Article

Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming

Yan Guo,[1,2] Jin Zhang,[3] Chengxin Yin,[4] Xiaonan Hu,[1] Yu Zou,[1] Zhipeng Xue,[1] and Wei Wang[3]

[1]*College of Information Engineering, Sichuan Agricultural University, Ya'an, Sichuan, China*
[2]*Key Laboratory of Agricultural Information Engineering of Sichuan Province, Sichuan Agricultural University, Ya'an, Sichuan, China*
[3]*College of Management, Sichuan Agricultural University, Ya'an, Sichuan, China*
[4]*College of Management, Chengdu Aeronautic Polytechnic, Chengdu, Sichuan, China*

Correspondence should be addressed to Wei Wang; wangwei@sicau.edu.cn

Received 4 June 2020; Accepted 6 July 2020; Published 18 August 2020

Guest Editor: jinchang ren

The identification of plant disease is the premise of the prevention of plant disease efficiently and precisely in the complex environment. With the rapid development of the smart farming, the identification of plant disease becomes digitalized and data-driven, enabling advanced decision support, smart analyses, and planning. This paper proposes a mathematical model of plant disease detection and recognition based on deep learning, which improves accuracy, generality, and training efficiency. Firstly, the region proposal network (RPN) is utilized to recognize and localize the leaves in complex surroundings. Then, images segmented based on the results of RPN algorithm contain the feature of symptoms through Chan–Vese (CV) algorithm. Finally, the segmented leaves are input into the transfer learning model and trained by the dataset of diseased leaves under simple background. Furthermore, the model is examined with black rot, bacterial plaque, and rust diseases. The results show that the accuracy of the method is 83.57%, which is better than the traditional method, thus reducing the influence of disease on agricultural production and being favorable to sustainable development of agriculture. Therefore, the deep learning algorithm proposed in the paper is of great significance in intelligent agriculture, ecological protection, and agricultural production.

1. Introduction

Plant disease can directly lead to stunted growth causing bad effects on yields [1–3]. An economic loss of up to $20 billion per year is estimated all over the world [4–6]. Diverse conditions are the most difficult challenge for researchers due to the geographic differences that may hinder the accurate identification [7, 8]. In addition, traditional methods mainly rely on specialists, experience, and manuals [9], but the majority of them are expensive, time-consuming, and labor-intensive with difficulty detecting precisely [10]. Therefore, a rapid and accurate approach to identify plant diseases seems so urgent for the benefit of business and ecology to agriculture.

Internet technologies, in particular the availability of multimodality data from various sensors including the Internet of things and sensor networks, have developed rapidly [11]. Herein, a novel plant leaf identification model based on deep learning algorithm is designed to solve the above issues. The function contains leaf retrieval, image segmentation, and identification with the utilization of integrated deep learning algorithm throughout the whole process. The first task is leaf retrieval, but many factors pose the challenge of identification accuracy such as soil and illumination in the complex environment [12]. Hence, the model is investigated RPN algorithm for manipulating retrieval and represents the good adaption in practice. Image segmentation is the second step that is considered to be the most crucial because diagnostic precision plays an important role in detection results. The Chan–Vese algorithm based on region shows promising results for segmenting images free

of noise and weak edge. The last step is to identify the disease of leaves based on the migration learning algorithm. Based on the pretrained model, the migration learning model uses the dataset of disease leaves in a simple background to train the model. The rest of the paper is constructed as follows: Section 2 previews other scholars' researches thoroughly. The detailed information about the model is shown in Section 3. Section 4 demonstrates the procedure of experiment and study. Conclusions and discussions are in Section 5.

2. Literature Review

At present, the research of plant disease recognition in the complex environment mainly focuses on three aspects: disease leaves image segmentation, feature extraction, and disease identification.

2.1. Image Segmentation. In the complex environment, the most crucial task is how to segment the images while localizing and detecting diseased plant leaves, since the major aim of image segmentation is to set the symptom information apart from the background. There are many researchers making a deep investigation on it. In 2017, Ali et al. applied the Delta E color difference algorithm to separate the disease-infected area [13]. In general, four major methods are used to perform the image segmentation which are discussed the detail in the following paragraph [14].

Some researchers integrate the region of interest (ROI) and other methods to segment images. For example, Kao et al. claimed that the convolutional autoencoder served as the background filter to determine the ROI in an image [15]. The second method only concerns region segmentation. In 2013, Pujari et al. claimed that images were divided into various regions which had a special meaning and extracted the images' feature [16]. Akram and other colleagues provided an image processing model with real-time synchronous processing. By dividing the image into three color spaces, it can carry out contrast stretching, feature vector, and salient region recognition [17]. In addition, other researchers chose deep learning techniques to segment and detect images. Marko et al. recommended a depth-based target detection algorithm and used the two-stage algorithm to optimize plant disease images detection [18]. At last, the thresholding is common in segmentation. In 2018, Li et al. applied multilevel thresholding techniques based on gray histogram for image segmentation [19]. Mohamed and Diego presented a new multiobjective metaheuristic on the basis of a multiverse optimization algorithm to segment grayscale images via multilevel thresholding [20].

However, there is a fact that cannot be ignored. Because of the complexity of color information in the complicated environment, the machine vision algorithm based on color, ROI, and threshold performs poorly in practice.

2.2. Feature Extraction. The feature extraction of plant disease faces many problems in identifying plant disease. The distinct image features include textures, shape, color, and motion-related attributes, which are the essential conditions for disease feature extraction [21, 22]. Raza and his colleagues described a method that uses color and texture features to extract disease spots [23]. Hu et al. proposed the Dempster–Shafer (D-S) evidence theory and multifeature fusion for extracting features as well as the results were processed by introducing variance to improve decision rules of D-S evidence theory [24]. In addition, Turkoglu depicted improved versions of the Local Binary Patterns (LBP) methodology, which uses the original LBP local quadratic value to transform the image into grayscale and processes the R and G channels of the image by considering overall and region [25]. Li et al. researched an IoT feature extraction for the intelligent city based on the deep migration learning model [26]. There was an application in music, which can extract meaningful audio features in order to enable the visualizations to be responsive to the music [27]. And in recent studies, lots of novel approaches have been put forward to implementing feature extraction. For example, concerning the challenging task that the extraction of relevant and distinct features from electroencephalogram (EEG), Meziani et al. proposed two new spectral estimators that were robust against non-Gaussian, nonlinear, and nonstationary signals [28]. What is more, as Liu et al. reported, the high-dimensional time–frequency spectrum features were extracted by using the residual neural network and the improved signal-to-clutter ratio (SCR) [29]. Xu et al. introduced a feature extraction method based on the Hilbert marginal spectrum to perform the wear of milling tools [30].

2.3. Disease Identification. As for the precise identification, so many techniques are developed and researched to get accurate results. The identification model focused on using class labels for training images and built a fine-grained image classification system [31]. Zhang et al. reported a recognition method for plant disease leaf images based on a hybrid clustering [32]. In 2017, Patil et al. described a content-based image retrieval (CBIR) system to extract texture features and means value to compute color features, and support vector machine (SVM) classifier was used for classification [33]. Through above researches, the major goal was to design the classification schemes and image analysis for feature extraction and identification. Recently, other new approaches have been introduced to identify the disease more accurately and precisely. A novel system based on the selection of pictures and short text descriptions helped nonexperts in identifying plant diseases that can be used remotely from a desktop as well as in a smart phone or personal digital assistants [34]. Pertot et al. presented a scheme that used mobile phones for real-time on-field imaging of diseased plants and used mobile devices for leaf image segmentation and spotting of disease patch with improved k-means clustering [35]. Yang et al. presented a microscopy image detection methodology based on the synergistic judgment of texture and shape features and the decision tree-confusion matrix [36].

Additionally, the convolutional neural network is numerously utilized in identifying diseases. Chad et al. established a system capable of automatically identifying plant disease in field-acquired images of maize plants [37].

Ni et al. used the deep convolution neural network to train 1632 images of corn kernels and designed an automatic corn detector [38]. Lu et al. proposed a rice diseases identification method based on deep convolutional neural networks (CNNs) techniques [39]. Zhang et al. designed an agricultural machinery image recognition network using the deep learning algorithm [40]. Zhang et al. improved deep convolution neural network to improve the accuracy of maize leaf disease identification [41]. Images were input into two deep learning-based architectures, namely, AlexNet and VGG-16 net, to perform detection [42]. Coulibaly et al. suggested a method using transfer learning for feature extraction to build an identification system [43]. However, due to the requirement for high hardware resources and traditional neural network models of high quality and quantity of data sets in the training process, the training wastes much time that is not conducive to the promotion and use of the model. In this paper, we recommend a transfer learning model for identification combined with the pretrained model, using the dataset of disease leaves to train the model.

From the above research findings, some achievements have been achieved in three aspects: leaf image segmentation, leaf lesion feature extraction, and leaf disease recognition. However, there are still many problems to be solved to realize plant disease identification in the complex environment.

3. Modeling

3.1. The Solution Framework. The full plant disease identification model framework based on deep learning is shown in Figure 1, including three steps, the localization of plant leaves, the segmentation of images, the extraction of plant disease, and the identification of disease. The model used in this paper mainly consists of the following three steps. The first step is to locate the diseased leaves. The RPN algorithm is used to train the leaf dataset in the complex environment, and the frame regression neural network and classification neural network are used to locate and retrieve the diseased leaves in the complex environment. The second step is the segmentation of diseased leaves. The Chan–Vese algorithm is used to segment the image of diseased leaves. Based on the set zero level set and the minimum energy function as the goal, the leaf contour is obtained by iterative calculation, so as to realize the image segmentation of diseased leaves in the complex environment. The third step is the identification of leaf disease species. The pretrained transfer learning model is trained to realize plant disease recognition in the simple background.

3.2. The Leaf Localization. Aiming at the localization of disease-plant leaves, the paper manipulates the leaf dataset under complex background to train the RPN algorithm and integrates boundary regression neural network and classification neural network to perform localization and retrieval.

As for the classification neural network, the core task is to distinguish whether the image in the boundary box is an object or a background. During the process of training, making Intersection over Union (IoU) as a criterion of classification, the boundary box with IoU greater than 0.5 is annotated as an object and the boundary box with IoU less than 0.1 is labeled as a background. IoU is applied in calculating the relevance between predicting boundary box and artificial marked boundary box. The formula of IoU is shown as follows:

$$\mathrm{IoU} = \frac{S_1}{S_2}, \tag{1}$$

where S_1 represents the overlap area of predicting boundary box and artificially marked boundary box, and S_2 represents the total area of it. Due to the fact that the classification neural network is only used for binary classification problem, sigmoid function is employed as loss function.

With regard to the adjustment parameters of boundary regression neural network and output boundary box, one boundary box can be represented by four-dimensional variable (x, y, w, h). (P_x, P_y, P_w, P_h) represents the given boundary box, (G_x, G_y, G_w, G_h) represents the target boundary box, and $(\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h)$ represents the predicting boundary box. In order to find a mapping relationship f of boundary regression neural network, $f(P_x, P_y, P_w, P_h) = (\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h)$ and $(\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h) \approx (G_x, G_y, G_w, G_h)$ are defined.

The movement of boundary consists of pan and zoom.

The parameter of pan is $(\Delta x, \Delta y)$, given that $\Delta x = P_w d_x(P)$ and $\Delta y = P_h d_h(P)$. The formula is shown as

$$\hat{G}_x = P_w d_w(P) + P_x, \tag{2}$$

$$\hat{G}_y = P_h d_h(P) + P_y. \tag{3}$$

The parameter of zoom is (S_w, S_h), given that $S_w = \exp(d_w(P))$ and $S_h = \exp(d_h(P))$. The formula is shown as

$$\hat{G}_w = P_w \exp(d_w(P)), \tag{4}$$

$$\hat{G}_h = P_h \exp(d_h(P)). \tag{5}$$

According to the above formula, the real learning objectives of boundary regression neural network are represented with $d(P) = (d_x(P), d_y(P), d_w(P), d_h(P))$, and the real transform parameters between predicting boundary box and artificially marked boundary box are shown as $t = (t_x, t_y, t_w, t_h)$.

$$t_x = \frac{(G_x - P_x)}{P_x}, \tag{6}$$

$$t_y = \frac{(G_y - P_y)}{P_y}, \tag{7}$$

$$t_w = \log\left(\frac{G_w}{P_w}\right), \tag{8}$$

$$t_h = \log\left(\frac{G_h}{P_h}\right). \tag{9}$$

The objective function of boundary regression neural network is $d(P) = w^T P$, where w represents the learning

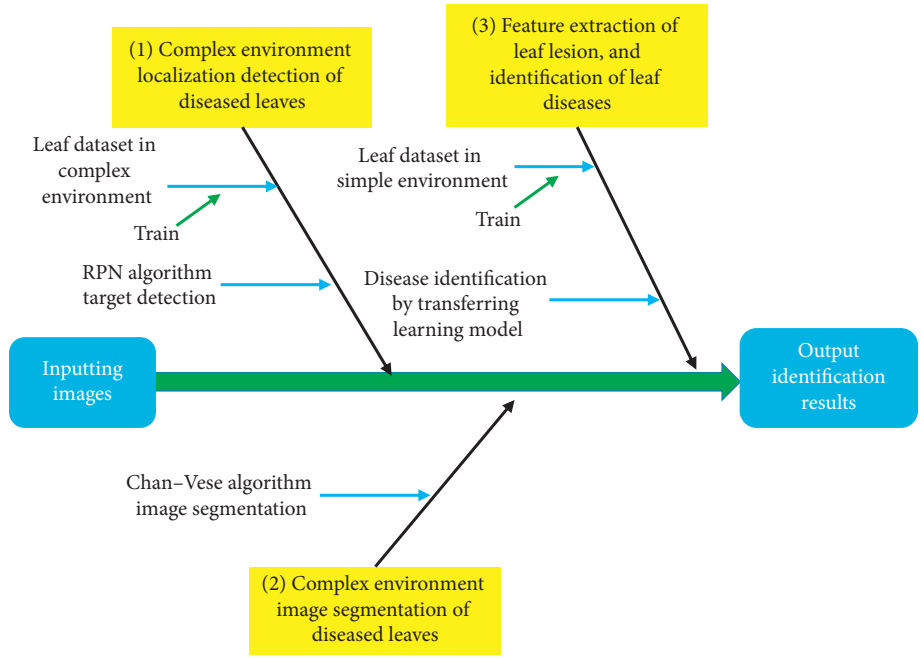


Figure 1: The identification model framework.

parameter of boundary regression neural network. The loss function is shown as follows:

$$\text{Loss} = \sum_{i=1}^{N} (t_i - d_i(P)). \tag{10}$$

3.3. The Leaf Image Segmentation. As for the segmentation of images in complex environment, based on the results of previous step, the model performs the segmentation by Chan–Vese algorithm. Laying the foundation of the set zero level set, aiming at minimizing the energy function and obtaining blade profiles by iterative computing, the model may perform the segmentation of diseased plant leaves images. The Chan–Vese algorithm uses the level set to constructing an energy function to constrain the whole region rather than to control surface evolution by the explicit control speed F. The energy function is defined as the minimum sum of variances between the gray values of the image inside and outside the contour, and the contour length is increased to make it converge. Given a closed curve in the image, the energy function is expressed as follows:

$$\begin{aligned} E = {} & \mu \text{Length}(C) + \lambda_1 \iint_{c_1} |u(x,y) - u_1|^2 dx\, dy \\ & + \lambda_2 \iint_{c_2} |u(x,y) - u_2|^2 dx\, dy. \end{aligned} \tag{11}$$

c_1 represents inside the contour, c_2 represents outside the contour, $u(x, y)$ represents the gray values of the image, u_1 represents the average gray values in contour, and u_2 represents the average gay values out contour. Then, the given formulas are as follows:

$$F_1 = \lambda_1 \iint_{c_1} |u(x,y) - u_1|^2 dxdy, \tag{12}$$

$$F_2 = \lambda_2 \iint_{c_2} |u(x,y) - u_2|^2 dxdy. \tag{13}$$

When $F_1 \approx 0$ and $F_2 \approx 0$, the computing ends.

Level set method is used to solve (11) and zero level set is used to express contour lines. Heaviside's function and Dirac's function are introduced:

$$T = Y, f(X), \tag{14}$$

$$\delta(\varphi) = \frac{dH}{d\varphi}. \tag{15}$$

The level set equation of energy function is as follows:

$$\begin{aligned} E = {} & \eta \iint_{\Omega} \delta(\varphi) |\nabla \varphi| dx\, dy + \lambda_1 \iint_{\Omega} |u(x,y) - u_1|^2 H(\varphi) dx\, dy \\ & + \lambda_2 \iint_{\Omega} |u(x,y) - u_2|^2 (1 - H(\varphi)) dx\, dy. \end{aligned} \tag{16}$$

By minimizing (16) with variational method and combining Euler–Lagrange equation, the following partial differential equations are obtained, where $u_1 = \iint_{\Omega} u(x,y) H(\varphi) dxdy / \iint_{\Omega} H(\varphi) dxdy$ and $u_2 = \iint_{\Omega} u(x,y)\ (1 - H(\varphi))\, dxdy / \iint_{\Omega} (1 - H(\varphi)) dxdy$.

$$\frac{\partial E}{\partial \varphi} = \delta(\varphi) \left(\eta \cdot div \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) \right) + \delta(\varphi) \left[-\lambda_1 (1 - u_1)^2 + \lambda_2 (1 - u_2)^2 \right]. \tag{17}$$

3.4. The Diseased Leaf Identification. In identifying disease types, the paper utilizes the disease leaf dataset training model under simple background to train the pretrained transfer learning model. This method finishes training in a short period of time and performs the disease identification in the simple environment, reducing the requirement of deep learning algorithms for the hardware equipment. Due to the fact that shallow network has similar characteristics for various learning objects, the shallow neural network for source task can be transferred to the neural network for a target task by using the transfer learning algorithm. Transfer learning has better performance in convergences and ultimate results than new learning in practice.

Setting the domain as D, it includes two contents, where X represents feature space and contains all possible characteristic values. $P(X)$ represents a specific feature sampling instance in the feature space:

$$D = X, P(X). \tag{18}$$

Setting the task as T, it includes two parts, where Y represents label space, that is, all vector space consisting of all tags. As prediction function, $f(x)$ is obtained by learning from the features and labels of input data:

$$T = Y, f(X). \tag{19}$$

4. Experimental Study

4.1. The Acquisition of Data. First of all, this study needs to obtain the leaf dataset in the complex environment. The paper employs Crawler technology and obtains 1000 leaf photos from the Plant Photo Bank of China (PPBC), including the leaves of various plants at each growth stage. The shapes of these leaves are different, and the health of the leaves is also different. Aiming at watermark less shelter, obvious leaves, and easy labeling, 189 images are screened out as leaf photos in the complex environment. This dataset is used to train the RPN algorithm to detect and locate the leaf in the complex environment.

Then, using LabelImg, images are quickly annotated and XML files are generated in PASCAL VOC format. It can directly input the target detection neural network as training data. Finally, this study needs to obtain the dataset of diseased leaves in the simple environment. This paper downloads four kinds of images of black rot, bacterial plaque, rust, and healthy leaves from PlantVillage Agricultural Question-and-Answer Forum as training data of transfer learning model, including 537 black rot, 1032 bacterial plaque disease, 293 rust, and 2852 healthy leaves. This dataset is used to train the transfer learning model.

4.2. The Parameter Setup

4.2.1. The Parameter Setup of Leaf Localization. The parameter setup of classification neural network and boundary regression neural network is shown in Tables 1 and 2. The anchors in the table represent the number of candidate boxes generated.

4.2.2. The Parameter Setup of Leaf Segmentation. The main parameter setup of Chan–Vese algorithm is initial zero level set and iteration number setting. In this paper, we set the initial zero level set as a circle with the center of the picture and one-third of the diagonal length of the picture as the radius and set up the Chan–Vese algorithm to calculate 500 iterations.

The image obtained by RPN algorithm is input into the Chan–Vese algorithm and the image in the zero level set is preserved. The image outside the zero level set is set to black to get the image segmentation result.

4.2.3. The Parameter Setup of Leaf Retrieval. Resnet-101 is selected as the pretraining model, and the network is trained by using the dataset of disease leaves under a simple background in this paper. Its network parameters are shown in Table 3.

In this paper, all its parameters are modified and initialized in the last output layer of Resnet-101, and the classification number is changed from 1000 to 4, which corresponds to the identification results of four kinds of leaf diseases.

5. Results

5.1. The Result of Leaf Localization. The test image is input into VGG-16 model and RPN algorithm, and the results are shown in Figure 2.

With regard to the above images, there is inaccuracy of the frame selection range in Figure 2(b) and the blades in Figure 2(d) are missing. But RPN algorithm can basically frame the main blade structure, which has better performance than the original model.

5.2. The Result of Leaf Segmentation. The iterative calculation process of Chan–Vese algorithm is shown in Figures 3–6. The results of Chan–Vese algorithm compared with watershed algorithm are shown in Figure 7.

According to the above results, it can be found that after 500 iterations, Chan–Vese algorithm can get better leaf image segmentation results. Although Chan–Vese algorithm cannot effectively extract the edge contour of the blade compared with the watershed algorithm, it retains the complete structure of the central blade including leaf venation, spot color, and spot shape. The complete central structure of the blade obtained by Chan–Vese algorithm can be used for disease identification of the next step.

5.3. The Result of Disease Identification. The test image is input into VGG-16 model and RPN algorithm, and the results are shown in Figure 8.

According to the above pictures, although it can be found that the frame selection range in Figure 8(b) is not accurate enough and there are omissions in Figure 8(d), RPN algorithm can basically frame the main blade structure and can be used for the next operation.

Table 1: The composition and parameter setup of neural network.

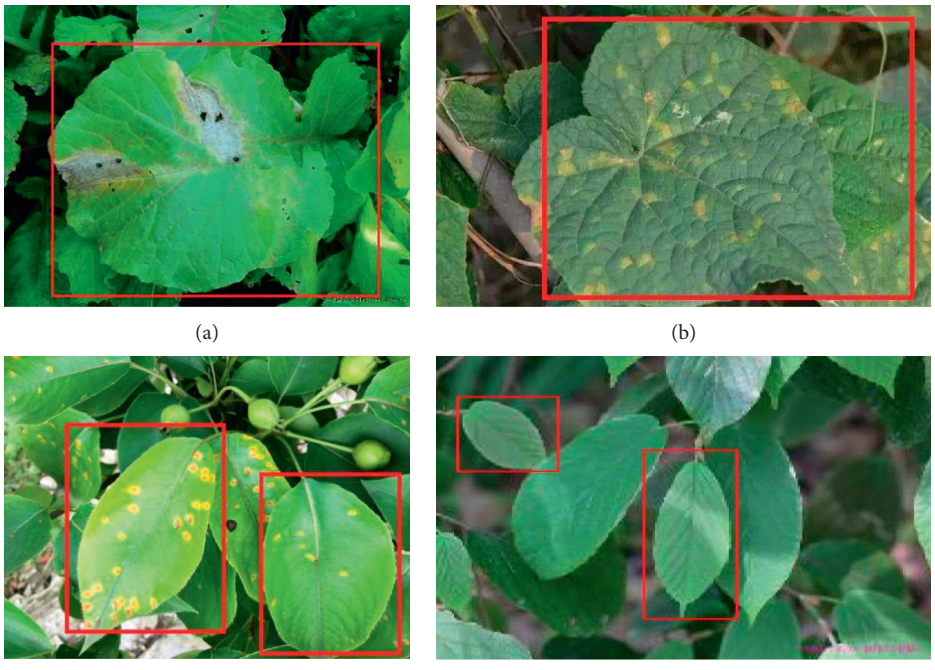
Network layer	Number of kernels	Size of kernel	Output shape	Number of parameters
Convolution	512	(3, 3)	(14, 14, 512)	2359296
Convolution	Anchors * 2	(1, 1)	—	—
Softmax	4096	—	—	—

Table 2: The composition and parameter setup of regression network.

Network layer	Number of kernels	Size of kernel	Output shape	Number of parameters
Convolution	512	(3, 3)	(14, 14, 512)	2359296
Convolution	Anchors * 4	(1, 1)	—	—

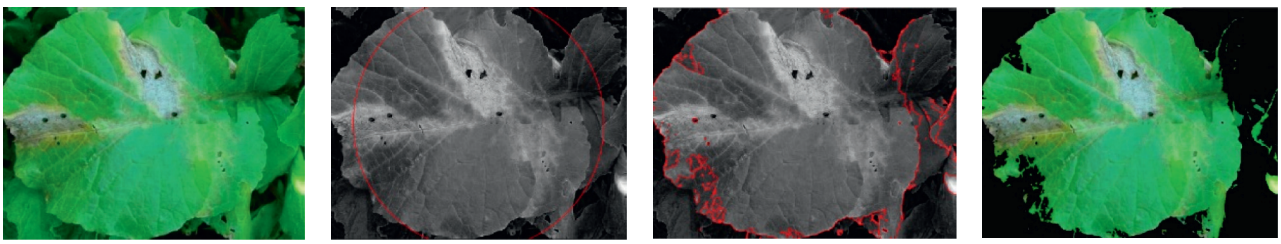
Table 3: The composition and parameter setup of ResNet-101.

Network layer	Number of kernels	Size of kernel	Output shape	Number of parameters
Convolution	64	(7, 7)	(112, 112, 64)	9408
Maxpooling	—	(2, 2)	(56, 56, 64)	0
5 * Convolution	64	(3, 3)	(56, 56, 64)	36864
Convolution	128	(3, 3)	(28, 28, 128)	73728
7 * Convolution	128	(3, 3)	(56, 56, 128)	147456
Convolution	256	(3, 3)	(28, 28, 256)	294912
11 * Convolution	256	(3, 3)	(28, 28, 256)	589824
Convolution	512	(3, 3)	(14, 14, 512)	1179648
5 * Convolution	512	(3, 3)	(14, 14, 512)	2359296
Averagepooling	—	(2, 2)	(7, 7, 512)	0
Softmax	1000	(7, 7)	(1, 1, 1000)	25088000



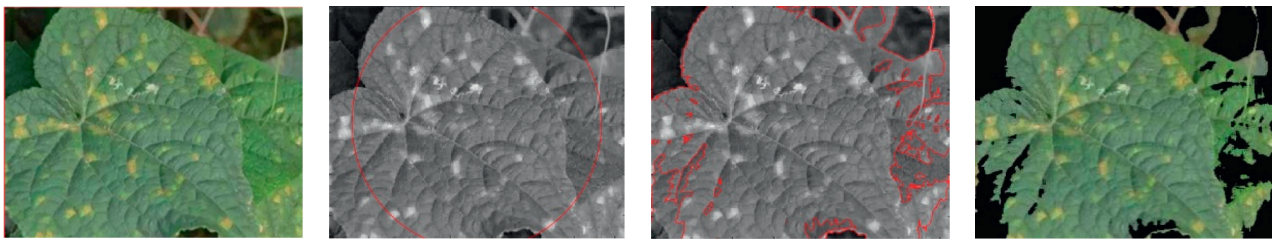
(a) (b) (c) (d)

Figure 2: The result of leaf identification: (a) black rot disease; (b) bacteria plaque disease; (c) rust disease; (d) healthy leaf.



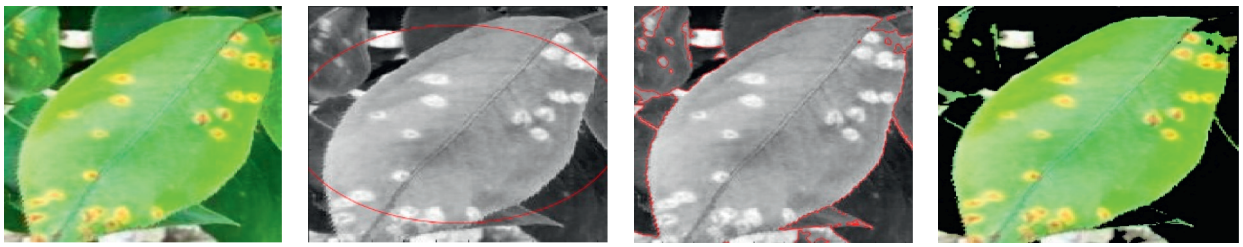
(a) (b) (c) (d)

Figure 3: The result of Chan–Vese algorithm segmenting black rot diseased leaf: (a) image capture; (b) initial zero level set; (c) contour image after 500 iterations; (d) segmentation results.



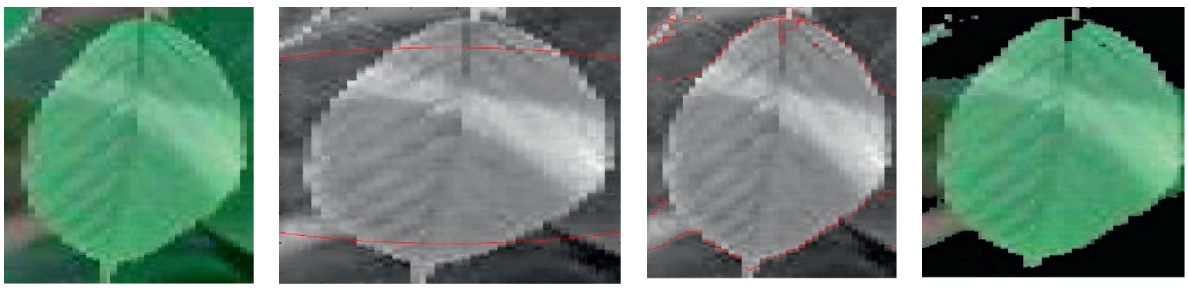
(a) (b) (c) (d)

Figure 4: The result of Chan–Vese algorithm segmenting bacterial plaque diseased leaf: (a) image capture; (b) initial zero level set; (c) contour image after 500 iterations; (d) segmentation results.



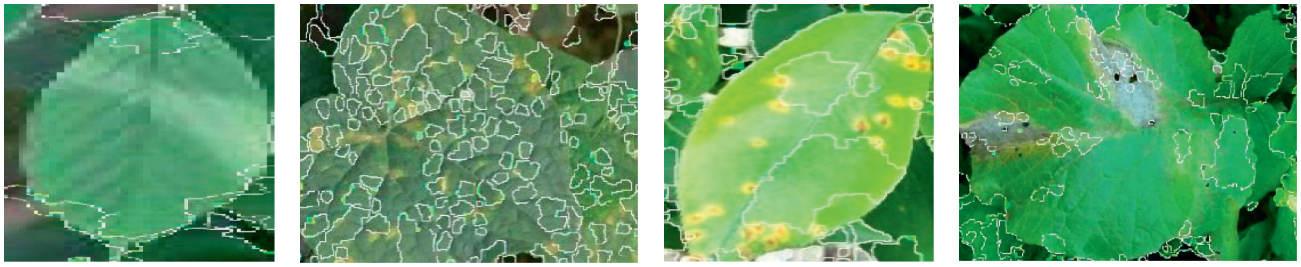
(a) (b) (c) (d)

Figure 5: The result of Chan–Vese algorithm segmenting rust diseased leaf: (a) image capture; (b) initial zero level set; (c) contour image after 500 iterations; (d) segmentation results.



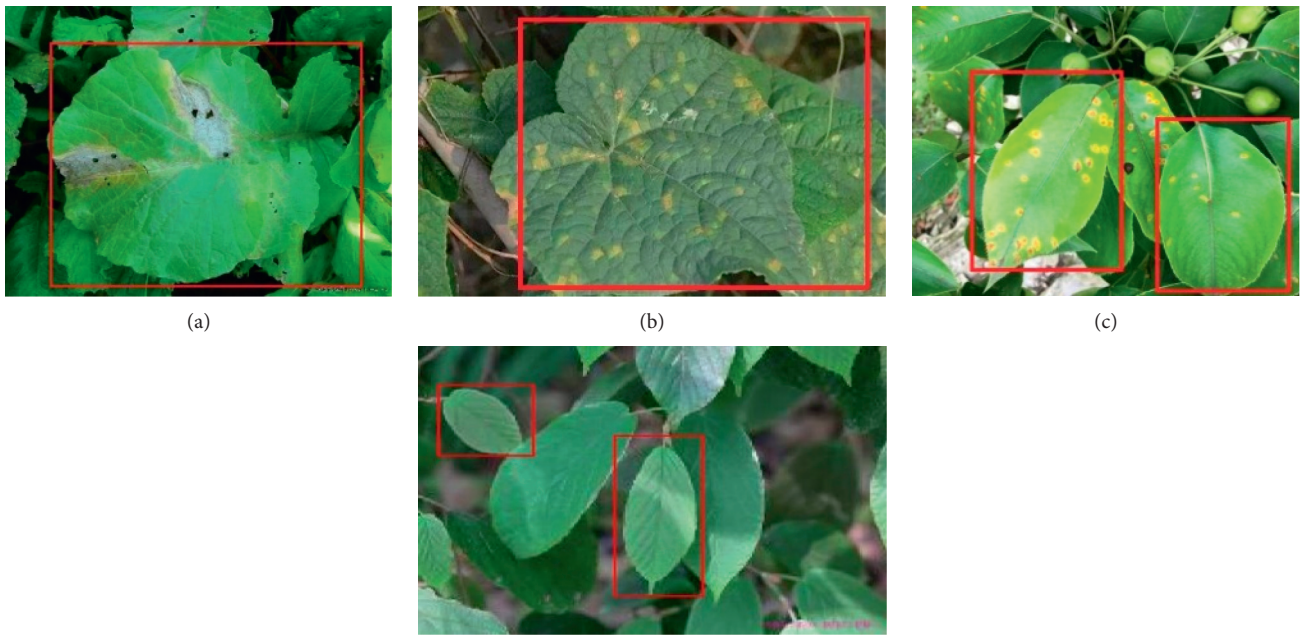
(a) (b) (c) (d)

Figure 6: The result of Chan–Vese algorithm segmenting healthy leaf: (a) image capture; (b) initial zero level set; (c) contour image after 500 iterations; (d) segmentation results.



(a) (b) (c) (d)

Figure 7: The result of watershed algorithm: (a) black rot disease; (b) bacteria plaque disease; (c) rust disease; (d) healthy leaf.



(a) (b) (c)

(d)

Figure 8: Leaf retrieval results in complex environment: (a) black rot disease; (b) bacteria plaque disease; (c) rust disease; (d) healthy leaf.

5.4. Comparative Test. In this paper, the parameters of the transfer learning model include gradient descent optimization parameters and training parameters. The specific parameters are set as shown in Table 4.

After 4000 iterations of training, the loss value and training set accuracy of the transfer learning model and the traditional model are shown in Figure 9.

In Figure 9(a), ResNet-101 represents traditional model. According to Figure 9(a), after the same 4000 iterations training, the transfer learning model has faster convergence speed and lower model loss value than the traditional model. According to Figure 9(b), it can be found that in the process of model training, transfer learning has higher accuracy, lower variance, and better recognition effect than new learning.

Therefore, compared with the new learning, this paper uses the transfer learning to converge faster and achieve better model identification effect. It can meet the requirements of smart agriculture for low hardware resources, fast training time, and high training efficiency.

Then, the image is input into transfer learning model based on the segmentation of Chan–Vese algorithm. As a contrast, the image that has not been processed in this paper is input into the traditional ResNet-101 model for identification, and the results are shown in Table 5.

According to the comparison results in the above table, the average correct rate of the proposed method is 83.75%, which is significantly better than that of the traditional ResNet-101 model (42.5%). Comparing the performances of this method in four samples, we can find that rust and healthy leaves can get better results than black rot and bacterial plaque.

6. Discussion and Conclusion

This paper shows that the plant disease recognition model based on deep learning has the characteristics of unsupervised, high accuracy, good universality, and high training efficiency. However, there are many challenges in accuracy practicability of plant disease detection in the complex

Table 4: The parameter setup of transfer learning.

The type of parameter	The name of parameter	The setup of parameter
Gradient descent optimization parameter	Learning rate	0.001
	Weight decay	0.0005
	Learning impulse	0.9
	Decay of learning rate	0.1
Input data parameters	Picture size	(224, 224)
	Batch size	256
	Iteration times	30000

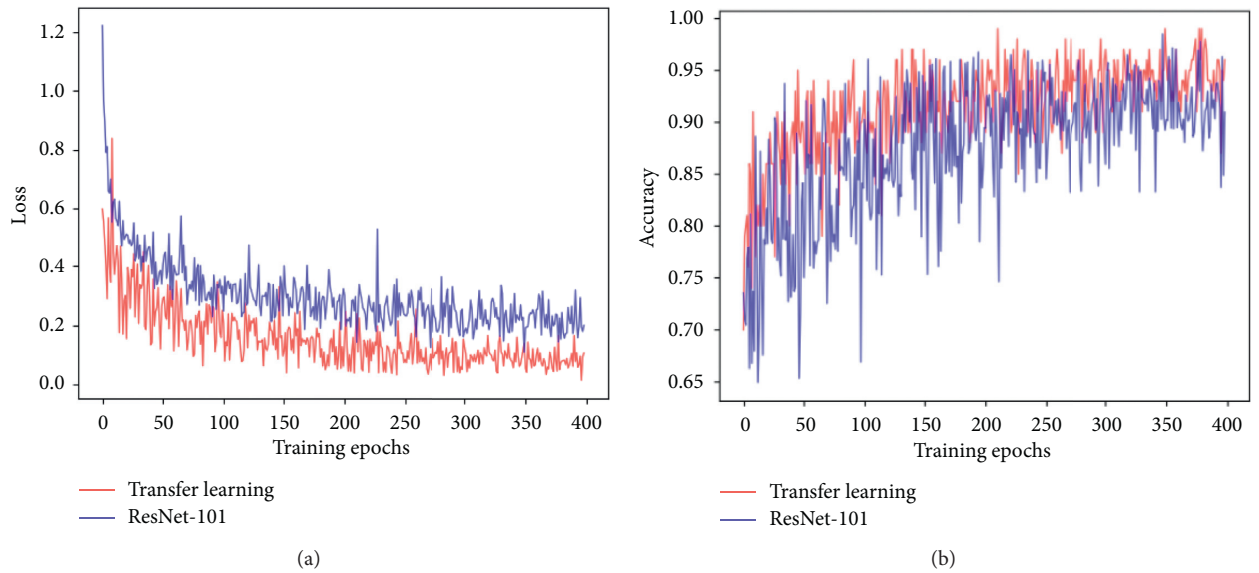


Figure 9: The comparison between transfer learning and traditional learning: (a) loss value comparison; (b) accuracy comparison.

Table 5: The comparison between the proposed method and ResNet-101 model.

The type of disease	The proposed method		ResNet-101 model	
	The correct number	The correct rate (%)	The correct number	The correct rate (%)
Black rot disease	15	75	8	40
Bacterial plaque disease	16	80	6	30
Rust disease	18	90	9	45
Healthy	18	90	11	55
Total	67	83.75	34	42.5

environment. In order to solve these problems and optimize the identification method, this paper proposes a recognition model integrating RPN algorithm, CV algorithm, and TL algorithm, which can effectively solve the problem of plant disease identification in the complex environment. The model not only adapts to complex environments, but also increases the accuracy of identification. Compared with the traditional model, the model proposed in this paper not only guarantees the robustness of the convolutional neural network, but also reduces the number and quality requirements of the convolutional neural network on the data set and obtains better results. Therefore, the model could help agricultural production personnel to prevent and cure the plant disease quickly. The model which overcomes the problem of environment complexity can get an accurate identification result in practical application. Furthermore, this study enriches the existing theory and helps to improve the accuracy. At the same time, it is of great significance for the study of plant disease identification in the field of environmental complexity and helps researchers pay attention to the important role of environmental complexity in plant disease identification. Therefore, the model applies information technology to agricultural production and is favorable to sustainable development of smart agriculture.

Although the plant disease identification model based on deep learning proposed in this paper can overcome the complexity of the environment and improve the accuracy of identification, there are still some problems to be pointed out. For example, the Chan–Vese algorithm needs repetitive iterative calculation and runs for a long time, which is not

conducive to the fast identification results of this method. In future research, we will use the neural network to generate zero initial set corresponding to different leaves, which will increase the end of calculation limit for the iterative process of Chan–Vese algorithm, speed up the training speed, and end the iteration ahead of time.

Data Availability

Data were made available with the help of the Key Laboratory of Agricultural Information Engineering of Sichuan Province. The data used to support the findings of this study were provided by the laboratory under license. Access to these data will be considered by the corresponding author upon request, with permission of the laboratory.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Key Laboratory of Agricultural Information Engineering of Sichuan Province and Social Science Foundation of Sichuan Province in 2019 (19GL030).

References

[1] Y. Ampatzidis, L. De Bellis, and A. Luvisi, "iPathology: robotic applications and management of plants and plant diseases," *Sustainability*, vol. 9, no. 6, p. 1010, 2017.

[2] A. Breukers, D. L. Kettenis, M. Mourits, W. V. D. Werf, and A. O. Lansink, "Individual-based models in the analysis of disease transmission in plant production chains: an application to potato brown rot," *Academy of Sciences*, vol. 90, no. 1–3, pp. 112–131, 2006.

[3] S. Ghosal, D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh, and S. Sarkar, "An explainable deep machine vision framework for plant stress phenotyping," *Proceedings of the National Academy of Sciences*, vol. 115, no. 18, pp. 4613–4618, 2018.

[4] E.-C. Oerke, "Crop losses to pests," *The Journal of Agricultural Science*, vol. 144, no. 1, pp. 31–43, 2006.

[5] X. E. Pantazi, D. Moshou, and A. A. Tamouridou, "Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers," *Computers and Electronics in Agriculture*, vol. 156, pp. 96–104, 2019.

[6] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems Engineering*, vol. 172, pp. 84–91, 2018.

[7] G. Geetharamani and J. Arun Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Computers & Electrical Engineering*, vol. 76, pp. 323–338, 2019.

[8] P. F. Konstantinos, "Deep learning models for plant disease detection and diagnosis," *Computers & Electrical Engineering*, vol. 145, pp. 311–318, 2018.

[9] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41–49, 2017.

[10] S. P. Mohanty, D. P. Hughes, and S. Marcel, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

[11] Y. Guo, X. Hu, Y. Zou et al., "Maximizing E-tailers' sales volume through the shipping-fee discount and product recommendation system," *Discrete Dynamics in Nature and Society*, vol. 2020, pp. 1–14, 2020.

[12] R. Amanda, B. Kelsee, M. C. Peter, A. Babuali, L. James, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers in Plant Science*, vol. 8, p. 1852, 2017.

[13] H. Ali, M. I. Lali, M. Z. Nawaz, M. Sharif, and B. A. Saleem, "Symptom based automated detection of citrus diseases using color histogram and textural descriptors," *Computers and Electronics in Agriculture*, vol. 138, pp. 92–104, 2017.

[14] H. M. Alexander, K. E. Mauck, A. E. Whitfield, K. A. Garrett, and C. M. Malmstrom, "Plant-virus interactions and the agro-ecological interface," *European Journal of Plant Pathology*, vol. 138, no. 3, pp. 529–547, 2014.

[15] I.-H. Kao, Y.-W. Hsu, Y.-Z. Yang, Y.-L. Chen, Y.-H. Lai, and J.-W. Perng, "Determination of Lycopersicon maturity using convolutional autoencoders," *Scientia Horticulturae*, vol. 256, p. 108538, 2019.

[16] D. Pujari, R. Yakkundimath, and A. S. Byadgi, "Grading and classification of anthracnose fungal disease of fruits based on statistical texture features," *International Journal of Advanced Science and Technology*, vol. 52, pp. 121–132, 2013.

[17] T. Akram, S. R. Naqvi, S. A. Haider, and M. Kamran, "Towards real-time crops surveillance for disease classification: exploiting parallelism in computer vision," *Computers & Electrical Engineering*, vol. 59, pp. 15–26, 2017.

[18] A. Marko, K. Mirjana, S. Srdjan, A. Andras, and S. Darko, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry-Baseline*, vol. 11, no. 7, p. 939, 2019.

[19] J. Li, W. Tang, J. Wang, X. Wang, and X. Zhang, "Multilevel thresholding selection based on variational mode decomposition for image segmentation," *Signal Processing*, vol. 147, pp. 80–91, 2018.

[20] M. A. Elaziz, D. Oliva, A. A. Ewees, and S. Xiong, "Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer," *Expert Systems with Applications*, vol. 125, pp. 112–129, 2019.

[21] A. Cruz, Y. Ampatzidis, R. Pierro et al., "Detection of grapevine yellows symptoms in Vitis vinifera L. with artificial intelligence," *Computers and Electronics in Agriculture*, vol. 157, pp. 63–76, 2019.

[22] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. ur Rehman, and K. Javed, "An automated detection and classification of citrus plant diseases using image processing techniques: a review," *Computers and Electronics in Agriculture*, vol. 153, pp. 12–32, 2018.

[23] M. Raza, M. Sharif, M. Yasmin, M. A. Khan, T. Saba, and S. L. Fernandes, "Appearance based pedestrians' gender recognition by employing stacked auto encoders in deep learning," *Future Generation Computer Systems*, vol. 88, pp. 28–39, 2018.

[24] M. Hu, X. Bu, X. Sun, Z. Yu, and Y. Zheng, "Rape plant disease recognition method of multi-feature fusion based on D-S evidence theory," *Mathematical and Computational Applications*, vol. 22, no. 1, p. 18, 2017.

[25] M. Turkoglu and D. Hanbay, "Leaf-based plant species recognition based on improved local binary pattern and extreme

learning machine," *Physica A: Statistical Mechanics and Its Applications*, vol. 527, p. 121297, 2019.

[26] D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *International Journal of Information Management*, vol. 49, pp. 533–545, 2019.

[27] Dhiraj, R. Biswas, and N. Ghattamaraju, "An effective analysis of deep learning based approaches for audio based feature extraction and its visualization," *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 23949–23972, 2019.

[28] A. Meziani, K. Djouani, T. Medkour, and A. Chibani, "A Lasso quantile periodogram based feature extraction for EEG-based motor imagery," *Journal of Neuroscience Methods*, vol. 328, p. 108434, 2019.

[29] Y. Xu, H. Ding, Y. Xue, and J. Guan, "High-dimensional feature extraction of sea clutter and target signal for intelligent maritime monitoring network," *Comput. Commun.*vol. 147, pp. 76–84, 2019.

[30] C. Xu, Y. Chai, H. Li, Z. Shi, L. Zhang, and Z. Liang, "A feature extraction method for the wear of milling tools based on the Hilbert marginal spectrum," *Machining Science and Technology*, vol. 23, pp. 847–868, 2019.

[31] Y. Zhang, X.-S. Wei, J. Wu et al., "Weakly supervised fine-grained categorization with part-based image representation," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1713–1725, 2016.

[32] S. Zhang, Z. You, and X. Wu, "Plant disease leaf image segmentation based on superpixel clustering and EM algorithm," *Neural Computing and Applications*, vol. 31, no. S2, pp. 1225–1232, 2019.

[33] J. K. Patil and R. Kumar, "Analysis of content based image retrieval for plant leaf diseases using color, shape and texture features," *Engineering in Agriculture, Environment and Food*, vol. 10, pp. 69–78, 2016.

[34] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.

[35] I. Pertot, T. Kuflik, I. Gordon, S. Freeman, and Y. Elad, "Identificator: a web-based tool for visual plant disease identification, a proof of concept with a case study on strawberry," *Computers and Electronics in Agriculture*, vol. 84, pp. 144–154, 2012.

[36] N. Yang, Y. Qian, H. S. EL-Mesery, R. Zhang, A. Wang, and J. Tang, "Rapid detection of rice disease using microscopy image identification based on the synergistic judgment of texture and shape features and decision tree-confusion matrix method," *Journal of the Science of Food and Agriculture*, vol. 99, no. 14, pp. 6589–6600, 2019.

[37] D. Chad, W. H. Tyr, S. Chen et al., "Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning," *Phytopathology*, vol. 107, pp. 1426–1432, 2017.

[38] C. Ni, D. Wang, R. Vinson, M. Holmes, and Y. Tao, "Automatic inspection machine for maize kernels based on deep convolutional neural networks," *Biosystems Engineering*, vol. 178, pp. 131–144, 2019.

[39] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.

[40] Z. Zhang, H. Liu, Z. Meng, and J. Chen, "Deep learning-based automatic recognition network of agricultural machinery images," *Computers and Electronics in Agriculture*, vol. 166, p. 104978, 2019.

[41] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access*, vol. 6, pp. 30370–30377, 2018.

[42] R. A. Krishnaswamy, P. Raja, and R. Aniirudh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia Computer Science*, vol. 133, pp. 1040–1047, 2018.

[43] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Computers in Industry*, vol. 108, pp. 115–120, 2019.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 4013185, 13 pages
https://doi.org/10.1155/2020/4013185

Research Article

A Crash Severity Prediction Method Based on Improved Neural Network and Factor Analysis

Chen Zhang,[1,2] Jie He,[1] Yinhai Wang,[2] Xintong Yan,[1] Changjian Zhang,[1] Yikai Chen,[3] Ziyang Liu,[1] and Bojian Zhou[1]

[1]*School of Transportation, Southeast University, Sipailou 2#, Nanjing 210018, Jiangsu, China*
[2]*Smart Transportation Applications and Research Laboratory, University of Washington, Seattle, WA 98195-2700, China*
[3]*School of Automobile and Traffic Engineering, Hefei University of Technology, Hefei 230009, China*

Correspondence should be addressed to Jie He; hejie@seu.edu.cn

Received 25 April 2020; Accepted 4 June 2020; Published 30 June 2020

Guest Editor: Longzhuang Li

Crash severity prediction has been raised as a key problem in traffic accident studies. Thus, to progress in this area, in this study, a thorough artificial neural network combined with an improved metaheuristic algorithm was developed and tested in terms of its structure, training function, factor analysis, and comparative results. Data from I5, an interstate highway in the Washington State during the period of 2011–2015, were used for fitting and prediction, and after setting the theoretical three-layer neural network (NN), an improved Particle Swarm Optimization (PSO) method with adaptive inertial weight was offered to optimize the NN, and finally, a comparison among different adaptive strategies was conducted. The results showed that although the algorithms produced almost the same accuracy in their predictions, a backpropagation method combined with a nonlinear inertial weight setting in PSO produced fast global and accurate local optimal searching, thereby demonstrating a better understanding of the entire model explanation, which could best fit the model, and at last, the factor analysis showed that non-road-related factors, particularly vehicle-related factors, are more important than road-related variables. The method developed in this study can be applied to a big data analysis of traffic accidents and be used as a fast-useful tool for policy makers and traffic safety researchers.

1. Introduction

1.1. Crash Severity. Traffic safety is a challenging task to be accomplished and has been identified as crash hotspots around the world. The total number of fatal crashes in the U.S. increased to around 35,000 in 2016. In addition, according to the Washington State Collision Summary report, a total of 117,053 crashes were identified in the Washington State, including 499 fatal collisions, 36,531 injury collisions, and 77,358 property-damage-only collisions, indicating a crash occurred every 4.5 min and a person died in a crash every 16 hours [1]. Figure 1 shows the traffic fatality rates between the U.S. and Washington State [1]. Billions of dollars in personal and property damage are wasted in traffic crashes each year around the world [2].

To achieve the intrinsic goal of exploring numerous factors that trigger the crash, crash severity is often used for crash analyses to represent the degree of injury. A KABCO scale was proposed by WSDOT to represent the level of the injury: K—fatal injury; A—incapacitating-injury; B—nonincapacitating injury; C—minor injury; and O—property-damage-only injury. The KABCO scale has been widely adopted and adapted by many scholars (e.g., [3–5]). In this paper, the predicted targets were regrouped into three categories based on the combination of the KABCO scale and crash severity category of 5-year data (2011–2015) obtained from the HISI data system, namely, incapacitating-injuries, injuries, and noninjuries.

1.2. Crash Severity Prediction. Crash prediction problems have long been a popular area of study around the world. Numerous studies conducted the prediction analyses based on classic statistical models, e.g., the linear, nonlinear,

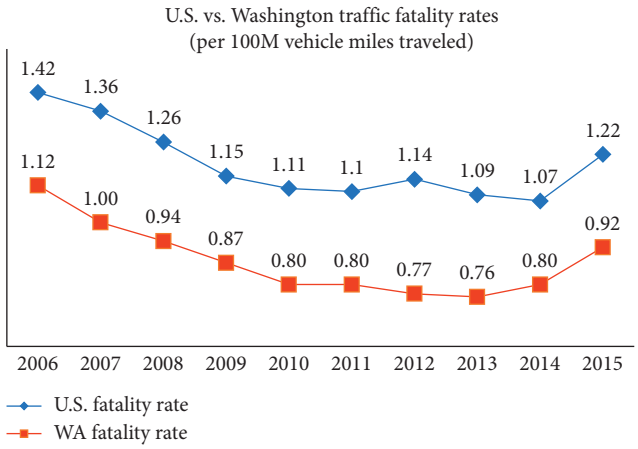


Figure 1: Traffic fatality rates in the U.S. vs. Washington State (per 100M vehicle miles traveled). Source: CLAS (WSDOT) and FARS (WTSC).

generalized linear model (GLM), generalized estimating equation (GEE), nominal binary (NB), and Poisson regression models, which are regarded as a good attempt at thoroughly formulating the relationship between tens or hundreds of explanative variables. However, it should be noted that the traditional statistical method has its limitation. The artificial intelligence (AI) technique, particularly, the deep learning methodology [6], as an emerging but promising tool for addressing the problems faced by the traditional statistical domain, deserves more attention and exploration.

Safety performance functions (SPFs) are frequently utilized to demonstrate the relationships between different crashes and crash impact parameters. Such functions usually use the crash frequency as the targeting variable. The Highway Safety Manual [7] has several chapters demonstrating the average crash frequency of an entire network, facility, or individual site. Elvik et al. [8] developed six power functions to demonstrate the relationship between speed and road safety; these six equations are slightly adjusted according to the road characteristics under the following categories: fatal injuries, fatal and serious injuries, all injuries, fatal accidents, fatal and serious accidents, and all injury accidents. Russo et al. [9] used a negative binomial regression model to develop four sets of SPFs based on the crash frequency and conducted a residual analysis to prove the accuracy. Park and Abdel-Aty [10] developed several crash modification factors (CMFs) for a combination of traffic and roadway cross-sectional elements on noncurved and curved roadway sections using a cross-sectional method and found that CMFs for increasing the lane and shoulder width were decreasing as the annual average daily traffic (AADT) level increased. Using a Bayesian ranking technique, Ahmed et al. [11] examined the safety effects of the roadway geometry on the crash occurrence rate along a freeway section that features mountainous terrain and an adverse weather condition and confirmed that segments with steep downgrades are more crash-prone along the studied section.

A number of researchers are eager to dig into the use of statistical analysis for traffic safety research, such as linear, nonlinear, GLM, GEE, NB, or Poisson regression models. Such models have performed well when the number of explanatory variables is constrained. Debrabant et al. [12] applied an autoregressive Poisson–Tweedie model to mine spatially and temporally aggregated hospital records of traffic accidents, and it was confirmed that this method is very accurate when applied to a black spot identification problem. Pei et al. [13] developed a joint probability model combined with a Markov chain, Monte Carlo (MCMC) approach, and full Bayesian to estimate the effects of explanatory factors, and their results indicated that the model achieves a good statistical fit and provides an accurate analysis of the influences of the explanatory factors. El-Basyouny et al. [14] applied a multivariate model based on a MCMC simulation to address the impact of weather elements, and their results showed that temperature and snowfall are statistically significant with intuitive signs for all crash types, whereas rainfall is mostly insignificant, as is the maximum wind gust with a few exceptions that are positively related to the crash type.

To address the shortcomings of the traditional statistical model, scholars in the crash analysis field are more willing to use AI methods at present. Karlaftis and Vlahogianni [15] discussed the differences and similarities between a statistical method and neural networks (NNs), and their results showed that the goals of the analysis are more important than the tools used, and that there are always assumptions to all modeling approaches. Specific to a traffic accident analysis, although classic statistical models such as NB, Poison, and a Bayesian network can achieve a good identification of a broad range of risk factors, they are also limited to a finite factor assumption as compared with a deep-learning method [16]. Aided by the powerful hardware and software of modern computers, deep-learning methods are becoming powerful tools for many aspects of our daily lives [6]. For example, in a crash analysis, Zeng and Huang [17] used a pruned NN for the crash severity, adopting a convex combination (CC) training algorithm and a NN pruning for a function approximation (N2PFA) structure optimization method, and found that the CC outperforms the backpropagation (BP) method in both convergence ability and training speed; in addition, simplification of the nodes in an NN structure can obtain a better performance. Huang et al. [18] developed an optimized radial basis function neural network (RBFNN) model to analyze the relationships between crash frequency and the relevant risk factors, and their comparative work showed that RBFNN models outperform negative binomial models and back-propagation neural network (BPNN) models. Li et al. [5] developed a data-driven method combining the non-dominated sorting genetic algorithm (NSGA-II) with an NN to identify the key factors in a fatal highway crash analysis. All of the abovementioned studies have focused more on the NN structure itself, using complicated mathematical equations to illustrate their abstract concepts; nevertheless, they have seldom given a thoughtful, detailed, and general

procedure to deal with a complete traffic severity analysis problem.

However, in short, both statistical and AI methods in the previous studies are still facing some challenges, and especially for neural networks, the traditional NNs are more easily stuck in the local optimum in accordance with its random weights initialization at the very first beginning. Although some studies [5, 18] were carried out to address the abovementioned problems, the efficiency of the global optimum search for particle swarm optimization (PSO) and local optimum search for some other optimization methods is still to be improved.

To solve the aforementioned problems, the purpose of this paper is as follows: (1) to provide a BPNN algorithm integrating the PSO with adaptive inertial weights for the establishment of the crash severity prediction model; (2) to conduct a detailed factor analysis (FA) based on the refined model to quantify the internal relationship and heterogeneity of different variables that trigger the crash of distinguished severity. The prediction target in the first phase is the crash severity. It should be noted that the novelty of the paper lies in the fact that an integrated method incorporating an emerging AI technique and a traditional statistical model was provided for crash severity analysis. Besides, it is marginally original to use FA and PSO for the calculation of the parameters of the crash triggers.

This paper is organized as follows: in the first part, the background and severity levels are discussed to demonstrate the reason for conducting this research. In the second part, classic statistical models and NN models dealing with crash analysis are reviewed to illustrate their advantages and limitations. The following section demonstrates the processing of the dataset, including its description and simplification. In the fourth part, the entire methodology for the developed model and data incorporation are presented to highlight the process of conducting a severity prediction process. Finally, the results are presented with the conclusion.

2. Data Description

The dataset used in this study consisted of the data acquired from the Highway Safety Information System (HSIS); in this system, data from nine states (California, Washington, Minnesota, Michigan, Maine, Ohio, North Carolina, and Illinois) in the U.S. are available. Considering the author's time studying in the Washington State during the period of 2016 to 2017, the crash data from this were selected as the target data.

The HSIS data contained, roughly, two tables for the variables. The first one is related to Accident, Vehicle, and Occupant files, which involve TIME, ENVIRONMENT, ACCIDENT-RELATED INFORMATION, VEHICLE INFORMATION, DRIVER INFORMATION, OCCUPANT, ROADWAY ELEMENTS, and PEDESTRIAN/BICYCLIST INFORMATION, whereas the second table was more concerned with the roadway containing LOCATION/LINKAGE ELEMENTS, ROADWAY CLASSIFICATION, ROAD ALIGNMENT, CROSS SECTION, ROAD FEATURES, TRAFFIC CONTROL/OPERATIONS, and TRAFFIC DATA. In addition, there were tens of subvariables for both the tables.

The crash data from I5 in the Washington State covering the years 2011–2015 were extracted. The data from the first four years, 2011–2014, were used to fit the model, and the data from the later year were used as the prediction validation set. The total crashes were 9926, 10083, 10127, 11628, and 12804 from 2011–2015. Based on these raw samples, the following steps need to be conducted before digging into the model input procedure:

(1) Exclude apparently irrelevant variables. More than 40 features were requested from the HSIS database system, and some features, such as "CASENO" (accident case number), "MILEPOST" (milepost), "RD_INV" (a linkage variable on the Accident file which is used in the merging operation), and "RTE_NBR" (route number) are not related to the crash severity and were omitted for simplicity.

(2) Samples with features such as "LIGHT," "WEATHER," and "ACCTYPE" (accident type) which have values such as "UNKNOWN," "NAN," "UNSTATED," and "NULL" were also omitted for simplicity.

(3) Some nominal variables which cannot be denoted by the continuous number such as "DIR_CURV" (the horizontal curve direction) and "DIR_GRAD" (the vertical curve grade direction), both representing the relative direction of left or right, were transformed into discrete scale values ("1" or "0").

(4) The vehicle-related and driver-related variables such as "DRV_AGE" (driver age) and "DRV_SEX" (driver sex) were incorporated with accident-related data files through the "CASENO" label, whereas the grad/curve-related variables were incorporated with accident-related data through "MILEPOST"; here, a data process computer program written in MATLAB was developed to locate the "MILEPOST" between "BEGPOST" and "ENDPOST" in the grad/curve files.

(5) "VEHYR," which indicates the vehicle model year, was transformed into the vehicle operation year through the following formula:

$$\mathrm{Vehyr}_m^i = \begin{cases} 100 - \mathrm{Vehyr}_{\mathrm{raw}}^i + \mathrm{val}\left(\mathrm{year}^i\right) \\ \quad \left(\mathrm{Vehyr}_{\mathrm{raw}}^i > \mathrm{val}\left(\mathrm{year}^i\right)\right), \\ \mathrm{val}\left(\mathrm{year}^i\right) - \mathrm{Vehyr}_{\mathrm{raw}}^i \\ \quad \left(\mathrm{Vehyr}_{\mathrm{raw}}^i \le \mathrm{val}\left(\mathrm{year}^i\right)\right), \end{cases} \tag{1}$$

where Vehyr_m^i refers to the vehicle operation year in the year of i, $\mathrm{Vehyr}_{\mathrm{raw}}^i$ refers to the vehicle model year i in the raw file (for example, "11" represents the year 2011 and "98" represents the year "1998"), and $\mathrm{val}\left(\mathrm{year}^i\right)$ refers to the value of the year. For example,

if a car was produced in the year 2011 and a crash occurred in the year 2013, Vehyr_m^{2013} should be $(\text{year}^i) - \text{Vehyr}_{\text{raw}}^i$, indicating $13 - 11 = 2$.

(6) The output file contains the vectors derived from the "SEVERITY" variable in the raw dataset. In detail, noninjury was derived from "1, No Injury," injury was derived from "6, Nondisabling Injury; 7, Possible Injury," and incapacitating injury was derived from the remaining. The vectors are described through the following formula:

$$\text{Out}_i = \left(B_{\text{Injury}}, B_{\text{Non-Injury}}, B_{\text{incapacitating-injury}}\right), \quad (2)$$

where Out_i refers to the output of the crash item i, and $B_{\text{Injury}}, B_{\text{Non-Injury}}$ $B_{\text{incapacitating-injury}}$ refer to the Boolean index of the crash severity for an injury, noninjury, and incapacitating injury.

After the processing was conducted through the abovementioned steps, a total of 4310, 4494, 4436, 4666, and 4984 samples from 2011 to 2015 were used for model fitting and validation; glancing at the data, crashes seemed to be slightly more prone to occur during the winter (cold season) and on work days, whereas younger drivers contribute to a significant number of accidents (Figure 2).

After the selection of 20 features (Table 1) available from the raw file, the next step for the data cleaning and processing work was variable standardization which was carried out using the min-max method through the following formula:

$$x_n = \frac{(x - \text{MinValue})}{(\text{MaxValue} - \text{MinValue})}. \quad (3)$$

From Table 1, we can see that, among all 20 features, there are 15 categorical and five continuous variables. In addition, for a research perspective, we divided all variables into two large categories: road-related and nonroad-related.

3. Methodology

3.1. NN with the BP Method. An artificial neural network uses information technology to mimic the human neurons and can process complicated connections between the input, hidden, and output layers. Among the multiple-layer neural networks, a three-layer simple NN has been proven to be most adopted and effective in a previous research [19].

In the forward propagation three-layer NN, the input variables can be defined as an input vector X:

$$X = \left(x_1, x_2, x_3, \ldots x_i, \ldots, x_I\right)^T, \quad (4)$$

where x_i refers to the ith input variable, $I = 20$, and T refers to a transpose in the matrix calculation.

Similarly, the expectation of the crash severity level output vectors Ψ can be

$$\Psi = \left(\psi_{\text{Injury}}, \psi_{\text{Non-Injury}}, \psi_{\text{Incapacitating-Injury}}\right)^T, \quad (5)$$

where ψ_{Injury} refers to the Boolean index for an injury-related crash, $\psi_{\text{Noninjury}}$ refers to a Boolean index for a noninjury-related crash, and $\psi_{\text{Incapacitating-Injury}}$ refers to a Boolean index for an incapacitating injury.

In addition, the weight matrix between the input and hidden layers, W^1, should be

$$W_{j,i}^{(1)}\,(j = 2, \ldots, J; i = 1, \ldots, I), \quad (6)$$

where $W_{j,i}^{(1)}$ denotes the weight between the ith input node and jth hidden node and J refers to the total number of the hidden layers, $I = 20$.

The weight matrix between the hidden and output layers, W^2, should be

$$W_{k,j}^{(2)}\,(k = \{\text{Injury, Non} - \text{injury, Incapacitating} - \text{Injury}\}; \; j = 1, \ldots, J), \quad (7)$$

where $W_{k,j}^{(2)}$ denotes the weight between the kth output node and the jth hidden node, J refers to the total number of hidden layers, and the k vector has values indicating injuries, noninjuries, and incapacitating injuries.

The structure of a general three-layer neural network is shown in Figure 3.

Using the forward calculation method [17], the outputs of the nodes in the hidden layer $H_j(m)$ can be depicted as

$$H_j(m) = g_j\left(\sum_{i=1}^{I} W_{j,i}^{(1)} \cdot x_i(m) + \beta_j\right), \quad (8)$$

where m is the number of the output neurons, $W_{j,i}^{(1)}$ denotes the weight between the ith input node and the jth hidden node, J refers to the total number of nodes in hidden layer, g_j is the activation function between the input and hidden layers, and β_j is the bias term.

Similarly, the outputs calculated from the hidden layer $\psi_k(m)$ are shown as

$$\psi_k(m) = g_k\left(\sum_{j=1}^{J} W_{k,j}^{(2)} \cdot H_j(m) + \theta_k\right), \quad (9)$$

where $W_{k,j}^{(2)}$ denotes the weight between the kth output node and the jth hidden node, J refers to the total number of nodes in the hidden layer, the k vector has values including injuries, noninjuries, and incapacitating injuries, and g_k is the activation function between the output layer and hidden layers. θ_k is the bias term.

Generally, the activation functions such as sigmoid or tan$h$ are selected and have the ability to transform the input signal into a certain range. If the network adopts sigmoid function as the output activation function, the output can be narrowed into a small scale as $(0, 1)$; however, between the hidden and input layers, usually, the tan$h$ function is adopted because it usually can converge fast.

Another aspect used for building an NN is the definition of the number of nodes in the hidden layer, and there is no easy and complete mathematical way of defining this number; however, based on the experience from former research [20], the empirical equation used is as follows:

$$n = \sqrt{n_i + n_0} + \alpha, \quad (10)$$

where n is the number of hidden nodes, n_i is the number of input nodes, n_0 is the number of outputs, and α is a constant varying from 1 to 10.

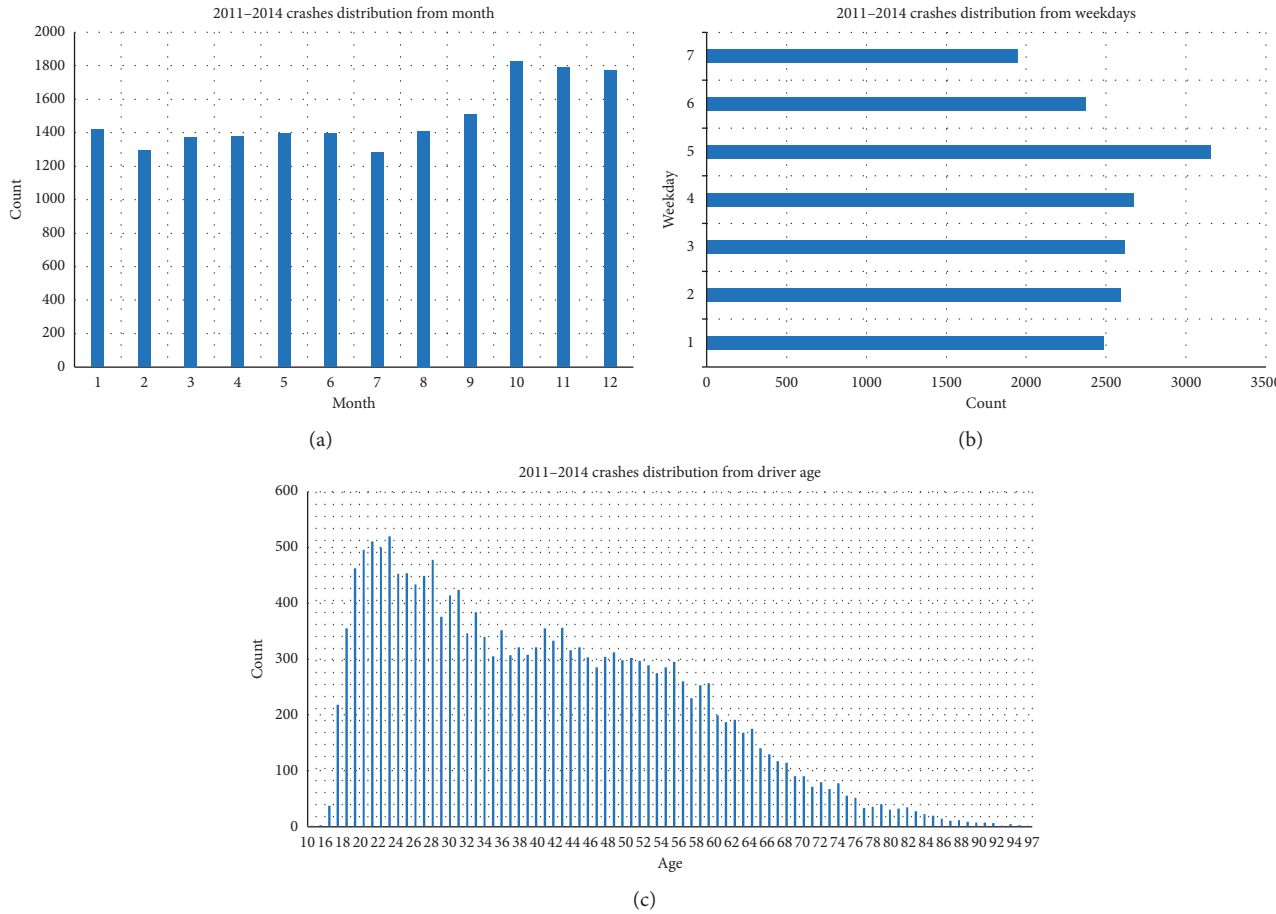


Figure 2: Crash distribution based on a (a) month, (b) weekday, and (c) driver age for 2011–2014.

Table 1: Summary of the selected variables.

Category	Variables	Value definition
Nonroad	(1) Accident type, (2) month, and (3) weekday	Categorical, categorical, and categorical
	(4) Location type, (5) light, and (6) Driver sex	Categorical, categorical, and categorical
	(7) Driver age, (8) driver restrain, and (9) vehicle year	Continuous, categorical, and categorical
	(10) Vehicle type and (11) weather	Categorical and categorical
Road	(12) Road characteristics and (13) road surface	Categorical and categorical
	(14) Road functional class and (15) curve angle	Categorical and continuous
	(16) Curve direction and (17) gradient direction	Categorical and categorical
	(18) Gradient percentage and (19) curve radius	Continuous and continuous
	(20) Curve degree	Continuous

Based on the BP method [17], the local gradients $\delta_k^{(2)}(m)$ and $\delta_j^{(1)}(m)$ of the output and hidden layer neurons and the correction values $\Delta w_{k,j}^{(2)}(m)$ and $\Delta w_{j,i}^{(1)}(m)$ of their connection weights are as follows:

$$\begin{aligned}
\delta_k^{(2)}(m) &= e_k(m) \cdot g_k'\left(v_k^2(m)\right),\\
\Delta w_{k,j}^{(2)}(m) &= a(m) \cdot \Delta w_{k,j}^{(2)}(m-1) + \eta(m) \cdot \delta_k^{(2)}(m) \cdot \psi_k(m),\\
\delta_j^{(1)}(m) &= g_j'\left(v_j^{(1)}(m)\right) \sum_K \delta_k^{(2)}(m) \Delta w_{k,j}^{(2)}(m),
\end{aligned} \tag{11}$$

where $a(m)$ and $\eta(m)$ are the momentum and step size, respectively.

The weights in the network can be updated as

$$W_{j,i}^{(1)} = W_{j,i}^{(1)} + \Delta w_{j,i}^{(1)}(m), \tag{12}$$

$$W_{k,j}^{(2)} = W_{k,j}^{(2)} + \Delta w_{k,j}^{(2)}(m). \tag{13}$$

Other than the general BP method, there are some other modified training functions, including resilient backpropagation (RPROP), conjugate gradient backpropagation,

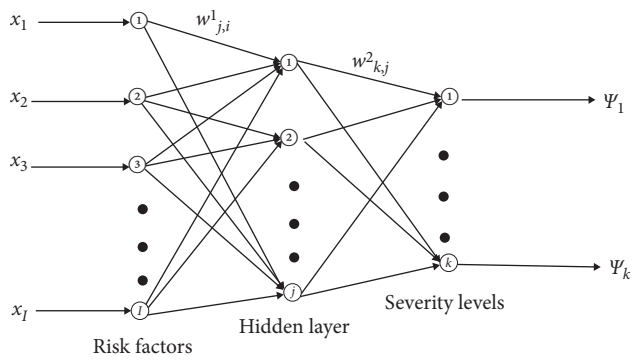


FIGURE 3: General structure of a typical NN used in this study.

gradient descent/momentum, and adaptive backpropagation. These functions have different levels of accuracy and training speeds, and thus, an attempt should be made to find a better solution.

In the traditional BP, method, the initialized weights in formulas (6) and (7) are randomly initialized following two different uniform distributions. However, this implementation will highly possibly cause the whole model solution stopping at a local optimum. In order to address this problem, the authors offered a particle swarm optimization method with refined adaptive inertial weights to enhance both local and global searching for the optimal initial weights for BPNN. The details are provided in the following section.

3.2. PSO with Adaptive Inertial Weights. The particle swarm optimization method was a well-known metaheuristic computation method provided in 1995 [21]. Also, it was easy to use in different optimization applications [22]. The standard and original formulas for this method are

$$V_s^{r+1} = V_s^r + c_1\lambda_1\left(p\text{best}_s^r - U_s^r\right) + c_2\lambda_2\left(g\text{best}_s^r - U_s^r\right), \tag{14}$$

$$U_s^{r+1} = U_s^r + V_s^{r+1}, \tag{15}$$

where U_s^r is the sth particle at the rth generation and V_s^{r+1} denotes this particle's velocity to the $r+1$th generation. c_1 and c_2 are two constants usually taken around the value of 2. λ_1 and λ_2 are the two uniform random numbers in the range of $[0, 1]$, $p$best is the best position experienced by the particle itself, and $g$best is the best position experienced by the particle swarm.

In this paper, the initial weights from BPNN are treated as the particles in the PSO algorithm, and the optimization problem can be described as mapping a decision space X to Y, and for a typical 3-layer neural network, they are encoded in the following set:

$$\begin{cases} \min\limits_{x\in X} Y = f(x), \\ X = \left[W_1, B_1, W_2, B_2\right], \end{cases} \tag{16}$$

where W_1 and B_1 refer to the weights and bias connecting the input and hidden layer, while W_2 and B_2 refer to the weights and bias connecting the hidden and output layer. Also, the transfer objective function Y is the neural network mean squared error (MSE).

For the standard or original PSO, it could solve nonlinear or nondifferentiable problems easily, but the searching space for a particular particle is almost fixed during each phase of generation, which means the model could then be easy and fast to find a solution, a possible solution near the local optimal. Thus, bringing the tradeoff between the local search ability and global search ability ahead [23], an inertial weight is introduced into formula (14), which could be written as follows:

$$V_s^{r+1} = w_s V_s^r + c_1\lambda_1\left(p\text{best}_s^r - U_s^r\right) + c_2\lambda_2\left(g\text{best}_s^r - U_s^r\right), \tag{17}$$

where w_s is the inertial weights controlling the global and local optimal searching speed, and it iterated during each generation in a linear or nonlinear form.

Thus, in this paper, different inertial weight setting methods [23–25] including the linear and nonlinear form are compared as follows:

$$w1(s) = \frac{w_{\text{end}} + \left(\left(w_{\text{start}} - w_{\text{end}}\right)\left(S_{\max} - s\right)\right)}{S_{\max}}, \tag{18}$$

$$w2(s) = w_{\text{start}} - \left(w_{\text{start}} - w_{\text{end}}\right)\left(\frac{s}{S_{\max}}\right)^2, \tag{19}$$

$$w3(s) = w_{\text{start}} - \left(w_{\text{start}} - w_{\text{end}}\right)\left[\frac{2s}{S_{\max}} - \left(\frac{s}{S_{\max}}\right)^2\right], \tag{20}$$

where w_{start} and w_{end} refer to the weights at the start and the end of the generation, and they are usually set to 0.9 and 0.4, respectively, s is the current iteration, and $S_{\max}$ is the max generation.

The function graph for formulas (18)–(20) is depicted in Figure 4. From Figure 4, can we see that the weight is of a relatively high value at first in order to expand the search space for global optimal; however, at the end of the iteration, it is converging slowly to enhance the local optimal search.

In conclusion, the pseudocode for the whole procedure in Sections 3.1 and 3.2 is formulated as follows:

```
FOR I = 1: MAX GENERATION
  FOR J = 1: POPULATION SIZE
    PARTICLE INITIALIZE;
    CALCULATE FITNESS (MSE of BP NN);
    UPDATE PARTICLE VELOCITY (WITH
ADAPTIVE INERTIAL WEIGHTS);
    UPDATE PARTICLE POSITION;
  END
END
ASSIGN SOLUTION TO BP NN;
BP NN TRAINING, TESTING, VALIDATION;
NN PREDICTION;
```

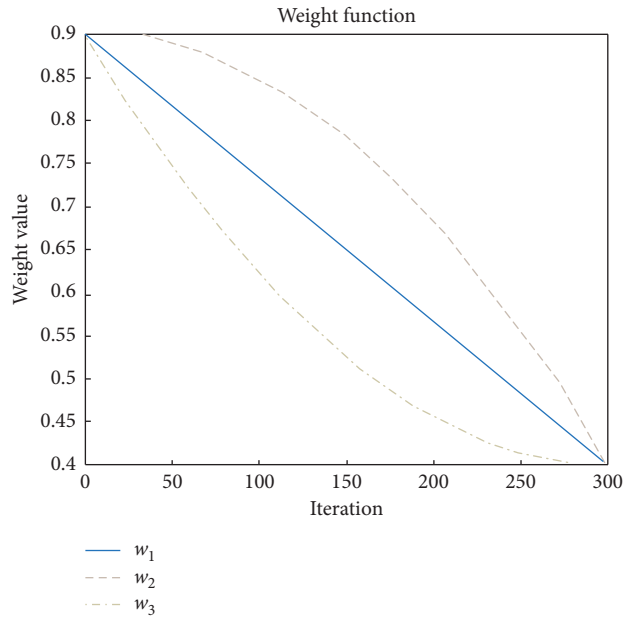


Figure 4: Weight function (18)–(20).

The calculation process is given in Figure 5.

3.3. Factor Analysis. To carry out a factor analysis (FA), the factor importance index (FII) is introduced in this paper. According to the nonlinear and classification function in practice, the n-dimensional input vector constructs the whole input space. Thus, from the perspective of engineering math, the first order partial derivative of the outcome Y with respect to the ith variable x_i could explain the connecting weight of that input variable vector, according to the chain rule in calculus, and the math description is as follows:

$$\frac{\partial Y}{\partial X_i} = \frac{\partial Y}{\partial \alpha}\frac{\partial \alpha}{\partial X_i}, \tag{21}$$

where α is the linear output value from the hidden layer to the output layer in BPNN, which could be described as formula (9), and thus, equation (21) is transformed to

$$\frac{\partial Y}{\partial X_i} = \sum_{j=1}^{L} W_{k,j}^{(2)} \frac{\partial H_j}{\partial X_i} g(\alpha)', \tag{22}$$

where j is the jth nodes in the hidden layer, $W_{k,j}^{(2)}$ refers to the weights through the hidden layer to the output layer, $g(\alpha)'$ denotes the derivative with respect to the activation function between the output and hidden layers.

Considering formula (8) combined with chain rule in calculus, formula (22) could be written as

$$\frac{\partial Y}{\partial X_i} = \sum_{j=1}^{L} W_{k,j}^{(2)} \frac{\partial H_j}{\partial \beta_j}\frac{\partial \beta_j}{\partial X_i} g(\alpha)', \tag{23}$$

$$\frac{\partial Y}{\partial X_i} = \sum_{j=1}^{L} W_{j,i}^{(1)} W_{k,j}^{(2)} f(\beta)' g(\alpha)', \tag{24}$$

where β is a short note for formula (8) and $W_{j,i}^{(1)}$ refers to the weights through the input layer to the hidden layer. $f(\beta)'$ denotes the derivative with respect to the activation function between the input layer and the hidden layer.

Through formula (24), we can see that while given a fixed input vector in the ith dimension, the value of $f(\beta)'$ and $g(\alpha)'$ is fixed with respect to all X_i; thus, considering the remaining part, the FII for the ith variable can be written as

$$R_i = \frac{\sum_{j=1}^{L} W_{j,i}^{(1)} W_{k,j}^{(2)}}{\sum_{i}^{N}\sum_{j=1}^{L} W_{j,i}^{(1)} W_{k,j}^{(2)}}, \tag{25}$$

where the value of $W_{j,i}^{(1)}$ and $W_{k,j}^{(2)}$ can be calculated through formulas (12) and (13). These values are stored in a dictionary in a program code during the calculation process.

Finally, in order to ease the simulation variance of the model training process, the FII expectation is introduced by running the model for a certain k times:

$$E(R_i) = \frac{1}{K}\sum_{k=1}^{K} R_i^k. \tag{26}$$

4. Results and Discussion

4.1. NN Model Structure Test. Based on the theory discussed in the previous section, the most primary step in building an NN is to define the number of good hidden layer nodes and a better training function. Usually, the model performance (mean square error, MSE) combined with the total iteration number of convergence is used to test the structure. For the number of hidden layer nodes, based on formula (10), consecutive numbers of 5 through 14 were selected; for the training function, however, one of the following (Table 2) is applied.

The last two methods in Table 2 usually provide a fast calculation speed, but tend to be challenging and inefficient when dealing with the big data issues, especially, for the GPU hardware with low configuration. For the present study, considering the sample size, GPU support is not a problem, and thus, this minor difference is not a significant concern.

Theoretically, the number of nodes in the hidden layers should be within the range of $(5, 14)$ based on formula (10); usually, however, the number of hidden layer nodes does not fall below 10, and thus, a combined test using the popular training functions and 10–14 hidden layer nodes number was carried out based on a loop test.

We randomly separated the sample data from 2011 to 2014 to form the dataset of training, testing, and validation with respect to 70%, 15%, and 15%. In detail, a total data of 17839 were divided into 12487, 2676, and 2676, in accordance with training, testing, and validation. The outcome is shown in Table 3.

It can be seen from Table 3 that 12 and 14 hidden layer nodes achieve the best performance (in other words, the lowest MSE) and the BR training function has the lowest MSE. However, when adopting these methods, the gradient value of the GDA, GDX (with an adaptive learning rate), and LM decreases quickly during the very early validation stage

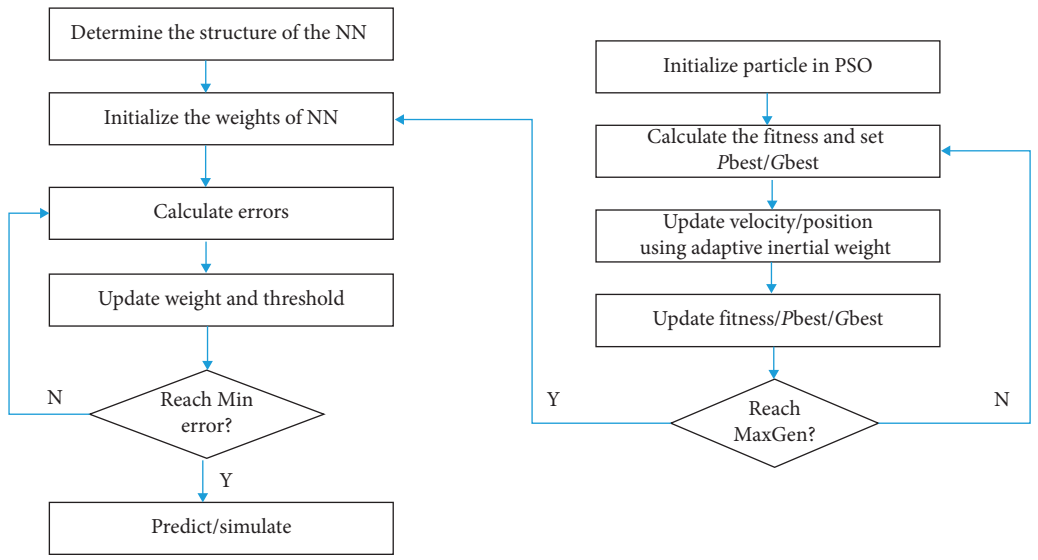


Figure 5: Calculation process for the method used in this study.

Table 2: Summary of training functions.

No.	Function name	Abbreviation
1	BFGS quasi-Newton backpropagation	BFGS
2	Conjugate gradient backpropagation with Powell-Beale restarts	CGB
3	Conjugate gradient backpropagation with Fletcher–Reeves updates	CGF
4	Conjugate gradient backpropagation with Polak–Ribiere updates	CGP
5	Gradient descent backpropagation	GD
6	Gradient descent with adaptive learning rate backpropagation	GDA
7	Gradient descent with momentum	GDM
8	Gradient descent w/momentum and adaptive learning rate backpropagation	GDX
9	One-step secant backpropagation	OSS
10	RPROP backpropagation	RP
11	Scaled conjugate gradient backpropagation	SCG
12	Levenberg–Marquardt backpropagation	LM
13	Bayesian regulation backpropagation	BR

Table 3: Test on the number of neural network hidden layer nodes (best validation performance in terms of MSE).

	10	11	12	13	14	Average
BFGS	0.260	0.252	0.250	0.250	0.208	0.242
CGB	0.202	0.206	0.254	0.244	0.246	0.230
CGF	0.208	0.240	0.204	0.242	0.232	0.224
CGP	0.220	0.222	0.202	0.240	0.202	0.218
GD	0.230	0.232	0.236	0.224	0.232	0.230
GDA	0.204	0.206	0.208	0.204	0.206	0.204
GDM	0.466	0.472	0.228	0.216	0.240	0.324
GDX	0.206	0.202	0.204	0.198	0.204	0.202
OSS	0.200	0.202	0.204	0.244	0.206	0.212
RP	0.206	0.202	0.198	0.206	0.210	0.204
SCG	0.204	0.206	0.198	0.206	0.204	0.204
LM	0.204	0.204	0.198	0.208	0.202	0.202
BR	0.199	0.199	0.199	0.198	0.198	0.198
Average	0.230	0.234	0.214	0.220	0.214	

and, thus, quickly converges to the preset goal, and methods such as BR, GDM, and GD converge slowly, requiring more validation than usual. Thus, in conclusion, the choice of GDA, GDM, or LM should be better, and considering a lack of GPU support for the simulation environment, we selected the Levenberg–Marquardt backpropagation (LM) as the training function. During the number test, LM with 12 nodes of the hidden layer showed the best performance.

4.2. Results for PSO Optimization. After setting the adaptive inertial weights for the PSO optimizer, we can conclude the following performance graph through each iteration.

To eliminate the random data separation variance, 100-time simulation was conducted, and the average performance for each method is described in Table 4.

From Figure 6 and Table 4, it could be seen that the performance refers to the MSE of the training model, training accuracy refers to the average classification accuracy among the training set (17839), and the prediction accuracy refers to the classification accuracy among the 4984 samples from 2015. Although the fitness (MSE) of PSO with w_1 drops fast in the early stage (almost starts with 22), it converges poorly and has the poor condition of performance and training/testing accuracy, which means it lacks of certain

Table 4: Summary of the performance relating to different models.

Category	Performance	Training accuracy (percentage)	Prediction accuracy (percentage)
NN	0.235	78.5	71.6
NN with std. PSO	0.232	78.6	72.4
NN with PSO (w_1)	0.298	73.2	70.5
NN with PSO (w_2)	0.194	80.4	73.1
NN with PSO (w_3)	0.196	79.3	73.6

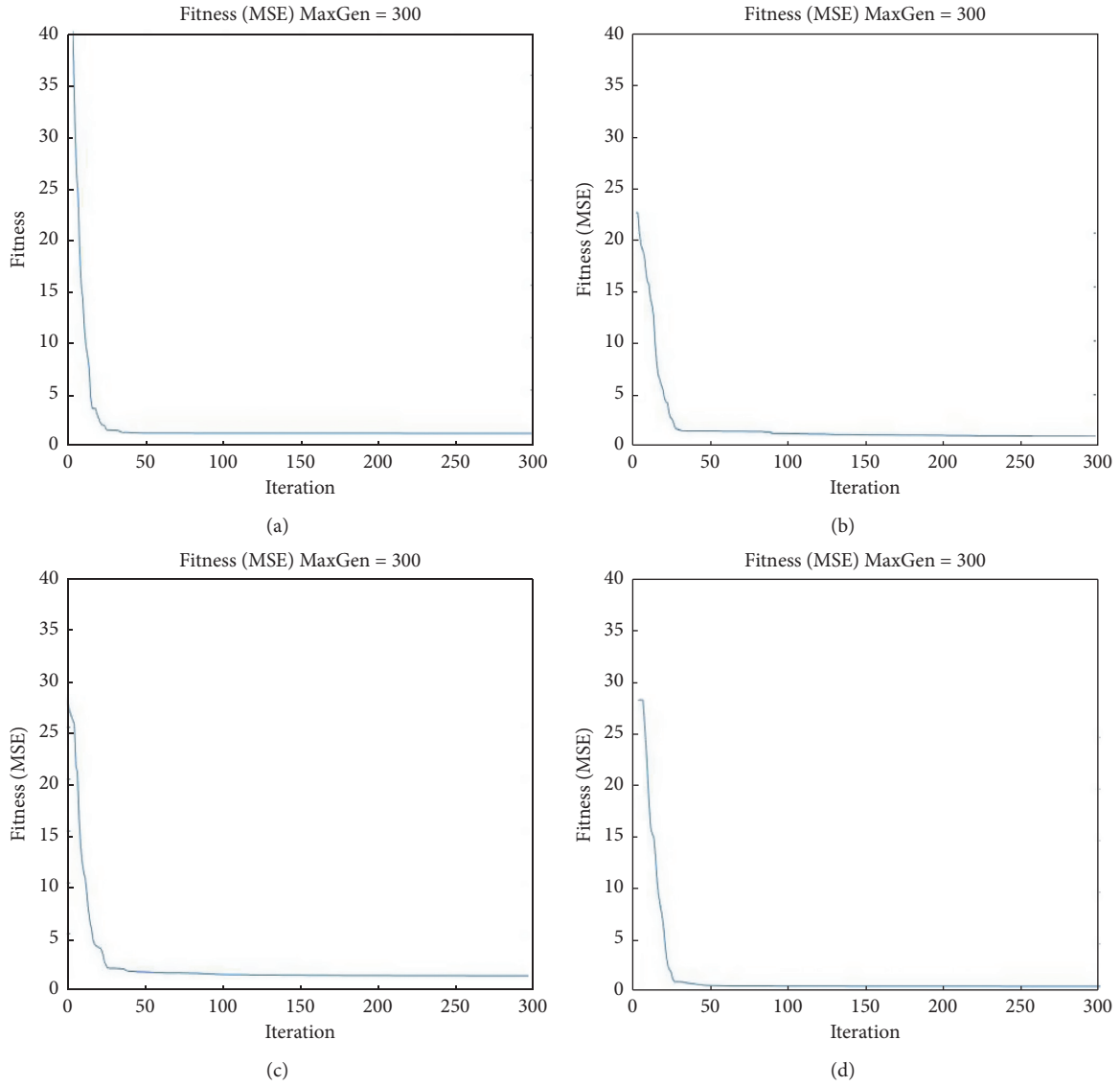


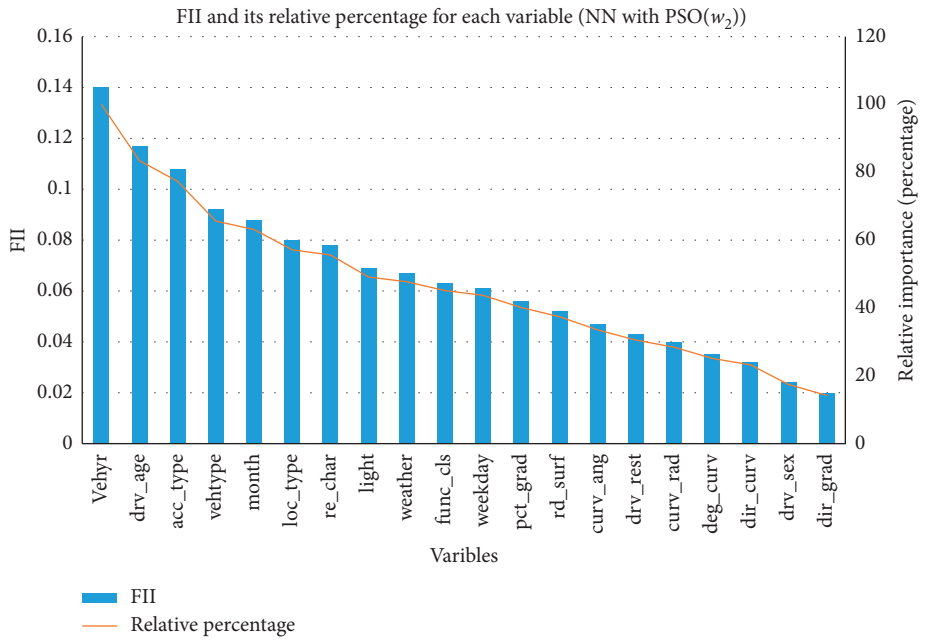
Figure 6: NN performance (MSE) adopting different PSO optimizations. (a) PSO without inertial weight; (b) PSO with w_1; (c) PSO with w_2; and (d) PSO with w_3.

ability for local optimal searching. On the other hand, PSO with w_2 and w_3 (two nonlinear formulas) outperforms the other methods not only in global optimal search (both from 28, while NN with standard PSO drops from nearly 40) but also in the local optimal search (with good performance and accuracy), and the NN and NN with standard PSO are in the middle level with an acceptable result. It can be assumed that with more data flushing in, the PSO with nonlinear will perform a dominating advantage over the other methods.

As shown in Table 4, the accuracy of training is better than the results of the prediction; the relatively lower sample number may be a major contributor. Notably, the performance from all categories is at the same rate related to the sample size, and it can be concluded that although the

Table 5: Summary of FII expectation ($K = 100$) and relative percentage for two different methods.

Variables	Category	NN with PSO (w_2)		NN with PSO (w_3)	
		FII	Relative percentage (%)	FII	Relative percentage (%)
Acctype	Nonroad	0.108	77.3	0.080	63.4
Month	Nonroad	0.088	63.1	0.102	81.0
Weekday	Nonroad	0.061	43.7	0.054	43.5
loc_type	Nonroad	0.080	57.1	0.076	60.1
rd_char1	Road	0.078	55.6	0.063	50.2
Rdsurf	Road	0.052	37.4	0.456	36.2
Light	Nonroad	0.069	49.1	0.071	56.1
Weather	Nonroad	0.067	47.7	0.059	47.2
func_cls	Road	0.063	45.1	0.058	46.1
drv_sex	Nonroad	0.024	17.4	0.017	13.1
drv_rest	Nonroad	0.043	30.5	0.039	31.0
Vehtype	Nonroad	0.092	65.6	0.090	72.0
dir_curv	Road	0.032	23.2	0.028	22.1
dir_grad	Road	0.020	14.2	0.019	15.1
drv_age	Nonroad	0.117	83.4	0.113	90.0
Vehyr	Nonroad	0.140	100.0	0.126	100
curv_ang	Road	0.047	33.5	0.045	35.4
pct_grad	Road	0.056	40.1	0.052	41.3
deg_curv	Road	0.035	25.1	0.032	25.4
curv_rad	Road	0.040	28.4	0.034	26.8



(a)

Figure 7: Continued.

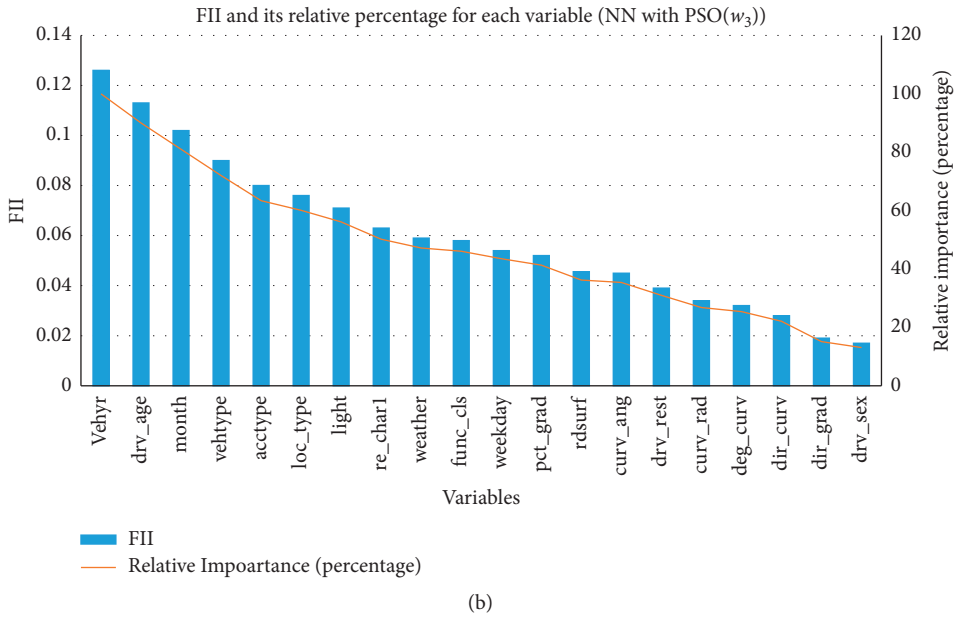


Figure 7: Summary of results for factor analysis. (a) FII and its relative percentage for each variable (NN with PSO (w_2)); (b) FII and its relative percentage for each variable (NN with PSO (w_3)).

prediction accuracy applied to other samples could result in a marginally worse accuracy, the model can explain all variance and contributors.

4.3. Results for Factor Analysis. The final results for two PSO with nonlinear adaptive inertial weight are described in Table 5.

From Figure 7 and Table 5 we can see that two methods (PSO with w_2 and PSO with w_3) provide almost the same ranking with respect to each variable, despite some minor difference, for example, MONTH and LIGHT are ranked slightly higher in the second method, while DRIVER SEX are ranked slightly lower. Nearly seven or eight of the 20 variables have a relative importance value exceeding 50%, including the vehicle year, driver age, accident type, month, location type, and road functional class (also LIGHT in the second method), with vehicle year and driver age taking the top two values. Additionally, the related road factors such as curve and gradient variables have the least importance, and the only important factor contributing to the severity prediction is the road functional class, which indicates whether a crash occurred on a certain type of road. Thus, it can be concluded that the relative road variables contribute less than the relative nonroad variables, particularly, compared with the relative vehicle factors (vehicle year and type). Therefore, the policy makers should pay more attention to the vehicle and driver regulation rules, as well as the road design, to reduce the possible severity level in the future.

Driver age and month are two other important factors in predicting a crash severity. From the sample size, we can see that the most severe crashes occur during the winter in the Washington State (December, January, and February), and drivers below the age of 25 and above the age of 60 are more prone to encountering severe injury crashes. The month may account for the rainy season in the mountainous Seattle area, whereas age may be derived from the fact that younger people and older people are more prone to making severe mistakes.

5. Conclusions

In this study, a thorough artificial neural network (ANN) was developed to address the problems of the crash severity level modeling and factor analysis (FA). Besides the test of different types of training structure and methods, more importantly, a nonlinear adaptive PSO optimization method was proposed in order to solve the tradeoff problem between the global and local search ability among the previous studies. The detail test of different algorithm confirmed our hypothesis. The additional contributing factor analysis also offers a different point of view compared with former statistical analysis. The main conclusions can be concluded as follows:

(1) The number 12 hidden layer nodes fit the model developed in this paper well; and the BP method (Levenberg–Marquardt) can be better utilized when aided by fast hardware

(2) The simulation result showed that the PSO optimizer with nonlinear adaptive inertial weight outperforms the standard PSO and PSO with linear adaptive inertial weight

(3) Through the factor analysis (FA), it can be found that, among all 20 variables, nonroad-related variables can account for most of the severity prediction variance, and the rainy mountainous area in Seattle may be the reason for the importance of the month as a factor and, also, the impact of driver age, where

younger and older people are more prone to encountering a severe crash

The main innovations can be concluded as follows:

(1) Traditional studies often used statistical methods like the Poisson regression, negative binary regression, and generalized logit or probit model for the identification and mathematical qualification of the inner internal triggers and their impact on crash severity, while this paper utilized FA as the analytical tool, which is unusual for the current research system of crash severity, and we think that our attempt extended the methods of crash severity analyses, and more research could be conducted in the future work.

(2) FA, as a traditional statistical implement, also can serve as a powerful explanatory tool in the last stage of the model, and our work has proved it. The application of FA in this paper indicated that the basic statistical method is still useful and efficient while the AI methods sometimes did not have an agreeable explanation for the inner mechanism of the data.

The method developed in this study can be applied to a big data analysis of traffic accidents and be used as a fast-useful tool for policy makers and traffic safety researchers. The authors recognize that much can be further investigated. In this paper, only crash severity was discussed. Further research could be conducted from the perspective of the collision type (e.g., head-on collisions and rear-end collisions). Besides, the dataset could be enlarged in the future research to improve the accuracy.

Data Availability

The dataset used in this study was made up of data requested from the Highway Safety Information System (HSIS), and requests for access to these data should be made by filling the form at the following link: https://www.hsisinfo.org/datarequest.cfm.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors would like to thank the Highway Safety Information System, U.S.A, Smart Transportation Application and Research Laboratory (STAR Lab) at the University of Washington, U.S.A, and Pacific Northwest Transportation Consortium Region 10, U.S.A. They also thank National Natural Science Foundation of China (Grant nos. 51778141 and 71871078), China Scholarship Council, and Jiangsu Creative PhD Student Sponsored Project (KYLX15_0157) for providing essential data and support.

References

[1] Washington State Department of Transportation, *Fatality Analysis Reporting System (FARS) Auxiliary Datasets Analytical User's Manual 2007–2016, Annual Collision Summary*, Washington State Department of Transportation, Olympia, WA, USA, 2015.

[2] D. Lord and F. Mannering, "The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives," *Transportation Research Part A: Policy and Practice*, vol. 44, no. 5, pp. 291–305, 2010.

[3] X. Ma, S. Chen, and F. Chen, "Multivariate space-time modeling of crash frequencies by injury severity levels," *Analytic Methods in Accident Research*, vol. 15, pp. 29–40, 2017.

[4] R. O. Mujalli, G. López, and L. Garach, "Bayes classifiers for imbalanced traffic accidents datasets," *Accident Analysis & Prevention*, vol. 88, pp. 37–51, 2016.

[5] Y. Li, D. Ma, M. Zhu, Z. Zeng, and Y. Wang, "Identification of significant factors in fatal-injury highway crashes using genetic algorithm and neural network," *Accident Analysis and Prevention*, vol. 111, pp. 354–363, 2018.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] AASHTO, *Highway Safety Manual*, American Association of State Highway Transportation Officials, Washington, DC, USA, 2010.

[8] R. Elvik, P. Christensen, and A. Amundsen, "Speed and road accidents: an evaluation of the power model," *TOI Report*, vol. 740, 2004.

[9] F. Russo, M. Busiello, and G. Dell Acqua, "Safety performance functions for crash severity on undivided rural roads," *Accident Analysis & Prevention*, vol. 93, pp. 75–91, 2016.

[10] J. Park and M. Abdel-Aty, "Safety performance of combinations of traffic and roadway cross-sectional design elements at straight and curved segments," *Journal of Transportation Engineering, Part A: Systems*, vol. 143, no. 6, Article ID 4017015, 2017.

[11] M. Ahmed, H. Huang, M. Abdel-Aty, and B. Guevara, "Exploring a Bayesian hierarchical approach for developing safety performance functions for a mountainous freeway," *Accident Analysis & Prevention*, vol. 43, no. 4, pp. 1581–1589, 2011.

[12] B. Debrabant, U. Halekoh, W. H. Bonat, D. L. Hansen, J. Hjelmborg, and J. Lauritsen, "Identifying traffic accident black spots with Poisson-Tweedie models," *Accident Analysis & Prevention*, vol. 111, pp. 147–154, 2018.

[13] X. Pei, S. C. Wong, and N. N. Sze, "A joint-probability approach to crash prediction models," *Accident Analysis & Prevention*, vol. 43, no. 3, pp. 1160–1166, 2011.

[14] K. El-Basyouny, S. Barua, and M. T. Islam, "Investigation of time and weather effects on crash types using full bayesian multivariate poisson lognormal models," *Accident Analysis and Prevention*, vol. 73, pp. 91–99, 2014.

[15] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[16] L.-Y. Chang, "Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network," *Safety Science*, vol. 43, no. 8, pp. 541–557, 2005.

[17] Q. Zeng and H. Huang, "A stable and optimized neural network model for crash injury severity prediction," *Accident Analysis & Prevention*, vol. 73, pp. 351–358, 2014.

[18] H. Huang, Q. Zeng, X. Pei, S. C. Wong, and P. Xu, "Predicting crash frequency using an optimised radial basis function neural network model," *Transportmetrica A: Transport Science*, vol. 12, no. 4, pp. 330–345, 2016.
[19] S. S. Haykin, *Neural Networks and Learning Machines*, Prentice Hall, Piscataway, NJ, USA, 3rd edition, 2009.
[20] J. S. Judd, *Neural Network Design & the Complexity of Learning*, The MIT Press, Cambridge, MA, USA, 2000.
[21] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks 1942–1948*, Piscataway, NJ, USA, 1995.
[22] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
[23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Computational Intelligence*, pp. 69–73, Ankorage, AK, USA, June 1998.
[24] L. Zhang, H. Yu, and S. Hu, "A new approach to improve particle swarm optimization," *Genetic and Evolutionary Computation—GECCO 2003*, vol. 2327, pp. 134–139, 2003.
[25] S. Liang and W. Dong, "PSO algorithm with dynamic inertial weight vector and dimension mutation," *Computer Engineering & Application*, vol. 47, no. 5, pp. 29–32, 2011.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 8215389, 9 pages
https://doi.org/10.1155/2020/8215389

Research Article

RGBD Scene Flow Estimation with Global Nonrigid and Local Rigid Assumption

Xiuxiu Li,[1,2] **Yanjuan Liu**,[1,2] **Haiyan Jin**,[1,2] **Lei Cai**,[1,2] **and Jiangbin Zheng**[3]

[1]*Xi'an University of Technology, Xi'an 710048, China*
[2]*Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China*
[3]*Northwestern Polytechnical University, Xi'an 710029, China*

Correspondence should be addressed to Xiuxiu Li; lixiuxiu@xaut.edu.cn

Received 27 March 2020; Accepted 30 May 2020; Published 29 June 2020

Guest Editor: Longzhuang Li

RGBD scene flow has attracted increasing attention in the computer vision with the popularity of depth sensor. To estimate the 3D motion of object accurately, a RGBD scene flow estimation method with global nonrigid and local rigid motion assumption is proposed in this paper. Firstly, the preprocessing is implemented, which includes the colour-depth registration and depth image inpainting, to processing holes and noises in the depth image; secondly, the depth image is segmented to obtain different motion regions with different depth values; thirdly, scene flow is estimated based on the global nonrigid and local rigid assumption and spatial-temporal correlation of RGBD information. In the global nonrigid and local rigid assumption, each segmented region is divided into several blocks, and each block has a rigid motion. With this assumption, the interaction of motion from different parts in the same segmented region is avoided, especially the nonrigid object, e.g., a human body. Experiments are implemented on RGBD tracking dataset and deformable 3D reconstruction dataset. The visual comparison shows that the proposed method can distinguish the motion parts from the static parts in the same region better, and the quantitative comparisons proved more accurate scene flow can be obtained.

1. Introduction

Vedula et al. [1] proposed the scene flow first, which describes a 3D motion field formed by the motion in 3D space scene. Scene flow is the fundamental input to high-level tasks such as scene understanding and analysis. With the development and applications of computer vision and artificial intelligence, the related technologies have been used in the object detection and segmentation [2, 3], depth interpolation, and 3D reconstruction in many dynamic scenes, such as autonomous driving [4, 5], high-speed video generation [6], and 3D reconstruction [7].

Some research efforts have been dedicated to the estimation of the scene flow, which involve different environments, monocular vision [8], stereo vision [1, 2, 9, 10], and RGBD [11–13]. Affordable RGBD cameras can directly capture both colour and depth information simultaneously, so we focus on the RGBD scene flow estimation. Among the existing methods, methods based on segmentation are attractive, which can deal with large displacement and occlusion better. For this method, the correlation of motion in the local area is considered, such as the assumption of local rigid area, which can improve the accuracy of the scene flow estimation. In the local rigid area, it is assumed that all pixels in a segmented region share a rigid motion.

However, if the segmented region is a nonrigid object, pixels with different motion degrees would affect each other and then affect the overall scene flow estimation effect. In this paper, the local rigid and global nonrigid assumption in segmented regions is introduced into the RGBD scene flow estimation. In this assumption, the local motion in a segmented object area is correlated, and the motion of the whole segmented object is nonrigid. With this assumption, the interaction of motion from different parts in the same segmented region is avoided.

2. Related Work

According to the difference of solving process, these approaches are divided into two categories roughly: the variational approaches [1, 8, 12, 13], which construct the objective function on scene flow directly, and the methods based on segmentation with the assumption of local rigid motion [14–16].

The variational approaches estimate the dense scene flow with constraints of the spatial-temporal vision information commonly. An objective function is constructed to estimate the dense scene flow [1, 8, 17]. Xiao et al. [8] construct an objective function on scene flow in a monocular camera environment, which includes a brightness constancy assumption, a gradient constancy assumption, a short time object velocity constancy assumption, etc. Jaimez et al. [17] considered the depth information from the RGBD camera and presented a dense real-time scene flow algorithm with brightness constancy and geometric consistency.

The methods based on segmentation estimate the rigid motion of each segmented region, and then the local rigid motion and nonrigid motion are mixed to get dense scene flow [16, 18–20]. In [20], an efficient RGBD PatchMatch was used to solve large displacement motion patterns and stage, and further occlusion model and spatial smoothness regularization were used to compute the RGBD scene flow field. Golyanik et al. [18] presented a multiframe scene flow approach that assumes scene transformations to be locally rigid in RGBD image sequences. Xiang et al. [19] used a 3D local rigidity assumption to estimate the dense scene flow in a variational framework. Schuster et al. [21] interpolated the sparse matches between stereoscopic image pairs to estimate scene flow, in which the initial sparse match is the local rigid assumption actually.

Sun et al [16] proposed a layered RGBD scene flow method, in which the depth ordering from RGBD is used to segment the scene, and solved the occlusions. The layered RGBD scene flow method is a promising method as spatial smoothness is separated from the model of discontinuities and occlusions, which can model occlusion boundaries by obtaining the relative depth order. Depth image is layered based on the depth information. In order to estimate the motion of the scene, it assumed that pixels belonging to the same layer have the same rigid motion.

The result of estimating scene flow directly is high dimensional, so the solution space is large and the calculation complexity is high. And methods with the assumption of local rigid motion reduce the solution space. However, for most of the methods, the assumption of local rigid motion, a local region is semantic, such as a superpixel or a specific object. So the assumption cannot be well applied to nonrigid objects because the internal motion of nonrigid object is not consistent. In this paper, we propose an assumption of global nonrigid and local rigid motion based on the study of Sun et al. [16], which can accurately estimate the motion of each segmented region by dividing each segmented region into different blocks. Besides, affordable RGBD cameras provide both colour and depth information simultaneously. Therefore, we would focus on approaches with colour and depth information.

3. Methodology

In this section, a framework for estimating RGBD scene flow is shown in Figure 1. In this framework, two steps are presented to get scene flow: the preprocessing and the scene flow estimation.

The preprocessing mainly performs basic processing on the input RGBD image sequence (the red box in Figure 1), thus providing materials for estimating scene flow efficiently, involving the colour-depth registration and depth image inpainting. Details will be introduced in Section 3.1. The scene flow estimation would present the calculating processing of scene flow. It includes two parts: depth image segmentation and scene flow estimation with the preprocessing result and spatiotemporal constraints from the RGBD image sequence (Section 3.2).

3.1. Preprocessing. In the preprocessing, the color-depth registration and the depth image inpainting are implemented. The color-depth registration is used to associate the depth image with the RGB image. And the depth image inpainting is used to repair holes and noises, which are from occlusions, lack of point correspondences, sensor imperfection, etc.

3.1.1. Colour-Depth Registration [22]. To register the RGB image and depth image, a projective matrix M is calculated as shown in equation (1). In equation (1), (x, y) is a pixel coordinate in the depth image, and (X, Y) is the corresponding coordinate in the RGB image:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{1}$$

Furthermore, equation (1) can be rewritten as follows:

$$X = m_1 x + m_2 y + m_3 - m_7 x X - m_8 y X, \tag{2}$$

$$Y = m_4 x + m_5 y + m_6 - m_7 x Y - m_8 y Y. \tag{3}$$

In M, eight unknown parameters need to be solved, and four pairs of corresponding points in the depth image and RGB image are needed at least. In our paper, corners are used.

3.1.2. Inpainting. To process the holes and noises in the depth image, the inpainting algorithm with the guidance of RGB image information is used [23]. In this algorithm, holes and small noises are all regarded as noises, but holes have larger connected areas and the depth value is 0, while small noises have smaller connected areas. In this paper, holes are inpainted based on depth domain similarity and colour consistency from the aligned depth image and RGB image. And small noises are removed with the local bilateral filter.

3.2. Scene Flow Estimation. In order to estimate scene flow accurately, the depth image is segmented into different regions roughly since there is stronger local motion correlation in the same region. Based on the inpainted depth image, K-means clustering algorithm is used to segment and label the depth image, by which scene can be quickly and simply segmented based on the depth information. The value of K depends on the number of moving regions in the scene.

To calculate the RGBD scene flow, an assumption of global nonrigid and local rigid motion is proposed to describe the behaviours of the scene in this paper. In a segmented region, the pixels' motion of its inner local area is highly consistent, so it is assumed that the local motion of a segmented region is rigid.

3.2.1. Global Nonrigid and Local Rigid Assumption. Each segmented region is divided into a number of sufficiently small blocks and the size of the block is 3×3 (Figure 2). In the global nonrigid and local rigid assumption, pixels in each block share the common 3D rigid motion R, which includes the rotation and the translation relative to the camera coordinate system (local rigid assumption), and different blocks have different motions (global nonrigid assumption).

Let a 2D point $p_1 = (x_1, y_1)$ at frame t, and its corresponding 2D point $p_2 = (x_2, y_2)$ in frame $t+1$. The depth values of p_1 and p_2 are z_1 and z_2, which are from the depth images. According to the camera imaging principle and the 2D-3D transformation model in [16], the corresponding 3D point $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ of p_1 and p_2 are as follows:

$$\begin{cases} X_1 = z_1 \cdot \dfrac{(x_1 - c_x)}{f_x, Y_1} = z_1 \cdot \dfrac{(y_1 - c_y)}{f_y, Z_1} = z_1, \\ \\ X_2 = z_2 \cdot \dfrac{(x_2 - c_x)}{f_x, Y_2} = z_2 \cdot \dfrac{(y_2 - c_y)}{f_y, Z_2} = z_2, \end{cases} \tag{4}$$

where $(f_x, f_y)^T$ and $(c_x, c_y)^T$ represent the camera focal length and distortion coefficient, respectively.

The rigid motion R from P_1 to P_2 can be expressed as follows:

$$P_2 = R \cdot \begin{bmatrix} P_1 \\ 1 \end{bmatrix}. \tag{5}$$

In equation 5, the image coordinate p_2 corresponding to the spatial point P_2 is given by

$$p_2 = \left(f_x \frac{X_2}{z_2} + c_x, f_y \frac{Y_2}{z_2} + c_y \right). \tag{6}$$

The corresponding local rigid RGBD scene flow from p_1 to p_2 is as follows:

$$\begin{aligned} u^R(p_1) &= f_x \frac{X_2}{z_2} + c_x - x_1, \\ v^R(p_1) &= f_y \frac{Y_2}{z_2} + c_y - y_1, \\ w^R(p_1) &= z_2 - z_1, \end{aligned} \tag{7}$$

where u, v, and w are the horizontal motion, vertical motion, and depth change of p_1.

Furthermore, a term on spatial constraints for scene flow is presented as follows:

$$\begin{aligned} E_{\text{spa}}(u_{tk}, v_{tk}, w_{tk}, R_{tk}) = {} & E_{\text{spa\_}u}(u_{tk}, R_{tk}) + E_{\text{spa\_}v}(v_{tk}, R_{tk}) \\ & + E_{\text{spa\_}w}(w_{tk}, R_{tk}), \end{aligned} \tag{8}$$

where

$$\begin{aligned} E_{\text{spa\_}u}(u_{tk}, R_{tk}) &= \sum_p \left(\sum_{p' \in N_p} \left(u_{tk}(p) - u_{tk}^R(p') \right)^2 + \sum_{p' \in N_p} \left(\left(u_{tk}(p) - u_{tk}^R(p) \right) - \left(u_{tk}(p') - u_{tk}^R(p') \right) \right)^2 \right), \\ E_{\text{spa\_}v}(v_{tk}, R_{tk}) &= \sum_p \left(\sum_{p' \in N_p} \left(v_{tk}(p) - v_{tk}^R(p') \right)^2 + \sum_{p' \in N_p} \left(\left(v_{tk}(p) - v_{tk}^R(p) \right) - \left(v_{tk}(p') - v_{tk}^R(p') \right) \right)^2 \right), \\ E_{spa\_w}(w_{tk}, R_{tk}) &= \sum_p \left(\sum_{p' \in N_p} \left(w_{tk}(p) - w_{tk}^R(p') \right)^2 + \sum_{p' \in N_p} \left(\left(w_{tk}(p) - w_{tk}^R(p) \right) - \left(w_{tk}(p') - w_{tk}^R(p') \right) \right)^2 \right), \end{aligned} \tag{9}$$

where u_{tk}, v_{tk}, and w_{tk} are the scene flow of in directions x, y, and z for the segmented region k at frame t, and N_p is 4 nearest spatial neighbours of the pixel p.

$E_{\text{spa\_}u}$, $E_{\text{spa\_}v}$, and $E_{\text{spa\_}w}$ reflect motion correlation in different directions within the same segmented region.

3.2.2. Spatiotemporal Correlation. Referring to the objective function in [16, 24, 25], the spatiotemporal correlation of the RGBD image sequence is also considered besides the global nonrigid and local rigid assumption. The spatial-temporal correlation of the RGBD image sequence contains two

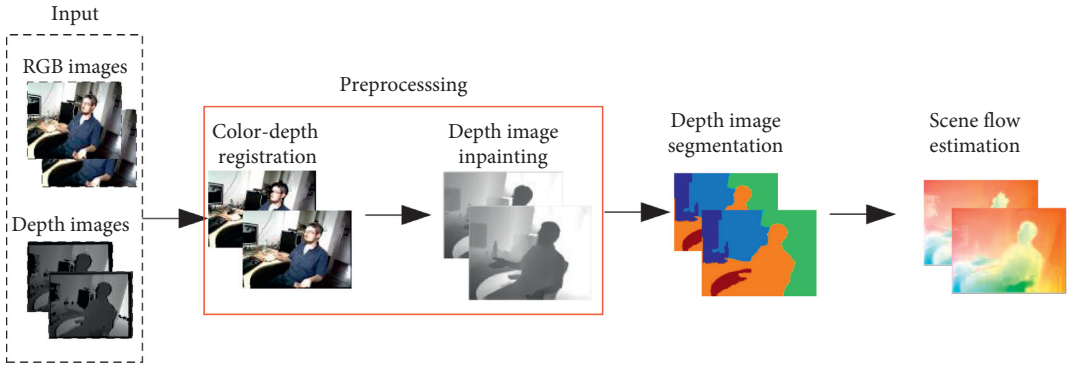


Figure 1: The framework for RGBD scene flow estimation. This framework shows the steps of RGBD scene flow estimation.

terms: the consistency of RGBD data and the coherence of the segmented regions.

(1) *The Consistency of RGBD Data.* If p is visible in frame t and $p + (u_{tk}(p), v_{tk}(p))$ is also visible in frame $t+1$ in the depth image and the aligned RGB image, the point has a constant appearance with the motion $(u_{tk}(p), v_{tk}(p), w_{tk}(p))$. The term consistency of RGBD data can be represented as follows:

$$E_{\text{data}}(u_{tk}, v_{tk}, w_{tk}) = \sum_{p} \Big((I_t(p) - I_{t+1}(p + (u_{tk}(p), v_{tk}(p))))^2 + (Z_t(p) + w_{tk}(p) - Z_{t+1}(p + (u_{tk}(p), v_{tk}(p))))^2 \Big), \tag{10}$$

(2) *The Coherence of the Segmented Region.* If p in frame t belongs to the segmented region k, $p + (u_{tk}(p), v_{tk}(p))$ in frame $t+1$ belongs to the segmented region k. The term coherence of the segmented region can be represented as follows:

$$E_{\text{sup}}(u_{tk}, v_{tk}, g_{tk}) = \sum_{p} \sum_{p' \in N_{(x,y)}} \left(g_{t,k}(p) - g_{t,k}(p') \right)^2 + \sum_{p} \left(g_{t,k}(p) - g_{t+1,k}(p + (u_{tk}(p), v_{tk}(p))) \right)^2, \tag{11}$$

where g_{tk} is a support function, which represents the probability size that a pixel belongs to the segmented region k in frame t.

According to equations (8), (10), and (11), a total objective function is constructed as follows:

$$E(u, v, w, g, R) = \sum_{t=1}^{T-1} \left(\sum_{k=1}^{K} \left(\lambda_{\text{data}} E_{\text{data}}(u_{tk}, v_{tk}, w_{tk}) + \lambda_{spa} E_{\text{spa}}(u_{tk}, v_{tk}, w_{tk}, R_{tk}) \right) \right) + \sum_{t=1}^{T} \sum_{k=1}^{K-1} \lambda_{\text{sup}} E_{\text{sup}}(u_{tk}, v_{tk}, g_{tk}), \tag{12}$$

where λ_{data}, λ_{spa}, and λ_{sup} represent the corresponding weight of E_{data}, E_{spa}, and E_{sup}, respectively.

The coordinate descent method is used to minimize the RGBD scene flow energy function in equation (12). Firstly, estimate the initial scene flow according to the interframe optical flow and segmentation of the depth image. Secondly, obtain the optimized scene flow by image warping while keeping the layering result fixed. Thirdly, calculate the optimized layered support function with coordinate descent method while keeping the scene flow fixed. Finally, get the final scene flow by looping the second and third operations.

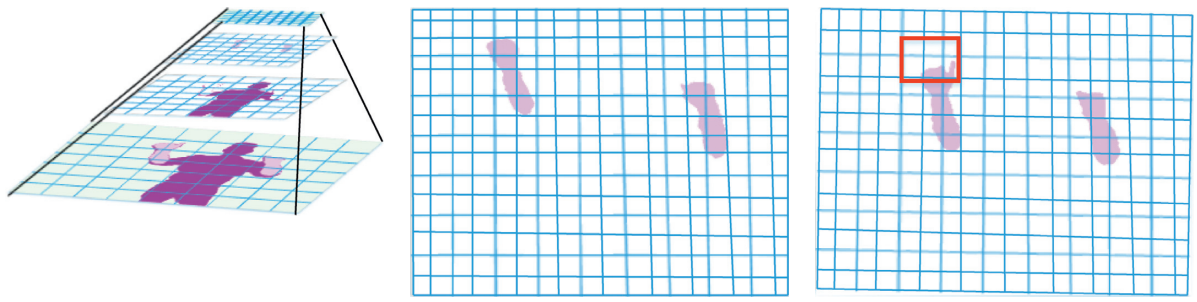
4. Experiments

In this section, the performance of the proposed method is evaluated by analysing the results without the assumption of global nonrigid and local rigid. Then, the method is implemented on Princeton Tracking Benchmark and Deformable 3D reconstruction dataset, and some qualitative or quantitative comparisons are presented.

4.1. Performance Evaluation on the Term on Spatial Constraints of Scene Flow. The term on spatial constraints of scene flow reflects the relationship between the scene flow of a pixel and its neighbourhood. To evaluate its performance, some experiments are implemented without this term.

In Figures 3 and 4, the scene flow is estimated without the spatial constraints of scene flow. For clarity, the figures for scene flow are not shrunk too much. It is obvious that scene flow loses smoothness in the same segmented region. That means the scene flow of pixels in the same region is discontinuous since the correlation of scene flow of pixels in the same region is not considered.

4.2. Princeton Tracking Benchmark. This dataset contains multiple independent moving targets and large areas of occlusion [28]. In this section, "Bear_back" sequence is used to test the method in this paper, and the results are shown in Figure 3. In the "Bear_back" sequence, the motion of the scene is produced by the opposite movement



(a) (b) (c)

Figure 2: Illustration for the assumption of global nonrigid and local rigid motion: (a) the depth layered image; (b) the layered result of the t^{th} frame k^{th} layer and (c) the layered result of the $t+1^{\text{th}}$ frame k^{th} layer. The block size is $3 * 3$.

of two hands mainly, in addition to some slight motion of the body.

In Figure 5, the first two columns are two consecutive images from Bear_back sequence, including RGB and depth images. The third column is the segmentation results, and K is set to 5 in the K-means clustering algorithm. The first and second rows of the fourth and fourth columns are the results of Sun's method [16] and our method, respectively. By comparing the scene flow in the red box between Sun's method and ours, it can be found that, under the same segmentation condition, the proposed method is closer to the moving region of the real image.

4.3. Deformable 3D Reconstruction Dataset [26]. Deformable 3D reconstruction dataset is a nonrigid dataset. In this paper, "Hat" and "Alex" sequence are used to test the proposed method, and different poses from different times are selected in these two sequences, respectively, to validate the proposed method is invariant to pose variation.

In "Hat" sequence, the motion is caused by the off-cap behaviour, and two poses are used, which is called Pose 1 and Pose 2. Pose 1 has small amplitude, involving the slight motion of hat, arm, and twist (Figure 6). Pose 2 includes the motion of hat mainly, and the direction of scene flow is the same basically (Figure 7). In Figures 6 and 7, the first two columns are the consecutive RGB and depth images, the third column is the segmented results with $K = 2$ in the K-means clustering algorithm, and the fifth column is the scene flow of Sun's method and ours. Occlusion calculation is an important part of Sun's method; therefore, the occlusions are also presented in this section.

In the fifth column of Figures 6 and 7, the estimation result of scene flow with Sun's method covers the whole human body which contains some stationary part. The reason for this problem may be that pixels in the same segmented region share a common rigid motion, which results in pixels without motion are also estimated scene flow. However, our method can estimate the scene flow of motion part, such as arm, head, and hat because each segmented region is divided into different blocks and the scene flow is estimated based on 3×3 block in each segmented region.

In "Alex" sequence, Pose 3 and Pose 4 are used. Pose 3 is produced by waving arms and some movement of clothes (Figure 8), and Pose 4 is obtained by the motion of arms (Figure 9). In the segmentation of "Alex" sequence, K is also set to 2. In Pose 3 and Pose 4, the motion amplitude of arms is greater than the rest of the human body. In Sun's method, the motion amplitude of the whole body is considered to be the same; however, the motion of arms is significantly greater than the rest of the body.

By comparing the scene flow estimation results visually (Figures 6~9), it can be found that our method can accurately estimate the scene flow of the nonrigid objects which involves different motion parts.

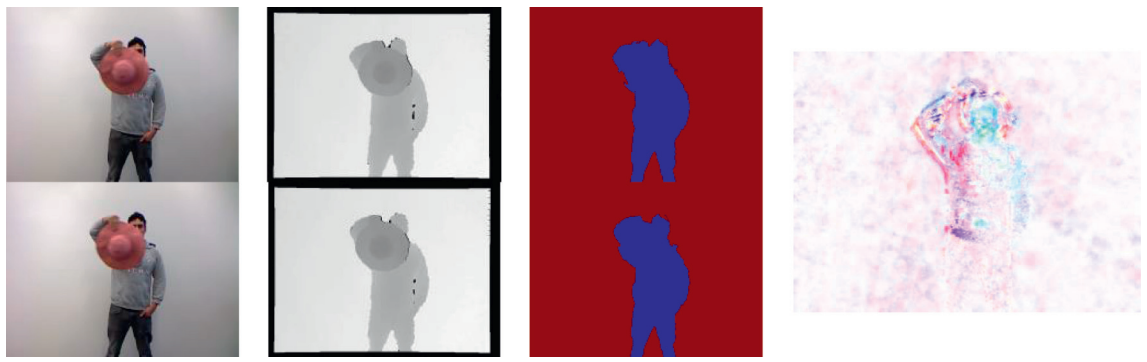
4.4. Evaluation Results. Quantitative results, RMS and AAE, are used to compare the proposed method and Sun's method.

RMS and AAE traverse all the pixels in the image, map the 3D scene flow acquired by the algorithm into a 2D optical flow, and compare it with the real optical flow value. The smaller the difference is, the more accurate the calculation is. Let that the estimated optical flow is $(u, v)^T$, and the true optical flow is $(u_{GT}, v_{GT})^T$, then the calculation formula of the RMS and AAE is as follows:

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{(x,y)}\left(\left(u_{GT}(x,y) - u(x,y)^2\right) + \left(v_{GT}(x,y) - v(x,y)\right)^2\right)},$$

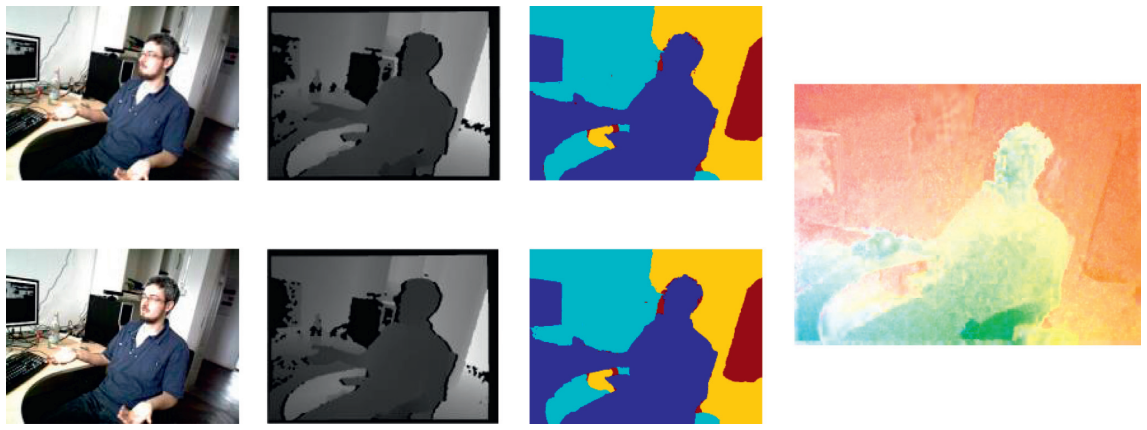
$$\text{AAE} = \frac{1}{N}\arccos\left(\frac{1 + u_{GT} \times u + v_{GT} \times v}{\sqrt{u_{GT}^2 + v_{GT}^2 + 1} \cdot \sqrt{u^2 + v^2 + 1}}\right), \tag{13}$$

where N is the number of pixels in the image.



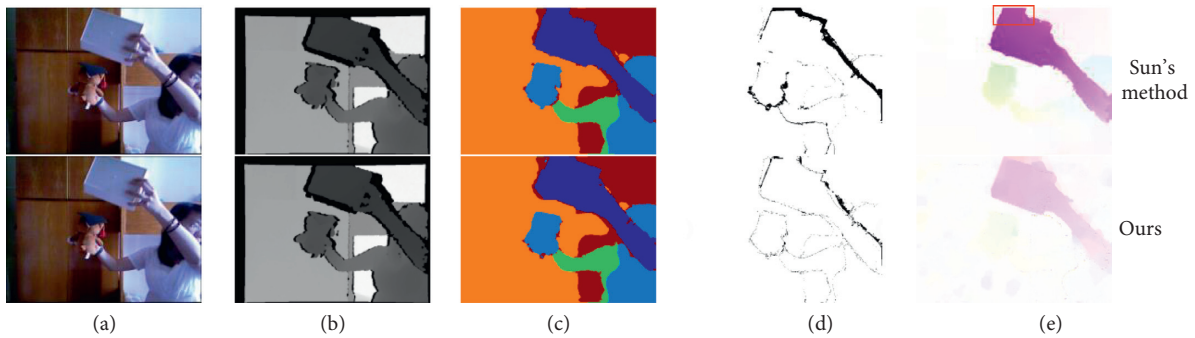
(a) (b) (c) (d)

Figure 3: "Hat" sequence [26] without spatial constraints of scene flow. Two consecutive frames from "Hat" sequence are input and segmented into 2 regions to estimate scene flow. (a) RGB images. (b) Depth images. (c) Segmentation $K = 2$. (d) Scene flow.



(a) (b) (c) (d)

Figure 4: SRSF 20 sequence [27] without spatial constraints of scene flow. Two consecutive frames from SRSF 20 sequence are input and segmented into 4 regions to estimate scene flow. (a) RGB images. (b) Depth images. (c) Segmentation $K = 4$. (d) Scene flow.



(a) (b) (c) (d) (e)

Figure 5: "Bear_back" sequence test results. Two consecutive frames from "Bear_back" sequence are input and segmented into 5 regions to estimate occlusion and scene flow. (a) RGB images. (b) Depth images. (c) Segmentation $K = 5$. (d) Occlusions. (e) Motion.

Errors of the method in this paper and Sun's are shown, respectively, in Figures 10(a) and 10(b), where the blue bar represents the errors of ours and the orange bars are the errors of Sun's method. From Figure 9, it is obvious that the blue bars are shorter than the orange bars, that is, RMS and AAE of the proposed method are lower than those of Sun's method in the test datasets.

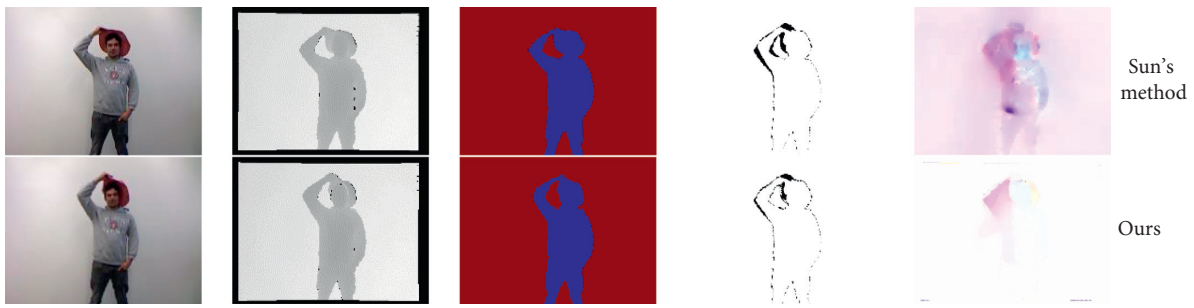


Figure 6: Pose 1 in "Hat" sequence." Input two consecutive frames about Pose 1 in "Hat" sequence, segment them into 2 regions (human body with hat and background), and estimate the occlusion and scene flow. (a) RGB images. (b) Depth images. (c) Segmentation $K = 2$. (d) Occlusions. (e) Motion.

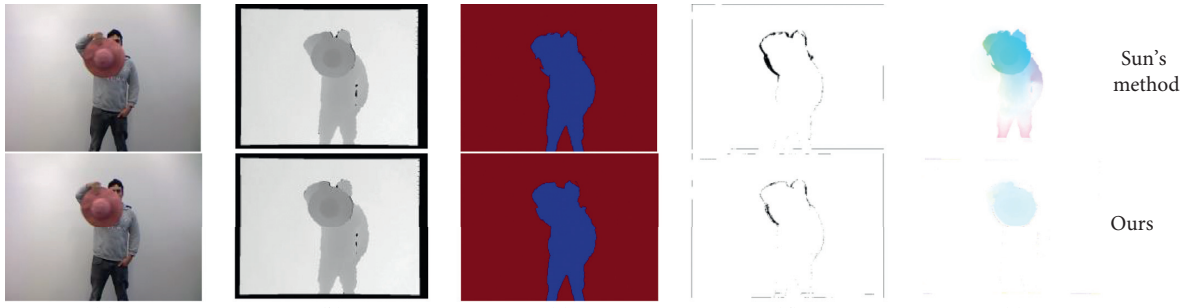


Figure 7: Pose 2 in "Hat" sequence." Input two consecutive frames about Pose 2 in "Hat" sequence. (a) RGB images. (b) Depth images. (c) Segmentation $K = 2$. (d) Occlusions. (e) Motion.

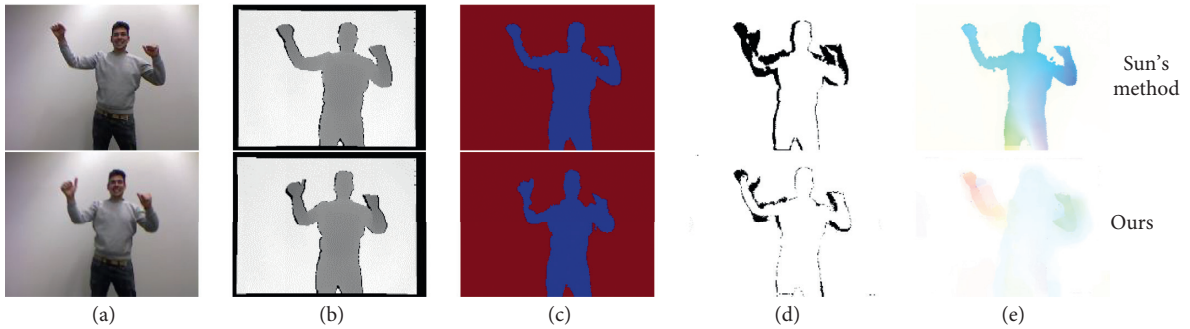


Figure 8: Pose 3 in "Alex" sequence. Input two consecutive frames about Pose 3 in "Alex" sequence. (a) RGB images. (b) Depth images. (c) Segmentation $K = 2$. (d) Occlusions. (e) Motion.

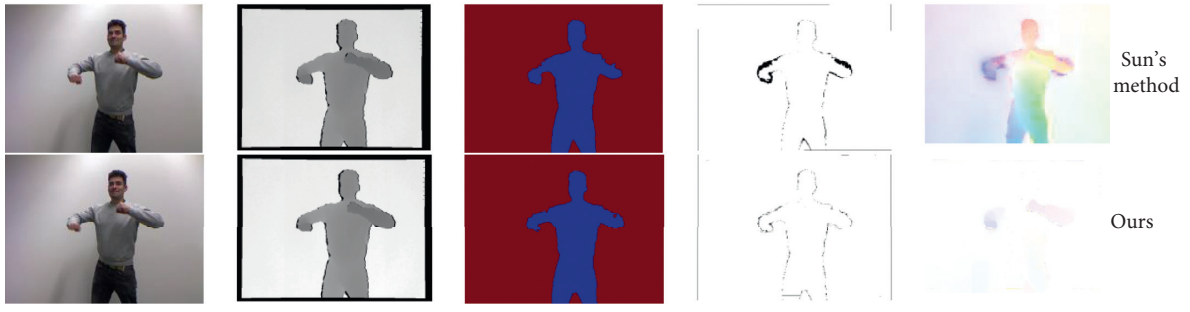


Figure 9: Pose 4 in "Alex" sequence. Input two consecutive frames about Pose 4 in "Alex" sequence. (a) RGB images. (b) Depth images. (c) Segmentation $K = 2$. (d) Occlusions. (e) Motion.

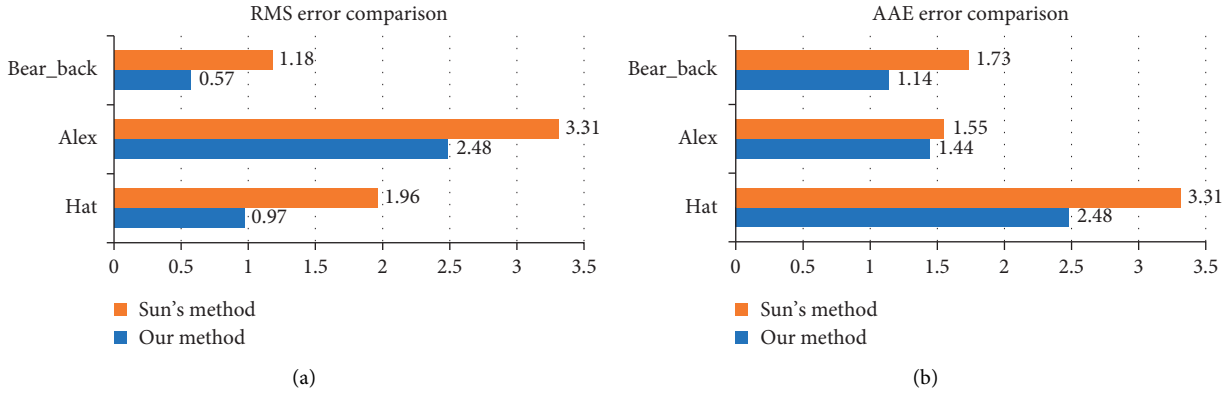


Figure 10: Comparisons of RMS and AAE. (a) RMS error. (b) AAE error.

5. Conclusions

In this paper, a RGBD scene flow estimation method with global nonrigid and local rigid motion assumption is presented. In this method, the preprocessing and the scene flow estimation are carried out. The preprocessing is used to get the registered RGB image and depth image, which would provide material for estimating scene flow. In the scene flow estimation, the K-means clustering algorithm is used to segment the depth image and process the occlusions, and then scene flow is estimated with the spatial-temporal correlation of the RGBD image sequence and global nonrigid and local rigid assumption in each segmentation region. To represent the global nonrigid and local rigid assumption, each segmented region is divided into a number of sufficiently small blocks since the pixels' motion in the same block is consistent and the pixels' motion in the different block is inconsistent. Experiments on different datasets and different poses show that the scene flow can be estimated more accurately with the proposed method.

However, the running time of the code is longer than [16] because each segmented region is divided into different blocks. In the future work, we will refer to the optimization of the model. For trained deep neural network methods can predict scene flow rapidly, we will refer to the existing methods to study learning-based methods.

Data Availability

The data used to support the findings of this study are available at http://tracking.cs.princeton.edu/dataset.html and http://campar.in.tum.de/personal/slavcheva/deformable-dataset/index.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank Deqing Sun et al. for providing the code. The authors would also gratefully acknowledge the support from the "International Conference on Brain Inspired Cognitive Systems-BICS" in 2019 for presenting our abstract which can be found in https://link.springer.com/chapter/10.1007/978-3-030-39431-8_21 [29]. This work was supported by the National Natural Science Foundation of China under grant nos. 6150238, 61501370, and 61703333.

References

[1] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Proceedings of the Seventh IEEE International Conference on IEEE Computer Vision*, vol. 2, pp. 722–729, Kerkyra, Greece, September 1999.

[2] D. Zhou, V. Frémont, B. Quost, Y. Dai, and H. Dai, "Moving object detection and segmentation in urban environments from a moving platform," *Image and Vision Computing*, vol. 68, pp. 76–87, 2017.

[3] S. Javed, T. Bouwmans, M. Shah, and S. Jung, "Moving object detection on RGB-D videos using graph regularized spatio-temporal RPCA," in *Proceedings of the International Conference on Image Analysis and Processing*, pp. 1–11, Catania, Italy, September 2017.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, Rhode Island, USA, June 2012.

[5] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (CVPR)*, pp. 3061–3070, Boston, MA, USA, June 2015.

[6] X. Zuo, S. Wang, J. Zheng, and R. Yang, "High-speed depth stream generation from a hybrid camera," in *Proceedings of the 2016 ACM International Conference on Multimedia (ACM MM)*, pp. 878–887, Klagenfurt Austria, May 2016.

[7] S. Wang, X. Zuo, C. Du, R. Wang, J. Zheng, and R. Yang, "Dynamic non-rigid objects reconstruction with a single RGB-D sensor," *Sensors*, vol. 18, no. 3, p. 886, 2018.

[8] D. Xiao, Q. Yang, B. Yang, and W. Wei, "Monocular scene flow estimation via variational method," *Multimedia Tools and Applications*, vol. 76, no. 8, pp. 10575–10597, 2017.

[9] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Proceedings of the the 11th International Conference on Computer Vision*, pp. 1–7, Rio de Janeiro, Brazil, October 2007.

[10] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1377–1384, Sydney, Australia, December 2013.

[11] J. Quiroga, F. Devernay, and J. Crowley, "Scene flow by tracking in intensity and depth data," in *Proceedings of the CVPRW 2012–Computer Vision and Pattern Recognition Workshops IEEE*, pp. 50–57, Ostrava, Czech Republic, January 2012.

[12] S. Hadfield and R. Bowden, "Kinecting the dots: particle based scene flow from depth sensors," in *Proceedings of the 2012 IEEE Conference on Computer Vision*, pp. 2290–2295, Rhode Island, USA, June 2012.

[13] J-M Gottfried, J. Fehr, and C. S. Garbe, "Computing range flow from multi-modal kinect data," in *Proceedings of the International Symposium on Visual Computing*, pp. 758–767, Las Vegas, NV, USA, September 2011.

[14] T. Taniai, S. N. Sinha, and Y. Sato, "Fast multi-frame stereo scene flow with motion segmentation," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pp. 6891–6900, Honolulu, HI, USA, July 2017.

[15] Z. Lv, C. Beall, P. F. Alcantarilla et al., "A continuous optimization approach for efficient and accurate scene flow," in *Proceedings of the European Conference on Computer Vision*, pp. 757–773, Amsterdam, The Netherlands, October 2016.

[16] D. Sun, E. B. Sudderth, and H. Pfister, "Layered RGBD scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 548–556, Boston, MA, USA, June 2015.

[17] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers, "A primal-dual framework for real-time dense RGB-D scene flow," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 98–104, Seattle, WA, USA, May 2015.

[18] V. Golyanik, K. Kim, R. Maier et al., "Multiframe scene flow with piecewise rigid motion," in *Proceedings of the International Conference on 3D Vision (3DV)*, pp. 273–281, Qingdao, China, October 2017.

[19] X. Xiang, M. Zhai, R. Zhang, W. Xu, and A. EI Saddik, "Scene flow estimation based on 3D local rigidity assumption and depth map driven anisotropic smoothness," *IEEE Access*, vol. 6, pp. 30012–30023, 2018.

[20] Y. Wang, J. Zhang, Z. Liu et al., "Handling occlusion and large displacement through improved RGB-D scene flow estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 7, pp. 1265–1278, 2016.

[21] R. Schuster, O. Wasenmuller, G. Kuschk, C. Bailer, and D. Stricker, "Sceneflowfields: dense interpolation of sparse scene flow correspondences," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1056–1065, Lake Tahoe, NV, USA, March, 2018.

[22] W. Song, A. V. Le, S. Yun, S-W. Jung, and C. S. Won, "Depth completion for kinect v2 sensor," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4357–4380, 2017.

[23] Y. Zhang, J. Dai, H. Zhang, and L. Yang, "Depth inpainting algorithm of RGB-D camera combined with color image," in *Proceedings of the 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1391–1395, Xi'an, Shaanxi, China, May 2018.

[24] D. Sun, E. Sudderth, and M. Black, "Layered image motion with explicit occlusions, temporal consistency, and depth ordering," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 2226–2234, Vancouver, British Columbia, Canada, December 2010.

[25] D. Sun, J. Wulff, E. Sudderth, H. Pfister, and M. Black, "A fully-connected layered model of foreground and background flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2451–2458, Portland, OR, USA, June 2013.

[26] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic, "KillingFusion: non-rigid 3D reconstruction without correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5474–5483, Honolulu, HI, USA, July 2017.

[27] J. Quiroga, T. Brox, F. Devernay, and J. Crowley, "Dense semi-rigid scene flow esti- mation from rgbd images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 567–582, Zürich, Switzerland, September 2014.

[28] http://tracking.cs.princeton.edu/dataset.html.

[29] X. Li, Y. Liu, H. Jin, L. Cai, and J. Zheng, "Layered RGBD scene flow estimation with global non-rigid local rigid assumption," in *Advances in Brain Inspired Cognitive Systems*, vol. 11691, BICS, Brussels, Belgium, 2019.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 9475782, 11 pages
https://doi.org/10.1155/2020/9475782

Research Article

Fuzzy Matching Template Attacks on Multivariate Cryptography: A Case Study

Weijian Li,[1] **Xian Huang**,[1] **Huimin Zhao**,[1] **Guoliang Xie**,[2] **and Fuxiang Lu**[1]

[1]*School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*
[2]*Dept of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G11XQ, UK*

Correspondence should be addressed to Huimin Zhao; zhaohuimin@gpnu.edu.cn

Received 19 April 2020; Accepted 25 May 2020; Published 20 June 2020

Guest Editor: Jianbiao Zhang

Multivariate cryptography is one of the most promising candidates for post-quantum cryptography. Applying machine learning techniques in this paper, we experimentally investigate the side-channel security of the multivariate cryptosystems, which seriously threatens the hardware implementations of cryptographic systems. Generally, registers are required to store values of monomials and polynomials during the encryption of multivariate cryptosystems. Based on maximum-likelihood and fuzzy matching techniques, we propose a template-based least-square technique to efficiently exploit the side-channel leakage of registers. Using QUAD for a case study, which is a typical multivariate cryptosystem with provable security, we perform our attack against both serial and parallel QUAD implementations on field programmable gate array (FPGA). Experimental results show that our attacks on both serial and parallel implementations require only about 30 and 150 power traces, respectively, to successfully reveal the secret key with a success rate close to 100%. Finally, efficient and low-cost strategies are proposed to resist side-channel attacks.

1. Introduction

With the upcoming quantum computers, traditional cryptosystems face huge challenges. Public-key cryptosystems such as Rivest–Shamir–Adleman (RSA) and elliptic curve cryptography (ECC), whose security relies on the difficulty of certain number theoretic problems, are under great threat of quantum attack. In as early as 1994, Peter Shor proposed an algorithm on a quantum computer that efficiently solved such number theoretic problems in polynomial time. Afterward, Monz et al. [1] presented the realization of a scalable Shor algorithm in 2016, which means that once large-scale quantum computers appear, public-key cryptosystems will become insecure. Meanwhile, for symmetric-key primitives, larger keys are required to resist the quantum attack to some extent.

Since Shor's discovery, the theory of post-quantum cryptography has developed significantly. Many cryptographic schemes proposed in the literature, such as code-based cryptography [2] and lattice-based cryptography [3], show great potentiality to resist quantum attacks, while multivariate cryptography is one of the most promising candidates [4]. Afterward, numerous cryptosystems based on multivariate quadratic polynomials have been proposed, such as unbalanced oil-and-vinegar (UOV) and its variant [5], Rainbow [5, 6], ZHFE [7], and CHNN-MVC [8].

At Eurocrypt 2006, Berbain et al. [9] presented the first multivariate stream cipher scheme denoted as QUAD, which is referred to as a practical and provable secure stream cipher, as well as a pseudorandom number generator (PRNG). In 2009, Berbain et al. [10] revisited the stream cipher of QUAD and proposed the provable security arguments supporting its conjectured strength for suitable parameter values. The provable security of QUAD relies on the hardness of solving systems of multivariate quadratic equations. Bardet et al. [11] presented a cryptanalysis algorithm with a complexity bounded by $O(2^{134.56})$, which means this cryptanalysis method cannot put into practice.

In recent years, GPUs are widely used in cloud computing and blockchain, which faces huge security challenges

to guarantee data security and user privacy [12–14]. Several GPU acceleration schemes for multivariate systems are proposed to make it suitable for security of cloud computing and blockchain in the quantum world [15, 16]. In 2014, Tanaka et al. [15] proposed two efficient parallelization algorithms and a GPU-based multivariable quadratic polynomial system. Furthermore, they proposed several effective parallel implementations of QUAD on GPU to accelerate the computing of quadratic polynomials. In 2018, Liao et al. [16] proposed a GPU acceleration framework for high-order multivariate cryptography systems, where the GPU acceleration schemes made multivariate cryptosystems feasible for cloud computing and blockchain.

Moreover, multivariate cryptosystems are in general computationally efficient, which supports the use of the Internet of Things (IoT) devices. IoT is essentially a network of pervasive devices such as RFID tags, sensors, ASICs, and smart cards, which have rigid cost constraints in terms of area, memory, computing power, and battery supply. Traditional cryptosystems are not entirely applicable to the IoT devices since they are too expensive for such pervasive devices. At fast software encryption (FSE) 2010, Billet et al. [17] showed that QUAD can be converted to efficiently construct a privacy-preserving authentication protocol for RFID with provable security. Arditti et al. [18] presented a QUAD implementation and regarded it as the smallest provably secure stream cipher so far. The smallest QUAD implementation requires only 2961 GE, which makes it a competitive candidate for IoT security. Also, Hamlet et al. [19] proposed a throughput-optimized parallel implementation of QUAD for more secure application scenarios in 2015.

The implementation of cryptography needs to take a wide range of physical attacks into account, especially side-channel attacks and fault attacks. Side-channel attacks exploit the dependency between physical information (e.g., power consumption, electromagnetic leaks, and timing information) and secret key to enable a divide-and-conquer attack to reveal the key part by part. Typical side-channel attacks include nonprofiled attacks (e.g., correlation power analysis (CPA) [20], mutual information analysis (MIA) [21]) and profiled attacks (e.g., template attacks (TA) [22–28] and other machine learning-based side-channel attacks [28–34]). Profiled side-channel attacks are the most powerful attacks, which received a lot of attention in recent years. Samples of power traces are regarded as features, and feature selection methods are needed to reduce the computational complexity and increase the prediction accuracy [28]. Afterward, machine leaning techniques including maximum-likelihood strategy [22–28], SVM [28–30], random forest (RF) [28, 29], k-nearest neighbors (KNN) [31], neural networks (NNs) [32], and deep learning (DL) [33, 34] are widely applied to build the prediction model. Profiled side-channel attacks include a profiling/training phase and a matching/predicting phase. In the profiling/training phase, machine learning algorithms are fed with labelled power traces captured from a reference device to build the prediction model. In the matching/predicting phase, prediction models are used to predict the correct labels for those power traces captured from a target device.

Template attack was first proposed at CHES'02 [22], which efficiently revealed the key by a maximum-likelihood strategy, and was rapidly accepted as the strongest form of side-channel attack. Original template attack matched only a single power trace, which sometimes failed in the practical attack. Agrawal et al. [23] proposed template-based DPA attack to accumulate the matching results of power traces, which significantly improved the success rate. Özgen et al. [24] combined classification algorithms with template attacks in the matching phase to improve the efficiency of attacks. Choudary and Kuhn [25] tackled some of the practical obstacles of template attacks, such as pooled covariance matrices, compression methods, and incompatibility of templates across different devices. Zhang [27] theoretically analyzed the exact relationship between the success rate of template attack and values of different parameters, including signal-to-noise, number of interesting points, and number of power traces. From the viewpoint of machine learning, Picek et al. [28] adopted feature selection techniques to improve the attack efficiency. They concluded that L1 regularization wrapper and linear SVM hybrid methods performed consistently well for all data sets.

Although side-channel attacks have been developed over 20 years, research about side-channel attacks on multivariate cryptosystems is still in the early stages. Several literatures about side-channel attack on multivariate cryptosystems were published. In as early as 2005, Okeya et al. [35] analyzed the power leakage of addition operations modulo 2^{32} of SHA-1 and successfully recovered the secret information of SFLASH, which is the first successful power analysis attack on multivariate cryptography in practice. Later, in 2013, Hashimoto et al. [36] proposed a theoretical method based on fault attack to reveal the partial key of MPKC systems. Yi and Li [37] proposed a fault attack and DPA on ASIC implementation of enTTS scheme in 2017. In 2018, Park et al. [38] presented a correlation power analysis attack against the Rainbow and UOV schemes on an 8-bit AVR microcontroller that yields full secret key recoveries. In 2019, based on the work of Hashimoto et al., Krämer and Loiero [39] complemented the research on fault attacks of multivariate signature schemes. However, their attacks do not lead to complete key recovery on Rainbow and UOV. Recently, Li et al. [40] proposed a CPA attack against serial implementation of QUAD on FPGA. Their work efficiently revealed the secret key but still requires further work to improve success rate.

Li et al. proposed the practical CPA cryptanalysis on serial QUAD (2, 160, 160) with a much lower complexity, but the success rate is only around 85%. Because of the low signal-to-noise ratio, classic template attack and template-attack DPA attack cannot exactly match the templates to achieve a satisfactory success rate. To tackle this issue, we have proposed template-based least-square power analysis on serial QUAD (2, 160, 160). The main contributions of our paper can be highlighted as follows:

(1) By applying the least-square technique to enable fuzzy matching of the templates, which can find the best matching via minimizing the squared sum of errors. As a result, the proposed practical can achieve a success rate of nearly 100%.

(2) We also extend the template-based least-square power analysis attack to explore the leakage of parallel implementation of QUAD (2, 160, 160), which has successfully and efficiently revealed the secret key with a success rate also close to 100%.

(3) For multivariate cryptography, all monomials and polynomials can be computed in an arbitrary order to break the link between the power consumption and the secret key. We propose two low-cost hiding countermeasures for serial and parallel implementations, respectively, which show great potential to resist side-channel attacks.

The remaining paper is organized as follows: in Section 2, we review the mathematical definition, serial and parallel FPGA implementations of the QUAD stream cipher; in Section 3, the template-based least-square power analysis attacks on the serial and parallel FPGA implementation of the QUAD are presented; experimental results of our attacks are given in Section 4; efficient and low-cost countermeasures to resist side-channel attacks are discussed in Section 5; and Section 6 concludes the paper.

2. Preliminaries

2.1. Multivariate Cryptography. Generally, the mathematical definition of a multivariate quadratic equation with n variables over GF (q) can be written as follows:

$$Q(x_1, \ldots, x_n) = \sum_{1 \le i \le j \le n} \alpha_{ij} x_i x_j + \sum_{1 \le i \le n} \beta_i x_i + \gamma, \quad (1)$$

where α_{ij}, β_i, and γ are all coefficients over GF (q). Note that the degree of polynomial is up to 2; otherwise, new variables will be introduced to keep the polynomial of degree 2. A multivariate quadratic system $Q(X)$ consisting of m multivariate quadratic equations in n variables over GF (q) is defined as

$$Q(X) = \{Q_1(X), \ldots, Q_m(X)\}, \quad X = x_1, \ldots, x_n. \quad (2)$$

Given a multivariate quadratic system $Q(X)$, the MQ problem is defined as to find a value $X = x_1, \ldots, x_n$, if any, such that $Q_i(X) = 0$ for all $1 \le i \le m$. The MQ problem is proved to be NP hard, even in the smallest finite filed GF (2) [10].

A particular QUAD stream cipher in n variables over GF (q) is specified as QUAD (q, n, r), which computes $n + r$ polynomials per round. As shown in Figure 1, QUAD (q, n, r) consists of an output function $S_{\text{out}}(X) = (Q_{n+1}(X), \ldots, Q_{n+r}(X))$ to produce r outputs as the keystream, and an update function $S_{\text{in}}(X) = (Q(X)1, \ldots, Q_n(X))$ is used to generate n outputs to update X for the next round. The parameters q, n, and r and the coefficients α_{ij}, β_i, and γ for S_{in} and S_{out} are public.

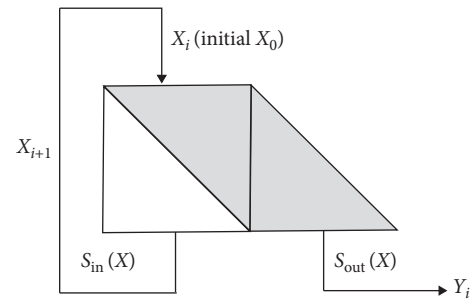


Figure 1: Stream cipher generation in QUAD [9].

The QUAD cipher expands a secret initial state $X_0 \in \text{GF}(q)^n$ into a sequence of secret states $X_0, X_1, X_2, \ldots \in \text{GF}(q)^n$ and a sequence of output vectors $Y_0, Y_1, Y_2, \ldots \in \text{GF}(q)^r$.

QUAD (2, 160, 160) is a practical version with the security level of at least 2^{80}, which is strongly recommended in [10]. QUAD (2, 160, 160) has 160 variables over GF (2), which outputs 160 bits per round, resulting in a set of 320 multivariate quadratic equations.

From a perspective of implementation, operations over GF (2) are more efficient than those over larger fields. Moreover, the monomial forms $x_i \cdot x_i$ and x_i are equal over GF (2); therefore, $\alpha_{ij} x_i x_j$ and $\beta_i x_i$ can be computed together. In the case of randomly generated α_{ij} and γ, equations of QUAD over GF (2) can be simplified as

$$Q(X) = \sum_{1 \le i \le j \le n} \alpha_{ij} x_i x_j + \gamma, \quad (3)$$

which brings great benefits in terms of efficiency and security.

2.2. FPGA Serial Implementation of QUAD. Arditti et al. [18] proposed a compact serial implementation of QUAD, which is believed to be the smallest provably secure stream cipher. As shown in Figure 2, the implementation consists of two main components. The first one is a nonlinear feedback shift register (NFSR), in which the coefficients of α and γ are randomly generated. Each monomial of the equation is computed by the second component at every clock tick and accumulated to a result register. Multivariate quadratic equations $Q_1(X)$, $Q_2(X)$, ..., $Q_{n+r}(X)$ are computed sequentially. At every clock tick, the NFSR generates the coefficient. Once a new monomial $\alpha_{ij} x_i x_j$ of polynomial $Q_k(X)$ is computed, its contribution will be accumulated to the temporary register Q_k. After $n(n+1)/2 + 1$ clock cycles, the polynomial $Q_k(X)$ is computed, and the above process is repeated for $Q_{k+1}(X)$.

2.3. FPGA Parallel Implementation of QUAD. Hamlet and Brocato [19] presented two throughput-optimized parallel implementations of QUAD for a much higher throughput. A QUAD (2, 128, 128) version with the security level of approximately 2^{64} is considered, which can be easily extended to another version in GF (2) such as QUAD (2, 160, 160). The coefficients α and γ are randomly generated and stored in ROM. Multivariate quadratic equations

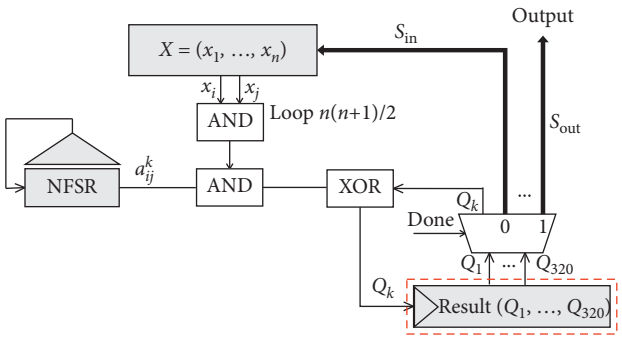


Figure 2: Serial implementation of QUAD (2, 160, 160) [18].

$Q_1(X), Q_2(X), \ldots, Q_{n+r}(X)$ are still computed sequentially, while n monomials of polynomial $Q_k(X)$ are computed in parallel at a time to achieve a higher throughput.

As shown in Figure 3, the FPGA parallel implementation of QUAD (2, 160, 160) is summarized as follows:

(1) Sequentially compute the equations $Q_1(X)$, $Q_2(X)$, ..., $Q_{320}(X)$ by step (2) to step (6).

(2) Initialize the internal state X and rotated X with the secret key and the initialization vector.

(3) Calculate $\eta_i = x_i$ AND rotated x_i, $1 \le i \le 160$ simultaneously.

(4) Load 160-bit coefficients α from ROM, feed η and α into 8-input AND-XOR modules to compute $M_c = \sum_{3 \ge i \ge 0} \alpha_{4c-i} \eta_{4c-i}$, $1 \le c \le 40$, and store M_c in temporary register P_c.

(5) Calculate $V_{c_1} = \sum_{3 \ge i \ge 0} M_{4c_1-i}$, $1 \le c_1 \le 10$, by XOR modules. Compute $Q_k(X) = Q_k(X) \oplus \sum_{1 \le i \le 10} V_i$ and store the value in result register Q.

(6) Rotate the internal state rotated X by one bit, and go to step 3 to compute next 160 monomials until all monomials of polynomial $Q_k(X)$ are completed, which requires $\lceil (n+1)/2 \rceil = 81$ loops. Note that, in the last loop, only half of the above modules are enabled to compute the last 80 monomials.

(7) Repeat the above steps until all quadratic equations are computed.

3. Proposed Attack on Implementation of QUAD

3.1. Power Leakage Model. A typical CMOS transistor consumes dynamic power when its output signal is converted. Figure 4(a) shows the changing process of a register when the output signal is converted from 0 to 1. A charging current from the power supply to the output capacitance C_L and a transient short-circuit current from CMOS transistor are generated. On the contrary, Figure 4(b) shows the discharging process when the output signal is converted from 1 to 0. Only the instantaneous short-circuit current is generated through CMOS transistor.

As a result, conversions of the output signal are focused since dynamic power is the major power consumption of the digital logical circuits of ASIC and FPGA. Denote the power consumption of CMOS transistor by P_{ij} when its signal converts from i to j, where i and j equal to 0 or 1. P_{01} and P_{10} consume dynamic power, while P_{00} and P_{11} consume only static power. As a result, it generally holds that $P_{00} \approx P_{11} \ll P_{01}, P_{10}$.

Therefore, the power consumption when writing data to a register depends on the number of bit-flips. A hamming distance (HD) model well summarizes the power consumption of a register transition from a previous state to a new state.

Regarding multivariate cryptosystems, which consist of a large number of monomials and polynomials, registers are indeed required to store monomial and polynomial values during the encryption. Serial implementation, for instance, monomials are computed sequentially and accumulated to the temporary register Q_k, as identified by rectangle in Figure 2.

The value of register Q_k changes to $Q_k \oplus \alpha_{ij} x_i x_j$ for all monomials. The power consumption of register Q_k can be concluded as follows:

$$L(Q(x)) = HD\left(Q_k, Q_k \oplus \alpha_{ij} x_i x_j\right) = HW\left(\alpha_{ij} x_i x_j\right), \quad 1 \le k \le 160. \tag{4}$$

Consequently, an attacker is possible to predict secret keys x_i and x_j by observing the power consumption of registers Q_l.

Other than serial implementations, parallel implementations compute 160 monomials simultaneously. 4 monomials are accumulated by an AND-XOR module and stored into temporary register P_c. According to the parallel implementation described in Section 2.3, when computing the first 160 monomials, the values M_c, $1 \le c \le 40$, stored into the temporary register P_c are

$$M_c = a_{4c-3,4c-3} x_{4c-3} x_{4c-3} \oplus a_{4c-2,4c-2} x_{4c-2} x_{4c-2} \oplus a_{4c-1,4c-1} x_{4c-1} x_{4c-1} \oplus a_{4c,4c} x_{4c} x_{4c}, \quad 1 \le c \le 40. \tag{5}$$

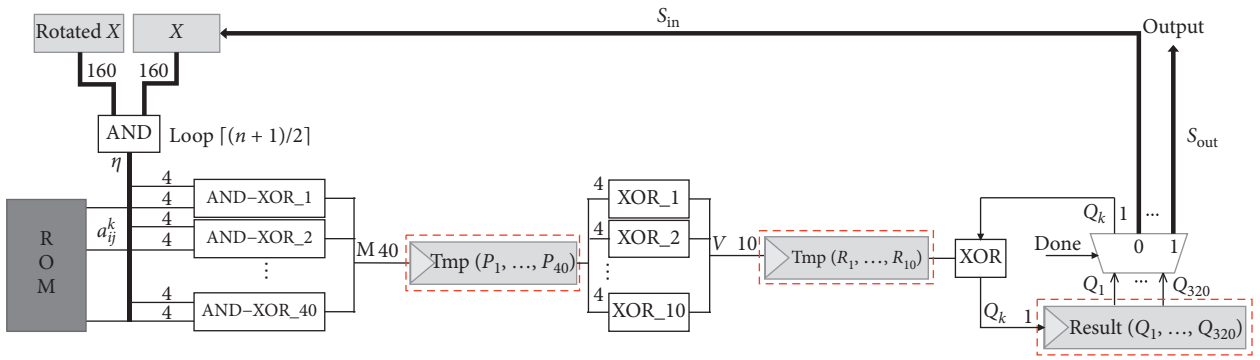


Figure 3: Parallel implementation of QUAD (2, 160, 160) [19].

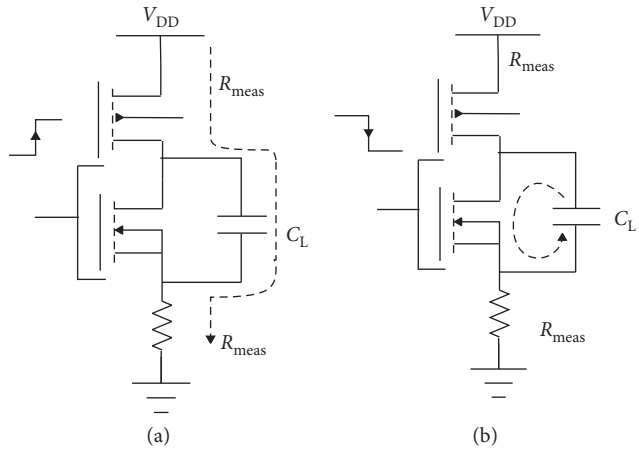


Figure 4: Charging and discharging process of register.

After rotating the internal state rotated x by one bit to compute the next 160 monomials, the values $M_c', 1 \le c \le 40$, stored into registers P_c are

$$\begin{aligned} M_c' &= a_{4c-3,4c-2}x_{4c-3}x_{4c-2} \oplus a_{4c-2,4c-1}x_{4c-2}x_{4c-1} \oplus a_{4c-1,4c}x_{4c-1}x_{4c} \oplus a_{4c,4c+1}x_{4c}x_{4c+1}, \quad 1 \le c \le 39, \\ M_c' &= a_{157,158}x_{157}x_{158} \oplus a_{158,159}x_{158}x_{159} \oplus a_{159,160}x_{159}x_{160} \oplus a_{1,160}x_{160}x_1, c = 40. \end{aligned} \tag{6}$$

Therefore, the values of registers $P_c, 1 \le c \le 40$, change from M_c to M_c', and the power leakage model of parallel implementations can be defined as

$$L(Q(x)) = HD(M_c, M_c') = HW(M_c \oplus M_c'), \quad 1 \le c \le 40. \tag{7}$$

3.2. Template-Based Least-Square Power Analysis Attack. Classic side-channel attacks, such as DPA, CPA, and MIA, require a large number of power traces to reveal the key, which means that different plaintexts are needed to be encrypted with the same key for obtaining as much power traces as possible. However, multivariate cryptosystems usually contain limited quadratic equations. Take QUAD as an example, and the key of QUAD is constantly updated after each round of encryption, which only generates $n + r$ power traces with the same key. In this case, machine learning-based side-channel attacks such as template attack have inherent advantages, which can extract the key with much fewer target power traces.

Machine learning-based side-channel attacks are the most powerful attacks. Based on a maximum-likelihood strategy, template attacks reveal the secret key efficiently, which consist of a profiling phase and a matching phase. Classic template attacks match only a single power trace and reveal the key by the Bayes theorem in the matching phase. However, there is not enough valuable information in a

single power trace to reveal the correct key in practical situation; hence, the classic template attack has little prospect of success rate.

To solve this problem, template-based DPA attack was proposed as follows [23]:

$$p\left(k_j \mid T\right) = \frac{\left(\prod_{1\le i\le D} p\left(t_i \mid k_j\right)\right)\cdot p\left(k_j\right)}{\sum_{1\le l\le K}\left(\left(\prod_{1\le i\le D} p\left(t_i \mid k_l\right)\right)\cdot p\left(k_l\right)\right)}, \tag{8}$$

which accumulates the matching degree of each power trace during template matching to improve the success rate.

Unfortunately, due to the low signal-to-noise ratio, the accurate matching method of template-based DPA attack is not applicable. For this reason, we proposed a template-based least-square (LSQ) power analysis attack, which reveals the key by fuzzy matching. As described in Figure 5, the main idea of template-based LSQ is as follows:

(1) Choose a strategy to build templates: according to the power leakage models in equations (4) and (7), two templates need to be built, corresponding to leakage values 0 and 1.

(2) Collect power traces to build templates: two groups of power traces are collected according to different leakage values.

(3) Select interesting points (features): samples of power traces are regarded as features, which include relevant, irrelevant, and redundant features. Feature selection methods are needed to select the most relevant features to improve the attack efficiency. Feature selection methods [28] such as squared pairwise T-differences (SOST), Pearson correlation, principal component analysis (PCA), linear SVM wrapper, and L1 regularization are investigated. In our experiments, the Pearson correlation method is chosen to search interesting points, which can lead to excellent classification performance. 25 interesting points for serial implementation and 35 interesting points for parallel implementation with the highest $\rho_{t,hw}$ are selected, where $\rho_{t,hw}$ is defined as

$$\rho_{t,hw} = \frac{\operatorname{cov}(t, hw)}{\sigma_t \sigma_{hw}}. \tag{9}$$

(4) Build templates with interesting points: two templates $h_i = (m_i, C_i)$ corresponding to leakage values 0 and 1 are built, respectively, by covariance matrix C_i and mean vector m_i, where m_i and C_i are defined as

$$m_i = \frac{1}{d}\sum_{1\le j\le d} t_j^i,$$
$$C_i(u,v) = \frac{1}{d-1}\sum_{1\le l\le d}\left(N_{ul} - \overline{N}_u\right)\cdot\left(N_{vl} - \overline{N}_v\right), \tag{10}$$

(5) Match templates: power traces $T = \{t_1, t_2, \ldots, t_D\}$ with the same key are captured from the device under attack to match the templates, respectively. Template that leads to the highest probability $p(t_j; (m_i, C_i))$ indicates the correct leakage value, where $p(t_j; (m_i, C_i))$ is defined as

$$p\left(t_j; (m_i, C_i)\right) = \frac{\exp\left(-(1/2)\cdot\left(t_j - m_i\right)'\cdot C_i^{-1}\cdot\left(t_j - m_i\right)\right)}{\sqrt{(2\cdot\pi)^T\cdot\det\left(C_i\right)}}. \tag{11}$$

Denote such leakage values corresponding to $T = \{t_1, t_2, \ldots, t_D\}$ as $H = [h_1, h_2, \ldots, h_D]$.

(6) Reveal the correct key. We map the hypothetical intermediate values into leakage values by equation (7) and compare with $H = [h_1, h_2, \ldots, h_D]$ to reveal the correct key. Taking attack on parallel implementations, for instance, denote the hypothetical leakage values by $S = \{s_1, s_2, \ldots, s_{32}\}$, where $s_i = \{s_{i1}, s_{i2}, \ldots, s_{iD}\}$. The least-square method defined as

$$F(i) = \sum_{1\le j\le D}\left(s_{ij} - h_j\right)^2, \quad 1\le i\le 32, 1\le j\le D, \tag{12}$$

is applied to compare the hypothetical leakage values with the leakage values revealed by the template attack, where i is the key hypothesis. Finally, the correct key is revealed by

$$\text{key} = \arg\min F(i). \tag{13}$$

4. Experimental Results and Discussion

As shown in Figure 6, our experimental setup includes a standard evaluation board SAKURA-G, an oscilloscope, and a computer. SAKURA-G is designed for hardware security, which equips with two separate Spartan-6 FPGA chips. One chip serves as the control chip, while another serves as the cryptographic chip. Cryptographic chip performs encryption operations, while the control chip controls the data flow and communicates with the oscilloscope and computer. During encryption, power consumptions of the cryptographic chip are measured by the oscilloscope which is triggered by the control chip. Finally, power analysis attacks are performed on the computer.

We first perform a side-channel attack on serial implementation of QUAD (2, 160, 160). In the template building phase, 3000 power traces with different keys and coefficients are captured from a reference device, based on which 25 interesting points are selected by the CPA peak method. Next, we collect two groups of power traces corresponding to leakage values 0 and 1 in equation (4), and each group consists of 25 power traces. Finally, we build two templates, and the result is shown in Figure 7.

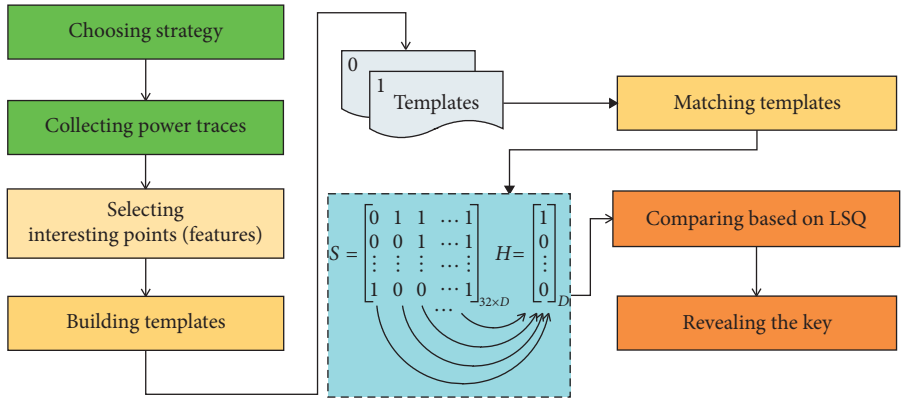


Figure 5: The flow of template-based LSQ attack.

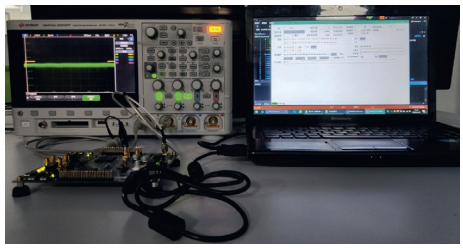


Figure 6: Experimental setup.

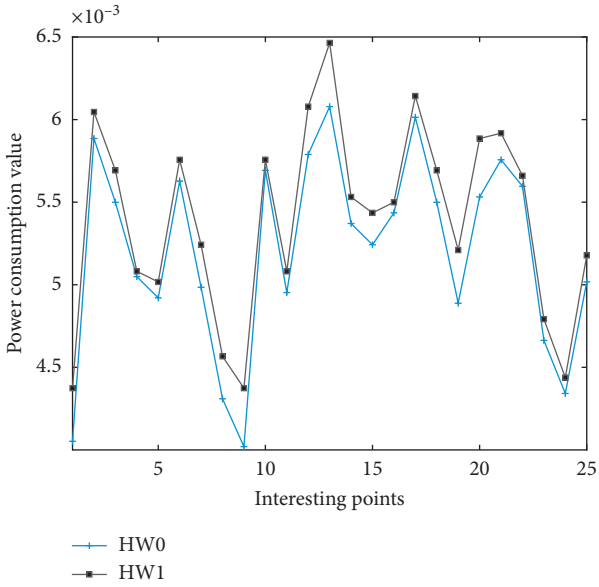


Figure 7: Templates built for serial QUAD.

In the template matching phase, 320 power traces with the same key are captured from the target device. Template that leads to the highest probability indicates the correct key. The success rate of template-based LSQ attack on serial implementation is shown in Figure 8. When the number of power traces approach 30, the success rate tends to 100%. Therefore, a successful attack only requires 30 power traces in serial implementation.

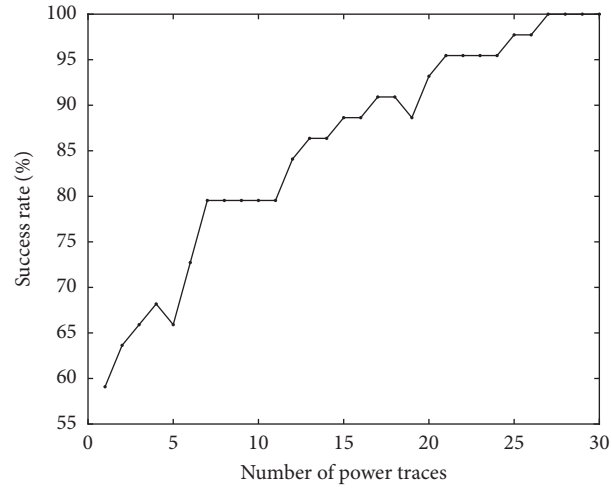


Figure 8: Success rates of our attack on serial implementation.

To further illustrate the effectiveness of our attack, the time required for our practical attack on serial QUAD is discussed here. Our attack is performed on a personal computer, which integrates an Intel i5-7500 CPU and 12 GB of RAM. The time for templates building and templates matching depends on the number of power traces for building and matching, respectively. Figure 9 shows the time required for the template building with the number of power traces ranging from 1 to 3000. Figure 10 shows the time required for template matching with the number of power traces ranging from 1 to 30. As our successful attack on the serial implementation of QUAD (2, 160, 160) requires less than 3000 power traces for templates building and 30 power traces for templates matching, the total time required for our successful attack is less than 1010 seconds, according to Figures 9 and 10.

According to the leakage model of parallel implementation in equation (7), 4 bits of the key are simultaneously accumulated into temporary register. Consequently, we need to guess 4 bits at a time. In the template building phase, 10000 power traces with different keys and coefficients are captured from a reference device, based on which 35 interesting points are selected by the CPA peak method. Next, we collect two groups of power traces corresponding

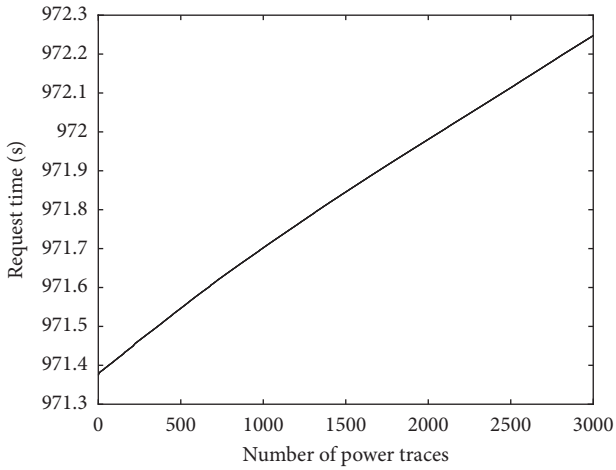


Figure 9: Time required for templates building.

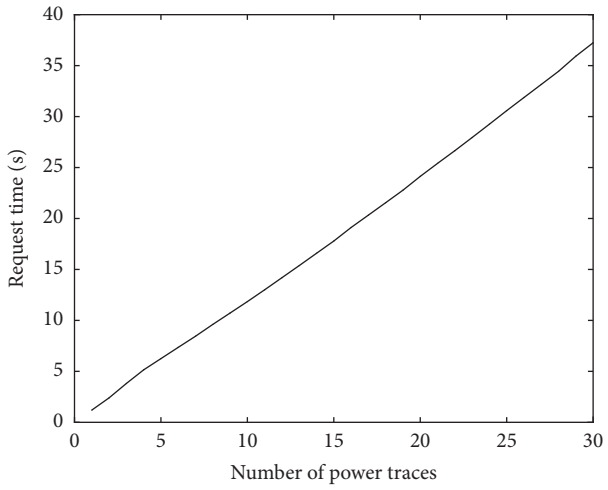


Figure 10: Time required for templates matching of serial QUAD of serial QUAD.

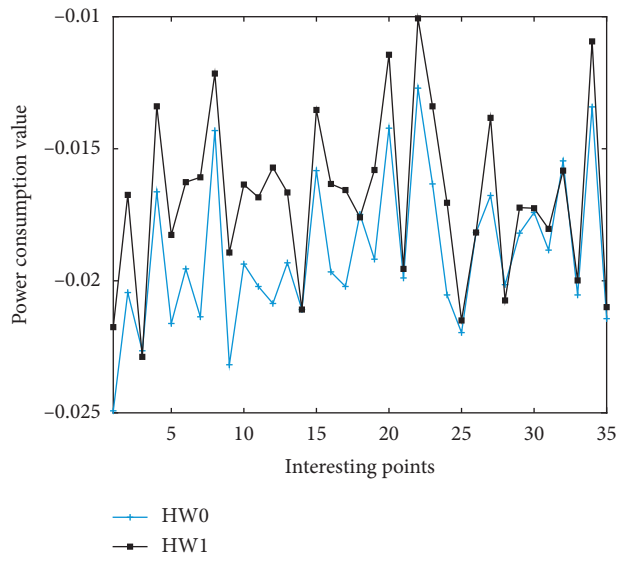


Figure 11: Templates built for parallel QUAD.

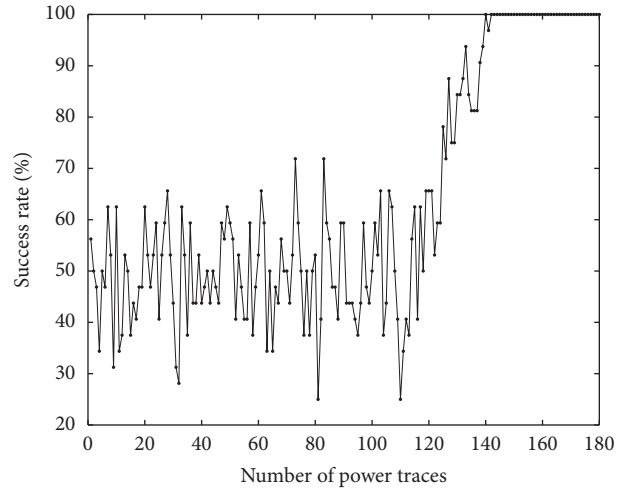


Figure 12: Success rates of our attack on parallel implementation.

to leakage values 0 and 1 in equation (7), and each group consists of 35 power traces. Finally, we build two templates, and the result is shown in Figure 11.

In the template matching phase, 320 power traces with the same key are captured from the target device. Template that leads to the highest probability indicates the correct key. The success rate of template-based LSQ attack on parallel implementation is shown in Figure 12. When the number of power traces approach 150, the success rate tends to 100%. Therefore, 150 power traces are sufficient for a successful attack in parallel implementation.

Figure 13 shows the time required for the template building with the number of power traces ranging from 1 to 10000. Figure 14 shows the time required for template matching with the number of power traces ranging from 1 to 180. As our successful attack on the parallel implementation of QUAD (2, 160, 160) requires less than 10000 power traces for templates building and 150 power traces for templates matching, the total time required for our successful attack is less than 1977 seconds, according to Figures 13 and 14.

In order to compare the success probability of the attacks, we performed our attack, template attack, and template-based DPA attacks dozens of times, respectively. We compare the typical results in Table 1, which shows that our proposed attack has the highest accuracy, greatly outperforming template attack, and template-based DPA attack.

5. Suggested Countermeasures

Side-channel countermeasures aim at reducing the data dependency between physical information and secret key. Usually, masking and hiding technologies are adopted. For multivariate cryptography, all monomials and polynomials can be computed in an arbitrary order. Therefore, the basic idea of countermeasures for multivariate cryptography is to randomly change the sequence of these operations.

A QUAD (q, n, r) has $(n + r) \times n(n + 1)/2$ monomials, which can be randomly computed in $((n + r) \times n(n + 1)/2)!$

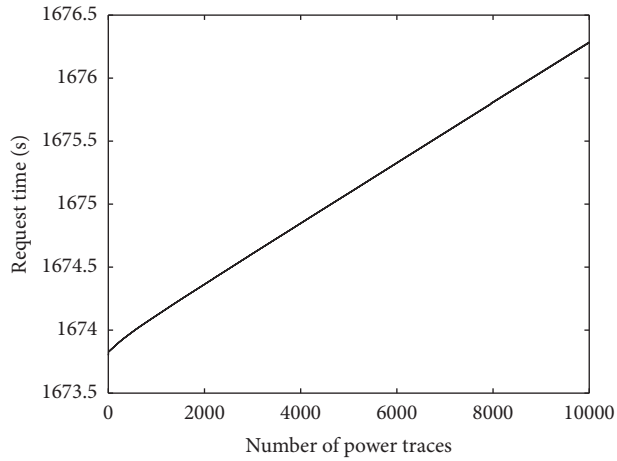


Figure 13: Time required for templates building.

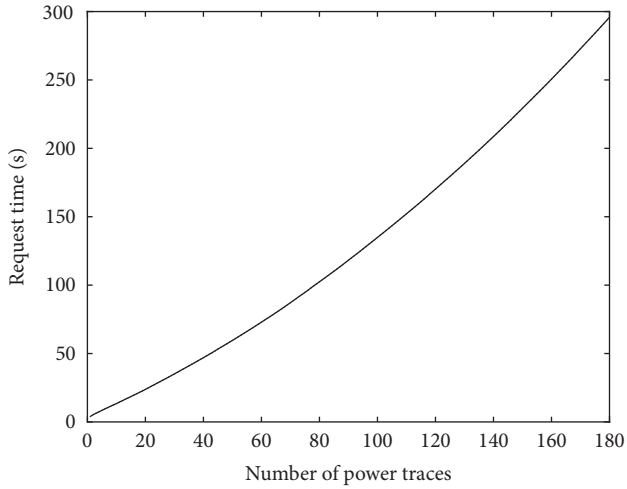


Figure 14: Time required for templates matching of parallel QUAD.

Table 1: Comparison of the success rates of different template attacks.

No. of experiments/success rate of attack	1	2	3	4	5	6	7	8	9	10
Template-based LSQ attack (%)	98.12	97.25	100	97.25	100	100	100	97.25	96.875	100
Template-based DPA attack (%)	71.88	65.63	71.88	68.75	81.25	75	81.25	78.13	68.75	75
Template attack (%)	61.75	71.88	53.125	65.63	75	68.75	61.75	65.63	63.75	59.375

orders. However, it is too expensive to implement such algorithm. We propose a low-cost shuffling countermeasure by partially changing the orders of monomials for each polynomial equation $Q_k(X)$. Starting with two randomly generated index i_s and j_s, $1 < i_s \le j_s \le n$, each polynomial is computed in the order as follows:

$$Q(x) = \sum_{j_s \le j \le n} \alpha_{i_s j} x_{i_s} x_j + \sum_{(i_s+1) \le i \le j \le n} \alpha_{ij} x_i x_j + \sum_{1 \le i \le (i_s-1), i \le j \le n} \alpha_{ij} x_i x_j + \sum_{i_s \le j \le (j_s-1)} \alpha_{i_s j} x_{i_s} x_j + \gamma. \tag{14}$$

A random index generator is required to generate such an order index, as shown in Figure 15, whose implementation requires only 556 GE.

For the parallel implementations, we proposed a low-cost hiding countermeasure by partially randomizing the initial value of rotated x to shuffle the computation orders of

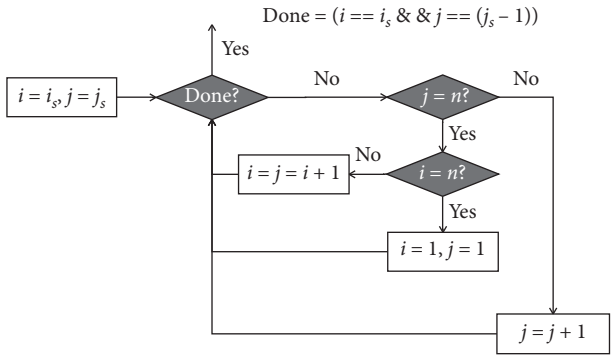


Figure 15: The random index generator of shuffle countermeasure.

monomials. Before each calculation of $Q_k(X)$, the initial value of rotated x is partially randomized with the random starting index i_s as follows:

$$\text{rotated } x = \left(x_{i_s}, x_{i_s+1}, \ldots, x_n, x_1, \ldots x_{i_s-1}\right). \tag{15}$$

6. Conclusions

Multivariate cryptosystems consist of a large number of monomials and polynomials, where registers are required to store monomial and polynomial values during the encryption. Therefore, a hamming distance (HD) model of the register will leak the secret of the implementation.

By applying the least-square technique to enable fuzzy matching of the templates, we propose a practical template-based least-square power analysis, where both the serial and parallel implementations of QUAD (2, 160, 160) can achieve a success rate close to 100%. The proposed two low-cost hiding countermeasures for serial and parallel implementations are also validated to be effective, where all monomials and polynomials can be computed in an arbitrary order to break the link between the power consumption and the secret key in multivariate cryptography. Our proposed attacks require only 30 and 150 power traces, respectively, to successfully reveal the secret key. Future work will focus on low-cost countermeasures of multivariate cryptography for IoT devices to resist side-channel attacks.

Data Availability

The mat data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61872096, 61672008, and 61772144), Innovation Team Project of the Education Department of Guangdong Province (no. 2017KCXTD021), Guangdong Provincial Key Laboratory of Intellectual Property and Big Data (Grant no. 2018B030322016), Key Laboratory of the Education Department of Guangdong Province (no. 2019KSYS009), and Guangdong Provincial Project of Science and Technology (no. 2016A010101030).

References

[1] T. Monz, D. Nigg, E. A. Martinez et al., "Realization of a scalable shor algorithm," *Science*, vol. 351, no. 6277, pp. 1068–1070, 2016.

[2] M. F. Ezerman, H. T. Lee, S. Ling, K. Nguyen, and H. Wang, "Provably secure group signature schemes from code-based assumptions," *IEEE Transactions on Information Theory*, p. 1, 2020.

[3] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, and R. Cammarota, "Post-quantum lattice-based cryptography implementations," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–41, 2019.

[4] T. Takagi, "Recent developments in post-quantum cryptography," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E101.A, no. 1, pp. 3–11, 2018.

[5] D. H. Duong, L. Van Luyen, and H. T. N. Tran, "Choosing subfields for LUOV and lifting fields for Rainbow," *IET Information Security*, vol. 14, no. 2, pp. 196–201, 2020.

[6] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," *Applied Cryptography and Network Security*, Springer, New York, NY, USA, pp. 164–175, 2005.

[7] J. Porras, J. Baena, and J. Ding, "ZHFE, a new multivariate public key encryption scheme," in *Proceedings of the International Workshop on Post-Quantum Cryptography*, pp. 229–245, Waterloo, ON, Canada, 2014.

[8] J. Wang, L.-M. Cheng, and T. Su, "Multivariate cryptography based on clipped hopfield neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 353–363, 2018.

[9] C. Berbain, H. Gilbert, and J. Patarin, "QUAD: a practical stream cipher with provable security," *Advances in Cryptology - EUROCRYPT 2006*, in *Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques*, pp. 109–128, St. Petersburg, Russia, 2006.

[10] C. Berbain, H. Gilbert, and J. Patarin, "QUAD: a multivariate stream cipher with provable security," *Journal of Symbolic Computation*, vol. 44, no. 12, pp. 1703–1723, 2009.

[11] M. Bardet, J.-C. Faugère, B. Salvy, and P.-J. Spaenlehauer, "On the complexity of solving quadratic boolean systems," *Journal of Complexity*, vol. 29, no. 1, pp. 53–75, 2013.

[12] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 964–975, 2018.

[13] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2017.

[14] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[15] S. Tanaka, C. Cheng, T. Yasuda, and K. Sakurai, "Parallelization of QUAD stream cipher using linear recurring sequences on graphics processing units," in *Proceedings of the 2014 Second International Symposium on Computing and Networking*, pp. 543–548, Shizuoka, Japan, 2014.

[16] G. Liao, Z. Gong, Z. Huang, and W. Qiu, "A generic optimization method of multivariate systems on graphic processing units," *Soft Computing*, vol. 22, no. 23, pp. 7857–7864, 2018.

[17] O. Billet, J. Etrog, and H. Gilbert, "Lightweight privacy preserving authentication for RFID using a stream cipher," *Fast Software Encryption*, in *Proceedings of the 17th International Conference on Fast Software Encryption*, pp. 55–74, Seoul, South Korea, 2010.

[18] D. Arditti, C. Berbain, O. Billet, and H. Gilbert, "Compact FPGA implementations of QUAD," in *Proceedings of the 2nd ACM symposium on Information, computer and communications security-ASIACCS '07*, pp. 347–349, Dallas, TX, USA, 2007.

[19] J. R. Hamlet and R. W. Brocato, "Throughput-optimized implementations of QUAD," *Journal of Cryptographic Engineering*, vol. 5, no. 4, pp. 245–254, 2015.

[20] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proceedings of the International workshop on Cryptographic Hardware and Embedded Systems (CHES'04)*, pp. 16–29, Cambridge, MA, USA, 2004.

[21] L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F.-X. Standaert, and N. Veyrat-Charvillon, "Mutual information analysis: a comprehensive study," *Journal of Cryptology*, vol. 24, no. 2, pp. 269–291, 2011.

[22] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'02)*, pp. 13–28, San Francisco Bay (Redwood City), CA, USA, 2002.

[23] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side—channel(s)," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems (CHES'02)*, pp. 13–28, San Francisco Bay (Redwood City), CA, USA, 2002.

[24] E. Özgen, L. Papachristodoulou, and L. Batina, "Template attacks using classification algorithms," in *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 242–247, McLean, VA, USA, 2016.

[25] M. O. Choudary and M. G. Kuhn, "Efficient, portable template Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 490–501, 2018.

[26] L. Lerman, R. Poussier, O. Markowitch, and F.-X. Standaert, "Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version," *Journal of Cryptographic Engineering*, vol. 8, no. 4, pp. 301–313, 2018.

[27] H. Zhang, "On the exact relationship between the success rate of template attack and different parameters," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 681–694, 2019.

[28] S. Picek, A. Heuser, A. Jovic, and L. Batina, "A systematic evaluation of profiling through focused feature selection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2802–2815, 2019.

[29] B. Hettwer, S. Gehrer, and T. Güneysu, "Applications of machine learning techniques in side-channel attacks: a survey," *Journal of Cryptographic Engineering*, vol. 10, no. 2, pp. 135–162, 2019.

[30] S. Hou, Y. Zhou, H. Liu, and N. Zhu, "Wavelet support vector machine algorithm in power analysis attacks," *Radioengineering*, vol. 26, no. 3, pp. 890–902, 2017.

[31] L. Malina, V. Zeman, J. Martinasek, and Z. Martinasek, "*K*-nearest neighbors algorithm in profiling power analysis attacks," *Radioengineering*, vol. 25, no. 2, pp. 365–382, 2016.

[32] P. Saravanan and P. Kalpana, "A novel approach to attack smartcards using machine learning method," *Journal of Scientific & Industrial Research*, vol. 76, no. 2, p. 99, 2017.

[33] L. Masure, C. Dumas, and E. Prouff, "A comprehensive study of deep learning for side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 348–375, 2020.

[34] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient CNN architectures in profiling attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 1–36, 2020.

[35] K. Okeya, T. Takagi, and C. Vuillaume, "On the importance of protecting in SFLASH against side channel attacks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 1, pp. 123–131, 2005.

[36] Y. Hashimoto, T. Takagi, and K. Sakurai, "General fault attacks on multivariate public key cryptosystems," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E96.A, no. 1, pp. 196–205, 2013.

[37] H. Yi and W. Li, "On the importance of checking multivariate public key cryptography for side-channel attacks: the case of enTTS scheme," *The Computer Journal*, vol. 60, no. 8, pp. 1197–1209, 2017.

[38] A. Park, K. Shim, N. Koo, and D. Han, "Side-channel attacks on post-quantum signature schemes based on multivariate quadratic equations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 3, pp. 500–523, 2018.

[39] J. Krämer and M. Loiero, "fault attacks on UOV and rainbow," *Constructive Side-Channel Analysis and Secure Design*, in *Proceedings of the International Workshop on Constructive Side-Channel Analysis and Secure Design*, pp. 193–214, Darmstadt, Germany, 2019.

[40] W. Li, F. Lu, and H. Zhao, "Power analysis attacks against QUAD," *IAENG International Journal of Computer Science*, vol. 46, no. 1, pp. 54–60, 2019.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 3838547, 16 pages
https://doi.org/10.1155/2020/3838547

Research Article

Efficient Coded-Block Delivery and Caching in Information-Centric Networking

Yan Liu[1], **Jun Cai**[1], **Huimin Zhao**[1], **Shunzheng Yu**[2], **JianLiang Ruan**[1] and **Hua Lu**[3]

[1]*Guangdong Polytechnic Normal University, Guangzhou, China*
[2]*School of Data and Computer Science, Sun Yat-San University, Guangzhou, China*
[3]*Network Technology Innovation Center of Guangdong Communication & Network Institute, Guangzhou, China*

Correspondence should be addressed to JianLiang Ruan; ruanjianliang@gpnu.edu.cn

Received 28 March 2020; Accepted 15 May 2020; Published 10 June 2020

Guest Editor: Jianbiao Zhang

Information-centric networking (ICN) provides request aggregation and caching strategies that can improve network performance by reducing content server loads and network traffic. Incorporating network coding into ICN can offer several benefits, but a consumer may receive the same coded block from multiple content routers since the coded block may be cached by any of the content routers on its forwarding path. In this paper, we introduce a request-specific coded-block scheme to avoid linear dependency of blocks that are utilizing in-network caching. Additionally, a non-cooperative coded caching and replacement strategy is designed to guarantee that the cached blocks can be reused. Our experimental results show that the proposed scheme has superior performance to conventional CCN and two network coding-based ICN schemes.

1. Introduction

Trends in recent years have shown that Internet users care more about *what* the content is rather than *where* the content is. Information-centric networking (ICN) [1] is a novel design for a future networking architecture that has been proposed as a promising alternative to the current Internet. In ICN, IP addresses are replaced by content names and content routers (CRs) are equipped with storage capabilities to cache the content passing through each router. Content is requested by Interest packets that are sent by the consumer. With in-network caching [2, 3], the content can be cached by multiple CRs, and any content router (CR) that contains the content that is being requested by the Interest can respond with a data packet, where both the Interest and the data are identified by the content name. Content-centric networking (CCN) has been shown to be a promising ICN architecture [4].

Network coding proposed by Ahlswede et al. [5] has been proven to be helpful in several different network scenarios, including peer-to-peer (P2P) [6], content distribution networks (CDNs) [7], and wireless networks [8, 9]. Recently, several studies have shown that network coding can also offer benefits to ICN [10–21], as network coding can be employed in ICN to effectively utilize multiple paths and reduce the complexity of the cache coordination. However, due to the ICN caching strategy, the same coded block may be cached by multiple CRs on its forwarding path and provided to the same consumer at a later time in response to their multicast requests [22].

In this case, the consumer will not be able to recover the content from the received coded blocks. Several solutions have been proposed to guarantee that all the coded blocks that are provided to the consumer are linearly independent of each other. In some centralized schemes [11, 15], central routers are used to ensure that content caching and routing strategies can provide independent blocks. In some distributed schemes [20, 21], information on the coded blocks which have already been received by the consumer must be carried by the Interest to retrieve linearly independent blocks. The CR can decide whether to respond to the Interest according to the information carried by the Interest. Therefore, several round trips will be required to obtain sufficient linearly independent coded blocks. In our previous

work [23], the CRs only cached the original received blocks to guarantee that all the coded blocks provided to consumers were linearly independent. Any coded blocks generated and transferred were wasted.

To increase the caching efficiency and reduce the cost of computation and communication induced by centralized schemes, we propose a request-specific coded-block (RSCB) scheme to reduce the transmission volume and download delay and ensure that only a single round trip is required for the consumer to retrieve sufficient linearly independent blocks. A non-cooperative coded caching and replacing strategy is then proposed to guarantee that any two coded blocks that are cached in a network will be linearly independent. It is assumed that chunk-based routing and traffic control schemes are in place. The contributions of this paper are as follows:

(i) We propose a special content delivery strategy to retrieve blocks from multiple CRs simultaneously. Each CR on the forwarding path will aggregate Interests received from multiple consumers for chunks of the same content, to eliminate duplicates. Interests received by a CR will be separated again and forwarded in different directions. A mechanism is proposed for the aggregation and separation of Interests for the chunks of content to guarantee that the minimum number of coded blocks will be requested and will be linearly independent.

(ii) An on-path non-cooperative coded caching mechanism is designed to guarantee that the cached blocks can be reused. Blocks received by a CR can be encoded and cached depending on pending Interests and the proposed caching strategy.

(iii) In our model, only chunks (i.e., original blocks) and coded-from-original blocks can be cached. One coded-from-original block can satisfy multiple Interests sent by different consumers requesting a set of its component chunks. A chunk-level coding-instead-of-evicting cache replacement scheme is designed to effectively increase the caching efficiency and optimize cache capacity.

(iv) Our strategy is evaluated by comparison with conventional CCN and two network coding-based ICN strategies. Our experimental results demonstrate that the proposed strategy achieves the highest performance in terms of parameters such as average download time, server hit reduction rate, and cache hit rate.

2. Related Works

Network coding techniques have received much attention in a variety of network scenarios including P2P networks [6], CDNs [7], and wireless networks [9]. Recently, several works have been proposed that apply network coding in ICN. There are two categories of solutions that can be used to ensure consumers are provided with sufficient linearly independent coded blocks: centralized strategies and distributed strategies.

Wang et al. [24] proposed a novel SDN-based framework to implement content caching and routing in ICN with linear network coding. The SDN controllers determine how to cache and route based on the information collected by the CR. Thus, a near-optimal caching and routing strategy can be obtained. Sadjadpour [11] proposed an architecture based on index coding for ICN, which groups the nodes into several clusters. The central router of each cluster maintains information on which content is cached by each node. Coded blocks generated by the central router are used to satisfy Interests for different content sent by different nodes. However, this strategy does not reduce traffic the first time content is requested. Llorca et al. [14] presented a multicast scheme based on network coding to achieve maximum network efficiency. However, the proposal does not mention a solution to deploy the proposed strategy in ICN. Talebifard et al. [15] proposed a method based on network coding that reduces the costs of coding and decoding by breaking the network into several clusters, with network coding only performed by selected nodes or clusters.

As well as centralized strategies based on network coding, some works have obtained enough linearly independent coded blocks by sending Interests repeatedly. Zhang and Xu [21] proposed two checking strategies to guarantee that the consumer will receive sufficient linearly independent coded blocks, which were called precise matching and RB matching. In precise matching, each Interest carries the global coefficients X of the coded blocks that have already been received by the consumer. Each CR performs Gaussian elimination to check linear dependencies. Precise matching is an efficient approach to guarantee that all blocks received by consumers will be linearly independent. However, it has very high communication and computation overheads. Therefore, RB matching was proposed as a more lightweight approach, where the Interest only carries the rank of the global coefficients X of the coded blocks already received by the consumer. If the number of coded blocks cached by the CR is larger than the rank of the global coefficients X, the CR can respond to the Interest with a coded block. The larger the value of $|X|$ is, the more difficult it is to serve the Interest. Wu et al. [16] proposed a network coding and random forwarding-based caching strategy, CodingCache, to enhance the caching efficiency. To guarantee all the blocks provided to consumer are linearly independent, each Interest carries the global coefficients of the coded blocks already received to retrieve the next block, similar to precise matching. Therefore, N rounds will be required in order to retrieve N blocks. Nguyen et al. [20] proposed a lightweight caching and Interest aggregation strategy to ensure that all the coded blocks received by the consumer are independent. Like RB matching, the rank of the global coefficients of the coded blocks already received by the consumer is carried by the Interest packet. Saltarin et al. [19] proposed a protocol named NetCodCCN to permit Interest aggregation and pipelining. Each node responds to an Interest once it has received enough coded blocks to recover the content or $|X|$ is larger than the number of coded blocks already sent out over face i previously, where $|X|$ is the rank of the global coefficients X of the coded blocks cached in ContentStore.

However, NetCodCCN has a weakness also shared by RB matching in that it may provide false negative decisions, i.e., a node may falsely decide it cannot provide an innovative coded block for the consumer while actually the block is available. Montpetit et al. [17] proposed an architecture based on network coding, NC3N, where each Interest retrieves one coded block. However, there method does not include a strategy to ensure all the received blocks are independent. Liu et al. [18] proposed an ICN-NC method to guarantee that all the blocks received are provided by different CRs to increase the probability of obtaining linearly independent blocks. Each Interest packet contains a record of the Interest exploration range of the previous round. Only CRs within a new exploration area are permitted to respond to these Interests. Several rounds are required to retrieve enough independent coded blocks, and the Interest may retrieve linearly dependent coded blocks. The authors in [25] proposed a framework based on network coding for cache management in ICNs. Saltarin et al. [26] proposed a distributed caching strategy for ICNs enabled for network coding, which gives CRs the responsibility of estimating the popularity of contents and ensuring that the most popular content is cached near the network edge.

Most of the existing schemes require several round trips to obtain sufficient linearly independent coded blocks to recover the content. In this paper, we propose a novel content delivery strategy to ensure enough blocks can be retrieved within a single round. An on-path non-cooperative caching and replacing strategy based on network coding is proposed to guarantee that all blocks received by consumers are linearly independent. Moreover, in our scheme, coded blocks are generated only if the traffic can be saved instead of generated at the server and all CRs on the forwarding path in order to reduce the cost of coding and decoding.

3. Method of Interest Aggregation and Separation

In ICN, chunk-based delivery strategies route chunks separately. Chunks may meet on an intermediate node during their forwarding paths to several consumers. Motivated by this, we propose a special request-specific coded-block (RSCB) scheme to encode chunks that meet during transport in order to reduce traffic.

3.1. Overview of RSNC. The definitions given in our previous study (referred to as RSNC) will be followed here. Each Interest (S, N) requests a specific set of chunks, where $S = \{1, \ldots, N\}$ is the set of chunk indices and N is number of independent coded blocks required to recover the content. Since chunks may be cached by different CRs, each CR can aggregate, separate, and forward Interests. If Interest 1 requests a set of chunks S_1 and Interest 2 requests a set of chunks S_2, then (S, n) satisfies both Interests, where $S = S_1 \cup S_2$ is the set of chunks used to generate n linearly independent coded blocks, n is the number of coded blocks to be sent by the upstream CR, and $n = \max\{|S_1|, |S_2|\}$. When $n < |S|$, the traffic required to deliver chunks from upstream will be reduced.

Since an Interest sent by a consumer for multiple chunks will be copied and forwarded along a multicast tree, requests for different chunks sent from the same consumer will not meet again in a CR on the multicast tree. Therefore, the Interest aggregation operation "$\oplus$" is defined to combine two Interests originating from at least two different consumers, i.e.,

$$\begin{aligned} &(S_1, n_1) \oplus (S_2, n_2) \overset{\text{def}}{\Rightarrow} (S, n), \\ &\text{where } S = S_1 \cup S_2, \\ &n = \max\{n_1, n_2\}. \end{aligned} \tag{1}$$

Similarly, a separation operation "Div" is defined to split an Interest into several sub-Interests:

$$\begin{aligned} &\text{Div}(S, n) \overset{\text{def}}{\Rightarrow} \{(S_3, n_3), (S_4, n_4)\}, \\ &\text{where } (S_3, n_3) \oplus (S_4, n_4) = (S, n), \\ &S_3 \cap S_4 = \varnothing, \\ &S = S_3 \cup S_4, \\ &n_3 = \min\{|S_3|, n\}, \\ &n_4 = \min\{|S_4|, n\}. \end{aligned} \tag{2}$$

3.2. Interest Aggregation and Separation in RSCB. In contrast with RSNC [23], RSCB includes information on (S_1, n_1) and (S_2, n_2) in the aggregated Interest (S, n) which guarantees that linearly independent coded blocks are provided to consumers and minimize the number of coded blocks transported in the network. To reduce the size of the Interest, the sub-Interest information is presented as a binary number, $b(S_i)$. For example, if Interest $(S = \{1, 2, 3, 4\}, b(S), n = 2)$ is an aggregated Interest of Interest 1 $(S_1 = \{1, 3\}, n_1 = 2)$ and Interest 2 $(S_2 = \{2, 4\}, n_2 = 2)$, the binary information of Interest 1 is $b(S_1) = 1010$, and the binary information of Interest 2 is $b(S_2) = 0101$, and thus $b(S) = b(S_1) \cup b(S_2) = \{1010, 0101\}$. Therefore, an Interest can be expressed as $I(p, [S, b(S)], n)$, where p is the name of the requested content, S is a set of chunks that match the name of the content p, $b(S)$ is a set of binary numbers representing the sub-Interests (each sub-Interest is a subset of S), and n is the number of linearly independent coded blocks being requested. Therefore, any n linearly independent coded blocks that contain all chunks specified by S will satisfy the sub-Interests specified in $b(S)$.

Equation (1) can thus be further modified as

$$\begin{aligned} &([S_1, b(S_1)], s_1) \oplus ([S_2, b(S_2)], s_2) \overset{\text{def}}{=} ([S, b(S)], s), \\ &\text{where } S = S_1 \cup S_2, \\ &s = \max\{s_1, s_2\}, \\ &b(S) \subseteq b(S_1) \cup b(S_2), \end{aligned} \tag{3}$$

where s is the minimum number of linearly coded blocks satisfying both Interests $([S_1, b(S_1)], s_1)$ and $([S_2, b(S_2)], s_2)$.

It should be noted that the binary number $b(S_i)$ is used to represent the subset information, which is required to guarantee that the requested number of coded blocks is minimized. When $s = 1$ or $s = |S|$, this subset information is not necessary, as shown in Figure 1(a). Moreover, if $S_i \subseteq S_j$, the information on S_i, i.e., $b(S_i)$, will be deleted from $b(S)$.

For instance, Figure 1(a) shows that CR_1 receives two Interests for content C_p from different interfaces, $I(p, S_1 = \{2, 4\}, 2)$ and $I(p, S_2 = \{1, 3\}, 2)$. Before these two Interests are forwarded, CR_1 aggregates the two requests into a single Interest $I(p, [S = \{1, 2, 3, 4\}, b(S) = \{0101, 1010\}], 2]$ using equation (3). $b(S)$ can then be used to reconstruct the subsets $\{2, 4\}$ and $\{1, 3\}$.

Similarly, the separation operation used to split an Interest $([S, b(S)], s)$ into several sub-Interests is modified, which is used to distribute the sub-Interests out over several interfaces of the CR towards different content sources:

$$\begin{aligned}
&\mathrm{Div}([S, b(S)], s) \stackrel{\mathrm{def}}{=} \{([S_1, b(S_1)], s_1), ([S_2, b(S_2)], s_2)\},\\
&\text{where } S_1 \cap S_2 = \text{null},\\
&S = S_1 \cup S_2;\\
&s_1 = \min\{|S_1|, s\},\\
&s_2 = \min\{|S_2|, s\}.
\end{aligned} \tag{4}$$

If an Interest $([S, b(S)], s)$ is formed by merging multiple Interests, the subsets (S_i, s_i) should be reconstructed based on $b(S)$, and these subsets should then be separated into sub-subsets using equation (2). Then, new Interests, i.e., $([S_1, b(S_1)], s_1)$ and $([S_2, b(S_2)], s_2)$ in equation (4), are generated by aggregating the sub-subsets using equation (3). This procedure is described in Algorithm 1. The complexity of Algorithm 1 is $O(n)$, where n is the number of subsets. According to Algorithm 1, $I(p, [\{1, 2, 3, 4\}, b(S)], 2)$ can be reconstructed into two subsets $(\{2, 4\}, 2)$ and $(\{1, 3\}, 2)$ for $b(S) = \{0101, 1010\}$. These subsets can be further divided into sub-subsets: $(\{2\}, 1)$, $(\{4\}, 1)$, $(\{1\}, 1)$, $(\{3\}, 1)$ using equation (2), and then aggregated into Interest $I(p, \{1, 2\}, 1)$ and Interest $I(p, \{3, 4\}, 1)$ according to equation (3). If the original blocks ob_1 and ob_2 are located in the same direction and the original blocks ob_3 and ob_4 are located in another direction, the new Interests can be sent from two interfaces in two different directions, as shown in Figure 1(a). In this case, only two blocks will transmitted which is in contrast with RSNC [23], which requires four blocks to be transmitted.

If Interest 2 (S_2, n_2) arrives after Interest 1 (S_1, n_1) has already been sent upstream, then the aggregated pending Interest will be $(S, n) \stackrel{\mathrm{def}}{=} (S_1, n_1) \oplus (S_2, n_2)$. Since (S_2, n_2) may contain some chunks that have also been requested by Interest (S_1, n_1), these chunks should be removed from Interest 2. Therefore, we define an operation to determine incremental Interest based on the separation operation:

$$\begin{aligned}
&(\Delta S_2, \Delta n_2) \stackrel{\mathrm{def}}{=} (S_2, n_2) \setminus (S_1, n_1),\\
&\text{where } \Delta S_2 = S_2 \setminus \Delta S_1, \Delta n_2 = \min\{|\Delta S_2|, n_2\},\\
&\Delta S_1 \subseteq S_1 \cap S_2,\\
&\Delta n_1 = \min\{|\Delta S_1|, n_1\}.
\end{aligned} \tag{5}$$

Since Interest 1 will return at most n_1 coded blocks, $\Delta n_1 \le n_1$ is required. If $|S_1 \cap S_2| > n_1$ and $n_2 > n_1$, we let $\Delta S_1 \subset S_1 \cap S_2$ and $\Delta n_1 = |\Delta S_1| = n_1$. Similarly, if a CR has cached a subset (W, w) of blocks, only the remaining (S', n') blocks need to be requested from the upstream CRs, where $(S', n') = (S, n) \setminus (W, w)$. In RSCB, the coded blocks generated by the original blocks (referred to as *coded-from-original block*) are cached by the first node *en route* to consumers. The coded-from-original blocks are used as the original blocks to satisfy future Interests. For example, in Figure 1(b), the coded-from-original block, $ocb_1 = \alpha_{11} ob_1 + \alpha_{12} ob_2$, can be presented as $(W = \{1, 2\}, w = 1)$. When Interest $(S = \{1, 2\}, n = 2)$ is received by CR_2, the incremental Interest $(S' = 2, n = 1)$ is determined and sent to the next node.

The benefits of RSCB are illustrated in Figures 1 and 2. In Figure 1(a), two consumers connected to router CR_5 and CR_6 have requested content, which contains four original blocks, ob_1, ob_2, ob_3, and ob_4 at the same time. Each original block has a size of one unit, each CR has a two-unit cache capacity, and each link has a one-unit transmission cost. Figure 1 shows the communication and caching in RSCB. For RSCB, CR_5 and CR_6 receive two coded-from-original blocks generated by CR_3 and CR_4, respectively. Coded-from-original blocks ocb_1 and ocb_2 are received from CR_1, where

$$\begin{aligned}
ocb_1 &= \alpha_{11} ob_1 + \alpha_{12} ob_2,\\
ocb_2 &= \alpha_{21} ob_3 + \alpha_{22} ob_4.
\end{aligned} \tag{6}$$

The total transmission cost is eight units. Figure 2 shows the conventional ICN communication. CR_2 receives the four original blocks (ob_1, ob_2, ob_3, ob_4) from CR_3 and CR_4 and then forwards these original blocks to CR_1. CR_1 responds to Interest $I(p: \{2, 4\}, 2)$ with two original blocks, ob_2 and ob_4, and Interest $I(p, \{1, 3\}, 2)$ with two further original blocks, ob_1 and ob_3. The total transmission cost is 12 units. Therefore, our proposed solution saves 33% of the transmission cost compared with conventional ICN and 20% more than our previous work [23].

3.3. Caching in RSCB. In RSCB, the original/coded-from-original blocks are cached by CRs to respond to future Interests. In order to provide consumers with sufficient linearly independent blocks in a single round, none of coded blocks that were encoded by coded blocks are cached in the network. The coded-from-original block only can be cached by a single CR, which is the immediate downstream neighbor of the CR that generated the block. Thus, the two coded-from-original blocks, ocb_1 and ocb_2, will be cached only by CR_2 (Figure 1(b). The coded-from-original block ocb_1 can be used as the original block ob_1 or ob_2 to respond to future Interests. When cache replacement happens, CR encodes several original blocks into one coded-from-original block to release the caching space. This ensures that all information contained in the original blocks is retained in the CR.

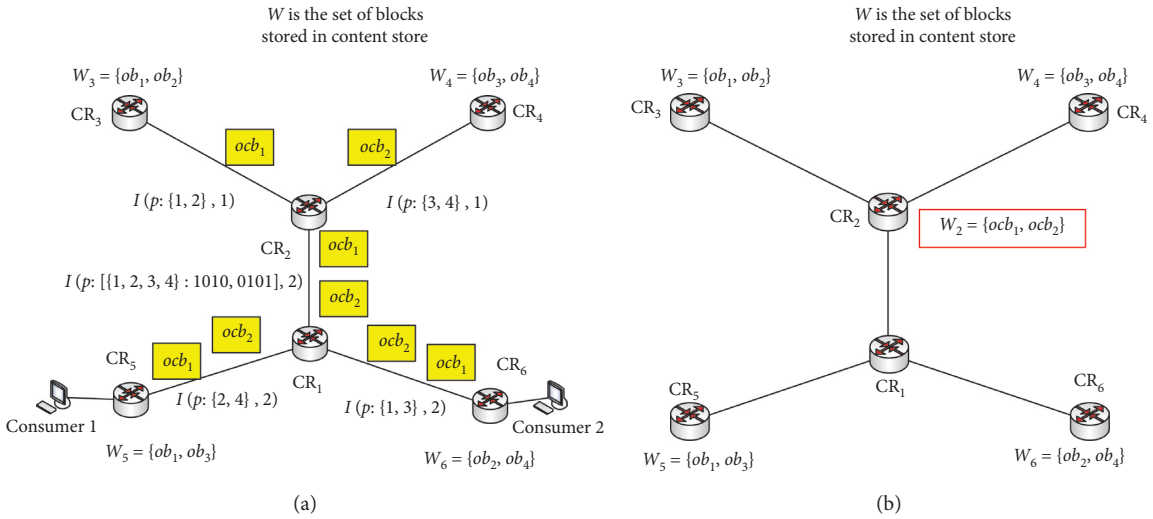


Figure 1: An example of RSCB. (a) Communication in RSCB. (b) Caching in RSCB.

Inputs

(1) Input Interest: $I(p, [S, b(S)], s)$

Steps

(1) Reconstruct subsets $[S_1, s_1]$,. . .and $[S_m, s_m]$ according to $b(S) = \{b(S_1), \ldots, b(S_m)\}$, where $s_i = |S_i|$, for $i = 1, \ldots, m$;
(2) **if** $S \neq \cup_i S_i$, $i = 1, \ldots, m$ **then**
(3) Determine subset $[S_{m+1}, s_{m+1}]$, where $S_{m+1} = S \backslash \cup_i S_i$, $s_{m+1} = 1$;
(4) **end if**
(5) **for** each subset $[S_i, s_i]$ **do**
(6) Use equation (2) to divide subset $[S_i, s_i]$ into several sub-subsets $[T_{i1}, t_{i1}], \ldots, [T_{ik}, t_{ik}]$ based on the forwarding interface of each chunk name (each and every sub-subset corresponds to an interface), where $t_{ij} = \min\{|T_{ij}|, s_i\}$, for $j = 1, \ldots, k$;
(7) **end for**
(8) Use equation (3) to generate new Interests by aggregating the sub-subsets $[T_1, t_1], \ldots, [T_n, t_n]$ according to the forwarding interfaces.

Algorithm 1: Separation.

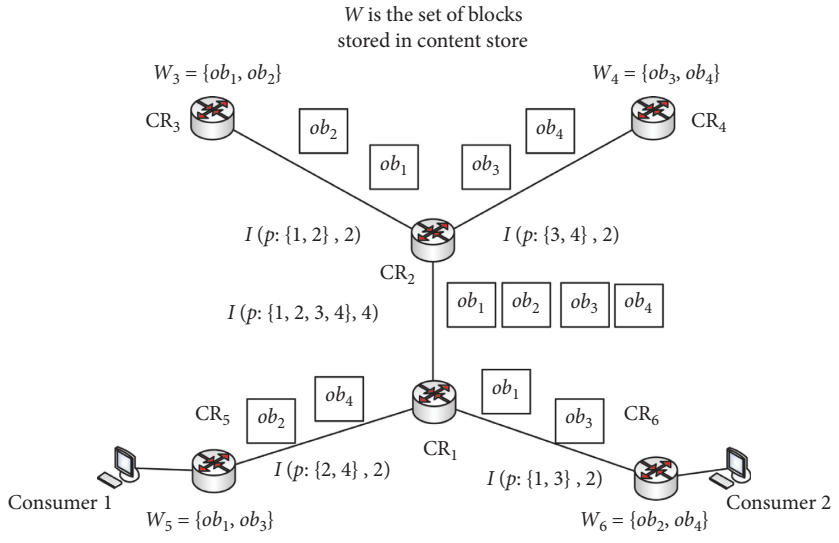


Figure 2: Communication in ICN.

4. Architecture

CCN architecture is the most popular architecture of ICN, and we have selected this architecture to implement RSCB. To enable network coding in CCN, some changes are required.

4.1. Content Publishing and Requesting. In CCN, content is split into several smaller-sized chunks, with each chunk identified by a unique hierarchical name. In RSCB, content is firstly divided into h generations and each generation is then divided into N chunks, i.e., *original blocks*. We denote content as $C_p = \{\{c_{p,1,1}, \ldots, c_{p,1,N}\}, \ldots, \{c_{p,h,1}, \ldots, c_{p,h,N}\}\}$, where p is the name of the content, N is a design parameter, and h can be calculated based on the content size and N. The content name p should contain information on h and N. For instance, the name of chunk $c_{p,2,1}$ is */sysu.edu.cn/largefile/h/N/2/1*, where */sysu.edu.cn/largefile/h/N* is the content name, p, 2 is the generation ID, and 1 is the chunk ID (referred to as the original block index).

A consumer will generate a set of Interests, $\{I_{p,1}, \ldots, I_{p,h}\}$, for each generation in order to request content C_p. Each Interest $I_{p,i}$ requests a set of original blocks, $\{c_{p,i,1}, \ldots, c_{p,i,N}\}$, where i is the number of generations. The number of Interests sent by the consumer depends on the flow control schemes which have been placed in the network. The consumer can send Interests either sequentially or simultaneously.

Since the forwarding paths of requests for different chunks generated by the same consumer will form a multicast tree, these requests will not meet in any intermediate CR on the multicast tree. Interests are responded to with chunks or coded blocks which are linear combinations of chunks that have been specified by the Interest. Random linear network coding (RLNC) is used to generate the coded blocks within each generation. For convenience, in the remainder of this paper, we will not explicitly state which generation each chunk belongs to. Our model makes the assumption that a chunk-based routing and flow control scheme is in place.

4.2. Interest and Data. All communications are driven by consumers in CCN. Consumers can receive chunks of content from multiple sources, which may include the content provider and CRs. A consumer interested in C_p will send a set of requests $I_p = \{i_{p,1}, \ldots, i_{p,N}\}$ with one request for each chunk. Before these requests are forwarded, the CR determines the forwarding interface of each chunk using the forwarding information base (FIB). Requests that have the same forwarding interface will be aggregated into a single Interest. Each Interest can contain multiple requests for a set of chunks, S_i, where S_i is the set of chunk names.

There are two types of CCN packets, Interests and data. In our model, the network coding information is appended to the selector field of the Interest packet and includes the set of chunk names S, the sub-Interests $b(S)$, and the number of required blocks n. The coefficient of the coded blocks, the caching flag Fq, is contained within the signed info field of the data packet. The data field of the data packet contains the original/coded block(s).

The Interest and data packets used in our model are formulated using the following method:

(i) $I(p, [S, B], m)$ defines an Interest for content C_p, where p is the content name, $S = (S_1 \cup S_2 \cup \ldots \cup S_n)$ is the set of chunk names, and $m = \max\{|S_1|, |S_2|, \ldots, |S_n|\} \leq |S|$ is the number of required blocks. $B = \{b(S_1), b(S_2), \ldots, b(S_n)\}$ is used to represent the subsets $S_1, S_2, \ldots, S_n$ and $S_i \not\subseteq S_j$, if $i \neq j$. For convenience, we express the Interest as $I(p, S, m)$.

(ii) $D(p, S, block, Fq)$ defines a data packet for content C_p containing *block*, which is either a chunk or a coded block (a linear combination of several chunks specified by set S). Fq is a caching flag which indicates whether the *block* is cacheable or not. If the *block* is a coded block, we obtain $S_D \supseteq S_I$, where S_D is the set of chunk names carried by the data and S_I is carried by the Interest.

4.3. Forwarding Module. The forwarding module of RSCB contains three components: the ContentStore (CS), the pending interest table (PIT), and the forwarding information base (FIB). The blocks received by the CRs are cached by the CS module. A CS entry can be formulated by $\mathrm{CS}(p, W, w)$, which is defined as follows:

(i) p: content name.

(ii) Index (W, w): W is a set of chunk names and w is the number of cached blocks. Since both the original and the coded-from-original blocks can be cached, we obtain $|W| \geq w$.

(iii) Data: the original or coded-from-original blocks.

In contrast with CCN, each CR interface in RSCB maintains a PIT. The PIT can have two types of entry, PIT-OUT and PIT-IN, which record information on Interests already sent or received to the interface, respectively. PIT-OUT and PIT-IN can be defined as follows:

(i) $\mathrm{PIT}_{\mathrm{out}-i}([p, S, s], facelist)$: this is a PIT-OUT entry that indicates that (p, S, s) is an aggregated Interest generated by aggregating several Interests received from interface(s) of $facelist$. The aggregated Interest has already been sent out over the interface i but a response has not been received from the upstream CR.

(ii) $\mathrm{PIT}_{\mathrm{in}-i}(p, S, s)$: this is a PIT-IN entry that indicates that (p, S, s) is an aggregated Interest generated by aggregating several Interests received from interface i. A response has not yet been sent over interface i.

The FIB is the same as for CCN. When an Interest is received by a CR, its CS is first consulted, followed by PIT and finally FIB. Data packets will be sent back to consumers using the same path that was created by the Interest, but in the opposite direction.

5. Communication Scheme

5.1. Forwarding Interest. When a CR receives an Interest $I(p, X, x)$ over interface f, the first step is to check its CS. If the CS contains all chunks or the coded-from-original blocks containing all the information in the Interested set X, the CR will respond to Interest $I(p, X, x)$ directly, as described in Algorithm 2. The complexity of Algorithm 2 is $O(n)$, where n is the number of coded blocks. If $|X| = x$, the CR responds to Interest $I(p, X, x)$ with the x cached blocks (chunks or coded-from-original blocks) without coding; otherwise, the CR will respond with the x coded blocks generated from the blocks cached in the CS, which contain the chunk information specified by set X. The caching flag, Fq, will be turned on, i.e., $Fq = 1$ if the block used to respond to the Interest is an original block or a coded-from-original block encoded by that CR; otherwise, $Fq = 0$. In RSCB, the CR performs network coding only if it will save on the transmission costs. For instance, as shown in Figure 1(a), CR responds to Interest $I(p, [\{1, 2, 3, 4\}: 1010, 0101], 2)$ with two coded-from-original blocks, ocb_1 and ocb_2, which were received from CR_3 and CR_4, respectively, without further coding. In this case, there is a saving on the cost of coding.

If the Interest cannot be satisfied by the CR, PIT-IN of the arrival interface f will be checked. If there is a matched entry $\mathrm{PIT}_{\mathrm{in}-f}(p, Z, z)$, CR will aggregate the PIT-IN entry and the Interest using equation (3) and will then update the PIT-IN entry $\mathrm{PIT}_{\mathrm{in}-f}(p, Z\prime, z\prime)$, where $Z\prime = X \cup Z$, $z\prime = \max\{x, z\}$. Therefore, the incremental Interest $(p, \Delta X, \Delta x) = (p, X, x) \backslash (p, W, w)$ will be determined.

The CR will split the incremental requests for $(p, \Delta X, \Delta x)$ into several Interests using Algorithm 1. If one of the Interests, e.g., (p, X_j, x_j), needs to be transmitted over the interface j, PIT-OUT of interface j will be obtained. If there is a matching PIT-OUT entry $\mathrm{PIT}_{\mathrm{out}-j}(p, V, v)$, a new incremental Interest for $(p, \Delta X_j, \Delta x_j) = (p, X_j, x_j) \backslash (p, V, v)$ will be generated using equation (5) and transmitted over interface j if $\Delta X_j \neq \mathrm{null}$. The PIT-OUT entry will then be updated to $\mathrm{PIT}_{\mathrm{out}-j}(p, V\prime, v\prime, facelist)$, where $V\prime = V \cup X_j$ and $v\prime = \max\{v, x_j\}$, and interface f will be added to $facelist$. Algorithm 3 describes the process used to forward an Interest. The complexity of Algorithm 3 is $O(n)$, where n is the number of sub-Interests.

5.2. Forwarding Data. When data packet $D(p, Y, block, Fq)$ is received by a CR over interface f, the PIT-OUT of interface f will be checked in the tables. If there is no PIT-OUT match, the data $D(p, Y, block, Fq)$ will be discarded directly since the CR has not requested the block; otherwise, the caching flag will be checked and the matching PIT-OUT entry will be updated according to Algorithm 4. If $Fq = 1$, the block carried by data will be cached into CS; otherwise, it will be temporarily cached into CACHE. The CR will then check whether more chunks can be obtained by decoding the blocks cached in CS and CACHE. If the CR has already received enough blocks of content p to recover the content, the chunks decoded from the received blocks will be cached into CS and all of the blocks of content p that were cached in CACHE will be deleted. In this case, CR can satisfy any Interest of content p.

If interface i is included in the $facelist$ of the matching PIT-OUT entry, the corresponding PIT-IN entry is $\mathrm{PIT}_{\mathrm{in}-i}(p, Z, z)$. If $Y \cap Z \neq \emptyset$, CR checks whether the $\mathrm{PIT}_{\mathrm{in}-i}(p, Z, z)$ can be satisfied using blocks cached in CS and CACHE. If it can be satisfied, CR will generate z linearly independent combinations of the blocks specified by the set Z and will send z data packets over interface i. Each data packet carries a coded block and $\mathrm{PIT}_{\mathrm{in}-i}(p, Z, z)$ is then deleted. Once the PIT-IN entries of all interfaces in $facelist$ of the matched PIT-OUT entry are satisfied, the coded blocks of content p that are cached in CACHE are deleted, as described by Algorithm 4. The complexity of Algorithm 4 is $O(nm)$, since the complexity of Algorithm 2 is $O(m)$, where n is the number of interfaces in the $facelist$ of the matched PIT-OUT and m is the number of coded blocks.

The network will try to deliver chunks without introducing extra traffic in order to increase the independence of the blocks cached in CRs. When the CR receives a data packet $D(p, Y, block, Fq)$ carrying a chunk (i.e., $|Y| = 1$), if $z = |Z|$, the data packet $D(p, Y, block, Fq)$ will be sent out over interface i without further processing or waiting for other data packets. The CR will then update $\mathrm{PIT}_{\mathrm{in}-i}(p, Z, z)$ to $\mathrm{PIT}_{\mathrm{in}-i}(p, Z\prime, z\prime)$, where $Z\prime = Z \backslash Y$, $z\prime = z - 1$. If $z = 0$, the CR will delete the PIT-IN entry. In this case, the time to download chunk Y will be reduced and the cost of coding and decoding is saved without introducing additional traffic.

5.3. Cache Policy. Network coding-enabled ICN (NC-ICN) will divide the content into n original blocks. For traditional NC-ICN, each coded block is a linear combination of the n original blocks. The n coded blocks will be cached by n' $(n' \geq n)$ CRs along their forwarding paths to a group of consumers. Figure 3(a) shows that for traditional NC-ICN, network $N1$ will provide m coded blocks, $cb_1, \ldots, cb_m$, to consumers in group $G1$ and network $N2$ will provide the remaining $(n - m)$ coded blocks. During the process of responding to consumers in group $G1$, $m\prime$ $(m\prime \geq m)$ coded blocks generated by $cb_1, \ldots, cb_m$ will be cached by multiple CRs in network $N1$, while n' $(n' \geq (n - m))$ coded blocks will be cached by multiple CRs in network $N2$. At a later time, when the consumers in $G2$ multicast their Interests for n coded blocks of content p, these Interests will be received first by CRs in network $N1$. Each CR will respond to the Interest with its cached coded blocks independently, and thus t $(t > m)$ coded blocks cached in network $N1$ may be provided to consumers in $G2$. However, the maximum number of independent coded blocks that a consumer can receive from network $N1$ is m. In this case, at least $(t - m)$ blocks are not beneficial to the consumer and are a waste of resources. Therefore, the conventional in-network caching strategy is not suitable for NC-ICN.

To address this issue, we propose to introduce a simple cache mechanism to guarantee that the blocks provided to consumers will be independent. In RSCB, only original blocks and coded-from-original blocks will be cached by CRs. None of coded blocks that were encoded by other

```
Input: I(p, X, x) ← Interest arriving on interface f,
CS_p(W, w)←W is the set of blocks specified by I(p, X, x) and w is the number of blocks;
(1) if |X| = x or x = w then
(2) for each block cs_i in CS_p do
(3) if cs_i is an original block then
(4) Fq = 1;
(5) else
(6) Fq = 0;
(7) end if
(8) Create a data packet D(p, Y_i, cs_i, Fq) and send over interface f;
(9) end for
(10) else
(11) Generate x coded blocks, ob_1, ..., and ob_x, using the blocks in CS_p;
(12) if all the blocks in CS_p are original blocks then
(13) Fq = 1;
(14) else
(15) Fq = 0;
(16) end if
(17) for each coded block cb_i do
(18) Generate a data packet D(p, Y_i, cb_i, Fq) and send over interface f;
(19) end for
(20) end if
```

Algorithm 2: Responding Interest.

```
Input: Interest I(p, X, x); interface f
(1) if CS matches then
(2) Respond to Interest according to Algorithm 2;
(3) else
(4) if PIT-IN of interface f matches then
(5) Update PIT-IN;
(6) else
(7) Establish a new PIT-IN entry;
(8) end if
(9) Calculate the incremental Interest ΔI;
(10) if ΔI ≠ null then
(11) Separate the incremental Interest ΔI into several sub-Interests according to Algorithm 1;
(12) for each sub-Interest I_i which needs to be sent over interface i do
(13) if PIT-OUT of i matches then
(14) Update PIT-OUT;
(15) Calculate the incremental Interest ΔI_i;
(16) if ΔI_i = null then
(17) return;
(18) end if
(19) else
(20) Establish a new PIT-OUT entry;
(21) end if
(22) Send Interest I_i (or ΔI_i) from interface i;
(23) end for
(24) end if
(25) end if
```

Algorithm 3: Forwarding Interest.

coded blocks can be cached in the network. The received/decoded original blocks can be cached by any CR. The coded-from-original blocks can only be cached by a single CR, which is the immediate downstream neighbor of the CR that generated the block. Thus, any n coded-from-original blocks cached in the network will be linearly independent, where n is the number of blocks required to recover the content. Fq in data packet $D(p, S, block, Fq)$ is a caching flag that indicates whether the *block* is cacheable or not.

Input: data packet $D(p, Y, block, Fq)$; interface f
(1) **if** PIT-OUT of interface f matches **then**
(2) **if** $Fq = 1$ **then**
(3) Cache data into CS;
(4) **else**
(5) Cache data into CACHE;
(6) **end if**
(7) **for** each interface i in the $facelist$ of the matched PIT-OUT **do**
(8) Find the matched PIT-IN entry, i.e. $\mathrm{PIT}_{\mathrm{in}-i}(p, Z, z)$, where $Y \cap Z \neq \varnothing$;
(9) **if** the matched PIT-IN entry of interface i is satisfied **then**
(10) Respond to the PIT-IN entry $\mathrm{PIT}_{\mathrm{in}-i}(p, Z, z)$, according to Algorithm 2;
(11) Delete the PIT-IN entry;
(12) **else**
(13) **if** the block carried by data is an original block and $|Z| = z$ **then**
(14) Send data $D(p, Y, block, Fq)$ over interface i;
(15) Update $\mathrm{PIT}_{\mathrm{in}-i}(p, Z, z)$ to $\mathrm{PIT}_{\mathrm{in}-i}(p, Z\prime, z\prime)$, where $Z\prime = Z \backslash Y$, $z\prime = z - 1$;
(16) **end if**
(17) **end if**
(18) **end for**
(19) **if** the matched PIT-OUT of interface f is satisfied **then**
(20) Delete the PIT-OUT entry;
(21) **end if**
(22) **else**
(23) Discard data $D(p, Y, block, Fq)$;
(24) **end if**

Algorithm 4: Forwarding data.

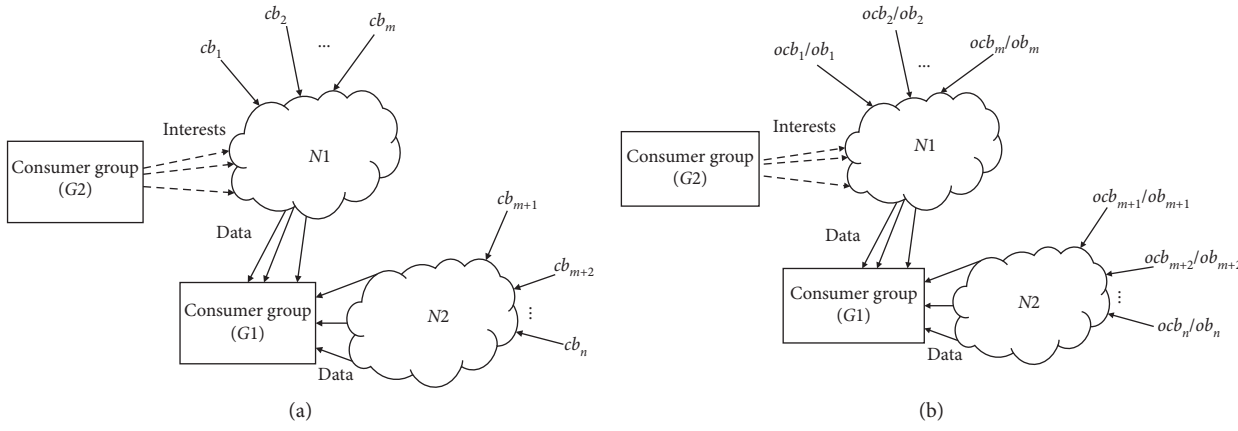


Figure 3: Caching. (a) Caching in ICN. (b) Caching in RSCB.

Since CRs have limited storage capacity, a cache replacement policy is required. When cache replacement occurs, the candidate content p that is to be discarded is selected using the existing content-level cache replacement policy, e.g., least recently used (LRU). Assume t units of cache space are required to cache newly received/decoded blocks and the cache space used to cache the candidate content p is n units (one unit for each block). If $t \geq n$, content p is deleted and $t = t - n$. The first step is repeated until only some of cached blocks of the candidate content p_i need to be discarded to cache new blocks, i.e., $t < n_i$. Chunk-level cache replacement is then introduced to discard blocks in content p_i. Firstly, any original blocks that are contained in the cached coded-from-original blocks are discarded, i.e., the information contained in the original blocks ob_1 and ob_2 may also be contained in the coded-from-original blocks $ocb_1 = \alpha_{11} ob_1 + \alpha_{12} ob_2$. Secondly, the remaining original blocks are coded into a single coded-from-original block. Finally, the received coded-from-original blocks are randomly discarded. These three steps are performed in turn until there is sufficient space for the newly received/decoded blocks, as described in Algorithm 5. The complexity of Algorithm 5 is $O(n)$, where n is the number of evicted content. If content is rarely accessed, only the coded-from-original blocks will be cached to respond to future Interests. Since a single coded-from-original block can respond to an aggregated Interest containing multiple requests for different chunks sent by multiple consumers, our chunk-level

Input: t newly received/decoded blocks
(1) Determine the candidate content p with n blocks using LRU;
(2) **if** $t \geq n$ **then**
(3) Let $t = t - n$;
(4) Discard content p;
(5) **if** $t > 0$ **then**
(6) go to step 1;
(7) **end if**
(8) **else**
(9) Discard the n_1 original blocks which are contained in the coded-from-original blocks;
(10) Let $\Delta t_1 = t - n_1$;
(11) **if** $\Delta t_1 > 0$ **then**
(12) Encode the remaining n_2 original blocks into a single coded blocks;
(13) Let $\Delta t_2 = \Delta t_1 - n_2 + 1$;
(14) **if** $\Delta t_2 > 0$ **then**
(15) Randomly discard Δt_2 received coded-from-original blocks;
(16) **end if**
(17) **end if**
(18) **end if**
(19) Cache the newly received/decoded blocks into the CS;

Algorithm 5: Cache replacement strategy.

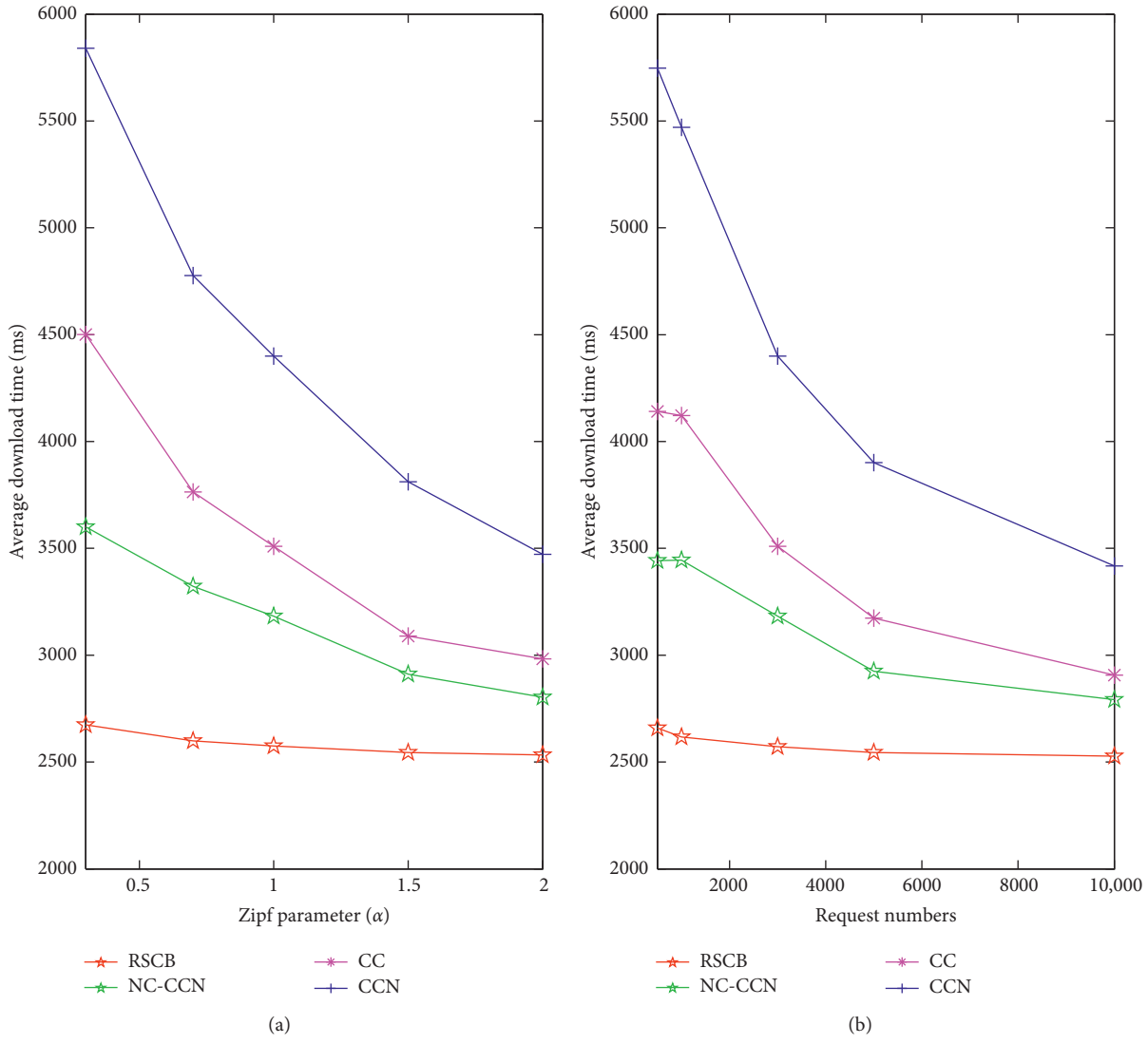


Figure 4: Average download time. (a) Zipf parameter VS average download time. (b) Request numbers VS average download time.

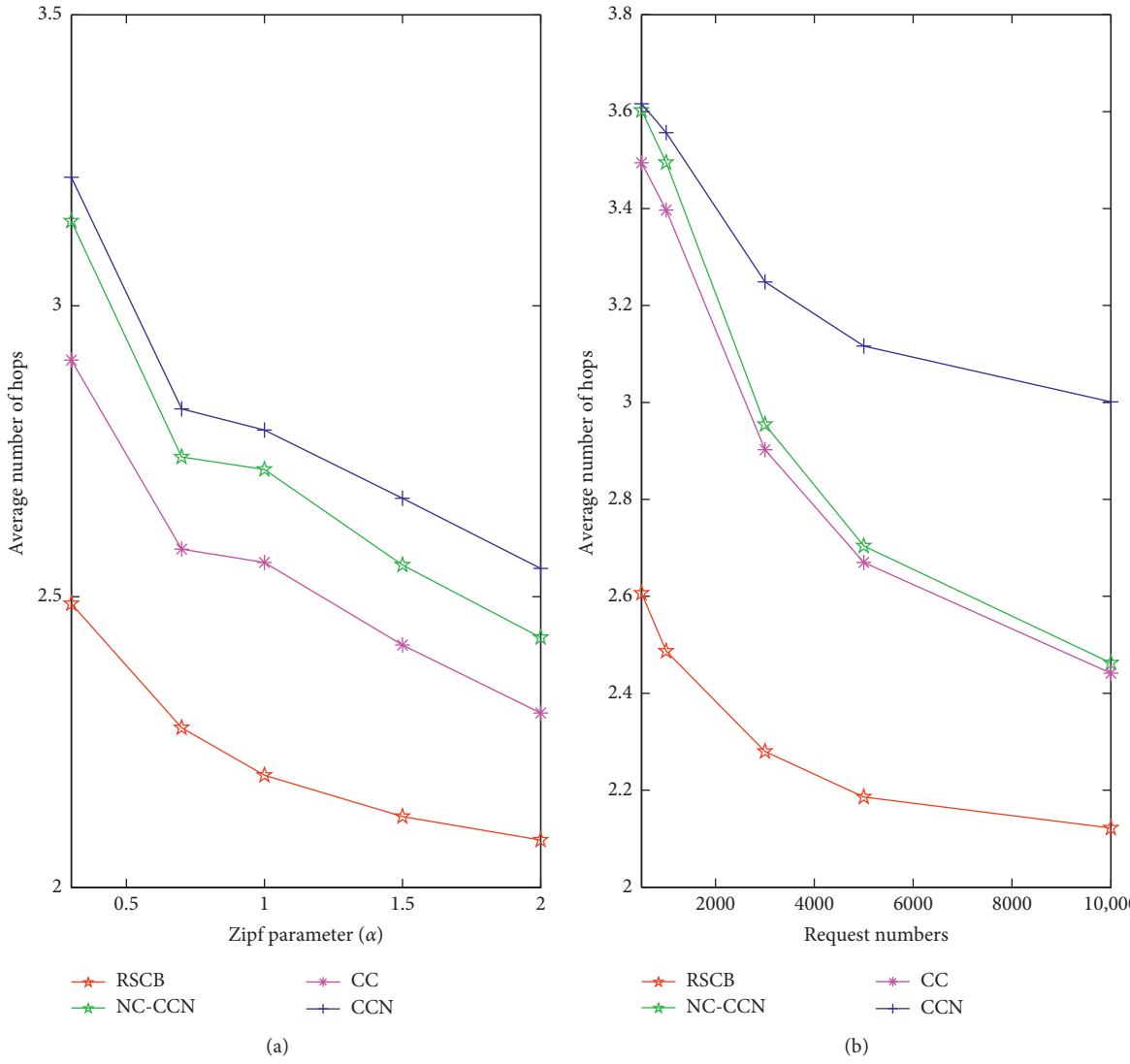


Figure 5: Average number of hops. (a) Zipf parameter VS average number of hops. (b) Request numbers VS average number of hops.

cache replacement policy can effectively increase the cache efficiency.

6. Simulation

In this section, the performance of our model is investigated by comparison with other three schemes: chunk-level CCN strategy (CCN) [4], NC-CCN [21], and CodingCache (CC) [16]. The caching strategy leave copy everywhere (LCE) is incorporated into the above strategies. In LCE, each block or chunk is cached by all CRs on the forwarding path between the content provider and the consumer.

6.1. Simulation Model. BRITE [27, 28] was used to generate the network topology, since it can roughly reflect the actual Internet topology. The Dijkstra algorithm was used to generate the FIB tables. All links have a bandwidth of 1 Gbps. There were a total of 1000 end hosts that were connected to 100 CRs and 10 original content providers were randomly connected to the CRs. 10,000 files were equally partitioned into 400 classes. Each content packet was 1 GB and was divided into 10 generations with each generation containing 10 chunks; each chunk size was 10 MB. In our simulation, only chunks in the same generation could be encoded. The content popularity follows a Zipf distribution with $\alpha \in [0.3, 0.7, 1, 1.5, 2]$. Interests sent by consumers follow a Poisson process. The request number was defined as the number of Interests sent by consumers during the processing period. In our simulations, each CR was equally configured to have a cache space of 0.1%, 0.25%, 0.5%, 1%, and 2% of the overall content catalog size. The default cache size of each CR was set to 10 GB for caching, i.e., 1% of the total content catalog size. Random linear network coding (RLNC) was used for coding. The size of a finite field was 2^8 [29]. The coefficient vector and the generation ID are contained within the signed info field of the data packet. The performance of all four strategies was evaluated under the same simulation environment.

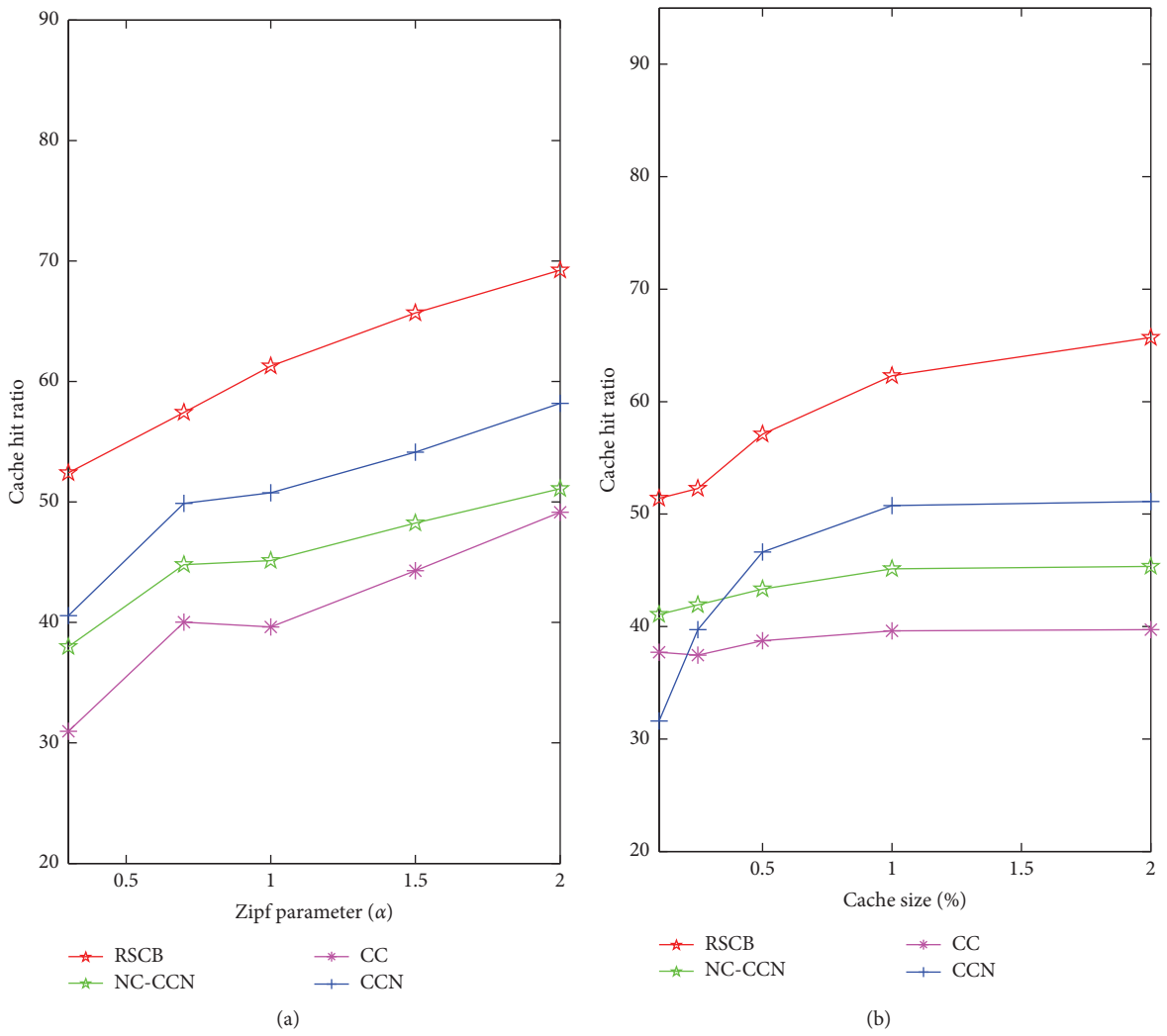


Figure 6: Cache hit ratio. (a) Zipf parameter VS cache hit ratio. (b) Cache size VS cache hit ratio.

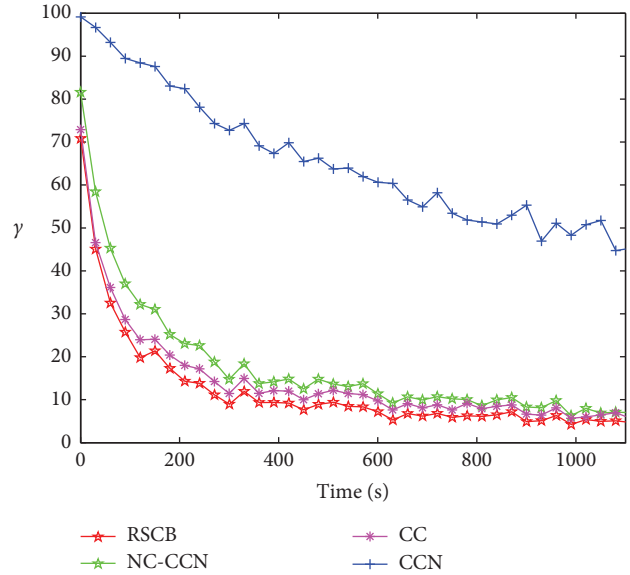


Figure 7: Server hit reduction ratio.

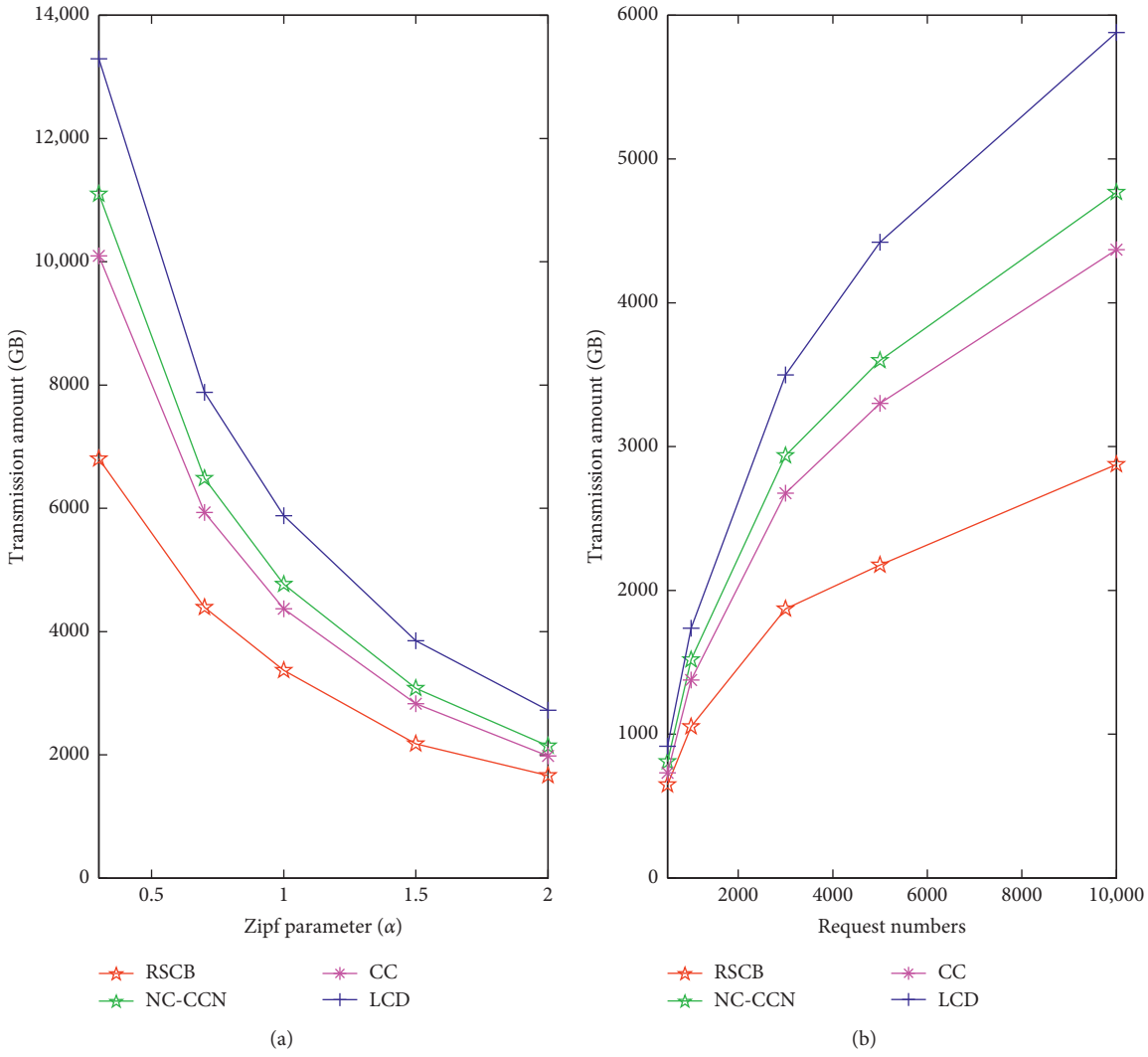


Figure 8: Traffic. (a) Zipf parameter VS transmission amount. (b) Request numbers VS transmission amount.

The following parameters were used for the evaluation:

(i) Average download time: the average time for consumers to download each successfully received content request response.

(ii) Average number of hops: the average number of hops for each successfully received chunk from the provider to the consumer.

(iii) Cache hit ratio: the ratio of the number of Interests that were satisfied by the caches to the number of Interests that were satisfied by either the caches or the server.

(iv) Server hit reduction ratio $\gamma(t)$:

$$\gamma(t) = \frac{\sum_{i=1}^{N(t)} w_i(t)}{N(t)}, \tag{7}$$

where $w_i(t) = 0$ if the chunk i is sent from a cache or an aggregated Interest; otherwise, $w_i(t) = 1$. $N(t)$ is the number of chunks received by all consumers. t indicates that the data were collected from time $(t - \Delta t)$ to t [20].

(v) Traffic: the total traffic to deliver the data packets over the whole request process.

(vi) Average number of Interests: the average number of Interest packets that were handled by each CR for each chunk that was successfully received by the consumer, as in [21].

6.2. Simulation Results. Due to its network coding-based content delivery and caching strategies, our proposed RSCB always achieves the best performance in terms of having the shortest average download time, the highest cache hit ratio, the lowest server hit reduction ratio, and the lowest transmission volume. RSCB ensures that consumers will receive sufficient independent coded blocks within a single round and the coded blocks cached by the CRs can be used as multiple chunks.

Figure 4 plots the average download time of the four caching strategies for different system parameters. The

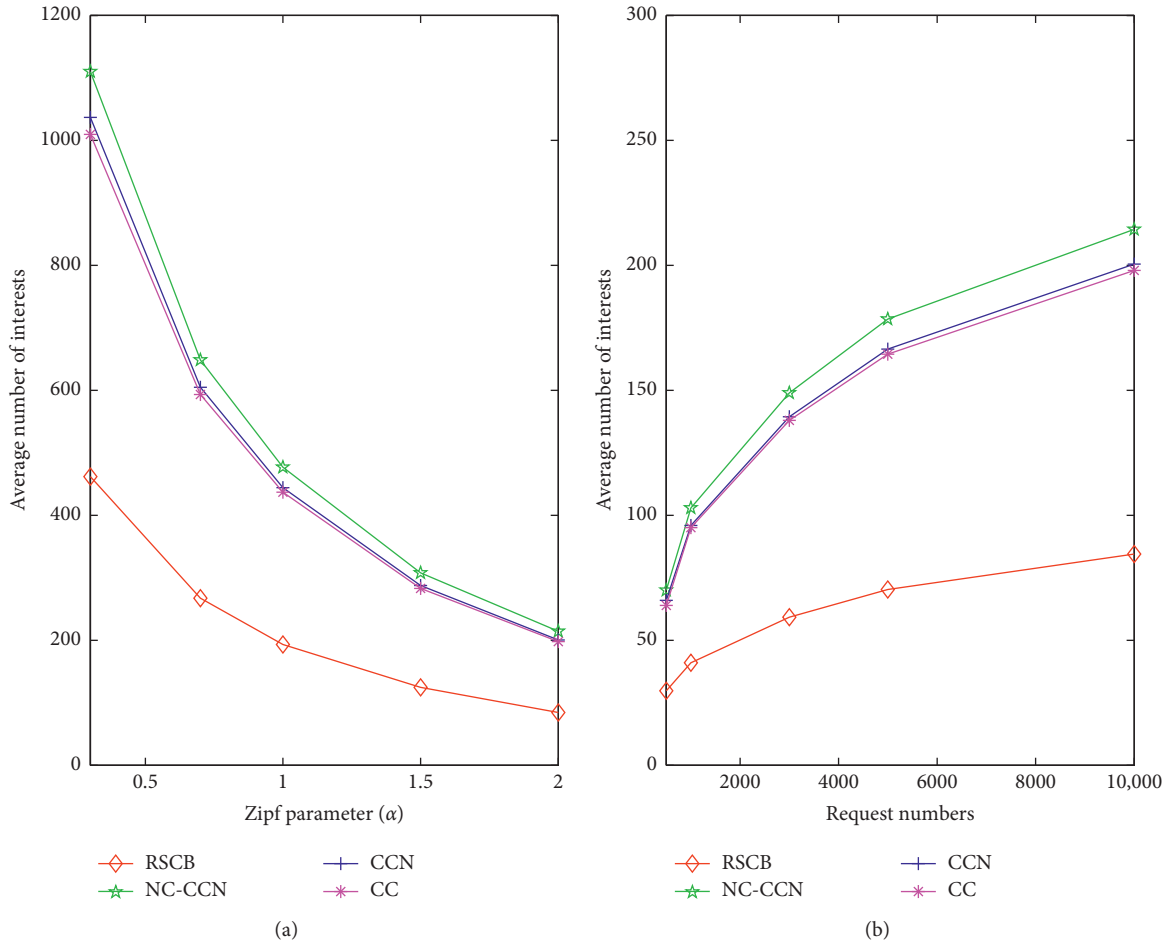


Figure 9: Average number of Interests. (a) Zipf parameter VS average number of Interests. (b) Request numbers VS average number of Interests.

average download time decreases as the Zipf parameter α is increased, as shown in Figure 4(a), since a larger Zipf parameter α indicates that the Interests sent by consumers are concentrated on a smaller set of contents. As the number of requests increases, chunks that have already been requested will be cached on more CRs and thus consumers can retrieve chunks directly from the CRs, which are much closer to the consumers. Therefore, the average download time will be reduced (Figure 4(b)). RSCB performs much better even for a small Zipf parameter and a low number of Interests, since it can retrieve chunks or coded blocks from multiple CRs simultaneously. Compared with other schemes, RSCB provides consumers with enough independent coded blocks in a single round.

In RSCB, one coded-from-original block can be used to respond to an aggregated Interest for multiple chunks requested by different consumers. For instance, the coded-from-original block, $ocb_1 = \alpha_{11}ob_1 + \alpha_{12}ob_2$, can satisfy the Interest for chunk ob_1 from consumer 1 and the Interest for chunk ob_2 from consumer 2, as shown in Figure 1(b). Thus, RSCB achieves the best caching performance, in terms of average download hops (Figure 5), cache hit ratio (Figure 6), and server hit reduction ratio (Figure 7).

Figure 8 shows the traffic for different caching schemes, and it can be seen that RSCB has the lowest transmission volume. Moreover, we can see that as the number of requests increases, RSCB has a higher traffic saving too due to its Interest aggregation scheme which saves on traffic required to deliver $n - (n_1 + n_2)$ blocks, as per equation (3).

RSCB can also aggregate Interests for different chunks into a single Interest. As shown in Figure 9, the average number of Interests processed by the CR is much lower compared with other schemes. In ICN, a consumer requests content with N chunks by sending out N Interests, and thus the CR needs to process N Interests. However, in RSCB, only one aggregated Interest containing several Interests will be processed by the CR. This can reduce the cost of transmitting and processing the Interest.

7. Conclusion and Discussion

In this paper, we have proposed a request-specific coded-block strategy to reduce the transmission volume. Additionally, a chunk-level on-path non-cooperative coded caching and replacing strategy has been proposed to

improve the caching efficiency. Our method enables a consumer to multicast a set of Interests in order to obtain multiple content chunks simultaneously from multiple CRs. The traffic can be reduced by encoding chunks that meet in an intermediate CR and have been requested by different consumers. A novel Interest forwarding-responding strategy has been proposed to guarantee that the minimum number of coded blocks will be requested and that the blocks will be linearly independent. A network coding-based caching and replacing mechanism has been designed to guarantee that the cached blocks can be reused. A chunk-level coded cache replacement strategy has been proposed to discard blocks. Rather than discarding the original blocks, the CR will encode the original blocks into a single coded-from-original block to release cache space when cache replacement is required. A single coded-from-original block can satisfy multiple Interests from different consumers for a set of its component original blocks. Therefore, this will increase the caching diversity without requiring extra cache space. The simulation results have confirmed that the RSCB scheme outperforms the other three strategies.

However, although there are many benefits in deploying network coding in ICN, it also introduces some additional costs for computation and communication. Some studies have already proven that RLNC is a practical method which has acceptable costs. Since ICN is a new architecture, there are still many issues that need to be resolved before ICN can be deployed, such as efficient operation of PIT and FIB at a chunk level [30, 31].

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (nos. 61571141, 61702120, 61972104, and 61902080), the National Key Research and Development Project (no. SQ2019YFB180098), the Guangdong Natural Science Foundation (no. 2017A030310591), the Guangdong Provincial Application-Oriented Technical Research and Development Special Fund Project (nos. 2017B010125003 and 2015B010131017), the Key Areas of Guangdong Province (nos. 2019B010118001 and 2017B030306015), the Guangdong Future Network Engineering Technology Research Center (no. 2016GCZX006), the Science and Technology Program of Guangzhou (no. 201604016108), the Project of Youth Innovation Talent of Universities in Guangdong (nos. 2017KQNCX120 and 2016KQNCX091), the Guangdong Science and Technology Development Project (no. 2017A090905023), the Key Projects of Guangdong Science and Technology, and the Science and Technology Project in Guangzhou (no. 201803010081).

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

[2] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: strategies, challenges, and future research directions," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 2, pp. 1443–1474, 2018.

[3] S. Lee, I. Yeom, and D. Kim, "T-caching: enhancing feasibility of in-network caching in icn," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 7, pp. 1486–1498, 2020.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," *Communications of the ACM*, vol. 55, no. 1, pp. 117–124, 2012.

[5] R. Ahlswede, N. Ning Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[6] B. Saleh and D. Qiu, "Performance analysis of network-coding-based p2p live streaming systems," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2140–2153, 2016.

[7] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2235–2245, Miami, FL, USA, March 2005.

[8] M. Karmoose, M. Cardone, and C. Fragouli, "Simplifying wireless social caching via network coding," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5512–5525, Nov 2018.

[9] C. Xu, P. Wang, C. Xiong, X. Wei, and G.-M. Muntean, "Pipeline network coding-based multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Broadcasting*, vol. 63, no. 2, pp. 376–390, 2017.

[10] M. Bilal and S.-G. Kang, "Network-coding approach for information-centric networking," *IEEE Systems Journal*, vol. 13, no. 2, pp. 1376–1385, 2019.

[11] H. R. Sadjadpour, "A new design for information centric networks," in *Proceedings of the 48th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, Princeton, NJ, USA, March 2014.

[12] J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal, "An optimal cache management framework for information-centric networks with network coding," in *Proceedings of the 2014 IFIP Networking Conference*, pp. 1–9, Trondheim, Norway, June 2014.

[13] Q. Xiang, H. Zhang, J. Wang, G. Xing, S. Lin, and X. Liu, "On optimal diversity in network-coding-based routing in wireless networks," in *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 765–773, Kowloon, Hong Kong, April 2015.

[14] J. Llorca, A. M. Tulino, K. Guan, and D. C. Kilper, "Network-coded caching-aided multicast for efficient content delivery," in *Proceedings of the 2013 IEEE International Conference on Communications (ICC)*, pp. 3557–3562, Budapest, Hungary, 2013.

[15] P. Talebifard, H. Nicanfar, and V. C. Leung, "A content centric approach to energy efficient data dissemination," in

Proceedings of the 2013 IEEE International Systems Conference (SysCon), pp. 873–877, Orlando, FL, USA, April 2013.

[16] Q. Wu, Z. Li, and G. Xie, "Codingcache: multipath-aware ccn cache with network coding," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking-ICN'13*, pp. 41-42, USA, 2013.

[17] M.-J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: an architectural case for information dispersion through native network coding," in *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications-NoM'12*, pp. 31–36, USA, June 2012.

[18] W.-X. Liu, S.-Z. Yu, G. Tan, and J. Cai, "Information-centric networking with built-in network coding to achieve multi-source transmission at network-layer," *Computer Networks*, vol. 115, pp. 110–128, 2017.

[19] J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun, "Netcodccn: a network coding approach for content-centric networks," in *Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, April 2016.

[20] D. Nguyen, M. Fukushima, K. Sugiyama, and A. Tagami, "CoNAT: a network coding-based interest aggregation in content centric networks," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)*, pp. 5715–5720, London, UK, June 2015.

[21] G. Zhang and Z. Xu, "Combing CCN with network coding: an architectural perspective," *Computer Networks*, vol. 94, pp. 219–230, 2016.

[22] C. Shan, J. Cai, Y. Liu, and J. Luo, "Node importance to community based caching strategy for information centric networking," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 21, Article ID e4797, 2019.

[23] Y. Liu and S.-Z. Yu, "Network coding-based multisource content delivery in content centric networking," *Journal of Network and Computer Applications*, vol. 64, pp. 167–175, 2016.

[24] J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal, "A minimum cost cache management framework for information-centric networks with network coding," *Computer Networks*, vol. 110, pp. 1–17, 2016.

[25] N. Lal, S. Kumar, and V. K. Chaurasiya, "A network-coded caching-based multicasting scheme for information-centric networking (ICN)," *Iranian Journal of Science and Technology*, vol. 43, no. 3, pp. 427–438, 2019.

[26] J. Saltarin, T. Braun, E. Bourtsoulatze, and N. Thomos, "Popnetcod: a popularity-based caching policy for network coding enabled named data networking," 2019, http://arxiv.org/abs/1901.01187.

[27] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: an approach to universal topology generation," in *Proceedings of the Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 346–353, Cincinnati, OH, USA, August 2001.

[28] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in internet topologies," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 2, pp. 18–28, Apr. 2000.

[29] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 450–463, 2008.

[30] Y. Wang, K. He, H. Dai et al., "Scalable name lookup in ndn using effective name component encoding," in *Proceedings of the ICDCS '12*, pp. 688–697, June 2012.

[31] T. Song, H. Yuan, P. Crowley, and B. Zhang, "Scalable name-based packet forwarding: from millions to billions," in *Proceedings of the ACM-ICN'15*, pp. 19–28, San Francisco, CA, USA, 2015.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 5476896, 9 pages
https://doi.org/10.1155/2020/5476896

Research Article

Chinese Tone Recognition Based on 3D Dynamic Muscle Information

JianRong Wang,[1,2] Li Wan,[1] Ju Zhang,[1] Qiang Fang,[3] Fan Yang,[1] and Jing Hu[1]

[1]*College of Intelligence and Computing, Tianjin University, Tianjin 300350, China*
[2]*Tianjin Key Laboratory of Advanced Networking, Tianjin University, Tianjin 300350, China*
[3]*Institute of Linguistics, Chinese Academy of Social Sciences, Beijing 100732, China*

Correspondence should be addressed to Jing Hu; mavis_huhu@tju.edu.cn

Received 20 November 2019; Revised 13 February 2020; Accepted 11 April 2020; Published 31 May 2020

Guest Editor: Jianbiao Zhang

To advance the study of lip-reading recognition in accordance with Chinese pronunciation norms, we carefully investigated Mandarin tone recognition based on visual information, in contrast to that of the previous character-based Chinese lip reading technique. In this paper, we mainly studied the vowel tonal transformation in Chinese pronunciation and designed a lightweight skipping convolution network framework (SCNet). And, the experimental results showed that the SCNet was sensitive to the more detailed description of the pitch change than that of the traditional model and achieved a better tone recognition effect and outstanding antiinterference performance. In addition, we conducted a more detailed study on the assistance of the deep texture information in lip-reading recognition. We found that the deep texture information has a significant effect on tone recognition, and the possibility of multimodal lip reading in Chinese tone recognition was confirmed. Similarly, we verified the role of the SCNet syllable tone recognition and found that the vowel and syllable tone recognition accuracy of our model was as high as 97.3%, which also showed the robustness of our proposed method for Chinese tone recognition and it can be widely used for tone recognition.

1. Introduction

In recent years, the superior performance of lip reading in robust speech recognition has received widespread attention. The goal of lip reading is to improve the robustness of speech recognition in special situations such as low signal-noise ratio (SNR) or silent environments. However, due to the complexity and variability of Chinese pronunciation, the performance of lip-reading recognition in Chinese is not always satisfactory in real-world scenarios.

One of the most important tasks of lip-reading recognition is feature extraction. Currently, there are two main categories of visual information extraction in the lip reading system, i.e., pixel-based methods and model-based methods. Pixel-based methods extract visual features from the image directly or after some preprocessing and transformation. Yuhas et al. [1] used the greyscale image pixel information of the lip and its surrounding areas as features. Wolff et al. [2] used the horizontal and vertical scanning lines centred on the lips as the eigenvector. Since the method of directly using the pixel information of the image as a feature is blind, more effective and targeted approaches, such as discrete cosine transform (DCT), principle component analysis (PCA), singular value decomposition (SVD), discrete wavelet transform (DWT), and linear discriminant analysis (LDA) [3–5], were proposed to reduce the information redundancy. The pixel-based method can make full use of pixel information to extract more comprehensive lip features. However, the feature vectors are high dimensional and redundant. Also, the pixel-based method is very sensitive to light, shadow, pronunciation, and other conditions. Besides, model-based methods aim to establish a parametric mathematical model and then use the model parameters to describe lip contour

information. Kaynak et al. [6] used the horizontal and vertical distance of lip contours, the lip corner angle, and the first-order derivative of the lip corner angle. Zhang et al. [7] proposed geometric features of the lips, containing mouth width, upper/lower lip width, lip opening height/width, and the distance between the horizontal lip line and the upper. Model-based methods utilize low dimensional features to express image features, and the feature is typically not changed by factors such as translation, rotation, scaling, or illumination. Nevertheless, both methods extract relevant information directly from the region of interest (ROI) in the planar image [8].

With the development of high-sensitivity RGB-D cameras, the three-dimensional information of the speaker's face can be extracted more accurately. For instance, Yargıç and Muzaffer [9] developed a lip reading system that uses a Kinect camera to acquire the depth feature points and then extracts the angular features of the lip reading. Palecek et al. [10] studied the fusion performance of face depth data in isolated word visual speech recognition tasks. Rekik et al. [11, 12] proposed an adaptive lip-reading system based on image and depth data. Wang et al. [13] used 3D lip points obtained from Kinect, improving the performance of multimodal speech recognition. Studies by these pioneers have demonstrated the effectiveness of depth information in lip-reading recognition. Since the depth information is not affected by illumination, skin colour, etc. [14], the defects of the two-dimensional image information are compensated for. However, since the characteristics of the lips are usually obtained from discrete three-dimensional points or facial depth images, it is difficult to fully represent the characteristics of the lips.

The currently proposed lip-reading recognition based on 3D depth information does not consider the inherent texture problem of driving the lip motion during natural speech changes. In our previous work [15], to explore the internal mechanism of the speech process, we conducted an in-depth study on the facial texture information that drives the changes in lip reading and explored the facial texture information for lip movement changes in Chinese vowel pronunciation that have significant influence. However, since Chinese pronunciation is a strict tone-changing language, the transformation of the pitch has a significant role in the understanding of Chinese. Therefore, the exploration of Chinese tonal transformation in the current lip-reading research based on 3D information is important.

In this work, we focus on the study of the vowel tonal changes in Chinese pronunciation. Our main contributions are as follows. (1) For Chinese pronunciation tonal changes, we propose a new lightweight network framework, the SCNet, which is more sensitive to the transformation of details compared with the traditional network architecture. (2) We explore in detail the important influence of our proposed deep facial texture information on the change of vowel tones in auxiliary lip reading. (3) In syllable recognition with the depth texture, the experimental results show the ubiquity and good performance of the SCNet model in integrated tone recognition.

The rest of this paper is organized as follows. Section 2 introduces the data collection and preprocessing. Section 3 presents the proposed model architecture. Section 4 introduces our experimental results. Section 5 summarizes our work and introduces the future work.

2. Data Collection and Feature Preprocessing

2.1. Data Collection. Eight native speakers of Chinese, four males and four females, served as the subjects. All the subjects used standard Mandarin pronunciations without any accent influence. In the pronunciation of Chinese, each syllable has four different pitch changes (tones 1–4). In fact, there is a fifth pronunciation types in Chinese pronunciation, which is the unvoiced sound (i.e., a special silent tone in Chinese pronunciation) commonly spoken in Chinese. In order to explore the effects of different pitch transformations, we eliminated the unvoiced sounds that are rarely pronounced in Chinese, so in the experiment each syllable contained only one of the four commonly used tones. In terms of experimental data, we collected 5 vowels (/a/, /e/, /i/, /o/, and /u/) and 5 syllables (/ta/, /te/, /ti/, /fo/, and /tu/), a total of 40 tones. During the recording process, each tone was pronounced 10 times per person. For example, four tuned syllables (/ā/, / á/, /ǎ/, and /à/) were obtained by combining four lexical tones with the atonal syllable /a/.

The data acquisition device used a Microsoft Kinect V2 face real-time tracking camera and this camera through facial key points to generate real-time 3D point clouds (1347 facial key points). In [16], Mallick et al. have proved that the muscles of the facial expression recognition based on point cloud is successful, and it has been verified that the generation of 3D face point clouds is related to muscle distribution. At the same time, their experiments show that the shape of the face of point cloud generated face has nothing to do and can be very stable in different faces of the same position. Meanwhile, [17, 18] also prove the stability and effectiveness of Kinect V2. To ensure its quality, we collected the data in a standard silent room. The data collection scenario is shown in Figure 1.

During the process, we reindexed the 1347 points. The index of feature points in the lip area is shown in Figure 2(b), which used only the collected image information and 3D depth information. By considering the changes in the head model during movement, we corrected the head rotation angle in the $X-$ axis, $Y-$ axis, and $Z-$ axis directions. As an example, the angle between vector $\overrightarrow{P_{11}P_{31}}$ (P_{11} and P_{13} are two points in Figure 2(b)) and plane XY is calculated as follows:

$$\alpha_{XY} = -1 \times \arctan\left(\frac{(Z_{31} - Z_{11})}{(x_{31} - x_{11})}\right), \tag{1}$$

where (x_{11}, y_{11}, z_{11}) and (x_{31}, y_{31}, z_{31}) are the coordinates of P_{11} and P_{31} and 31 and 11 represent the coordinate point numbers on the plane XY. The rotated face point coordinates parallel to the XY plane are constructed by the following algorithm.

Figure 1: Kinect V2 recording data experimental scene.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\alpha_{XY}) & 0 & \sin(\alpha_{XY}) \\ 0 & 1 & 0 \\ -\sin(\alpha_{XY}) & 0 & \cos(\alpha_{XY}) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (2)$$

Finally, we acquired the standard point set of the real speaker's face.

2.2. Feature Preprocessing

2.2.1. Image Feature Preprocessing. For the collected image information, we used the open source OpenCV lib library to intercept a 128×100 lip region of interest, as shown in Figure 4(a), and then used the image sequence representation method proposed by Saitoh et al. The pronunciation of the syllables extracts 16 consecutive frames (center − 8, center + 8) in the middle of the pronunciation to form a continuous sequence of image lip motion changes (4×4, from left to right, top to bottom) and uses a gamma transform ($V_{\text{out}} = V_{\text{in}}^{\gamma}$) for light enhancement to augment the data, as shown in Figure 4(b) (take 16 sheets and then sort).

2.2.2. Muscle Dynamics Features. According to this study, there are six main types of muscles that drive lip movement in facial muscles. The distribution of the facial functions and characteristics of each muscle are presented in Tables 1 and 2 reflect the specific names of each muscle and the characteristic point identification of each muscle in the kinect data. In the specific depth texture feature representation, we extracted the two most representative depth, muscle length change, and muscle dynamic characteristic data points.

(1) Muscle Length Change Information. The length feature is expressed as $[1/R]$, where l represents the muscle length vector at the time of speech and R represents the muscle length vector at the time of relaxation, which eliminates the differences between different speakers.

(2) Muscle Dynamics Information. The muscle dynamics information characterizes the relationship between the facial muscles and facial feature points and reflects the intrinsic commonality between different speakers. We also analysed the effects of different muscles on the displacement of the feature points as the drivers of muscle dynamic transformation. Regarding the feature information, the vector variation between the muscles is obtained by calculating the transformation trend of different feature points in adjacent frames. The specific expression is as follows:

$$F_{\text{muscle\_}i} = \left[\frac{P_{j\text{–end}} - P_{j\text{–start}}}{l_j} \right] \cdot \overline{V}_{\text{muscle}}, \quad (3)$$

where $F_{\text{muscle\_}i}$ represents the momentum change of the feature point i, $P_{j\text{–start}}$ and $P_{j\text{–end}}$ represent the start and end points, respectively, of the muscle j, and the direction of the muscle movement at each point is represented by decomposing the displacement subvector of each point. $\overline{V}_{\text{muscle}}$ Indicates the length of movement of each muscle point.

3. Network Architecture

Considering the subtle differences in the mouth shape changes in Chinese tonal changes, we designed a lightweight skip convolutional structure network (SCNet) with subtle descriptions of feature changes to evaluate our proposed 3D lip features and to explore the feasibility of tonal changes and syllable lip-reading recognition. The overall architecture is shown in Figure 3.

The network architecture was inspired by that of VGG [19] and ResNet [20]. In the initial phase of the network, we used three 3×3 convolutional layers with a stride of 2 to extract the surface features of the image. This network structure reduces not only the overall parameters of the network but also the accuracy loss of the feature map.

The main body of network structure is two connected feature extraction blocks, and they different from the current remaining block structure. Two subconnection blocks adopt different subsampling expressions. At the back of block 1, to make the edge features more obvious, the maximum pool was used to indicate the specificity of different features, highlighting the features of different feature maps. And, at the block 2, to make the features, the map was associated with the specificity of the feature maps more smoothly and effectively using global average pooling. The two connection block structures in the frame were slightly different. In the second block, to maximize the smoothing effect after block 1, the last convolutional layer output channel in block 2 was doubled and the rest was the same as that of block 1. This structure also showed good performance in the experiment. At the last end, a 128-dimensional linear layer was connected, and then the classification probability was obtained.

3.1. Skip Convolution Structure.

We used a skip connection in each block. The structure of each block is shown in Figure 2(b), and the connection of each block is defined as follows:

$$y = F(x) + G(x), \quad (4)$$

where x and y represent the input and output, respectively, of each block and $F(x)$ represents the learning function of

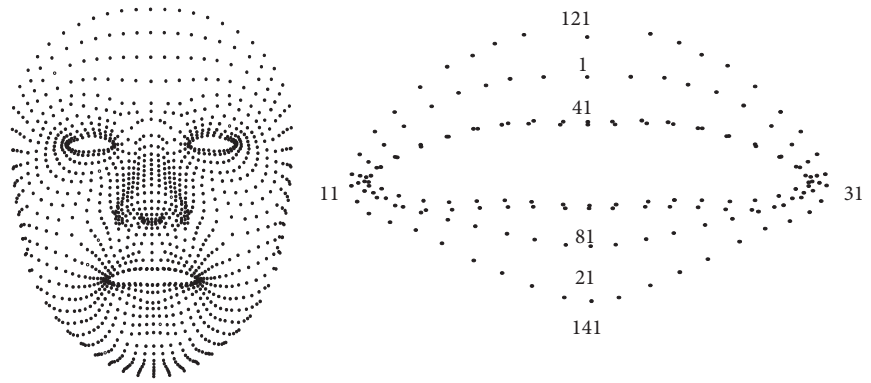


Figure 2: (a) Predefined 1347 planar facial points. (b) Reindexed 160 points of lip area.

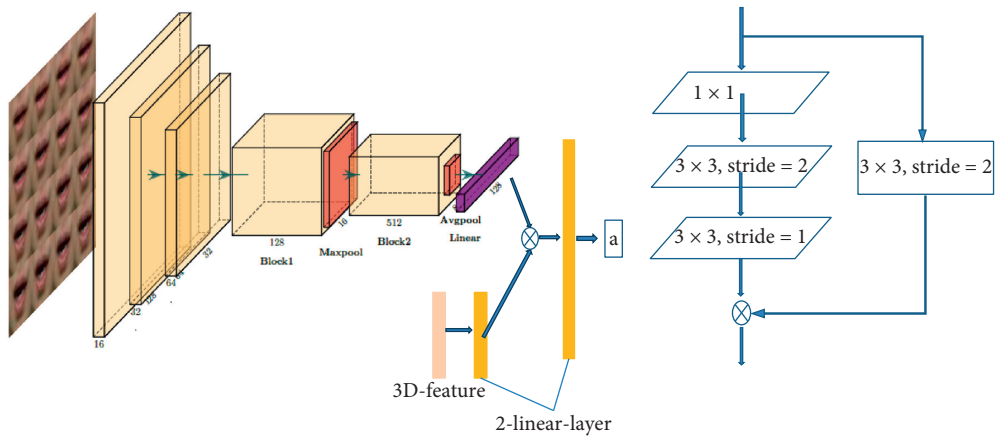


Figure 3: Our SCNet structure. (a) The overall structure of the model and (b) the skip connection structure.

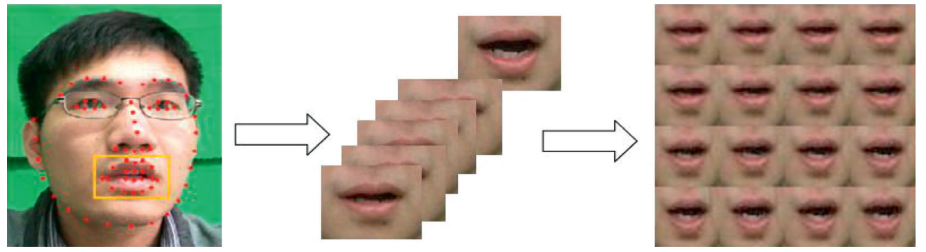


Figure 4: Picture-stitching process. (a) Feature extraction of interest and (b) representation of the image sequence splicing process.

Table 1: The structures and functions of the major muscles.

Muscle name	Structure	Function
Levator labli superioris	From the medial infraorbital margin to the skin and muscle of the upper lip	Elevates the upper lip
Levator anguli oris	From the canine fossa, below the infraorbital foramen	Draws the angle of the mouth
Zygomaticus	Extends from the zygomatic arch to the corners of the mouth	Draws the angle of the mouth
Buccinator	From the alveolar processes of the maxilla and mandible and the temporomandibular joint	Pulls back the angle of the mouth
Orbicularis oris	Composed of four independent quadrants, gives an appearance of circularity	Encircles the mouth
Depressor anguli oris	From the tubercle of the mandible to the modiolus of the mouth	Depresses the angle of mouth

Table 2: The starting and ending coordinates of each muscle.

Muscle name	Start point	End point	Affected lip points
Levator anguli oris	603	126	125, 126, 127, 128
Zygomaticus	650	131	129, 130, 131
Buccinator	522	131	127, 128, 129, 130, 131
Levator labli superioris	769	165	125, 126
Orbicularis oris	717	126	125, 126
Depressor anguli oris	665	127	127, 128, 129, 1230, 131

direct connection. As Figure 3(b) shows, the direct connection is composed of three convolution layers, so $F(x)$ is specifically expressed as $F(x) = W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 \cdot x))$, in which σ is LeakyReLU and $G(x)$ is the skip connection, which represents the connection structure of a layer and is given by the formula $G(x) = W \cdot x$. Since the regularization layer was introduced, to reduce the parameter changes in this architecture, the bias item was not led into. Finally, the $F(x) + G(x)$ operation represents the direct weight addition of the direct and skip connection, rather than the corresponding result splicing.

Equation (4) is mainly divided into two parts: direct connection structure and skip structure. In the stage of direct connection structure, first we used a 1×1 convolution, followed by a 3×3 convolution, with a stride of 2 to obtain more detailed feature information, and then the network optimization is connected to a 3×3 convolution kernel, with a stride of 1 to simulate the processing of the Sobel matrix on the feature boundary. This structure makes the boundary features more obvious, so that the feature was better characterized in the feature judgement area. In the skip module, we used a 3×3 convolution block, with a stride of 2, and the number of channels was increased. This procedure generates the same channel for the network, and the same size is more convenient for feature stitching. This method also ensures the fusion of the image on the feature structure. The purpose of the traditional Res block is to ensure the characterization of the local structure and the global feature to make the network structure more representative. We use this structure to consider that the 1×1 convolution has retained the global feature, using a 3×3 convolution. This convolution ensures the multiscale representation of the network structure.

3.2. Feature Fusion Structure. The expression for feature fusion structure is given as follows:

$$\text{Infor}_{\text{cat}} = F_{\text{fusion}}\left(\text{Infor}_{\text{img}}, \text{Infor}_{\text{depth}}\right). \tag{5}$$

To better integrate the depth information and picture information, we adopted a decision fusion method to deeply integrate the two different kinds of information. The specific expression is shown in formula (5), where $\text{Infor}_{\text{img}}$ represents the 128-dimensional information acquired by the SCNet. The depth feature, $\text{Infor}_{\text{depth}}$ represents the depth feature of the shallow stitching after two layers are fully connected, and F_{fusion} indicates the fusion strategy. Thus, the feature, $\text{Infor}_{\text{cat}}$, after the fusion of the two, was decoded by a linear layer of one layer and output.

3.3. Implementation Detail. In the experiment, the input size of our image is 112×112. Since the image was adjusted before input, no corresponding data enhancement method was used during the experiment. Batch normalization (BN) [21] was adopted in the network after each convolution, before activation and after the BN. For the network weights, the random initialization method was adopted and the network was trained from zero. An Adam optimizer was used in the experiment, and the small batch size was set to 30. The learning rate started at 0.0003, and the expression of the learning rate attenuation functions is shown in the following formula:

$$\text{new\_lr} = \text{lr} \times \gamma^{(\text{epoch} - \text{sleep}_{\text{epoch}}+1)/\text{half}}, \tag{6}$$

where lr represents the last round of the learning rate, $\text{sleep}_{\text{epoch}}$ (20) iterations decay once, and each damping coefficient is γ (0.5) times (epoch − sleep_epoch + 1)/half (5) − th. We did not use dropout during the implementation.

4. Experiments and Results

In the experiment, to verify the smoothness of the proposed model on the whole dataset, we set the experimental scheme to a five-fold cross-validation and calculated the average of all the results as the final experimental result.

4.1. Cross-Validation. To ensure the full use of the data and the accuracy of the experimental results in our experiments, we designed a 5-fold cross-validation. We randomly divided all the experimental data into 5 parts. Water sampling was used for the data division. The data in each sample set consisted of only 1860 groups. Four tests were used to train one test, and the experiment was performed for a total of 5 rounds, so that each could be used as the training set and test set and each experiment would give an independent result.

Because vowels play a leading role in the whole pronunciation process, in the experiment, in order to verify the difference between the entire syllable recognition effect and the different syllable recognition performance of each syllable, we first aimed at each vowel recognition accuracy was discussed, and then further analysis of tone recognition of vowels with different tones. By using different speech expressions, we ignore the unvoiced sounds in Chinese pronunciation to verify that our proposed SCNet has considerable experimental results in terms of accuracy of tone recognition and accuracy of the entire syllable recognition.

4.2. Vowel Detection and Vowel Tone Detection. We first verified the validity of our proposed model and compared it with the traditional models (VGG, ResNet, DenseNet [22]); in addition, we tested the effects of the different models on vowel recognition and vowel tone recognition. To ensure the fairness of the comparison, a linear 1000×128 layer and a softmax classification layer were added to the traditional model, and the optimal values the parameter settings were selected.

Figures 5 and 6 show the single vowel recognition results and the vowel tone recognition results, respectively. By comparing the two images quantitatively, we found that all the models showed good recognition performance; specifically, the proposed vowel distinction SCNet reached a recognition rate of almost 100%, and the tone recognition effect was significantly higher than that of the traditional model structure. A comparison of the overall results of several models in terms of the network depth, parameters, and accuracy is shown in Table 3. It was found that the SCNet gave the optimal values of the three parameters, especially those of the parametric variables. Compared with those of the previous models, the SCNet parameters were only 1/50 of the VGG value, 1/4 of ResNet value, and 1/3 of DenseNet value and even more advantages of the experimental results. These results indicated that our designed model was advantageous for processing real-time data and had better performance than that of the existing traditional framework.

Our analysis of this experimental phenomenon is based on the application of the SCNet architecture to the transformation of subtle differences in the datasets. This architecture showed good results for the description of the data details.

As a whole, the experiment can show such excellent results and attribute the success to the following characteristics of the network structure: (1) in tone recognition, the degree of differentiation of the mouth shape between different tones of the same syllable is very small, and we used a $3\times$ filter in the experiment. The use of such a small convolution kernel can enhance the fine feature structure discrimination. (2) Based on several previous verifications, it was proved that skipping convolutions can preserve the feature transformations between feature maps, which in addition is more conducive to the propagation of gradients than are traditional direct connections. The jump connection proposed in this paper showed that our method can capture more delicate network structure features and thus improve the fine discrimination performance. (3) Different downsampling methods between different structural blocks can be used in feature selection, highlight the propagation between different features, and make the network structure smoother, which is more conducive to the expression of different detailed features.

4.3. Texture Depth Information Fusion. To better verify the validity of the depth texture information in tone recognition, we designed a series of experiments to confirm the correctness of our conjecture.

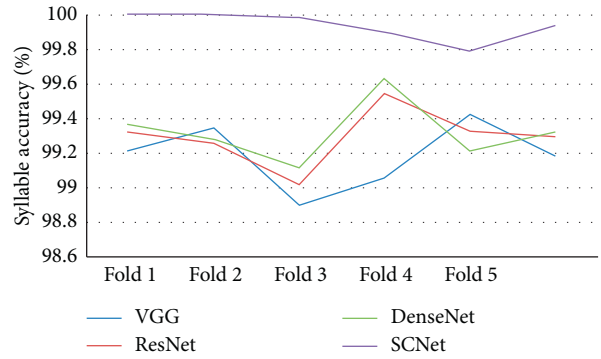


Figure 5: Vowel recognition results of different models.

The results of the tone recognition of the picture only and the tone recognition after the fusion of the depth information are shown in Figure 7. The experimental results showed that after fusion of the texture depth information, the recognition result of the image-only tone recognition increased by 2%, and especially in the case of low picture recognition rate, the effect on the tone recognition was obvious, which indicated that our proposed 3D depth texture information significantly influenced the auxiliary tone recognition. This effect occurred because image-based features are not sufficient to fully represent continuous lip motion. The feature tone recognition of colour images is sensitive to light, speaker skin colour, and camera acquisition quality. However, 3D information has good anti-interference for this kind of disadvantage and is hardly affected. Our proposed facial texture depth information largely compensates for the defect of lip pronunciation in tone recognition caused by environmental problems and complements the image-only lip pronunciation method.

Figure 8 shows the results of the model recognition for adding different noise types. In the experiment, the random Gaussian noise with the mean $\in [0, 10]$ and variance $\in [10, 20]$ was added to simulate the recognition scenario for different photographic definitions, and the gamma algorithm with the gamma interval $\in [1, 8]$ was used to adapt to changes in the lighting due to real-life changes. Adding such dynamic noise can better reflect the robustness of different models in natural scenes and the ubiquitous ability of different frameworks. Unexpectedly, the performance of the proposed SCNet model was much higher than that of the traditional model, which shows that our framework has better application performance in real-world scenarios. Similarly, for the performance of the recognition effect before and after the texture depth information, there was a stable improvement effect of more than 0.5% after the fusion of the depth information, indicating that the fusion depth information is more meaningful for the recognition of the real scene.

4.4. Syllable Recognition. Since tone change occurs in all Chinese pronunciations and the consonant is attached to the vowel, the difficulty of syllable recognition is greater than that of the vowels. To further verify the effectiveness of our

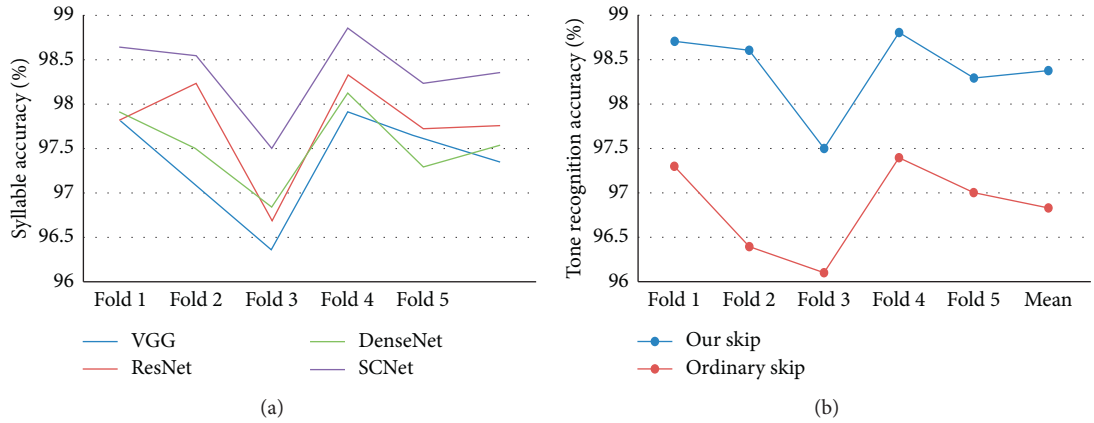


Figure 6: Vowel tone results of different models. The influence of (a) different methods and (b) different skip connections on the accuracy of vowel recognition.

Table 3: Comparison of the network depth, parameters and experimental accuracy of the four different models.

Method	Depth	Params	Accuracy
VGG	11	531.5M	97.35
ResNet	18	46.9M	97.75
DenseNet	121	33.4M	97.528
SCNet	10	10.9M	98.352

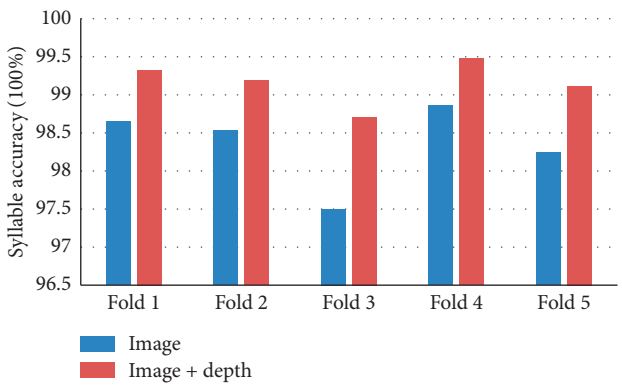


Figure 7: Convergence depth information and comparison of image-only results.

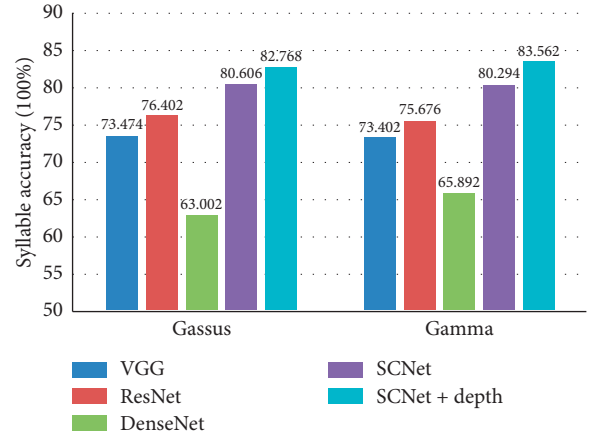


Figure 8: Robustness test of several different dynamic noise models.

proposed SCNet in the recognition of all Chinese tones, we also verified the performance of the model in the recognition of 40 mixed tones based on 5 vowels (/a/, /e/, /i/, /o/, and /u/) and 5 syllables (/ta/, /te/, /ti/, /fo/, and /tu/).

The recognition results are shown in Figure 9. Although the pitch recognition of syllables is more difficult according to the theory, our SCNet model was robust, and a high recognition rate of 97.364% was obtained, indicating that our model had not only a good vowel tone recognition performance but also an excellent Chinese tone recognition performance. Moreover, after adding the depth texture information, the average recognition result of the pitch showed a 0.2% improvement. Since the pronunciation of the syllable is more complicated than that of the vowel and the pronunciation organ is more involved, the facial depth may be relevant. Texture information has a greater impact on the recognition of syllables. A comparison with our previous conjectures indicates that deep texture information has a very clear effect on the recognition of the Chinese lip to assist in lip reading for both consonant and vowel tone recognition.

5. Summary

This work was mainly focused on the difficulty of tone recognition in Chinese lip-reading recognition. In this paper, we designed an efficient lightweight network framework, SCNet, based on a comprehensive and effective lip-

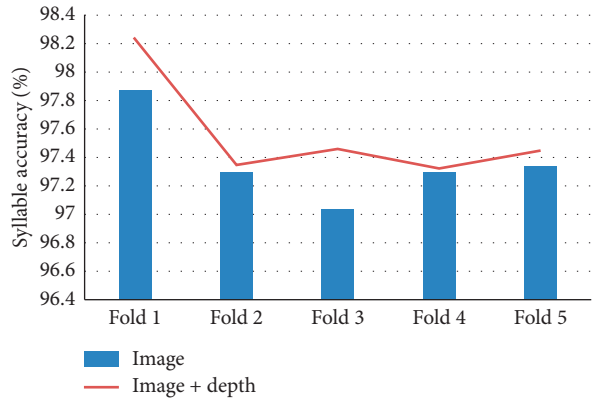


Figure 9: Tone recognition results of vowels and syllables.

reading feature extraction method and verified the effectiveness of our proposed network framework by several experiments. In the study, we carried out an in-depth verification on the proposed framework. Comparison experiments showed that the framework can accurately identify the tones of Chinese pronunciation. In addition, the facial texture depth information and picture information fusion demonstrated the feasibility of facial texture depth information to help the recognition of Chinese tones.

With the wide application of depth cameras on video equipment, lip reading will better assist speech recognition in the future and improve the robustness of speech recognition in different environments. The dataset used in this paper consisted of independent syllables, but the results show that the proposed method is practical and can be effectively applied to future large-scale datasets.

Data Availability

The data used to support the findings of this study are available from the first author upon request.

Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the authorship and/or publication of this article.

Acknowledgments

This study was financially supported by the National Natural Science Foundation of China (grant no. 61977049) and by the Tianjin Key Laboratory of Advanced Networking.

References

[1] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski, "Integration of acoustic and visual speech signals using neural networks," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 65–71, 1989.

[2] G. J. Wolff, K. V. Prasad, D. G. Stork, and M. E. Hennecke, "Lipreading by neural networks: visual preprocessing, learning, and sensory integration," in *Advances in Neural Information Processing Systems*, Morgan Kaufmann Publishers Inc. Burlington, MA, USA, 1993.

[3] P. Scanlon and R. Reilly, "Feature analysis for automatic speech reading," in *Proceedings of the IEEE Fourth Workshop on Multimedia Signal Processing*, Cannes, France, October 2001.

[4] P. S. Aleksic and A. K. Katsaggelos, "Comparison of low- and high-level visual features for audio-visual continuous automatic speech recognition," in *Proceedings of the IEEE International Conference on Acoustics*, Montreal, Canada, May 2004.

[5] I. Matthews, G. Potamianos, C. Neti, J. Luettin, and A. Ascom Systec, "A comparison of model and transform-based visual features for audio-visual lvcsr," in *Proceedings of the IEEE International Conference on Multimedia & Expo*, Tokyo, Japan, August 2001.

[6] M. N. Kaynak, Q. Zhi, A. D. Cheok, K. Sengupta, Z. Jian, and K. C. Chung, "Analysis of lip geometric features for audio-visual speech recognition," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 34, no. 4, pp. 564–570, 2004.

[7] X. Zhang, R. M. Mersereau, and M. A. Clements, "Audio-visual speech recognition by speechreading," in *Proceedings of the International Conference on Digital Signal Processing*, Orlando, FL, USA, May 2002.

[8] J. Bin, Y. Jiachen, L. Zhihan, T. Kun, M. Qinggang, and M. Yan, "Internet cross-media retrieval based on deep learning," *Journal of Visual Communication & Image Representation*, vol. 48, pp. 356–366, 2017.

[9] A. Yargıç and D. Muzaffer, "A lip reading application on MS Kinect camera," in *Proceedings of the IEEE INISTA*, Albena, Bulgaria, June 2013.

[10] K. Palecek, *Extraction of Features for Lip-Reading Using Autoencoders*, Springer, Berlin, Germany, 2014.

[11] A. Rekik, A. Ben-Hamadou, and W. Mahdi, *A New Visual Speech Recognition Approach for RGB-D Cameras*, Springer, Berlin, Germany, 2014.

[12] A. Rekik, A. Ben-Hamadou, and W. Mahdi, "An adaptive approach for lip-reading using image and depth data," *Multimedia Tools and Applications*, vol. 75, no. 14, pp. 8609–8636, 2015.

[13] J. Wang, J. Zhang, H. Kiyoshi, W. Jianguo, and D. Jianwu, "Audio-visual speech recognition integrating 3D lip information obtained from the Kinect," *Multimedia Systems*, vol. 22, no. 3, pp. 315–323, 2016.

[14] J. Yang, B. Jiang, B. Li, K. Tian, and Z. Lv, "A fast image retrieval method designed for network big data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2350–2359, 2017.

[15] J. Wei, F. Yang, J. Zhang, R. Yu, M. Yu, and J. Wang, "Three-dimensional joint geometric-physiologic feature for lip-reading," in *Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence*, Volos, Greece, November 2018.

[16] T. Mallick, P. Goyal, P. P. Das, and A. K. Majumdar, "Facial emotion recognition from kinect data—an appraisal of kinect face tracking library," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, Rome, Italy, February 2016.

[17] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[18] The Difference between Kinect v2 and v1, 2020, https://skarredghost.com/2016/12/02/the-difference-between-kinect-v2-and-v1.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[20] S. R. K. He, X. Zhang, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 770–778, Las Vegas, NV, USA, July 2016.

[21] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, July 2015.

[22] G. Huang, Z. Liu, L. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, July 2017.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 4705982, 11 pages
https://doi.org/10.1155/2020/4705982

Research Article

CNID: Research of Network Intrusion Detection Based on Convolutional Neural Network

Guojie Liu[1,2] **and Jianbiao Zhang**[1,2]

[1]*Beijing University of Technology, Beijing 100124, China*
[2]*Beijing Key Laboratory of Trusted Computing, Beijing 100124, China*

Correspondence should be addressed to Jianbiao Zhang; zjb@bjut.edu.cn

Received 20 December 2019; Revised 5 April 2020; Accepted 29 April 2020; Published 21 May 2020

Academic Editor: Luca Pancioni

Network intrusion detection system can effectively detect network attack behaviour, which is very important to network security. In this paper, a multiclassification network intrusion detection model based on convolutional neural network is proposed, and the algorithm is optimized. First, the data is preprocessed, the original one-dimensional network intrusion data is converted into two-dimensional data, and then the effective features are learned using optimized convolutional neural networks, and, finally, the final test results are produced in conjunction with the Softmax classifier. In this paper, KDD-CUP 99 and NSL-KDD standard network intrusion detection dataset were used to carry out the multiclassification network intrusion detection experiment; the experimental results show that the multiclassification network intrusion detection model proposed in this paper improves the accuracy and check rate, reduces the false positive rate, and also obtains better test results for the detection of unknown attacks.

1. Introduction

Network security is one of the most important security issues facing cloud computing, with frequent cyber attacks and cyber intrusions, such as a DDoS attack by a botnet controlled by the malware Mirai in October 2016 which caused widespread outages on the East Coast of the United States. The ransomware software WannaCry, which broke out in May 2017, exploited system vulnerabilities to poison the computers of hundreds of thousands of users in several countries around the world. In China, the annual losses caused by digital crimes such as pseudo-base-stations and malware extortion amount to tens of billions of yuan. The above examples show that network security not only affects the development of national economy but also affects social stability and national security [1].

Deep learning for network intrusion detection is one of the hot spots in recent academic research. With the enhancement of hardware computing power and the rapid growth of data volume, the development of deep learning has been promoted, so that the practicality and popularity of deep learning have greatly improved [2]. Deep learning is a machine learning technique designed to enable artificial intelligence through experience and data to improve computer systems. Deep learning uses multiple nonlinear feature transformations, that is, processing layers formed by multilayer perception mechanisms, to characterize data learning [3]. Deep learning has been applied to computer vision [4], speech recognition [5], natural language processing [6], biomedicine [7], and malicious code detection [8], as well as many other fields. Since 2015, the research applied to deep learning in network security has gradually emerged, which has attracted wide attention from the academic circles. At present, deep learning is mainly used in the two major areas of network security for malware detection and network intrusion detection, and, compared with traditional machine learning, deep learning improves detection efficiency and reduces false positives. In addition, deep learning algorithms

get rid of the reliance on feature engineering and are able to intelligently identify attack features, helping to identify potential security threats.

Convolutional neural network algorithm (CNN) [9] is an effective algorithm of deep learning; convolutional neural network is designed to process multidimensional array data, and its greatest advantage is to be able to accurately extract the local correlation of features and improve the accuracy of feature extraction. Using convolutional neural network algorithm, combined with mainstream deep learning technology such as Dropout and ADAM and Softmax classifiers, this paper proposes a multiclassification network intrusion detection model based on convolutional neural network and implements the code based on TensorFlow. Finally, the model established in this paper is applied to the standard network intrusion detection dataset such as KDD-CUP 99 and NSL-KDD [10].

The main contributions of this article are as follows:

(i) A multiclass network intrusion detection model based on convolutional neural networks is proposed. This model can automatically and intelligently learn and identify attack features, which is helpful to find potential security threats.

(ii) Multiclass network intrusion detection experiments were performed using KDD-CUP 99 and NSL-KDD standard network intrusion detection datasets. The experimental results show that the network intrusion detection model proposed in this paper improves the accuracy and recall and reduces the false positive rate. The detection of unknown attacks has also achieved better detection results.

(iii) Compared with the common deep learning models such as DNN, LSTM-RNN, GRU-RNN, and DBN, the experimental results show that the network intrusion detection model proposed in this paper has higher accuracy and check rate and lower false positive rate.

The rest of the paper is arranged as follows: Section 2 describes the relevant work, Section 3 introduces the proposed network intrusion detection model, Section 4 discusses the experiments and results, and Section 5 summarizes the paper.

2. Related Works

Network intrusion detection is one of the important security defence means to protect computer systems and networks. Deep learning for network intrusion detection is a hot topic of recent academic research, and many literatures have proposed the successful application of deep learning technology in solving network intrusion detection problems [11, 12]. At present, the experimental results of network intrusion detection using deep learning are mostly distinguished between normal and attack, and there is no distinction between the types of attack. The next focus is on several commonly used deep learning models for multiclassification network intrusion detection: deep neural networks, recursive neural networks, and deep belief networks.

Network intrusion detection is one of the important security defence means to protect computer systems and networks. Deep learning for network intrusion detection is a hot topic of recent academic research, and many literatures have proposed the successful application of deep learning technology in solving network intrusion detection problems [11, 12]. At present, the experimental results of network intrusion detection using deep learning are mostly distinguished between normal and attack, and there is no distinction between the types of attack. The next focus is on several commonly used deep learning models for multiclassification network intrusion detection: deep neural networks, recursive neural networks, and deep belief networks.

2.1. Deep Neural Networks. Deep neural network (DNN) [13] is a neural network model of deep structure, which is widely used in the field of network intrusion detection. Deep neural networks typically consist of input layers, multiple hidden layers, and output layers, as shown in Figure 1. Kim et al. [14]. proposed refined data for the KDD-CUP 99 dataset using a deep neural network model (DR = 99%, FAR = 0.08%). As a method for network attack detection, an accelerated deep neural network model is used together with AEs and Softmax layers for fine-tuning of supervised learning [15]. Evaluate their accelerated deep neural network models using the NSL-KDD dataset, where DR is 97.5% and FAR is 3.5%.

2.2. Recurrent Neural Networks. Recursive neural network (RNN) is another deep structural model widely used in network traffic anomaly detection in recent years. Recursive neural networks mainly include LSTM-RNN [16] and GRU-RNN [17]. Figure 2(a) shows the structure of the LSTM-RNN storage unit. Figure 2(b) shows the structure of the GRU-RNN unit. Ponkarthika and Saraswathy [18] explored the network intrusion detection system based on the LSTM-RNN architecture model. They trained and tested their models on the KDD-CUP 99 dataset with an accuracy of 83%. Kim et al. [19] introduced a long-term-short-term memory recursive neural network (LSTM-RNN) classifier for network intrusion detection on the KDD-CUP 99 dataset, with DR being 98.88% and FAR being 10.04%. Yin et al. [20] proposed a network intrusion detection system based on recursive neural network and applied it to the NSL-KDD dataset (DR = 72.95% percent, FAR = 3.44%). In Kim et al.' [21] study, an integrated method based on LSTM-RNN was proposed, and an ADFA dataset was evaluated, resulting in DR being 90% and FAR being 16%.

2.3. Deep Belief Networks. Deep belief network (DBN) [22] is a layered structure of layer-to-layer restricted Boltzmann machine (RBM). As a well-known deep learning model, it has been widely used in network intrusion detection tasks. Figure 3 describes the typical structure of a DBN. Fiore et al.

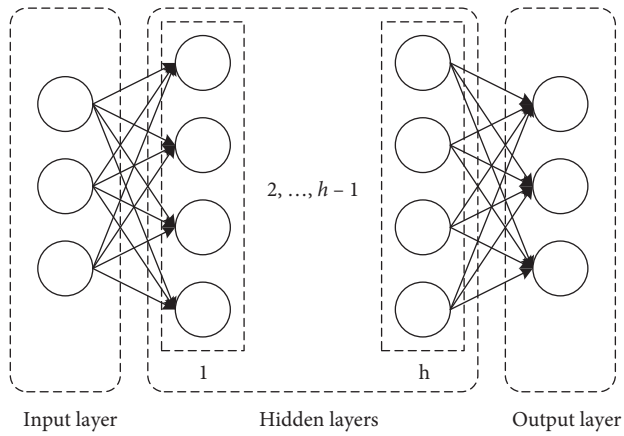


Figure 1: Deep neural networks.

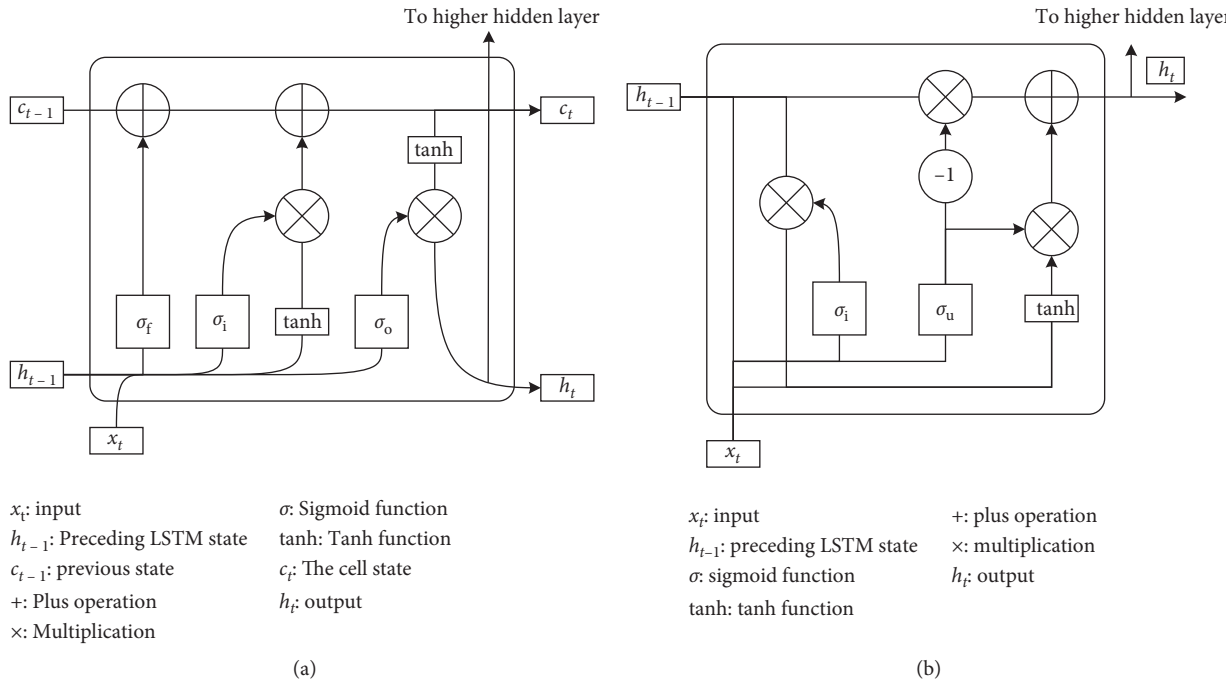


Figure 2: Recurrent neural networks. (a) LSTM-RNN. (b) GRU-RNN.

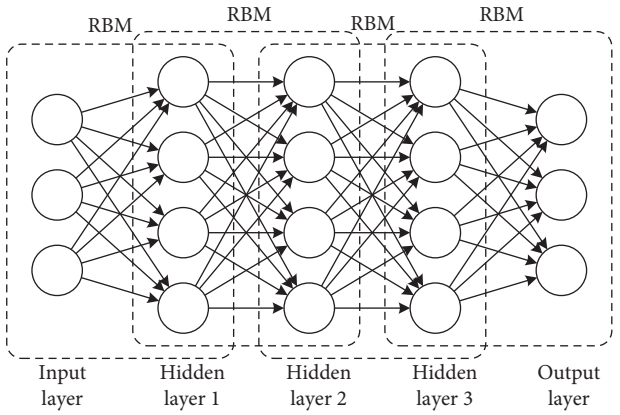


Figure 3: Deep belief networks.

[23] used an RBM-based discriminant model to detect anomalies on 10% of the KDD-CUP 99 dataset. Gao et al. [24] proposed a DBN-based network intrusion detection model and performed experiments on the KDD-CUP 99 dataset (DR = 92.33%, FAR = 0.76%). Alom et al. [25] explored the DBN model, 40% of NSL-KDD ability to detect the abnormal data sets, to obtain a 97.5% detection accuracy. In Liu and Zhang's research [26], the extreme learning machine (ELM) was applied to the learning process of the DBN model and then evaluated using the NSL-KDD dataset (DR = 91.8%). Alrawashdeh and Purdy [27] proposed based on RBM and DBN deep learning method for of KDD-CUP 99 in 10% of the intrusion detection system abnormality, where the DR is 97.9% and FAR is 2.47%.

The comparison of the detection results of the above three deep models is shown in Table 1, which is helpful for researchers to compare the detection results of different deep learning models. We can see from the table that, using the same method, the detection results of the KDD-99 dataset are better than those of the NSL-KDD dataset. This is because the KDD-99 dataset contains a large number of identical data records, while the NSL-KDD dataset removes a large number of duplicate records.

Although the above studies have improved the recognition ability and performance of network intrusion detection samples, there are shortcomings such as overfitting and poor generalization ability in network training, and the detection accuracy and detection efficiency need to be improved. In order to avoid network trained to be merged to enhance the generalization ability, we use convolution neural network combined with the structural characteristics of cross-layer aggregation design concept proposed based on convolutional neural network of multiclassification network intrusion detection model.

Table 1: Summary of test results for different depth models.

Deep learning model	Reference	Dataset	Results		
			Accuracy	DR	Far
DNN	[14]	KDD-99	—	99.00%	0.08%
	[15]	NSL-KDD	—	97.50%	3.50%
LSTM-RNN	[18]	KDD-99	83.00%	—	—
	[19]	KDD-99	—	98.88%	10.04%
	[20]	NSL-KDD	—	72.95%	3.44%
	[21]	ADFA	—	90.00%	16.00%
DBN	[24]	KDD-99	—	92.33%	0.76%
	[25]	NSL-KDD	97.50%	—	—
	[26]	NSL-KDD	—	91.80%	—
	[27]	KDD-99	—	97.90%	2.47%

3. The Proposed Model

The functional composition of the network intrusion detection model based on convolutional neural network is shown in Figure 4, which is composed of three functional modules: the data preprocessing module, the feature self-learning module, and the classifier module. Based on convolutional neural networks, the model is trained by preprocessed original sample datasets and optimized by circular feature extraction and iteration, so that the model can achieve good convergence effect.

3.1. Convolutional Neural Networks. Compared with other machine learning methods, network intrusion detection methods using convolutional neural networks significantly improve the accuracy of classification. As a semisupervised neural network, convolutional neural networks have the ability to abstractly represent low-level intrusion traffic data features as high-level features and outstanding feature learning capabilities, so they have been gradually applied to the field of network intrusion detection in recent years.

Convolutional neural networks are neural networks that use convolution operations in place of ordinary matrix multiplication operations in at least one layer of the network [28], as shown in Figure 5. Convolution is a special linear operation, such as image recognition tasks; each convolution corresponds to the different characteristics of the image; the network's lower-level convolution tends to learn the simple properties of the image, including the edge of the space frequency and colour [29].

The proposed convolutional neural network effectively solves the problem of the explosion of neural network parameters and also ensures the accuracy of classification. The three important core concepts in convolutional neural networks are local perception, parameter sharing, and pooling. Local perception means that neurons in the hidden layer do not need to be connected to all the input pixels, and different hidden layer neurons need only to be connected to a specific area of the input pixel. In convolutional neural networks, local perception is realized by convolutional computations of the convolutional layers, which are realized on input data by convolution nucleus.

3.2. Data Preprocessing. The data preprocessing module characterizes the data, including the numericalization of text features and the standardization of numerical features, and the original intrusion data is usually one-dimensional vector data, which needs to be converted into two-dimensional data similar to the image, so that the convolutional neural network can process it. Using a data-based transformation algorithm, based on retaining all the information of the original sample, the sample is extended with features and normal data is used to fill the extended features, which is to expand the original data sample, thus preserving all the useful information in the original data sample. The expanded features increase the information capacity of the data sample, increase the distance between different categories of data in the sample space, and improve the accuracy of detection to a certain extent.

This data needs to be processed during the data preprocessing phase because each characteristic value of the intrusion detection data has a different range of values and is very different. In this paper, the numerical characteristics of the intrusion data are standardized by using the mainstream z-score standardized method, as shown in formula.

$$x_i' = \frac{x_i - \overline{x}}{v}. \tag{1}$$

In the formula, $\overline{x} = (1/n)\sum_{i-1}^{n} x_i$, $v = \sqrt{(1/n-1)\sum_{i=1}^{n}(x_i - \overline{x})^2}$, n is the total number of samples, x_i is the characteristic value of a dimension of the sample data before standardization, and x_i' is the characteristic value of the dimension corresponding to the sample data after standardization.

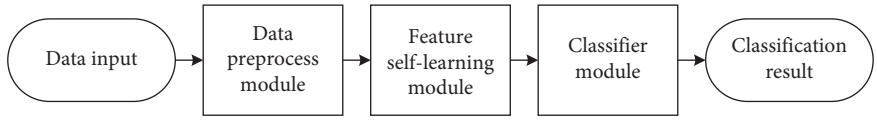


Figure 4: CNN-based network intrusion detection architecture.

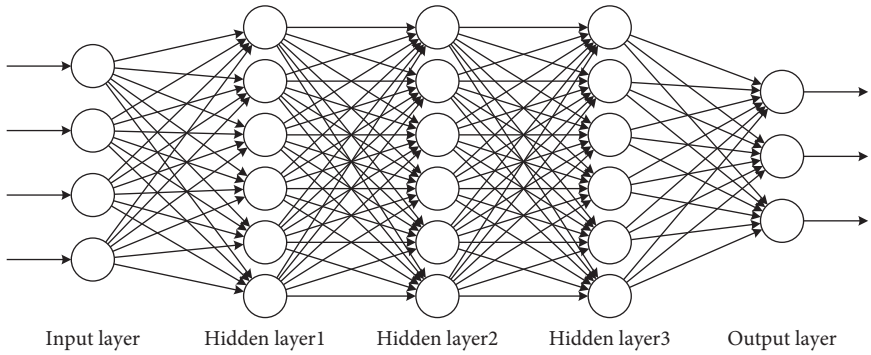


Figure 5: Convolutional neural networks.

3.3. Feature Self-Learning. The main function of the feature self-learning module is to use convolutional neural networks to automatically learn and extract useful features from the original data samples and to learn, map, and generate new features from the original data samples. Lecun et al. [30] systematically expounded convolutional neural networks. The basic structure of the feature self-learning module based on convolutional neural network designed in this paper is shown in Figure 6. The deep learning technology used in the feature self-learning module mainly includes convolution operations and pooling operations, dropout, activation functions, and ADAM optimization algorithms [31].

Convolutional neural network effectively solves the problem of neural network parameter explosion. The three important core concepts of convolutional neural networks are local perception, parameter sharing, and pooling. The local perception of convolutional neural networks is realized through convolutional operations. Convolution is as shown in formula (2) [3]. In the formula, s is the output data also called feature map, x is the input sample data, w is the weight value of the kernel function, b is the offset value, and f is the activation function.

$$s = f(x \times w + b). \tag{2}$$

Convolutional neural networks introduce parameter sharing to further reduce the parameters of the neural network, the essence is that all hidden neurons share a set of weight parameters and bias parameters, and the statistical characteristics based on different parts of the image are usually the same [3]. A set of weight parameters and bias parameters generate a feature map, and the representation capability of a feature map is limited. Therefore, in practical applications, a convolution layer will generate multiple feature maps. The pooling process is mainly to reduce the dimensions of features. The pooling operation generally calculates the average or maximum value of multiple features in a local area. Therefore, the pooling operation in convolutional neural networks is divided into maximum pooling and average pooling. The model proposed in this paper uses the average pooling operation.

The common activation functions of convolutional neural networks are sigmoid, tanh, ReLU [32], and so forth, where tanh is also known as the double-curveting function, and the tanh function will have a good effect when the characteristics differ significantly and will expand the feature effect in the course of the cycle. Therefore, tanh is used as the activation function of the convolutional neural network.

The common method of preventing overfitting include regularization, early stopping, increasing the sample size, dropout, and batch normalization. This paper uses the method of inserting a dropout layer between the feature self-learning module and the classifier to prevent overfitting. The implementation process of dropout is as follows: during model training, some neurons in the neural network are randomly dropped according to the probability p; and, during the test phase, all neurons are online, which can be mitigated by preventing the synergy of certain features overfitting [33, 34]. Using dropout later, each subnetwork is trained neural network of the original, thus, for containing n neural network hidden nodes, $2n$ models can be obtained. When making predictions, the prediction results of all submodels are averaged to improve the model's capacity and generalization ability. Srivastava et al. [33] pointed out that when $p = 0.05$, Dropouthas the best effect, and the network structure generated at this time is therichest.

The ADAM algorithm has been the most widely used first-order optimization algorithm in the field of deep learning in recent years. Kingma and Ba [31] pointed out that the ADAM algorithm includes the advantages of both adaptive gradient algorithms and root mean square propagation algorithmsand has designed different adaptive learning rates for different parameters, so it can converge faster. Network intrusion detection data usually has problems of noise and sparseness. Therefore, this paper chooses

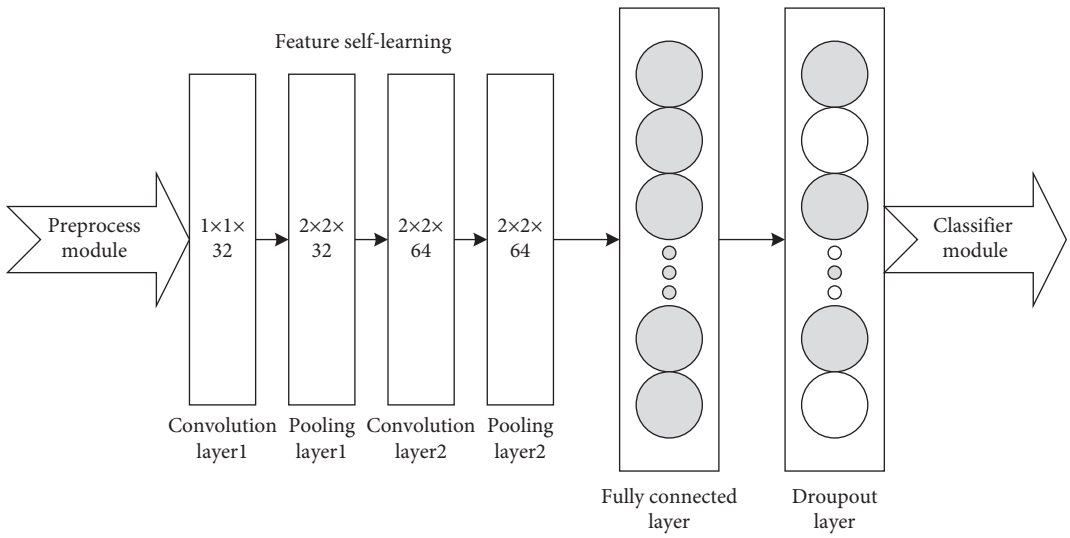


Figure 6: Feature self-learning module.

the ADAM algorithm as the optimization algorithm of the convolutional neural network model.

3.4. Classifier Module. The classifier module gives the final test results based on the characteristics learned by the self-learning module. This article uses the Softmax classifier as a classification module for convolutional neural networks. The Softmax classifier is shown in formula (3). j in the formula is the j weight vector, and $x^{(i)}$ is the i data sample.

$$y_j = \frac{e^{\theta_j x^{(i)}}}{\sum_k \theta_k x^{(i)}}. \tag{3}$$

Common loss functions are mean square error (MSE) and cross entropy error (CEE). The equal square error loss function is mostly used for linear regression and is suitable for predicting values, that is, regression problem model. The cross-entropy error loss function is mostly used for logical regression and is suitable for prediction probability, that is, classification problem. Therefore, the cross-entropy error loss function is used as the loss function of the convolutional neural network model.

4. Experiments and Evaluation

4.1. Experiment Setting. The computer configuration used in the experiment in this paper is as follows: CPU i7-3920XM, 32 Gb of memory, 1 Tb SSD, installed Ubuntu 16.04 operating system with Docker 19.03.5 container virtualization environment, using TensorFlow 1.12.0 as a deep learning framework and Python 3.7 as the programming language.

This paper conducts multiple types of network intrusion classification tests, in which each dataset has a normal (negative) and a mixture of various attack (positive) samples. As shown in Table 2, the number of classes marked in each dataset is different. Therefore, when each model is applied to a specific dataset, a multiclass combined matrix is created to visualize the performance of the model [35]. This confusion

Table 2: KDD-CUP 99 data details.

Data type		Training set	Test set
Normal		97278	60593
Attack	DoS	391458	223298
	Probe	4107	2377
	R2L	1126	5993
	U2R	52	39
Total		**494021**	**292300**

matrix maintains information about actual and predictive classes. Four main results can be extracted from the confusion matrix, namely, true positive (TPs), true negative (TNs), false positive (FPs), and false negative (FNs).

Unlike the two classification schemes, these four results have slightly different meanings in multiclass classification tasks. First, TN is the correct predictor of a normal sample. FP can be calculated by formula (4), where N is the number of attack classes and FP_i is misclassified as the normal number of samples of the i attack class. TP is the sum of all attack samples that are actually marked as their appropriate attack category using formula (5), where TP_i is the exact predictor of the i attack category. Finally, FN is the sum of all attack samples that are misclassified into normal classes. FN can be calculated according to formula (6), where FNi is the number of samples of the attack class misjudged as normal; that is,

$$\mathrm{FP} = \sum_{i=1}^{N} \mathrm{FP}_i, \tag{4}$$

$$\mathrm{TP} = \sum_{i=1}^{N} \mathrm{TP}_i, \tag{5}$$

$$\mathrm{FN} = \sum_{i=1}^{N} \mathrm{FN}_i. \tag{6}$$

These four results are then used to calculate five evaluation indicators, allowing us to evaluate the performance of the model on the dataset. In order to adapt to the terminology definition of the multiclass NIDS system described earlier, some equations have been adjusted. The evaluation indicator definition used and its corresponding equation are shown below.

(i) Accuracy shows the true prediction rate for all test sets; that is,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (7)$$

(i) Precision is the accuracy of the classifier, that is, the rate at which the attack is correctly marked from all samples classified as an attack from the test set; that is,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (8)$$

(i) Recall is the integrity of the classifier, that is, the correct labelled attack rate for all attack samples in the test set. It is also called true positive rate (TPR), detection rate (DR), or sensitivity; that is,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (9)$$

(i) F-Score can be viewed as the harmonic mean of the precision (P) and recall (R) indicators; that is,

$$\text{F} - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (10)$$

(i) The error alert rate (FAR) shows that all normal samples in the test set are misclassified as normal sample rates for any attack category. It is also known as false positive rate (FPR); that is,

$$\text{FAR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (11)$$

4.2. Experiment Datasets. This paper uses the commonly used network intrusion datasets KDD-CUP 99 and NSL-KDD as experimental datasets, which can verify the effect of the network intrusion detection model proposed in this paper. KDD-CUP 99 and NSL-KDD are standard datasets in the field of network intrusion detection and are used by a large number of security research works [36]. In this paper, two experiments are designed for these two datasets.

4.2.1. KDD-CUP 99 Dataset. The KDD-CUP 99 dataset is widely used in the field of network intrusion detection and can be downloaded on the official website [37]. The complete dataset includes approximately 5 million records in the training set and approximately 2 million records in the test set. In fact, only 10% of the KDD-CUP 99 dataset is used for training and testing. There were 494,021 samples in the training data and 292,300 samples in the test data. Each sample is marked as normal or attack recorded. In 10% of the dataset, there are 38 types of attacks. In order to evaluate the effectiveness of the test model testing new attacks that did not appear in the training set, only a sample of 24 types of attacks appeared in the training set. In addition, similar attacks are grouped into one category, forming four main attack categories, namely, DoS, Probe, R2L, and U2R. Details of the KDD-CUP 99 dataset are shown in Table 2.

4.2.2. NSL-KDD Dataset. Tavallae et al. [38] improved and simplified the 10% KDD-CUP 99 dataset in 2009 to form the NSL-KDD dataset. They solved the disadvantage of 10% KDD-CUP 99 in two ways. First, they removed all the extra records from the training and test ingress. Second, they divided the records into different difficulty levels and then selected records from each difficulty level which were inversely proportional to the 10% record percentage in the original KDD-CUP 99 dataset. As a result, NSL-KDD has a reasonable number of records in the training and test set, enabling it to run experiments on a complete set. Although it is no longer a good representation of the real network, it is still considered a benchmark and is widely used in network intrusion detection research. In addition, the NSL-KDD dataset is public on the Internet [39]. Each record in the NSL-KDD dataset consists of 41 characteristics that represent a network connection. The data in the dataset is marked as normal and attacked, and the attack types are divided into four broad categories, with a total of 39 attack types. Twenty-two attacks appear in training and test sets, and 17 attacks appear only in test sets. Details of the NSL-KDD dataset are shown in Table 3.

4.3. Experiment Results

4.3.1. KDD-CUP 99 Experiment. In the data preprocessing stage, the three text features in the dataset are first digitized, each text feature is converted to a corresponding integer value, and then the data sample of 41 features is expanded to 42 dimensions, and the extended feature is transformed using a data transformation algorithm. Fill it and convert it into two-dimensional data. Take 75% of the training data as the training set and 25% as the validation set. A total of 30 iterative trainings were performed. After the training was completed, the test set was used for testing. The experimental results are shown in Table 4. The first 4 columns of the data in Table 4 are the average of the experimental results of the pretrained (w/) and nonpretrained (w/o) stages in [40]; the 5th and 6th columns of the data are the experimental results in [41, 42]. Figure 7 can intuitively compare the evaluation indexes of all models. Experimental results show that the model proposed in this paper obtains 98.02% accuracy and 0.02% false positive rate and has good generalization ability and also has good detection ability for unknown attack types. The accuracy of the detection results is higher than the best detection result in the literature [40–42], 95.00%, and the false alarm rate is lower than the best detection result in the literature [40–42], 0.97%. Figure8(a)intuitively describes the detection rate of each type in KDD-CUP 99. It can be seen from Figure 8(a) that if the amount of data of the attack type

Table 3: NSL-KDD data details.

Data type		Training set	Test set
Normal		67343	9711
Attack	DoS	45927	7458
	Probe	11656	2421
	R2L	995	2887
	U2R	52	67
Total		**125973**	**22544**

Table 4: KDD-CUP 99 experiment result.

Metric	DNN [40]	LSTM-RNN [40]	GRU-RNN [40]	DBN [40]	KNN [41]	CNNA [42]	CNID (this paper) (%)
Accuracy	91.43	93.28	92.41	95.00	94.35%	92.18%	**98.02**
Precision	97.60	97.55	97.29	97.42	93.55%	90.95%	**99.98**
Recall	91.43	93.87	93.02	96.24	—	—	**99.81**
F1-Score	94.41	95.68	95.10	96.82	—	—	**99.89**
FAR	8.61	6.99	7.92	3.24	2.34%	0.97%	**0.02**

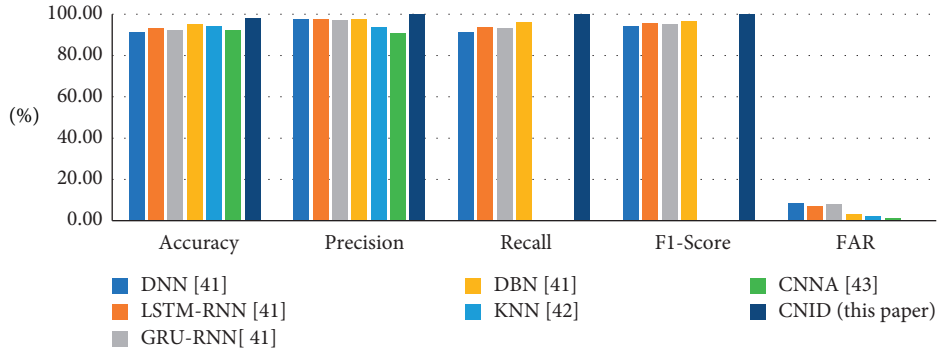


Figure 7: KDD-CUP 99 experiment result.

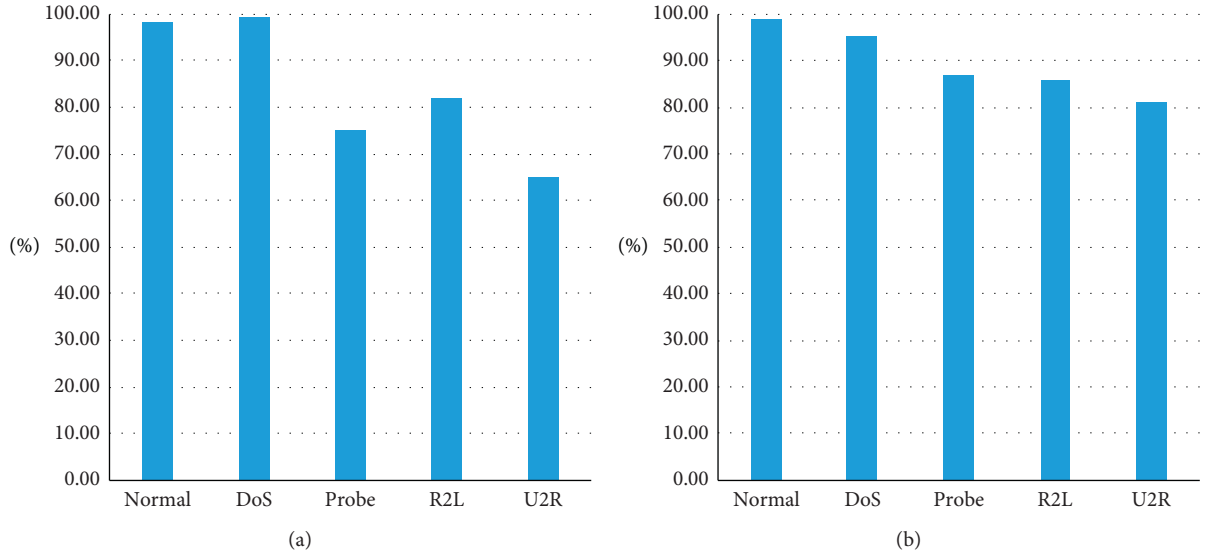


Figure 8: Detection rate for each class in the dataset. (a) KDD-CUP 99. (b) NSL-KDD.

Table 5: NSL-KDD experiment result.

Metric	DNN [40] (%)	LSTM-RNN [40] (%)	GRU-RNN [40] (%)	DBN [40] (%)	ICNN [43]	CNID (this paper) (%)
Accuracy	85.74	90.39	89.58	92.66	91.79%	**97.09**
Precision	96.73	97.52	97.02	97.43	93.65%	**99.98**
Recall	77.19	83.70	82.95	89.56	—	**97.14**
F1-Score	86.05	90.99	89.79	93.33	—	**98.49**
FAR	2.75	2.03	2.58	1.74	2.32%	**0.87**

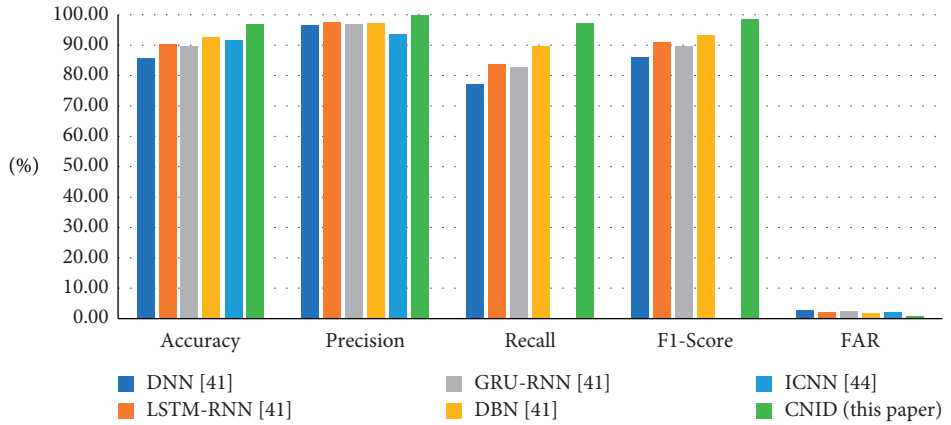


Figure 9: NSL-KDD experiment result.

training set is large, the detection rate is correspondingly higher. In the experiment, it is also found that the attack type data unknown to the training set will also be correctly classified. For example, the upper class of mscan is PROBE, which only appears in the test set. It is an unknown threat to the training set and will be recognized as PROBE in the experiment.

4.3.2. KDD-NSL Experiment. The NSL-KDD dataset was processed using the same processing method as KDD-CUP 99, and a total of 30 iterative trainings were performed. The test results are shown in Table 5. The first 4 columns of the data in Table 5 are the average of the experimental results of the pretraining stage (w/) and the nonpretraining stage (w/o) in [40]; the fifth column of the data represents the experimental results of [43]. Figure 9 can intuitively compare the various evaluation indexes of all models. From the experimental results in Table 5, it can be seen that using the network intrusion detection model proposed in this paper has an accuracy rate of 97.09% and a false alarm rate of 0.87%. The accuracy rate in the detection results is higher than the best detection result in the literature [40, 43], 92.66%, and the false alarm rate is lower than the best detection result in the literature [40, 43], 1.74%. Figure 8(b) intuitively describes the detection rate of each type in NSL-KDD. Similar to the KDD-CUP 99 data and the experimental results, it can be seen from Figure 8(b) that if the amount of data in the attack type training set is large, the detection rate is correspondingly higher. In the experiment, it is also found that the attack type data unknown to the training set will also be correctly classified.

4.3.3. Comparison with Other Related Works. The effectiveness of the model for detecting network intrusion depends on the reasonable setting of its evaluation indicators. The higher the accuracy, accuracy, recall, and F-Score, the lower the FAR value, indicating that the classifier is effective. The accuracy and recall of an ideal classifier reach 1, and the FAR value reaches 0. The experimental results on the KDD-CUP 99 dataset and NSL-KDD dataset are compared with the four deep learning models in the latest literature [40–42] on the same datasets. On KDD-CUP 99 data, the accuracy rate of 98.02% in this paper is better than the accuracy rate of 95.00% in the literature [40–42]. On NSL-KDD data, the accuracy rate of 97.09% in this article is better than the accuracy rate of 92.66% in the literature [40, 43]. Literature [44] proposed a new network intrusion detection model using convolutional neural network (CNN), using CNN to automatically select traffic features from the original data set, and set the cost function weight coefficient according to the number of categories to solve the problem of balance. This model is used for large-scale network intrusion detection, using NSL-KDD dataset, and its accuracy is lower than the model proposed in this paper. Literature [45] proposed a network intrusion detection system based on the convolutional neural network model Lenet-5 and introduced OHE coding and normalization method to process the feature matrix; using KDD-CUP99 dataset, its accuracy is lower than this article's proposed model. Similar literatures [46–48] use convolutional neural networks for intrusion detection in different research areas. The method proposed in this paper has better generalization ability, good detection ability for unknown attack types, and good detection performance in distinguishing normal data and attack data, but

there is room for further improvement in distinguishing different attack types.

5. Conclusions

Network intrusion detection is very important in the field of network security. In recent years, although there has been a lot of research on network intrusion detection, there is very little in-depth research on this issue, especially in multiclass network intrusion detection. In this paper, a multiclass network intrusion detection model based on convolutional neural network is proposed, and the algorithm is optimized. It was tested on a computer configured with 32 Gb of memory, 1 Tb of solid-state drive, and Ubuntu 16.04 operating system and Docker 19.03.5 container virtualization environment. The experiment uses KDD-CUP99 dataset and NSL-KDD dataset and compares the experimental results with the deep learning models of DNN, LSTM-RNN, GRU-RNN, DBN, KNN, ICNN, and so on. The experimental results show that the network intrusion detection model proposed in this paper improves the accuracy and recall, reduces the false positive rate, and obtains better detection results for the detection of unknown attacks.

The accuracy of the model proposed in this paper in multiclass experiments needs to be improved; in particular, the classification results of unknown different attack types still have room for improvement, which needs to be explored in future work. The dataset used in the experiment in this paper is a dataset that has been manually processed and optimized. In the future work, the following datasets will be studied: the newly emerging dataset for network intrusion detection will extract the corresponding data from real network traffic features to verify the method proposed in this article.

Data Availability

The basic data used in this article was downloaded from the Internet. There are two-part datasets: The KDD-CUP 99 is a public dataset that can be downloaded from http://kdd.ics.uci.edu/databases/kddcup99/. The NSL-KDD is a public dataset that can be downloaded from https://github.com/defcom17/NSL_KDD.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was sponsored by the Opening-up Project of National Defense Science and Technology Laboratory of Information Security (No. 2015XXAQ08).

References

[1] J.-Z. Luo, M. Yang, L. Zhen et al., "Architecture and key technologies of cyberspace security," *SCIENTIA SINICA Information*, vol. 46, no. 8, pp. 939–968, 2016.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 2, pp. 2012–2025, 2012.

[5] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[6] X. F. Xi and G. D. Ghou, "A survey on deep learning for natural language processing," *Acta Automatica Sinica*, vol. 42, no. 10, pp. 1445–1465, 2016.

[7] A. Cruz-Roa, J. E. A. Ovalle, A. Madabhushi et al., ""A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection," in *Proceedings of the 9th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 403–410, Springer, Nagoya, Japan, September 2013.

[8] W. Huang and J. W. Stokes, "MtNet: a multi-task neural network for dynamic malware classification," in *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 399–418, Springer, Como, Italy, July 2016.

[9] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[10] M. Tavallaee, Bagherie, W. Lu et al., *NSL-KDD [Z/OL]*, University of New Brunswick, Fredericton, Canada, 2018, http://www.unb.ca/cic/datasets/nsl.html.

[11] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: an overview,," in *Proceedings of the 2016 International Research Conference on Engineering and Technology (2016 IRCET)*, Higher Education Forum, Kuta, Indonesia, January 2016.

[12] R. Vani, "Towards efficient intrusion detection using deep learning techniques: a review," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 6, no. 10, pp. 375–384, 2017.

[13] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] J. Kim, N. Shin, S. Y. Jo et al., "Method of intrusion detection using deep neural network,," in *Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (Big Comp)*, Jeju, South Korea, February 2017.

[15] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in *Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, September 2016, Berlin, Germany.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder–decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[18] M. Ponkarthika and V. R. Saraswathy, "Network intrusion detection using deep neural networks," *Asian Journal of Science and Technology*, vol. 2, no. 2, pp. 665–673, 2018.

[19] J. Kim, J. Kim, H. L. T. Thu et al., "Long short term memory recurrent neural network classifier for intrusion detection," in

Proceedings of the 2016 International Conference on Platform Technology and Service (Plat Con), Jeju, South Korea, January 2016.

[20] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[21] G. Kim, H. Yi, J. Lee et al., "LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," 2016, https://arxiv.org/abs/1611.01726.

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[23] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[24] N. Gao, L. Gao, Q. Gao et al., "An intrusion detection model based on deep belief networks,," in *Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data (CBD)*, Huangshan, China, November 2014.

[25] Z. Alom, V. R. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks,," in *Proceedings of the 2015 National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, USA, June 2015.

[26] Y. Liu and X. Zhang, "Intrusion detection based on IDBM," in *Proceedings of the 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, 14th Intl Conference on Pervasive Intelligence and Computing, 2nd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, Auckland, New Zealand, 2016.

[27] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Anaheim, CA, USA, December 2016.

[28] L. Deng and Y. Dong, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.

[29] RSIP, *Deep Learning and Convolutional Neural Networks: RSIP Vision Blogs [EB/OL]*, RSIP, Jerusalem, Israel, 2016, http://www.rsipvision.com/exploring-deep-learning/.

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[31] D. Kingma and J. Ba, "ADAM: a method for stochastic optimization," 2017, https://arxiv.org/abs/1412.6980v9.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing System*, Curran Associates, Lake Tahoe, CA, USA, pp. 1097–1105, December 2012.

[33] N. Srivastava, G. Hinton, A. Krizhevsky et al., "Dropout: a simple way to prevent neural networks from over fitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] E. Hintong, N. Srivastava, A. Krizhevsky et al., "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.

[35] J. Swets, "Measuring the accuracy of diagnostic systems," *Science*, vol. 240, no. 4857, pp. 1285–1293, 1988.

[36] J. McHugh, "Testing Intrusion detection systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 262–294, 2000.

[37] University of California, *KDD CUP 1999 Data Set*, University of California, Irvine, CA, USA, 1999, http://kdd.ics.uci.edu/databases/kddcup99/.

[38] M. Tavallaee, E. Bagheri, W. Lu et al., "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, July 2009.

[39] NSL-KDD dataset, https://github.com/defcom17/NSL_KDD.

[40] W. Elmasry, A. Akbulut, and A. H. Zaim, "Empirical study on multiclass classification-based network intrusion detection," *Computational Intelligence*, vol. 35, no. 4, pp. 915–954, 2019.

[41] J. Ren, X. Liu, Q. Wang et al., "An multi-level intrusion detection method based on KNN outlier detection and random forests," *Journal of Computer Research and Development*, vol. 56, no. 3, pp. 566–575, 2019.

[42] Y. Liu, S. Liu, X. Zhao et al., "Intrusion detection algorithm based on convolutional neural network," *Transactions of Beijing Institute of Technology*, vol. 37, no. 12, pp. 1271–1275, 2018.

[43] H. Yang and F. Wang, "Network intrusion detection model based on improved convolutional neural network," *Journal of Computer Applications*, vol. 39, no. 9, pp. 2604–2610, 2019.

[44] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.

[45] P. Liu, "An intrusion detection system based on convolutional neural network," in *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering ICCAE*, pp. 62–67, Perth, Australia, February 2019.

[46] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64366–64374, 2019.

[47] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, Article ID 100198, 2020.

[48] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 4163825, 9 pages
https://doi.org/10.1155/2020/4163825

Research Article

Deep Learning-Based Network Security Data Sampling and Anomaly Prediction in Future Network

Lan Liu[1], **Jun Lin**[2], **Pengcheng Wang**[1], **Langzhou Liu**[1] **and Rongfu Zhou**[1]

[1]*Guangdong Polytechnic Normal University School of Electronic and Information Engineering, Guangzhou 510655, Guangdong, China*
[2]*China Electronic Product Reliability and Environmental Testing Research Institute, Guangzhou 510610, Guangdong, China*

Correspondence should be addressed to Jun Lin; linjun@ceprei.com

Received 16 March 2020; Accepted 23 April 2020; Published 17 May 2020

Guest Editor: Jianbiao Zhang

Based on the design idea of future network, this paper analyzes the network security data sampling and anomaly prediction in future network. Through game theory, it is determined that data sampling is performed on some important nodes in the future network. Deep learning methods are used on the selected nodes to collect data and analyze the characteristics of the network data. Then, through offline and real-time analyses, network security abnormal events are predicted in the future network. With the comparison of various algorithms and the adjustment of hyperparameters, the data characteristics and classification algorithms corresponding to different network security attacks are found. We have carried out experiments on the public dataset, and the experiment proves the effectiveness of the method. It can provide reference for the management strategy of the switch node or the host node by the future network controller.

1. Introduction

At present, people attach great importance to the research and application deployment of new technologies and new networks. Scientists are actively exploring the use of technologies such as IPv6, software-defined network (SDN), and 5G to build future networks that meet the requirements of high reliability, low delay, and wide coverage [1]. We need to pay attention to the new features of security events of future network.

5G has brought about massive communications and tens of billions of device access scenarios, all of which require flexible network architecture and high-performance networks. Software defined networking (SDN) is being strongly considered as the next promising networking platform [1, 2]. The logical centralization of network has brought new opportunities and challenges of the field of network security. In future network, the detection and prediction of network data anomaly caused by network malicious attack is an important problem to be solved. Research on the network data sampling strategy and the appropriate anomaly detection model of network security event in the future network has guiding significance for preventing future network. In this paper, we design and simulate a kind of network data sampling strategy of SDN using zero-sum game. After those steps, we can find out some important nodes to protected. And then, we intend to use the method of deep-learning to establish and analyze the network anomaly flow in future network.

The remainder of the paper is organized as follows. Section 2 summarizes the background and related work of deep learning-based network security data sampling and anomaly prediction in future network. In Section 3, we introduce the sampling model of SDN security data and the method of deep learning-based security flow anomaly prediction in detail. Experimental results and comparisons are presented in Section 4. Finally, conclusion is given in Section 5.

2. Related Work

2.1. SDN Network Architecture and Security Data Sampling Model. In recent years, Major mainstream manufacturers

have begun to deploy SDN networks. Many commercial cases have been applied. For example, Google built a B4 [3] network based on SDN to transform its network; Cimorelli [4] propose a distributed load balancing algorithm based on game theory to balance the traffic of the controller cluster. Abraxas of Switzerland adopted Huawei's SDN-based data center network solution to build a virtualized multitenant cloud data center network. In order to provide users with a better experience, Tencent use SDN to achieve differentiated path differentiation calculation and flow control. And, in the development of Internet communication technology in the coming decades, SDN also has broad prospects for development.

SDN is based on the granularity of data flow control, so that it does not understand the internal information of the data stream, which makes SDN vulnerable to attacks by Trojan, worms, spam, etc [5]. In order to ensure the security of the network, it is necessary to detect packets in the future network. Lan [6] propose a dynamic model with a time-varying community network, inspired by research models on the spread of epidemics in complex networks across communities. The results may help to decide the SDN control strategy to defend against network malware and provide a theoretical basis to reduce and prevent network security incidents.

Data packet sampling under limited network resources is necessary to reduce latency, improve the network bandwidth, and ensure network security of future network at the same time. Afek [7] present techniques for traffic sampling and large flows detection in SDN with OpenFlow. They make use of the sampling mechanisms for the development of an efficient method to detect large flows. Tang [8] propose an efficient sampling and classification approach with the two-phase elephant flow detection. They demonstrate their system can provide accurate detection with less sampled packets and short detection time. Aiming at the problem of existing flow statistical sampling in anomaly detection, the authors [9–11] analyze the distortion cause that packet sampling and time domain polymerization lead to flow record time series in theory. They propose different methods to solve it. Result shows that their methods can reduce impact of sampling rate on the signal to noise ratio and improve the performance of the anomaly detection.

Zero-sum game is a concept of game theory and it is a noncooperative game. As its model is relatively simple, a zero-sum game model can be built between the attacker and the defender in network attack and defense [12]. When the attacker attacks successfully, the attacker gains positive scores, while the defender gains negative scores, and the sum of the two is zero. In network attack and defense model, both attack and defense resources are limited. By quantifying network nodes and allocating the corresponding profit value, the game model of attack and defense is established, and we can improve the defense capability, reduce the attack loss, and find a reasonable packet sampling strategy in the future network.

2.2. Deep Learning and Anomaly Detection. As an important subfield of machine learning, deep learning has made breakthroughs in many artificial intelligence fields, such as speech recognition, computer vision, autonomous driving, and natural language processing [13]. Data flow in future network is usually high dimensional and heterogeneous. Deep learning can learn different levels of features from a large number of raw network data streams, and these automatic learning features do not require the domain knowledge of human experts, saving a lot of labor and time costs. We take these learned important features as the input of machine learning algorithm to complete the classification task, which can solve the problem of false alarm rates (FAR) and false positives (FP) of the intrusion detection system (IDS) in the future network security and realize the identification of network traffic [14].

In recent years, some scholars have introduced the method of deep learning into the field of network security [15–18]. They used convolutional neural networks (CNNs) to learn the spatial characteristics of network traffic and used the method of image classification to identify malicious network traffic. Recurrent neural network (RNN) is used to learn the temporal characteristics of network traffic and identify the traffic characteristics to improve the detection rate.

In the deep learning [19–21], CNNs have obtained good performance and wide application in the field of computer vision, and the recognition of handwritten numbers has achieved an extremely low false positive rate on the MNIST test set. The long short-term memory (LSTM) improves the original RNN algorithm [22–24], solves the problem of gradient disappearance or gradient explosion after training of time series modeling, and conducts deep learning through long-term state preservation and forward calculation and uses the back-propagation algorithm to train time series to establish the prediction model [25, 26].

3. Models and Methods

When the network is attacked in future network, we need to have a certain strategy, as soon as it is possible to find the existence of the attack and obtain the attack category and location information. The SDN controller is used to allocate defense resources according to the importance of nodes under the condition of limited defense resources to reduce network losses.

For the important nodes selected from the model, the spatial-temporal characteristics of network traffic are learned by combining CNN and LSTM in deep learning, so as to realize abnormal detection of network traffic. The processing process consists of three parts. First, the advantages of CNN in spatial feature extraction of image processing are utilized, and the spatial feature training is carried out after the network traffic data are processed graphically to form a traffic spatial classification model. Secondly, the traffic vectors processed by CNN are processed in time series, and the time characteristics of the traffic are learned through LSTM to form a traffic time feature recognition model. Then, combining spatial classification model and temporal feature recognition model, the current network traffic is classified. The model is shown in Figure 1.

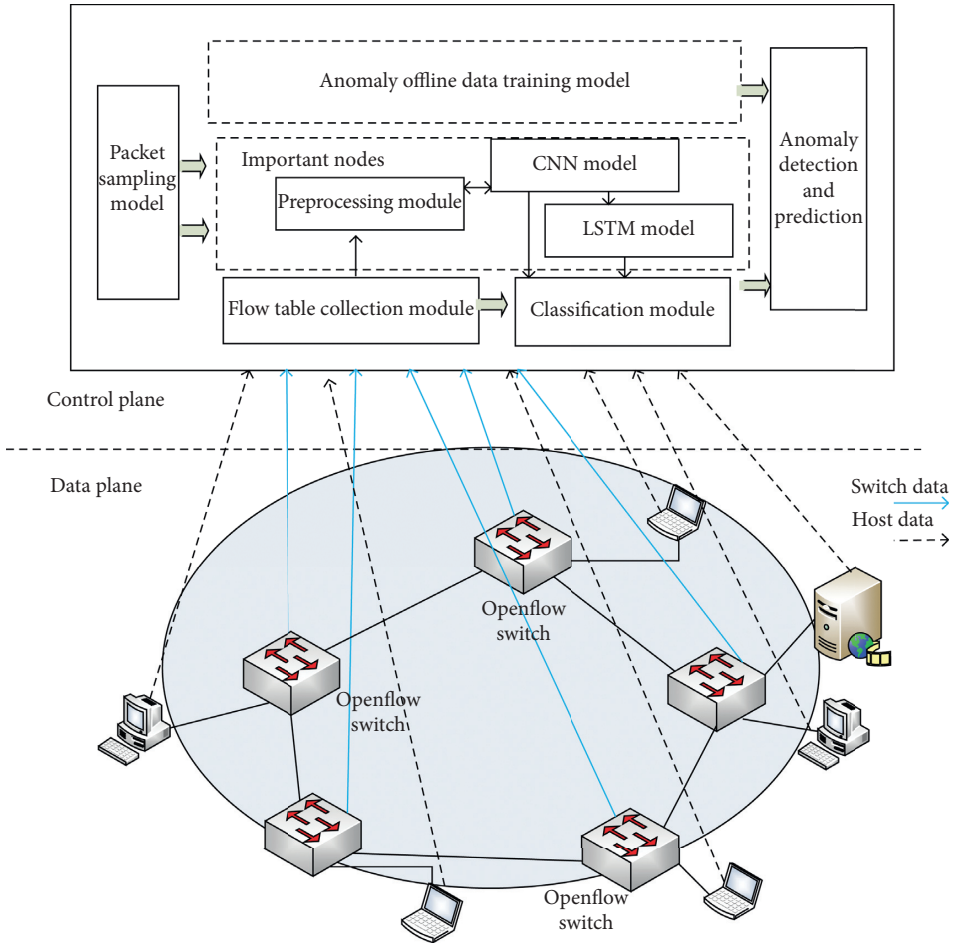


Figure 1: Deep learning-based network security data sampling and anomaly prediction diagram.

3.1. Packet Sampling Model of Future Network. For the sampling and classification of future network security data, the analysis is carried out from the aspects of SDN attack loss calculation method, node importance calculation, attack and defense strategy game model analysis, etc.

3.1.1. Attack Loss Score Calculation. The attacker's behavior can be seen as sending attack packets from a controlled computer to one or more network devices. When the defender nodes checked the attack packet by sampling strategy, the attack will fail and the defender will get a positive score; otherwise, the defender will get a negative score. The attacker sends packets from one network device to one or more network devices; if the packet is not intercepted by the defender, the attack is successful and the score is positive; otherwise, the attack is considered as a failure and the score is negative.

Based on the above background, the following hypotheses are considered:

Hypothesis 1. Under the limited defensive resource constraints, the probability that a defender detects a packet is directly proportional to its importance.

Hypothesis 2. Attackers always pursue maximum revenue, so they will prioritize attacks on network devices of high importance.

In the process of attack and defense game, both the attacker and the defender will use the optimal strategy to maximize their own benefits, and the SDN packet sampling problem will be simulated as a zero-sum game in which both sides of the attack and defense participate.

The SDN network is constructed into an undirected graph, and the set of vertices is V, the graph of the edge set E is recorded as $G = (V, E)$, and the number of vertices and the number of edges of $G = (V, E)$ are, respectively, $|V|$ and $|E|$. Connect two vertices u, and the edges of v are denoted as $e = (u, v)$.

When an attacker launches an attack, the probability of sending an attack packet is proportional to the importance. It is assumed that k packets are extracted for every n packets of the network device of importance x and m packets are included in the n packets. Then, the probability of extracting k out of n packets in n packets is c_{n-m}^{k}/C_{n}^{k}; then this is the probability that no attack packets are detected.

For the attacker, the benefit score is

$$U_a = \frac{c_{n-m}^k}{C_n^k} * x. \quad (1)$$

For the defender, the benefit score is

$$U_X = -\frac{c_{n-m}^k}{C_n^k} * x. \quad (2)$$

3.1.2. Node Importance Calculation. When an attacker successfully attacks the network node v_t, the score that can be obtained is based on the importance $\varphi(v_t)$ corresponding to the node v_t, and the attacker tends to attack the higher-priority nodes in the network to cause greater impact on the network. The network node value is quantified according to the importance of the network node, and the higher value is given to the more important network nodes. The nodes in the network are divided into switch nodes $S_k \in$ S, and the host nodes $H_k \in$ H, S, and H are included in N. For the normal operation of the network, the importance of the switch node (Switching device) is equal to the sum of importance value of all the host nodes (Terminal devices) connected to it. The importance of different switches in future network may be different, such as the core switch is more important than the edge switch; there is no difference between hosts. In summary, Theorem 1 and Theorem 2 are proposed.

Theorem 1. *The importance value of each S node is divided into direct importance value and indirect importance value.*

Theorem 2. *The direct importance value of a S node is equal to the sum importance value of the H nodes which it is connected, and the indirect importance value is equal to the direct importance value of the S node which it is connected.*

Theorem 3. *The importance value of each S node may be different, and the importance value of each H node is equal.*

According to Theorem 1–3, the importance value of the S node and the H node is divided. The importance SI value of the switch node is often higher than the importance HI value of the host node. The specific values can be used to represent different network nodes according to different network scenarios, for example, we may set HI value as 1. When SI value and HI value are set, attention is paid to the size relationship between them, that is, the value of S node $SI = \sum_{i=1}^{n} HI$, where n is the H node connected to the S node.

According to Theorem 2, assuming that the importance value of each H node is 1, then the direct importance value of a S node is equal to the sum of all the H nodes connected to it. And, the indirect importance value the S node is equal to the sum of all S nodes connected to it. We add the two values when we calculate the importance value of S node.

3.1.3. Zero-Sum Game Model of Attack and Defense Strategies. For an attacker, there are two main attack strategies:

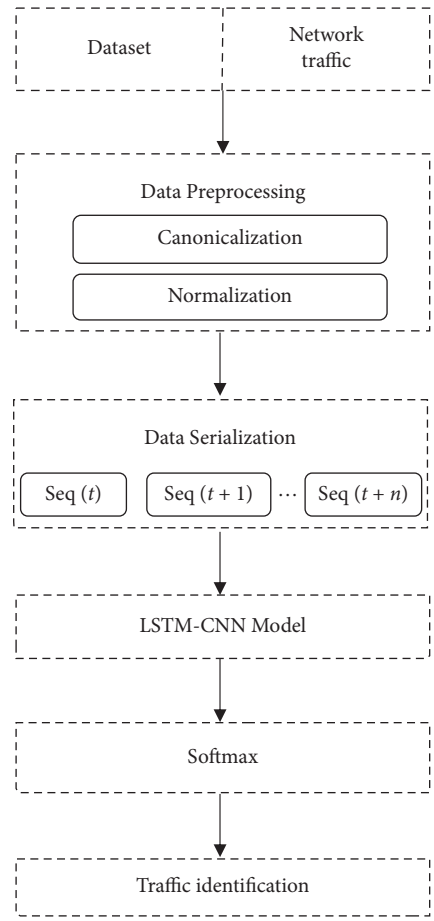


Figure 2: The network anomaly detection model based on deep learning in future network.

Table 1: Network topology.

	Topology 1	Topology 2	Topology 3	Topology 4
Number of switches	3	1	5	3
Number of hosts	4	4	5	5
Number of links	7	4	9	7
Topology	Tree type	Star type	Line type	Hybrid type

(1) Sends attack packets to the defender network device on average if the importance of network nodes is unknown

(2) Sends attack packets to the defender network device by its proportion if the importance of network nodes is known.

Suppose an attacker uses attack strategy 1 to distribute attack packets evenly to n network devices, this n is exactly equal to the number of defender network devices. It is assumed that when an attacker uses an attack strategy, it may be randomly assigned to attack a network device of high

Table 2: Defender's detection success rate of attack and score of the attacker.

Experimental method		Defensive detection success rate	Attacker score
Topology 1	Attack strategy 1 vs. defensive strategy 1	0.41	7.2
	Attack strategy 1 vs. defensive strategy 2	0.75	6.8
	Attack strategy 2 vs. defensive strategy 1	0.75	6.3
	Attack strategy 2 vs. defensive strategy 2	0.98	5.7
Topology 2	Attack strategy 1 vs. defensive strategy 1	0.62	3.1
	Attack strategy 1 vs. defensive strategy 2	0.96	2.1
	Attack strategy 2 vs. defensive strategy 1	0.96	2.2
	Attack strategy 2 vs. defensive strategy 2	0.99	2.4
Topology 3	Attack strategy 1 vs. defensive strategy 1	0.19	16.1
	Attack strategy 1 vs. defensive strategy 2	0.29	17.1
	Attack strategy 2 vs. defensive strategy 1	0.29	17.1
	Attack strategy 2 vs. defensive strategy 2	0.39	13.7
Topology 4	Attack strategy 1 vs. defensive strategy 1	0.41	8.5
	Attack strategy 1 vs. defensive strategy 2	0.75	7.3
	Attack strategy 2 vs. defensive strategy 1	0.75	7.3
	Attack strategy 2 vs. defensive strategy 2	0.98	6.1

importance value of defender network or may be randomly assigned to attack a network device of low importance value defender network.

When defenders deal with attackers, there are two main defense strategies:

(1) The probability of network device packet detection is equal

(2) The probability of network device packet detection is directly proportional to its importance value

3.2. Network Anomaly Detection Based on Deep Learning. After describing the sampling model in Section 3.1, we find the secure nodes that need sampling in future network. On these nodes, we use the network traffic anomaly detection method based on deep learning and combine CNN and LSTM to detect and classify network security data. The spatial-temporal characteristics of network traffic can be obtained through training, which has great potential to improve the overall performance of network traffic detection technology in future network. The algorithms analyze the possible security events and submit them to the controller of SDN for further analysis and optimization of the whole network.

The network anomaly detection model based on deep learning in future network is shown in Figure 2.

For the data on the important nodes found by the sampling model, the data are firstly preprocessed, including numerical coding and normalization. Then, the preprocessed data were input into the LSTM-CNN model, and the spatial and time feature learning of network traffic were carried out. Finally, the two kinds of neural networks were combined, and the output was classified by Softmax and the attack events were classified.

The experimental process is as follows:

Step 1 open IDS datasets or simulated attacks are used as training datasets, and real-time network traffic is collected as test data

Network traffic types in CICIDS2017	
Benign	2273097
DoS hulk	231073
Portscan	158930
DDoS	128027
DoS goldeneye	10293
FTP patator	7938
SSH patator	5897
DoS slowloris	5796
DoS slowhttptest	5499
Bot	1966
Web attack brute force	1507
Web attack XSS	652
Infiltration	36
Web attack sql injection	21
Heartbleed	11
Total	*2830743*

Figure 3: Dataset statistics.

Step 2 data preprocessing is carried out, and the flow data after feature extraction is numerically coded and feature normalized

Step 3 the preprocessed data were coded with one-hot coding, the matrix was converted into $m \times m$ traffic images, and the image data were classified through the CNN neural network

Step 4 the preprocessed data were divided into time series and trained by LSTM neural network to obtain the abnormal flow probability of the next period.

Finally, the two training models are combined to predict and identify the current network traffic and realize the real-

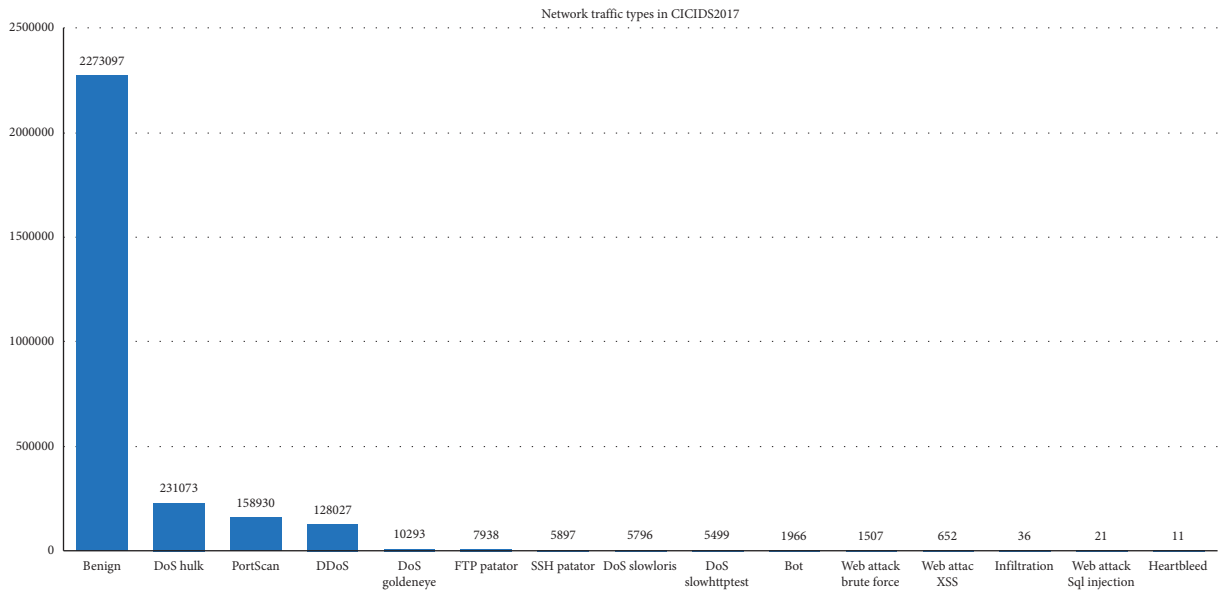


Figure 4: Network traffic types in CICIDS2017.

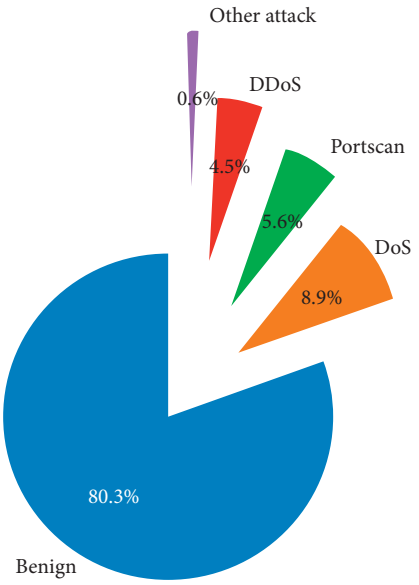


Figure 5: Complete CICIDS2017 dataset distribution.

time automatic monitoring of future network traffic anomaly detection function.

4. Experimental Results and Analysis

4.1. Experimental Method of Packet Sampling. In order to verify the difference of sampling strategy described in 3.1, Matlab and graph theory were used to build the model and construct the network topology and node sampling function. Four kinds of topology structure and four kinds of attack and defense strategies were used to carry out 16 groups of simulation, each group of simulation was repeated 10 times, and then the average value of each group of data was calculated. Under different combinations of attack strategies and topologies, SDN packet sampling strategy based on zero-sum game is compared with random sampling strategy. The experimental topology is shown in Table 1. The

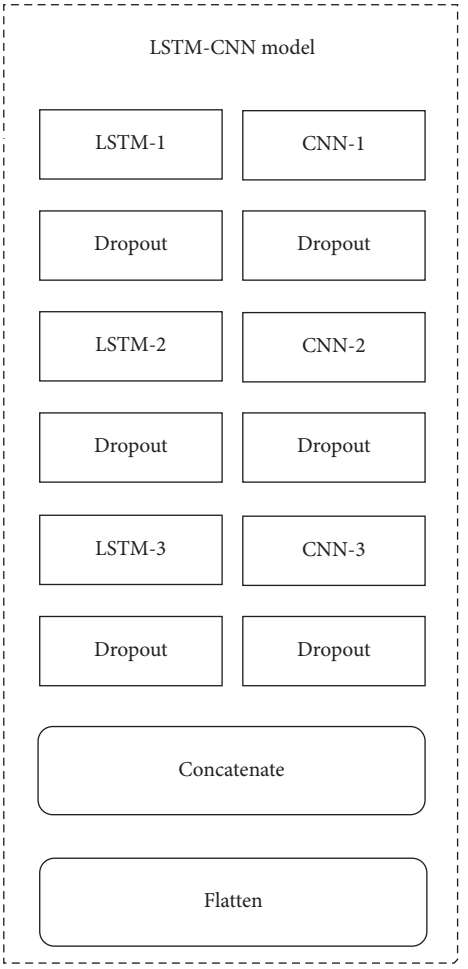


Figure 6: LSTM-CNN model.

detection success rate of defenders against attacks and the scores of attackers are shown in Table 2.

In the experiment, there is only one attack host and the attack data sent to the network every second is 20 per second. Each network device can receive 20 packets per second. The total number of sampling nodes per second for network devices in the entire topology is 20, and the attack score is reserved to decimal.

Experimental data show that, compared with attack strategy 1, attack strategy 2 can improve the defense success rate and reduce the attack score, which indicates that, in network attack, increasing the power of sending attack packets to the target host will make the target host easy to detect the attack and take active defense. Compared with defensive strategy 1, defensive strategy 2 can improve the detection success rate and reduce the attack score. The reason is that SDN packet sampling strategy based on zero-sum game tends to protect important nodes, so this strategy is effective.

4.2. Datasets and Experimental Methods of Anomaly Detection. In this section, the mentioned network traffic anomaly detection method is tested. All models of this method are designed and verified on the Google Colab platform, and the TPU accelerator provided by Google Colab is used. The framework of deep learning selects Keras based on TensorFlow 2.1 and CICIDS2017 [27] as the dataset for anomaly detection.

We use CICIDS2017 as the dataset for anomaly detection, published by the Canadian Institute for Cybersecurity. CICIDS2017 is a dataset for simulating real attacks and contains the necessary features for common network events. Among them, the traffic data are captured by packet and extracted by CICFlowMeter. Each data contains more than 80 dimensions of network traffic characteristics.

Before the experiment, we first conducted data statistics on CICIDS2017, and its traffic types is shown in Figure 3 and its traffic distribution is shown in Figure 4. It can be seen that there are 15 types of traffic, including normal traffic and 14 types of attack traffic.

Then, we carried out numerical normalization and traffic label coding on the dataset, and the numerical normalization was mapped by the MinMax method. In the process of traffic label coding, according to the traffic distribution characteristics of CICIDS2017, we can see that the normal traffic occupies more than 80%, and the attack traffic is mainly DOS, PortScan, and DDoS. Therefore, we divided 15 types of traffic into Benign, DOS, DDoS, PortScan, and other attacks, as shown in Figure 5; it make our experimental training and statistics more convenient.

We input the serialized preprocessed data into LSTM and CNN neural network, where LSTM predicts the temporal characteristics of the traffic sequence and CNN learns the spatial characteristics of the network traffic sequence. This experiment of deep learning framework using Keras LSTM and neural network (CNN) in the model structure as shown in Figure 6, including CNN and LSTM three-layer neural network, is adopted, and each layer neural network

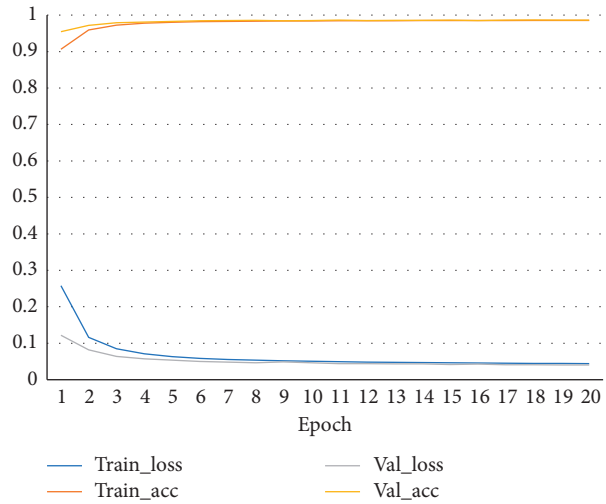


Figure 7: Loss-accuracy change rate.

Table 3: Classification report.

Types	Precision	Recall	F1-score
BENIGN	1.00	0.98	0.99
DoS	0.99	1.00	0.99
PortScan	0.95	1.00	0.98
DDoS	1.00	1.00	1.00
Other attacks	0.89	0.94	0.92
Total	0.966	0.984	0.976

using the dropout discard part features, to prevent overfitting, on the fourth floor, LSTM is combined with CNN through the flatten layer for dimension reduction and finally the output was sorted through the softmax layer.

After experimental tests, as the number of epochs increased, we obtained the variation trend of loss value and accuracy value in the traffic classification of the LSTM-CNN model. It can be seen from Figure 7 that when the epoch reached 7.5 times, the performance of this model tended to be stable. The loss value was 0.0441, and the accuracy value was 0.9853 when the epoch was 20 times.

After training the data, we tested the model and the accuracy reached 0.966, with a better recognition rate of network attacks. Table 3 shows the comparison of detection rates between normal traffic and attack traffic using CNN-LSTM. The evaluation criteria include precision, F1Score, and recall.

Through the experiment, DDOS can achieve 100% successful detection, and the average F1-score of normal traffic and other attack traffic can reach 97.6%, indicating that this method has excellent performance in the future network anomaly detection.

5. Conclusion

The global view and centralized control of the future network make the network traffic control in the big data environment convenient and effective, but most of the anomaly traffic detection often needs to be detected through a large

number of data samples and the number of abnormal traffic explosive growth, resulting in a decline in detection efficiency.

This paper proposes a sampling and classification prediction model of anomaly traffic of future networks based on game theory and deep learning. The defense performance of network is improved by protecting important nodes. The experimental platform has been built, and we also use public datasets to test our method. The results show that the sampling strategy of SDN packets based on zero-sum game and the method of deep learning analysis for the selected important nodes are effective. In the future, further research can be carried out on the game model, different types of deep learning methods, and super-parameter selection.

Data Availability

The data used to support the findings of this study are available at https://www.unb.ca/cic/datasets/ids-2017.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61972104), the Special Project for Research and Development in key areas of Guangdong Province (2019B010121001), and the Special Fund for Science and Technology Innovation Strategy of Guangdong Province (2020a0332).

References

[1] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: a survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 325–346, 2017.

[2] J. Ren, A. Hussain, H. zhao et al., "Advances in brain inspired cognitive systems," in *International Conference on Brain Inspired Cognitive Systems*, vol. 11691 of Lecture Notes in Computer Science, Springer, Berlin, Germany, 2020.

[3] S. Jain, A. Kumar, S. Mandal et al., "B4," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[4] F. Cimorelli, F. D. Priscoli, A. Pietrabissa, L. R. Celsi, V. Suraci, and L. Zuccaro, "A distributed load balancing algorithm for the control plane in software defined networking," in *Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED)*, pp. 1033–1040, Athens, Greece, June 2016.

[5] W. Zhang, X. Wang, S. Zhang, and M. Huang, "SDN data packet sampling strategy based on security game," *Journal of Zhengzhou University (Science Edition)*, vol. 50, no. 1, pp. 15–19, 2018.

[6] L. Lan, K. L. K. Ryan, R. Guangming, and X. Xu, "Malware propagation and prevention model for time-varying community networks within software defined networks," *Security and Communication Networks*, vol. 2017, Article ID 2910310, 8 pages, 2017.

[7] Y. Afek, S. A. Bremler-Barr, and L. SchiffLandau Feibish, "Sampling and large flow detection in SDN," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 345-346, 2015.

[8] F. Tang, L. Li, L. Barolli, and C. Tang, "An efficient sampling and classification approach for flow detection in SDN-based big data centers," in *Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, Taipei, Taiwan, March 2017.

[9] J. Zhao, J. Sun, Y. Zhai, Y. Ding, C. Wu, and M. Hu, "A novel clustering-based sampling approach for minimum sample set in big data environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 2, 2018.

[10] H. Grushka-Cohen, O. Biller, O. Sofer, L. Rokach, and B. Shapira, "Simulating user activity for assessing effect of sampling on DB activity monitoring anomaly detection," in *Policy-Based Autonomic Data Governance*, Springer, Berlin, Germany, 2019.

[11] Y. Yong-Qiang, S. Chao, and Z. Jian-Hui, "Research on impact of packet sampling on anomaly detection and its elimination method," *Computer Engineering*, vol. 39, no. 1, pp. 131–135, 2013.

[12] S. D. Bopardikar, A. Borri, J. P. Hespanha, M. Prandini, and M. D. Di Benedetto, "Randomized sampling for large zero-sum games," *Automatica*, vol. 49, no. 5, pp. 1184–1194, 2013.

[13] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "Intelligent approach to build a deep neural network based IDS for cloud environment using combination of machine learning algorithms," *Computers & Security*, vol. 86, pp. 291–317, 2019.

[14] X. Shao, M. Zhang, and J. Meng, "Data stream clustering and outlier detection algorithm based on shared nearest neighbor density," in *Proceedings of the 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, Xiamen, China, January 2018.

[15] W. Huang and J. W. Stokes, "MtNet: a multi-task neural network for dynamic malware classification," *Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, Berlin , Germany, pp. 399–418, 2016.

[16] T. Bolukbasi, J. Wang, and O. Dekel, "Adaptive neural networks for efficient inference," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 527–536, Sydney, Australia, August 2017.

[17] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[18] A. Wankhade and K. Chandrasekaran, "Distributed-intrusion detection system using combination of ant colony optimization (ACO) and support vector machine (SVM)," in *Proceedings of the 2016 International Conference on Micro-Electronics and Telecommunication Engineering, ICMETE 2016*, pp. 646–651, Ghaziabad, India, September 2016.

[19] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, Madeira, Portugal, January 2018.

[20] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: a survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, 2017.

[21] L. Lan and L. Jun, "Some special issues of network security monitoring on big data environments," in *Proceedings of the 2013 IEEE 11th International Conference on*

Dependable, Autonomic and Secure Computing, Chengdu, China, December 2013.

[22] L. P. Dias, J. J. F. Cerqueira, K. D. R. Assis, and R. C. Almeida, "Using artificial neural network in intrusion detection systems to computer networks," in *Proceedings of the 2017 9th Computer Science and Electronic Engineering (CEEC)*, pp. 145–150, Colchester, UK, September 2017.

[23] Hu W., Tan Y., Black-box Attacks against RNN Based Malware Detection algorithms, 2017, https://arxiv.org/pdf/1705.08131.

[24] Grosse K., Papernot N., Manoharan P., Adversarial Perturbations against Deep Neural Networks for Malware classification, 2016, https://arxiv.org/pdf/1606.04435.

[25] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[26] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the 2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, Da Nang, Vietnam, January 2017.

[27] A. Boukhamla and J. C. Gaviro, "CICIDS2017 dataset: performance improvements and validation as a robust intrusion detection system testbed. International Journal of Information and Computer Security," vol. 9, 2018.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 4565769, 10 pages
https://doi.org/10.1155/2020/4565769

Research Article

GSPSO-LRF-ELM: Grid Search and Particle Swarm Optimization-Based Local Receptive Field-Enabled Extreme Learning Machine for Surface Defects Detection and Classification on the Magnetic Tiles

Jun Xie,[1] Jin Zhang,[1] Fengmei Liang,[1] Yunyun Yang ,[2] Xinying Xu ,[2] and Junjie Dong[1]

[1]*College of Information and Computer Science, Taiyuan University of Technology, Jinzhong 030600, China*
[2]*College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan 030024, China*

Correspondence should be addressed to Xinying Xu; xuxinying@tyut.edu.cn

Received 8 January 2020; Revised 14 April 2020; Accepted 30 April 2020; Published 15 May 2020

Guest Editor: Zheng Wang

Machine vision-based surface defect detection and classification have always been the hot research topics in Artificial Intelligence. However, existing work focuses mainly on the detection rather than the classification. In this article, we propose GSPSO-LRF-ELM that is the grid search (GS) and the particle swarm optimization- (PSO-) based local receptive field-enabled extreme learning machine (ELM-LRF) for the detection and classification of the surface defects on the magnetic tiles. In the ELM-LRF classifier, the balance parameter C and the number of feature maps K via the GS algorithm and the initial weight A^{init} via the PSO algorithm are optimized to improve the performance of the classifier. The images used in the experiments are from the dataset collected by Institute of Automation, Chinese Academy of Sciences. The experiment results show that the proposed algorithm can achieve 96.36% accuracy of the classification, which has significantly outperformed several state-of-the-art approaches.

1. Introduction

The magnetic tile is an important component of the motor, whose surface defects directly affect the performance and the life of the motor. Therefore, defective surfaces on the magnetic tiles need be detected and analyzed during the production process [1]. The common types of surface defects of the magnetic tiles mainly include "break," "crack," "fray," "uneven," and "blowhole" [2], which are shown in Figure 1. These surface defects are used to be inspected by humans, which has inevitably suffered from several downsides such as the low detection efficiency, the poor detection consistency, and the high laboring cost. As a result, the automatic detection of such surface defects using the visual inspection and image processing attracts more and more attention [3]. However, conventional automatic detection methods either have low detection accuracy or fail to classify the detected defects, which have severely affected the following process of industrial production [4].

In recent years, a number of approaches based on machine vision have been proposed for improving the detection and classification of surface defects on magnetic tiles. Valavanisa and Kosmopoulos [5] proposed a method which uses the geometric and the texture features to detect and classify defects in the weld radiographs. This method improved the detection speed, but the extracted features were too complicated. Li et al. [6] used the fast discrete curvelet transform (FDCT) and texture analysis for the detection of "cracks" in magnetic tiles longer than 0.8 mm, but it could only detect single "cracks" whilst other types of defects were not considered. Yang et al. [7] proposed to use non-subsampled shearlet transform for surface defects detection of the magnetic tiles, which could effectively remove "uneven" background, grinding texture and noise interference during defect detection rather than any other kinds of defects. He et al. [8] proposed a framework for the detection of steel surface defects, classification priority network (CPN), and a new classification network, multigroup convolutional

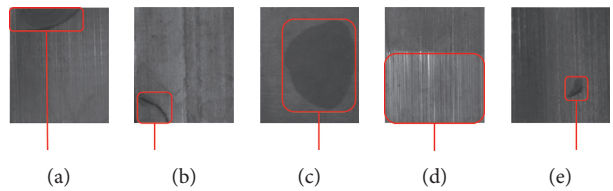


Figure 1: Five common types of the surface defects on magnetic tiles: (a) break. (b) Crack. (c) Fray. (d) Uneven. (e) Blowhole.

neural network (MG-CNN). The framework has better classification performance, but there is the problem that the classification results are unstable in early training.

In summary, most of the existing approaches for surface defect detection of the magnetic tiles can detect a single type of surface defects and often have relatively low detection accuracy. After detection of the possible defects, their types are not further classified, which is not conducive to finding the cause of the defect for improving the subsequent production process. In this article, we aim to solve the aforementioned problems. We use the GS method to obtain the optimal parameter combination (C, K) more accurately and divide the GS method into two parts: rough optimization and fine optimization; PSO algorithm is proposed to optimize the initial weight A^{init} of ELM-LRF and further classify the defect categories. The main contributions of our work can be highlighted as follows.

(1) Because ELM-LRF has poor initial weights stability, we use the particle swarm optimization (PSO) to optimize the initial weights of ELM-LRF, which improves the classification accuracy of the classifier

(2) In order to improve the performance of the classifier, the method of grid search (GS) rough optimization and fine optimization is used to optimize the balance parameter C and the number of feature maps K in ELM-LRF

(3) Using the optimized ELM-LRF to classify the surface defect categories in the detected images and compared with some advanced multicategory classification algorithms, our proposed method has higher classification accuracy

The remainder of this article is organized as follows. In Section 2, the related work and technical background of ELM-LRF are introduced. Section 3 presents the proposed GSPSO-LRF-ELM algorithm, that is, the grid search and the particle swarm optimization optimized ELM-LRF. The experiments results and analysis are given in Section 4. Finally, Section 5 concludes the article along with future prospects for the next phase of research.

2. Related Work and Technical Background

There are six common types of the magnetic tiles, which are "break," "crack," "fray," "uneven" "blowhole," and "free," on the surface of the magnetic tile. In order to further classify the test results, it is necessary to extract the feature information of each category as the input of classifier to realize the defect classification.

Vision-based defect detection and classification systems have great advantages for industrial production, which have promoted a large number of related work in relevant fields [9]. The general workflow of the system is illustrated in Figure 2.

2.1. Extraction of Region of Interest (ROI). Firstly, the industrial camera and the video acquisition equipment are used to obtain the image of the magnetic tile. After background removal, the region of interest (ROI) is extracted as the input for next stage of processing.

2.2. Preprocessing and Image Segmentation. Image preprocessing includes enhancement, sharpening, and denoising of images. For image segmentation or detection of defects, several methods can be used. Commonly used image segmentation methods include region growing [10], mean iterative segmentation [11], maximum entropy segmentation [12], and Otsu [13]. In this article, considering that the surface of the magnetic tile image is dim and the image is complicated, an entropy weighted automatic threshold Otsu maximum interclass variance image segmentation method is chosen for the image segmentation [14].

2.3. Image Feature Extraction and Defection Classification. Some commonly used features include color, shape, texture, and spatial relationship. ELM-LRF with convolution layer and pooling layer can realize feature self-extraction of input image. For image classification, some classical algorithms include support vector machine (SVM), artificial neural network (ANN), Bayesian classification (BC), and K-nearest neighbor (K-NN). Zhou et al. [15] extracted features as input to SVM for classification of automobile surface defects. Kumar et al. [16] used the gray level cooccurrence matrix and the texture shape geometry as the features of the detected weld image, followed by using the ANN for detection and classification of the defects. Yapi et al. [17] trained BC to distinguish the defect-free fabrics from the defect ones and achieved good detection results. Cetiner et al. [18] used features obtained from wavelet distance as the input of K-NN to further classify wood materials.

With the development of deep learning, convolutional neural network (CNN) has been successfully applied in many different applications. Tao et al. [19] classified the detected metal defects through a compact CNN, which satisfies the robustness and accuracy of detection. Wang et al. [20] realized the function of automatically extracting

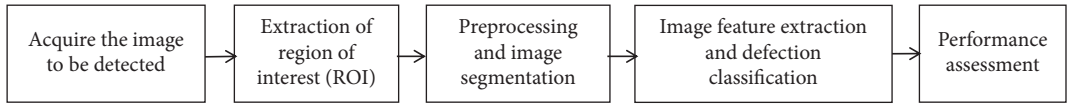


Figure 2: Workflow of machine vision-based defect detection and classification.

image features in the case of less prior knowledge of the defect detection images.

Due to the advantages of CNN sharing convolution kernel and feature self-extraction, it has been widely used in current classification research. But CNN adopts traditional BP training, which has the disadvantages of slow convergence and being easy to fall into local optimum. In 2015, Huang et al. [21], inspired by CNN and the extreme learning machine (ELM), proposed a local receptive field-based extreme learning machine [22] (ELM-LRF). In this article, the ELM-LRF algorithm is used to detect and classify the surface defects of the magnetic tiles, where the grid search (GS) and particle swarm optimization (PSO) algorithm are used to optimize the parameters and weight in the ELM-LRF algorithm. The experimental results show that this proposed method has significantly improved the accuracy of the defect detection and classification.

After detecting and classifying the surface defects of the magnetic tile, the overall defect detection accuracy and classification accuracy of the different defect types are obtained and compared with existing approaches for performance assessment and evaluation.

2.4. Technical Background of ELM-LRF. In 2004, a new type of single hidden layer feedforward neural network (SLFNs) was proposed by Huang and his team as extreme learning ,machine [21], which is characterized by easy parameter selection, fast learning speed, and good generalization performance. In this article, a special ELM, namely, ELM-LRF, is adopted [22]. The feature of this method is to add a single layer convolution and pooling network similar to CNN on the basis of ELM to realize the self-extraction of image features and to classify the input by the output weights of ELM.

In order to make the input more adequate, K different feature maps can be obtained by using K different input weights in the ELM-LRF [23]. Its specific implementation is divided into the following three steps.

(1) Randomly generating the initial weight A^{init}. The specific calculation formula is as follows:

$$\begin{aligned} &A^{\text{init}} \in R^{r^2 \times K}, \\ &A^{\text{init}} = \left[\alpha_1^{\text{init}}, \alpha_2^{\text{init}}, \ldots, \alpha_K^{\text{init}}\right], \\ &\alpha_k^{\text{init}} \in R^{r^2}, \\ &k = 1, 2, \ldots, K, \end{aligned} \tag{1}$$

where A^{init} is the initial weight, K is the number of feature maps, r^2 is the size of the local receptive field, and each column α_k in A^{init} is a set of the orthogonal bases of A^{init}. The input weight of the feature map k^{th} is $\alpha_k \in R^{r \times r}$, which is arranged by the column α_k.

The initial weight A^{init} is orthogonalized by the singular value decomposition (SVD), and the result of the orthogonalization is A. The value $c_{i,j,k}$ of the convolution node (i, j) of the feature map k^{th} is calculated by

$$\begin{aligned} c_{i,j,k}(x) = \sum_{m=1}^{r} \sum_{n=1}^{r} \left(x_{i+m-1, j+n-1} \cdot \alpha_{m,n,k}\right), \\ i, j = 1, 2, \ldots, (d - r + 1), \end{aligned} \tag{2}$$

where $d \times d$ is the input image size, $(d - r + 1) \times (d - r + 1)$ is the size of the feature map, and $\alpha_{m,n,k}$ is the input weight of the k^{th} feature map at (m, n) point.

(2) Square root pooling. The specific calculation formula is as follows:

$$h_{p,q,k} = \sqrt{\sum_{i=p-e}^{p+e} \sum_{j=q-e}^{q+e} c_{i,j,k}^2}, \quad p, q = 1, 2, \ldots, (d - r + 1). \tag{3}$$

If (i, j) is out of bound, then $c_{i,j,k} = 0$,

where the pooling size e represents the distance from the center of the pool to the edge [24]. In the ELM-LRF, the pooling map has the same size as the feature map, and both are $(d - r + 1) \times (d - r + 1)$. $c_{i,j,k}$ and $h_{p,q,k}$ represent the nodes (i, j) in the feature map k^{th} and the combined nodes (p, q) in the k^{th} pool map.

(3) Calculating the output weight matrix. To calculate the corresponding feature map and pooling map for each input sample x, the row vector is formed by concatenating the combined nodes in the pooling graph and then connecting the row vectors of the N input samples to obtain the combined layer matrix $H \in R^{N \times K \cdot (d-r+1)^2}$. The final combined layer and the output layer are fully connected. The output weight is β, which is calculated using regularized least squares analysis. The specific equation is as follows.

(a) If $N \leq K \cdot (d - r + 1)^2$,

$$\beta = H^T \left(\frac{1}{C} + HH^T\right)^{-1} T. \tag{4}$$

(b) If $N > K \cdot (d - r + 1)^2$,

$$\beta = \left(\frac{1}{C} + H^T H\right)^{-1} H^T T, \tag{5}$$

where N is the number of input samples, K is the number of feature maps, $(d - r + 1) \times (d - r + 1)$ is the size of the feature map, β is the output weight, T

is the expected output matrix, and C is the regularization parameter.

3. The Proposed Grid Search and Particle Swarm Optimization-Based Local Receptive Field-Enabled Extreme Learning Machine (GSPSO-LRF-ELM) Algorithm

The surface of the magnetic tile is curved and the curvature is different. The collected magnetic tile images have uneven illumination. In this article, we use the entropy weighted automatic threshold Otsu image segmentation method to segment magnetic tile images [14]. In view of the different local changes in the magnetic tile images, we use adaptive thresholds to segment uniform and nonuniform regions.

The GS is used to optimize the balance parameter C and the number of feature maps K in the ELM-LRF to find the optimal parameter combination (C, K). The PSO algorithm is used to optimize the initial weight in the ELM-LRF to find the optimal A^{init}. The optimized ELM-LRF classifier is called GSPSO-LRF-ELM. The classification algorithm flow chart is shown in Figure 3.

3.1. Optimization of the Balance Parameter C and the Number of Feature Maps K by GS. In the ELM-LRF algorithm, the most important parameter combination is the balance parameter C and the number of the feature maps K. The selection of these two parameters directly affects the performance of the algorithm. Therefore, the GS is used to optimize the parameter combination (C, K) in the ELM-LRF. The optimization Algorithm 1 of GS for the parameters C and K is as follows:

The array matrix is obtained by combining the values of the arrays A and B. Each array in the matrix is inputted into the classifier to obtain the corresponding classification accuracy. Comparing the accuracy of the each classification, the parameters Best$C$ and Best$K$ corresponding to the highest classification accuracy are selected as the balance parameter C and the number of feature maps K in the classifier.

3.2. Optimization of the Initial Weight A^{init} by PSO. The idea of PSO is derived from the foraging behavior of birds, in which each particle represents a set of possible solutions, and all particles form a group. The particles determine their speed and position according to their historical information and group historical information until the optimal solution is found. The iterative update equation is as follows:

$$\begin{aligned} v_{id}^{k+1} &= w v_{id}^{k} + c_1 r_1 \left(P_{id}^{k} - x_{id}^{k}\right) + c_2 r_2 \left(G_{d}^{k} - x_{id}^{k}\right), \\ x_{id}^{k+1} &= x_{id}^{k} + v_{id}^{k+1}, \end{aligned} \tag{6}$$

where v_{id}^{k} and x_{id}^{k} are the velocity and position of the number d dimension of particle i at the number k iteration and w is the weight, respectively, c_1 and c_2 are the learning factor of the individual and the group, P_{id}^{k} is the optimal position of particle i in the d dimension in the number k iteration, G_{d}^{k} is the optimal position of the individual in the d dimension of the whole population, and r_1, r_2 is a random number of [0, 1] intervals.

In order to make the PSO have the better global search ability in the early stage and the better local search ability in the later stage, the work adopts the nonlinear inertia weighting factor w [25], as shown in the following :

$$w = w_{\max} - \left(w_{\max} - w_{\min}\right) \times \arcsin\left(\frac{t}{t_{\max}} \times \frac{\pi}{4}\right), \tag{7}$$

where $w_{\max}$ and $w_{\min}$ are maximum and minimum weights, respectively, and t and $t_{\max}$ are the current iteration number and the maximum iteration number.

The work uses the PSO algorithm to optimize the initial weight A^{init} in the ELM-LRF algorithm. Firstly, D initial particles are generated, and the corresponding feature map matrix, pool graph matrix, and output weight matrix β are calculated; secondly, it uses the formula $H\beta = T$ to calculate the prediction label T; finally, the classification accuracy of the image is taken as the fitness function, and the optimization goal of the PSO is to maximize the fitness function. The PSO for initial weight optimization of ELM-LRF Algorithm 2 is as follows:

4. Experiments and Results

4.1. Experiment Settings. The dataset used in the experiment was from the dataset on surface defect detection of the magnetic tile collected by Institute of Automation, Chinese Academy of Sciences [2]. The folder name of the dataset is magnetic-tile-defect-datasets (magnetic-tile-defect-datasets dataset acquisition address: https://github.com/abin24/Magnetic-tile-defect-datasets.). A total of 1344 images were collected. In order to make the experiment more reasonable and reliable, the experimental data were randomly selected as the defect and defect-free images by 1 : 1. The types and quantities of experimental data selected are shown in Table 1. Due to the different ROI of different magnetic tiles, the size of the image is different. For this reason, the image is uniformly converted into 64 dip × 64 dip size before preprocessing.

The classification accuracy of the surface defect of the magnetic tile is used as a criterion for judging the experimental results. The higher the classification accuracy is, the better the classification performance of the algorithm has.

In addition, in order to better analyze the experimental results, the false detection rate and the missed detection rate of each category are separately counted as follows:

$$\begin{aligned} \text{false detection rate} &= \frac{\text{number of errors}}{\text{number of samples}}, \\ \text{missed detection rate} &= \frac{\text{number of not recognized}}{\text{number of samples}}. \end{aligned} \tag{8}$$

All experimental environments in this article are operating system Windows 8.1 64 bit, processor Intel Core i5-

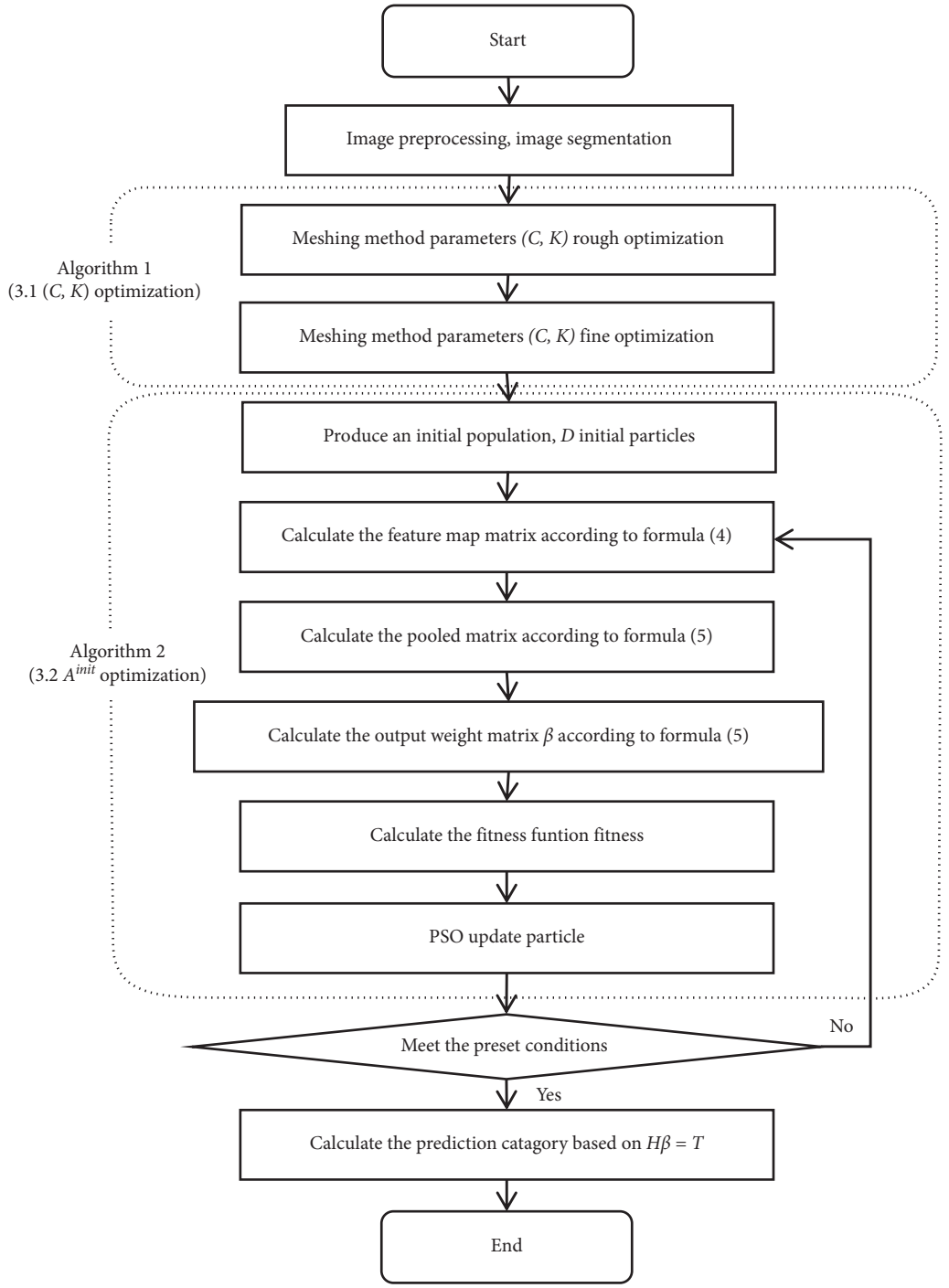


Figure 3: GSPSO-LRF-ELM classification algorithm flow chart.

4200M @ 2.50 GHz, memory (ARM) 8 GB, and software MATLAB R2018a.

4.2. Experiment Results and Analysis. We use ELM-LRF algorithm for defect detection and classification experiments. Dividing the dataset into training set and test set randomly, the types and quantities of images in training set and test set are shown in Table 2.

In the ELM-LRF algorithm, in order to analyze the influence of the balance parameter C and the number of feature maps K on the algorithm, the GS is used to optimize the two parameters. The parameter optimization is divided into two parts: rough optimization and fine optimization. In the rough optimization, the range of the parameter C is set to $\{10^{-4}, 10^{-3}, 10^{-2}, \ldots, 10^{2}, 10^{3}\}$, the range of the parameter K is set to $\{10, 20, 30, 40, 50, 60\}$, and the rough optimization 3D map of the results is shown in Figure 4. It can be seen from

```
(1) Setting the maximum, minimum and step size of C and K to get array A and array
(2) B respectively. The numerical angle in array A is 1: m, and the numerical angle in
(3) array B is 1: n,
(4) Bastacc = 0;
(5) For C = 1: m %C is the balance parameter
(6) {
(7)    For K = 1: n %K is the number of feature maps
(8)      {
(9)         Substituting C and K into the ELM-LRF algorithm, the classification
(10)        accuracy of the algorithm is obtained;
(11)        If Acc (C, K) > Bastacc
(12)           Bestacc = Acc (C, K);
(13)           BestC = C;
(14)           BestK = K;
(15)           End
(16)     }
(17) }
```

Algorithm 1: GS optimization for the parameters C and K.

```
(1) Particle swarm algorithm initialization begins:
(2)    generating D initial particles;
(3)    Calculating a feature map matrix, a pool map matrix, and an output weight
(4) matrix β corresponding to the particles;
(5)    The prediction label T is obtained by the formula Hβ = T;
(6)    Obtaining a fitness function;
(7) The update operation begins:
(8)    Individual update obtains the best value of the individual;
(9)    Global update obtains the best value of the global;
(10)   The optimal value is the optimal initial weight A^init;
```

Algorithm 2: PSO optimization for the initial weight A^{init}.

Table 1: Data categories and quantities used in the experiment.

Data category	Defect image					Defect-free image
	Break	Crack	Fray	Uneven	Blowhole	Free
Data quantity	85	57	32	103	114	391

Table 2: Training and testing data categories and quantities used in the experiment.

Data category	Defect image					Defect-free image	Total
	Break	Crack	Fray	Uneven	Blowhole	Free	
Training	60	40	20	70	80	270	540
Testing	25	17	12	33	34	121	242

Figure 4 that when lg$C$ is taken as −2 and K is taken as 50, a rough optimal parameter combination (C, K) is obtained.

For the further fine optimization, according to the rough optimization experiment results, the C and K setting range and step size are reduced, and the range of the parameter C is set to {0.005, 0.006, 0.007, ... 0.019, 0.020}, the range of the parameter K is set to {45, 46, 47, ... 53, 54, 55}, and the fine optimization 3D map of the results is shown in Figure 5. It can be seen from Figure 5 that when C is taken as 0.016 and K is taken as 55, a fine optimal parameter combination (C, K) is obtained. At this time, the highest classification accuracy of parameter optimization is 98.04%. In the parameter optimization experiment, the test classification accuracy is adopted as the criterion for judging the experimental results.

In the following experiment, the population size P was set first, followed by the number of iterations N, the maximum inertia weight $\omega_{\max}$ and the minimum $\omega_{\min}$, the learning factors $c1$ and $c2$, the maximum particle velocity $V_{\max}$ and the minimum $V_{\min}$, and the maximum particle position $X_{\max}$ and the minimum $X_{\min}$. The selection of the

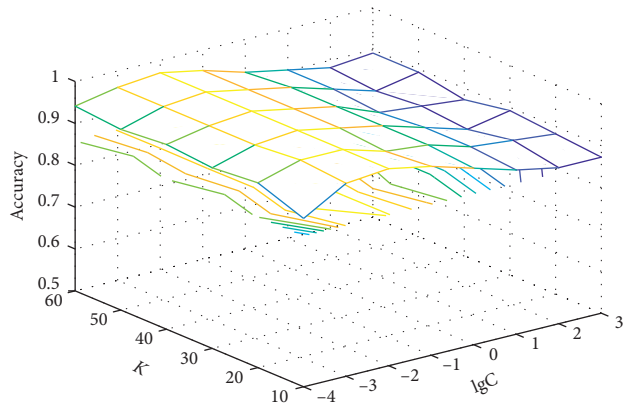


Figure 4: 3D results map of GS method for C, K parameters rough optimization.

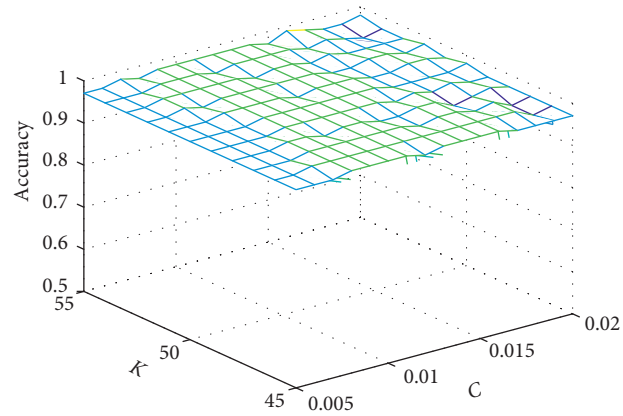


Figure 5: 3D results map of GS method for C, K parameters fine optimization.

balance parameter C, the number of feature maps K, the kernel size Ke, and the pool size e in the ELM-LRF algorithm are shown in Table 3.

The proposed classification algorithm counts the detection and classification results of six categories of the magnetic tile surface. The statistical results are shown in Table 4. As seen from Table 4, the classification algorithm proposed in this article can achieve 100% correct rate when identifying the defect and defect-free magnetic tiles. Among the five types of the defect categories, the classification rate of "break" is the highest, which is 100%; the lowest rate of "uneven" is 86.67%. In the experiment, the number of the defective magnetic tiles is falsely detected as the blowhole and the number of uneven defective magnetic tiles is missed from detection the most, both of which are 4 pieces. Based on the whole test results, the proposed algorithm achieves good results in the detection of surface defects of the magnetic tiles and can be applied to detect and identify surface defects of the magnetic tile in actual production.

4.3. Comparative Experiment. In order to verify the performance of GSPSO-LRF-ELM algorithm in image classification, the proposed algorithm was compared with four traditional classification algorithms, support vector machine (SVM), artificial neural network (ANN), extreme learning machine (ELM), and local receptive field-based extreme learning machine (ELM-LRF). The experimental results are shown in Table 5.

As can be seen from Table 5, the training accuracy of the proposed GSPSO-LRF-ELM algorithm is 99.07%, and the test accuracy is 96.36%, the highest among all five compared algorithms. The test accuracy of the proposed algorithm is 4.54% higher than that of the traditional ELM-LRF algorithm, and it is much higher than the other three classical classification algorithms. In terms of time consumption, ELM training time and test time are the shortest. SVM and ANN detection time is also very short. Comparatively speaking, the training and testing time of the proposed GSPSO-LRF-ELM algorithm in this article is longer. This is because the convolution and pooling layers are added to the algorithm, and the input weight is optimized by PSO, so the algorithm runs longer. Although the running time of the algorithm proposed in this article is slightly longer, its training and testing accuracy have been significantly improved, which is more consistent with the demand for detection efficiency in the offline defects detection of the magnetic tile.

In order to make the experiment more comprehensive, the accuracy, false detection rate, and missed detection rate

Table 3: Parameters related to the GSPSO-LRF-ELM algorithm.

ω_{max}	ω_{min}	$c1$, $c2$	V_{max}/V_{min}	X_{max}/X_{min}	P	N	C	K	Ke	e
1.2	0.4	1.49445	±0.5	±3	25	70	0.016	55	4 ∗ 4	4

Table 4: Classification results of various surface defect categories.

Category	Identification/ total	Correct rate (%)	False detection/ Total	False detection rate (%)	Missed detection/ total	Missed detection rate (%)
Free	109/109	100.00	1/109	0.92	0/109	0.00
Break	24/24	100.00	2/24	8.33	0/24	0.00
Crack	15/17	88.24	0/17	0.00	2/17	11.76
Fray	8/9	88.89	1/9	11.11	1/9	11.11
Uneven	26/30	86.67	0/30	0.00	4/30	13.33
Blowhole	30/31	96.77	4/31	12.90	1/31	3.23

Table 5: Comparison of classification results of different classification algorithms.

Algorithm	Training accuracy (%)	Training time (s)	Test accuracy (%)	Testing time (s)
SVM	95.56	0.2443	86.82	0.4672
ANN	92.15	6.9703	87.73	0.0215
ELM	73.18	**0.0066**	72.27	**0.0026**
ELM-LRF	97.85	47.8906	91.82	10.2463
GSPSO-LRF-ELM	**99.07**	26.4375	**96.36**	12.1420

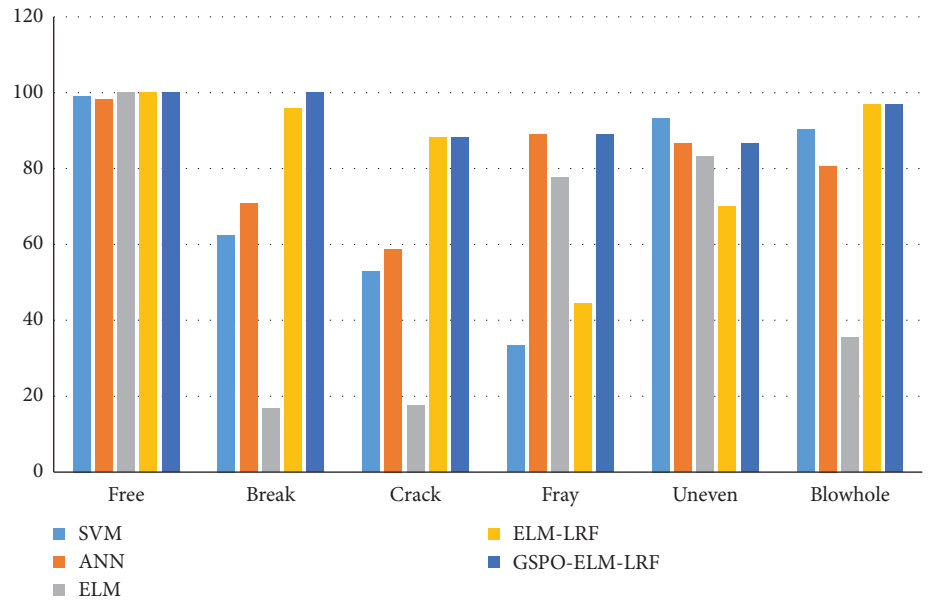


Figure 6: Comparison of five algorithms for *correct* rate (the horizontal axis is the defect type, and the vertical axis is the *correct rate*).

of the five algorithms for six categories of the magnetic tile defect as shown in Figures 6–8.

As can be seen from Figures 6–8, the classification accuracy of the GSPSO-LRF-ELM algorithm for the magnetic tile defect category is higher than that of the other four algorithms, and the classification rate of "free" and "break" reaches 100%. In terms of false detection rate, the overall false detection rate of the proposed algorithm is low, and the false detection rate of "crack" and "uneven" is 0%. In terms of missed detection rate, the algorithm proposed in this article is significantly lower than the other four classification algorithms, and the missed detection rate of "free" and "break" is 0%.

In summary, the algorithm proposed in this article is outstanding in the classification of the magnetic tile defects,

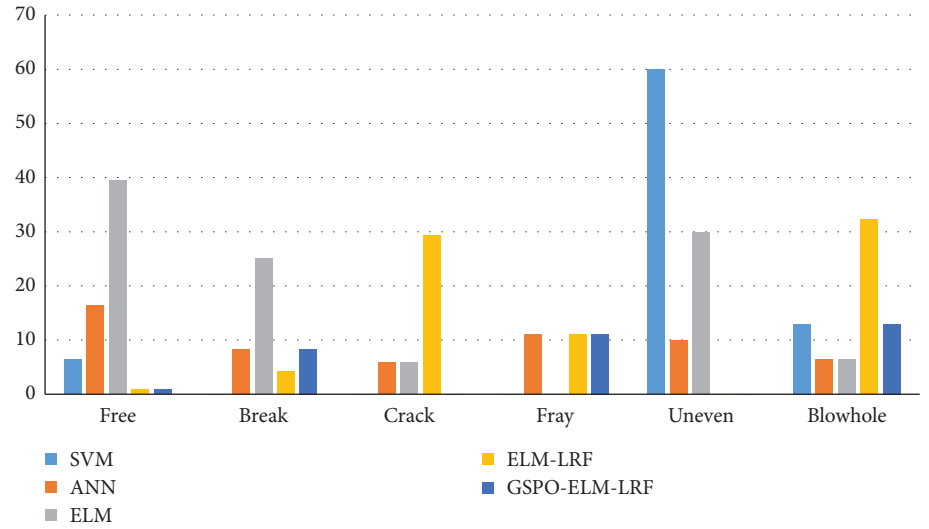


Figure 7: Comparison of five algorithms for *false detection* rate (the horizontal axis is the defect type, and the vertical axis is the *false detection rate*; less than five columns in each defect means *false detection rate* is 0% in the missing algorithm).

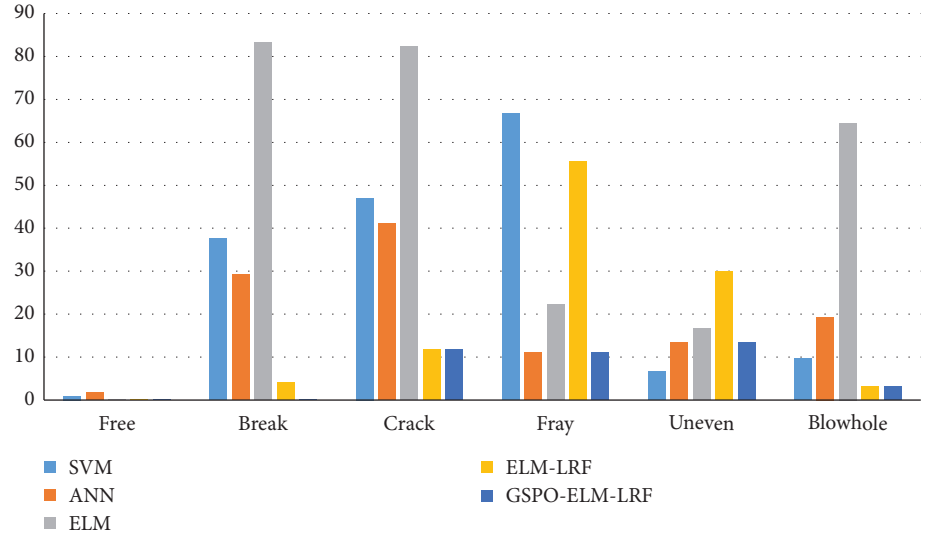


Figure 8: Comparison of five algorithms for *missed detection rate* (the horizontal axis is the defect type, and the vertical axis is the *missed detection rate*; less than five columns in each defect means *missed detection rate* is 0% in the missing algorithm).

and the classification accuracy is higher than the other four algorithms.

5. Conclusion

In this article, we propose an optimized local receptive field-based extreme learning machine for detecting and classifying the surface defects of the magnetic tile. In the ELM-LRF classifier, considering the difficulty of selecting the balance parameter C and the number of the feature maps K, the grid search method (GS) is used to optimize the balance parameter C and the number of the feature maps K, and in order to obtain the optimal initial weight A^{init}, the particle swarm optimization algorithm (PSO) is used to optimize the initial weight A^{init} of the classifier. The optimized classifier is named GSPSO-ELM- LRF. After preprocessing and segmenting, the magnetic tile images are inputted to the GSPSO-LRF-ELM classifier, and the surface defects of the magnetic tile will be detected and classified. Through experimental comparison and analysis, the method proposed in this article has the highest accuracy and better detection efficiency in the detection and classification of the magnetic tile surface defects.

In the future, the online detection and classification systems for the surface defects of the magnetic tiles will be further researched to achieve real-time detection, classification, and analysis.

Data Availability

The dataset used in the experiment was from the dataset on surface defect detection of the magnetic tile collected by

Institute of Automation, Chinese Academy of Sciences. The Magnetic-tile-defect-datasets master data used to support the findings of this study have been deposited in the https://github.com/abin24/Magnetic-tile-defect-datasets repository.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61503271 and 61603267) and Shanxi Natural Science Foundation of China (201801D121144 and 201801D221190).

References

[1] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: a survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, 2008.

[2] Y. B. Huang, C. Y. Qiu, and K. Yuan, "Surface defect saliency of magnetic tile," *The Visual Computer*, vol. 36, no. 1, pp. 1–12, 2020.

[3] Y. W. Wang, J. Y. Tao, X. C. Chen, and K. Wang, "Defects detection for rough magnetic tiles surface based on light sectioning," in *Proceedings of the 8th International Symposium On Advanced Optical Manufacturing and Testing Technologies: Optical Test, Measurement Technology, and Equipment*, vol. 9684, Article ID 968434, Suzhou, China, April 2016.

[4] C. Yang, P. Liu, G. Yin, H. Jiang, and X. Li, "Defect detection in magnetic tile images based on stationary wavelet transform," *NDT & E International*, vol. 83, no. 10, pp. 78–87, 2016.

[5] I. Valavanis and D. Kosmopoulos, "Multiclass defect detection and classification in weld radiographic images using geometric and texture features," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7606–7614, 2010.

[6] X. Li, H. Jiang, and G. Yin, "Detection of surface crack defects on ferrite magnetic tile," *NDT & E International*, vol. 62, pp. 6–13, 2014.

[7] C. Yang, P. Liu, G. Yin, and L. Wang, "Crack detection in magnetic tile images using nonsubsampled shearlet transform and envelope gray level gradient," *Optics & Laser Technology*, vol. 90, no. 5, pp. 7–17, 2017.

[8] D. He, K. Xu, and P. Zhou, "Defect detection of hot rolled steels with a new object detection framework called classification priority network," *Computers & Industrial Engineering*, vol. 128, pp. 290–297, 2019.

[9] L. Xie, L. Lin, M. Yin, L. Meng, and G. Yin, "A novel surface defect inspection algorithm for magnetic tile," *Applied Surface Science*, vol. 375, pp. 118–126, 2016.

[10] X. Zhou, Z. L. Long, J. Niu, X. J. Wu, and W. Chao, "Defect segmentation of ultrasonic aluminum bonding joint based on region growing and level-set," in *Proceedings of the 20th International Conference on Electronics Materials and Packaging (EMAP)*, pp. 1–4, Clear Water Bay, Hong Kong, December 2018.

[11] A. Mostafa, M. A. Elfattah, A. Fouad, A. Ella Hassanien, and K. Tai-Hoon, "Region growing segmentation with iterative K-means for CT liver images," in *Proceedings of the 4th International Conference on Advanced Information Technology and Sensor Application (AITS)*, pp. 88–91, Harbin, China, August 2015.

[12] S. C. Wang, P. Dai, X. Y. Du, Y. Huang, and J. Liu, "3D histogram based maximum entropy threshold segmentation for railway fence detection," in *Procccedings of the 14th IEEE International Conference on Signal Processing (ICSP)*, pp. 807–811, Beijing, China, August 2018.

[13] M. Malarvel, G. Sethumadhavan, P. C. R. Bhagi, S. Kar, and S. Thangavel, "An improved version of Otsu's method for segmentation of weld defects on X-radiography images," *Optik*, vol. 142, pp. 109–118, 2017.

[14] M. T. N. Truong and S. Kim, "Automatic image thresholding using Otsu's method and entropy weighting scheme for surface defect detection," *Soft Computing*, vol. 22, no. 13, pp. 4197–4203, 2018.

[15] Q. Zhou, R. Chen, B. Huang, C. Liu, J. Yu, and X. Yu, "An automatic surface defect inspection system for automobiles using machine vision methods," *Sensors*, vol. 19, no. 3, pp. 644–661, 2019.

[16] J. Kumar, R. S. Anand, and S. P. Srivastava, "Flaws classification using ann for radiographic weld images," in *International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 145–150, Noida, India, February 2014.

[17] D. Yapi, M. S. Allili, and N. Baaziz, "Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1014–1026, 2018.

[18] I. Cetiner, A. A. Var, and H. Cetiner, "Classification of knot defect types using wavelets and KNN," *Elektronika Ir Elektrotechnika*, vol. 22, no. 6, pp. 67–72, 2016.

[19] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Applied Sciences*, vol. 8, no. 9, pp. 1575–1589, 2018.

[20] T. Wang, Y. Chen, M. N. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3465–3471, 2017.

[21] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, Budapest, Hungary, July 2004.

[22] G.-B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 18–29, 2015.

[23] H. Vong, F. Li, X. Xu, and F. Sun, "Multi-modal local receptive field extreme learning machine for object recognition," *Neurocomputing*, vol. 277, pp. 4–11, 2018.

[24] H. Liu, F. Li, X. Xu, and F. Sun, "Active object recognition using hierarchical local-receptive-field-based extreme learning machine," *Memetic Computing*, vol. 10, no. 2, pp. 233–241, 2018.

[25] H. Lu, B. Du, J. Liu, H. Xia, and W. K. Yeap, "A kernel extreme learning machine algorithm based on improved particle swam optimization," *Memetic Computing*, vol. 9, no. 2, pp. 121–128, 2017.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 6152041, 8 pages
https://doi.org/10.1155/2020/6152041

Research Article

Learning from Large-Scale Wearable Device Data for Predicting the Epidemic Trend of COVID-19

Guokang Zhu, Jia Li, Zi Meng, Yi Yu, Yanan Li, Xiao Tang, Yuling Dong, Guangxin Sun, Rui Zhou, Hui Wang, Kongqiao Wang, and Wang Huang

Huami Corporation, Hefei, China

Correspondence should be addressed to Kongqiao Wang; kongqiao.wang@huami.com

Received 27 March 2020; Accepted 15 April 2020; Published 5 May 2020

Guest Editor: Jinchang Ren

The coronavirus disease 2019 (COVID-19) pandemic has triggered a new response involving public health surveillance. The popularity of personal wearable devices creates a new opportunity for tracking and precaution of spread of such infectious diseases. In this study, we propose a framework, which is based on the heart rate and sleep data collected from wearable devices, to predict the epidemic trend of COVID-19 in different countries and cities. In addition to a physiological anomaly detection algorithm defined based on data from wearable devices, an online neural network prediction modelling methodology combining both detected physiological anomaly rate and historical COVID-19 infection rate is explored. Four models are trained separately according to geographical segmentation, i.e., North China, Central China, South China, and South-Central Europe. The anonymised sensor data from approximately 1.3 million wearable device users are used for model verification. Our experiment's results indicate that the prediction models can be utilized to alert to an outbreak of COVID-19 in advance, which suggests there is potential for a health surveillance system utilising wearable device data.

1. Introduction

Since the outbreak of the coronavirus disease 2019 (COVID-19) pandemic, more than 300,000 people have been infected in at least 127 countries as of March 23, 2020, according to the World Health Organization's (WHO's) report [1]. COVID-19 spreads easily from person to person and has killed thousands of people [2–5]. Since the beginning of the COVID-19 outbreak, several studies have been carried out to forecast the epidemic trend of COVID-19 in China [6–8]. For example, Wu et al. built a Susceptible-Exposed-Infectious-Recovered (SEIR) model to simulate the epidemics across the major cities in China [7]. Yang et al. applied the Long Short Term Memory (LSTM) model to predict the number of newly infected COVID-19 cases by utilizing data from the outbreak of Severe Acute Respiratory Syndrome (SARS) in 2003 [6]. Although the models used in those studies could simulate the outbreak trend of the disease, they relied heavily on officially reported statistics; therefore, the timeliness of the models could be affected. On the contrary, big data analysis, such as analysis of Internet data, may provide real-time surveillance and improve the timeliness of the forecasting [9–16]. For instance, Google invented the influenza epidemic prediction tool Google Flu Trend (GFT) to estimate the level of in-fluenza activity based on the individual web search queries from different regions [9–11]. They assumed that more individuals in a certain region might search online for the information about specific diseases if the influenza disease risk was higher in that certain region. Therefore, Google built a database containing 50 million of the most common web search queries on all influenza-related topics and constructed the risk prediction model GFT using this search query data as the input [9]. Google showed that GFT could help predict the influenza-like illness outbreak 7–10 days before the Centers for Disease Control and Prevention (CDC) report [10]. In fact, the surveillance report from CDC usually has a lag time of around 1-2 weeks. Therefore, the result from Google indicated that big data analysis could improve timeliness for public health surveillance. However, search queries can be

greatly influenced by social hotspots, which weakens the correlation between the search queries and the occurrence of influenza-like diseases [17].

With the rise in popularity of fitness band and smartwatch devices, physiological signs, such as heart rate, activity, sleep, etc., can be conveniently acquired from these wearable biosensors [18–20]. As of 2019, more than 100 million consumers owned Huami wearable devices, and the number continues to grow. In contrast with the big data from web search engines, data from wearable devices can provide more objective information on the health status of the users. For example, once users are infected with an influenza-like illness, their physiological signs would be altered. Radin et al. explored the relationship between the physiological anomaly rate from wearable device users and the influenza-like illness rate reported by the US CDC [21] to build the regression models for predicting the influenza-like illness cases within different states of America. They utilized the heart rate and sleep data from the wearable devices to improve upon the standard models. The prediction results have strong correlation with the official data. Li et al. also investigated the role of physiological changes measured with wearable devices on the diagnosis and analysis of disease [22]. The researchers established a personalized disease detection framework, which identifies abnormal physical signs, e.g., from Lyme disease and other inflammatory responses, from the longitudinal data of the individuals. All the studies mentioned above can inform the way wearable device data is used for public health surveillance.

According to clinical studies [23–25], the most common symptoms at the onset of COVID-19 are fever, cough, and fatigue, which are closely related to the physiological signs measured by the wearable devices. Therefore, a good method to predict the epidemic trend of COVID-19 may involve building a prediction model based on the wearable device data.

The main purpose of this study is to provide a novel framework for predicting the trend of COVID-19 outbreak within different countries and cities, using big data collected from wearable devices. There are two major contributions from this study: (1) a physiological anomaly detection method is developed and can identify the anomalous signs reflected by the physiological data from wearable sensors; (2) an online learning framework is proposed for public health emergency surveillance.

2. Methodology

2.1. Physiological Anomaly Detection. According to a study on fever and cardiac rhythm [26], heart rate increases by 8.5 beats per minute, on average, for every 1°C increase in body temperature, so an elevated resting heart rate (RHR) might be related to fever caused by COVID-19 or influenza-like illness. The basic anomaly detection method is based on the elevated RHR. Because shortened sleep length also causes an increase in RHR [27], we weaken the contribution of this factor in the physiological anomaly detection method.

RHR and sleep length are directly acquired with the corresponding sensors of Huami wearable devices. Both kinds of synchronized data from the accelerometer (ACC) sensor and the photoplethysmography (PPG) sensor are used to analyze sleep status (including sleep recognition and stage) for measuring sleep length. During sleep, the PPG data is used to compute the RHR. For each user, overall mean and standard deviation (SD) of RHR and sleep length throughout the entire period are calculated. A daily RHR is defined as an anomaly if it is larger than the average RHR plus 1.5 SD, and if in addition, the daily sleep is longer than the average sleep minus 0.5 SD. Considering that COVID-19 or influenza-like illness persist for several days, we define the detection standard of physiological anomaly as continuous anomaly measured for at least five consecutive days.

2.2. Online Prediction of COVID-19 Infection Rate. The physiological anomaly detected by our method is an indication of fever, which in fact can be caused by COVID-19 or other influenza-like illness. Thus, the key point for COVID-19 infection rate prediction is to distinguish an anomaly arising from COVID-19 from the wider category of physiological anomalies. To this end, as shown in Figure 1, a heterogeneous neural network [28] regression model combining sparse categorical features and dense numerical features (CDNet) is proposed.

CDNet concatenates 2 subnetworks: CatNN and DenNN. The inputs of the CatNN are sparse categorical features, i.e., holiday activity, season, and weather. The inputs of the DenNN are historical physiological anomaly rate, active user density, and historical officially reported COVID-19 rate, where the historically detected physiological anomaly rate is calculated with dividing the number of users detected with a physiological anomaly by the number of total active users. The output layer of CDNet normalized by a Sigmoid function outputs the predicted physiological anomaly rate. The detailed inputs and outputs are summarized in

$$R'_{t+1,k} = \mathrm{CDNet}\Big(\big\{R_{t-j,k}, r_{t-j,k}, C_{t-j,k}, c_{t-j,k},\ j = 0, 1, \ldots, 6\big\}, RC_{t,k}, D_{t,k}\Big), \tag{1}$$

where, for country or city k, the output $R'_{t+1,k}$ is the predicted physiological anomaly rate in the next period, $R_{t-j,k}$ is the physiological anomaly rate the j-th period earlier, $r_{t-j,k}$ is the physiological anomaly rate in the same period of $R_{t-j,k}$ last year, $C_{t-j,k}$ and $c_{t-j,k}$ are the corresponding categorical information with the same temporal definition as $R_{t-j,k}$ and $r_{t-j,k}$, respectively, $RC_{t,k}$ is the officially reported COVID-19 rate (ratio of confirmed COVID-19 patient number to the number of residents in the country or city) in the current period, $D_{t,k}$ is the current active user density (ratio of active user number to the number of residents in the country or city). To distinguish regional disparity, four different CDNet models are trained separately for North China, Central China, South China, and South-Central Europe.

In order to get the predicted anomaly rate caused by COVID-19 for the next period, the predicted physiological anomaly rate with ($R_{t+1,k}$) and without ($R'_{t+1,k}|RC_{t,k} = 0$) the supervision of officially reported data is calculated separately. As shown in Figure 2, the supervision is removed by

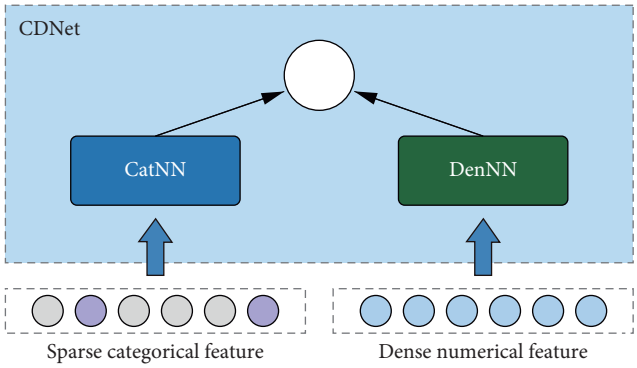


Figure 1: Diagram of the CDNet neural network architecture.

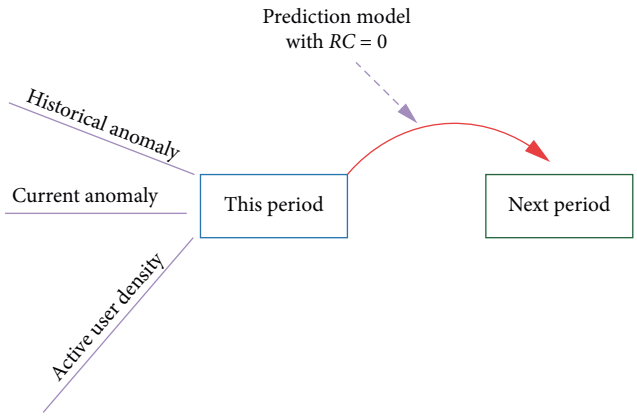


Figure 2: Illustration for the prediction stage of the model.

setting $RC_{t,k}$ as 0. Then, the predicted anomaly rate caused by COVID-19 for the next period $P'_{t+1,k}$ can be calculated as the difference between $R'_{t+1,k}$ and $R_{t+1,k} \mid RC_{t,k} = 0$:

$$\mathbf{P}'_{t+1,k} = \mathbf{R}'_{t+1,k} - \left\{\mathbf{R}'_{t+1,k} \mid \mathbf{RC_{t,k}} = 0\right\}. \quad (2)$$

To consecutively predict the epidemic trend of COVID-19, the CDNet model is trained in an online learning way. As shown in Figure 3, the initial CDNet model M_0 is trained with the input of $\{R_{t-j,k}, r_{t-j,k}, C_{t-j,k}, c_{t-j,k}, j = 1, 2, \ldots, 7\}$, and with the target as $R_{t,k}$. The weights of CDNet are updated step by step with the transmission of COVID-19, using the arriving data of newly officially reported COVID-19 rate and detected physiological anomaly rate. The step size of the sliding window for online learning is set as 1 week.

3. Experiments

3.1. Dataset. Anonymised sensor data of approximately 1.3 million users who wore Huami devices from July 1, 2017, to April 8, 2020 were obtained according to appropriate security control processes. All users are notified that their anonymised data could potentially be used for academic research under the Huami Privacy Policy.

All the users wore their Huami devices for at least 100 days throughout the entire period. Daily measures include RHR, activity, and sleep length, which are the bases of physiological anomaly detection. Data with missing RHR or sleep length were excluded. The daily COVID-19 infection rate data come from CDC of the corresponding countries.

We build separate models for different countries and cities listed in Table 1, according to the geographical segmentation considering the regional and lifestyle differences. Taking North China as an example, we utilized data from five representative cities (Beijing, Shijiazhuang, Jinan, Taiyuan, and Tianjin) for analysis and model building. The detailed summaries of the active user numbers are also listed in Table 1. The users enrolled in the study were chosen from 19 cities of Central, Southern, and Northern China and seven South-Central European countries to sufficiently reveal the regional disparity.

3.2. Analysis Result in China. The consecutive 3-year physiological anomaly rate curves in Wuhan together with the predicted physiological anomaly rate curves with and without the supervision of the officially reported COVID-19 infection rate in 2020 are illustrated in Figure 4. They are aligned by the time of the Chinese Spring Festival in the temporal axis. In the figure, all five curves peak around the time of Chinese Spring Festival. In addition, the predicted physiological anomaly rate with the supervision of official data in 2020 fits well with the rate calculated by the anomaly detection algorithm, which validates the prediction performance of the CDNet. Additionally, the physiological anomaly rate curve excluding COVID-19 in 2020 overlaps with both the predicted and the detected physiological anomaly rate curves including COVID-19 in 2020 before the outbreak of COVID-19, which verifies the basic reliability of the model. After that, all these three curves rise rapidly, which indicates that the outbreak of influenza-like illness is occurring alongside COVID-19. The predicted outbreak period aligns with the real-life situation. In addition, we also predicted the physiological anomaly rate curve from 2018 to 2019 with the prediction model and found that the predicted curve fits well with the total anomaly rate curve during the 2 years. This may indicate that the obvious separation happening around the Chinese Spring Festival between the predicted anomaly rate curves with and without the supervision of the officially reported COVID-19 infection rate in 2020 results from the outbreak of COVID-19.

Figure 5 illustrates the predicted COVID-19 infection rate across five Chinese cities and the officially reported accumulating COVID-19 infection cases in Wuhan. In the figure, there is a clear outbreak period in the predicted infection rate curve for each city, which may correspond to that of the newly confirmed cases. Taking Wuhan as an example, the predicted infection rate peaks around January 28, while the officially reported newly confirmed infection rate in Wuhan reached its highest on February 8 (the data after February 12 in Wuhan is omitted since the COVID-19 diagnostic criteria changed on that day, which causes a sudden sharp increase of 13,436 newly confirmed cases). The predicted disease peak is ahead of the officially reported peak by 11 days. The predicted earlier peak may indicate that health surveillance involving wearable sensors can play an

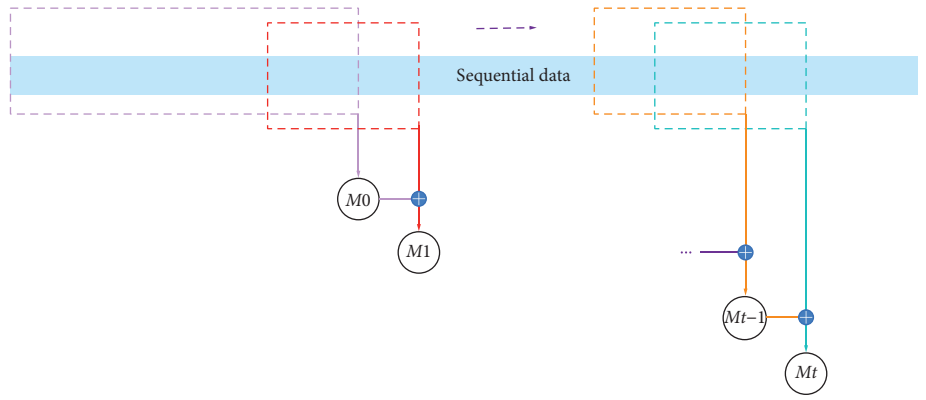


Figure 3: Work-flow of the online prediction framework for COVID-19 infection trend.

Table 1: Number of users enrolled in the study.

Region	Cities or countries	User number	Cities or countries	User number
North China	Beijing	126,575	Jinan	42,569
	Shijiazhuang	33,257	Taiyuan	16,753
	Tianjin	49,237		
Central China	Shanghai	153,711	Nanjing	76,204
	Hangzhou	78,840	Chengdu	64,436
	Wuhan	44,529	Hefei	33,257
	Nanning	19,641	Huanggang	3,412
	Xiaogan	2,358		
South China	Guangzhou	92,219	Shenzhen	71,669
	Foshan	20,229	Dongguan	29,146
	Fuzhou	23,061		
South-central Europe	Italy	68,494	Portugal	18,343
	Spain	187,788	Switzerland	4,431
	Germany	65,941	Greece	12,830
	France	34,711		

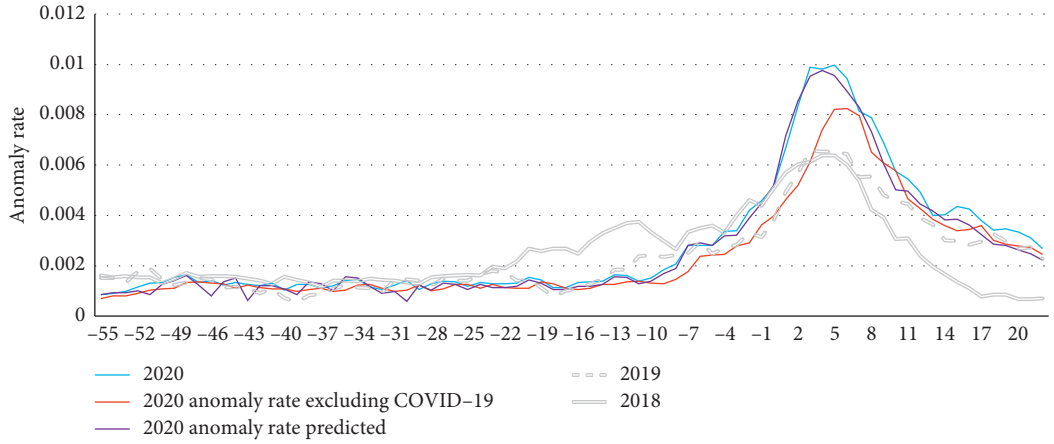


Figure 4: Physiological anomaly rate curves in Wuhan aligned by the time of Chinese Spring Festival. The two grey curves and the blue curve represent the physiological anomaly rate calculated by the anomaly detection algorithm from 2018 to 2020, respectively. The purple curve and the red curve represent the predicted physiological anomaly rate with and without the supervision of the official data, respectively.

important role in alerting to infectious disease outbreaks and in timely public health management. In fact, Wu and McGoogan also found there was a lag between the start of the illness and the diagnosis of COVID-19 by viral nucleic acid testing [2]. The newly infected cases actually peaked around January 28 if determined by the onset of the symptoms, which happens to be consistent with our findings. In addition, Figure 5 also shows that the predicted infection rate in Wuhan gradually decreases following January 28 and reaches a local minimum on February 1, which may

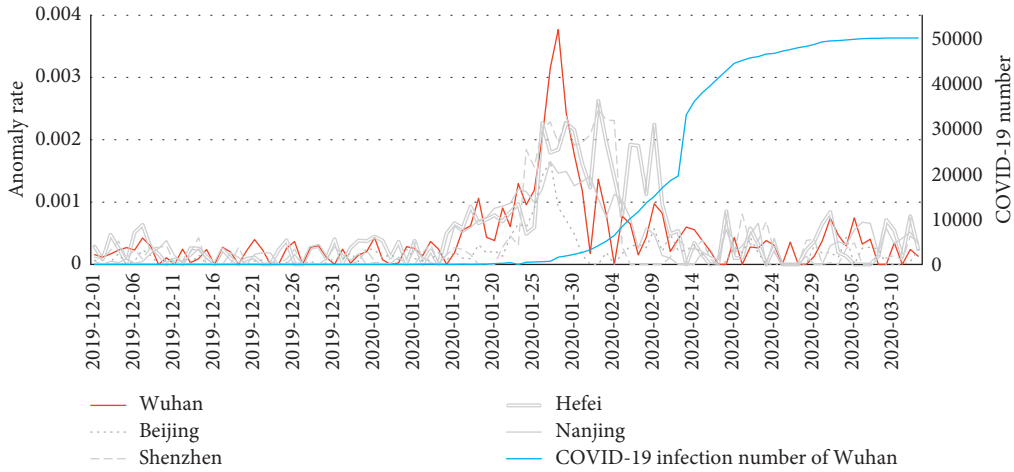


Figure 5: Predicted COVID-19 anomaly rate of 5 representative Chinese cities and the officially reported accumulating COVID-19 infection number of Wuhan.

correspond to the plateau in the officially reported accumulating infection curve that occurs after February 19. This result may indicate that the model can also predict the disease control outcome in advance. Moreover, Figure 5 shows Wuhan has the highest prediction disease peak among the five cities. This is also consistent with the fact that Wuhan is the most affected city in China.

3.3. Analysis Result in Italy and Spain. Figures 6(a) and 6(b) illustrate the predicted COVID-19 infection rate and the officially reported accumulating COVID-19 infection rate in Italy and Spain, respectively. The predicted infection rate in Italy rises rapidly from February 23, 2020, which coincides with the outbreak of COVID-19 in this country. As for Spain, the predicted infection rate starts to increase from February 29, which is 6 days later than Italy, and the predicted rate increases quickly following that. This is consistent with the real-life situation where the outbreak of COVID-19 was later in Spain.

As shown in Figure 6, the principal peak in the predicted COVID-19 infection curve of either Italy or Spain arrives as of April 8. In correspondence to the largest number of newly confirmed infection cases, which are reported officially by Italy on March 21 and Spain on March 25, the predicted principal peaks for the two countries occur around the time of March 13 and March 18, respectively. Both predicted principal peaks are ahead of the officially reported data by at least 1 week.

3.4. Correlation Analysis. To evaluate the appropriateness of predicting COVID-19 infection rate from physiological anomaly rate, we chose 19 Chinese cities to calculate the correlation between the officially reported COVID-19 infection rate and the detected physiological anomaly rate using Pearson's correlation coefficient shown in equation (3). In the equation, t_0 represents the start of the COVID-19 outbreak, t_1 stands for the end of the study period, and X, Y represent the officially reported COVID-19 infection rate and the physiological anomaly rate, respectively. The correlation analysis is performed in two steps. In the first step, we find the point, corresponding to the outbreak peak point of the officially reported COVID-19 infection curve, on the physiological anomaly rate curve. In the second step, we align the curves by the two points, and calculate the correlation coefficient.

$$\rho_{X,Y} = \frac{\sum_{t=t0}^{t1}(X_t - \overline{X})(Y_t - \overline{Y})}{\sqrt{\sum_{t=t0}^{t1}(X_t - \overline{X})^2}\sqrt{\sum_{t=t0}^{t1}(Y_t - \overline{Y})^2}}. \quad (3)$$

Pearson's correlation coefficients, ρ, for different cities in China are listed in Table 2. The average ρ value reaches around 0.68, which is strong correlation that further supports the opinion that physiological signs are useful for public health emergency alert. However, some cities do not show strong correlation, which may be due to the following reasons. Firstly, the officially reported cases of infection in some cities, e.g., Wuhan, were adjusted on certain days resulting in sudden changes. Secondly, the number of active users in some cities, e.g., Nanning, are relatively small which influences the performance of the model; therefore, the ρ value can be further improved when the number of active users increases. Finally, some cities, e.g., Beijing, have unstable user population and data noise due to the population shift.

3.5. Retention Effect. In the above correlation analysis, it is noticeable that there might be some retention effect in the detected physiological anomaly rate. To be specific, some people with anomalous measurements may continue to wear their devices so that they are calculated as anomalies on multiple days. This results in statistical error during the correlation analysis.

In order to analyze the impact, we calculate the retention rates of people detected as anomalies for several consecutive days. As shown in Figure 7, if a person is detected as anomaly on a certain day, the possibility of wearing the device is decreasing gradually from 3.5% down to 0.2% in the following 4 days. This indicates that the retention effect may have very limited influence on the correlation analysis.

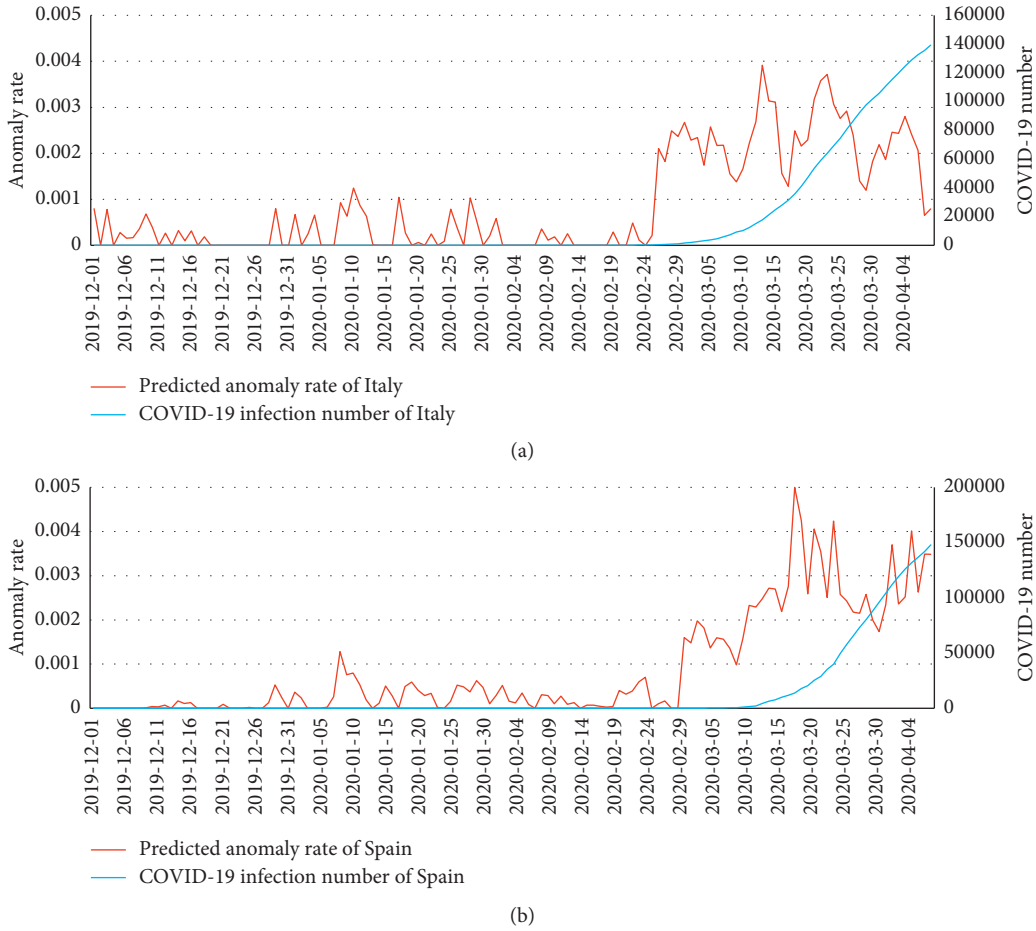


Figure 6: Predicted COVID-19 anomaly rate and the officially reported accumulating COVID-19 infection number of (a) Italy and (b) Spain.

Table 2: Correlation coefficient between the physiological anomaly rate and the officially reported number of newly confirmed COVID-19 cases.

City	ρ
Beijing	0.31
Shijiazhuang	0.58
Tianjin	0.53
Jinan	0.68
Taiyuan	0.55
Shanghai	0.51
Hangzhou	0.74
Wuhan	0.58
Nanning	0.52
Xiaogan	0.70
Nanjing	0.83
Chengdu	0.75
Hefei	0.84
Huanggang	0.87
Guangzhou	0.80
Foshan	0.81
Fuzhou	0.73
Shenzhen	0.82
Dongguan	0.75
Average	**0.68**

4. Discussion

In this study, a prediction model for COVID-19 epidemic trends has been realized using physiological data collected by wearable devices. The results show that prediction with dynamic physiological data may have an advantage in alerting to the infection outbreak in advance. However, the detection method for calculating the physiological anomaly rate has some limitations.

Firstly, on holidays, e.g., Chinese Spring Festival, Christmas, etc., transportation and population shift, social activities, and alcohol drinking might greatly influence the physiological signs of the users. For example, the elevated RHR due to heavy drinking on holidays might persist for several days and greatly influences the physiological anomaly rate to be detected. Especially for China, the outbreak of COVID-19 and influenza-like illness overlap with the Chinese Spring Festival. Thus, it is necessary to distinguish the elevated RHR cases induced by holiday activities from infection.

Secondly, the anomaly rate is the statistical description of wearable device users' physiological signs measured in the anomalous range. The validity of the statistical description depends on both the user scale and diversity. For example,

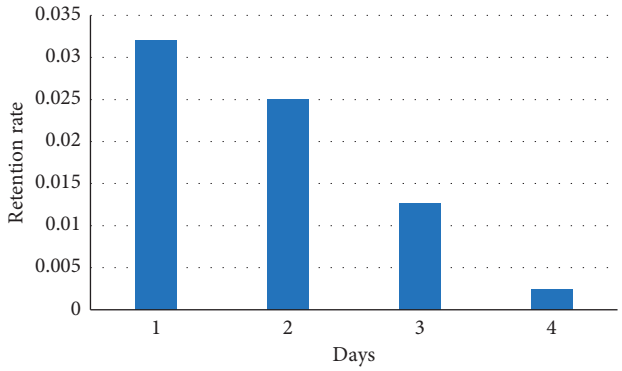


Figure 7: The retention rate across the timeline.

for a city with 0.1% officially reported infection rate of COVID-19, if the number of active users in the city is less than 10,000, there might be only 10 people infected among them. Such scale of data cannot support a convincing inference. Regarding the diversity, the prediction accuracy can be greatly improved if the distribution of active users is consistent with the natural distribution. For example, since elderly people and people with other diseases, e.g. cardiovascular disease (CVD), are more susceptible to COVID-19 [2, 3], the statistical performance of the model will be influenced if there is not enough coverage of such people.

Thirdly, although the current study provides a population evolution model for public health surveillance, it may be more meaningful for medical workers as well as individuals to take early precautions, if individualized health status prediction model is available. In the future, such prediction models based on wearable device data will be explored by incorporating more individual features, such as age, gender, body mass index (BMI), etc.

5. Conclusions

Public health emergencies can cause severe damage to the health and prosperity of our society. The popularity of wearable devices provides the opportunity for researchers to utilize big health data for public health emergency surveillance. In this study, a COVID-19 prediction framework using the health data from wearable devices was put forward. The proposed model could predict the epidemic trend of COVID-19 outbreak in various countries and cities. The results from the study may shed light on a nationwide solution for the infectious disease surveillance system.

Data Availability

The concerned sensor data cannot be shared due to user privacy. For academic purposes, anonymised region-level statistics can be shared under agreement.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Huami Corporation.

References

[1] World Health Organization, *Coronavirus Disease (COVID-2019) Situation Reports*, World Health Organization, Geneva, Switzerland, 2020, https://www.who.int/emergencies/diseases/noveloronavirus-situation-eports.

[2] Z. Wu and J. M. McGoogan, "Characteristics of and important lessons from the coronavirus disease 2019 (COVID-19) outbreak in China," *JAMA*, vol. 323, 2020.

[3] W. Guan, Z. Ni, Y. Hu et al., "Clinical characteristics of 2019 novel coronavirus infection in China," *New England Journal of Medicine*, vol. 395, pp. 1708–1720, 2020.

[4] J. F.-W. Chan, S. Yuan, K.-H. Kok et al., "A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster," *The Lancet*, vol. 395, no. 10223, pp. 514–523, 2020.

[5] N. Chen, M. Zhou, X. Dong et al., "Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study," *The Lancet*, vol. 395, no. 10223, pp. 507–513, 2020.

[6] Z. Yang, Z. Zeng, K. Wang et al., "Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions," *Journal of Thoracic Disease*, vol. 12, 2020.

[7] J. T. Wu, K. Leung, and G. M. Leung, "Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study," *The Lancet*, vol. 395, no. 10225, pp. 689–697, 2020.

[8] S. Zhao, Q. Lin, J. Ran et al., "Preliminary estimation of the basic reproduction number of novel coronavirus (2019-nCoV) in China, from 2019 to 2020: a data-driven analysis in the early phase of the outbreak," *International Journal of Infectious Diseases*, vol. 92, pp. 214–217, 2020.

[9] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2009.

[10] New York Times, *Google Uses Searches to Track flu's Spread*, New York Times, New York, NY, USA, 2008, http://www.nytimes.com/2008/11/12/technology/internet/12flu.html?_rp1.

[11] A. F. Dugas, Y.-H. Hsieh, S. R. Levin et al., "Google Flu Trends: correlation with emergency department influenza rates and crowding metrics," *Clinical Infectious Diseases*, vol. 54, no. 4, pp. 463–469, 2012.

[12] M. J. Paul, M. Dredze, and D. Broniatowski, "Twitter improves influenza forecasting," *PLoS Currents*, vol. 6, 2014.

[13] Q. Yuan, E. O. Nsoesie, B. Lv et al., "Monitoring influenza epidemics in China with search query from Baidu," *PLoS One*, vol. 8, no. 5, 2013.

[14] K. Liu, T. Wang, Z. Yang et al., "Using Baidu search index to predict Dengue outbreak in China," *Scientific Reports*, vol. 638040 pages, 2016.

[15] M. Santillana, A. T. Nguyen, M. Dredze et al., "Combining search, social media, and traditional data sources to improve influenza surveillance," *PLoS Computational Biology*, vol. 11, no. 10, 2015.

[16] Q. Xu, Y. R. Gel, L. L. R. Ramirez et al., "Forecasting influenza in Hong Kong with Google search queries and statistical model fusion," *PLoS One*, vol. 12, no. 5, 2017.

[17] D. Lazer, R. Kennedy, G. King, and A. Vespignani, "The parable of Google Flu: traps in big data analysis," *Science*, vol. 343, no. 6176, pp. 1203–1205, 2014.
[18] Y. Liu, H. Wang, W. Zhao, M. Zhang, H. Qin, and Y. Xie, "Flexible, stretchable sensors for wearable health monitoring: sensing mechanisms, materials, fabrication strategies and features," *Sensors*, vol. 18, no. 2, p. 645, 2018.
[19] D. Dias and J. Paulo Silva Cunha, "Wearable health devices-vital sign monitoring, systems and technologies," *Sensors*, vol. 18, no. 8, p. 2414, 2018.
[20] T. Arakawa, "Recent research and developing trends of wearable sensors for detecting blood pressure," *Sensors*, vol. 18, no. 9, p. 2772, 2018.
[21] J. M. Radin, N. E. Wineinger, E. J. Topol, and S. R. Steinhubl, "Harnessing wearable device data to improve state-level real-time surveillance of influenza-like illness in the USA: a population-based study," *The Lancet Digital Health*, vol. 2, no. 2, pp. e85–e93, 2020.
[22] X. Li, J. Dunn, D. Salins et al., "Digital Health: tracking physiomes and activity using wearable biosensors reveals useful health-related information," *PLoS Biology*, vol. 15, no. 1, Article ID e2001402, 2017.
[23] C. Huang, Y. Wang, X. Li et al., "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," *The Lancet*, vol. 395, no. 10223, pp. 497–506, 2020.
[24] D. Wang, B. Hu, C. Hu et al., "Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan, China," *JAMA*, vol. 323, 2020.
[25] Z. Xu, L. Shi, Y. Wang et al., "Pathological findings of COVID-19 associated with acute respiratory distress syndrome," *The Lancet Respiratory Medicine*, vol. 8, 2020.
[26] J. Karjalainen and M. Viitasalo, "Fever and cardiac rhythm," *Archives of Internal Medicine*, vol. 146, no. 6, pp. 1169–1171, 1986.
[27] L. Faust, K. Feldman, S. M. Mattingly et al., "Deviations from normal bedtimes are associated with short-term increases in resting heart rate," *Npj Digital Medicine*, vol. 3, no. 1, pp. 1–9, 2020.
[28] G. Ke, Z. Xu, J. Zhang et al., "DeepGBM: a deep learning framework distilled by GBDT for online prediction Tasks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 384–394, Anchorage, AK, USA, August 2019.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 6195189, 11 pages
https://doi.org/10.1155/2020/6195189

Research Article

Conceptual Cognitive Modeling for Fine-Grained Annotation Quality Assessment of Object Detection Datasets

Lei Guo,[1] **Xinying Xu**,[2] **Gang Xie**,[2,3,4] **and Jerry Gao**[2,5]

[1]*College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China*
[2]*College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan 030024, China*
[3]*School of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China*
[4]*Shanxi Key Laboratory of Advanced Control and Intelligent Information System, Taiyuan University of Science and Technology, Taiyuan 030024, China*
[5]*Department of Computer Engineering, San Jose State University, San Jose, CA, USA*

Correspondence should be addressed to Gang Xie; xiegang@tyut.edu.cn and Jerry Gao; jerry.gao@sjsu.edu

Received 29 February 2020; Accepted 8 April 2020; Published 5 May 2020

Guest Editor: Jianbiao Zhang

In many supervised computer vision tasks such as object detection, manual annotation crowdsourcing platforms are widely used for acquiring large-scale labeled data. However, the annotation quality may suffer low quality that can severely affect the training of models. As a result, the evaluation of the annotations within the dataset is critical, yet it has seldom been addressed in object detection. In this paper, we present a fine-grained annotation quality assessment (FGAQA) framework for evaluating the quality of object detection datasets. First, we formulate a generic annotation quality assessment framework based on the core general-purpose data quality dimensions, using the bounding box and the label. Second, cognition theory in terms of hierarchy and continuity is utilized to refine the basic framework, including the consistency of the bounding box, completeness of the category, hierarchical accuracy of the label, and the consistency of the label. Comprehensive experiments on the two object detection datasets are used for performance evaluation. It is found that the ground truth annotations of the Urban Traffic Surveillance dataset have more quality issues than the ones of the PASCAL VOC 2007 detection dataset. The proposed FGAQA framework performs an effective fine-grained evaluation of the annotations, which is significant for quality assurance of annotations from crowdsourcing platforms and the subsequent model's training.

1. Introduction

In supervised learning, annotation quality plays a vital role in training and assessment of the models for several computer vision tasks such as object classification [1, 2], detection [3–6], and segmentation [7–9]. The training of object detection models relies on accurate and sufficient annotations. For large-scale object detection datasets, annotations are usually obtained through crowdsourcing platforms, which results from anonymous participants, and can be collected for efficiency [10–12]. However, due mainly to the untrained participants involved in the professional and time-consuming annotation tasks, this has inevitably led to subjective inconsistency and relatively low quality of the collected annotations. As a result, the annotation quality cannot be guaranteed, where the quality assessment of such annotations becomes a challenge in this context.

Annotation quality in object detection is a specialized-purpose data quality problem. Data quality has been widely studied since the 1980s [13]. According to [14], data quality can be defined as the degree to which a set of characteristics of data fulfills the requirements. Data with high quality should represent the real-world entities accurately in the structure and fit for their intended uses. Besides, data quality is of multidimensional characteristics. By reviewing the related literature [14–19], a core set of data quality

dimensions is defined, including the completeness, accuracy, and consistency. Moreover, there are a fair number of researches about annotation quality. Regarding the annotation quality in classification, accuracy is employed generally [20], not considering the hierarchy of categories. For annotation quality in object detection, quality is evaluated by Intersection-over-Union (IoU) [21]. IoU is the ratio of the intersection area of the ground truth and human annotation to the total area, only considering the quality of the bounding box [22]. There are few systematic researches about annotation quality of object detection. Consequently, we refer to general-purpose data quality and construct an annotation quality framework.

To date, there are relatively few works reported on this topic. This is only addressed from the perspectives of the object category and IoU [21]. However, a few general-purpose metrics can also be applied for annotation quality assessment. And we should perform annotation quality assessment from various aspects of the two attributes: bounding box and label.

Evaluation measures for object classification, detection, and segmentation could serve as a reference for annotation quality in object detection. Regarding flat object classification, precision and recall are employed to assess the performance [23–26]. As for hierarchical object classification, distance in the tree or the directed acyclic graph (DAG) is used to assess the performance [27–30]. The distance can treat the prediction errors differently. In terms of object detection, the mAP is usually employed [31–36], integrating precision, recall, and IOU. The mAP is calculated according to the predicted results and confidence scores. However, for annotations, reasonable confidence scores are hard to obtain. As a result, in this paper, we employ the metrics of precision and recall. Regarding object segmentation, evaluation measures can be categorized into three types: area-based measures, location-based measures, and combined measures [37–41]. These image segmentation measures pay more attention to the details and the intrinsic visual characteristics. Consequently, the idea of image segmentation evaluation is introduced into the annotation quality assessment framework.

In this paper, we propose a fine-grained framework for annotation quality assessment of object detection datasets, containing three dimensions: accuracy, completeness, and consistency. First, we construct the basic quality assessment framework based on the core general-purpose data quality (DQ) measurement, including accuracy and completeness, which considers the characteristics of annotation. For consistency, we find that it is difficult to give a strict definition. Further, the relationship of classes should be considered. Previous literature indicates that the cognition of humans is hierarchical in concept [42, 43] and consistent in space-time representations [44–46]. Inspired by these observations, the consistency of bounding box, completeness of category, hierarchical accuracy of label, and consistency of label are extracted as four additional elements for annotation quality assessment. The main contributions of this paper are as follows:

(1) We present a fine-grained annotation quality assessment (FGAQA) framework for evaluating the quality of object detection datasets. By analyzing the characteristics of the attributes of the bounding box and the corresponding label, the annotation quality contains three dimensions: accuracy, completeness, and consistency.

(2) To tackle the limitations of the basic quality assessment framework, we introduce the theory of cognitive perception to analyze the annotation quality and add four elements of annotation quality, including the consistency of bounding box, completeness of category, hierarchical accuracy of the label, and consistency of label. Specifically, the hierarchical accuracy of the label can treat annotation errors distinctively and softly.

(3) Comprehensive case studies on the Urban Traffic Surveillance (UTS) dataset and the PASCAL VOC 2007 detection dataset verify the effectiveness of the proposed annotation quality assessment framework. We find that the ground truth annotations of the UTS dataset have more quality issues, compared to the ones of the PASCAL VOC 2007 detection dataset.

The rest of this paper is organized as follows. In Section 2, the proposed cognitive-driven FGAQA framework is presented in detail. Section 3 discusses experiments as two case studies on the UTS and PASCAL VOC datasets. Finally, concluding remarks and future work are given in Section 4.

2. Annotation Quality Assessment Framework

A novel annotation quality assessment framework in object detection is given in this section, which is shown in Figure 1. The annotation has two attributes: bounding box and label. Annotation quality depends on its characteristics. For the bounding box, the size, location, and quantity could have some quality issues. Regarding the label, there may exist the quality problems of value and quantity. And the annotation quality serves reference for the training of the object detection model. Therefore, we define the quality dimensions according to the quality problems and the use of annotation. Inspired by some existing work [14–19], the dimensions of completeness, accuracy, and consistency are selected as the core set of the data quality dimensions. By considering the theory of cognitive perception, we redefine some elements based on annotation characteristics. As a result, a fine-grained annotation quality assessment framework is proposed, as shown in Figure 1. The framework is constructed from the views of the bounding box and label. Regarding the quality of the bounding box, completeness, accuracy, and consistency are defined. The completeness of the bounding box can be divided into the completeness of the bounding box's quantity and the completeness of the bounding box's size. In terms of the quality of the label, we define completeness, accuracy, and consistency. The completeness of the label consists of the completeness of the bounding box's label and the completeness of the category. The accuracy of

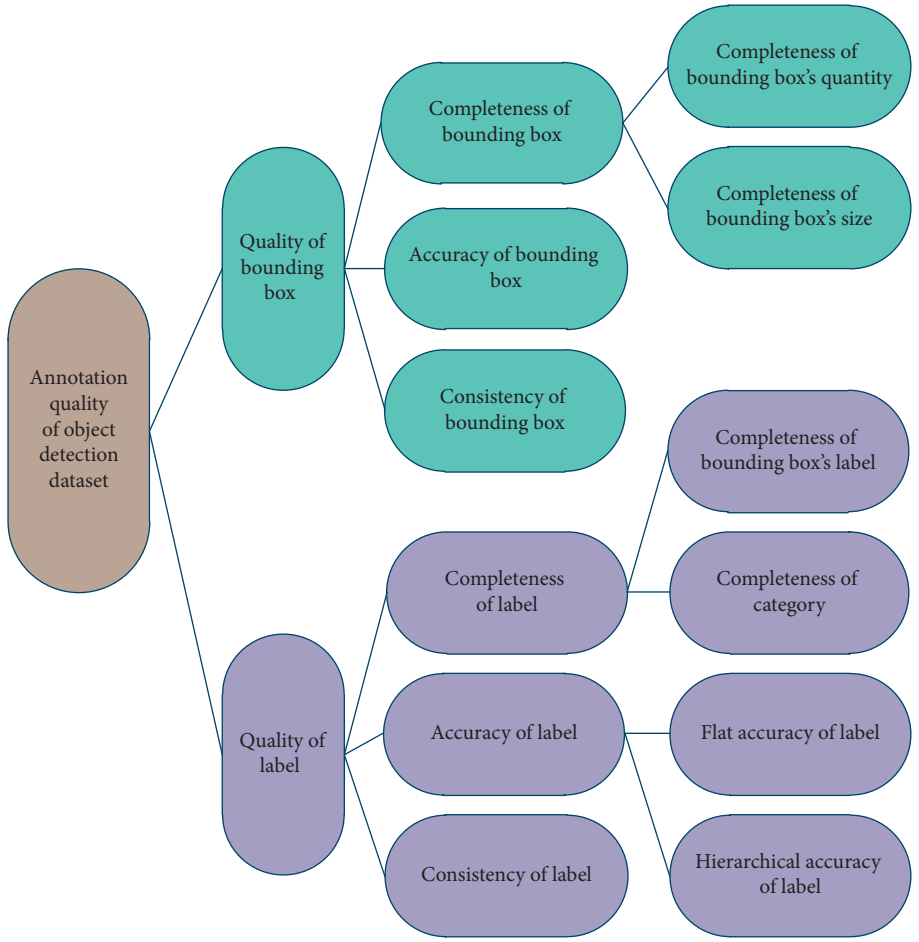


Figure 1: Annotation quality evaluation framework.

the label contains flat and hierarchical accuracy. And most of these dimensions are computed for every object and are averaged for an image and the total dataset.

2.1. Annotation Quality of Bounding Box's Quantity

2.1.1. Completeness of Bounding Box. The dimension can be defined as the extent to which bounding boxes are of sufficient quantity and coverage degree for the object. The dimension of completeness focuses on the null values. As for the completeness of the bounding box's quantity, the null values correspond to unannotated objects. In an object detection dataset, small objects are often be neglected. During the modeling process of object detection, the unannotated objects would be regarded as background. For the completeness of the bounding box's size, the null values correspond to the uncovered areas of the bounding boxes.

(1) Completeness of bounding box's quantity: for image i, completeness of bounding box's quantity is a metric that can be defined as follows:

$$\mathrm{CB}_i^{\mathrm{Quantity}} = \frac{n_i^{\mathrm{Hu}}}{n_i}, \tag{1}$$

where n_i is the true object number and n_i^{Hu} is the number of human annotations, namely, the number of bounding boxes. For the dataset, $\mathrm{CB}^{\mathrm{Quantity}}$ is

$$\mathrm{CB}^{\mathrm{Quantity}} = \frac{\sum_{i=1,\ldots,N}^{N} \mathrm{CB}_i^{\mathrm{Quantity}}}{N}, \tag{2}$$

where N is the number of images in the dataset.

(2) Completeness of bounding box's size: the completeness of the bounding box's size is a pixel-count-based metric and can be defined as follows. For the j^{th} object in image i, the metric is

$$\mathrm{CB}_{ij}^{\mathrm{Size}} = \frac{S_{ij}^{\mathrm{Int}}}{S_{ij}^{\mathrm{Obj}}}, \tag{3}$$

where S_{ij}^{Int} is the intersection area of the object and bounding box, and S_{ij}^{Obj} is the area of the object. For image i, $\mathrm{CB}_i^{\mathrm{Size}}$ is

$$\mathrm{CB}_i^{\mathrm{Size}} = \frac{\sum_{j=1,\ldots,n_i^{\mathrm{Hu}}}^{n_i^{\mathrm{Hu}}} \mathrm{CB}_{ij}^{\mathrm{Size}}}{n_i^{\mathrm{Hu}}}. \tag{4}$$

For the dataset, $\mathrm{CB}^{\mathrm{Size}}$ is

$$\mathrm{CB}^{\mathrm{Size}} = \frac{\sum_{i=1,\ldots,N}^{N} \mathrm{CB}_i^{\mathrm{Size}}}{N}. \tag{5}$$

2.1.2. Accuracy of Bounding Box. The dimension is intended to measure the closeness of the bounding box to the object. When the accuracy is low, the bounding box contains too much background affecting the distinction between the object and the background. For the bounding box of j^{th} object in image i, the accuracy is

$$\mathrm{Acc}\, B_{ij} = \frac{S_{ij}^{\mathrm{Int}}}{S_{ij}^{\mathrm{BB}}}, \tag{6}$$

where S_{ij}^{BB} is the area of the bounding box. In image i, the accuracy is

$$\mathrm{Acc}\, B_i = \frac{\sum_{j=1,\ldots,n_i^{\mathrm{Hu}}}^{n_i^{\mathrm{Hu}}} \mathrm{Acc}\, \mathrm{B}_{ij}}{n_i^{\mathrm{Hu}}}. \tag{7}$$

For a dataset, the accuracy can be given as follows:

$$\mathrm{Acc}\, B = \frac{\sum_{i=1,\ldots,N}^{N} \mathrm{Acc}\, B_i}{\sum_{i=1,\ldots,N}^{N} n_i^{\mathrm{Hu}}}. \tag{8}$$

2.1.3. Consistency of Bounding Box. The dimension focuses on the violation of spatiotemporal continuity of size and location. In crowdsourcing platforms, bounding boxes in adjacent frames may be drawn by different workers. As a result, they could conflict in size and location. Faced with the case, we can perform a quality assessment of the consistency of the bounding box during the corresponding postprocessing. Afterward, the annotations would satisfy the constraints. Concretely, for example, if an object moves toward the camera parallelly, the constraints are as follows:

$$\begin{cases} x_{\mathrm{center}}^{\mathrm{previous}} \approx x_{\mathrm{center}}^{\mathrm{current}} \approx x_{\mathrm{center}}^{\mathrm{next}}, \\ y_{\mathrm{center}}^{\mathrm{previous}} \le y_{\mathrm{center}}^{\mathrm{current}} \le y_{\mathrm{center}}^{\mathrm{next}}, \\ w^{\mathrm{previous}} \le w^{\mathrm{current}} \le w^{\mathrm{next}}, \\ h^{\mathrm{previous}} \le h^{\mathrm{current}} \le h^{\mathrm{next}}, \end{cases} \tag{9}$$

where x_{center} and y_{center} are the coordinates for the center of the bounding box, and w and h are the width and height of the bounding box. When the j^{th} object in image i satisfies the constraints, the metric $\mathrm{Con}\, B_{ij} = 1$. Otherwise, $\mathrm{Con}\, B_{ij} = 0$. For image i, the consistency is

$$\mathrm{Con}\, B_i = \frac{\sum_{j=1,\ldots,n_i^{\mathrm{Hu}}}^{n_i^{\mathrm{Hu}}} \mathrm{Con}\, B_{ij}}{n_i^{\mathrm{Hu}}}. \tag{10}$$

For the dataset, Con$B$ is

$$\mathrm{Con}\, B = \frac{\sum_{i=1,\ldots,N}^{N} \mathrm{Con}\, B_i}{N}. \tag{11}$$

2.2. Annotation Quality of Label

2.2.1. Completeness of Label. The dimension can be split into two types. The completeness of the bounding box's label is employed to measure if each box has a label. The completeness of category describes the completeness for the category's quantity from the aspect of computational learning theory. In the common benchmarks for object detection, there exist minority categories. For a category, if the metric does not meet the requirement, the detection accuracy would be affected.

(1) Completeness of bounding box's label: for image i, the completeness is

$$\mathrm{CL}_i = \frac{n_i^{\mathrm{Label}}}{n_i^{\mathrm{Hu}}}, \tag{12}$$

where n_i^{Label} is the number of labels. For a dataset, the metric is

$$\mathrm{CL} = \frac{\sum_{i=1,\ldots,N}^{N} \mathrm{CL}_i}{N}. \tag{13}$$

(2) Completeness of category: the completeness of category is a metric that measures whether the number of samples can meet the training for the object detection model. As for a dataset, the classes are usually organized in a semantic hierarchy tree. Regarding a leaf node, if it meets the condition $n^{\mathrm{leaf}} > n^{\mathrm{lowbound}}$, the completeness is 1. Otherwise, the completeness is 0. For a parent node, the completeness is

$$\mathrm{CC}_{\mathrm{parent}}^{\mathrm{Label}} = \frac{\sum_{k=1,\ldots,n^{\mathrm{child}}}^{n^{\mathrm{child}}} CC_k^{\mathrm{Label}}}{n^{\mathrm{child}}}, \tag{14}$$

where n^{child} is the number of the corresponding child nodes. As a result, we can have the completeness of the category for a dataset.

2.2.2. Accuracy of Label. The dimension is employed to measure the closeness of the human and ground truth annotations. Regarding a dataset collected by a crowdsourcing annotation platform, the label noise is the most common error and has a direct influence on the training of the object detection model. The dimension has two elements: flat accuracy and hierarchical accuracy. The flat accuracy of the label is the usual element. However, the label space is often hierarchical. The hierarchical element can treat annotation errors distinctively and is the foundation of the utilization of annotation errors. As a result, we introduce these two kinds of elements for label accuracy evaluation.

(1) Flat accuracy of label: the flat accuracy of the label includes two metrics: precision and recall. The precision and recall of class t are

$$\begin{aligned} P_t &= \frac{\mathrm{tp}_t}{\mathrm{tp}_t + \mathrm{fp}_t}, \\ R_t &= \frac{\mathrm{tp}_t}{n_t^{\mathrm{GTr}}}, \end{aligned} \tag{15}$$

where n_t^{GTr} is the number of ground truth annotations for class t, and tp_t and fp_t are the numbers of true

positive objects and false-positive objects, respectively. For a dataset, precision can be calculated as follows:

$$P = \frac{\sum_{t=1,\ldots,M}^{M} P_t}{M}, \tag{16}$$

which treats each class equally. And similarly, the recall is obtained.

(2) Hierarchical accuracy of label: the element also has two metrics. The metrics of class t are

$$\begin{aligned} \mathrm{HP}_t &= \frac{\sum_{k=1,\ldots,n_i^{\mathrm{Hu}}}^{n_i^{\mathrm{Hu}}} \left(\left|\mathrm{ans}(C_k) \cap \mathrm{ans}(C_k')\right| / \left|\mathrm{ans}(C_k')\right|\right)^{1/p}}{n_i^{\mathrm{Hu}}}, \\ \mathrm{HR}_t &= \frac{\sum_{k=1,\ldots,n_i^{\mathrm{GTr}}}^{n_i^{\mathrm{GTr}}} \left(\left|\mathrm{ans}(C_k) \cap \mathrm{ans}(C_k')\right| / \left|\mathrm{ans}(C_k)\right|\right)^{1/p}}{n_i^{\mathrm{GTr}}}, \end{aligned} \tag{17}$$

where n_i^{Hu} and n_i^{GTr} are the corresponding numbers of human and ground truth annotations, C_k and C_k' denote the ground truth and human annotation labels, and ans(C) is the operation for computing ancestors for class C, $p > 0$. Then, via macroaveraging the metrics for all classes, the hierarchical precision and recall can be calculated.

2.2.3. Consistency of Label. Similar to the consistency of the bounding box, consistency of label concentrates on the confliction of spatiotemporal continuity of label. In the crowdsourcing platform, the labels in the adjacent frames often conflict due to the existence of low-level workers. If the label of an object is consonant with the labels in the previous and next frames, the metric $\mathrm{Con}\,L_{\mathrm{object}}$ is 1; otherwise, $\mathrm{Con}\,L_{\mathrm{object}}$ is 0. For image i, the consistency is

$$\mathrm{Con}\,L_i = \frac{\sum_{j=1,\ldots,n_i^{\mathrm{Label}}}^{n_i^{\mathrm{Label}}} \mathrm{Con}\,L_{ij}}{n_i^{\mathrm{Label}}}. \tag{18}$$

For the dataset, $\mathrm{Con}\,L$ is

$$\mathrm{Con}\,L = \frac{\sum_{i=1,\ldots,N}^{N} \mathrm{Con}\,L_i}{N}. \tag{19}$$

3. Case Study

To verify the effectiveness of the quality framework, two case studies are conducted based on the UTS dataset [47] and PASCAL VOC 2007 detection dataset [48]. UTS dataset is a video dataset with varying illumination conditions and viewpoints. PASCAL VOC 2007 dataset is an image dataset and contains twenty categories. Note that a few dimensions of the quality assessment framework are not fit for the dataset. To acquire the annotations, we let a group of students fulfill the annotation work. Generally, ground truth annotations are employed as golden standard annotations. However, in the evaluation process, we find that, to a certain extent, the ground truth annotations have quality problems, especially for the UTS dataset. Consequently, ground truth annotations are evaluated, where human annotations are regarded as "ground truth annotations." Additionally, to verify the completeness of category, the relationship between this metric and detection performance is studied by conducting object detection experiments.

3.1. Case Study for UTS Dataset. In this case study, the UTS dataset is utilized for verification. To reduce the amount of annotation labor, four shots are selected, and we annotate an image for every four or five images. Finally, the numbers of images in the four shots are 75, 120, 100, and 120 with 1166, 686, 639, and 919 objects, respectively. The evaluation is presented from the aspects of an image and a dataset. We find that the ground truth annotations have quality problems, especially for the completeness of the bounding box's quantity and the flat recall of the label.

3.1.1. Annotation Quality of an Image. For the clarity of the description of annotation quality, an image is selected for evaluation, which is given in Figure 2. The semantic hierarchy tree we defined is presented in Figure 3. The quality evaluation results for an image are given in Table 1. The accuracy of the bounding box for each object is shown in Figure 4.

Now, the analysis is given below. According to Table 1, the flat precision of hatchback is 0.25. However, it is because of the quality problems of ground truth annotations. Reviewing the annotations, we find that there are two small unannotated objects as shown in Figure 2. Hierarchical measures can reflect the relation of the classes. For instance, hierarchical precision for the hatchback is 0.42, while the flat precision is 0.25. Further, the consistency of the label is less than 1. It shows that there are inconsistent labels with the labels in adjacent frames. In Table 1, four metrics are equal to 1, reflecting that there is no error from these aspects.

3.1.2. Annotation Quality of Human and Ground Truth Annotations. Afterward, we show the annotation quality of the UTS dataset for the human and ground truth annotations. The annotation accuracies of the label are given in Tables 2 and 3. The completeness of the category of the ground truth annotations for each class and the original vehicle dataset is given in Figure 3, where the threshold is set

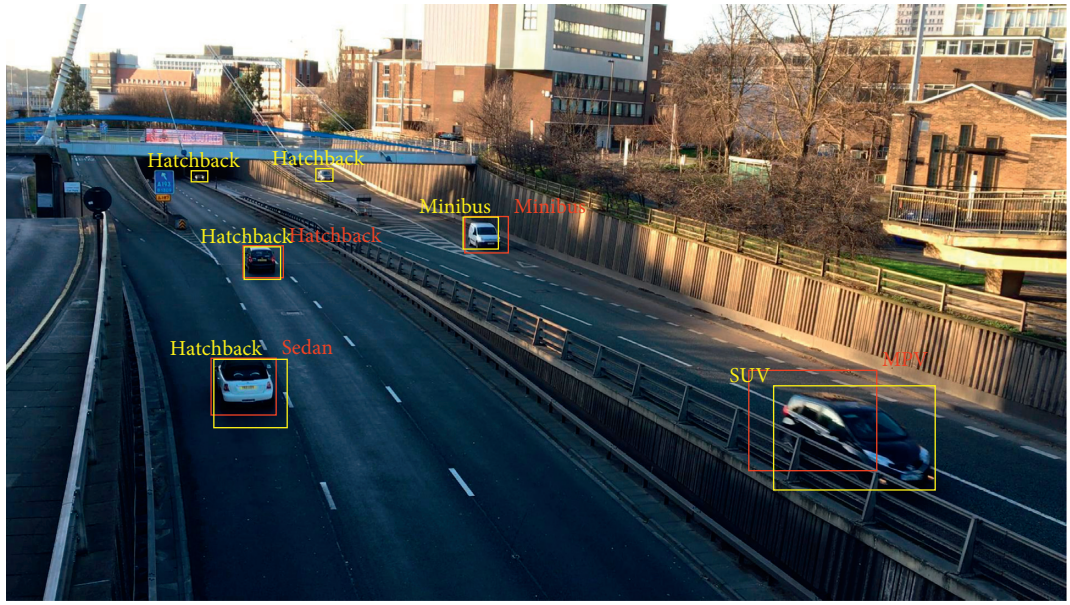


Figure 2: Human and ground truth annotations from the UTS dataset (ground truth and human annotations are shown in red and yellow, respectively).

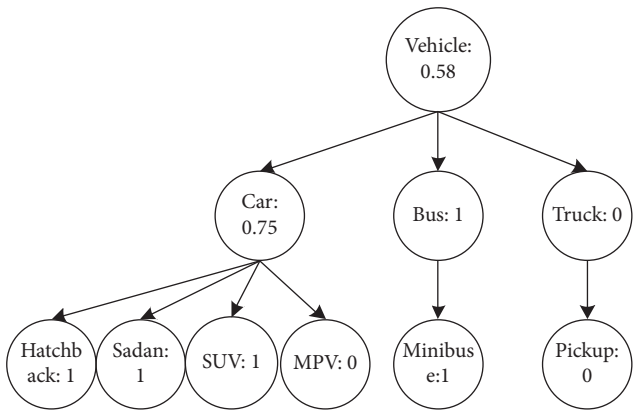


Figure 3: Semantic hierarchy tree and completeness of category of ground truth annotations for the original UTS training dataset.

Table 1: Results of other quality dimensions for an image.

Annotation quality dimension	Value
Completeness of bounding box's quantity	1
Completeness of bounding box's size	0.64
Accuracy of bounding box	0.67
Consistency of bounding box	1
Completeness of bounding box's label	1
Flat precision/recall of label	–/0.5
Hierarchical precision/recall of label	0.69/0.79
Flat precision/recall of hatchback	0.25/1
Hierarchical precision/recall of hatchback	0.42/0.83
Consistency of label	0.84

to 1000. The results of other quality dimensions are presented in Table 4.

The quality of human annotations is analyzed first. According to Tables 2 and 4, the overall annotation quality of the bounding box is good, while the annotation quality of the label is relatively poor. Accordingly, it can be inferred that the label's annotation is a more difficult task. In particular, for SUV and MPV, the accuracy and recall are too low. The hierarchical accuracy is higher than the flat accuracy, treating errors distinctively. According to Table 4, compared with other dimensions, the consistency of the label is lower on account of the own property.

The quality of ground truth annotations is evaluated here. According to Tables 2–4, the completeness of bounding box's quantity, flat and hierarchical recall of label, and consistency of label for ground truth annotations are lower than those for human annotations. When reviewing ground truth annotations, we find that ground truth annotations neglect some small and incomplete objects. But these small and incomplete objects can be annotated properly by experience. There are more inconsistent labels in ground truth annotations than in human annotations. Figure 3 shows that the completeness of category for MPV and pickup is 0, as the corresponding category's quantities do not reach the threshold. Generally, the quality problem exists in the ground truth annotations. Therefore, it is significant to perform a quality assessment in the process of annotation and ground truth inference.

3.1.3. Relationship between the Completeness of Category and Detection Performance. For the sake of exploring the relationship between the completeness of category and detection performance, the following experiment is conducted, which implies the effectiveness of the dimension. The object detection experiment on the UTS dataset is performed on the original dataset and downsampled dataset. As for downsampling, we just select images for every two images. The detection algorithm we use is Faster RCNN [3]. Table 5 presents the corresponding result.

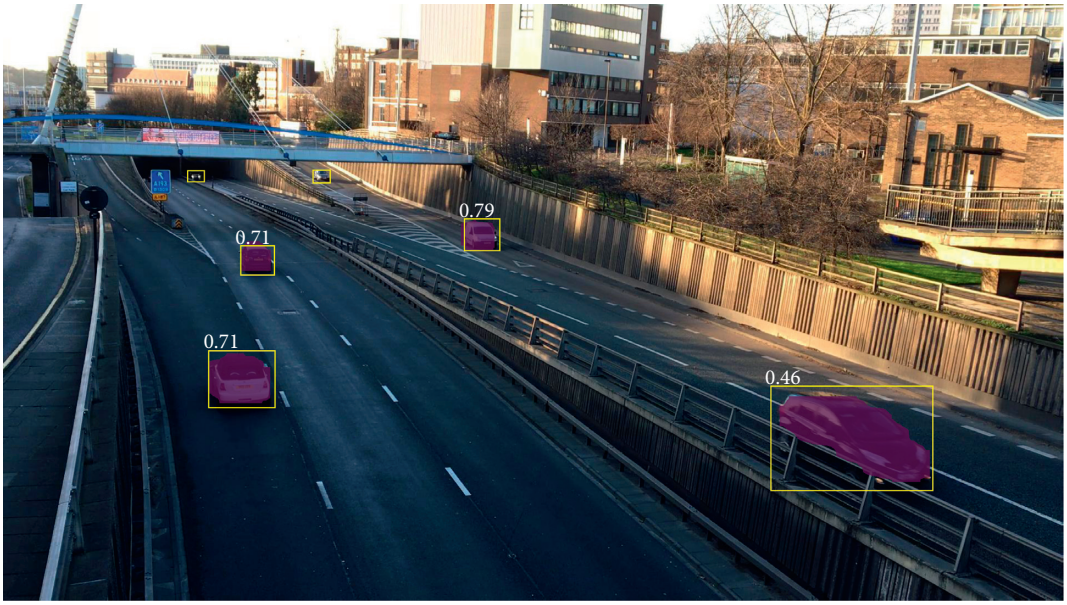


Figure 4: Accuracy of the bounding box in an image (note that two small objects are missed by the image instance segmentation algorithm.).

Table 2: Annotation accuracy of human annotations for the downsampled UTS dataset.

Class	Flat accuracy of label		Hierarchical accuracy of label	
	Precision	Recall	Precision	Recall
Hatchback	0.79 ± 0.02	0.55 ± 0.14	0.92 ± 0.01	0.81 ± 0.07
Sedan	0.58 ± 0.08	0.78 ± 0.12	0.86 ± 0.03	0.88 ± 0.07
Minibus	0.93 ± 0.10	0.57 ± 0.33	0.95 ± 0.06	0.70 ± 0.22
SUV	0.20 ± 0.06	0.27 ± 0.10	0.69 ± 0.03	0.74 ± 0.04
MPV	0.19 ± 0.14	0.26 ± 0.14	0.62 ± 0.14	0.72 ± 0.08
Pickup	0.57 ± 0.41	1 ± 0	0.72 ± 0.27	1 ± 0
On average	0.55 ± 0.11	0.57 ± 0.07	0.79 ± 0.07	0.81 ± 0.05

Table 3: Annotation accuracy of ground truth annotations for the downsampled UTS dataset.

Class	Flat accuracy of label		Hierarchical accuracy of label	
	Precision	Recall	Precision	Recall
Hatchback	0.57 ± 0.13	0.57 ± 0.04	0.85 ± 0.06	0.67 ± 0.05
Sedan	0.79 ± 0.12	0.44 ± 0.04	0.91 ± 0.08	0.65 ± 0.02
Minibus	0.58 ± 0.34	0.77 ± 0.17	0.72 ± 0.23	0.79 ± 0.17
SUV	0.27 ± 0.10	0.15 ± 0.05	0.75 ± 0.04	0.53 ± 0.09
MPV	0.26 ± 0.14	0.17 ± 0.15	0.72 ± 0.08	0.51 ± 0.18
Pickup	1 ± 0	0.28 ± 0.19	1 ± 0	0.28 ± 0.19
On average	0.58 ± 0.07	0.40 ± 0.07	0.83 ± 0.04	0.57 ± 0.07

Table 4: Results of other quality dimensions for the downsampled UTS dataset.

Annotation quality dimension	Human annotations	Ground truth annotations
Completeness of bounding box's quantity	0.98 ± 0.02	0.75 ± 0.05
Completeness of bounding box's size	0.96 ± 0.02	0.99
Consistency of bounding box	0.977 ± 0.002	0.96
Completeness of bounding box's label	0.52 ± 0.11	0.58
Consistency of label	0.86 ± 0.01	0.71

Table 5: Comparison of detection results based on the original training dataset and downsampled dataset.

Class	Object number in the training dataset	mAP (original)	mAP (downsampled)
Hatchback	12165	0.669	0.744
Sedan	5484	0.573	0.565
Minibus	3220	0.663	0.601
SUV	1761	0.560	0.576
MPV	898	0.154	0.142
Pickup	263	0.020	0.0001
On average	3965.2	0.440	0.438

According to Table 5, we argue that the detection result is closely related to the completeness of category. Overall, for the complete class whose training samples' quantity is over 1000, the corresponding mAP is high, while the detection mAPs of other classes are quite low. However, for SUV in the downsampled dataset, the quantity is about 880. The detection performance is still acceptable. It is due to its salient visual feature. Thus, the threshold varies with the class. Additionally, for the incomplete class, the performance declines with downsampling.

3.2. Case Study for PASCAL VOC 2007 Detection Dataset. In the case study, PASCAL VOC 2007 detection dataset is utilized for verification. To save labor, we select twenty images for each class as annotation samples. Finally, a random-selected dataset containing 353 images is obtained. The PASCAL VOC 2007 dataset is an image dataset. Consequently, a few quality dimensions are not fit for the dataset.

3.2.1. Annotation Quality for Human and Ground Truth Annotation. The quality of human and ground truth annotations for the PASCAL VOC 2007 dataset is given below. Accuracies of the label for the human and ground truth annotations are given in Tables 6 and 7. The semantic hierarchy tree and completeness of category quantity are given in Figure 5, where the threshold is set as 400. The results of other quality dimensions are provided in Table 8.

According to Tables 6 and 8, we can see that the human annotation quality for the dataset is good overall. However, the accuracies of the chair, potted plant, and dining table are relatively poor. For instance, the average flat recall for the potted plant is 0.54. This is because the potted plant is small and tends to be neglected. And for the other dimensions of human annotations, quality is relatively reliable.

Afterward, we evaluate the annotation quality of ground truth annotations. According to Tables 6–8, we find that the quality of ground truth annotations is slightly worse than that of human annotations. Specifically, the completeness of the bounding box's quantity and the flat recall of the label are relatively low. These dimensions indicate that there are more unannotated objects. As there are not enough images in the random-selected dataset, we calculate the completeness of category according to the original training set. The total completeness of category is 0.62, as 38% of the classes do not have enough samples.

Table 6: Annotation accuracy of human annotations for the selected images of the PASCAL 2007 dataset (the average values are computed for the twenty classes).

Class	Flat accuracy of label		Hierarchical accuracy of label	
	Precision	Recall	Precision	Recall
Person	0.98 ± 0.01	0.92 ± 0.05	0.99 ± 0.01	0.92 ± 0.04
Car	0.99 ± 0.02	0.94 ± 0.04	0.99 ± 0.01	0.94 ± 0.03
Chair	0.96 ± 0.03	0.74 ± 0.08	0.98 ± 0.01	0.82 ± 0.07
Bottle	0.98 ± 0.01	0.81 ± 0.08	0.99 ± 0.01	0.82 ± 0.07
Potted plant	1 ± 0	0.54 ± 0.31	1 ± 0	0.57 ± 0.29
Cow	0.99 ± 0.01	0.95 ± 0.01	0.997 ± 0.005	0.96 ± 0.01
Dining table	0.75 ± 0.16	0.59 ± 0.18	0.91 ± 0.06	0.64 ± 0.15
Bus	1 ± 0	0.94 ± 0.04	1 ± 0	0.96 ± 0.03
On average	0.96 ± 0.01	0.89 ± 0.04	0.983 ± 0.005	0.90 ± 0.04

Table 7: Annotation accuracy of ground truth annotations for the selected images of the PASCAL 2007 dataset (the average values are computed for the twenty classes).

Class	Flat accuracy of label		Hierarchical accuracy of label	
	Precision	Recall	Precision	Recall
Person	0.97 ± 0.01	0.79 ± 0.11	0.98 ± 0.01	0.81 ± 0.1
Car	0.94 ± 0.02	0.82 ± 0.08	0.96 ± 0.01	0.83 ± 0.08
Chair	0.90 ± 0.02	0.81 ± 0.09	0.96 ± 0.01	0.84 ± 0.08
Bottle	1 ± 0	0.74 ± 0.10	1 ± 0	0.81 ± 0.09
Potted plant	0.98 ± 0.02	0.77 ± 0.02	0.99 ± 0.01	0.78 ± 0.03
Cow	0.99 ± 0.01	0.81 ± 0.10	1 ± 0	0.83 ± 0.10
Dining table	0.68 ± 0.16	0.63 ± 0.11	0.87 ± 0.06	0.65 ± 0.09
Bus	0.91 ± 0.09	0.89 ± 0.08	0.94 ± 0.05	0.90 ± 0.07
On average	0.94 ± 0.02	0.84 ± 0.05	0.97 ± 0.01	0.87 ± 0.05

3.2.2. Relationship between the Completeness of Category and Detection Performance. To explore the relationship between the completeness of category and detection performance, an experiment is conducted in the same way as the previous section. We conduct object detection experiments on the original dataset and downsampled dataset of which the sampling ratio is 0.5. And the major classes of person, car, and chair are not downsampled. Table 9 presents the detection results, where classes are in descending order of quantity of training samples.

According to Table 9, on the whole, the detection performance declines after the dataset is downsampled. For the majority classes of person, car, and chair, there are no obvious declines of mAPs, as we do not make downsampling on these classes. As for the minority classes, mAPs for the bottle and potted plant decline a lot, which can be regarded

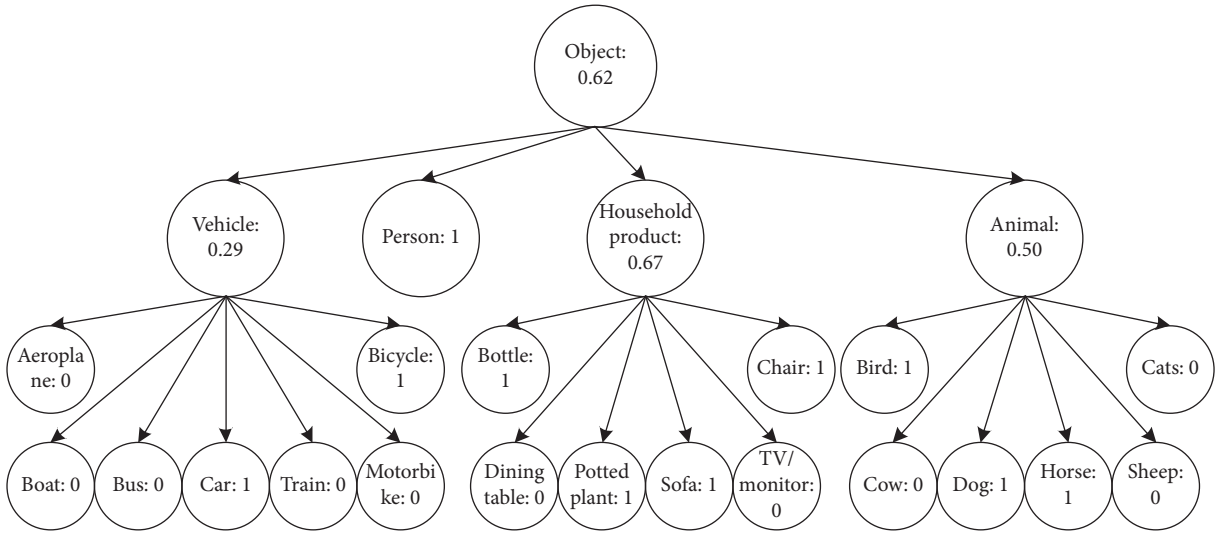


Figure 5: Semantic hierarchy tree and completeness of category for original PASCAL VOC 2007 training dataset.

Table 8: Results of other quality dimensions for the selected images and its original training dataset of the PASCAL VOC 2007 dataset.

Annotation quality dimension	Human annotations	Ground truth annotations
Completeness of bounding box's quantity	0.90 ± 0.04	0.88 ± 0.05
Completeness of bounding box's size	0.84 ± 0.02	0.85
Completeness of bounding box's label	0.9991 ± 0.0008	1

Table 9: Comparison of detection results based on the original training dataset and downsampled dataset (the average values are computed for the twenty classes).

Class	Object number in the training dataset	mAP (original)	mAP (downsampled)
Person	5447	0.779	0.778
Car	1644	0.831	0.807
Chair	1432	0.520	0.511
Bottle	634	0.576	0.519
Potted plant	625	0.459	0.376
Cow	356	0.767	0.721
Dining table	310	0.682	0.671
Bus	272	0.772	0.776
On average	783.1	0.714	0.678

as hard classes. But mAPs for the other classes of the minority are relatively high and change little, which should be regarded as easy classes. The hard classes are usually of small scale and have nonsalient visual features, hindering the learning of the object detection model. Therefore, the threshold for hard classes is relatively high. In the future process of constructing a dataset, the training samples' quantity for hard classes should be added.

4. Conclusion

Annotation quality is essential for the object detection model's training. In this paper, conceptual cognitive modeling for fine-grained annotation quality assessment is proposed. The annotation quality is calculated from the perspectives of the bounding box and label. To begin with, a generic framework based on general-purpose data quality dimensions is constructed from two aspects: the bounding box and the class label. This framework is used to assess the completeness and accuracy from the corresponding aspects. Nonetheless, the basic framework has limitations in assessing the consistency, the category's quantity, and the annotation errors. Thereupon, the cognitive theory is introduced, and we add the corresponding elements, including consistency of bounding box, hierarchical accuracy of label, consistency of label, and completeness of category. Case studies on the Urban Traffic Surveillance dataset and PASCAL VOC 2007 detection dataset indicate the validity of the framework. Currently, the annotation quality framework is constructed in an ideal condition. Future research is required to consider more practical factors.

Data Availability

The Urban Traffic Surveillance dataset and PASCAL VOC 2007 detection dataset used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Key Research and Development Plan of Shanxi Province (Nos. 201703D111027 and 201703D111023), Shanxi International Cooperation Project (No. 201803D421039), and Natural Science Foundation of Shanxi Province (No. 201801D121144).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, MIT Press, Cambridge, MA, USA, 2012.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, MIT Press, Cambridge, MA, USA, 2015.

[4] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1309–1321, 2014.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision,*, pp. 2980–2988, Venice, Italy, October 2017.

[6] Z. Fang, J. Ren, S. Marshall et al., "Triple loss for hard face detection," *Neurocomputing*, 2020.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, October 2017.

[9] X. Xu, G. Li, G. Xie, J. Ren, and X. Xie, "Weakly supervised deep semantic segmentation using CNN and ELM with semantic candidate regions," *Complexity*, vol. 2019, Article ID 9180391, 12 pages, 2019.

[10] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk," *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.

[11] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "Crowdforge: crowdsourcing complex work," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology,*, pp. 43–52, Santa Barbara, CA, USA, October 2011.

[12] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Turkit: tools for iterative tasks on mechanical turk," in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 29-30, Washington DC, USA, 2009.

[13] D. P. Ballou and H. L. Pazer, "Modeling data and process quality in multi-input, multi-output information systems," *Management Science*, vol. 31, no. 2, pp. 150–162, 1985.

[14] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Communications of the ACM*, vol. 45, no. 4, pp. 211–218, 2002.

[15] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–52, 2009.

[16] D. Firmani, M. Mecella, M. Scannapieco, and C. Batini, "On the meaningfulness of "big data quality" (invited paper)," *Data Science and Engineering*, vol. 1, no. 1, pp. 6–20, 2016.

[17] D. Ardagna, C. Cappiello, W. Samá, and M. Vitali, "Context-aware data quality assessment for big data," *Future Generation Computer Systems*, vol. 89, pp. 548–562, 2018.

[18] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Biessmann, and A. Grafberger, "Automating large-scale data quality verification," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 1781–1794, 2018.

[19] R. Zhang, M. Indulska, and S. Sadiq, "Discovering data quality problems," *Business & Information Systems Engineering*, vol. 61, no. 5, pp. 575–593, 2019.

[20] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 614–622, Las Vegas, NV, USA, August 2008.

[21] S. Vittayakorn and J. Hays, "Quality assessment for crowdsourced object annotations," in *Proceedings of the British Machine Vision Conference*, pp. 1–11, Dundee, UK, August 2011.

[22] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Proceedings 15th International Conference on Pattern Recognition (ICPR-2000)*, pp. 167–170, IEEE, Barcelona, Spain, 2000.

[23] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, 2009.

[24] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[25] W. Feng, W. Huang, and J. Ren, "Class imbalance ensemble learning based on the margin theory," *Applied Sciences*, vol. 8, no. 5, p. 815, 2018.

[26] J. Jiang, J. Kohler, C. Williams et al., "Live: an integrated production and feedback system for intelligent and interactive tv broadcasting," *IEEE Transactions on Broadcasting*, vol. 57, no. 3, pp. 646–661, 2011.

[27] S. Kiritchenko, S. Matwin, and F. Famili, "Functional annotation of genes using hierarchical text categorization," in *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology*, pp. 1–6, Detroit, MI, USA, 2005.

[28] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: a unified view and novel approaches," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 820–865, 2015.

[29] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International Conference on Machine Learning*, pp. 5075–5084, Stockholm, Sweden, July 2018.

[30] J.-Y. Park and J.-H. Kim, "Incremental class learning for hierarchical classification," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 178–189, 2018.

[31] C. Gu, J. J. Lim, P. arbeláez, and J. malik, "Recognition using regions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 1030–1037, IEEE, June2009.

[32] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proceedings of the 8th ACM International*

Workshop on Multimedia Information Retrieval, pp. 321–330, New York, NY, USA, 2006.

[33] C. Zhao, X. Li, J. Ren, and S. Marshall, "Improved sparse representation using adaptive spatial support for effective target detection in hyperspectral imagery," *International Journal of Remote Sensing*, vol. 34, no. 24, pp. 8669–8684, 2013.

[34] J. Han, D. Zhang, C. Gong, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3325–3337, 2014.

[35] Y. Xi, J. Zheng, X. Li, X. Xu, J. Ren, and G. Xie, "SR-POD: sample rotation based on principal-axis orientation distribution for data augmentation in deep object detection," *Cognitive Systems Research*, vol. 52, pp. 144–154, 2018.

[36] Z. Wang, J. Ren, D. Zhang, M. Sun, and J. Jiang, "A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos," *Neurocomputing*, vol. 287, pp. 68–83, 2018.

[37] N. Clinton, A. Holt, J. Scarborough, L. Yan, and P. Gong, "Accuracy assessment measures for object-based image segmentation goodness," *Photogrammetric Engineering & Remote Sensing*, vol. 76, no. 3, pp. 289–299, 2010.

[38] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "A measure for objective evaluation of image segmentation algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, p. 34, IEEE, San Diego, CA, USA, 2005.

[39] G. Sun, A. Zhang, J. Ren et al., "Gravitation-based edge detection in hyperspectral images," *Remote Sensing*, vol. 9, no. 6, p. 592, 2017.

[40] J. Ren, J. Jiang, D. Wang, D. Wang, and S. S. Ipson, "Fusion of intensity and inter-component chromatic difference for effective and robust colour edge detection," *IET Image Processing*, vol. 4, no. 4, pp. 294–301, 2010.

[41] X. Xie, G. Xie, X. Xu, L. Cui, and J. Ren, "Automatic image segmentation with superpixels and image-level labels," *IEEE Access*, vol. 7, pp. 10999–11009, 2019.

[42] J. M. Mandler and L. McDonough, "Concept formation in infancy," *Cognitive Development*, vol. 8, no. 3, pp. 291–318, 1993.

[43] J. L. McClelland and T. T. Rogers, "The parallel distributed processing approach to semantic cognition," *Nature Reviews Neuroscience*, vol. 4, no. 4, pp. 310–322, 2003.

[44] D. Casasanto, O. Fotakopoulou, and L. Boroditsky, "Space and time in the child's mind: evidence for a cross-dimensional asymmetry," *Cognitive Science*, vol. 34, no. 3, pp. 387–405, 2010.

[45] Y. Yan, J. Ren, G. Sun et al., "Unsupervised image saliency detection with Gestalt-laws guided optimization and visual attention based refinement," *Pattern Recognition*, vol. 79, pp. 65–78, 2018.

[46] Y. Yan, J. Ren, H. Zhao et al., "Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos," *Cognitive Computation*, vol. 10, no. 1, pp. 94–104, 2018.

[47] Yi Zhou, Li Liu, L. Shao, and M. Mellor, "DAVE: a unified framework for fast vehicle detection and annotation," in *Proceedings of the European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 278–293, October 2016.

[48] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 9242917, 11 pages
https://doi.org/10.1155/2020/9242917

Research Article

Face Detection and Segmentation Based on Improved Mask R-CNN

Kaihan Lin, Huimin Zhao, Jujian Lv, Canyao Li, Xiaoyong Liu, Rongjun Chen, and Ruoyan Zhao

School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China

Correspondence should be addressed to Huimin Zhao; zhaohuimin@gpnu.edu.cn and Jujian Lv; jujianlv@gpnu.edu.cn

Received 18 December 2019; Accepted 11 March 2020; Published 1 May 2020

Guest Editor: Zheng Wang

Deep convolutional neural networks have been successfully applied to face detection recently. Despite making remarkable progress, most of the existing detection methods only localize each face using a bounding box, which cannot segment each face from the background image simultaneously. To overcome this drawback, we present a face detection and segmentation method based on improved Mask R-CNN, named G-Mask, which incorporates face detection and segmentation into one framework aiming to obtain more fine-grained information of face. Specifically, in this proposed method, ResNet-101 is utilized to extract features, RPN is used to generate RoIs, and RoIAlign faithfully preserves the exact spatial locations to generate binary mask through Fully Convolution Network (FCN). Furthermore, Generalized Intersection over Union (GIoU) is used as the bounding box loss function to improve the detection accuracy. Compared with Faster R-CNN, Mask R-CNN, and Multitask Cascade CNN, the proposed G-Mask method has achieved promising results on FDDB, AFW, and WIDER FACE benchmarks.

1. Introduction

Face detection is a key link of subsequent face-related applications, such as face recognition [1], facial expression recognition [2], and face hallucination [3], because its effect directly affects the subsequent applications performance. Therefore, face detection has become a research hotspot in the field of pattern recognition and computer vision and has been widely studied in the past two decades.

Large amounts of approaches have been proposed for face detection. The early research on face detection [4–9] mainly focused on the design of handcraft feature and used traditional machine learning algorithms to train effective classifiers for detection and recognition. Such approaches are limited in that the efficient feature design is complex and the detection accuracy is relatively low. In recent years, face detection methods based on deep convolutional neural network [10–13] have been widely studied, which are more robust and efficient than handcraft feature methods. Besides, a series of efficient object detection frameworks are used for face detection to improve detection performance [14–18], including R-CNN [19], Fast R-CNN [20], and Faster R-CNN [21]. These methods mainly implement face detection and the location of the face bounding box, which may have some drawbacks such as the extracted face features have background noise, spatial quantization is rough and cannot be accurately positioned. These drawbacks will directly affect the follow-up subsequent face-related applications, such as face recognition, facial expression recognition, and face alignment [22]. Therefore, it is necessary to study a face detection and segmentation method.

Mask R-CNN [23], an improved object detection model based on Faster R-CNN, has an impressive performance on various object detection and segmentation benchmarks such as COCO challenges [24] and Cityscapes dataset [25]. Unlike traditional R-CNN series methods, Mask R-CNN adds a mask branch for predicting segmentation masks on each Region of Interest (RoI), which can fulfil both detection and segmentation tasks. In order to fulfil both face detection and segmentation tasks from the image to overcome the

drawbacks of the existing methods, a face detection and segmentation method based on improved Mask R-CNN (G-Mask) is proposed in this paper. In particular, our scheme introduces Generalized Intersection over Union (GIoU) [26] as the loss function for bounding box regression to improve detection accuracy of face detection. The main contributions of this paper are as follows:

(1) A new dataset was created (more details are described in Section 4.1), which annotated 5115 images randomly selected from the FDDB [27] and ChokePoint datasets [28].

(2) A face detection and segmentation method based on improved Mask R-CNN was proposed, which can detect faces correctly while also precisely segmenting each face in an image. Furthermore, the proposed method improves the detection performance by introducing GIoU as a bounding box loss function. The experimental results verify that our proposed G-Mask method achieves promising performance on several mainstream benchmarks, including the FDDB, AFW [29], and WIDER FACE [30].

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work. The G-Mask framework for face detection and segmentation is described in detail in Section 3. Section 4 presents the experiment and discussion of the proposed method. In the last section, the work is summarized and the direction of future work is proposed.

2. Related Work

Face detection as one of the important research directions of computer vision has been extensively studied in recent years. From the development process of face detection, we can simply classify previous work as handcraft feature based and neural networks based methods.

2.1. Handcraft Feature Based Methods. With the appearance of the first real-time face detection method called Viola-Jones [4] in 2004, face detection has begun to be applied in practice. The well-known Viola-Jones can perform real-time detection using Haar feature and cascaded structure, but it also has some drawbacks, such as large feature size and low recognition rate for complex situations. To address these concerns, a lot of new handcraft features are proposed, such as HOG [5], SIFT [6], SUFT [7], and LBP [8], which have achieved outstanding results. Apart from the above methods, one of the significant advances was Deformable Part Model (DPM), proposed by Felzenszwalb et al. [9]. In the DPM model, the face is represented as a set of deformable parts, and the improved HOG feature and SVM are used for detection, achieving remarkable performance. In general, the advantages of handcraft features are that the model is intuitive and extensible, and the disadvantage is that the detection accuracy is limited in the face of multi-objective tasks.

2.2. Neural Networks Based Methods. As early as 1994, Vaillant et al. [10] first proposed using neural network to detect faces. In this work, Convolutional Neural Networks (CNN) is used to classify whether each pixel is part of a face and then determine the location of the face through another CNN. After that, the researchers did a lot of research based on this work. In recent years, the deep learning approaches has significantly promoted the development of the computer vision technology, including face detection. Li et al. [11] proposed a cascade CNN network architecture for rapid face detection, which is a multiresolution network structure that can quickly eliminate background regions in the low-resolution stage and carefully evaluate challenging candidates in the last high resolution stage. Ranjan et al. [12] proposed a deformation part model based on normalized features extracted by deep convolutional neural network. Yang et al. [13] proposed a method called Convolutional Channel Feature (CCF) by combining the advantages of both filtered channel features and CNN, which has a lower computational cost and storage cost than the general end-to-end CNN method.

Recently, witnessing the significant advancement of object detection using region-based methods, researchers have gradually applied the R-CNN series of methods to face detection. Qin et al. [14] proposed a joint training scheme for CNN cascade, Region Proposal Network (RPN), and Fast R-CNN. In [15], Jiang et al. trained the Faster R-CNN model by using WIDER dataset and verified performance on the FDDB and IJB-A benchmarks. Sun et al. [16] improve the Faster R-CNN framework through a series of strategies such as multiscale training, hard negative mining, and feature concatenation. Wu et al. [17] proposed a different scales face detection method based on Faster R-CNN for the challenge of small-scale face detection. Liu et al. [18] proposed a cascaded backbone branches fully convolutional neural network (BB-FCN) and used facial landmark localization results to guide R-CNN-based face detection. The neural networks based methods are already the mainstream of face detection because of its high efficiency and stability. In this work, we propose a G-Mask scheme, which achieves fairly progress in face detection task compared to the original architecture.

3. Improved Mask R-CNN

3.1. Network Architecture. The proposed method is extended from the Mask R-CNN [23] framework, which is the state-of-the-art object detection scheme and demonstrated impressive performance on various object detection benchmarks. As stated in Figure 1, the proposed G-Mask method consists of two branches, one for face detection and the other for face and background image segmentation. In this work, the ResNet-101 backbone is used to extract the facial features of the input image, and the Region of Interest (RoI) is rapidly generated on the feature map through the Region Proposal Network (RPN). We also use the Region of Interest Align (RoIAlign) to faithfully preserve exact spatial locations and output the feature map to a fixed size. At the end of the network, the bounding box is located and classified in the

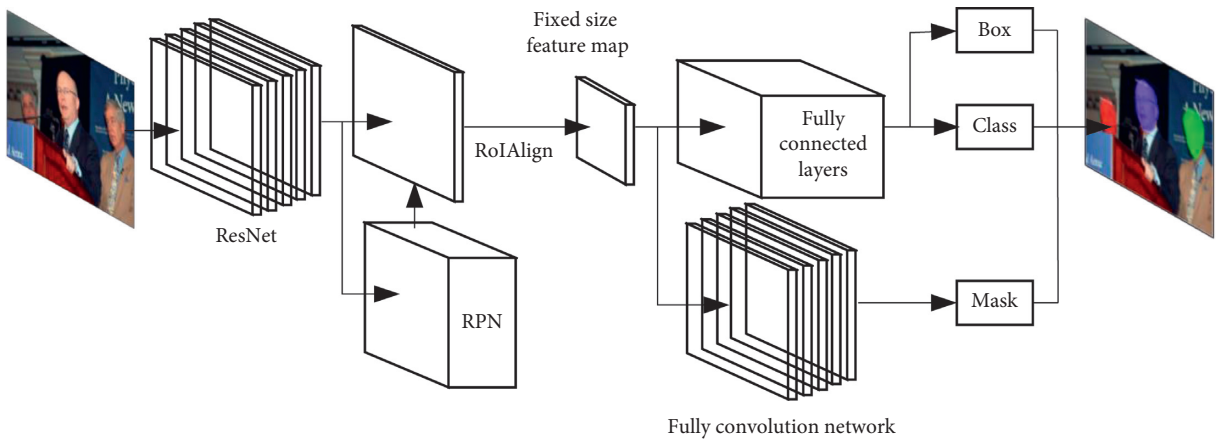


Figure 1: Network architecture of the G-Mask.

detection branch, and the corresponding face mask is generated on the image in the segmentation branch through the Fully Convolution Network (FCN) [31]. In the following, we will introduce the key steps of our network in detail.

3.2. Region Proposal Network. For images with human faces in our daily life, there are generally some face objects with different scales and aspect ratios. Therefore, in our approach, Region Proposal Network (RPN) generates RoIs by sliding windows on the feature map through anchors with different scales and different aspect ratios. Details are shown in Figure 2. The largest rectangle in the figure represents the feature map extracted by the convolutional neural network, and the dotted line indicates that the anchor is the standard anchor. Assume that the standard anchor size is 64 pixels, and the three anchors it contained represent three anchors with aspect ratios of 1 : 1, 1 : 2, and 2 : 1. The dot-dash line and the solid line represent the anchors of 32 and 128 pixels, respectively. Similarly, each of them also has three aspect ratios anchors. For traditional RPN, the above three scales and three aspect ratios are used to slide on the feature map to generate RoIs. In this paper, we use 5 scales (16^2, 32^2, 64^2, 128^2, and 256^2) and 3 aspect ratios (1 : 1, 1 : 2, and 2 : 1), leading to 15 anchors at each location, which was more effective in detecting objects of different scales.

3.3. RoIAlign Layer. G-Mask, unlike the general face detection methods, has a segmentation operation, which requires more refined spatial quantization for feature extraction. In the traditional region-based approaches, RoIPool is the standard operation for extracting small feature map from RoIs, which have two quantization operations that result in misalignments between the RoI and the extracted features. For traditional detection methods, this may not affect classification and localization, while for our approach, it has a great impact on prediction of pixel-accurate masks, as well as for small object detection.

In response to the above problem, we introduced the RoIAlign layer, following the scheme of [23]. As shown in Figure 3, suppose the feature map is divided into 2×2 bins.

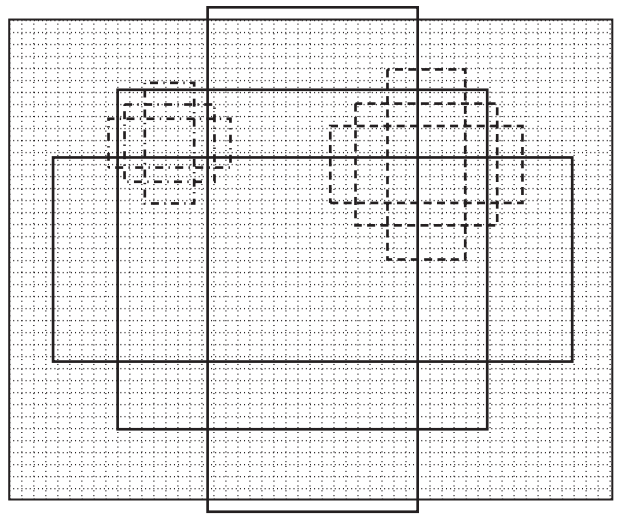


Figure 2: Illustration of RPN network.

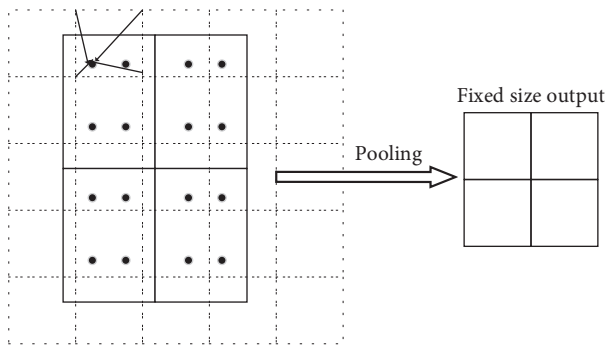


Figure 3: Bilinear interpolation in RoIAlign, where the dashed background grid represents the feature map, the solid grid represents an RoI (with 2×2 bins in this example), and the dots represent the four sample points in each bin.

It can be seen that the RoIAlign layer cancels the harsh quantization operations on the feature map and uses bilinear interpolation to preserve the floating-number coordinates, thereby avoiding misalignments between the RoI and the

extracted features. The bilinear interpolation function has two steps, which are defined as follows:

Interpolate on the x-axis direction as follows:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}), \quad R_1 = (x, y_1), \tag{1}$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}), \quad R_2 = (x, y_2). \tag{2}$$

Interpolate on the y-axis direction as follows:

$$f(P) = f(x, y) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2), \tag{3}$$

where $f(x, y)$ is the value of the sampling point P, $f(Q_{11})$, $f(Q_{12})$, $f(Q_{21})$, and $f(Q_{22})$ are the values of the four nearby grid points $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, and $Q_{22} = (x_2, y_2)$, and $f(R_1)$, $f(R_2)$ are the value obtained by interpolating in the x-axis direction.

3.4. Mask Branch. The mask branch realizes the segmentation of face object and background image in G-Mask model, which predicts the segmentation mask in a pixel to pixel manner by applying Full Convolutional Network (FCN) [31] to each RoI. The FCN scheme is one of the solutions for instance segmentation, which originates from CNN but is also different from general CNN. For the traditional CNN network architecture, in order to obtain the feature vector of fixed dimensions, the convolutional layer is generally connected with several full connection layers, and finally the output is a numerical description of the input, which is generally applicable to tasks such as image recognition and classification, object detection, and positioning. The FCN framework is similar to the traditional CNN network, which also includes the convolutional layer and the pooling layer. In particular, the FCN uses the deconvolution to up-sample the feature map in the end convolution layer so that the output image size can be restored to the original image size, and finally uses the Softmax classifier to predict the category of each pixel.

3.5. Generalized Intersection over Union. Bounding box regression, as one of the fundamental components of many computer vision tasks, deserves further study by researchers [32]. However, unlike the architecture and feature extraction strategy improvement researches, which have made great progress in recent years [33], the research of bounding box regression has lagged behind somewhat. The Generalized Intersection over Union (GIoU) [26], as the latest metric and bounding box regression method, demonstrates state-of-the-art results on various object detection benchmarks by incorporating with the general object detection frameworks. For traditional IoU, there are two weaknesses when it is used as a metric or a bounding box regression loss: (a) the IoU value is zero when two objects do not overlap, making it difficult to optimize the nonoverlapping bounding boxes; (b) the IoU value may be the same when two objects intersect in different orientations, so the IoU function does not reflect how the two objects overlap. To overcome these drawbacks, GIoU not only focuses on the situation where two objects overlap but also considers the situation of nonoverlapping. The details of the GIoU metric are shown in Figure 4. Suppose $B_p = (x_1^p, y_1^p, x_2^p, y_2^p)$ and $B_g = (x_1^g, y_1^g, x_2^g, y_2^g)$ are the coordinates of an object's predicted bounding box and the ground-truth bounding box, where $x_2 > x_1$ and $y_2 > y_1$ in B_P and B_g; then, the area of them is

$$A_p = \left(x_2^p - x_1^p\right) \times \left(y_2^p - y_1^p\right), \tag{4}$$

$$A_g = \left(x_2^g - x_1^g\right) \times \left(y_2^g - y_1^g\right). \tag{5}$$

The coordinates and area of intersection I of B_P and B_g can be calculated as

$$\begin{aligned} x_1^i &= \max\left(x_1^p, x_1^g\right), \\ x_2^i &= \min\left(x_2^p, x_2^g\right), \end{aligned} \tag{6}$$

$$\begin{aligned} y_1^i &= \max\left(y_1^p, y_1^g\right), \\ y_2^i &= \min\left(y_2^p, y_2^g\right), \end{aligned} \tag{7}$$

$$A_i = \begin{cases} \left(x_2^i - x_1^i\right) \times \left(y_2^i - y_1^i\right), & \text{if } x_2^i > x_1^i,\ y_2^i > y_1^i, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Similarly, the smallest enclosing box B_c can be found through

$$\begin{aligned} x_1^c &= \min\left(x_1^p, x_1^g\right), \\ x_2^c &= \max\left(x_2^p, x_2^g\right), \end{aligned} \tag{9}$$

$$\begin{aligned} y_1^c &= \min\left(y_1^p, y_1^g\right), \\ y_2^c &= \max\left(y_2^p, y_2^g\right), \end{aligned} \tag{10}$$

and the area of B_c can be computed as

$$A_c = \left(x_2^c - x_1^c\right) \times \left(y_2^c - y_1^c\right). \tag{11}$$

The IoU between B_P and B_g is defined as

$$\text{IoU} = \frac{A_i}{A_p + A_g - A_i}. \tag{12}$$

Therefore, GIoU can be calculated by the definition of

$$\text{GIoU} = \text{IoU} - \frac{A_c - \left(A_p + A_g - A_i\right)}{A_c}. \tag{13}$$

3.6. Loss Function. The proposed G-Mask model consists of two stages, which are the same as the general region-based model. In the first stage, RPN proposes the candidate bounding boxes of the object face. The second stage, follow the Fast R-CNN architecture, extracts features from each candidate box and then performs classification and

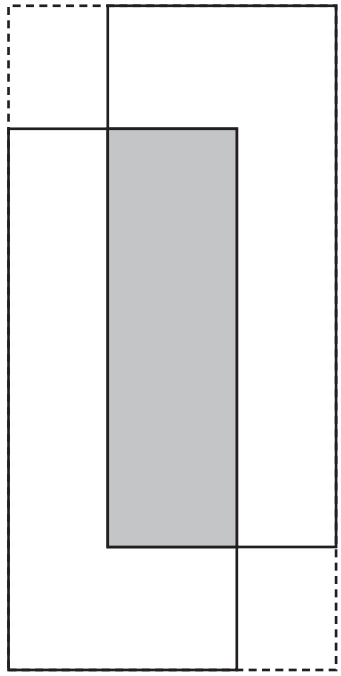


Figure 4: Illustration of GIoU metric. The solid line indicates the prediction box and ground truth box, the dotted line indicates the smallest enclosing box, and the shaded portion indicates the intersection of the prediction box and the ground truth box.

bounding box location. In addition, like the Mask R-CNN, we added a mask branch parallel to the classification branch and the bounding box location branch. Therefore, we define a multitasking objective function, which includes classification loss L_{cls}, bounding box location loss L_{box}, and segmentation loss L_{mask}. Our loss function for each image is defined as

$$L = L^*_{\text{cls}} + L^*_{\text{box}} + L^*_{\text{mask}}. \tag{14}$$

In (14), the classification loss L_{cls} and segmentation loss L_{mask} are defined the same as in Mask R-CNN. For the bounding box loss, we found that GIoU can better respond to face detection tasks through several experiments compared with the traditional bounding box regression method. Therefore, in this paper, we introduced GIoU as a bounding box loss function. In more detail, the classification loss is defined as in

$$L^*_{\text{cls}} = (\{p_i\}) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, p_i^*), \tag{15}$$

where N_{cls} is the minibatch size, i is the index of an anchor in a minibatch, and p_i is the prediction probability of whether anchor i is a face target. The ground-truth label $p_i^* = 1$ if the anchor is positive, and $p_i^* = 0$ when the anchor is negative. The classification loss L_{cls} of each anchor is log loss of whether an object is a face, which is defined as

$$L_{\text{cls}}(p_i, p_i^*) = -[p_i^* \log p_i + (1 - p_i^*)\log(1 - p_i)]. \tag{16}$$

For bounding box loss, we introduce GIoU as the loss function, and the definition of GIoU metric is described in (13), so the loss bounding box function is defined as follows:

$$L^*_{\text{box}} = 1 - \text{GIoU}. \tag{17}$$

For segmentation box loss, we adopt the average binary cross-entropy loss, which is defined in

$$L^*_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} \left[y_{ij} \log \hat{y}^k_{ij} + (1 - y_{ij}) \log\left(1 - \hat{y}^k_{ij}\right) \right], \tag{18}$$

where y_{ij} is the label value of a cell (i, j) for the region of size $m \times m$ and $\hat{y}^k_{ij}$ is the predicted value of the k-th class of this cell. L^*_{mask} is only defined on a specific mask, which is related to the ground-truth class k, and other mask outputs do not affect the loss.

4. Experiments

4.1. Experimental Setup. Unlike object detection and generic face detection, there are no off-the-shelf face datasets with masks annotation that can be employed to train our model [34]. Therefore, the first step of our work is to create a new dataset with mask annotations. In order to enhance the reliability of the samples, we selected 5115 samples from FDDB and ChokePoint datasets and annotated them with masks labels. After the annotation work, we trained the G-Mask model on this dataset.

For implementation, we adopt Keras [35] framework to train the G-Mask model in Ubuntu 16.04. ResNet-101 [36] is used as the backbone network architecture in our work. In the training phase, the G-Mask model is train on aforementioned dataset for 150,000 iterations (where the epoch is 50 and the steps of per epoch are 3000) with the learning rate set to 0.001 and the weight decay rate set to 0.0001. We randomly sample one image per batch for training [37], in which the short side of each image was resized to 800 and the long side was resized to 1024. In the RPN part, RoIs is generated by sliding the window on the feature map through anchors of different scales and different aspect ratios. It will have 2000 RoIs kept after nonmaximum suppression, and the RoIs will only be considered as foreground if its IoU with the ground truth is greater than 0.5. The testing phase settings are the same as the training phase, and the region proposal is considered to be a face only if the confidence score is greater than 0.7. The training and testing process is carried out on the same server, which is a Xeon E5 CPU of 128 GB flash memory and NVIDIA GeForce GTX 1080Ti GPU.

4.2. Experimental Results. In this work, G-Mask model not only realized the bounding box localization of the face target but also separated the face information from the background image by binary mask, so that more detailed face information could be obtained through the above process. The comparison experiment was carried out on three popular face benchmark datasets, including FDDB, AFW, and WIDER FACE.

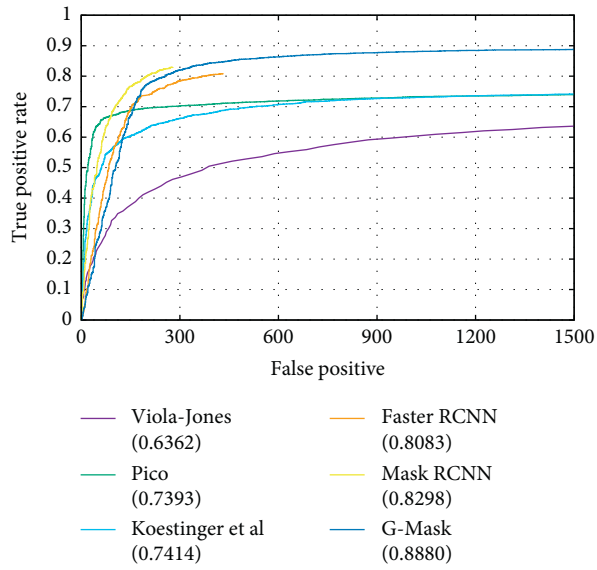
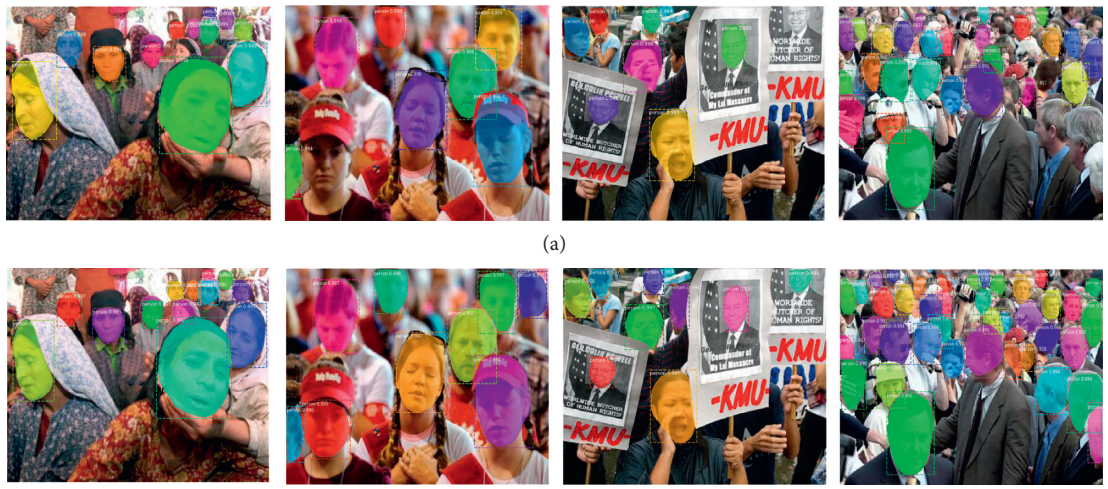


Figure 5: Comparisons of face detection with other methods on FDDB benchmark.



(a)

(b)

Figure 6: Different detection results of Mask R-CNN and G-Mask in the complex scene of FDDB dataset. (a) Mask R-CNN model and (b) G-Mask model.

The FDDB [27] dataset is a well-known face detection evaluation dataset and benchmark, which contains 2845 images of 5171 human faces. In this dataset, the faces of each image come from different scenes, which is quite challenging. We compared several methods on the FDDB dataset, including Faster R-CNN [15], Mask R-CNN [23], Pico [38], Viola-Jones [39], and Koestinger [40]. For effective comparison, the training data of the G-Mask, Mask R-CNN, and Faster R-CNN models are the same, which is the dataset constructed in this work. We compared the true positive rates at 1500 false positives, and the results are shown in Figure 5. It can be seen from Figure 5 that G-Mask performs better than Faster R-CNN when there are more than 160 false positives. When there are more than 280 false positives, the performance of G-Mask is better than that of Mask R-CNN. Furthermore, our method can achieve 88.80% true positive rate in 1500 false positives, which exceeded all the comparison methods. The comparison results of the FDDB dataset show that our proposed G-Mask method has achieved promising results, demonstrating that our method can segment face information while detecting effectively. Some detection results of the Mask R-CNN and G-Mask models in the complex scenario of FDDB dataset are shown in Figure 6. It is obvious that

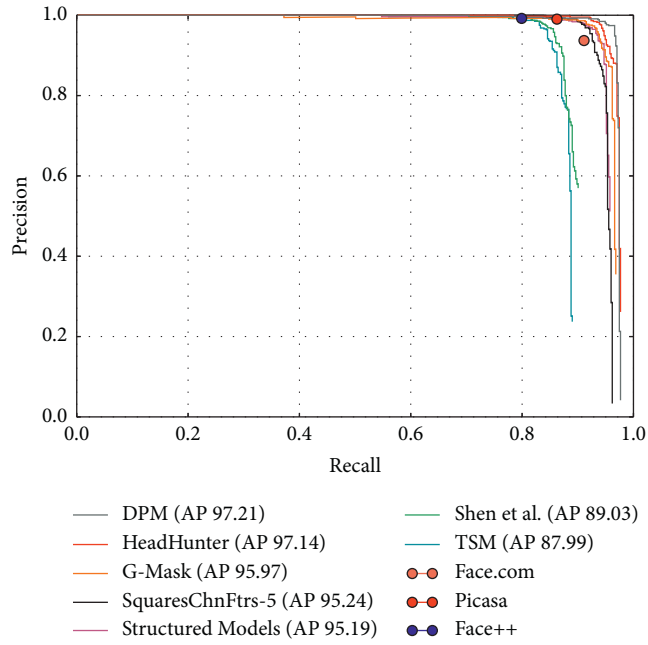


Figure 7: The precision-recall curve of our method on the AFW benchmark. Data of other models and evaluation code are derived from [41].

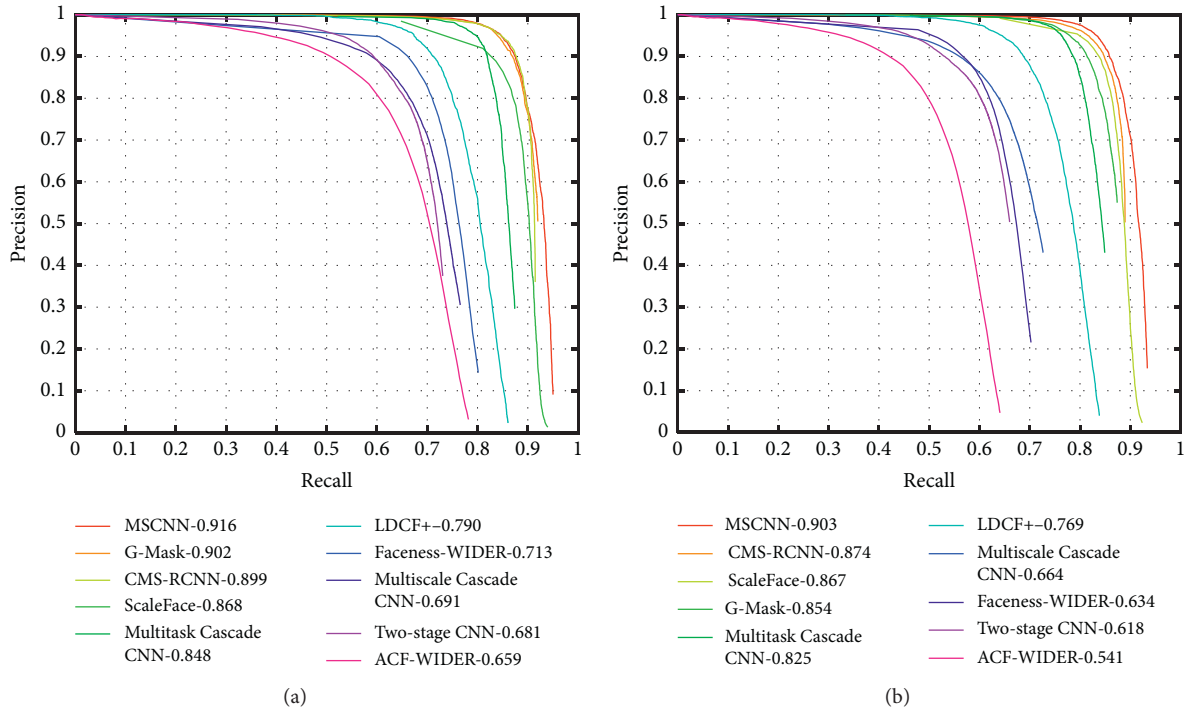
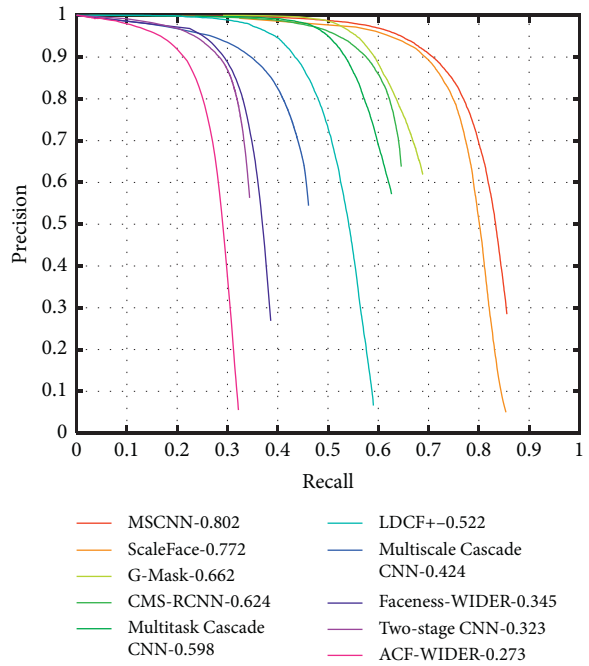


Figure 8: Continued.



(c)

Figure 8: The precision-recall curve on the WIDER FACE benchmark: (a) on the easy subset, (b) on the medium subset, and (c) on the hard subset.

Figure 9: More results of G-Mask method.

the G-Mask model performs better in the multiscale face task, which demonstrates the effectiveness of the proposed method in face detection.

The AFW dataset [29] is a face dataset and benchmark established by using Flickr image, which contains 205 images with 473 labeled faces. The precision-recall curve of our method

Table 1: Running time of different region-based methods.

Method	Running time (s)		
	FDDB	AFW	ChokePoint
R-CNN	14.75	15.32	14.51
Fast R-CNN	3.12	3.08	2.84
Faster R-CNN	**0.30**	**0.32**	**0.28**
Mask R-CNN	0.32	0.35	0.33
G-Mask	0.35	0.42	0.33

on the AFW benchmark is shown in Figure 7, and it can be seen that the G-Mask method achieved 95.97% average precision (AP). Although our dataset has a different label format from the AFW benchmark, as well as the moderately sized training dataset, we also demonstrate the generalization of our method.

WIDER FACE [30], one of the largest and most challenging face detection datasets in the open source data, has 32,203 images and 393,703 labeled faces. In this dataset, various changes in the face size, pose, and occlusion have brought great challenges to face detection, and the dataset is divided into easy, medium, and hard subsets according to the difficulty level. To further demonstrate the detection performance of our proposed method, we trained the G-Mask model on WIDER FACE dataset and verified it on the validation dataset. The proposed method is compared with several major methods including MSCNN [42], CMS-RCNN [43], ScaleFace [44], Multitask Cascade CNN [45], and Faceness-WIDER [46]. The precision-recall curves of G-Mask method on the WIDER FACE benchmark are shown in Figure 8. It can be seen that our method obtained 0.902 AP in the easy subset, 0.854 AP in the medium subset, and 0.662 AP in the hard subset, which exceeded most of the comparison methods. Compared with the state-of-the-art MSCNN method, the AP value of the proposed method is only 0.014 lower in the easy subset and 0.049 lower in the medium subset. There are some gaps between G-Mask and MSCNN methods on hard subset. The reason may be that the MSCNN method uses a series of strategies for small-scale faces detection and thus they can deal with more challenging cases. Nevertheless, the G-Mask method still achieves promising performance, which demonstrates the effectiveness of the G-Mask method.

We further demonstrate more qualitative results of G-Mask method in Figure 9. It can be observed that the proposed method can detect faces correctly while also precisely segmenting each face in an image.

We also compared the running time of different region-based methods in the a series of dataset such as FDDB, AFW, and ChokePoint. The WIDER FACE dataset was not used for testing because the running time of the hard and easy subset on the WIDER FACE was quite different. We randomly selected 100 images from each of the above datasets to test and calculate their average time, and the results are reported in Table 1. We can clearly see that Faster R-CNN has the shortest running time because of its relatively simple structure, while the proposed method has a running time similar to Mask R-CNN. Compared with Faster RCNN method, G-Mask adds a segmentation branch, which leads to an increase in computational complexity. However, the G-Mask method can achieve higher accuracy with less time consumption compared with other region-based methods and can also obtain more detailed face information through segmentation branches while accurately locating.

5. Conclusions

In this paper, a G-Mask method was proposed for face detection and segmentation. The approach can extract features by ResNet-101, generate RoIs by RPN, preserve the precise spatial position by RoIAlign, and generate binary masks through the full convolutional network (FCN). In doing so, the proposed framework is able to detect faces correctly while also precisely segmenting each face in an image. Experimental results with self-built face dataset as well as public available datasets have verified that our proposed G-Mask method achieves promising performance. For the future work, we will consider improving the speed of the proposed method.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partly supported by Innovation Team Project of the Education Department of Guangdong Province (2017KCXTD021), Key Laboratory of the Education Department of Guangdong Province (2019KSYS009), Foundation for Youth Innovation Talents in Higher Education of Guangdong Province (2018KQNCX139), Project for Distinctive Innovation of Ordinary Universities of Guangdong Province (2018KTSCX120), and the Ph.D. Start-Up Fund of Natural Science Foundation of Guangdong Province (2016A030310335).

References

[1] J. Deng, J. Guo, N. Xue et al., "Arcface: additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4690–4699, Long Beach, CA, USA, June 2019.

[2] N. Zeng, H. Zhang, B. Song et al., "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 4690–4699, 2018.

[3] Y. Shi, L. I. Guanbin, Q. Cao et al., "Face hallucination by attentive sequence optimization with reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[4] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, San Diego, CA, USA, June 2005.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[8] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester et al., "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[10] R. Vaillant, C. Monrocq, and Y. Le Cun, "Original approach for the localisation of objects in images," *IEEE Proceedings—Vision, Image, and Signal Processing*, vol. 141, no. 4, pp. 245–250, 1994.

[11] H. Li, Z. Lin, X. Shen et al., "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5325–5334, Boston, MA, USA, June 2015.

[12] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2017.

[13] B. Yang, J. Yan, Z. Lei et al., "Convolutional channel features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 82–90, Boston, MA, USA, June 2015.

[14] H. Qin, J. Yan, X. Li et al., "Joint training of cascaded CNN for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3456–3465, Las Vegas, NV, USA, July 2016.

[15] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 650–657, Washington, DC, USA, June 2017.

[16] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: an improved faster RCNN approach," *Neurocomputing*, vol. 299, no. 1, pp. 42–50, 2018.

[17] W. Wu, Y. Yin, X. Wang, and D. Xu, "Face detection with different scales based on faster R-CNN," *IEEE Transactions on Cybernetics*, vol. 49, no. 11, pp. 4017–4028, 2019.

[18] L. Liu, G. Li, Y. Xie et al., "Facial landmark machines: a backbone-branches architecture with progressive representation learning," *IEEE Transactions on Multimedia*, vol. 21, no. 9, 2019.

[19] R. Girshick, J. Donahue, T. Darrell et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, Columbus, OH, USA, June 2014.

[20] R. Girshick, "Fast r-CNN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1440–1448, Boston, MA, USA, June 2015.

[21] S. Ren, K. He, R. Girshick et al., "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 91–99, 2015.

[22] W. Wu, C. Qian, S. Yang et al., "Look at boundary: a boundary-aware face alignment algorithm,," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2129–2138, Salt Lake City, UT, USA, June 2018.

[23] K. He, G. Gkioxari, P. Dollár et al., "Mask r-CNN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2961–2969, Honolulu, HI, USA, July 2017.

[24] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," *Computer Vision—ECCV 2014*, Springer, Berlin, Germany, pp. 740–755, 2014.

[25] M. Cordts, M. Omran, S. Ramos et al., "The cityscapes dataset for semantic urban scene understanding,," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, Las Vegas, NV, USA, July 2016.

[26] H. Rezatofighi, N. Tsoi, J. Y. Gwak et al., "Generalized intersection over union: a metric and a loss for bounding box regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, Long Beach, CA, USA, June 2019.

[27] V. Jain and E. Learned-Miller, "FDDB: a benchmark for facedetection in unconstrained settings," Technical report UM-CS-2010-009, 2010.

[28] Y. Wong, S. Chen, S. Mau et al., "Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 74–81, Colorado Springs, CO, USA, June 2011.

[29] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2879–2886, Providence, RI, USA, June 2012.

[30] S. Yang, P. Luo, C. C. Loy et al., "Wider face: a face detection benchmark,," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5525–5533, Las Vegas, NV, USA, June 2016.

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.

[32] J. Ren, A. Hussain, J. Han, and X. Jia, "Cognitive modelling and learning for multimedia mining and understanding," *Cognitive Computation*, vol. 11, no. 6, pp. 761-762, 2019.

[33] J. Tschannerl, J. Ren, P. Yuen et al., "MIMR-DGSA: unsupervised hyperspectral band selection based on information theory and a modified discrete gravitational search algorithm," *Information Fusion*, vol. 51, pp. 189–200, 2019.

[34] K. Lin, H. Zhao, J. Lv et al., "Face detection and segmentation with generalized intersection over union based on mask R-CNN," in *Proceedings of the International Conference On Brain Inspired Cognitive Systems*, pp. 106–116, Guangzhou, China, July 2019.

[35] F. Chollet, "Keras, github repository," 2015, https://github.com/fchollet/keras.

[36] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, July 2016.

[37] P. Wan, C. Wu, Y. Lin et al., "Driving anger states detection based on incremental association markov blanket and least

square support vector machine," *Discrete Dynamics in Nature and Society*, vol. 2019, Article ID 2745381, 17 pages, 2019.

[38] N. Markuš, M. Frljak, I. S. Pandzic et al., "A method for object detection based on pixel intensity comparisons organized in decision trees," 2013, https://arxiv.org/abs/1305.4537.

[39] D. Hefenbrock, J. Oberg, N. T. N. Thanh et al., "Accelerating viola-jones face detection to Fpga-level using gpus," in *Proceedings of the IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, pp. 11–18, Charlotte, NC, USA, May 2010.

[40] M. Köstinger, P. Wohlhart, P. M. Roth et al., "Robust face detection by simple means," in *Proceedings of the DAGM 2012 CVAW Workshop*, Graz, Austria, August 2012.

[41] M. Mathias, R. Benenson, M. Pedersoli et al., "Face detection without bells and whistles," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 720–735, Zurich, Switzerland, September 2014.

[42] Z. Cai, Q. Fan, R. S. Feris et al., "A unified multi-scale deep convolutional neural network for fast object detection," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 354–370, Amsterdam, Netherlands, October 2016.

[43] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection," *Deep Learning for Biometrics*, Springer, Berlin, Germany, pp. 57–79, 2017.

[44] S. Yang, Y. Xiong, C. C. Loy et al., "Face detection through scale-friendly deep convolutional networks," 2017, https://arxiv.org/abs/1706.02863.

[45] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[46] S. Yang, P. Luo, C. C. Loy et al., "Faceness-net: face detection through deep facial part responses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 8, pp. 1845–1859, 2017.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 6703528, 13 pages
https://doi.org/10.1155/2020/6703528

Research Article

Reduced-Dimensional Capture of High-Dynamic Range Images with Compressive Sensing

Shundao Xie,[1] **Wenfang Wu**,[1] **Rongjun Chen**,[1,2] **and Hong-Zhou Tan**[1]

[1]*School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China*
[2]*School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*

Correspondence should be addressed to Hong-Zhou Tan; issthz@mail.sysu.edu.cn

Received 20 December 2019; Revised 26 February 2020; Accepted 5 March 2020; Published 27 April 2020

Guest Editor: Jianbiao Zhang

The range of light illumination in real scenes is very large, and ordinary cameras can only record a small part of this range, which is far lower than the range of human eyes' perception of light. High-dynamic range (HDR) imaging technology that has appeared in recent years can record a wider range of illumination than the perceptual range of the human eye. However, the current mainstream HDR imaging technology is to capture multiple low-dynamic range (LDR) images of the same scene with different exposures and then merge them into one HDR image, which greatly increases the amount of data captured. The advent of single-pixel cameras (compressive imaging system) has proved the feasibility of obtaining and restoring image data based on compressive sensing. Therefore, this paper proposes a method for reduced-dimensional capture of high dynamic range images with compressive sensing, which includes algorithms for front end (capturing) and back end (processing). At the front end, the K-SVD dictionary is used to compressive sensing the input multiple-exposure image sequence, thereby reducing the amount of data transmitted to the back end. At the back end, the Orthogonal Matching Pursuit (OMP) algorithm is used to reconstruct the input multiple-exposure image sequence. A low-rank PatchMatch algorithm is proposed to merge the reconstructed image sequence to obtain an HDR image. Simulation results show that, under the premise of reducing the complexity of the front-end equipment and the amount of communication data between the front end and the back end, the overall system achieves a good balance between the amount of calculation and the quality of the HDR image obtained.

1. Introduction

With the development of mobile Internet and Internet of Things (IoT) technology, devices with cameras are becoming more common, such as smart phones, network surveillance cameras, laptop computers, autonomous vehicles, and traffic monitoring cameras. Furthermore, camera is now an essential feature for smartphones and laptops. However, common cameras on the market can only capture low-dynamic range (LDR) images, i.e., these cameras can only capture a small part of the rang of illuminance in a real scene. The dynamic range of the real scene perceptible to the human eye is as high as $10^8:1$, but the dynamic range of the LDR images captured by these cameras is only $2^8:1$ or $2^{16}:1$, which makes the LDR images unable to truly represent the real scene. To solve this problem, high-dynamic range (HDR) imaging technique has been proposed, and it can capture a wider range of illumination than that of human eye. There are two ways to obtain HDR images: software and hardware. The hardware method directly captures HDR images by increasing the dynamic range of the sensor, but the range is very limited, and it is expensive [1]. Therefore, the software method is currently the main method, i.e., fusing multiple-exposure LDR images (hereinafter called image sequence or sequence) to obtain HDR images. The fusion method can be further divided into two categories: one is to restore the Camera Response Function (CRF) and then reconstruct the HDR light radiation pattern [2]; the other is to directly fuse the pixels of multiple-exposure sequences at the pixel level. Both categories of methods need to consider all the pixels of the multiple-exposure image sequence, increasing the computational complexity and

storage space. In the process of transmission and storage, the images are further compressed and transformed to remove redundancy to extract the required information. This method of sampling after compression results in sampling redundancy, excessive storage space, and increased transmission costs.

Compressive sensing (CS, also called compressed sensing) [3–5] can solve the above problem, which compresses signal while sampling. CS breaks through the limitation of the traditional Shannon sampling theorem and can perform high-probability reconstruction of incomplete signals at a condition far below the Nyquist sampling rate. Rice University has developed a single-pixel camera based on the theory of compressive sensing [6]. By replacing the CCD or CMOS sensors with a digital micromirror array (DMD) and a single photon detector, it only needs to sample the image fewer times than the number of pixels. Its appearance confirms the feasibility of compressive sensing applying to imaging systems. Therefore, this paper proposes a method for reduced-dimensional capture of high dynamic range images with compressive sensing, which includes algorithms for front end (capturing) and back end (processing). At the front end, the K-SVD dictionary is used to compressive sensing the input multiple-exposure image sequence, thereby reducing the amount of data transmitted to the back end. At the back end, the Orthogonal Matching Pursuit (OMP) algorithm is used to reconstruct the input multiple-exposure image sequence. A low-rank PatchMatch algorithm is proposed to merge the reconstructed image sequence to obtain an HDR image.

2. Materials and Methods

Figure 1 is the schematic of the proposed method. The whole system of this method includes three parts: front end, communication, and back end. In practical applications, especially for IoT applications, the front end is generally a low-power device with very limited computing resources, and its main role is to sense the real world. The back end is generally a cloud computing center or edge computing node, which has powerful computing resources, but is far away from the field that needs to be sensed. Communication between front end and back end includes Ethernet, mobile communication networks (including 2G, 3G, 4G, 5G, and NB-IoT), wireless local area networks (WLAN), and low-power wide area networks (LPWAN, including Lora, Sigfox). Among these communication methods, the wired network (such as Ethernet) is rarely used to directly connect front-end equipment because of high deployment costs and inflexibility. Because WLAN has a limited transmission distance, it is applicable but is relatively limited. The communication distance of LPWAN is very long, but its bandwidth is also very small, which is difficult to use for transmitting traditional image sequences. The most suitable technology for transmitting image sequences is the mobile communication network (especially 4G and 5G), but the larger the bandwidth used is, the higher the price is, and the more energy the front end uses for communication. Therefore, it is necessary to reduce the computational complexity of the front-end device and the amount of communication data between the front end and the back end. The method proposed in this paper uses compressive sensing technology to reduce the computational complexity and data volume of HDR image capturing front-end devices, thereby reducing the cost of the entire system.

2.1. Reduced-Dimensional Capture and Reconstruction of Multiple-Exposure Image Sequences. Since compressive-sensing cameras are not common now, we assume that the front end uses a common camera to capture a series of LDR images with different exposures. This assumption makes the system not only easier to implement, but also easier to compare with other methods. Every image in the image sequence is resampled using compressive sensing. Compressive sensing includes sparse representation of signals, design of measurement matrices, and design of signal reconstruction algorithms [7].

2.2. Sparse Representation of Image with Overcomplete Dictionaries. Signals are not sparse in practical applications, but when a suitable basis is used to represent the signals, they are sparse or compressible [8], i.e., the number of nonzero elements is small, which is conducive to the improvement of the sampling rate. The use of sparse representation in multiple fields has become increasingly mature, such as compression, regularization in inverse problems, and feature extraction [9]. Sparseness is the premise of compressive sensing, which means that the signal itself is sparse or sparse after some transformation, for example, transforming nonsparse signals into sparse ones by Fourier transform, discrete wavelet transform [7], i.e., the nonsparse signal is represented by a linear combination of several atoms in a fixed dictionary (such as a DCT dictionary, a wavelet dictionary, a Haar dictionary, and a Gabor dictionary). The fixed dictionary has a simple structure and simple calculation, but it can only be applied to a limited range of signals, and the sparse representation cannot be guaranteed to be optimal, i.e., the sparseness of the signal sparse representation cannot be guaranteed. To best suit a set of given signals, we can train an overcomplete dictionary with the given signals. The K-SVD [9] method can continuously iterate through sparse coding and dictionary update to optimize the sparse representation of the signal on the premise of a given training set. K-SVD can be regarded as a generalized form of K-means clustering. The only difference is that the number of atoms used for each signal is different.

Blocking each image in the multiple-exposure image sequence can further reduce storage space and the block size can usually be 8×8, 16×16, 32×32, and so on [10]. Figure 2 is the flowchart of the K-SVD algorithm. The input image sequence is $\{I_1, I_2, \ldots, I_M\}$, where M is the number of images in the sequence and I_m, $m = 1, 2, \ldots, M$, is an image in the sequence. Here we suppose that all images in the sequence have the same size of $r \times c$. All images in image sequence are divided into blocks (with block size $b \times b$) and pixels in each block are rearranged into a column vector y_i, $i = 1, 2, \ldots, N$. In case that b is not divisible by r or c, the

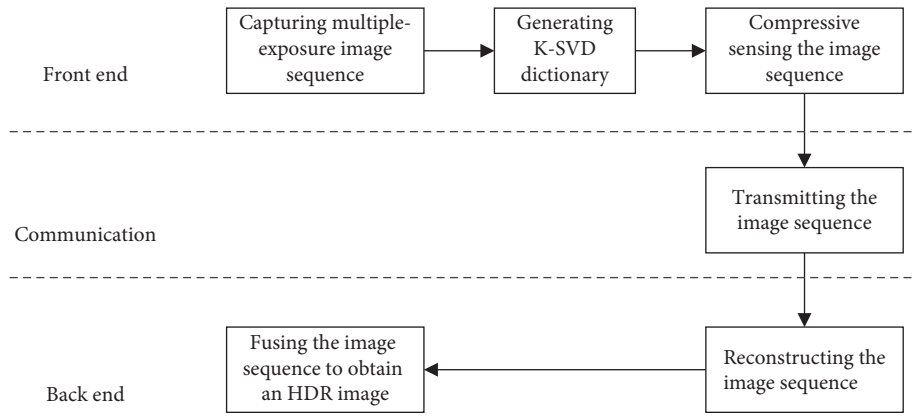


Figure 1: Schematic of the proposed method.

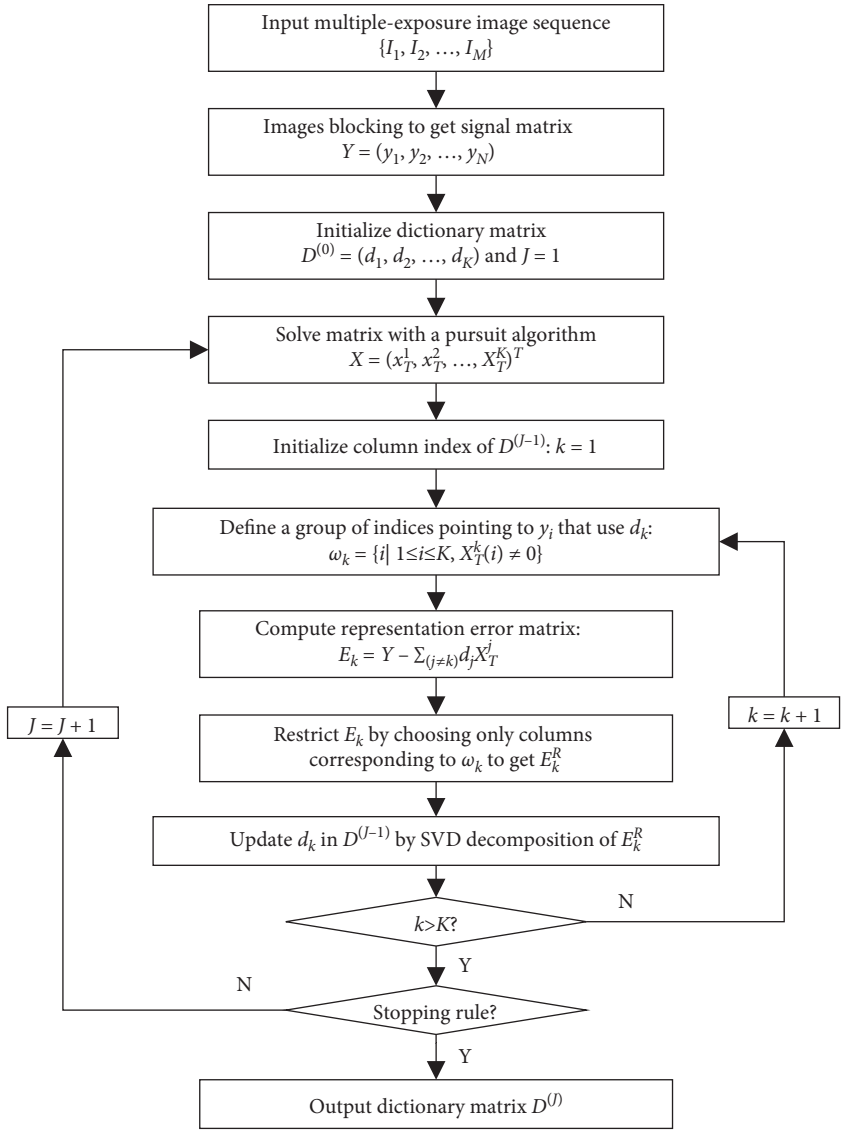


Figure 2: The flowchart of the K-SVD algorithm: I_m, (m) = *1, 2, . . .*, (M), is an image in image sequence. All images in image sequence are divided into blocks and pixels in each block are rearranged into a column vector y_i, $i = 1, 2, \ldots, N$. (N) is the number of vectors generated from the image sequence and (n) is the length of y_i. $Y \in R^{n \times N}$ is a matrix of column vectors y_i. The dictionary $D^{(J)} \in R^{n \times K}$ is made up of atom vector d_k, where K is the total number of atoms in $D^{(J)}$ and the superscript (J) is the number of iterations. $X \in R^{K \times N}$ is the sparse representation of (Y) under dictionary (D) and is made up of row vectors x_T^i, $i = 1, 2, \ldots, K$, where the subscript T of x_T^i indicates that x_T^i is a row vector and superscript T indicates matrix transpose. The matrix E_k is the error for all the input signal when the (k)th atom is removed. The detail of E_k^R and SVD decomposition of E_k^R can be found in [9].

image is expanded by 0. $N = r \times c \times M/b^2$ is the number of vectors generated from the image sequence and $n = b^2$ is the length of y_i. $Y \in R^{n \times N}$ is a matrix of column vectors y_i. The dictionary $D^{(J)} \in R^{n \times K}$ is made up of atom vector d_k, where K is the total number of atoms in $D^{(J)}$ and the superscript (J) is the number of iterations. $X \in R^{K \times N}$ is the sparse representation of Y under dictionary D and is made up of row vectors x_T^i, $i = 1, 2, \ldots, K$, where the subscript T of x_T^i indicates that x_T^i is a row vector and superscript T indicates matrix transpose. Equation (1) is the object function of K-SVD, where x_i is the ith column of matrix X and T_0 is the predetermined number of nonzero elements in x_i:

$$\min_{D,X} \left\{ \|Y - DX\|_F^2 \right\} \text{ subject to } \forall i, \quad \|x_i\|_0 \le T_0. \tag{1}$$

The matrix E_k is the error for all the input signal when the kth atom is removed. The detail of E_k^R and SVD decomposition of E_k^R can be found in [9].

2.3. Measurement Matrix Design. After the signal is sparsely represented, a suitable measurement matrix $\Phi \in R^{K \times n}$ is needed to compressive sense the signal. The design principle of the measurement matrix is that the sensing matrix $\Theta = \Phi D$ should meet the Restricted Isometry Property (RIP) [11], [12] to ensure one-to-one mapping from the original space to the sparse space. The compressive sensing of signal y_i is shown in (2), where $z_i \in R^{K \times 1}$ is the compressed sample of signal y_i:

$$z_i = \Phi y_i = \Phi D x_i = \Theta x_i. \tag{2}$$

When Φ is a Gaussian random matrix, the sensing matrix Φ can satisfy the RIP with large probability [13]. The advantage of a Gaussian measurement matrix is that it is not related to almost any sparse signal, so it requires very few measurements. Therefore, we use the Gaussian random matrix as measurement matrix.

2.4. Reconstructing Image Sequence. The reconstruction method is the core step of compressive sensing. The quality of the reconstruction method determines the quality of the reconstructed image. Compressive sensing reconstruction methods mainly include three categories [14]. The first is greedy algorithm (such as orthogonal matching pursuit (OMP) [15], stagewise orthogonal matching pursuit (StOMP) [16], and regular orthogonal matching pursuit) (ROMP) [17]). This method solves the local optimal solution to approximate the signal in each iteration. The second is a convex optimization algorithm (such as the base tracking algorithm (BP) [18], the interior point method [19], the gradient projection method [20], and the iterative threshold algorithm [21]). Convex optimization can achieve better reconstruction results with a small number of samples but has a higher computational complexity. The third is combination optimization algorithm, which uses the group testing to accurately reconstruct the signal. The reconstruction speed is fast, but the scope of application is limited, such as HHS Pursuit [22]. In this paper, we use the OMP algorithm to reconstruct the image sequence. The performance of the OMP algorithm is stable and the reconstruction accuracy is high, which can ensure that the original signal is accurately recovered at a lower sampling rate.

Given the sensing matrix $\Theta = \Phi D = (\theta_1, \theta_2, \ldots, \theta_K)$ and the compressed sample z_i of signal y_i, the OMP algorithm can estimate the sparse representation x_i of signal y_i. Then the signal y_i can be recovered by 3

$$y_i = D x_i. \tag{3}$$

The idea behind the OMP is to pick columns in a greedy fashion, i.e., at each iteration t, the column θ_t of Θ that is most strongly correlated with the remaining part of x_i is chosen [15]. Figure 3 is the flowchart of the OMP algorithm. The input is the sensing matrix Θ and one of the compressed signals $z = z_i$, $i = 1, 2, \ldots, N$ in (2). After running N times of OMP, we can get the matrix X (the sparse representation of Y) and Y can be calculated column by column using (2). At last the image sequence can be reconstructed from Y.

2.5. Low-Rank PatchMatch Algorithm. During the capture process of the multiple-exposure image sequence, camera shaking or unpredictable moving objects in the scene are inevitable, which will cause artifacts or noise to appear in the final fused HDR image. Currently, block matching fusion method is mainly used to eliminate the artifacts and noise. The essence of block matching fusion is to find a mapping relationship between two different images A and B (given the image block set of A and B as {PA} and {PB}, respectively), i.e., by calculating the correlation, find the nearest-neighbor field (NNF) of B, so that the error of similar image blocks in the two images is minimized. By looking for the block in {PA} that is closest to block in {PB}, the artifacts in the fused image are reduced.

If image block matching is performed through a full search, the complexity is as high as $O(mM^2)$, where m and M are size of the image and the size of the block, respectively. To reduce the complexity, Connelly Barnes et al. [23], [24] proposed a fast PatchMatch algorithm with randomized nearest neighbor and successfully reduced the complexity of the algorithm to O(mMlog(M)). The main steps of the algorithm can be summarized as initialization, propagation, and random search. Due to the high efficiency and better performance of the PatchMatch algorithm, it has a profound impact in the fields of image stitching, image completion, and image reorganization.

In fact, there are generally more than three multiple-exposure LDR images of the same scene to fuse into HDR image, so Pradeep Sen et al. [25] proposed multisource bidirectional similarity (MBDS), as shown in (4). S is the original image, and T is the target image. N is the number of source images. P and Q are patches in S and T, respectively. $\omega_k(P)$ weighs the source patches when calculating completeness based on how well-exposed they are. In order to measure the weight of a well-exposed image block, the well-exposed image block has a large weight, and vice versa. d $(\cdot)$ is a distance metric, which is usually calculated using the l_2 norm. $|T|$ is the total number of image blocks of the target

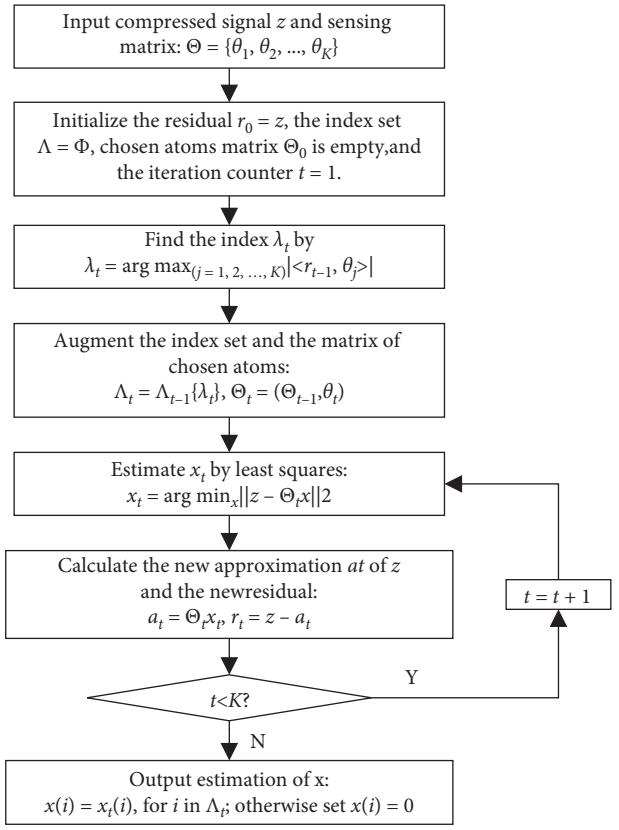


Figure 3: The flowchart of the OMP algorithm.

image. This formula mainly includes the integrity of mapping from S to T and the correlation from T to S. MBDS selects well-exposed blocks in the image sequence to fill the registration image, so it can achieve better registration results:

$$\mathrm{MBDS}(T|S_1, \ldots, S_N) = \frac{1}{N}\sum_{k=1}^{N}\sum_{P\in S_1,\ldots,S_N}\omega_k(P)\min_{Q\in T}\mathrm{d}(P,Q) + \frac{1}{|T|}\sum_{Q\in T}\min_{P\in S_1,\ldots,S_N}\mathrm{d}(Q,P). \tag{4}$$

From the perspective of low-rank matrix recovery, combined with the idea of MBDS, this paper proposes an improved algorithm for removing artifacts from HDR images. The objective function is shown in (5). The input image sequence has N images I_i, $i = 1, \ldots, N$, and I_{ref} is the reference image selected from the sequence. L_i, $i = 1, \ldots, N$, is the result image of I_i being aligned to the reference image I_{ref}; that is, the content of L_i is aligned with the reference image, and the exposure parameters remain the same with I_i. Function $g_i(I_j)$ is the mapping from exposure parameter i to exposure parameter j. Function $h(\cdot)$ maps the grayscale domain of LDR to the radiance domain of HDR. Function $\mathrm{vec}(\cdot)$ turns a two-dimensional image into a column vector:

$$\sum_{i=1, i\neq ref}^{N}\mathrm{MBDS}(L_i|I_i, g_i(I_{\mathrm{ref}})) \quad \text{s.t}\,\mathrm{rank}(\mathrm{vec}(h(L_1), \ldots, \mathrm{vec}(h(L_N)))) = 1. \tag{5}$$

Solve the MBDS problem to get L_i. More details can be found in [25]. In addition, the addition of a low-rank constraint enables the aligned images to ensure a sufficiently low rank, i.e., to maintain a linear correlation in brightness. The solution is to divide into two independent local optimization subproblems, namely, the problem of MBDS and low-rank matrix recovery. At the same time, the iterative solution under multiresolution scale is used to find the optimal solution of MBDS. The low-rank matrix finally obtained is the target HDR image with high dynamic range and linear brightness of the scene. The process is shown in Figure 4.

3. Results and Discussion

This section will analyze the convergence of the low-rank PatchMatch algorithm, simulate the multiexposure image compressive sensing and reconstruction algorithm and the antiartifact fusion algorithm, and evaluate the algorithms in terms of subjective and objective criteria.

3.1. Convergence of the Low-Rank PatchMatch Algorithm. Randomly generate data matrices with rank of r and size of 1000×500, and add sparse noise with a noise ratio of p. Validate the convergence by two sets of experiments. In first set, fix matrix rank r to 1, and observe the convergence under different noise ratios p. In the second set, fix the noise ratio p, which is set to 0.2 in the experiment, and observe the convergence under different ranks r. The results are shown in Figure 5. In both cases, the low-rank PatchMatch algorithm converges within 5 iterations.

3.2. Image Evaluation Criteria. In this paper, we will use the mean squared error (MSE), peak signal-to-noise ratio (PSNR), information entropy, average gradient, and running time to objectively evaluate the image quality and algorithm.

The definition of mean squared error (MSE) is shown in (4), where m and n are the width and height of the images, and f and g are two different images. The standard deviation represents how much the experimental data deviates from the mean. The higher the standard deviation, the more diverse the result data, and the lower the accuracy of the result:

$$\mathrm{MSE} = \frac{1}{\mathrm{mn}}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\|f(i,j) - g(i,j)\|^2. \tag{6}$$

PSNR is a commonly used criterion for reconstructed image quality evaluation. According to the definition of the standard deviation in equation (4), the definition of PSNR is given in equation (5), where MAX_I is the maximum value of the pixel value of an image, for example, 255 for an 8 bit grey image. The larger the PSNR value, the lower the degree of distortion of the image:

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right). \tag{7}$$

Information entropy represents the average information of an image, that is, the average information after removing

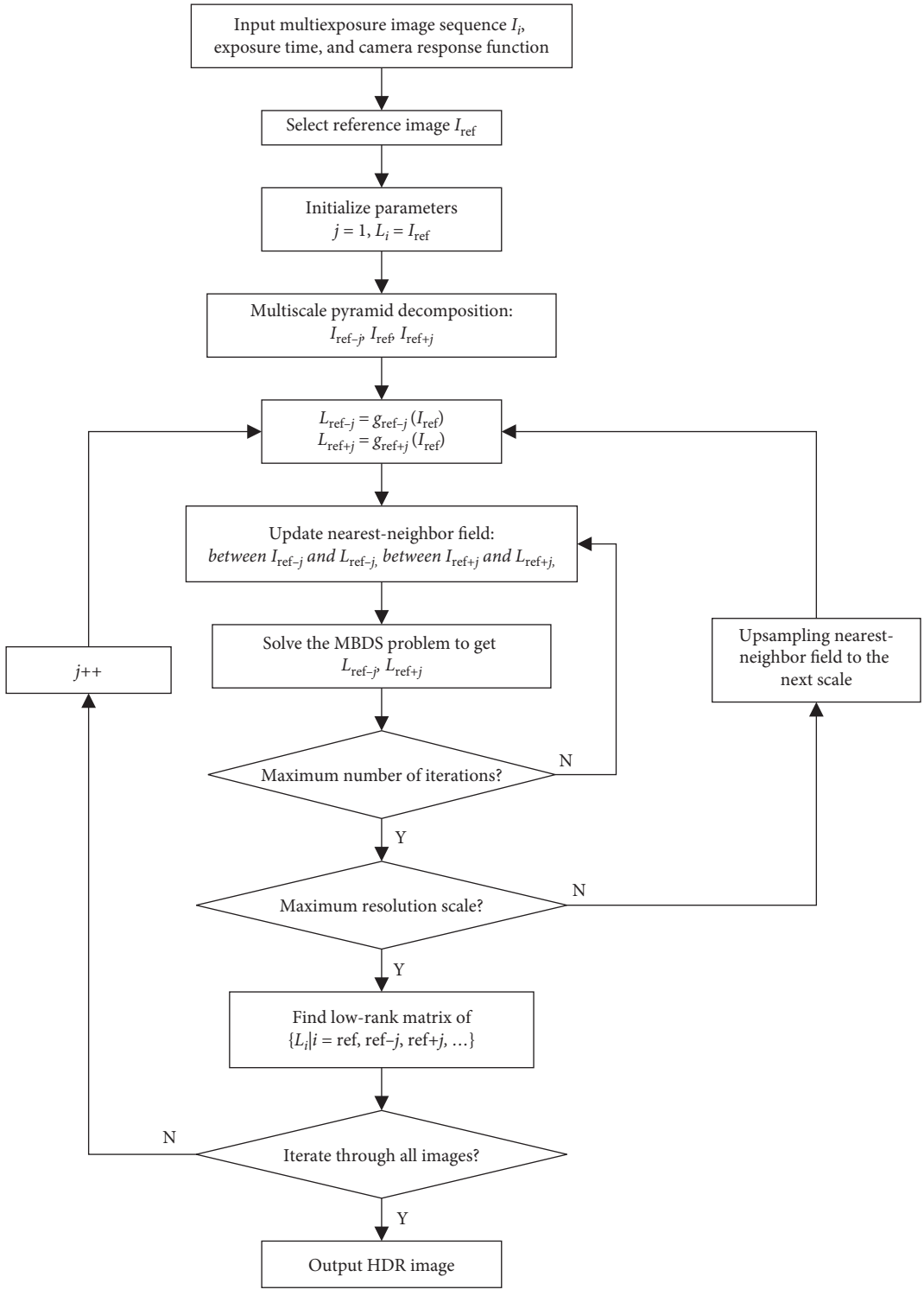


Figure 4: Process of low-rank PatchMatch.

redundant information. The definition is shown in equation (7), where $p(b_i)$ is the probability that the brightness b_i appears in the image, and L is the maximum grey value of the image:

$$\text{Entropy} = -\sum_{i=1}^{L} p(b_i)\log_2 p(b_i). \tag{8}$$

The average gradient characterizes the relative sharpness of the image and reflects the rate of change in contrast of details. The larger the average gradient is, the larger the changes of grey level are, and the richer the levels of the image are. The definition is shown in equation (8), where M and N are the number of rows and columns of the image f, respectively:

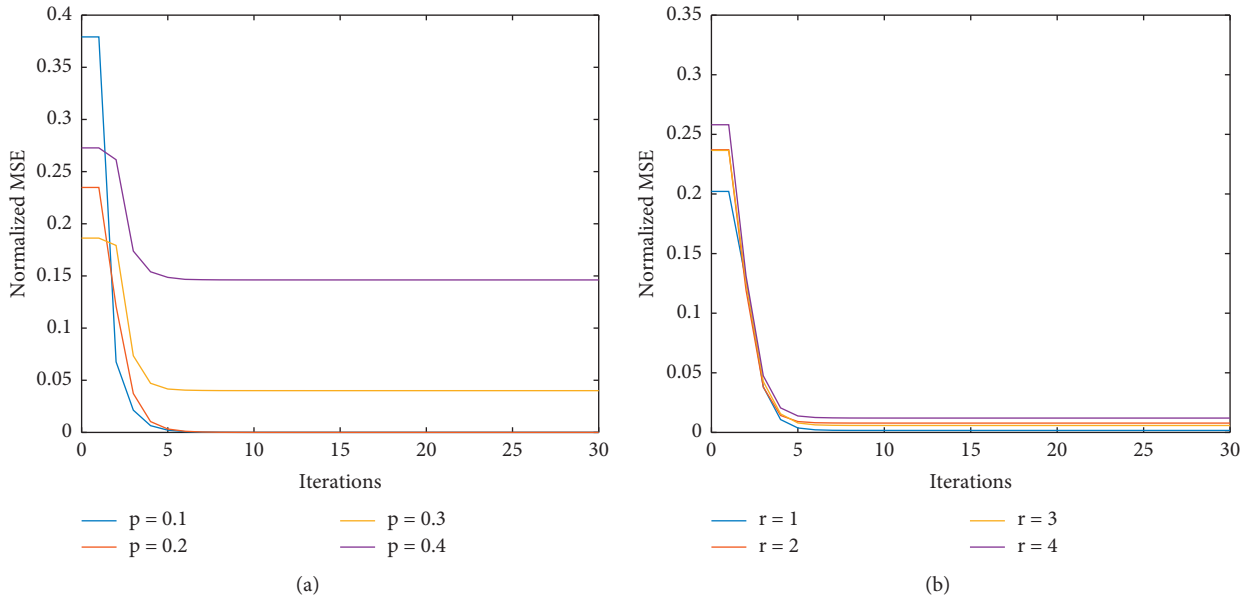


Figure 5: Convergence of the low-rank PatchMatch algorithm. (a) The convergence under different noise ratios (p) (the matrix rank (r) is fixed to 1). (b) The convergence under different ranks (r) (the noise ratio (p) is set to 0.2).

$$\overline{g} = \frac{1}{(M-1)(N-1)} \times \sum_{i=1}^{M-1}\sum_{j=1}^{N-1}\sqrt{\frac{(f(i,j)-f(i+1,j))^2+(f(i,j)-f(i,j+1))^2}{2}}. \quad (9)$$

For the fusion of multiple-exposure images of complex scenes with moving objects, the evaluation of deghosting needs to be further explored. At present, there is no mature objective criterion to evaluate the deghosting of HDR images. In this paper, the deghosting evaluation method proposed by Karaduzovic-Hadziabdic and Telaovic et al. [26] is used, and the test image set used is a complex real scene.

3.3. Simulation of Compressive Sensing and Reconstruction for Multiple-Exposure Images. The simulation platform for this experiment is MATLAB 2015b; the hardware is 32G memory, Intel Core i5-6600K processor (main frequency 3.5 GHz). Airplane and Lena with an image size of 512 × 512 were selected for simulation, and the simulation results were compared with BP, OMP, and StOMP algorithms at lower ($R = 0.3$), medium ($R = 0.5$), and higher ($R = 0.7$) sampling rates, respectively. The simulation results are shown in Table 1.

Among the three major types of algorithms for compressive sensing reconstruction, the convex optimization algorithm has the best performance, but then it has the highest complexity and the longest reconstruction time. As a representative of convex optimization algorithm, BP has better reconstruction performance than greedy algorithm and combinatorial optimization algorithm. At the sampling rate of 0.3 and 0.5, compared with BP algorithm, the performance of our algorithm is better than BP algorithm. With a sampling rate of 0.8, although the PSNR of our algorithm is slightly lower than BP, it is higher than the other algorithms. In addition, the reconstruction time of our algorithm is shorter than the reconstruction time of the BP algorithm except that the sampling rate is low ($R = 0.3$).

The simulation results are shown in Figure 6 when the sampling rate is 0.5. From a subjective point of view, for the letter area on the fuselage and wings of the airplane image, both the BP algorithm and our algorithm can recover the clear letters, but the letters recovered by the OMP algorithm and the StOMP algorithm are blurred. The images recovered by BP, OMP, and StOMP algorithms all have obvious noise. The particle noise of StOMP algorithm is the most obvious. Our algorithm can basically restore the image information correctly.

The reconstruction difference of the Lena image is not as obvious as the aircraft image from the subjective point of view, but it can be observed that the images recovered by BP, OMP, and StOMP all have different degrees of noise, and the particle noise of StOMP is the most obvious.

Among the above algorithms, the StOMP algorithm has the shortest reconstruction time, but the reconstruction effect is also the worst, and the particle noise is very obvious. Because of the existence of noise, the average gradient of the StOMP algorithm is higher than that of other algorithms. The average gradient of our algorithm is similar to the OMP algorithm, which is better than the BP algorithm. From the perspective of MSE, the MSE of BP algorithm is the smallest, and our algorithm is second. Information entropy is similar to the case of MSE.

3.4. Simulation Multiple-Exposure Images Fusion Algorithm. In this section, the multiple-exposure image sequences Arch, Sculpture Garden, and Puppet are compressive sensed,

Table 1: Comparison of difference compressive sensing and reconstruction algorithm.

Image	Sampling rate	Algorithm	PSNR	MSE	Entropy	Average gradient	Running time (s)
Airplane	0.3	BP	23.774	44.389	7.215	7.802	28.199
		OMP	23.609	46.271	7.188	8.543	15.442
		StOMP	25.271	47.781	7.065	10.049	0.075
		Ours	28.861	46.341	7.097	8.227	36.31
	0.5	BP	30.278	45.169	6.996	6.673	42.239
		OMP	28.506	46.545	7.057	7.167	30.557
		StOMP	24.913	48.297	7.113	11.379	0.079
		Ours	33.356	46.365	6.999	7.158	38.024
	0.8	BP	40.16	46.097	6.902	6.189	81.628
		OMP	34.257	46.368	6.952	6.462	60.081
		StOMP	26.219	47.882	7.117	10.788	0.311
		Ours	36.601	46.348	6.938	6.551	42.084
Lena	0.3	BP	27.394	46.153	7.862	6.303	27.167
		OMP	26.285	47.967	7.864	7.26	15.625
		StOMP	26.579	48.876	7.861	9.157	0.089
		Ours	30.116	47.846	7.862	7.12	28.709
	0.5	BP	32.867	47.262	7.865	5.88	39.993
		OMP	30.808	47.803	7.866	6.371	29.406
		StOMP	25.885	49.249	7.859	10.208	0.313
		Ours	33.442	47.743	7.863	6.308	29.429
	0.8	BP	40.415	47.678	7.862	5.795	80.541
		OMP	35.008	47.85	7.865	5.971	58.919
		StOMP	27.261	49.012	7.861	9.779	0.326
		Ours	36.171	47.799	7.863	6.051	32.847

(a) (b) (c)

(d) (e)

Figure 6: Continued.

(f) (g) (h)

(i) (j)

Figure 6: Comparison of difference compressive sensing and reconstruction algorithm at medium sample rate ($R = 0.5$). (a) Airplane. (b) BP. (c) OMP. (d) StOMP. (e) Ours. (f) Lena. (g) BP. (h) OMP. (i) StOMP. (j) Ours.

(a)

(b)

Figure 7: Continued.

(c) (d) (e) (f)

Figure 7: Multiple-exposure image sequences Arch fusing result. (a) Multiple-exposure image sequence Arch. (b) Arch after compressive sensed and reconstructed (c) Reference [2]. (d) RPCA. (e) PSSV. (f) OURS.

(a)

(b)

(c) (d) (e)

Figure 8: Multiple-exposure image sequences Puppet fusing result. (a) Multiple-exposure image sequences Puppet. (b) Puppet after compressive sensed and reconstructed (c) Reference [31]. (c) SEN [25]. (d) HU [30]. (e) OURS.

FIGURE 9: Multiple-exposure image sequences Sculpture Garden fusing result. (a) Multiple-exposure image sequence Sculpture Garden. (b) Sculpture Garden after compressive sensed and reconstructed. (c) Reference [31]. (d) Reference [2]. (e) SEN [25]. (f) HU [30]. (g) OURS.

(a) (b) (c) (d)

Figure 10: Details of the fusion result in Figure 9. (a) Reference [2]. (b) SEN [25]. (c) HU [30]. (d) OURS.

reconstructed, and then fused into HDR images. The results are compared with robust principal component analysis (RPCA) [27], partial sum of singular value (PSSV) [28], [29], and [2], MBDS algorithm (referred to as SEN) proposed by Pradeep Sen et al. [25], the brightness and texture consistency deghosting method (referred to as HU) proposed by Jun Hu et al. [30], and the low-rank restoration based deghosting fusion proposed by Tae-Hyun Oh et al. [31].

Figure 7 shows the Arch image sequence. There are moving people in the picture. Artifacts can occur with direct fusion. Reference [2] and RPCA cannot suppress the appearance of artifacts, and our algorithm and PSSV algorithm can both suppress artifacts well and have better subjective visual effects.

Figure 8 shows the results of the Puppet sequence. Our algorithm adds low-rank constraints to minimize the impact of misaligned regions and keep the resulting image as linear as possible. It can be seen from the result that our algorithm is better.

Figure 9 shows the results of Sculpture Garden sequence. There are many pedestrians in the picture, which makes it difficult to remove artifacts. From the results of the fusion, [30] is the worst, and there are obvious artifacts in [6], and the SEN method has a fuzzy phenomenon. Both HU and this algorithm suppress artifacts well, but due to the effect of image blocking, block effect exists in the fusion result.

Figure 10 is the local area details of the fusion result in Figure 9. Because result of [31] is the worst compared to other algorithms, the detail of it is not enlarged. The literature [6] is less effective in removing artifacts because of the obvious silhouette cross. There is obvious blurring at the pedestrian edges of the SEN method. HU and our algorithm achieve better results, but our algorithm has noise caused by block effects.

4. Conclusions

Aiming at the problems of traditional cameras with redundant sampling, large storage space consumption, and inability to fully record the radiance in the real scene due to the limitation of the dynamic range of the sensor, this article uses the K-SVD dictionary to compressive sensing LDR images of different exposure in the same scene. Then the LDR images is reconstructed and fused with low-rank PatchMatch algorithm to get an HDR image. The simulation results show that the method in this paper can effectively reduce the sampling rate and remove the image artifacts and blurring caused by the camera shake and the motion of the objects in the scene. It provides a method for compressive sensing to obtain HDR images.

However, due to the introduction of block compressive sensing, the size of the image block has become a factor that cannot be ignored. Simulation results show that when the image block is small, the block effect is more obvious and the edge details are distorted. But when increasing the image block, it will increase the storage space and computational complexity. In addition, adding compressive sensing and dictionary learning before fusion increases the computation time, sacrificing time in exchange for reduction in complexity, and sampling rate. Therefore, the next step is to perform pixel-level fusion in the compressive sensing domain of the HDR image to further reduce the time required for the algorithm and improve the quality of the fused image.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the Project for Distinctive Innovation of Ordinary Universities of Guangdong Province (no. 2018KTSCX120), the Ph.D. Start-Up Fund of Natural Science Foundation of Guangdong Province (no. 2016A030310335), and Guangdong Science and Technology Plan Project (no. 76120-42020022).

References

[1] F. Banterle, A. Artusi, K. Debattista, and A. Chalmers, *Advanced High Dynamic Range Imaging*, Taylor & Francis, CRC Press, Boca Raton, FL, USA, Second edition, 2018.

[2] P. E. Debevec and J. Malik, *Recovering High Dynamic Range Radiance Maps from Photographs*, pp. 31–10, ACM SIGGRAPH 2008 Classes, New York, NY, USA, 2008.

[3] E. Candès, *"Compressive Sampling," Presented at the Proceedings of the International Congress of Mathematicians*, Madrid, Spain, 2006.

[4] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[5] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal Processing*, vol. 86, no. 3, pp. 549–571, Mar. 2006.
[6] D. Takhar, J. N. Laska, M. B. Wakin et al., "A new compressive imaging camera architecture using optical-domain compression," *Computational Imaging IV*, vol. 6065, p. 606509, 2006.
[7] G. Shi, D. Liu, D. Gao, Z. Liu, J. Lin, and L. Wang, "Advances in theory and application of compressed sensing," *Acta Electronica Sinica*, vol. 37, no. 5, pp. 1070–1081, 2009.
[8] X. Zhang, *Matrix Analysis and Application*, Tsinghua University Press Co., Ltd., Beijing, China, 2004.
[9] M. Aharon, M. Elad, and A. Bruckstein, "$rm K$-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
[10] L. Gan, "Block compressed sensing of natural images," in *Proceedings of the 2007 15th International Conference on Digital Signal Processing*, pp. 403–406, Wales, UK, July 2007.
[11] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007.
[12] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.
[13] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
[14] D. Needell, J. A. Tropp, and " CoSaMP, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.
[15] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
[16] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.
[17] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, 2009.
[18] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
[19] K. S J, M. Lustig, An interior-point method for large-scale l1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.
[20] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.
[21] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.
[22] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, *One Sketch for All: Fast Algorithms for Compressed Sensing*, ACM, in *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pp. 237–246, San Diego, CA, USA, June 2007.
[23] C. Barnes, E. Shechtman, A. Finkelstein, D. B. Goldman, and " PatchMatch, *A Randomized Correspondence Algorithm for Structural Image Editing*, pp. 24–11, no. 1–24, ACM SIGGRAPH, New York, NY, USA, 2009.
[24] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, *The Generalized PatchMatch Correspondence Algorithm*, pp. 29–43, Computer Vision–ECCV 2010, Berlin, Heidelberg, 2010.
[25] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Shechtman, "Robust patch-based hdr reconstruction of dynamic scenes," *ACM Transactions on Graphics*, vol. 31, no. 6, p. 1, Nov.
[26] K. Karaduzovic–Hadziabdic and R. Mantiuk, "Expert evaluation of deghosting algorithms for multi-exposure high dynamic range imaging," in *Proccedings of the HDRi2014-Second International Conference and SME Workshop on HDR Imaging*, EBANGOR, Sarajevo, Bosnia, April 2014.
[27] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 11–37, 2011.
[28] T.-H. Oh, H. Kim, Y.-W. Tai, J.-C. Bazin, and I. So Kweon, "Partial sum minimization of singular values in rpca for low-level vision," in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, Harbour, Sydney, pp. 145–152, April 2013.
[29] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, "Partial sum minimization of singular values in robust PCA: algorithm and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 744–758, Apr. 2016.
[30] J. Hu, O. Gallo, K. Pulli, and X. Sun, "HDR deghosting: how to deal with saturation?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Harbour, Sydney, pp. 1163–1170, April 2013.
[31] T.-H. Oh, J.-Y. Lee, Y.-W. Tai, and I. S. Kweon, "Robust high dynamic range imaging by rank minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1219–1232, 2015.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 2528954, 10 pages
https://doi.org/10.1155/2020/2528954

Research Article

Incremental Instance-Oriented 3D Semantic Mapping via RGB-D Cameras for Unknown Indoor Scene

Wei Li,[1] **Junhua Gu**,[2] **Benwen Chen**,[2] **and Jungong Han**[3]

[1]*School of Electrical Engineering, State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability of Hebei Province, Hebei University of Technology, Tianjin 300401, China*
[2]*School of Artificial Intelligence, Key Laboratory of Big Data Computing, Hebei University of Technology, Tianjin 300401, China*
[3]*WMG Data Science, University of Warwick, CV4 7AL, Coventry, UK*

Correspondence should be addressed to Junhua Gu; jhgu@hebut.edu.cn

Received 12 January 2020; Revised 25 February 2020; Accepted 3 March 2020; Published 23 April 2020

Guest Editor: Jinchang Ren

Scene parsing plays a crucial role when accomplishing human-robot interaction tasks. As the "eye" of the robot, RGB-D camera is one of the most important components for collecting multiview images to construct instance-oriented 3D environment semantic maps, especially in unknown indoor scenes. Although there are plenty of studies developing accurate object-level mapping systems with different types of cameras, these methods either process the instance segmentation problem in completed mapping or suffer from a critical real-time issue due to heavy computation processing required. In this paper, we propose a novel method to incrementally build instance-oriented 3D semantic maps directly from images acquired by the RGB-D camera. To ensure an efficient reconstruction of 3D objects with semantic and instance IDs, the input RGB images are operated by a real-time deep-learned object detector. To obtain accurate point cloud cluster, we adopt the Gaussian mixture model as an optimizer after processing 2D to 3D projection. Next, we present a data association strategy to update class probabilities across the frames. Finally, a map integration strategy fuses information about their 3D shapes, locations, and instance IDs in a faster way. We evaluate our system on different indoor scenes including offices, bedrooms, and living rooms from the SceneNN dataset, and the results show that our method not only builds the instance-oriented semantic map efficiently but also enhances the accuracy of the individual instance in the scene.

1. Introduction

Robot vision plays an important role with the development of artificial intelligence industries. With aid of RGB-D cameras (such as Kinect), robots can "see" and analyze the surrounding environment easily. Then, how to make robots accurately and rapidly percept the meaning of objects in real-world environments without a prior knowledge is one of the most important technologies in robotic community. For tasks, such as path planning, object grabbing, or even autonomous driving, we need not only the semantic understanding of a single object but more important, the spatial relationships and layout among individual instances in a 3D environment. It thus leads to the demand of building high-level instance-oriented representations of the scene that would greatly advance the human-robotic interaction. Hence, building progressive semantic instance-level 3D map for indoor scenes with multiview RGB-D images has always been a major project for researchers.

The conventional methods of constructing object-aware semantic maps generally consist of two inseparable aspects: instance segmentation of 3D image and transformation across multiple views. The former focuses on obtaining semantic information via the convolutional neural network [1–6], which is followed by integrating geometric segmentation approach to label 3D objects of the scene. The latter usually carries out simultaneous localization and mapping (SLAM) [7–9], which completes 3D scene reconstruction using RGB-D cameras. Motivated by the mentioned technologies, several works efficiently combine them to generate

a semantically segmented 3D map [10–12] and have achieved impressive results. However, such methods suffer from the oversegment problem or lack of proper data association strategy, and meanwhile, they are computationally inefficient, making them unsuitable for the real-time applications. Some other works focus on processing large-scale video retrieval [13–15], but they mainly deal with the entire scene.

This paper intends to incrementally build instance-orient semantic 3D maps via RGB-D cameras in real time. Without the need of a prior knowledge, the proposed mapping system contains optimized semantic information about the individual object instances from the scene and, meanwhile, integrates semantic probabilities from multiple viewpoints to a globally consistent 3D semantic map. The entire algorithm is basically carried out in three steps. First, RGB images captured by cameras undergo the Mask R–CNN [1] algorithm to generate 2D instance and class predictions. In the second step, the proposed system associates prediction results online into corresponding point cloud mapping by the SLAM system. To improve the instance accuracy, we utilize a Gaussian mixture model with the EM algorithm to cluster and optimize semantical labels predicted from the convolutional neural network. In the last step, we propose a voxel-based Bayesian update strategy towards incremental class update across different frames, which will be incorporated into the truncated signed distance function- (TSDF-) based reconstruction maps for the purpose of accelerating the computational efficiency and reducing time complexity.

The major difference between our system and other works [10, 16, 17] is that we employ the projection relation between voxel and pixel directly to obtain instances semantic in the 3D map instead of using the combination between geometry segmentation on depth images and 2D instance segmentation methods. Doing so helps avoid oversegment with no computation increased. Moreover, our goal is to build an instance-level indoor map consisting of reconstructed object instances with semantic annotation. So, unlike many other dense reconstructions works [18–20] that pursuit accurate instance segmentation, the proposed approach aims to achieve the real-time performance, facilitating real-life robotic applications.

To sum up, the main contributions of this work are as follows:

(i) A novel incremental instance-oriented mapping system that utilizes an RGB-D camera to obtain sequential images and represents as a TSDF-based voxelization map

(ii) An optimization method based on a Gaussian mixture model that clusters the point cloud, further integrating TSDF volumes that contain semantic class and instance IDs

(iii) A voxel-based Bayesian update strategy that tracks and updates class probability distribution across different frames to perform consistent global scene mapping

(iv) Qualitative and quantitative analysis of the proposed system on the SceneNN [21] dataset in multiple scenarios

2. Related Works

2.1. Dense 3D Scene Reconstruction. We can roughly divide 3D reconstruction technologies based on RGB-D images into three categories: feature-based methods, voxel-based methods, and surfel-based methods. Feature-based methods, in general, involve front-end frame-to-frame motion through feature matching and back-end "loop closing" constraints from a heuristic search to perform pose graph optimization. The first popular open-source system was RGB-D SLAM [22] proposed by Endres et al. Subsequent similar methods include DVO-SLAM by Kerl et al. [23] and ORB-SLAM2 by Mur-Artal and Tardos [24]. Although such methods directly consume the point cloud, they could cause incomplete instance segmentation in object-level mapping tasks. Voxel-based methods, such as [8, 25, 26], integrate all depth data of the sensor into a volume model from a 3D space, which uses the iterative closest point (ICP) algorithm to track camera poses and reconstruct dense 3D scene maps.

2.2. Semantic Instance-Aware Mapping. Previous methods have addressed the task of mapping at the level of individual objects. Civera et al. [27] used a monocular SLAM system to create 3D environment maps and then inserted the modeled object from the built database. Similarly, Pavel et al. [28] also required priori 3D object models. Although these methods perform object-oriented semantic mapping, the requirement for priori knowledge of modeling objects makes it difficult for them to be applied in real-time human-robot interaction.

Recent developments in deep learning have also enabled the integration of rich semantic information within real-time simultaneous localization and mapping (SLAM) systems. The work in [11] fuses semantic predictions from a CNN into a dense map built with a SLAM framework. However, conventional semantic segmentation is unaware of object instances, i.e., it does not disambiguate between individual instances that belong to the same category. Thus, the approach in [11] does not provide any information about the geometry and relative placement of individual objects in the scene. A number of other works have addressed the task of detecting and segmenting individual semantically meaningful objects in 3D scenes without predefined shape templates [10, 16, 17, 27, 29–34]. Runz et al. [32] employed the object detector for the first step and then updated the class probabilities of each element consisting of the reconstructed 3D map. As it has a huge time complexity, these methods suggested to only extract semantic information on a subset of the input frames; McCormac et al. [29] utilized the same prediction model but aims at extending the SLAM system by means of object-level pose graph optimizations and relocalizations. [16, 17] are similar to that, but they employ depth segmentation methods to segment 3D instances, which led them to take different approaches and

reach different goals. [22] Proposes an object-oriented mapping system that combines a Single Shot MultiBox Detector (SSD) [6] with ORB-SLAM2 [24]. There are also several object-oriented dense 3D mapping methods [30, 31], the main idea of which is to obtain 2D semantic information by a CNN framework, create associated relationships between 2D semantic and 3D mapping, and then utilize conditional random fields (CRFs) as a postprocessing step to refine the results of semantic segmentation. Another project worth mentioning is [35]. Although it also combines a CNN and SLAM to generate 3D semantic mapping, it adds a recurrent neural network (RNN) [28] in data association.

2.3. Instance Detection and Segmentation. Nowadays, with the rapid development of the convolutional neural network, semantic-related tasks in real-world environments have shown some remarkable results. Beginning with the object detection [3, 28] in RGB images, soon afterwards, Mask R–CNN came out which is further able to predict a per-pixel semantically annotated mask for each of the detected instances, achieving state-of-the-art results on the COCO [36] instance-level semantic segmentation task. Other similar works that are worth to mention, including YOLO [5] and SSD [6], deliver an outstanding performance in terms of accurately segmenting instances. With the help of 2D semantic information, we explore semantical objects in 3D environments.

3. Materials and Methods

The architecture of our system is shown in Figure 1. Each RGB image from the incoming video stream is processed with the Mask R–CNN framework to detect a semantically annotated segmentation mask, then, along with the corresponding depth image, is initialized to the point cloud using the projection method between coordinate frames followed by an optimization strategy using a Gaussian mixture model (GMM) for a more accurate instance label. Next, we employ a voxel-based Bayesian update method to merge class semantic or instance IDs across different frames. Finally, we complete the construction of an incremental instance-oriented semantic mapping system. Details of the proposed system are discussed in the following sections.

3.1. Semantic Instance Segmentation Method. In order to annotate and segment the 3D instances in the scene, we needed to combine the 3D point cloud with its corresponding semantic class distribution and instance IDs. To label objects, we first employed the Mask R–CNN as an object detector to the input image. Mask R–CNN achieved real-time performance while showing high accuracy on the computer vision benchmarks, including the Microsoft COCO dataset [37] and the Pascal VOC collection of datasets [38]. Given the input image $\mathcal{I}_t(\vec{u})$, $\vec{u} = (x, y) \subset \mathbb{Z}^2$, $0 \le x < W$, $0 \le y < H$, Mask R–CNN provides a set of bounding boxes as b_i, $i \subset \mathbb{N}$, $1 \le i \le M$, and class probabilities are assigned to each bounding box as $P(c_i \mid_{I_k}) \subset \mathbb{R}$ by letting $M \in \mathbb{R}^{100*15*15}$ be the number of bounding boxes and $c\varepsilon\mathbb{R}^{100}$ be the class category. Note: although there is a good deal of related research, we chose Mask-R-CNN to achieve the task because of its stability and ability to obtain good results on different datasets. This way, our system can theoretically handle another similar network for an acceleration or accuracy request.

3.2. Incremental 3D Semantic Instance-Oriented Update

3.2.1. 2D-3D Association with Semantic Information. One requirement of the proposed system is to know the camera pose in the target scene. In view of real-time and computing costs, we chose voxel hashing [9] as our SLAM system. This takes advantage of volumetric approaches to achieve dense surface representation while using spatial hashing techniques to avoid memory overhead. The proposed system takes both RGB and depth information as the input and incrementally project them into a single 3D model to achieve the volumetric reconstruction. For each arriving RGB-D frame, the 6-DoF camera pose is estimated by combining ICP [36] and RGB alignment, denoted as $T_{WC} \in \mathrm{SE}(3)$, where W represents the world coordinate and C represents the camera coordinate. Then, we employ the homogeneous transformation matrix $T_{WC}^{-1}(k) = T_{CW}(k)$ to project the transformation from the world coordinate to the camera coordinate. In our case, instead of integrating the original incoming RGB image, the proposed system takes the semantic image $\mathcal{I}_k$ that was processed through the Mask R–CNN as the input, along with corresponding D_k, and then generates the 3D reconstruction with the estimated camera pose. Therefore, the initial point cloud with instance IDs has been generated.

3.2.2. Instance Refinement via the Gaussian Mixture Model. After the rough 2D-3D data association of the SLAM system, point cloud data instances are initially formed, but some false matching points occurred during the projection process. In order to obtain more accurate object representation, we optimized the objects by formulating an accelerated generative model in the form of a GMM with a highly parallel hierarchical expectation-maximization (EM) algorithm, inspired by [39]. Also, there is an alternative clustering approach which can be used for optimization, such as ROC algorithm [12]. As a cluster solution for 3D point cloud data, the advantages of GMM are suited to our work. First, the projected data are embedded into the covariance matrices of GMM, which provides an effective way of processing noisy data. Second, because the storage requirements for a GMM are much lower, the system's ability to perform in real time is not affected. However, due to the computational complexity of the GMM, processing is relatively slow. Normally, the processing method would employ a k-means algorithm to run on the sample set. Because our system already implements 2D-3D association using the projection method of the SLAM system, it generates the corresponding 3D cloud with semantic and instance annotations. This is equal to the process of the sample set, and

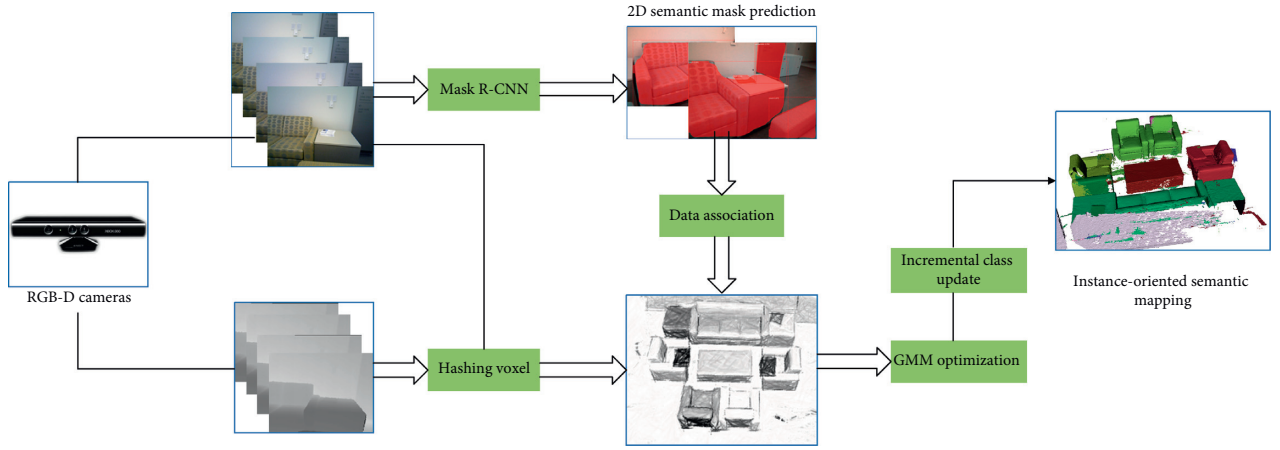


Figure 1: Overview of our incremental instance-level 3D scene reconstruction method. From continuous frames of an RGB-D sensor, our system performs on-the-fly reconstruction and 3D semantic prediction. All of our processing is performed on a frame-by-frame basis in an online fashion, thereby making it useful for real-time applications.

therefore, we can optimize the point cloud data clusters directly with the GMM.

(1) Model Definition. After masks m_j^k are produced by the Mask R–CNN integrated into depth map D_k, we obtained a corresponding point cloud $X = \{x_1, \ldots, x_N\}$ of size N. We assume that there are K classes that can be altered according to the demands of different scenarios. The latent variable represents as $Z = \{z_1, \ldots, z_N\}$, which is a discrete random variable related to sampled point cloud X. In our case, Z indicates classes, the purpose is to index which observed variable belongs to which Gaussian distribution, and the probability of Z represents as $p(Z) = \{p_1, \ldots, p_k\}$. For our formulation, the parameter $\Theta = \{p_k, \mu_k, \Sigma_k\}$ that needs to be estimated with $p_k \varepsilon p(Z)$ represents as class probability and μ_k and Σ_k being the mean and covariance matrix, respectively. Our function describing the generation of incoming point cloud data is a linear combination of Gaussians:

$$p(X \mid \Theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} p_k N\left(x_i \mid \mu_k, \Sigma_k\right), \tag{1}$$

with $\sum_{k=1}^{K} p_k = 1$, and the point cloud data are sets of independent and identically distributed (iid) points.

(2) Executive Parameters. In our case, we are trying to maximize the overall likelihood of a set of Gaussians producing a given point cloud. The general way to compute the maximizer of a parameter is maximum likelihood estimation, but it is only suitable for one Gaussian distribution-contained problem; otherwise, it would not provide an analytical solution. That is why we chose to solve this problem using the EM algorithm, which employs an iterative approach to finding the maximizer of a parameter.

Given initial value $\theta^{(0)}$, the function represents in E-step:

$$\begin{aligned} E_{Z|X,\theta^{(t)}} &= \int_Z^1 \log[p(X, Z \mid \Theta)] p\left(Z \mid X, \theta^{(t)}\right) \mathrm{d}z \\ &= \sum_{k=1}^{K} \sum_{n=1}^{N} \log\left[p_k N\left(x_i \mid \mu_k, \Sigma_k\right)\right] \frac{p_{z_i} N\left(x_i \mid_{\mu_{z_i}^{(t)}, \Sigma_{z_i}^{(t)}}\right)}{\sum_{k=1}^{K} p_k^{(t)} N\left(x_i \mid_{\mu_k^{(t)}, \Sigma_k^{(t)}}\right)}. \end{aligned} \tag{2}$$

In the M-Step, we maximize the expected log-likelihood with respect to θ. The objective function is

$$\theta^{(t+1)} = \arg\max EZ \mid X, \theta^{(t)}. \tag{3}$$

Given a fixed set of expectations, one can solve for the optimal parameters at iteration t:

$$\begin{aligned} p_k^{(t+1)} &= \frac{1}{N} \sum_{i=1}^{N} p\left(z_i = p_k \mid x_i, \theta^{(t)}\right), \\ \mu_k^{(t+1)} &= \frac{\sum_{i=1}^{N} p\left(z_i = p_k \mid x_i, \theta^{(t)} p\left(z_i = p_k\right)\right)}{\sum_{i=1}^{N} p\left(z_i = p_k \mid x_i, \theta^{(t)}\right)}, \\ \sum_k^{(t+1)} &= \frac{\sum_{i=1}^{N} p\left(z_i = p_k \mid x_i, \theta^{(t)}\left(x_i - \mu_k^{(t+1)}\right) p\left(x_i - \mu_k^{(t+1)}\right)^T\right)}{\sum_{i=1}^{N} p\left(z_i = p_k \mid x_i, \theta^{(t)}\right)}. \end{aligned} \tag{4}$$

3.2.3. Voxel-Based Bayesian Class Update Approach. Because frame-wise segmentation processes each incoming RGB-D image pair independently, it lacks any spatiotemporal information about corresponding segments and instances across the different frames. Therefore, we propose an incremental voxel-based Bayesian class update approach. According to Nießner et al. [9], given a series of RGB images $\mathcal{I}_1, \ldots, \mathcal{I}_k$ with semantic and instance IDs, as discussed in Section 3.2.1, and corresponding depth images $D_1, \ldots, D_k$, volumetric representation divides them into a small square called a voxel, v, which stores information such as location, color, and class. In order to update the class distribution of each voxel according to the given classes of pixels from the 2D images, we must first find the correspondence between the voxel and the pixel. This is performed by the SLAM system. Therefore, for the current incoming frame $\mathcal{I}_k$, the world coordinate of the corresponding voxel, $v_k(\vec{u})$, in a 3D map is computed by using backprojection:

$$v_k(\vec{u}) = D_k(\vec{u})K^{-1}\dot{u}, \tag{5}$$

where K denotes the intrinsic camera parameter and $\dot{u}$ denotes the corresponding homogeneous coordinate of the pixel's $\vec{u}$.

Each voxel is then projected onto the RGB image plane via camera projection as follows:

$$\vec{u}(v,k) = \pi\left(T_{WC}^{-1}(k)v_k(\vec{u})\right). \tag{6}$$

When a new image $\mathcal{I}_k$ comes in, the system feeds it to the Mask R–CNN to segment n masks denoted as m_j^k, $j = 1, 2, \ldots, n$. Mask R–CNN outputs masks that may overlap each other, so we do not directly gain a class distribution per pixel, as in semantic segmentation. Therefore, we update the class distribution mask by mask. With the relationship between each pair of voxel and pixel computed from (6), we update the class distribution by an optimized recursive Bayesian update algorithm [11], which fits better with our system:

$$\begin{aligned} P\left(c_v = c_i \mid \mathcal{I}_1, \ldots, \mathcal{I}_k\right) &= \frac{1}{Z}P\left(c_v = c_i \mid \mathcal{I}_1, \ldots, \mathcal{I}_{k-1}\right)P\left(c_v = c_i \mid \mathcal{I}_k\right) \\ &= \frac{1}{Z}P\left(c_v = c_i \mid \mathcal{I}_1, \ldots, \mathcal{I}_{k-1}\right)\prod_{j=1}^{n}P\left(c_{\vec{u}(v,k)} = c_i \mid m_j^k\right). \end{aligned} \tag{7}$$

The instance probability distribution update procedure is similar. Nonetheless, the two distributions are updated independently. We store a list of instance probabilities $P(I_v = I_i)$ for each voxel v with I representing instance IDs. We update the instance distribution according to the segmentation result given by the Mask R–CNN. The general update function for instance distribution adopts a recursive Bayesian update scheme as well:

$$\begin{aligned} &P\left(I_v = I_i \mid \mathcal{I}_1, \ldots, \mathcal{I}_k\right) = P\left(I_v = I_i \mid \mathcal{I}_1, \ldots, \mathcal{I}_{k-1}\right) \\ &\prod_{j=1}^{n}P\left(I_{\vec{u}(v,k)} = I_i \mid m_j^k\right). \end{aligned} \tag{8}$$

3.3. Map Integration. The instance segmentation in the 3D format mentioned above achieves associate class probabilities over multiple camera views. After voxel-based class update approach, every voxel's instance ID has been updated as I_v. For map integration, we attempt to integrate 3D semantic instances into a globally volumetric map with greater speed. To this end, each clustered instance is progressive and integrated into a TSDF-based voxel grid, which is measurement from a depth map, D_k, into a volume, V. V stores at each discrete voxel location, $v = (v_x, v_y, v_z)$, both the current normalized truncated signed distance value, its associated weight, and instance class I_v. And we use raycast, the main method for integrating information from sensor data into TSDF for tracking, data association, and visualization to render depth, normals, vertices, RGB, and object indices as shown in Figure 2. The fusion part of our system is incorporated with Voxblox [40], which is a real-time framework of 3D reconstruction based on volumetric TSDF representation. The main benefit of the Voxblox framework is that it has been extended to the label volume, which can store the instance label related with each voxel in the TSDF grid. At each view, the set of point clouds representing the 3D object with semantics is integrated into the voxel-based representation, and our system ensures consistency among the instance labels across different frames.

4. Results and Discussion

We evaluated the performance of our system on an Ubuntu operating system with an Intel Core i5-6500 CPU at 3.2 GHz and an Nvidia GeForce GTX1080 Ti GPU with 11 GB of RAM. Our system is built on top of ROS open-source middleware. The core function is implemented in Python and uses TensorFlow for instance predictions.

The Mask R–CNN uses ResNet-101 based on the publicly available implementation from Matterport Inc. [41], with the pretrained weights provided for the Microsoft COCO dataset [37].

The input stream is typically a 640×480 resolution RGB-D video. To display the ability of progressive building of instance-aware maps per frame, we perform a Mask R–CNN thread simultaneously with 3D reconstruction upon every frame.

Although there are many 3D databases [42, 43] for different research purposes, we chose the SceneNN dataset [21] to evaluate the 3D object accuracy of the proposed instance-level semantic mapping system, which contains 100 indoor scenes, including offices, bedrooms, living rooms, and kitchens, and scenes with repetitive objects; the SceneNN dataset also provides the annotations with fine-grained information, e.g., axis-aligned bounding boxes, oriented bounding boxes, and object poses. It is suited to the task of reconstruction of the instance-oriented semantic mapping.

4.1. Run-Time Performance. To demonstrate the efficiency of our system, we analyzed its run-time performance and compared it with other state-of-the-art systems, as shown in Table 1. These systems are mainly concentrated on object-level mapping tasks. Our system achieved a speed of 10.8 Hz while performing all processing components on every input frame, thereby outperforming other similar

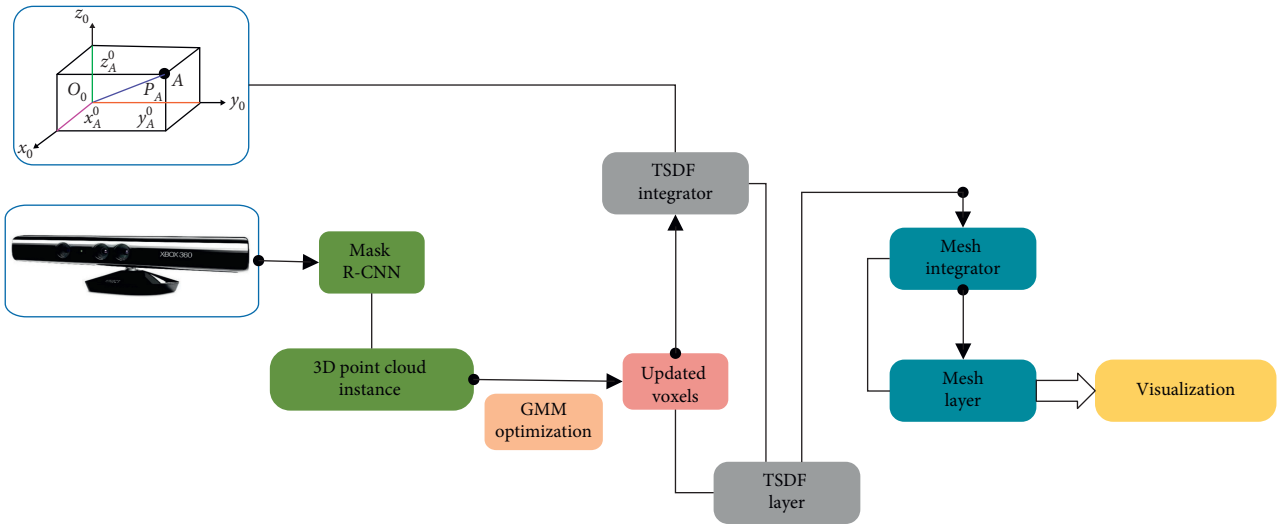


Figure 2: Map integration of the proposed system, showing the interaction between multiple layers and with incoming sensor data through integrators.

Table 1: Comparison of run-time performance. FQ denotes the frequency recognition of when the input frame is performed, and the class probabilities of the 3D map are updated.

Method	Representation	FQ	FPS
SemanticFusion [11]	Dense	Every 10 frames	Under 8 Hz
Hermans et al. [34]	Dense	Every 6 frames	3 Hz
PanopicFusion [44]	Dense	Every 10 frames	4.3 Hz
Voxblox++ [16]	Instance-oriented	Every frame	1 Hz
Pham et al. [45]	Instance-oriented	Every frame	1 Hz
Fusion++ [29]	Instance-oriented	Every frame	4 Hz
Ours	Instance-oriented	Every frame	**10.8 Hz**

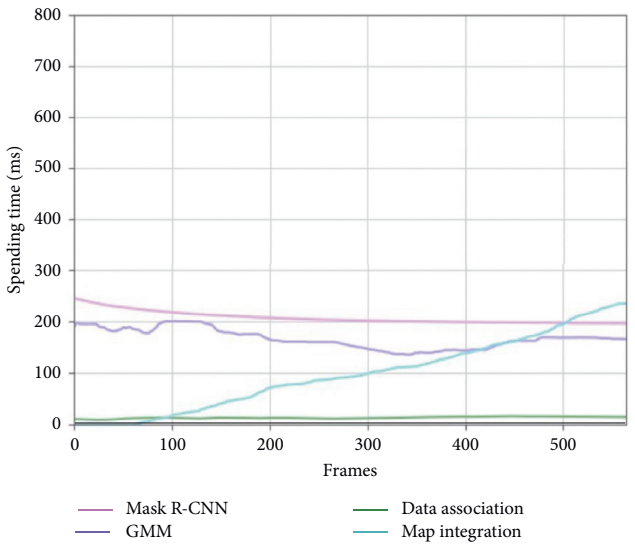


Figure 3: Measured execution times of each stage of the proposed incremental instance-oriented mapping system, averaged over the 5 evaluated sequences from the SceneNN [21].

systems in run-time tests. Compared to the process for utilizing the semantic information from the input image in conventional methods [16, 29, 32], the proposed system has substantially reduced the computational time by exploiting a voxel-based class probability update scheme. All systems were tested on the same sequences of the SceneNN dataset.

Figure 3 shows the evaluation of the execution times upon each individual stage of the proposed incremental instance-level mapping system averaged over five sequences in the SceneNN dataset. Input RGB-D images have 640×480 resolution. Mask R–CNN runs on the GPU, while the rest of the components run on the CPU. The trend lines in the figure showed the data association module running under low rate, which the proposed method effectively improves the operation speed of the system; GMM module maintained on a stable running rate; the map integration module slowed down after 500 frames, ensuring the real-time demand of the system. Note that, by speeding up the system, it is possible to change to a faster object detector network, and the processing of map fusion and Mask R–CNN can occur simultaneously.

4.2. Accuracy. Several recent research projects have focused on semantic instance segmentation of 3D scenes. The majority of these, however, takes as the input the full reconstructed scene, either processing it in chunks or directly as a whole. Because such methods are not constrained to

Table 2: Comparison to the 3D semantic instance segmentation approach from Voxblox++ [16] proposed by Grinvald et al. For 10 sequences from the SceneNN dataset [21], the per-class average precision (AP) is computed using an intersection over union (IoU) threshold of 0.5 over the predicted 3D segmentation masks.

Seq. ID	Method	Bed	Chair	Sofa	Table	Books	Refrigerator	TV	Toilet	Bag
011	Voxblox++	—	75	50	100	—	—	—	—	—
	Ours	—	68.7	**67**	100	—	—	—	—	—
016	Voxblox++	100	0.0	0.0	—	—	—	—	—	—
	Ours	75	0.0	0.0	—	—	—	—	—	—
030	Voxblox++	—	54.4	100	55.6	14.3	—	—	—	—
	Ours	—	**76**	100	50	8.3	—	—	—	—
061	Voxblox++	—	—	100	33.3	—	—	—	—	—
	Ours	—		59.9	33.3					
078	Voxblox++	—	33.3	—	0.0	47.6	100	—	—	—
	Ours	—	**50**	—	**100**	**54.2**	75			
086	Voxblox++	—	80	—	—	0.0	—	—	—	0.0
	Ours	—	66.7	—	—	**25**	—	—	—	**50**
096	Voxblox++	0.0	87.5	—	37.5	0.0	—	0.0	—	50
	Ours	0.0	55.7	—	**39.5**	**11.1**	—	0.0	—	**68.7**
206	Voxblox++	—	58.3	100	60	—	—	—	—	100
	Ours	—	**60**	100	55	—	—	—	—	100
223	Voxblox++	—	12.5	—	75	—	—	—	—	—
	Ours	—	**16.7**	—	75	—	—	—	—	—
255	Voxblox++	—	—	—	—	—	75	—	—	—
	Ours	—	—	—	—	—	75	—	—	—

progressively integrating predictions from partial observations into a global map but can learn from the entire 3D layout of the scene, they are not directly comparable with our work. Among the frameworks that study online, incremental instance-aware semantic mapping, we chose Grinvald et al. [16] as a comparison. Because we relied on a Mask R–CNN model trained on the 80 Microsoft COCO [38] object classes to get the instance IDs, we evaluated the segmentation accuracy on the nine object categories that were common to the SceneNN dataset [21]. The proposed approach was evaluated on the 10 indoor sequences from the SceneNN dataset, the same as Grinvald et al. [16] reported instance-level segmentation results. The results in Table 2 demonstrate that our approach achieves better accuracy in most sequences compared with [16], which is one of the advanced methods focused on real-time incremental instance-aware 3D mapping. It is worth mentioning that further comparing it with [16], our system runs faster and is more suitable for human-robot interaction.

To expand the evaluation of the accuracy of our system, we compared class-averaged mean average precision (mAP) values over the ten evaluated categories with [16, 45]. The results in Table 3 show that the proposed approach outperforms the baseline on six sequences. [45] focuses on building incremental 3D semantic maps of indoor scenes; although it is different from our system, there is an experiment designed for the accuracy of instance classes, and the author explained they only used a simple clustering algorithm to obtain instance semantic so that it can be used as a baseline to compare with similar systems. As the results shown in Table 3, our system highly outperformed in eight scenes compared to their system. Compared to Voxblox++, the proposed system exceeded in six sequences, which proved the advancement of our system. However, it did not perform better in sequences 16, 61, 96, and 206, through analyzing the categories in those sequenced, such as bed and sofa, had more clutter appearances, using the GMM model to optimize might cause oversegment which reduced accuracy. Also, Voxblox++ uses the geometric segmentation method which is better to segment objects with more details, such as chair. We will improve the algorithm in the future.

Table 3: Comparison to the 3D semantic instance-segmentation approach from Voxblox++ [16] and Pham et al. [45] on class-averaged mAP value.

Sequence ID	Voxblox++ [16]	Pham et al. [45]	Ours
011	75.0	52.1	**78.6**
016	33.3	34.2	25.0
030	56.1	56.8	**58.6**
061	66.7	59.1	46.6
078	45.2	34.9	**69.8**
086	20.0	35.0	**47.2**
096	29.2	26.5	26.7
206	79.6	41.7	78.0
223	43.8	40.9	**45.8**
255	75.0	48.6	**75.0**

Furthermore, we showed the qualitative results about the proposed framework on the SceneNN dataset. We presented the incremental instance-oriented 3D semantic mapping generation process in Figure 4. As can be seen, the left image showed the respective progressive semantic segmentation results of our method, the middle image shows the final mapping results, and the right one shows the ground truth segmentation, and the 3D shapes of the object instances, such as chair, sofa, and desk, were incrementally generated

Figure 4: Generation process of incremental instance-oriented semantic mapping in real time.

Proposed system without GMM optimization | Proposed system with GMM optimization | Ground truth

Figure 5: Ablation study on the effects of GMM optimization. The comparison shows the refinement help to improve the segmentation accuracy.

by our system. Because our system is designed to segment instances from the scene, the color of the instance is different from the ground truth, in which the color is assigned according to the classes. As our proposed mapping system focuses primarily on recovering instances of the scene, we have chosen to ignore the background and floor.

4.3. Ablation Analysis. To further illustrate the performance of our GMM model pertaining to the optimized instance cluster, we carried out an ablation analysis to evaluate the effects of accuracy of instance, as shown in Figure 5. Circle A shows that, after GMM optimization, the boundaries of the instance are clearer, and the

segmentation is more accurate. And circle B displays that two different instances are segmented after GMM optimization. The same optimization result is showed in C, and the boundaries of different objects are clearer. This proves that cluster operation in the point cloud based on predicted class information is valid in dense semantic instance-level mapping.

5. Conclusions

Our proposed system is an efficient instance-oriented semantic mapping system. We employed a projection method in the SLAM system that could rapidly associate 2D "masks" and the corresponding depth images to generate a 3D point cloud with instance labels and then used a cluster optimized algorithm to resolve the confusion if projection mismatch occurred. For the 3D reconstruction, the resulting instance-aware semantically annotated volumetric maps are expected to provide benefits in navigation and manipulation planning tasks.

However, as mentioned above, because our system focuses only on recovering 3D instances of an unknown scene, we overlooked the structure of the surrounding environment, such as walls and floors. In the future, we hope to come up with a method that could solve this problem in real time. And also, our system can be used in different applications, such as [44, 46–48]. We intend to research how the segmented instances can serve as semantic landmarks to promote the accuracy of the SLAM system in order to attain a full semantic SLAM system.

Data Availability

The experimental data of the SceneNN and Microsoft COCO dataset used to support the findings of this study are included within the paper.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by Hebei Provincal Innovation Capability Enhancement Project (199676146H).

References

[1] H. Kaiming, G. Georgia, D. Piotr, and G. Ross, "Mask R-CNN," in *Proceedings of the International Conference on Computer Vision (ICCV)*, IEEE, Venice, Italy, Transactions on Pattern Analysis and Machine Intelligence, Venice, Italy, October 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, June 2016.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] R. Girshick, J. Donahue, T. Darrelland, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, June 2014.

[5] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[6] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Computer Vision—ECCV 2016* in *European Conference on Computer Vision*, vol. 9905, Cham, Switzerland, Lecture Notes in Computer Science, 2016.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2017.

[8] R. A. Newcombe, S. Izadi, O. Hilliges et al., "Kinectfusion: real-time dense surface mapping and tracking," in *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality/ISMAR*, pp. 127–136, Basel, Switzerland, June 2011.

[9] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 1–11, 2013.

[10] N. Sunderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proceedings of the RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5079–5085, IEEE, Vancouver, BC, Canada, September 2017.

[11] J. Mccormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: dense 3D semantic mapping with convolutional neural networks," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Singapore, May 2017.

[12] X. Gu, Z. P. Angelov, and Z. Zhao, "A distance-type-insensitive clustering approach," *Applied Soft Computing*, vol. 77, pp. 622–634, 2019.

[13] G. Wu, J. Han, Y. Guo et al., "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1993–2007, 2018.

[14] G. Wu, J. Han, Z. Lin et al., "Joint image-text hashing for fast large-scale cross-media retrieval using self-supervised deep learning," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9868–9877, 2018.

[15] C. Yan, B. Gong, Y. Wei et al., "Deep multi-view enhancement hashing for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1. In press, 2020.

[16] M. Grinvald, F. Furrer, T. Novkovic et al., "Volumetric instance-aware semantic mapping and 3D object discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3037–3044, 2019.

[17] Y. Nakajima and H. Saito, "Efficient object-oriented semantic mapping with object detector," *IEEE Access*, vol. 7, p. 3206, 2019.

[18] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 1, 2017.

[19] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: dense tracking and mapping in real-time," in *Proceedings of the International Conference on Computer Vision, ICCV*, November 2011.

[20] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: real-time dense slam and light

source estimation," *International Journal of Robotics Research*, vol. 35, no. 14, p. 1697, 2016.

[21] B. S. Hua, Q. H. Pham, D. T. Nguyen et al., "SceneNN: a scene meshes dataset with aNNotations," in *Proceedings of the Fourth International Conference on 3D vision (3DV)*, IEEE Computer Society, Stanford, CA, USA, October 2016.

[22] F. Endres, J. Hess, J. Sturm et al., "3-D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2017.

[23] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proceedings of the RSJ International Conference on Intelligent Robots and Systems*, pp. 2100–2106, IEEE, Tokyo, Japan, November 2013.

[24] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[25] P. Henry, D. Fox, A. Bhowmik, and R. Mangnia, "Patch volumes: segmentation-based consistent mapping with RGB-D cameras," in *Proceedings of the International Conference on 3D Vision-3DV 2013*, pp. 398–405, IEEE, Seattle, WA, USA, June 2013.

[26] T. Whelan and J. McDonald, "Kintinuous: spatially extended kinectfusion," in *Proceedings of the RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Cambidge, MA, USA, July 2012.

[27] J. Civera, A. J. Davison, and J. M. M. Montiel, *Structure from Motion Using the Extended Kalman filter*, Springer Science & Business Media, Berlin, Germany, 2011.

[28] M. S. Pavel, H. Schulz, and S. Behnke, "Recurrent convolutional neural networks for object-class segmentation of RGB-D video," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Killarney, Ireland, July 2015.

[29] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: volumetric object-level slam," in *Proceedings of the International Conference on 3D Vision (3DV)*, pp. 32–41, IEEE, Verona, Italy, September 2018.

[30] X. Li and R. Belaroussi, "Semi-dense 3D semantic mapping from monocular slam," 2016, https://arxiv.org/abs/1611.04144.

[31] S. Yang, Y. Huang, and S. Scherer, "Semantic 3D occupancy mapping through efficient high order CRFs)," in *Proceedings of the RSJ International Conference on Intelligent Robots and Systems (IROS*, pp. 590–597, IEEE, Vancouver, BC, Canada, September 2017.

[32] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: real-time recognition, tracking and reconstruction of multiple moving objects," in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 10–20, IEEE, Munich, Germany, October 2018.

[33] M. Rünz and L. Agapito, "Co-fusion: real-time segmentation, tracking and fusion of multiple objects," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pp. 4471–4478, IEEE, Singapore, May 2017.

[34] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pp. 2631–2638, IEEE, Hong Kong, China, May 2014.

[35] Y. Xiang and D. Fox, "DA-RNN: semantic mapping with data associated recurrent neural networks," in *Proceedings of the Robotics: Science and Systems XIII*, Seattle, WA, USA, July 2017.

[36] A. Aldoma, M. Zoltan-Csaba, F. Tombari et al., "Tutorial: point cloud library: three-dimensional object recognition and 6 DOF pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, 2012.

[37] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *Computer Vision—ECCV 2014*, Lecture Notes in Computer Science, vol. 8693, Cham, Switzerland, Springer, 2014.

[38] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge, (VOC2007) results," *Lecture Notes in Computer Science*, vol. 111, no. 1, pp. 98–136, 2007.

[39] B. Eckart and A. Kelly, "REM-Seg: a robust em algorithm for parallel segmentation and registration of point clouds," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Tokyo, Japan, November 2013.

[40] H. Oleynikova, Z. Taylor, M. Fehr et al., "Voxblox: incremental 3D euclidean signed distance fields for on-board MAV planning," in *Proceedings of the RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, September 2016.

[41] https://github.com/matterport/MaskRCNN.

[42] A. Dai, A. X. Chang, M. Savva et al., "ScanNet: richly-annotated 3D reconstructions of indoor scenes," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[43] I. Armeni, S. Sax, A. R. Zamir et al., "Joint 2D-3D-semantic data for indoor scene understanding," 2017, https://arxiv.org/abs/1702.01105.

[44] Z. Fang, J. Ren, S. Marshall et al., "Triple loss for hard face detection," *Neurocomputing*, 2020, In press.

[45] Q. H. Pham, B. S. Hua, D. T. Nguyen, and S.-K. Yeung, "Real-time Progressive 3D Semantic Segmentation for Indoor Scene," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, January 2019.

[46] Y. Yan, J. Ren, G. Sun et al., "Unsupervised image saliency detection with Gestalt-laws guided optimization and visual attention based refinement," *Pattern Recognition*, vol. 79, pp. 65–78, 2018.

[47] Y. Yan, J. Ren, H. Zhao et al., "Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos," *Cognitive Computation*, vol. 10, no. 1, pp. 94–104, 2018.

[48] Z. Wang, J. Ren, D. Zhang, M. Sun, and J. Jiang, "A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos," *Neurocomputing*, vol. 287, pp. 68–83, 2018.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 1279253, 8 pages
https://doi.org/10.1155/2020/1279253

Research Article

Multilabel Classification Using Low-Rank Decomposition

Bo Yang,[1,2] **Kunkun Tong**,[1] **Xueqing Zhao**,[1] **Shanmin Pang**,[3] **and Jinguang Chen**[1]

[1]*Shaanxi Key Laboratory of Clothing Intelligence, National and Local Joint Engineering Research Center for Advanced Networking & Intelligent Information Services, School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China*
[2]*School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China*
[3]*School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

Correspondence should be addressed to Jinguang Chen; chenjinguang@xpu.edu.cn

Received 19 December 2019; Accepted 21 February 2020; Published 7 April 2020

Guest Editor: Longzhuang Li

In the multilabel learning framework, each instance is no longer associated with a single semantic, but rather with concept ambiguity. Specifically, the ambiguity of an instance in the input space means that there are multiple corresponding labels in the output space. In most of the existing multilabel classification methods, a binary annotation vector is used to denote the multiple semantic concepts. That is, +1 denotes that the instance has a relevant label, while −1 means the opposite. However, the label representation contains too little semantic information to truly express the differences among multiple different labels. Therefore, we propose a new approach to transform binary label into a real-valued label. We adopt the low-rank decomposition to get latent label information and then incorporate the information and original features to generate new features. Then, using the sparse representation to reconstruct the new instance, the reconstruction error can also be applied in the label space. In this way, we finally achieve the purpose of label conversion. Extensive experiments validate that the proposed method can achieve comparable to or even better results than other state-of-the-art algorithms.

1. Introduction

Classification is a high-frequency vocabulary in machine learning. We often say that classification generally refers to single-label classification, that is, an object is given a category. In multilabel learning, the meaning of classification is multilabel classification. Specifically, an instance is associated with more than one class label simultaneously. Multilabel learning has many application fields, such as web mining [1–3], text categorization [4–6], multimedia contents annotation [7–11], and bioinformatics [12–14].

In recent years, the field of multilabel learning has gradually attracted significant attention. A variety of algorithms have been proposed, which can be basically divided into two categories [15]: algorithm adaptation and problem transformation. The core idea of the former is to transform the previous supervised learning algorithm so that it can be used to solve multilabel learning problems, such as ML-kNN [16], while the latter is to convert the multilabel learning problem into other known problems to solve, such as BR [17]. Some multilabel algorithms solve the multilabel learning problem without using the correlation among different labels, such as LIFT [18]. The main idea of the LIFT is to obtain the identifying characteristics of each label and build a new feature space. It first obtains the positive and negative examples corresponding to each label and then performs cluster analysis on the corresponding set of examples to obtain the cluster centers and finally uses the cluster centers to construct the label-specific features. In the process of solving the multilabel learning problem, LIFT does not consider label correlations; hence, it can be regarded as a new feature conversion method. Some algorithms consider the label correlation [19–25] for solving the multilabel learning problem. For example, the basic idea in [20] is to model the correlation among labels based on the Bayesian network and to achieve efficient learning by using the approximate strategy. Indeed, the rational use of the correlation among labels can effectively boost the

performance of multilabel classification. For example, if an image has labels "football" and "rainforest," it is likely to be labeled "Brazil". It has a low probability of being labeled "river" if a document is annotated with "desert". Therefore, how to effectively explore and make full use of label correlations is a crucial problem for multilabel learning.

In fact, for an object with multiple labels, the importance of the related labels is still different. Although the importance of each label is not given directly, we can judge the importance of each label through external observation. Generally speaking, the larger the proportion in the original object, the more important the corresponding label. Accordingly, how to accurately express the importance of the label is also a challenge.

The method in [26] decomposes the original output space in order to obtain potential label semantic information, which can effectively increase the ability of the subsequent feature selection. Motivated by the decomposition of the label space in [26], in the paper, we propose a method named label low-rank decomposition (LLRD) for multilabel classification. The LLRD algorithm first performs low-rank decomposition on the label matrix, then combines the decomposed results with the original features to form new features, and mines the structural information of the feature through sparse reconstruction. Third, it transforms the binary label into the real-valued and finally converts the classification problem into a regression problem.

The contribution of this paper is as follows:

(1) Utilize low-rank decomposition to reveal the global label correlations and achieve good classification results

(2) Combine the low-rank decomposition results with the original features reducing the information loss in the subsequent label transformation process

(3) Carry out extensive experiments on different field datasets to verify the effectiveness of different algorithms

2. Materials and Methods

2.1. Datasets. In this experiment, a total of 13 datasets were used covering four fields: audio, text, image, and biology. All these data resources can be collected from Mulan (http://mulan.sourceforge.net/datasets.html) and Meka (http://meka.sourceforge.net/#datasetsru). Table 1 gives the specific details of the datasets. The number of instances, label space, and the dimension of features are denoted by |S|, L(S), and D(S), respectively. LDen (S) is the density of label, which is the result of the normalization of label cardinality LCard(S).

2.2. Notations. Formally, suppose $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional input space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the output domain of q class labels. Let $\mathcal{D} = \{(x_i, y_i) | 1 \le i \le p\}$ be the multilabel training dataset with p examples, where $x_i \in \mathcal{X}$ is a d-dimensional instance vector and $y_i \subseteq \mathcal{Y}$ is the label vector corresponding to x_i. Let

Table 1: Properties of the experimental datasets.

Datasets	|S|	D(S)	L(S)	LCard(S)	LDen(S)	Domain
cal500	502	68	174	26.044	0.150	Audio
Emotions	593	72	6	1.868	0.311	Audio
Medical	978	1449	45	1.245	0.028	Text
Llog	1460	1004	75	1.180	0.016	Text
Image	2000	294	5	1.236	0.247	Image
Scene	2407	294	5	1.074	0.179	Image
Yeast	2417	103	14	4.237	0.303	Biology
Slashdot	3782	1079	22	1.181	0.054	Text
rcv1subset1	6000	500	101	2.880	0.029	Text
rcv1subset2	6000	500	101	2.634	0.026	Text
rcv1subset3	6000	500	101	2.614	0.026	Text
rcv1subset4	6000	500	101	2.484	0.025	Text
rcv1subset5	6000	500	101	2.642	0.026	Text

$X = [x_1, x_2, \ldots, x_p] \in \mathbb{R}^{d \times p}$ represent the input data matrix, and $X_i = [x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p]$ denote the matrix from which x_i is removed from X. Let $Y = [y_1, y_2, \cdots, y_p] \in \{-1, 1\}^{q \times p}$ is a matrix composed of label vector.

2.3. The Process of LLRD. First, LLRD decomposes the label matrix with low-rank method. In the framework of multilabel learning, label matrix is often considered to be low rank [27, 28] due to the existence of label correlations. Low-rank structure is also a way to explore the global relationship between labels. Therefore, we can perform low-rank decomposition on the label matrix. Assuming that the rank of Y is $r < q$, Y can be written as follows:

$$Y \simeq AB, \tag{1}$$

where $A \in \mathbb{R}^{q \times r}$ represents the dependency of $B \in \mathbb{R}^{r \times p}$ on the original label space and B is a mapping of the original label and also contains label correlation information.

Second, we combine B with X to form a new feature space $N = [X; B][n_1, n_2, \ldots, n_p] = \in \mathbb{R}^{(r+d) \times p}$. In order to reveal the inner structure of the feature space, we use sparse reconstruction [29] method to model the relationship between the training instances. Specifically, we use $W[s_{ij}]_{=p \times p}$ to represent the training object relationship matrix, where s_{ij} is a measure of the relationship between n_i and n_j. Let $S_i = [s_{1i}, \ldots, s_{i-1,i}, s_{i+1,i}, \ldots, s_{pi}]^{\mathrm{T}}$ denote the corresponding sparse reconstruction coefficient related to n_i. According to the sparse representation theory, S_i can be calculated as follows:

$$\min_{S_i} \left\| N_i S_i - n_i \right\|_2^2 + \eta \left\| S_i \right\|_1, \tag{2}$$

where $N_i = [n_1, n_2, \ldots, n_{i-1}, n_{i+1}, \ldots, n_p]$ represent a combination of all training instances except n_i. We can solve the above problem using alternating direction method of multiplier [30].

Third, we transform the original binary label set $y_i = (l_{i1}, l_{i2}, \ldots, l_{iq})^{\mathrm{T}}$ associated with any x_i in the training set into a real-valued label vector $c_i = (c_{i1}, c_{i2}, \ldots, c_{iq})^{\mathrm{T}}$, where $l_{ij} \in \{-1, 1\}$ and $c_{ij} \in \mathbb{R}$. Because the real value contains more information, and through the size of the value, we can also infer the importance of the label. Since the input space

and the label space are often interrelated, it is assumed that the relationship between n_i and n_j in the input space also exists between c_i and c_j in the label space. Accordingly, the representation errors of different elements in the label space can be written as follows:

$$\min_{C} \sum_{i=1}^{p} \left\| c_i - \sum_{j=1}^{p} s_{ij} c_j \right\|_2^2 \quad s.t.\ k_1 \le l_{ij} c_{ij} \le k_2 \quad (1 \le i \le p, 1 \le j \le q), \tag{3}$$

where $c = [c_1, c_2, \ldots, c_p]$. The above quadratic programming problem can be solved by mature tools related to quadratic programming. The original multilabel classification problem can be transferred into a multioutput regression problem. There are many solutions [31] to solve it. The learning of LLRD method contains three phases: low-rank decomposition, sparse reconstruction, and multioutput regression. The time complexity of low-rank decomposition and sparse reconstruction is $O(d^2 p + d^3)$. If we choose multioutput support vector regression to realize the classification, the time complexity is $O(qp^3)$. Thus, the total complexity of LLRD is $O(d^2 p + d^3 + qp^3)$.

3. Results and Discussion

3.1. Experiment Setup. In this subsection, we investigate comparisons between our LLRD and other six multilabel learning methods on six multilabel evaluation criteria, which include two categories: example-based and label-based metrics [32]. The example-based metric is to first obtain the performance of the learning system on each test example and finally returns the average of the entire test set. Unlike the above example-based metric, the label-based metric first returns the performance of the system on each label and finally gets the macro/microaveraged $F1$ value on all labels.

In this paper, *one-error*, *coverage*, *ranking loss*, and *average precision* are employed for example-based performance evaluation. And *macroaveraging* and *microaveraging* $F1$ are label-based metrics. For example-based metrics except *average precision*, as their values increase, it means that the performance of the algorithm is worse. For the remaining metrics, their values are proportional to the performance of the algorithm.

Let $T = \{(x_i, y_i)\}_{i=1}^{m} \subset \mathbb{R}^d \times \{+1, -1\}^q$ be the multilabel test set and $f(x, l)$ can be seen as the confidence of l being the corresponding label associating with x. In addition, $f(x, l)$ can be converted into a ranking function $\text{rank}_f(\mathbf{x}, l)$. If $f(x, l_1) > f(x, l_2)$ holds, then the corresponding ranking function has $\text{rank}_f(x, l_1) < \text{rank}_f(x, l_2)$.

The six evaluation criteria for the algorithm used in the paper are defined as follows:

(1) *One-error*:

$$\text{one-error}(f) = \frac{1}{m} \sum_{i=1}^{m} \left[\left[\arg\max_{l \in \mathcal{Y}} f(x_i, l) \right] \notin y_i \right]. \tag{4}$$

(2) *Coverage*:

$$\text{coverage}(f) = \frac{1}{m} \sum_{i=1}^{m} \max_{l \in y_i} \text{rank}_f(x_i, l) - 1. \tag{5}$$

(3) *Ranking loss*:

$$\text{rloss}(f) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{|y_i||\overline{y}_i|} \left| \{(l', l'') | f(x_i, l') \le f(x_i, l''), \cdot (l', l'') \in y_i \times \overline{y}_i\} \right|. \tag{6}$$

(4) *Average precision*:

$$\text{avgprec}(f) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{|y_i|} \sum_{l \in y_i} \frac{\left| \{l' | \text{rank}_f(x, l') \le \text{rank}_f(x_i, l), l' \in y_i\} \right|}{\text{rank}_f(x_i, l)}. \tag{7}$$

(5) *Macroaveraging* $F1$:

$$F1_{\text{macro}}(h) = \frac{1}{q} \sum_{j=1}^{q} \frac{2\text{TP}_j}{2\text{TP}_j + \text{FN}_j + \text{FP}_j}. \tag{8}$$

(6) *Microaveraging* $F1$:

$$F1_{\text{micro}}(h) = \frac{2 \sum_{i=1}^{q} \text{TP}_j}{2 \sum_{i=1}^{q} \text{TP}_j + \sum_{i=1}^{q} \text{FN}_j + \sum_{i=1}^{q} \text{FP}_j}, \tag{9}$$

where FN_j, TN_j, FP_j, and TP_j indicate the number of false-negative, true-negative, false-positive, and true-positive instances with regard to l_j.

In order to test the effectiveness of LLRD, we chose six multilabel learning algorithms MLFE [33], RAKEL [34], ML^2 [35], CLR [36], LIFT [18], and RELIAB [37] for performance comparison. MLFE makes full use of the intrinsic information in feature space, making the semantics of the label space more abundant. The specific parameters of MLFE are set as follows: $\rho = 1$, $c_1 = 1$, $c_2 = 2$, and β_1, β_2, and β_3 searched from {1, 2, . . .,10}, {1, 10, 15}, and {1, 10}. RAKEL is a high-order approach. The basic idea of the algorithm is to transform the multilabel learning problem into integration of multiclass classification

Table 2: Performance of each multilabel algorithm (mean ± std. deviation) on the regular-scale datasets.

Comparing algorithms	cal500	Emotions	Medical	Llog	Image	Scene	Yeast
One-error ↓							
LLRD	0.136 ± 0.041	**0.248 ± 0.048**	**0.125 ± 0.031**	**0.657 ± 0.038**	**0.244 ± 0.018**	**0.116 ± 0.019**	**0.217 ± 0.013**
MLFE	0.168 ± 0.049	0.259 ± 0.050	0.131 ± 0.030	0.672 ± 0.041	0.257 ± 0.031	0.127 ± 0.022	0.233 ± 0.026
LIFT	0.125 ± 0.049	0.251 ± 0.027	0.156 ± 0.041	0.664 ± 0.034	0.276 ± 0.026	0.132 ± 0.012	0.226 ± 0.021
RELIAB	**0.116 ± 0.030**	0.255 ± 0.041	0.163 ± 0.028	0.754 ± 0.035	0.342 ± 0.032	0.258 ± 0.011	0.255 ± 0.016
ML^2	0.201 ± 0.090	0.261 ± 0.045	0.135 ± 0.032	0.674 ± 0.051	0.260 ± 0.027	0.144 ± 0.019	0.246 ± 0.034
CLR	0.243 ± 0.058	0.310 ± 0.019	0.362 ± 0.009	0.841 ± 0.036	0.449 ± 0.013	0.331 ± 0.031	0.234 ± 0.022
RAKEL	0.622 ± 0.065	0.289 ± 0.032	0.237 ± 0.032	0.871 ± 0.028	0.397 ± 0.019	0.314 ± 0.030	0.291 ± 0.031
Coverage ↓							
LLRD	0.774 ± 0.021	0.282 ± 0.034	**0.029 ± 0.009**	0.194 ± 0.025	**0.157 ± 0.010**	**0.008 ± 0.009**	**0.447 ± 0.010**
MLFE	0.769 ± 0.024	0.283 ± 0.030	0.033 ± 0.010	0.200 ± 0.027	0.162 ± 0.018	0.012 ± 0.008	0.449 ± 0.011
LIFT	0.753 ± 0.015	**0.271 ± 0.023**	0.040 ± 0.014	0.164 ± 0.007	0.172 ± 0.013	0.026 ± 0.007	0.454 ± 0.017
RELIAB	**0.746 ± 0.019**	0.306 ± 0.020	0.044 ± 0.013	**0.155 ± 0.013**	0.185 ± 0.007	0.114 ± 0.004	0.457 ± 0.015
ML^2	0.814 ± 0.033	0.292 ± 0.044	0.035 ± 0.013	0.201 ± 0.026	0.164 ± 0.009	0.010 ± 0.007	0.461 ± 0.016
CLR	0.789 ± 0.010	0.330 ± 0.011	0.073 ± 0.041	0.182 ± 0.050	0.233 ± 0.017	0.122 ± 0.011	0.484 ± 0.020
RAKEL	0.958 ± 0.011	0.335 ± 0.031	0.077 ± 0.014	0.332 ± 0.021	0.249 ± 0.006	0.161 ± 0.007	0.553 ± 0.016
Ranking loss ↓							
LLRD	0.185 ± 0.011	**0.144 ± 0.028**	0.018 ± 0.007	0.185 ± 0.022	**0.129 ± 0.010**	**0.042 ± 0.008**	**0.163 ± 0.008**
MLFE	0.188 ± 0.010	0.146 ± 0.030	0.014 ± 0.007	0.191 ± 0.025	0.134 ± 0.017	0.046 ± 0.010	0.167 ± 0.011
LIFT	**0.178 ± 0.008**	**0.144 ± 0.026**	0.029 ± 0.009	0.148 ± 0.014	0.148 ± 0.012	0.054 ± 0.015	0.164 ± 0.013
RELIAB	0.182 ± 0.007	0.165 ± 0.021	0.026 ± 0.008	**0.134 ± 0.011**	0.176 ± 0.008	0.076 ± 0.007	0.185 ± 0.021
ML^2	0.205 ± 0.021	0.153 ± 0.033	**0.011 ± 0.009**	0.194 ± 0.027	0.136 ± 0.012	0.050 ± 0.007	0.175 ± 0.015
CLR	0.231 ± 0.020	0.181 ± 0.020	0.072 ± 0.051	0.137 ± 0.028	0.241 ± 0.015	0.098 ± 0.021	0.196 ± 0.009
RAKEL	0.359 ± 0.012	0.213 ± 0.019	0.066 ± 0.019	0.281 ± 0.034	0.244 ± 0.016	0.155 ± 0.023	0.243 ± 0.010
Average precision ↑							
LLRD	**0.506 ± 0.018**	0.819 ± 0.031	**0.905 ± 0.020**	**0.421 ± 0.033**	**0.841 ± 0.009**	**0.934 ± 0.010**	**0.775 ± 0.008**
MLFE	0.490 ± 0.017	0.812 ± 0.032	0.901 ± 0.021	0.410 ± 0.029	0.835 ± 0.019	0.928 ± 0.013	0.766 ± 0.016
LIFT	0.502 ± 0.021	**0.824 ± 0.024**	0.880 ± 0.030	0.416 ± 0.031	0.820 ± 0.018	0.922 ± 0.008	0.768 ± 0.018
RELIAB	0.497 ± 0.016	0.801 ± 0.021	0.869 ± 0.020	0.405 ± 0.041	0.781 ± 0.009	0.851 ± 0.008	0.751 ± 0.010
ML^2	0.481 ± 0.030	0.816 ± 0.031	0.898 ± 0.022	0.404 ± 0.031	0.832 ± 0.014	0.930 ± 0.009	0.759 ± 0.020
CLR	0.425 ± 0.034	0.770 ± 0.019	0.695 ± 0.032	0.312 ± 0.059	0.722 ± 0.015	0.801 ± 0.012	0.755 ± 0.006
RAKEL	0.343 ± 0.009	0.772 ± 0.037	0.798 ± 0.018	0.228 ± 0.020	0.731 ± 0.017	0.777 ± 0.023	0.717 ± 0.007
Macroaveraging $F1$ ↑							
LLRD	0.231 ± 0.026	**0.676 ± 0.051**	**0.736 ± 0.050**	0.408 ± 0.028	**0.666 ± 0.024**	**0.800 ± 0.016**	0.420 ± 0.030
MLFE	0.239 ± 0.025	0.668 ± 0.050	0.702 ± 0.056	**0.415 ± 0.041**	0.655 ± 0.021	0.787 ± 0.015	0.430 ± 0.024
LIFT	0.179 ± 0.014	0.651 ± 0.035	0.694 ± 0.052	0.392 ± 0.045	0.624 ± 0.033	0.788 ± 0.018	0.377 ± 0.019
RELIAB	**0.288 ± 0.015**	0.639 ± 0.038	0.686 ± 0.058	0.394 ± 0.031	0.568 ± 0.030	0.671 ± 0.021	0.409 ± 0.023
ML^2	0.226 ± 0.024	0.656 ± 0.045	0.686 ± 0.058	0.382 ± 0.035	0.652 ± 0.018	0.783 ± 0.015	0.438 ± 0.017
CLR	0.220 ± 0.017	0.604 ± 0.032	0.616 ± 0.118	0.402 ± 0.056	0.523 ± 0.027	0.635 ± 0.013	0.386 ± 0.016
RAKEL	0.195 ± 0.010	0.615 ± 0.030	0.679 ± 0.037	0.377 ± 0.054	0.545 ± 0.018	0.654 ± 0.012	**0.441 ± 0.011**
Microaveraging $F1$ ↑							
LLRD	0.325 ± 0.011	**0.692 ± 0.048**	**0.814 ± 0.030**	0.126 ± 0.027	**0.665 ± 0.024**	**0.792 ± 0.017**	**0.656 ± 0.011**
MLFE	0.384 ± 0.017	0.683 ± 0.047	0.785 ± 0.031	0.137 ± 0.032	0.653 ± 0.024	0.781 ± 0.015	0.643 ± 0.013
LIFT	0.313 ± 0.013	0.664 ± 0.015	0.763 ± 0.031	0.168 ± 0.034	0.625 ± 0.031	0.779 ± 0.022	0.650 ± 0.016
RELIAB	**0.454 ± 0.011**	0.647 ± 0.038	0.748 ± 0.024	**0.188 ± 0.028**	0.562 ± 0.021	0.639 ± 0.013	0.631 ± 0.015
ML^2	0.366 ± 0.013	0.674 ± 0.042	0.780 ± 0.021	0.074 ± 0.031	0.650 ± 0.019	0.776 ± 0.018	0.635 ± 0.018
CLR	0.330 ± 0.012	0.626 ± 0.029	0.606 ± 0.143	0.165 ± 0.050	0.531 ± 0.008	0.634 ± 0.017	0.623 ± 0.010
RAKEL	0.356 ± 0.025	0.648 ± 0.024	0.669 ± 0.016	0.155 ± 0.019	0.533 ± 0.005	0.645 ± 0.009	0.637 ± 0.011

problem. We use the default settings recommended by RAKEL algorithm, namely, $k = 3$, ensemble size $n = 2q$. For ML^2, respective parameter values are recorded as follows: $\lambda = 1$, $K = l + 1$ and C_1 and C_2 selected from $\{1, 2, \ldots, 10\}$. ML^2 is the first multilabel learning algorithm to attempt to explore manifolds at the label level. CLR is a second-order problem transformation method. It solves the problem of multilabel classification by using label ranking, in which ranking among labels is implemented by pairwise comparison. The associated parameter ensemble size is set to $\binom{q}{2}$. LIFT uses different feature sets to distinguish different labels by clustering positive and negative examples. The value of ratio parameter r is 0.1, as suggested in [18]. RELIAB utilizes the implicit relative information of label to achieve the task of multilabel learning. The parameters τ and β take values from $\{0.1, 0.15, \ldots, 0.5\}$ and $\{0.001, 0.01, \ldots, 10\}$, respectively. For LLRD, $\eta = 1$, r can be selected from $\{1, 2, \ldots, q-1\}$. In a word, the parameter settings of the comparison algorithm are as recommended in the related papers.

Table 3: Performance of each multilabel algorithm (mean ± std. deviation) on the large-scale datasets.

Comparing algorithms	Slashdot	rcv1subset1	rcv1subset2	rcv1subset3	rcv1subset4	rcv1subset5
One-error ↓						
LLRD	**0.363 ± 0.026**	0.414 ± 0.013	0.411 ± 0.017	0.416 ± 0.029	**0.317 ± 0.015**	0.401 ± 0.018
MLFE	0.374 ± 0.027	0.406 ± 0.018	0.399 ± 0.013	0.402 ± 0.025	0.328 ± 0.013	0.392 ± 0.008
LIFT	0.393 ± 0.033	0.427 ± 0.011	0.434 ± 0.017	0.441 ± 0.020	0.363 ± 0.019	0.430 ± 0.019
RELIAB	0.508 ± 0.022	0.449 ± 0.015	0.458 ± 0.028	0.454 ± 0.012	0.433 ± 0.024	0.423 ± 0.009
ML^2	0.370 ± 0.025	**0.404 ± 0.017**	**0.395 ± 0.018**	**0.398 ± 0.021**	0.323 ± 0.021	**0.388 ± 0.010**
CLR	0.965 ± 0.013	0.513 ± 0.022	0.515 ± 0.009	0.518 ± 0.028	0.472 ± 0.031	0.521 ± 0.021
RAKEL	0.602 ± 0.009	0.605 ± 0.013	0.574 ± 0.012	0.585 ± 0.022	0.561 ± 0.022	0.614 ± 0.009
Coverage ↓						
LLRD	0.107 ± 0.010	**0.125 ± 0.008**	**0.121 ± 0.009**	**0.123 ± 0.006**	0.092 ± 0.004	**0.116 ± 0.009**
MLFE	0.126 ± 0.013	0.136 ± 0.005	0.130 ± 0.010	0.129 ± 0.007	0.094 ± 0.007	0.124 ± 0.007
LIFT	0.112 ± 0.008	0.144 ± 0.020	0.135 ± 0.008	0.156 ± 0.008	0.113 ± 0.012	0.148 ± 0.013
RELIAB	0.131 ± 0.007	0.152 ± 0.012	0.128 ± 0.014	0.144 ± 0.011	0.105 ± 0.020	0.131 ± 0.014
ML^2	**0.103 ± 0.011**	0.138 ± 0.008	0.132 ± 0.010	0.126 ± 0.006	**0.078 ± 0.006**	0.129 ± 0.009
CLR	0.254 ± 0.003	0.146 ± 0.018	0.141 ± 0.007	0.137 ± 0.010	0.109 ± 0.018	0.136 ± 0.011
RAKEL	0.226 ± 0.020	0.426 ± 0.023	0.372 ± 0.016	0.381 ± 0.014	0.365 ± 0.009	0.388 ± 0.020
Ranking loss ↓						
LLRD	**0.090 ± 0.010**	**0.049 ± 0.004**	**0.050 ± 0.004**	**0.052 ± 0.002**	0.038 ± 0.002	**0.047 ± 0.003**
MLFE	0.107 ± 0.013	0.052 ± 0.002	0.055 ± 0.007	0.055 ± 0.002	0.040 ± 0.004	0.050 ± 0.003
LIFT	0.098 ± 0.016	0.058 ± 0.007	0.057 ± 0.009	0.068 ± 0.004	0.059 ± 0.010	0.055 ± 0.007
RELIAB	0.124 ± 0.003	0.066 ± 0.010	0.063 ± 0.008	0.062 ± 0.004	0.052 ± 0.006	0.063 ± 0.005
ML^2	0.103 ± 0.012	0.056 ± 0.004	0.057 ± 0.004	0.056 ± 0.003	**0.031 ± 0.003**	0.050 ± 0.004
CLR	0.237 ± 0.008	0.062 ± 0.011	0.066 ± 0.008	0.065 ± 0.012	0.047 ± 0.006	0.071 ± 0.005
RAKEL	0.211 ± 0.019	0.226 ± 0.019	0.215 ± 0.017	0.230 ± 0.015	0.235 ± 0.014	0.214 ± 0.016
Average precision ↑						
LLRD	**0.725 ± 0.019**	0.611 ± 0.010	0.638 ± 0.011	0.634 ± 0.017	**0.717 ± 0.008**	0.643 ± 0.011
MLFE	0.712 ± 0.021	0.618 ± 0.016	0.645 ± 0.009	0.639 ± 0.014	0.708 ± 0.012	0.647 ± 0.012
LIFT	0.703 ± 0.010	0.586 ± 0.009	0.598 ± 0.012	0.595 ± 0.011	0.674 ± 0.013	0.598 ± 0.011
RELIAB	0.624 ± 0.014	0.578 ± 0.021	0.611 ± 0.011	0.614 ± 0.018	0.655 ± 0.018	0.604 ± 0.009
ML^2	0.715 ± 0.022	**0.621 ± 0.012**	**0.647 ± 0.013**	**0.643 ± 0.016**	**0.717 ± 0.013**	**0.650 ± 0.010**
CLR	0.269 ± 0.002	0.575 ± 0.013	0.584 ± 0.021	0.571 ± 0.032	0.614 ± 0.020	0.588 ± 0.013
RAKEL	0.522 ± 0.020	0.395 ± 0.012	0.445 ± 0.018	0.431 ± 0.014	0.450 ± 0.012	0.437 ± 0.016
Macroaveraging $F1$ ↑						
LLRD	0.427 ± 0.035	0.235 ± 0.020	0.259 ± 0.019	0.213 ± 0.031	0.300 ± 0.019	0.211 ± 0.020
MLFE	0.466 ± 0.035	0.198 ± 0.017	0.195 ± 0.056	0.202 ± 0.030	0.249 ± 0.021	0.204 ± 0.021
LIFT	0.429 ± 0.037	0.223 ± 0.025	0.186 ± 0.024	0.200 ± 0.031	0.238 ± 0.013	0.196 ± 0.031
RELIAB	0.425 ± 0.029	**0.342 ± 0.022**	**0.338 ± 0.016**	**0.348 ± 0.014**	**0.342 ± 0.028**	**0.352 ± 0.014**
ML^2	**0.472 ± 0.029**	0.216 ± 0.020	0.206 ± 0.024	0.195 ± 0.030	0.244 ± 0.023	0.208 ± 0.011
CLR	0.174 ± 0.032	0.285 ± 0.032	0.264 ± 0.021	0.272 ± 0.022	0.311 ± 0.031	0.305 ± 0.017
RAKEL	0.354 ± 0.037	0.269 ± 0.030	0.251 ± 0.014	0.255 ± 0.014	0.263 ± 0.014	0.274 ± 0.018
Microaveraging $F1$ ↑						
LLRD	0.496 ± 0.021	0.393 ± 0.013	0.381 ± 0.017	0.406 ± 0.027	0.470 ± 0.013	0.402 ± 0.018
MLFE	0.545 ± 0.019	0.373 ± 0.014	0.375 ± 0.031	0.392 ± 0.024	0.403 ± 0.020	0.381 ± 0.017
LIFT	0.510 ± 0.030	0.320 ± 0.017	0.353 ± 0.014	0.347 ± 0.018	0.342 ± 0.024	0.363 ± 0.008
RELIAB	0.453 ± 0.011	**0.408 ± 0.010**	**0.449 ± 0.008**	**0.451 ± 0.021**	**0.478 ± 0.016**	**0.454 ± 0.012**
ML^2	**0.556 ± 0.022**	0.371 ± 0.014	0.391 ± 0.010	0.383 ± 0.026	0.393 ± 0.022	0.410 ± 0.015
CLR	0.104 ± 0.005	0.367 ± 0.011	0.368 ± 0.024	0.320 ± 0.024	0.381 ± 0.015	0.372 ± 0.008
RAKEL	0.365 ± 0.020	0.359 ± 0.023	0.348 ± 0.016	0.341 ± 0.016	0.371 ± 0.015	0.342 ± 0.006

3.2. Experimental Results. For each dataset in our experiment, we adopt the tenfold cross-validation strategy. Our experimental results are mainly distributed in Tables 2 and 3, where we record the performance of different algorithms in different multilabel datasets. Specifically, the average and standard deviation of the corresponding evaluation criteria are recorded in the tables. For each evaluation metric, "↓" indicates "the smaller the better" and "↑" indicates "the larger the better". The best results are shown in bold form.

Table 4: The Friedman statistics F_F and the critical value.

Evaluation metric	F_F	Critical value
One-error	34.0909	2.2274
Coverage	20.3765	
Ranking loss	21.1642	
Average precision	39.8409	
Macroaveraging F1	2.6520	
Microaveraging F1	7.6088	

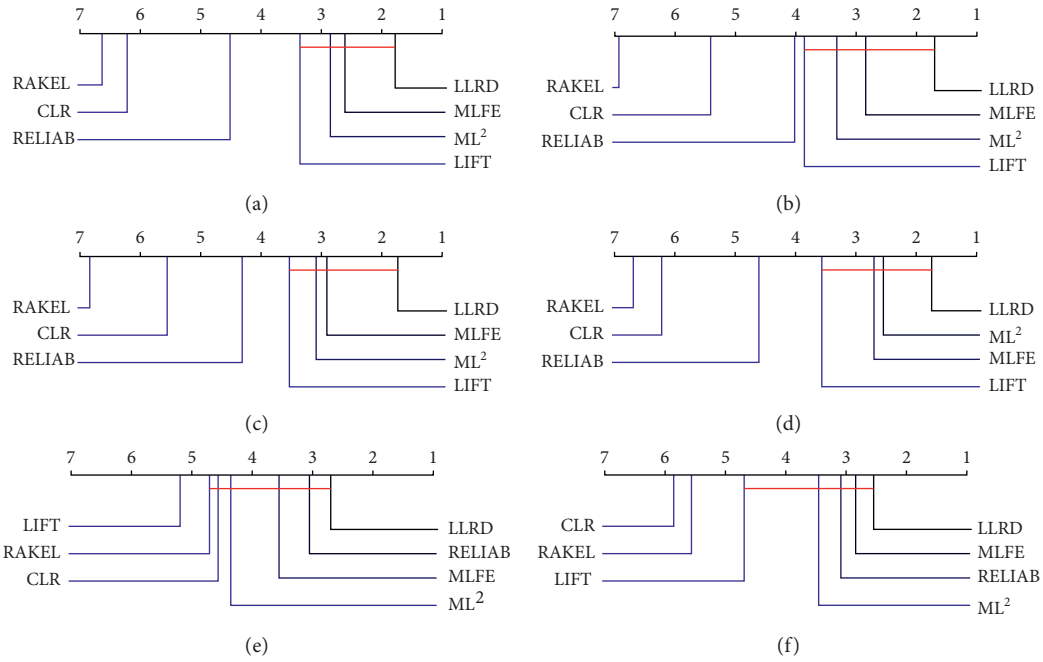


Figure 1: Comparison of LLRD (control algorithm) against other related approaches with the Bonferroni–Dunn test. Approaches that are not connected to LLRD are significantly different in performance from LLRD. (a) One-error. (b) Coverage. (c) Ranking loss. (d) Average precision. (e) Macroaveraging $F1$. (f) Microaveraging $F2$.

We use *Friedman test* [38] based on the average ranks for verifying whether the difference between algorithms is statistically significant. If the assumption that "all algorithms have equal performance" is rejected, it means that the performance of each algorithm is significantly different. As can be seen from the data presented in Table 4, the hypothesis that there is no significant difference among the algorithms is not valid under the condition of 0.05 significance level. Therefore, we need to conduct a post hoc test to further distinguish the various algorithms. Usually, there are two options for post hoc test, one is the Nemenyi test [38] and the other is the Bonferroni–Dunn test [39]. For k algorithms, the former needs to compare $k(k-1)/2$ times, while the latter only needs $k-1$ times in some cases. Thus, we choose the latter. The Bonferroni–Dunn test is used to test whether LLRD is more competitive than the comparative algorithm, in which LLRD plays a role of control algorithm. When the difference of average rank between two algorithms is more than one critical difference CD, the performance of two algorithms is obviously different. The CD value mentioned here can be calculated from $CD = q_\alpha \sqrt{k(k+1)/6N}$, where $k = 7$ and $N = 13$, when the significance level is 0.05, the corresponding $q_\alpha = 2.638$.

The CD diagram associated with LLRD and its comparison algorithm is shown in Figure 1. The numbers on the horizontal axis of the coordinate indicate the average rank value of each algorithm under different evaluation criteria. There is no significant difference in performance among the various algorithms connected by solid lines.

Through the analysis of the above experimental results, we can draw the following conclusions:

(1) In terms of the four evaluation criteria of *one-error*, *coverage*, *ranking loss*, and *average precision*, LLRD is obviously superior to RELIAB, RAKEL, and CLR.

(2) The smaller the average rank value, the better the performance of the corresponding. For LLRD, five of the average rank value in the six CD subdiagrams are optimal, which shows LLRD outperforms other algorithms.

(3) For regular-size datasets, LLRD ranks first in 69% of the cases under different evaluation criteria, while for large-scale datasets, it ranks first in 36.1%.

4. Conclusions

In this work, we propose a novel multilabel classification algorithm named LLRD, which adopts the low-rank decomposition to gain the internal information of label and further reduce the information loss of the label transformation via the new feature space. Experimental results show that the performance of the proposed LLRD is better than many state-of-the-art multilabel classification techniques. In the future, we will explore alternative models combining the low-rank decomposition and classification into a joint optimization problem for considering more complex correlation of labels.

Data Availability

The datasets used in our manuscript are all public datasets, which can be downloaded from "http://mulan.sourceforge.net/datasets.html" and "http://meka.sourceforge.net/#datasetsru".

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (2019YFC1521400), National Natural Science Foundation of China (61806159 and 61806160, 61972312), and China Postdoctoral Science Foundation (2018M631192).

References

[1] L. Tang, S. Rajan, and V. K. Narayanan, "Large scale multi-label classification via metalabeler," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 211–220, Madrid, Spain, July 2009.

[2] B. Yang, J. T. Sun, T. Wang, and Z. Chen, "Effective multi-label active learning for text classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 917–926, Paris, France, July 2009.

[3] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda, "Maximal margin labeling for multi-topic text categorization," in *Advances In Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., pp. 649–656, MIT Press, Cambridge, MA, USA, 2005.

[4] A. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Proceedings of the Working Notes of the AAAI Workshop on Learning for Text Categorization*, Orlando, FL, USA, July 1999.

[5] R. E. Schapire, Y. Singer, and Boostexter, "A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[6] N. Ueda and K. Saito, "Parametric mixture models for multi-label text," in *Advances In Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., pp. 721–728, MIT Press, Cambridge, MA, USA, 2003.

[7] F. Kang, R. Jin, and R. Sukthankar, "Correlated label propagation with application to multi-label learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1719–1726, New York, NY, USA, June 2006.

[8] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multilabel classification of music into emotions," in *Proceedings of the 9th I International Conference on Music Information Retrieval*, pp. 325–330, Philadephia, PA, USA, 2008.

[9] C. Sanden and J. Z. Zhang, "Enhancing multi-label music genre classification through ensemble techniques," in *Proceedings of the 34th SIGIR*, pp. 705–714, Beijing, China, July 2011.

[10] R. C. Hong, M. Wang, Y. Gao et al., "Image annotation by multiple-instance learning with discriminative feature mapping and selection," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 669–680, 2014.

[11] Y. Xia, L. Nie, L. Zhang, Y. Yang, R. Hong, and X. Li, "Weakly supervised multilabel clustering and its applications in computer vision," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3220–3232, 2016.

[12] Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances In Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., pp. 681–687, MIT Press, Cambridge, MA, USA, 2002.

[13] M. L. Zhang and Z. H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[14] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," *Data Mining and Knowledge Discovery Handbook*, Springer pp. 667–685, 2010.

[15] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, pp. 1–38, 2015.

[16] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[17] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[18] M.-L. Zhang and L. Wu, "Lift: multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2015.

[19] Y. H. Guo and S. C. Gu, "Multi-label classification using conditional dependency networks," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, Catalonia, Spain, July 2011.

[20] M. L. Zhang and K. Zhang, "Multi label learning by exploiting label dependency," in *Proceedings Of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 999–1008, Washington, DC, USA, 2010.

[21] B. Fu, G. D. Xu, Z. H. Wang, and L. B. Cao, "Leveraging supervised label dependency propagation for multi-label learning," in *Proceedings Of the IEEE 13th International Conference on Data Mining*, pp. 1061–1066, Dallas, TX, USA, December 2013.

[22] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *In Lecture Notes in Artificial Intelligence 5782, W*, M. G. Buntine and J. Shawe-Taylor, Eds., pp. 254–269, Springer, Berlin, Germany, 2009.

[23] J. Huang, G. R. Li, S. H. Wang, and Q. M. Huang, "Categorizing social multimedia by neighborhood decision using local pairwise label correlation," in *Proceedings Of the IEEE International Conference on Data Mining Workshop*, pp. 913–920, Shenzhen, China, December 2014.

[24] H. W. Liu, Z. J. Ma, S. C. Zhang, and X. D. Wu, "Penalized partial least square discriminant analysis with ℓ1-norm for multi-label data," *Pattern Recognition*, vol. 48, no. 5, pp. 1724–1733, 2015.

[25] S. J. Huang and Z. H. Zhou, "Multi-label learning by exploiting label correlations locally," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 949–955, Toronto, Canada, July 2012.

[26] J. Ling, J. D. Li, and H. Liu, "Exploiting multilabel information for noise-resilient feature selection," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 5, pp. 1–23, 2018.

[27] C. K. Yeh, W. C. Wu, W. J. Ko, and Y. C. F. Wang, "Learning deep latent space for multi-label classification," in *Proceedings Of the 31st AAAI Conference On Artificial Intelligence*, pp. 2838–2844, San Francisco, CA, USA, February 2017.

[28] K.-H. Huang and H.-T. Lin, "Cost-sensitive label embedding for multi-label classification," *Machine Learning*, vol. 106, no. 9-10, pp. 1725–1746, 2017.

[29] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2008.

[30] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2014.

[31] D. Tuia, J. Verrelst, L. Alonso, F. Perez-Cruz, and G. Camps-Valls, "Multioutput support vector regression for remote sensing biophysical parameter estimation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 804–808, 2011.

[32] M. L. Zhang and Z. H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[33] Q. W. Zhang, Y. Zhong, and M. L. Zhang, "Feature-induced labeling information enrichment for multi-label learning," in *Proceedings of the IEEE International Conference on Artificial Intelligence*, pp. 4446–4453, Taichung, Taiwan, January 2018.

[34] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2010.

[35] P. Hou, X. Geng, and M. L. Zhang, "Multi-label manifold learning," in *Proceedings of the 30th AAAI Conference On Artificial Intelligence*, Phoenix, AZ, USA, February 2016.

[36] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[37] Y. K. Li, M. L. Zhang, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," in *Proceedings Of 15th IEEE International Conference On Data Mining*, pp. 251–260, Atlantic City, NJ, USA, November 2015.

[38] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[39] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?" *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 152–161, 2016.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 7954393, 8 pages
https://doi.org/10.1155/2020/7954393

Research Article

Personalized Clothing Recommendation Based on User Emotional Analysis

Xueping Su,[1] **Meng Gao**,[1] **Jie Ren**,[1] **Yunhong Li**,[1] **and Matthias Rätsch**[2]

[1]*School of Electronics and Information, Xi'an Polytechnic University, Xi'an710048, China*
[2]*Interactive and Mobile Robotics and Artificial Intelligence, Department of Engineering, Reutlingen University, Reutlingen, Germany*

Correspondence should be addressed to Xueping Su; yifeichongtian1201@163.com

Received 6 November 2019; Revised 30 November 2019; Accepted 5 December 2019; Published 5 March 2020

Guest Editor: jinchang ren

With the continuous development of economy, consumers pay more attention to the demand for personalization clothing. However, the recommendation quality of the existing clothing recommendation system is not enough to meet the user's needs. When browsing online clothing, facial expression is the salient information to understand the user's preference. In this paper, we propose a novel method to automatically personalize clothing recommendation based on user emotional analysis. Firstly, the facial expression is classified by multiclass SVM. Next, the user's multi-interest value is calculated using expression intensity that is obtained by hybrid RCNN. Finally, the multi-interest value is fused to carry out personalized recommendation. The experimental results show that the proposed method achieves a significant improvement over other algorithms.

1. Introduction

With the rapid development of e-commerce, online shopping has become one of the main ways people spend shopping. On the one hand, there is too much information about online clothing, which may drown users in the mass clothing information; how to quickly choose the clothing they need and improve the shopping efficiency are crucial for the merchant. On the other hand, users have their own preferences and focus on individual needs of clothing. Therefore, research on the clothing personalized recommendation method is very important to improve the user's shopping efficiency and meet the user's personalized needs. However, in the traditional personalized recommendation, due to the lack of user information, the recommendation quality of the clothing recommendation system is not high enough to meet the user's expectations. In addition, personalized recommendation function is limited to recommending products related to the user or favoured by the user.

Furthermore, with the development of emotional computing and intelligent human-computer interface, the computers are required to perceive and understand human's expression. There are a variety of facial expression-recognition methods, but existing expression analysis frameworks are less likely to measure the intensity of expressions. In the field of human-computer interaction, the measurement of facial expression intensity is also very necessary; it can help computers understand people's emotions. For example, when a user browses clothing information, the intensity of his expression can reflect the degree of affection of clothing. The computer can use it to recommend the clothing of interest to the user. Therefore, we propose an efficient method for the personalized recommendation of clothing based on user emotional analysis. The novel contributions of the proposed work are as follows:

(1) The scheme of the hybrid recurrent convolutional neural network (RCNN) is proposed to compute the expression intensity. These implementations improve the precision of personalized clothing recommendations.

(2) To the best of our knowledge, this is the first time to describe the user's multi-interest value by combining

expression intensity and the expression duration, which well captures the user's preferences and improves the recall of personalized clothing recommendation.

The remainder of this paper is organized as follows. We first describe some related research for facial expression recognition and personalized clothing recommendation. In section 3, we introduce our method, focusing on facial expression recognition by multi-class support vector machine (SVM), computing the expression intensity that is obtained by the hybrid RCNN, fusing the user's multi-interest value and personalized recommendation. In section 4, we present detailed experimental results and compare the performance of our proposed method with other current approaches. Finally, we conclude with discussions.

2. Related Work

Deep neural networks have been successfully applied in computer vision, especially in face recognition, where the use of convolutional neural networks (CNNs) outperforms all the previously proposed methods, and the obtained results surpass the human performances [1–5]. Subsequently, Zhou et al. [6] proposed the recurrent convolutional neural network (RCNN) for object recognition by applying recurrent connections with the same layer. With fewer parameters, the RCNN achieved better results than the state-of-the-art CNNs by testing object recognition datasets [7]. The end-to-end RCNN framework can predict the pain intensity of each frame by considering sufficiently large historical frames while limiting the scale of the parameters within the model [6]. Besides that, the RCNN outputs continuous scores rather than discrete labels as in the problem of classification.

Facial expression is salient information to understand certain target's emotional situation. Most of the human emotional expressions are able to be observed on their face than any other signs. At the same time, the CNN is used for the facial expression recognition task with Tang [8], Bergstra [9], and Jeon et al. [10] and achieved the best performance on Kaggle facial expression recognition challenge. Tang used the CNN with linear-SVM instead of the SoftMax layer in the classification phase. His model performed the best accuracy of 69.77% on the challenge. Bergstra's model is concentrated to hyperparameter optimization. Jeon constructed a real-time facial expression recognizer using a deep neural network which is invariant to the subject. Soon after, many deep learning methods are used for facial expression recognition and have achieved good performance [11–14]. In summary, there are a variety of facial expression recognition methods, but the method of expression intensity based on deep learning is less [15].

Meanwhile, with the continuous development of economy, consumers pay more attention to the demand for personalization of clothing. Personalized clothing recommendation not only meets the personalized needs of consumers but also greatly saves time for consumers to choose clothes. Therefore, the personalized clothing recommendation has attracted the attention of domestic and foreign clothing experts, and the method of personalized clothing recommendation has emerged [16–18]. The clothing personalized recommendation system mainly obtains user preferences based on the user's purchase record, browsing history, and neighbouring user information analysis. There are problems such as cold start and low degree of personalization, which cannot satisfactorily satisfy the personalized recommendation effect.

In summary, we propose a novel method to automatically personalize clothing recommendation based on user multi-interest value, which is calculated using expression intensity that is obtained by the hybrid RCNN.

3. The Proposed Framework

3.1. Initialization Recommendation. Clothes are divided into multiple classes by the affinity propagation cluster. For each class of clothes $(\overline{c}_i, \sigma_i)$, $\overline{c}_i$ represents the mean of the class and σ_i represents the variance of the class. For the user u, we calculate the similarity between each class of clothes and the user, and recommend the suitable class of clothes for the user according to the ranking of similarity. The formula for calculating similarity between the class of clothes and user is as follows:

$$d_{cu}(\overline{c}_i, u) = \sqrt{\sum_{k=1}^{K} \left(\frac{(\overline{c}_{ik} - u_k)}{\sigma_{ik}} \right)^2},$$
$$S_{cu}(\overline{c}_i, u) = -\exp(d_{cu}(\overline{c}_i, u)), \quad i = 1, 2, \ldots, N, \tag{1}$$

where $d_{cu}(\overline{c}_i, u)$ is the Euclidean distance of the class of clothes $\overline{c}_i$ and the user u, $\overline{c}_{ik}$ and u_k are the k-th features of the class of clothes $\overline{c}_i$ and the user u, N is the total number of the class of clothes, and K is the feature dimension number. $S_{cu}(\overline{c}_i, u)$ represents the similarity between the class of clothes $\overline{c}_i$ and the user u.

Similarly, the other users are also divided into multiple clusters by the affinity propagation cluster. For each class of users $(\overline{u}_j, \sigma_j)$, $\overline{u}_j$ represents the mean of the class, σ_j represents the variance of the class. We calculate the similarity between each class of users and the user and recommend a suitable other class of users' clothes for the user according to the ranking of similarity. The formula for calculating similarity between the class of users and user is as follows:

$$d_{uu}(\overline{u}_j, u) = \sqrt{\sum_{k=1}^{K} \left(\frac{(\overline{u}_{jk} - u_k)}{\sigma_{jk}} \right)^2},$$
$$S_{uu}(\overline{u}_j, u) = -\exp(d_{uu}(\overline{u}_j, u)). \quad j = 1, 2, \ldots, M \tag{2}$$

where $d_{uu}(\overline{u}_j, u)$ is the Euclidean distance of the class of users $\overline{u}_j$ and the user u, $\overline{u}_{jk}$ and u_k are the k-th features of

other user $\overline{u}_j$ and the user u, M is the total number of the class of other users, and K is the feature dimension number. $S_{uu}(\overline{u}_j, u)$ represents the similarity between the class of users $\overline{u}_j$ and the user u.

3.2. Calculation of Expression Intensity. The high-definition camera was used to obtain the user's expression feedback of the initialization recommendation clothing video and recognize the user's expression in the video. We adopt the multiclass SVM method for dynamic expression recognition and use the method of the recurrent convolution neural network (RCNN) to evaluate the expression (happy, angry, etc.) intensity.

3.2.1. Facial Expression Recognition. We convert facial expression recognition into a classification problem, and recognition of the expression (happy, sad, etc.) in the video. The specific expression recognition framework is shown in Figure 1:

The specific steps are as follows:

Firstly, the video is transformed into a series of frames of image sequences, and the active shape model method is used for face detection, and the video volume is created [7]. Secondly, the Local Gabor Binary Pattern Histogram Sequence (LGBPHS) features [19] of the three planes XY, XT, and YT are extracted, and all the features of the video volume are combined as the features of the final image sequence (the specific steps are shown in Figure 2).

Multiclass SVM aims to assign labels to instances by using SVM, where the labels are drawn from a finite set of several elements (Sad, Happy, Angry, Disgust, Fear, Surprise, and Neutral). The dominant approach for doing this is to reduce the single multiclass problem into multiple binary classification problems. Common methods for such reduction include one versus all and one versus one. Both methods have been found to produce approximately similar results when dealing with face recognition. Compared with one versus one, one versus all constructed a much less number of decision planes. When the number of classes is large, the prediction speed is faster. In our paper, we use one versus all to train multiclass SVM.

We train multiclass SVM for expression recognition. The procedure is as follows. Firstly, for the samples of happy, we assign 1 as the class label (Sad/Angry/Disgust/Fear/Surprise/Neutral set to 2/3/4/5/6/7). Secondly, the samples of all emotions are trained for multiclass SVM.

3.2.2. Calculation of Expression Intensity

(1) Calculation of Expression Duration. We use the expression duration as one of the expression intensities, and calculate the expression duration based on the recognized frame expressions. The calculation formula is as follows:

$$I_i = \frac{\mathrm{TE}_i}{T_i}, \quad i = 1, 2, 3, \ldots, M, \tag{3}$$

where I_i is the time value of the i-th interest measure and TE_i is the duration of the interest in the i-th interest measure (expression: happy, angry, etc.). T_i is the viewing total time. Then, the time values of all interest measures are sorted in descending order. The smaller the serial number, the more interested in trying on the clothing. Set a threshold for I_i, and recommend it to the user if I_i is greater than the threshold.

(2) Evaluate of Expression Intensity. From the field of face recognition, the face model strained on several specific facial parts can significantly improve the recognition accuracy [20, 21]. Compared with the full-face model, the specific part model is able to extract more detailed information. For the sake of exploring the effectiveness of different face parts, we divide the entire face into several parts. In addition, based on the promising results obtained by the RCNN, we trained a hybrid RCNN using different face parts. The main idea of our method can be concluded as (1) train a hybrid RCNN based on the face region, eye region, and mouth region. (2) Concatenate the last fully connected layer of the hybrid RCNN to constitute the features.

To save computation and reduce the time consumption, we simplified the architecture of the RCNN [6]. The RCNN used in our paper contains one convolutional layer, three Recurrent Convolutional Layers (RCLs), three max pooling layers, and one fully connected layer. The first layer is the standard feed-forward convolutional layer without recurrent connections, followed by max pooling. Three RCLs are used with a max pooling layer in the middle. Between neighbouring RCLs, there are only feed-forward connections. The output of the third RCL follows a global max pooling layer, which outputs the maximum over every feature map, yielding a feature vector representing the image. The main architecture of the hybrid RCNN is shown in Figure 3.

The framework of the hybrid RCNN to evaluate expression intensity is shown in Figure 4. The specific steps are as follows.

For each frame of the face image in the video sequence, firstly, the active shape model method is used to detect face feature points [7]. Secondly, in the process of face aligning and warping, we warped every facial image in R, G, and B channels separately, then combined all channels back to get the final RGB warped faces. Third, the input samples of our hybrid RCNN structure should be no more than two dimensions, but to reserve the temporal information among frames and the spatial pixel information of warped facial images at the same time, each frame is converted into a one-dimensional vector by flattening. After flattening, we concatenated all 1D flattened warped facial images in frame order to achieve frame vector sequences.

3.2.3. Calculation of Interest Value. We combine the expression intensity and the expression duration as the user interest value. The calculation formula of interest value is as follows:

$$I_e = \alpha I_i + \beta I_t, \tag{4}$$

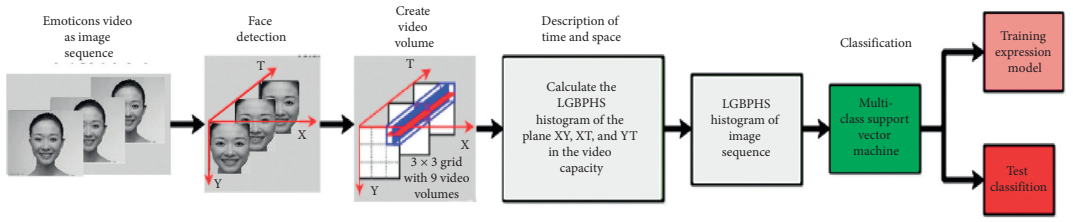


Figure 1: Expression recognition framework.

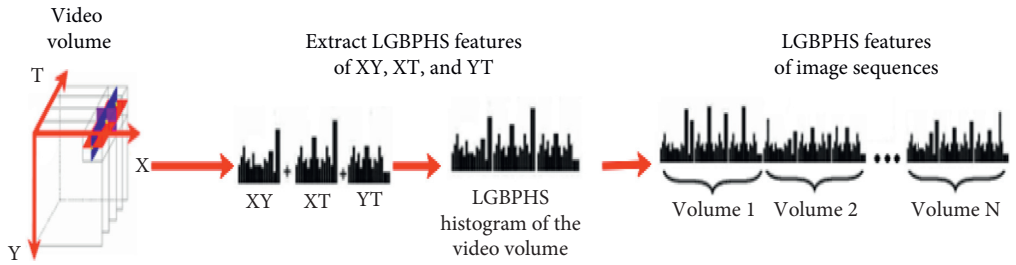


Figure 2: LGBPHS feature flow chart for extracting image sequences.

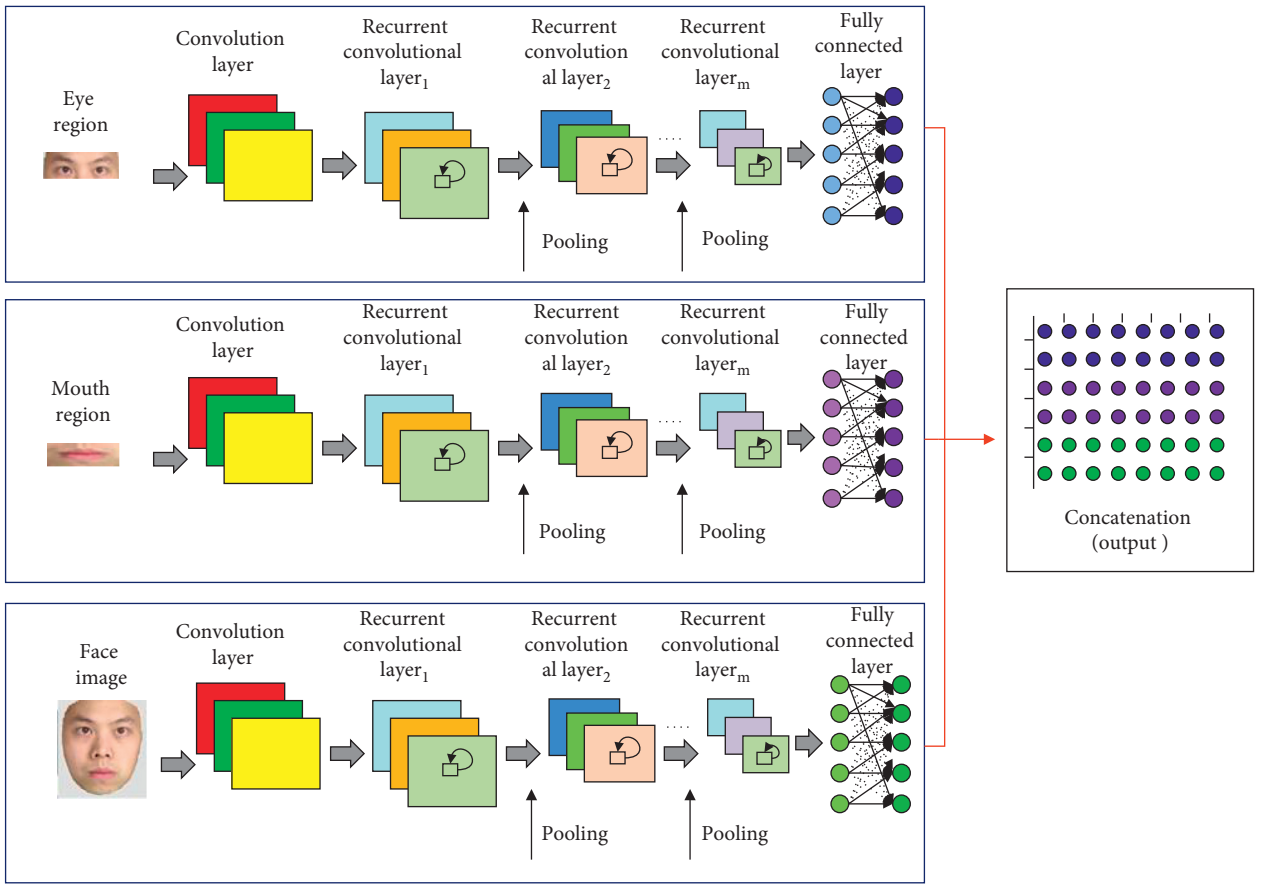


Figure 3: Primary architecture of the hybrid RCNN.

where I_e is the interest value, I_i and I_t are the duration value and intensity value of the expression, respectively, and α β are the weights value, and it meets the formula $\alpha + \beta = 1$.

3.3. Fusion User's Multi-Interest Value and Personalized Recommendation. We intend to calculate the user's multi-interest (colour, style, texture, price, etc.) value. The method

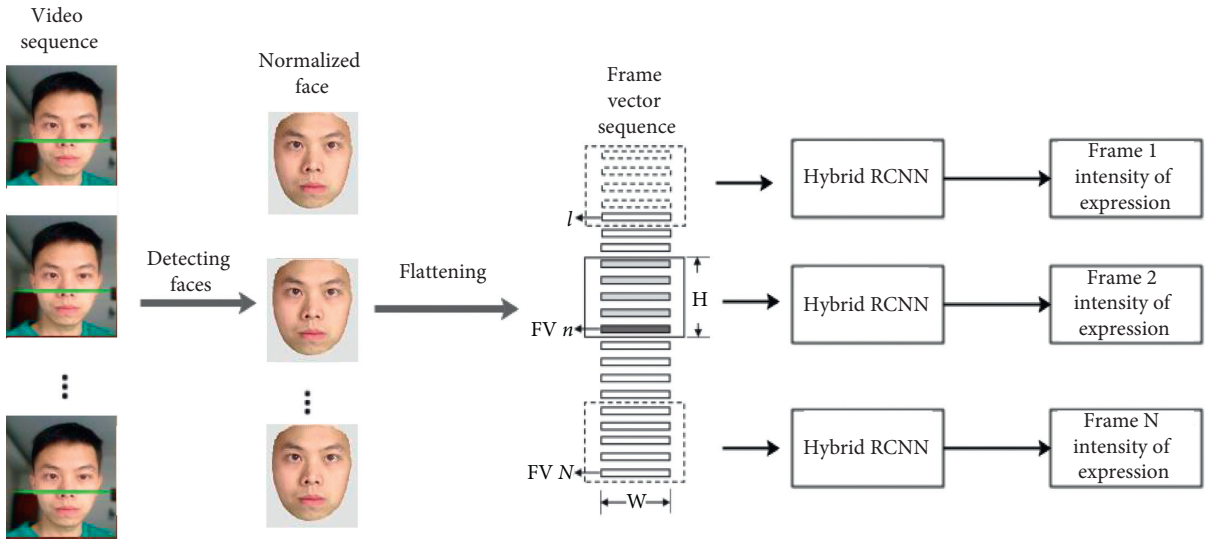


Figure 4: Frame diagram of the expression intensity evaluation method.

of multi-interest value fusion is adopted to carry out personalized recommendation.

Rank aggregation is the fusion of decision-making results which is expressed by the order list. Because the order list expresses the decision-making result, it is necessary to facilitate direct comparison of the different results, and contains a wealth of information for decision-making result. Therefore, we exploit the weighted Borda count method [22] to fuse multi-interest value. Ballots in the Borad count are counted by assigning a point value to each place in the hierarchy, and the choice with the largest number of points is selected. The Borda method scores each sequence of the list of interests (colour, style, texture, and price) linearly, and the score of objects i in the sequence τ_j of interest measure is estimated as follows:

$$B^j(i) = -\tau_j(i), \quad i = 1, 2, 3, \ldots, M, j = 1, 2, 3, \ldots, N; \tag{5}$$

where $\tau_j(i)$ is the ordinal function of the sequence τ_j of interest value, and it indicates the order of the object i in the interest value list τ_j, M is the total number of objects, and N is the total number of interest lists. The symbol in the formula is to place the value of the object in front of the order of bits higher. When J sequence of interest value is fused, the weighted Borda method is adopted according to the performance difference of sequences of different interest measure, which is:

$$B(i) = \sum_{j=1}^{J} w_j B^j(i), \quad i = 1, 2, 3, \ldots, M, j = 1, 2, 3, \ldots, N; \tag{6}$$

where $B^j(i)$ is the score of the object i in the j-th sequence of interest value, M is the total number of objects, N is the total number of interest value lists, and w_j is the weight of the interest measure sequence j. The calculation formula is as follows:

$$w_j = \mathrm{AP}_j, \quad j = 1, 2, 3, \ldots, N;$$
$$\sum_{j=1}^{J} w_j = 1, \quad 0 < w_j < 1, \tag{7}$$

where AP_j (average precision) is the average accuracy of the j-th sequence of interest value.

We sort the multi-interest (colour, style, texture, price, etc.) value and then use the weighted Borda method to perform rank aggregation. The final rank aggregation result is the sorting of the weighted scores from high to low, which gives a personalized recommendation based on the results.

4. Experiments

4.1. Dataset. Kaggle [23] facial expression recognition challenge database is used for training and testing performance. This dataset has 7 facial expression categories (angry, disgust, fear, happy, sad, surprise, and neutral), 28,709 training images, 3,589 validation images, and 3,589 test images. This dataset contains the human frontal face with various illumination, poses, and domains, and even cartoon characters are included. Moreover, in Kaggle facial recognition challenge training dataset, 7215 images are in the happy category and 436 images are in the disgust category.

Forty female university students from Soochow University conducted a verification experiment on the personalized recommendation system. Forty subjects had a rating of points for the system's recommended clothing, and it contains 132 sample pictures of women's winter wool coats [23]. Among them, 32 clothing images (See Figure 5) are used to train and 100 clothing images are used to recommend. Women's winter woollen coat has 7 attributes, of which the colour attribute has 8 attribute values, and the other 6 attributes have 4 attribute values.

Figure 5: Training images.

4.2. Experimental Setup. For the clothing sample, we let each image in the sample to automatically play at intervals of 3 seconds. During the playback, the user is asked to evaluate the clothing according to their own preferences, so as to obtain the user's evaluation form for the sample. E is used to indicate the user's evaluation of the sample clothing. The values of E are 1, 2, 3, 4, and 5, respectively, indicating that the consumer's evaluation of each garment in the sample is "very dislike, "dislike," "general," "like," and "very like." Similarity, the expression intensity values are graded into 5 intensity levels: [0,0.2] for level 1, the weakest level, [0.2, 0.4] is level 2, and so on, [0.8,1] is level 5.

The hybrid RCNN was implemented and run on two GeForce GTX TITAN Black GPUs. Initially, the weight of feed-forward/recurrent is set to 0.02, and the bias is set to 1. In addition, the parameter β in equation (4) are analysed in Figure 6. In order to better trade off average precision, average recall and average E, α is set to 0.2 and β is set to 0.8 for calculating interest value.

4.3. Results and Analysis. We compare the proposed method with Wang's method [16], Hu's method [17], and Melo's method [18]. Wang's method is tested on the same clothing dataset which we use in our experiments. Hu's method used the user interest degree to express the user preference model; the idea is similar to Melo's method and our paper. To evaluate the effectiveness of the proposed method, we select 40 people to evaluate the recommend clothing, and evaluate the performance by calculating the average precision, average recall, and average E. The experiment result is shown in Table 1.

As shown in Table 1, our proposed method obtained the better result for personalized clothing recommendation.

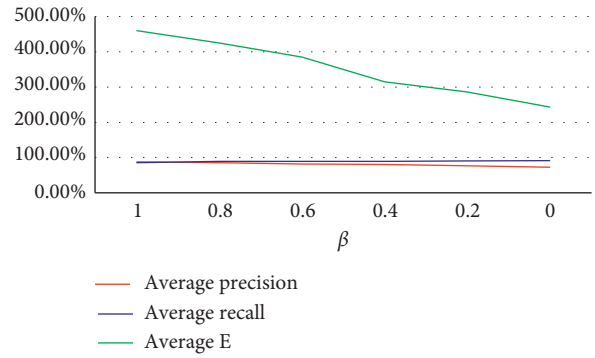


Figure 6: Compared results with varying β.

Because the measurement of facial expression intensity can help computers understand people's emotions, when a user browses clothing information, the intensity of his expression can reflect the degree of affection of the clothing. The computer can use it to recommend the clothing of interest to the user, which can satisfactorily satisfy the personalized recommendation effect.

In addition, we compare the classification accuracy of the facial expression recognition method with Tang's method [8] and Jeon's method [10]. These two methods are tested on the same facial expression dataset that we use in our experiments. The classification accuracy of facial expression is shown in Table 2.

As shown in Table 2, the average accuracy for all categories was 72.36% in our method. Accuracy for the happy and surprise category was higher than the others, but accuracy for the fear category was poor.

Meanwhile, we also compare the precision of classification of intensity levels for happy expressions with SVM

Table 1: Experiment result of different methods.

Method	Average precision (%)	Average recall (%)	Average E
Wang's method [16]	72.5	82.6	3.625
Hu's method [17]	72.3	80.5	3.86
Melo's method [18]	80.6	83.4	3.98
Our method	85.4	88.9	4.25

Table 2: Classification accuracy of facial expression in different methods.

Method	Angry (%)	Disgust (%)	Fear (%)	Happy (%)	Sad (%)	Surprise (%)	Neutral (%)
Tang's method [8]	48.2	55.4	38.6	69.7	50.8	70.9	59.2
Jeon's method [10]	61.1	67.3	49.5	88.7	65.2	83.2	69.2
Our method	65.4	69.9	55.8	90.6	66.8	86.7	71.3

Table 3: Precision of classification of intensity levels for happy expressions.

Method	Precision (%)
SVM [15]	81.3
RCNN [6]	84.2
Our method	90.6

[15] and the RCNN [6]. Although Zhou proposed an automatic frame-by-frame pain (not facial expression) intensity estimation framework in a video based on the RCNN [6], the solution to intensity estimation is similar.

As shown in Table 3, our proposed method obtained the better result for facial expression intensity estimation. Compared with the full-face model, the specific part model is able to extract more detailed information, so the face model strained on several specific facial parts can significantly improve the precision of expression intensity estimation.

5. Conclusion

We have presented a method for personalized recommendation of clothing based on the user's emotional analysis. Particularly, the hybrid RCNN is used to compute the expression intensity, which improves the precision of personalized clothing recommendation. In addition, to capture the user's preferences, the user's multi-interest value is computed by combining expression intensity and expression duration, which improves the recall of personalized clothing recommendation. For the datasets used in the experiments, our proposed method is superior to other existing methods.

There are still some possible directions to improve the performance of our method. In this study, we only process fewer datasets, and multiclass SVM can get much better results than other algorithms on small sample training sets, so multiclass SVM is used for expression classification. However, for a large-scale dataset, a large amount of storage space is required, which multiclass SVM cannot handle. Most deep learning methods can get good results for expression classification. Besides that, we use rank aggregation for recommendation, and collaborative filtering and knowledge graph could be also used for recommendation. The proposed idea can also be applied to other problems such as personalized news recommendations, personalized travel recommendations, and so on.

Data Availability

Kaggle facial expression recognition challenge database: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61902301, Shaanxi Natural Science Basic Research Project under Grant 2017JQ6058 and 2019JQ-255, the Scientific Research Program funded by Shaanxi Provincial Education Department under Grant 19JK0364 and 18JK0334, and Xi'an Science and Technology Bureau Science and Technology Innovation Leading Project.

References

[1] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification verification," *Advances in Neural Information Processing Systems*, pp. 1988–1996, 2014, http://arxiv.org/abs/1406.4773.

[2] X. Chen, E. Zhou, Y. Mo, J. Liu, and Z. Cao, "Delving deep into coarse-to-fine framework for facial landmark localization," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2088–2095, IEEE, Honolulu, HI, USA, July 2017.

[3] Y. Sun, X. Wang, and X. Tang, "Sparsifying neural network connections for face recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4856–4864, IEEE, Las Vegas, NV, USA, June 2016.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, IEEE, Boston, MA, USA, June 2015.

[5] Z. Zhu, G. Huang, W. Zou, D. Du, and C. Huang, "UCT: learning unified convolutional networks for real time visual tracking," in *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 1973–1982, IEEE, Venice, Italy, October 2017.

[6] J. Zhou, X. Hong, F. Su, and G. Zhao, "Recurrent convolutional neural network regression for continuous pain intensity estimation in video," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 84–92, IEEE, Las Vegas, NV, USA, July 2016.

[7] X. Su and H. Zhou, "Automatic focus personage identification in multilingual news image," in *in Proceedings of the 2017 International Conference on the Frontiers and Advances in Data Science (FADS)*, pp. 1–6, IEEE, Xi'an, China, October 2017.

[8] Y. Tang, "Deep learning using linear support vector machines," 2013, https://arxiv.org/abs/1306.0239.

[9] J. Bergstra and D. D. Cox, "Hyperparameter optimization and boosting for classifying facial expressions: how good can a "null" model be?," 2013, https://arxiv.org/abs/1306.3476.

[10] J. Jeon, J.-C. Park, Y.J. Jo et al., "A real-time facial expression recognizer using deep neural network," in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication—IMCOM '16*, pp. 91–94, Association for Computing Machinery, New York, NY, USA, January 2016.

[11] H. Yang, U. Ciftci, and L. Yin, "Facial expression recognition by de-expression residue learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2168–2177, IEEE, Salt Lake City, UT, USA, June 2018.

[12] F. Zhang, T. Zhang, Q. Mao, and C. Xu, "Joint pose and expression modeling for facial expression recognition," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3359–3368, IEEE, Salt Lake City, UT, USA, June 2018.

[13] C.-M. Kuo, S.-H. Lai, and M. Sarkis, "A compact deep learning model for robust facial expression recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2121–2129, IEEE, Salt Lake City, UT, USA, June 2018.

[14] A. T. Lopes, E. de Aguiar, A. F. de Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: coping with few data and the training sample order," *Pattern Recognition*, vol. 61, pp. 610–628, 2017.

[15] Li Yan, "*Research on methods for facial expression intensity measurement based on video*," pp. 1–57, Hunan University, Changsha, China, 2006, M.S. thesis.

[16] L.-L Wang, H.-Q Dai, and J. Wang, "The research on consumer's clothing preference based on clothing sample," *TBIS*, vol. 10, pp. 528–534, 2017.

[17] J. Hu, Z. Wang, and S. Han, "Research on personalized clothing recommendation mode based on user preference," *Journal of Zhejiang Science-Technology University (Social Sciences)*, vol. 40, no. 2, pp. 136–143, 2018.

[18] E. V. D. Melo, E. A. Nogueira, and D. Guliato, "Content-based filtering enhanced by human visual attention applied to clothing recommendation," in *2015 Proceedings of the IEEE 27th International Conference on Tools with Artificial Intelligence*, pp. 644–651, IEEE, Vietri sul Mare, Italy, November 2015.

[19] X. Su, J. Peng, X. Feng, J. Wu, J. Fan, and L. Cui, "Cross-modality based celebrity face naming for news image collections," *Multimedia Tools and Applications*, vol. 73, no. 3, pp. 1643–1661, 2013.

[20] J. Lu, V. E. Liong, G. Wang, and P. Moulin, "Joint feature learning for face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1371–1383, 2016.

[21] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *Proceedings of the 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, IEEE, Oulu, Finland, December 2017.

[22] O. Melnik, Y. Vardi, and C.-H. Zhang, "A probability model for combining ranks," in *Proceedings of the Multiple Classifier Systems: 6th International Workshop*, pp. 74–85, Seaside, CA, USA, June 2005.

[23] I. J. Goodfellow, D. Erhan, P. L. Carrier et al., "Challenges in representation learning: a report on three machine learning contests," in *Proceedings of the International Conference on Neural Information Processing*, pp. 117–124, Springer, Berlin, Germany, 2013.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 1469090, 11 pages
https://doi.org/10.1155/2020/1469090

Research Article

Upper Bound on the Bit Error Probability of Systematic Binary Linear Codes via Their Weight Spectra

Jia Liu,[1] **Mingyu Zhang**,[1] **Chaoyong Wang**,[1] **Rongjun Chen**,[2] **Xiaofeng An**,[1] **and Yufei Wang**[1]

[1]*School of Information Engineering, Jilin Engineering Normal University, Changchun 130052, China*
[2]*School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*

Correspondence should be addressed to Rongjun Chen; crj321@163.com

Received 18 November 2019; Accepted 11 December 2019; Published 29 January 2020

Guest Editor: jinchang ren

In this paper, upper bound on the probability of maximum *a posteriori* (MAP) decoding error for systematic binary linear codes over additive white Gaussian noise (AWGN) channels is proposed. The proposed bound on the bit error probability is derived with the framework of Gallager's first bounding technique (GFBT), where the Gallager region is defined to be an irregular high-dimensional geometry by using a list decoding algorithm. The proposed bound on the bit error probability requires only the knowledge of weight spectra, which is helpful when the input-output weight enumerating function (IOWEF) is not available. Numerical results show that the proposed bound on the bit error probability matches well with the maximum-likelihood (ML) decoding simulation approach especially in the high signal-to-noise ratio (SNR) region, which is better than the recently proposed Ma bound.

1. Introduction

Upper bounds on the maximum *a posteriori* (MAP) decoding error probability, as a key technique for evaluating the performance of the binary linear codes over additive white Gaussian noise (AWGN) channels, bring a profound impact on the reliable transmission of the next-generation mobile communication systems since they can be used to not only predict the performance without resorting to computer simulations but also guide the design of coding [1]. In order to improve the looseness of the union bound in the low signal-to-noise ratio (SNR) region, many improved upper bounds, on the bit error probability [2–5] and on the frame error probability [2–4, 6–15], are proposed. As surveyed in [1], the improved upper bounds on the bit error probability [2–4] are based on Gallager's first bounding technique (GFBT):

$$\Pr\{E_b\} = \Pr\{E_b, \underline{y} \in \mathscr{R}\} + \Pr\{E_b, \underline{y} \notin \mathscr{R}\}, \quad (1)$$

$$\leq \Pr\{E_b, \underline{y} \in \mathscr{R}\} + \Pr\{\underline{y} \notin \mathscr{R}\}, \quad (2)$$

in which E_b denotes the event that represents an error in one of the information bits of the decoded codeword, $\underline{y}$ denotes the received signal vector, and $\mathscr{R}$ denotes an arbitrary region around the transmitted signal vector (called the Gallager region). Divsalar [2] chose the region $\mathscr{R}$ to be an Euclidean sphere centered at the point along the line connecting the origin to the transmitted signal vector. Sason and Shamai [3] chose the region $\mathscr{R}$ to be a circular cone whose central line passes through the origin and the transmitted signal vector. The upper bounds [2, 3] on the bit error probability based on equation (2) can be considered to be simply replaced by the weight spectra $\{A_d, 1 \leq d \leq n\}$ in the upper bound on the frame error probability by

$$A_d^* \triangleq \sum_{i=0}^{k} \frac{i}{k} A_{i,d}, \quad 1 \leq d \leq n, \quad (3)$$

where $A_{i,d}$ denotes the number of code words of Hamming weight d encoded by information bits of Hamming weight i

and k denotes the dimension of the linear code. However, as noted by Zangl and Herzog [4], computing the expression $\Pr\{\underline{y} \notin \mathscr{R}\}$ by the factor 1.0 in (2) means that the worst case of k bit errors is assumed if $\underline{y}$ falls outside the good region $\mathscr{R}$, and then Zangl and Herzog [4] improved the tangential-sphere bound (TSB) on the bit error probability [3] by computing this probability in a more accurate way. The upper bounds on the bit error probability [2–4] require the whole input-output weight enumerating function (IOWEF), which can be applied to both systematic codes and non-systematic codes. The upper bound on the bit error probability by Ma et al. [16] can be evaluated by calculating partial IOWEF with truncated information weight $(\{A_{i,d}, 0 \le i \le T, 0 \le d \le n-k+T\}$, where $T \ge 0$ is a positive integer), which holds only for systematic codes. However, for most codes, the IOWEF is usually not computable. In contrast, it is reasonable to assume that the weight spectra $\{A_d, 0 \le d \le n\}$ of codes are available, such as the BCH code [17]. In this paper, different from most of the existing bounds, we derive a tighter upper bound on the bit error probability of systematic binary linear codes via their weight spectra.

The main results as well as the structure of this paper are summarized as follows:

(1) In Section 2, we present the preliminaries and necessary notation. The conventional union bound and four reported upper bounds based on GFBT are also reviewed in Section 2.

(2) In Section 3, in a detailed way, we rederive the recently proposed bound on the bit error probability by Liu [5], in which the union bound is firstly truncated and then amended for the systematic linear codes over AWGN channels. In this paper, the proposed upper bound on the bit error probability is derived in a much more detailed way by considering more information of the Gallager region $\mathscr{R}$ and the truncated IOWEF of the code. Finally, with the framework of GFBT, we derive the upper bound on the bit error probability which requires only the knowledge of weight spectra of the code.

(3) In Section 4, numerical examples are given to show that the proposed bound is helpful to evaluate the performance of the systematic binary linear codes which can predict the performance of the code in the high-SNR region, avoiding the time-consuming computer simulations.

(4) Section 5 concludes this paper.

2. Preliminaries

Let $\mathbb{F}_2 = \{0, 1\}$ be the binary field. Let $\mathscr{C}[n, k]$ be a systematic binary linear block code of dimension k and length n with a generator matrix $G = [I_k, P]$, where I_k is the $k \times k$ identity matrix. Let $\underline{u} \in \mathbb{F}_2^k$ be the information vector and $\underline{c} \in \mathbb{F}_2^n$ be the associated codeword. We have the encoding function as follows:

$$\underline{u} \longrightarrow \underline{c} = \underline{u}G. \tag{4}$$

Considering the binary phase shift keying (BPSK) mapping, we have $\underline{c} \longrightarrow \underline{s}$ by $s_t = 1 - 2c_t$ for $0 \le t \le n-1$. Suppose that $\underline{s}$ is transmitted over an AWGN channel. Let $\underline{y} = \underline{s} + \underline{z}$ be the received vector, where $\underline{z}$ is a vector of independent Gaussian random variables with zero mean and variance σ^2. We have the decoding function as follows:

$$\underline{y} \longrightarrow \widehat{\underline{u}}. \tag{5}$$

Without loss of generality, assume that the all-zero codeword $\underline{c}^{(0)}$ is transmitted. The conventional union bound and four reported upper bounds based on GFBT are also reviewed in the following sections.

2.1. Union Bound. The simplest upper bound on the bit error probability is the union bound:

$$\begin{aligned} \Pr\{E_b\} &\le \sum_{i=1}^{k} \frac{i}{k} \sum_{d=1}^{n} \Pr\left\{E_b^{i,d}\right\} \\ &\le \sum_{i=1}^{k} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right), \end{aligned} \tag{6}$$

where $E_b^{i,d}$ is the event that there exists at least one codeword of Hamming weight d encoded by information bits of Hamming weight i that is nearer than $\underline{c}^{(0)}$ to $\underline{y}$, and $Q(\sqrt{d}/\sigma)$ is the pairwise error probability with

$$Q(x) \triangleq \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-(z^2/2)} \mathrm{d}z. \tag{7}$$

However, the above conventional union bound is loose and even diverges (≥ 1) in the low-SNR region. Then, the improved upper bounds on the bit error probability based on GFBT were proposed, such as the Divsalar bound [2], the tangential-sphere bound (TSB) [3], the improved tangential-sphere bound (ITSB) [4], and the Ma bound [16].

2.2. The Divsalar Bound. In 1999, Divsalar derived a simple upper bound [2] on the bit error probability based on GFBT, where the region $\mathscr{R}$ is chosen to be an n-dimensional sphere centered at a scaled transmitted signal vector. Both the radius and the center of the sphere can be optimized. Let $d_{\min}$ denote the minimum Hamming weight. Taking into account the definition of A_d^* in (3), we have the Divsalar bound on the bit error probability:

$$P_b \le \sum_{d=d_{\min}}^{n-k+1} \min\left\{e^{-nE_b(\delta,\beta,\gamma)}, A_d^* Q\left(\sqrt{2d\gamma}\right)\right\}, \tag{8}$$

where

$$E_b(\delta,\beta,\gamma) = -r_b(\delta) + \frac{1}{2}\ln\left(\beta + (1-\beta)e^{2r_b(\delta)}\right) + \frac{\beta\gamma\delta}{1-(1-\beta)\delta},$$

$$\gamma = \frac{1}{2\sigma^2},$$

$$\delta = \frac{d}{n},$$

$$r_b(\delta) = \frac{\ln A_d^*}{n},$$

$$\beta = \sqrt{\frac{\gamma(1-\delta)}{\delta}\frac{2}{1-e^{-2r_b(\delta)}} + \left(\frac{1-\delta}{\delta}\right)^2\left[(1+\gamma)^2 - 1\right]} - \frac{1-\delta}{\delta}(1+\gamma). \tag{9}$$

2.3. The Tangential-Sphere Bound. In 2000, Sason and Shamai [3] derived the tangential-sphere bound on the bit error probability based on GFBT, where the region $\mathscr{R}$ is chosen to be an n-dimensional circular cone whose central line passes through the origin O and the transmitted signal. Let

$$\overline{\gamma}(a,x) = \frac{1}{\Gamma(a)}\int_0^x t^{a-1}e^{-t}\mathrm{d}t, \quad a>0,\ x\geq 0, \tag{10}$$

denote the normalized incomplete gamma function. Taking into account the definition of A_d^* in (3), we have the TSB with a parameter r on the bit error probability:

$$P_b = \int_{-\infty}^{+\infty} P_b(z_1)\frac{1}{\sqrt{2\pi}\sigma}e^{-\left(z_1^2/2\sigma^2\right)}\mathrm{d}z_1, \tag{11}$$

where

$$P_b(z_1) \leq \sum_{d:\,(\delta_d/2)<\alpha_d}\left\{A_d^*\int_{\beta_d(z_1)}^{r_{z_1}}\frac{e^{-\left(z_2^2/2\sigma^2\right)}}{\sqrt{2\pi}\sigma}\overline{\gamma}\left(\frac{n-2}{2},\frac{r_{z_1}^2-z_2^2}{2\sigma^2}\right)\mathrm{d}z_2\right\} + 1 - \overline{\gamma}\left(\frac{n-1}{2},\frac{r_{z_1}^2}{2\sigma^2}\right),$$

$$\delta_d = 2\sqrt{d},$$

$$\alpha_d = r\sqrt{1-\frac{\delta_d^2}{4n}},$$

$$r_{z_1} = r\left(1-\frac{z_1}{\sqrt{n}}\right),$$

$$\beta_d(z_1) = \frac{r_{z_1}\sqrt{d}}{r\sqrt{1-(d/n)}}. \tag{12}$$

The parameter r in the TSB can be optimized by a numerical solution of the following equation:

$$\sum_{d:\ (\delta_d/2)<\alpha_d} A_d^*\int_0^{\theta_d}\sin^{n-3}\phi\,\mathrm{d}\phi = \frac{\sqrt{\pi}\Gamma(n-2/2)}{\Gamma(n-1/2)}, \tag{13}$$

where

$$\theta_d = \arccos\left(\frac{\delta_d}{2r\sqrt{1-\left(\delta_d^2/4n\right)}}\right). \tag{14}$$

2.4. The Improved Tangential-Sphere Bound. In 2001, Zangl and Herzog [4] derived the improved tangential-sphere bound on the bit error probability based on GFBT by computing the expression $\Pr\{\underline{y}\notin\mathscr{R}\}$ in a more accurate way. We have the ITSB with a parameter r_Ψ on the bit error probability:

$$P_b \leq \int_{-\infty}^{+\infty}\left\{\sum_{d:(\delta_d/2)<\alpha_d}\left\{A_d\frac{\overline{g}_d}{k}\overline{\gamma}\left(\frac{n-2}{2},\frac{r_{z_1}^2-\beta_d^2(z_1)}{2\sigma^2}\right)\cdot\left(Q\left(\frac{\beta_d(z_1)}{\sigma}\right)-Q\left(\frac{r_{z_1}}{\sigma}\right)\right)\right\} + \frac{\max\left(\overline{g}_0,\ldots,\overline{g}_{\mathrm{opt}}\right)}{k}\cdot\overline{\gamma}\left(\frac{n-1}{2},\frac{\beta_{d_{\mathrm{opt}}+1}^2(z_1)}{2\sigma^2}\right)\right.$$

$$\left.\cdot\left(\sum_{d=d_{\mathrm{opt}}+1}^{n}\frac{\max\left(\overline{g}_0,\ldots,\overline{g}_{d-1}\right)-\max\left(\overline{g}_0,\ldots,\overline{g}_d\right)}{k}\cdot\overline{\gamma}\left(\frac{n-1}{2},\frac{r_{z_1}^2}{2\sigma^2}\right)\right) + \frac{\max\left(\overline{g}_0,\ldots,\overline{g}_n\right)}{k}\right\}\cdot\frac{1}{\sqrt{2\pi}\sigma}e^{-\left(z_1^2/2\sigma^2\right)}\mathrm{d}z_1. \tag{15}$$

The parameter r_Ψ in the ITSB can be optimized by a numerical solution of the following equation:

$$\sum_{d:\ (\delta_d/2)<\alpha_d} A_d \frac{\overline{g}_d}{k} \int_0^{\theta_d} \sin^{n-3}\phi \, \mathrm{d}\phi = \frac{\max\left(\overline{g}_0, \ldots, \overline{g}_{\mathrm{opt}}\right)}{k} \cdot \frac{\sqrt{\pi}\Gamma(n-2/2)}{\Gamma(n-1/2)}, \tag{16}$$

where

$$\theta_d = \arccos\left(\frac{\delta_d}{2r_\Psi\sqrt{1-\left(\delta_d^2/4n\right)}}\right). \tag{17}$$

2.5. The Ma Bound. In 2018, Ma et al. [16] derived the upper bound on the bit error probability under maximum *a posteriori* (MAP) decoding. The Ma bound can be evaluated by calculating partial IOWEF with truncated information weight. We have the Ma bound with a parameter r^* on the bit error probability

$$\mathrm{BER}_{\mathrm{MAP}} \le \min_{0\le r^*\le T/2}\left\{\sum_{i\le 2r^*}\frac{i}{k}\left(\sum_d A_{i,d}Q\frac{\sqrt{d}}{\sigma}\right) + \sum_{i=r^*+1}^{k}\frac{\min\{i+r^*,k\}}{k}\binom{k}{i}p_b^i(1-p_b)^{k-i}\right\}, \tag{18}$$

where

$$p_b = Q\left(\frac{1}{\sigma}\right). \tag{19}$$

3. Upper Bound on the Bit Error Probability Based on GFBT

3.1. The Gallager Region $\mathscr{R}$. We define the region $\mathscr{R}$ by the Hamming distance based on a list decoding algorithm which is shown in Figure 1, resulting in an irregular high-dimensional geometry (Algorithm 1).

The list decoding algorithm is similar to but different from the algorithm presented in [14]. The *list region* in [14] is an n-dimensional Hamming sphere with center at the hard decision of the whole received sequence, while the list region here is a k-dimensional Hamming sphere with center at the hard decision of the information part of the received sequence.

The Gallager region $\mathscr{R}$ can be defined by

$$\mathscr{R} \triangleq \left\{\underline{y}\ \middle|\ \overset{(0)}{\underline{u}} \in \mathscr{L}_{\underline{y}}\right\} = \left\{\underline{y}\ \middle|\ W_H\left(\widehat{\underline{y}}_0^{k-1}\right) \le r^*\right\}. \tag{20}$$

3.2. Upper Bound on the Bit Error Probability via IOWEF. We assume that $A_{i,d} \ge 1$ and denote all the code words of Hamming weight d encoded by information bits of Hamming weight i by $\underline{c}^{(\ell)}$, $1 \le \ell \le A_{i,d}$. Let $E_{0\longrightarrow\ell}$ be the event that $\underline{c}^{(\ell)}$ is nearer than $\underline{c}^{(0)}$ to $\underline{y}$.

With the framework of GFBT, we have

$$\Pr\{E_b\} = \Pr\left\{E_b, \underline{y} \in \mathscr{R}\right\} + \Pr\left\{E_b, \underline{y} \notin \mathscr{R}\right\}. \tag{21}$$

As shown in Figure 1(b), we have

$$\Pr\left\{E_b, \underline{y} \in \mathscr{R}\right\} \le \sum_{i=1}^{2r^*}\frac{i}{k}\sum_{d=1}^{n}\Pr\left\{E_b^{i,d}, \underline{y} \in \mathscr{R}\right\}, \tag{22}$$

$$\le \sum_{i=1}^{2r^*}\frac{i}{k}\sum_{d=1}^{n}A_{i,d}\Pr\left\{E_{0\longrightarrow 1}, \underline{y} \in \mathscr{R}\right\}. \tag{23}$$

As shown in Figure 1(a), we have

$$\Pr\left\{E_b, \underline{y} \notin \mathscr{R}\right\} \le \sum_{i=r^*+1}^{k}\frac{\min\{i+r^*,k\}}{k}\binom{k}{i}p_b^i(1-p_b)^{k-i}, \tag{24}$$

which means that the decoder outputs at most $i + r^*$ erroneous bits.

Assuming a binary vector of total length N_t passes through a BSC with cross error probability p, we denote $B(p, N_t, N_\ell, N_u)$ to be the probability that the number of bit errors occurring ranges from N_ℓ to N_u, that is,

$$B\left(p, N_t, N_\ell, N_u\right) \triangleq \sum_{m=N_\ell}^{N_u}\binom{N_t}{m}p^m(1-p)^{N_t-m}. \tag{25}$$

Then, we define

$$C(r, p, N_t, N_\ell, N_u) \triangleq \sum_{m=N_\ell}^{N_u}\frac{\min\{m+r, N_t\}}{N_t}\binom{N_t}{m}p^m(1-p)^{N_t-m}. \tag{26}$$

Theorem 1. *We have the upper bound on the bit error probability of systematic binary linear codes under MAP decoding*

$$\Pr\{E_b\} \le \min_{0\le r^*\le k}\left\{\sum_{i=1}^{2r^*}\frac{i}{k}\sum_{d=1}^{n}A_{i,d}Q\left(\frac{\sqrt{d}}{\sigma}\right)B\left(p_b, k-i, 0, \left\lfloor r-\frac{i}{2}\right\rfloor\right) + C\left(r^*, p_b, k, r^*+1, k\right)\right\}. \tag{27}$$

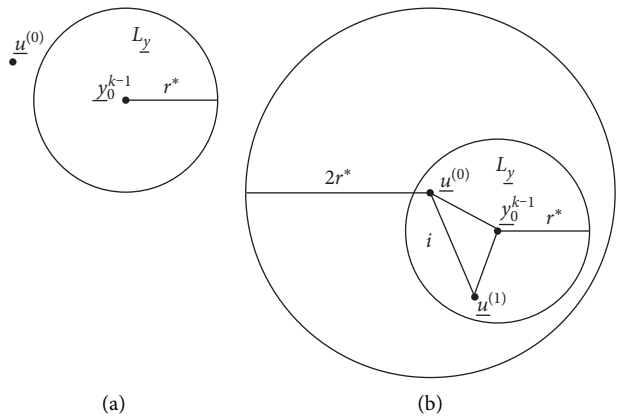


Figure 1: Graphical illustrations of the decoding error events. (a) The error event that the all-zero sequence $\underline{u}^{(0)}$ is not in the list. (b) The error event that the all-zero sequence $\underline{u}^{(0)}$ is in the list but not the closest one.

(1) We denote
$\underline{y} \triangleq (\underline{y}_0^{k-1}\, \underline{y}_k^{n-1})$
in which
$\underline{y}_0^{k-1} = \underbrace{(y_0 \cdots y_{k-1})}_{k}$
and
$\underline{y}_k^{n-1} = \underbrace{(y_k \cdots y_{n-1})}_{n-k}$
Make hard decisions on the information part $\underline{y}_0^{k-1} = \underbrace{(y_0 \cdots y_{k-1})}_{k}$ of the received vector $\underline{y}$, resulting in a vector $\widehat{\underline{y}}_0^{k-1} = \underbrace{(\widehat{y}_0 \cdots \widehat{y}_{k-1})}_{k}$ of length k. Then, the channel becomes a memoryless binary symmetric channel (BSC) with cross probability
$p_b \triangleq Q(1/\sigma)$
(2) List all sequences of length k within the Hamming sphere with center at $\widehat{\underline{y}}_0^{k-1}$ of radius r^*, where r^* is a positive integer. The resulting list is denoted as $\mathscr{L}_{\underline{y}}$.
(3) Encode each sequence in $\mathscr{L}_{\underline{y}}$ by the encoding algorithm of the systematic code, resulting in a list of code words, denoted as $\mathscr{L}_{\underline{c}}$.
(4) Find the codeword $\underline{c}^* \in \mathscr{L}_{\underline{c}}$ that is closest to $\underline{y}$. Output the information part $\widehat{\underline{u}}$ of $\underline{c}^*$ as the decoding result.

Algorithm 1: A list decoding algorithm.

Proof. Without loss of generality, we denote

$$\begin{aligned}\underline{c}^{(1)} &\triangleq \left(\underbrace{1\cdots1}_{i}\underbrace{0\cdots0}_{k-i}\underbrace{1\cdots1}_{d-i}\underbrace{0\cdots0}_{n-k-d+i}\right),\\ \widehat{\underline{y}}_0^{k-1} &\triangleq \left(\underbrace{1\cdots1}_{i_1}\underbrace{0\cdots0}_{i-i_1}\underbrace{1\cdots1}_{i_2}\underbrace{0\cdots0}_{k-i-i_2}\right).\end{aligned} \tag{28}$$

Notice that a necessary condition for the event $E_{0\longrightarrow 1}$ is that the corresponding input information sequence of the codeword $\underline{c}^{(1)}$ is in the list $\mathscr{L}_{\underline{y}}$. Hence,

$$W_H\left(\underline{c}_0^{(1)k-1} - \widehat{\underline{y}}_0^{k-1}\right) \le r^*. \tag{29}$$

We have

$$i - i_1 + i_2 \le r^*. \tag{30}$$

Also notice that, for $\underline{y} \in \mathscr{R}$,

$$W_H\left(\widehat{\underline{y}}_0^{k-1}\right) \le r^*. \tag{31}$$

We have

$$i_1 + i_2 \le r^*. \tag{32}$$

By combining (30) and (32), we can verify that

$$i_2 \le \lfloor r^* - \frac{i}{2} \rfloor. \tag{33}$$

By the union bounds, we have

$$\begin{aligned}&\Pr\left\{E_{0\longrightarrow 1}, \underline{y} \in \mathscr{R}\right\}\\ &= \Pr\left\{E_{0\longrightarrow 1}, W_H\left(\widehat{\underline{y}}_0^{k-1}\right) \le r^*\right\}\\ &\le \sum_{i_2=1}^{r^*} \Pr\left\{E_{0\longrightarrow 1}, W_H\left(\widehat{\underline{y}}_0^{i-1}\right) \le r^* - i_2, W_H\left(\widehat{\underline{y}}_i^{k-1}\right) = i_2\right\}\\ &\le \sum_{i_2=1}^{r^*} \Pr\left\{E_{0\longrightarrow 1}, W_H\left(\widehat{\underline{y}}_i^{k-1}\right) = i_2\right\}\\ &= \sum_{i_2=1}^{\lfloor r^* - (i/2)\rfloor} \Pr\left\{E_{0\longrightarrow 1}, W_H\left(\widehat{\underline{y}}_i^{k-1}\right) = i_2\right\},\end{aligned} \tag{34}$$

for $i_2 \le r^* - (i/2)$ from (33).

Since the event $\Pr\{E_{0\longrightarrow 1}\}$ is independent of $\widehat{\underline{y}}_i^{k-1}$ and $\Pr\{E_{0\longrightarrow 1}\} = Q(\sqrt{d}/\sigma)$, we have

$$\sum_{i_2=1}^{(r^*-i/2)} \Pr\left\{E_{0\longrightarrow 1}, W_H\left(\widehat{y}_i^{k-1}\right) = i_2\right\}, \tag{35}$$

$$= \sum_{i_2=1}^{(r^*-i/2)} \Pr\{E_{0\longrightarrow 1}\}\Pr\left\{W_H\left(\widehat{y}_i^{k-1}\right) = i_2\right\}, \tag{36}$$

$$= \Pr\{E_{0\longrightarrow 1}\} \sum_{i_2=1}^{r^*-(i/2)} \Pr\left\{W_H\left(\widehat{y}_i^{k-1}\right) = i_2\right\}, \tag{37}$$

$$= Q\left(\frac{\sqrt{d}}{\sigma}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2}\rfloor\right). \tag{38}$$

Then, by substituting (38) in (23), we have

$$\Pr\left\{E_b, \underline{y} \in \mathscr{R}\right\} \le \sum_{i=1}^{2r^*} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2}\rfloor\right). \tag{39}$$

Therefore, it can be verified by substituting (24) and (39) in (1) to complete the proof. □

Corollary 1. *The proposed upper bound on the bit error probability (Theorem 1) can improve the conventional union bound on the bit error probability.*

Proof. As to the proposed bound (Theorem 1), note that

$$C\left(r^*, p_b, k, r^* + 1, k\right) = 0, \tag{40}$$

by setting

$$r^* = k. \tag{41}$$

we have

$$\begin{aligned}\Pr\{E_b\} &\le \sum_{i=1}^{k} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2}\rfloor\right) \\ &\le \sum_{i=1}^{k} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right).\end{aligned} \tag{42}$$

Since $B(p_b, k-i, 0, \lfloor r^* - (i/2)\rfloor) \le 1$, the proof is completed. □

Corollary 2. *The proposed upper bound on the bit error probability (Theorem 1) can improve the Ma bound on the bit error probability (18).*

Proof. Assuming that we know only partial IOWEF with truncated information weight $\{A_{i,d}, 0 \le i \le T, 0 \le d \le n-k+T\}$, the parameter r^* in the proposed bound (Theorem 1) is optimized in the interval $[0, \lfloor T/2\rfloor]$. Theorem 1 can be written as

$$\Pr\{E_b\} \le \min_{0\le r^*\le \lfloor T/2\rfloor} \left\{ \sum_{i=1}^{2r^*} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right) B\left(p_b, k-i, 0, \lfloor r - \frac{i}{2}\rfloor\right) + C\left(r^*, p_b, k, r^*+1, k\right) \right\}. \tag{43}$$

Since $B(p_b, k-i, 0, \lfloor r - (i/2)\rfloor)$ is the probability that the number of bit errors occurring in a binary vector of total length $k-i$, when passing through a BSC with cross error probability p_b, it ranges from 0 to $r - (i/2)$. Then, it can be verified by

$$B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2}\rfloor\right) \le 1, \tag{44}$$

to complete the proof. □

The objective of this paper is to derive the upper bound on bit error probability with only knowing of the weight spectrum.

3.3. Upper Bound on the Bit Error Probability via Weight Spectra. In this section, we focus on how to derive the upper bound on the bit error probability via weight spectra. The IOWEF is usually not computable, but the weight spectra $\{A_d, 0 \le d \le n\}$ of the code are usually available, such as the BCH code [17]. Let $T \ge 0$ be a positive integer that is relatively small. Assuming that we know only the truncated IOWEF $\{A_{i,d}, 0 \le i \le T, 0 \le d \le n-k+T\}$ which can be obtained by using a brute-force method and the weight spectrum $\{A_d, 0 \le d \le n\}$.

Define

$$A'_d = A_d - \sum_{i=1}^{T} A_{i,d}, \tag{45}$$

for $0 \le d \le n$.

Then, we focus on how to obtain the upper bound on the bit error probability by using the IOWEF $\{A_{i,d}, 0 \le i \le T, 0 \le d \le n-k+T\}$ and the weight spectrum $\{A_d, 0 \le d \le n\}$. We derive the upper bound in the two following cases.

Case 1: if the radius of the Hamming sphere $r^* \in [0, \lfloor T/2\rfloor]$ in Figure 1, we can get the IOWEF $\{A_{i,d}, 0 \le i \le T, 0 \le d \le n-k+T\}$ by using a brute-force method, and we have

$$\Pr\{E_b\} \le \min_{0\le r^*\le \lfloor T/2\rfloor} \left\{ \sum_{i=1}^{2r^*} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right) B \cdot \left(p_b, k-i, 0, \lfloor r - \frac{i}{2}\rfloor\right) + C\left(r^*, p_b, k, r^*+1, k\right) \right\}, \tag{46}$$

which can be verified by Theorem 1.

Case 2: if the radius of the Hamming sphere $r^* \in [\lfloor T/2\rfloor + 1, k]$ in Figure 1, we can derive the upper bound on the bit error probability by employing both the IOWEF $\{A_{i,d}, 0 \le i \le T, 0 \le d \le n-k+T\}$ and the weight spectrum $\{A_d, 0 \le d \le n\}$.

From Theorem 1, we have

$$\Pr\{E_b\} \le \min_{\lfloor T/2\rfloor+1\le r^*\le k} \left\{ \sum_{i=1}^{2r^*} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\sqrt{\frac{d}{\sigma}}\right) B\left(p_b, k-i, 0, \lfloor r - \frac{i}{2} \rfloor\right) + C\left(r^*, p_b, k, r^*+1, k\right) \right\}, \tag{47}$$

in which

$$\begin{aligned} &\sum_{i=1}^{2r^*} \frac{i}{k} \left(\sum_{d=1}^{n} A_{i,d} Q\left(\sqrt{\frac{d}{\sigma}}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2} \rfloor\right) \right) \\ &= \sum_{i=1}^{T} \sum_{d=1}^{n} \frac{i}{k} A_{i,d} Q\left(\sqrt{\frac{d}{\sigma}}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2} \rfloor\right) \\ &\quad + \sum_{i=T+1}^{2r^*} \sum_{d=1}^{n} \frac{i}{k} A_{i,d} Q\left(\sqrt{\frac{d}{\sigma}}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2} \rfloor\right). \end{aligned} \tag{48}$$

Note that

$$\sum_{i=T+1}^{2r^*} \sum_{d=1}^{n} A_{i,d} = \sum_{i=T+1}^{2r^*} \sum_{d=i}^{2r^*} A_{i,d} + \sum_{i=T+1}^{2r^*} \sum_{d=2r^*+1}^{2r^*+n-k} A_{i,d}. \tag{49}$$

Firstly, it is easy to verify that the first term in the right-hand side (RHS) of (49)

$$\sum_{i=T+1}^{2r^*} \sum_{d=i}^{2r^*} A_{i,d} = \sum_{d=T+1}^{2r^*} \sum_{i=T+1}^{d} A_{i,d}. \tag{50}$$

Since

$$\sum_{i=T+1}^{d} A_{i,d} = A'_d, \tag{51}$$

for $d \in [T+1, 2r^*]$ and i is obviously not greater than $\min\{d, k\}$, we have

$$\sum_{i=T+1}^{d} \frac{i}{k} A_{i,d} \le \frac{\min\{d,k\}}{k} A'_d. \tag{52}$$

Therefore,

$$\begin{aligned} &\sum_{i=T+1}^{2r^*} \sum_{d=i}^{2r^*} \frac{i}{k} A_{i,d} Q\left(\sqrt{\frac{d}{\sigma}}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2} \rfloor\right) \\ &\le \sum_{d=T+1}^{2r^*} \frac{\min\{d,k\}}{k} A'_d Q\left(\sqrt{\frac{d}{\sigma}}\right), \end{aligned} \tag{53}$$

for

$$B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2} \rfloor\right) \le 1. \tag{54}$$

Secondly, it is easy to verify that the second term in the RHS of (49)

$$\sum_{i=T+1}^{2r^*} \sum_{d=2r^*+1}^{2r^*+N-K} A_{i,d} \le \sum_{d=2r^*+1}^{2r^*+N-K} A'_d. \tag{55}$$

Since

$$\sum_{i=T+1}^{2r^*} A_{i,d} \le A'_d, \tag{56}$$

for $d \in [T+1, 2r^*+n-k]$ and i is obviously not greater than $\min\{2r^*, k\}$, we have

$$\sum_{i=T+1}^{2r^*} \frac{i}{k} A_{i,d} \le \frac{\min\{2r^*,k\}}{k} A'_d. \tag{57}$$

Therefore,

$$\begin{aligned} &\sum_{i=T+1}^{2r^*} \sum_{d=2r^*+1}^{2r^*+n-k} \frac{i}{k} A_{i,d} Q\left(\sqrt{\frac{d}{\sigma}}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2} \rfloor\right) \\ &\le \sum_{d=2r^*+1}^{2r^*+n-k} \frac{\min\{2r^*,k\}}{k} A'_d Q\left(\sqrt{\frac{d}{\sigma}}\right), \end{aligned} \tag{58}$$

for $B(p_b, k-i, 0, \lfloor r^* - (i/2) \rfloor) \le 1$.

Then, we have

$$\begin{aligned} &\sum_{i=T+1}^{2r^*} \sum_{d=1}^{n} \frac{i}{k} A_{i,d} Q\left(\sqrt{\frac{d}{\sigma}}\right) B\left(p_b, k-i, 0, \lfloor r^* - \frac{i}{2} \rfloor\right) \\ &\le \sum_{d=T+1}^{2r^*} \frac{\min\{d,k\}}{k} A'_d Q\left(\sqrt{\frac{d}{\sigma}}\right) + \sum_{d=2r^*+1}^{2r^*+n-k} \frac{\min\{2r^*,k\}}{k} A'_d Q\left(\sqrt{\frac{d}{\sigma}}\right), \end{aligned} \tag{59}$$

by combining (53) and (58) with (59).

Finally, we have

$$\begin{aligned} \Pr\{E_b\} \le \min_{\lfloor T/2\rfloor+1\le r^*\le k} \Bigg\{ &\sum_{i=1}^{T} \frac{i}{k} \sum_{d=1}^{n} h(i,d,r^*) \\ &+ \sum_{d=T+1}^{2r^*} \frac{\min\{d,k\}}{k} A'_d Q\left(\sqrt{\frac{d}{\sigma}}\right) \\ &+ \sum_{d=2r^*+1}^{2r^*+n-k} \frac{\min\{2r^*,k\}}{k} A'_d Q\left(\sqrt{\frac{d}{\sigma}}\right) \\ &+ C\left(r^*, p_b, k, r^*+1, k\right) \Bigg\}, \end{aligned} \tag{60}$$

by combining (48) and (59) with (47).

Then, we have the following theorem.

Theorem 2. *We have the upper bound on the bit error probability of systematic binary linear codes via their weight spectra*

$$\Pr\{E_b\} \le \min\left\{\Pr\left\{E_{b_1}\right\}, \Pr\left\{E_{b_2}\right\}\right\}, \tag{61}$$

where

$$\Pr\{E_{b_1}\} = \min_{0\le r^*\le \lfloor T/2\rfloor} \left\{ \sum_{i=1}^{2r^*} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right) B\left(p_b, k-i, 0, \lfloor r - \frac{i}{2} \rfloor\right) + C(r^*, p_b, k, r^*+1, k) \right\},$$

$$\Pr\{E_{b_2}\} = \min_{\lfloor T/2\rfloor+1\le r^*\le k} \left\{ \sum_{i=1}^{T} \frac{i}{k} \sum_{d=1}^{n} h(i,d,r^*) + \sum_{d=T+1}^{2r^*} \frac{\min\{d,k\}}{k} A'_d Q\left(\frac{\sqrt{d}}{\sigma}\right) + \sum_{d=2r^*+1}^{2r^*+n-k} \frac{\min\{2r^*,k\}}{k} A'_d Q\left(\frac{\sqrt{d}}{\sigma}\right) + C(r^*, p_b, k, r^*+1, k) \right\}. \tag{62}$$

Proof. We can complete the theorem by combining (46) and (60). □

Corollary 3. *The proposed upper bound on the bit error probability (Theorem 2) can improve the proposed upper bound on the bit error probability (Theorem 1).*

Assuming that we know only the truncated IOWEF $\{A_{i,d}, 0 \le i \le T,\ 0 \le d \le n-k+T\}$ and the proof weight spectrum $\{A_d, 0 \le d \le n\}$, Theorem 1 can be written as

$$\Pr\{E_b\} \le \min_{0\le r^*\le \lfloor T/2\rfloor} \left\{ \sum_{i=1}^{2r^*} \frac{i}{k} \sum_{d=1}^{n} A_{i,d} Q\left(\frac{\sqrt{d}}{\sigma}\right) B\left(p_b, k-i, 0, \lfloor r - \frac{i}{2} \rfloor\right) + C(r^*, p_b, k, r^*+1, k) \right\}, \tag{63}$$

implying that $\Pr\{E_b\} \le \Pr\{E_{b_1}\}$. Theorem 2 needs a minimization over $0 \le r^* \le k$ if the optimal parameter r^* is in the interval $[0, \lfloor T/2 \rfloor]$, and Theorem 2 is exactly Theorem 1; if r^* is in the interval $[\lfloor T/2 \rfloor + 1, k]$, Theorem 2 is tighter than Theorem 1. Therefore, we claim that Theorem 2 can improve Theorem 1 to complete the proof.

Corollary 4. *The proposed upper bound on the bit error probability (Theorem 2) can improve the Ma bound on the bit error probability (18).*

Proof. It can be verified by combining Corollaries 2 and 3. □

Remark. The proposed bound (Theorem 2) has a little higher computational loads than the conventional union bound. Firstly, the overhead is caused by recursively computing $B(\cdot,\cdot,\cdot,\cdot)$ and $C(\cdot,\cdot,\cdot,\cdot,\cdot)$. The probability $B(\cdot,\cdot,\cdot,\cdot)$ and $C(\cdot,\cdot,\cdot,\cdot,\cdot)$ is the summation with at most k summands, which are independent of the IOWEF and hence can be calculated and stored for use. Secondly, the overhead is caused by minimizing over r^* $(0 \le r^* \le k)$. A brute-force method can be implemented by computing the bound for each r^*, which can be done recursively.

4. Numerical Examples

In this section, we need to point out that the weight spectra of the compared BCH codes can be found in [17]. For all the upper bounds on the bit error probability except the Ma bound, we need the whole IOWEF. Then, in this paper, the compared bounds are the Ma bound (18) and the proposed bound (61) in Theorem 2 on the bit error probability, which are also compared with the simulation results under the maximum-likelihood (ML) decoding.

Figures 2 and 3 show the comparisons between the upper bounds of BCH codes [127, 106] and [127, 113], respectively, which are also compared with the simulation results under ML decoding. A partial IOWEF $\{A_{i,d},\ 0 \le i \le T,\ 0 \le d \le n-k+T\}$ with $T = 8$ of the BCH codes [127, 106] and [127, 113] can be obtained by using a brute-force method, respectively. The computed IOWEF $\{A_{i,d}, 0 \le i \le 8, 0 \le d \le 29\}$ of the BCH code [127, 106] is given in Table 1. The Ma bound is obtained by this truncated IOWEF according to (18). The proposed upper bound is obtained by this truncated IOWEF and the weight spectrum according to (61) in Theorem 2 (note that (61) is different from [5], (23)) since Theorem 2 here is derived in a much more detailed way when the Hamming weight $d \in (T, 2r^*]$). As pointed out in [18], multidimensional signal processing plays a very important role in effective data analytics and interpretation. In this paper, we tighten the upper bound by analysing the k-dimensional vector. We can see that the proposed bound is tighter than the Ma bound. We can also see that, for the same code length n, the higher the code rate is, the tighter the Ma bound is. The proposed bound is always tighter whatever the code rate is.

Figure 4 shows the comparisons between the upper bounds on the bit error probability of the BCH code [255, 239], which are also compared with the simulation results under ML decoding. A partial IOWEF $\{A_{i,d},\ 0 \le i \le T,\ 0 \le d \le n-k+T\}$ with $T = 5$ of the BCH code [255, 239] can be obtained by using a brute-force method. The Ma bound is obtained by this truncated IOWEF according to (18). The proposed upper bound is obtained by this truncated IOWEF and the weight spectrum according to (61). We can see that the proposed bound is tighter than the Ma bound. We can also see that, the proposed bound coincides nicely with the ML decoding results in the high-SNR region when we only

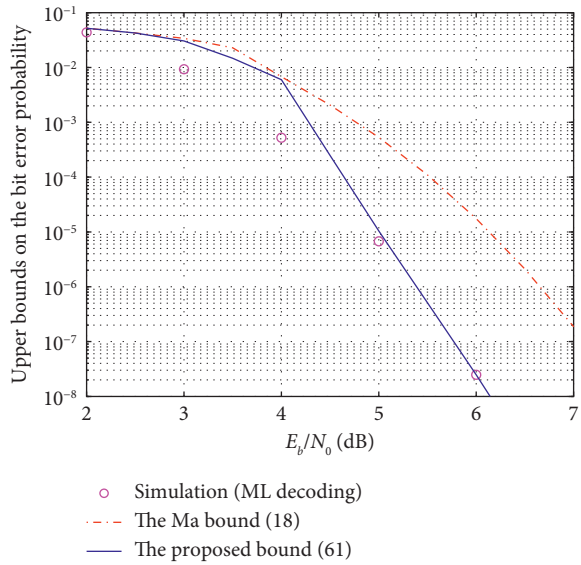


Figure 2: comparison between the proposed bound (Theorem 2) and the Ma bound on the bit error probability of the BCH code [127, 106], which is also compared with the simulation results under ML decoding.

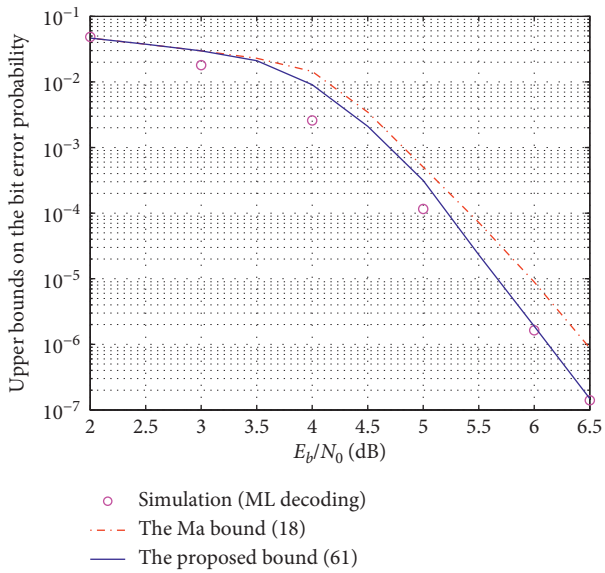


Figure 3: comparison between the proposed bound (Theorem 2) and the Ma bound on the bit error probability of the BCH code [127, 113], which is also compared with the simulation results under ML decoding.

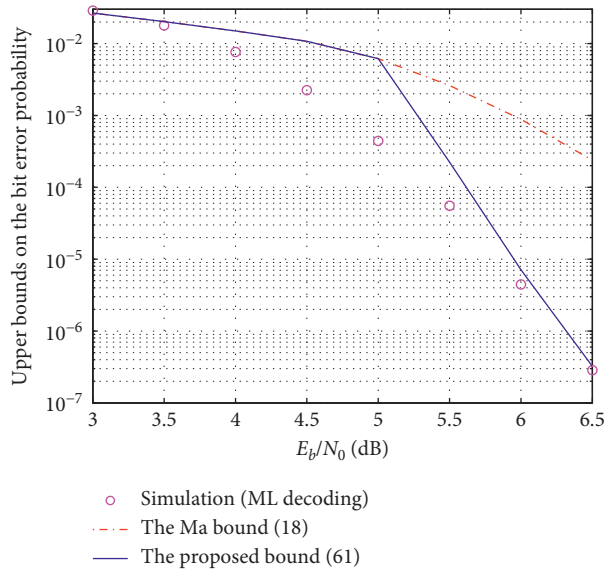


Figure 4: comparison between the proposed bound (Theorem 2) and the Ma bound on the bit error probability of the BCH code [255, 239], which is also compared with the simulation results under ML decoding.

know less IOWEF $\{A_{i,d}, 0 \le i \le 5, 0 \le d \le 21\}$ of the BCH code [255, 239].

Figure 5 shows the comparisons between the upper bounds on the bit error probability of the BCH code [31, 21], which are also compared with the simulation results under ML decoding. The whole IOWEF $A_{i,d}$, $0 \le i \le 21$, $0 \le d \le 31\}$ of the BCH code [31, 21] can be obtained by using a brute-force method. The Ma bound is obtained by the whole IOWEF according to (18), where $T = 21$. The proposed upper bound is obtained by the whole IOWEF and the weight spectrum according to (61). We can see that the proposed bound is tighter than the Ma bound in the low-SNR region when we know the whole IOWEF. We can also see that, the proposed bound and the Ma bound coincide nicely with the ML decoding results in the high-SNR region.

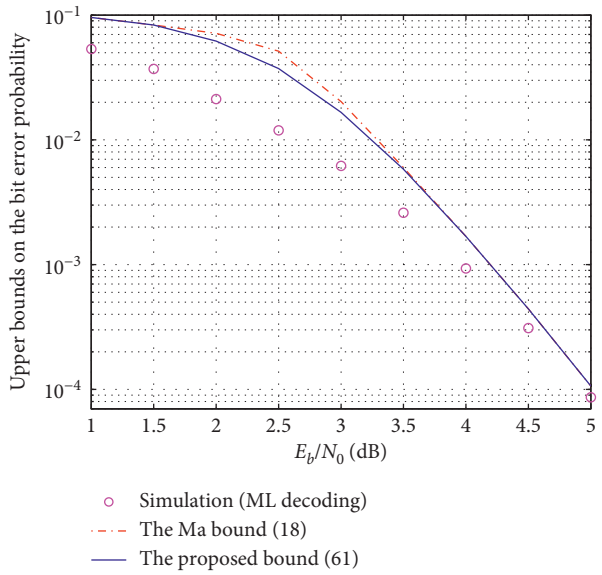


Figure 5: comparison between the proposed bound (Theorem 2) and the Ma bound on the bit error probability of the BCH code [31, 21], which is also compared with the simulation results under ML decoding.

Table 1: The partial IOWEF $\{A_{i,d},\ 0 \le i \le 8,\ 0 \le d \le 29\}$ with $T = 8$ of the BCH code [127, 106].

1	7	17	3	19	1847	5	20	2627123	7	15	$2.364896e+009$
1	8	8	3	20	584	5	21	984117	7	16	$3.415877e+009$
1	9	2	3	21	127	5	22	288687	7	17	$4.098554e+009$
1	10	4	3	22	21	5	23	64158	7	18	$4.098554e+009$
1	11	23	4	7	3470	5	24	10105	7	19	$3.415535e+009$
1	12	27	4	8	15952	5	25	1049	7	20	$2.364660e+009$
1	13	13	4	9	46995	5	26	59	7	21	1351255504
1	14	3	4	10	125279	6	7	19462	7	22	630548101
1	15	8	4	11	275791	6	8	194386	7	23	236462455
1	16	1	4	12	484126	6	9	1059704	7	24	69546072
2	7	71	4	13	697948	6	10	4752113	7	25	15457425
2	8	188	4	14	834510	6	11	16623036	7	26	2439979
2	9	305	4	15	834260	6	12	44331232	7	27	243848
2	10	472	4	16	696623	6	13	94527372	7	28	11614
2	11	711	4	17	482965	6	14	165420289	8	8	163532
2	12	1007	4	18	275130	6	15	239125721	8	9	2946230
2	13	964	4	19	128772	6	16	286915986	8	10	29469636
2	14	794	4	20	47734	6	17	286915986	8	11	192045523
2	15	511	4	21	14385	6	18	239077900	8	12	864339336
2	16	340	4	22	3172	6	19	165519571	8	13	$2.924072e+009$
2	17	135	4	23	513	6	20	94593564	8	14	$7.797383e+009$
2	18	48	4	24	37	6	21	44141017	8	15	$1.672354e+010$
2	19	12	4	25	2	6	22	16553366	8	16	$2.926719e+010$
2	20	7	5	7	11611	6	23	4864138	8	17	$4.226740e+010$
3	7	614	5	8	73116	6	24	1082409	8	18	$5.072263e+010$
3	8	2097	5	9	281110	6	25	170775	8	19	$5.072258e+010$
3	9	4816	5	10	960971	6	26	17090	8	20	$4.226969e+010$
3	10	10482	5	11	2633699	6	27	823	8	21	$2.926271e+010$
3	11	18637	5	12	5642851	7	7	13142	8	22	$1.672131e+010$
3	12	27236	5	13	9825582	7	8	276537	8	23	$7.803514e+009$
3	13	32706	5	14	14197834	7	9	2380926	8	24	$2.926377e+009$
3	14	32212	5	15	17043850	7	10	15092372	8	25	860707256
3	15	27027	5	16	17047259	7	11	69866721	8	26	191259366
3	16	18689	5	17	14202094	7	12	237457686	8	27	30195788
3	17	10826	5	18	9829514	7	13	630106871	8	28	3021987
3	18	5000	5	19	5614707	7	14	1350174932	8	29	143670

5. Conclusions

In this paper, upper bound on the bit error probability of systematic binary linear codes under MAP decoding is derived. The proposed bound just requires the weight spectra of the code, which is helpful when the whole IOWEF of the code is not available. The proposed bound (Theorem 2) is proved to be tighter than the recently proposed Ma bound. The numerical results show that the proposed bound on the bit error probability via weight spectra coincides nicely with the ML decoding results in the high-SNR region, which can predict the BER performance without resorting to computer simulations since the simulation is time-consuming in high-SNR region.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Scientific Research Planning Project of Education Department of Jilin Province (JJKH20200180KJ), the Doctoral Initiated Research Foundation Project (BSKJ201820, BSKJ201821, and BSKJ201822), the Science and Technology Planning Project of Jilin Province in 2020 named Research on Key Technologies of Wireless Wearable Health Monitoring Equipment, the Distinctive Innovation of Ordinary Universities of Guangdong Province (2018KTSCX120), and the PhD Start-up Fund of Natural Science Foundation of Guangdong Province (2016A030310335).

References

[1] I. Sason and S. Shamai, "Performance analysis of linear codes under maximum-likelihood decoding: a tutorial," in *Foundations and Trends in Communications and Information Theory*, vol. 3, pp. 1–225, , no. 1-2, NOW, Delft, Netherlands, 2006.

[2] D. Divsalar, "A simple tight bound on error probability of block codes with application to turbo codes," TMO Progress Report 42–139, pp. 1–35, JPL, Pasadena, CA, USA, 1999.

[3] I. Sason and S. Shamai, "Improved upper bounds on the ML decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 24–47, 2000.

[4] J. Zangl and R. Herzog, "Improved tangential sphere bound on the bit error probability of concatenated codes," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 825–830, 2001.

[5] J. Liu, "Performance analysis of systematic linear codes over AWGN channels," in *Proceedings of the 2016 IEEE International Conference on RFID Technology and Applications*, pp. 133–137, Foshan, China, September 2016.

[6] E. R. Berlekamp, "The technology of error-correcting codes," *Proceedings of the IEEE*, vol. 68, no. 5, pp. 564–593, 1980.

[7] T. Kasami, T. Fujiwara, T. Takata, K. Tomita, and S. Lin, "Evaluation of the block error probability of block modulation codes by the maximum-likelihood decoding for an AWGN channel," in *Proceedings of the 15th Symposium on Information Theory and its Applications*, Minakami, Japan, September 1992.

[8] T. Kasami, T. Fujiwara, T. Takata, and S. Lin, "Evaluation of the block error probability of block modulation codes by the maximum-likelihood decoding for an AWGN channel," in *Proceedings of the 1993 IEEE International Symposium on Information Theory*, p. 68, San Antonio, TX, USA, January 1993.

[9] H. Herzberg and G. Poltyrev, "Techniques of bounding the probability of decoding error for block coded modulation structures," *IEEE Transactions on Information Theory*, vol. 40, no. 3, pp. 903–911, 1994.

[10] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1284–1292, 1994.

[11] D. Divsalar and E. Biglieri, "Upper bounds to error probabilities of coded systems beyond the cutoff rate," *IEEE Transactions on Communications*, vol. 51, no. 12, pp. 2011–2018, 2003.

[12] S. Yousefi and A. K. Khandani, "Generalized tangential sphere bound on the ML decoding error probability of linear binary block codes in AWGN interference," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2810–2815, 2004.

[13] A. Mehrabian and S. Yousefi, "Improved tangential sphere bound on the ML decoding error probability of linear binary block codes in AWGN and block fading channels," *IEE Proceedings-Communications*, vol. 153, no. 6, pp. 885–893, 2006.

[14] X. Ma, J. Liu, and B. Bai, "New techniques for upper-bounding the ML decoding performance of binary linear codes," *IEEE Transactions on Communications*, vol. 61, no. 3, pp. 842–851, 2013.

[15] J. Liu, "The parameterized Gallager's first bounds based on conditional triplet-wise error probability," *Mathematics and Computers in Simulation*, vol. 163, pp. 32–46, 2019.

[16] X. Ma, K. Huang, and B. Bai, "Systematic block Markov superposition transmission of repetition codes,," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1604–1620, 2018.

[17] The OEIS Foundation Inc., *List of Weight Distributions*, The OEIS Foundation Inc., NJ, USA, 2013, http://oeis.org/wiki/List_of_weigth_distributions.

[18] J. Ren, J. Han, and M. Dalla Mura, "Special issue on multimodal data fusion for multidimensional signal processing," *Multidimensional Systems and Signal Processing*, vol. 27, no. 4, pp. 801–805, 2016.

Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 4706576, 11 pages
https://doi.org/10.1155/2020/4706576

Research Article

A Full Stage Data Augmentation Method in Deep Convolutional Neural Network for Natural Image Classification

Qinghe Zheng,[1] **Mingqiang Yang**,[1] **Xinyu Tian**,[2] **Nan Jiang**,[3] **and Deqiang Wang**[1]

[1]*School of Information Science and Engineering, Shandong University, Qingdao 266237, China*
[2]*College of Mechanical and Electrical Engineering, Shandong Management University, Jinan 250357, China*
[3]*School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China*

Correspondence should be addressed to Mingqiang Yang; imageinstitute@outlook.com and Deqiang Wang; wdq_sdu@sdu.edu.cn

Received 2 November 2019; Accepted 16 December 2019; Published 11 January 2020

Guest Editor: Zheng Wang

Nowadays, deep learning has achieved remarkable results in many computer vision related tasks, among which the support of big data is essential. In this paper, we propose a full stage data augmentation framework to improve the accuracy of deep convolutional neural networks, which can also play the role of implicit model ensemble without introducing additional model training costs. Simultaneous data augmentation during training and testing stages can ensure network optimization and enhance its generalization ability. Augmentation in two stages needs to be consistent to ensure the accurate transfer of specific domain information. Furthermore, this framework is universal for any network architecture and data augmentation strategy and therefore can be applied to a variety of deep learning based tasks. Finally, experimental results about image classification on the coarse-grained dataset CIFAR-10 (93.41%) and fine-grained dataset CIFAR-100 (70.22%) demonstrate the effectiveness of the framework by comparing with state-of-the-art results.

1. Introduction

Computer vision is the first and most widely used field of deep learning technology. After the advent of AlexNet [1], deep convolutional neural networks (CNNs) have been quickly applied for various tasks in computer vision, including pedestrian detection [2], face recognition [3], image classification [4–6], semantic segmentation [7, 8], and target tracking [9, 10]. Due to the availability of big data and massive computing resources, overparameterized deep learning models have demonstrated their superior performance depending on the highly nonlinear fitting capabilities. So far, many kinds of deep learning models have been developed and improved, including different structures and connections [11]. The corresponding training methods are also constantly updated [12, 13].

However, deep learning still has many unintelligible properties and the theory behind it is not perfect. Typically, due to its difficulty of interpretation, deep learning models are difficult to be improved in a targeted manner. Researchers usually need to consider both optimization and generalization. Moreover, big data driven mode based deep CNNs still have the "overfitting" problem; that is, the neural network can perform well on the training set but cannot be effectively generalized on the unseen test data. On the other hand, a larger model tends to perform better [14], but it also requires people to make tradeoffs between accuracy and reasoning speed in practice. The noise in the natural image also affects the mining of implicit knowledge and the extraction of expressive features of the object. These challenges have hindered its successful application in some special scenarios, such as the medical diagnosis tasks [15], where there is lack of training data and automatic driving systems [16] that require high real-time performance.

At present, many methods have been developed to alleviate the "overfitting" problem of deep CNNs, and they can be summarized as follows:

(i) Regularization techniques for limiting network complexity, such as L2-regularization [17] and Hierarchical Guidance and Regularization (HGR) learning [18]

(ii) Data augmentation methods for expanding sample set, such as translation [19], horizontal flipping [20], and noise disturbance [21]

(iii) Model ensemble for reducing dependence on single network, for example, auxiliary classification nodes in GoogleLeNet [22], Dropout [23], and DropConnect [24]

(iv) Some special training tricks like well-designed initialization [25], early stopping [26], and learning rate decay [27]

In this paper, we propose the full stage (i.e., training and testing stages) data augmentation framework in deep learning for natural image classification. Data augmentation in the training process is used to ensure that the network can mine the structural information of samples and finally converge in the appropriate position, and the data augmentation in the test process can play the role of model ensemble to reduce the dependence on a single network. Augmentation in two stages needs to be consistent to ensure accurate transfer of domain information. It is worth noting that the framework is universal to any network architecture and data augmentation strategy and can therefore be applied to a variety of deep learning based tasks. We have done extensive experiments on fine-grained and coarse-grained image classification datasets, that is, CIFAR-10 and CIFAR-100 [28]. Compared with different algorithms, our framework shows significant improvement on deep CNN and achieves state-of-the-art results.

The remainder of the paper is organized as follows. Section 2 gives a brief review of the related work on data augmentation in deep learning. In Section 3, we introduce the proposed full stage data augmentation framework in detail. Experimental results and comparisons are presented in Section 4. Finally, we conclude our work and discuss future directions in Section 5.

2. Related Work

Data augmentation is an effective method to reduce the "overfitting" of deep CNN caused by limited training samples, which approximates the data probability space by manipulating input samples, such as horizontal flipping, random crop, scale transformation, and noise disturbance. In general, as long as the quantity, quality, and diversity of the data in the dataset are increased, the effectiveness of the model can be improved. Sample pairing [29] is a simple but surprisingly effective data augmentation technique for image classification task, which can create the new image from an original one by overlaying another image randomly picked from the training set. However, many special training tricks hinder its real application. Neural Augmentation [30] and Smart Augmentation methods [31] teach the neural network autonomous learning how to generate new samples by minimizing the error of that network. The appearance of Generative Adversarial Networks (GANs) provides a new research direction for data augmentation. Frid-Adar et al. [32] have illustrated that training with adversarial samples generated by GANs can improve the generalization ability of deep CNNs and help to overcome the defects of activation functions. But, in practice, GANs require considerable time for training and are difficult to converge. As for data augmentation in testing phase, Wang et al. [33] have used different underpinning network structures and augmented the image by 3D rotation, flipping, scaling, and adding random noise. Experiments showed that test-time augmentation can achieve higher segmentation accuracy and obtain uncertainty estimation of the segmentation results. There have been many data augmentation methods in deep learning community, but how to efficiently apply them is currently the most important research direction.

In addition, there are many regularization methods at the loss layer which can also be interpreted as an implicit data augmentation, such as Dropout [23], DropConnect [24], DisturbLabel [34], and SoftLabel [35]. Dropout and DropConnect can be interpreted as data augmentation methods by projecting the introduced noises back into the input space. DisturbLabel and SoftLabel add specially distributed noises to ground-truth category labels of randomly selected samples during the training process. The noises have been distributed in the implicit augmented samples. Although the above methods can improve the generalization ability of the model, the impact of additional noise on the decision boundary has not been analyzed rigorously.

In fact, approximating real and natural input spaces through data augmentation is intuitionistic. A more comprehensive input space allows the model to better converge on a global minimum or a better local minimum. However, the "overfitting" problem of deep CNNs still exists, which prompts us to rethink of the influence of data augmentations during training and testing process on the optimization and generalization of deep CNNs.

3. Full Stage Data Augmentation Framework

3.1. Problem Formulation. Given a deep CNN model $\mathbb{M}_0$: $f(\mathbf{x}; \boldsymbol{\theta}_0)$ trained on the training set D: $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, $(\mathbf{x}, \mathbf{y})$ and $\boldsymbol{\theta}_0$: $\{\mathbf{W}_0^l, \mathbf{b}_0^l\}_{l=1}^{L}$ represent the inputs (i.e., the natural images and corresponding ground-truth labels) and initialized network parameters, respectively. Parameters are organized into four-dimensional tensors and two-dimensional matrices in the convolutional and fully connected layers, respectively. The network is optimized by mini-batch stochastic gradient descent (SGD) method based on back propagation.

In the forward propagation stage, the output of each layer is the input of the next; the output $\mathbf{h}_l$ of l-th layer in deep CNN for $l = 1, \ldots, L-1$ can be given by

$$\mathbf{h}_l = \sigma\left(\mathbf{W}_0^l \mathbf{h}_{l-1} + \mathbf{b}_0^l\right), \tag{1}$$

where $\mathbf{h}_0 = \mathbf{x}$ and $\sigma(\cdot)$ represents the element-wise nonlinear activation function, such as Leaky-ReLU [36], which is defined as

$$\sigma(x) = \begin{cases} x, & \text{if } x > 0, \\ \dfrac{x}{a}, & \text{if } x \le 0, \end{cases} \tag{2}$$

where a is a fixed hyperparameter in $(1, +\infty)$. Then the final output of deep CNN model can be obtained by

$$f(\mathbf{x}) = \text{softmax}\left(\mathbf{W}_0^L \mathbf{h}_{L-1} + \mathbf{b}_0^L\right), \tag{3}$$

where softmax $(\cdot)$ is defined as the logarithmic normalization function of finite term discrete probability distribution and can be calculated according to

$$\text{softmax}(f)_i = \frac{e^{f_i}}{\sum_{j=1}^{C} e^{f_j}}, \quad \text{for } i = 1, 2, \ldots, C, \tag{4}$$

where C is the number of neurons in the last layer, that is, the number of classification categories. Finally, the training loss of deep CNN can be given by

$$\mathscr{L}(\mathbf{x}_i, \mathbf{y}_i) = -\frac{1}{C} \sum_{j=1}^{C} \left[y_i^j \log f(\mathbf{x}_i)^j + \left(1 - y_i^j\right) \log\left(1 - f(\mathbf{x}_i)^j\right) \right] + \lambda \sum_{k=1}^{L} \left\| \mathbf{W}^k \right\|_F. \tag{5}$$

The first term is the negative log-likelihood loss and the second term is L2-regularization of all the weights. λ is the weight decay rate that controls the regularization intensity and $\|\cdot\|_F$ represents the Frobenius norm. By continuously optimizing the loss function and updating the network parameters, the model is trained for convergence and used for testing.

In the back propagation stage, our goal is to minimize $\mathscr{L}$ through updating parameters (weights $\mathbf{W}$ and biases $\mathbf{b}$) in deep CNN. Based on mini-batch SGD, parameters at t-th training iteration can be updated as

$$\mathbf{W}_t^l = \mathbf{W}_{t-1}^l - \alpha \cdot \frac{1}{M} \sum_{i=1}^{M} \frac{\partial \mathscr{L}(\mathbf{x}_i, \mathbf{y}_i)}{\partial \mathbf{W}_{t-1}^l},$$
$$\mathbf{b}_t^l = \mathbf{b}_{t-1}^l - \alpha \cdot \frac{1}{M} \sum_{i=1}^{M} \frac{\partial \mathscr{L}(\mathbf{x}_i, \mathbf{y}_i)}{\partial \mathbf{b}_{t-1}^l}, \tag{6}$$

where α and M represent the learning rate and batch size, respectively. Through continuous iteration (each of which includes M forward propagation steps and 1 back propagation step), a convergent model $\mathbb{M}_*$: $f(\mathbf{x}; \boldsymbol{\theta}_*)$ is obtained. In the test process, the convergent deep CNN model is used to output the category labels of test samples. Finally, the entire flow chart is drawn in Figure 1.

3.2. Data Augmentation during Training Process. From the perspective of image acquisition, an acquired image is only one of many possible observations of the potential anatomy that can be observed by different spatial transformations and noise disturbance. Direct inference of the acquired images may result in biased results affected by specific transformations and noise associated with equipment and environment. In order to obtain a more reliable and robust prediction, we propose a full stage data augmentation framework to decrease the "overfitting" problem in deep CNN.

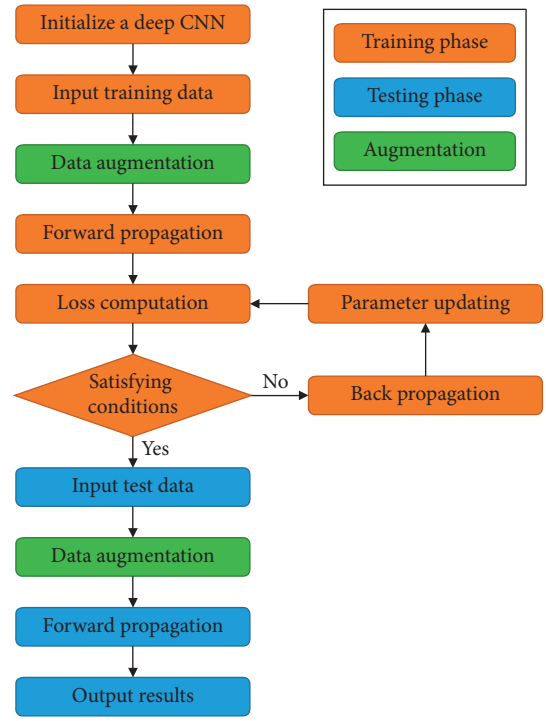


Figure 1: The overall flow of training and testing of deep CNNs.

At the first level, that is, training stage, the training samples D_t: $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{M}$ in a mini-batch set at t-th training iteration can be expanded to $\tilde{D}_t$: $\{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^{\tilde{M}}$ through various data augmentation techniques when they are fed into the deep CNN, such as translation [19], horizontal flipping [20], and noise disturbance [21]. All the augmentation parameters like translation step, rotation range, and noise intensity are set and retained. Furthermore, it is worth noting that all data augmentations are performed at the data input stage rather than at the beginning of the entire training process. In this way, the training data are expanded to $\tilde{M}/M$ times of the original data and the number of training iterations remains almost unchanged.

3.3. Data Augmentation during Testing Process. At the second level, that is, test stage, we use the same distributions of augmentation parameters for the convergent deep CNN. Each test image is augmented to $\tilde{M}/M$ images through the same data augmentations used in the training process. The consistency of data augmentation in the two stages is helpful to ensure the accurate transfer of domain information. The $\tilde{M}/M$ prediction results are combined to obtain the final prediction based on majority voting:

$$f(\mathbf{x}) = \frac{M}{\widetilde{M}} \sum_{i=1}^{\widetilde{M}/M} f(\widetilde{\mathbf{x}}_i). \tag{7}$$

Then the label corresponding to the location of the largest value in the one-dimensional vector $f(\mathbf{x})$ is the final prediction result. If there exists a balanced vote, the category with largest probability is chosen as the final prediction result. The whole framework is shown in Figure 2.

3.4. Interpretation as Model Ensemble. Researchers [37] have reported that the combination of deep CNNs trained on different noisy datasets is usually helpful. However, training each neural network separately is prohibitively expensive, since this requires exponentially many large sets containing noisy data. At test stage, data augmentation operation of each test sample ($\mathbf{x} \longrightarrow \widetilde{\mathbf{x}}$) can be viewed as $\widetilde{\mathbf{x}} = g(\mathbf{x})$ where $g(\cdot)$ represents corresponding data augmentation operation. Each different data augmentation strategy can be represented by a different $g(\cdot)$. Therefore, the final prediction based on majority voting can be rephrased as

$$f(\mathbf{x}) = \frac{M}{\widetilde{M}} \sum_{i=1}^{\widetilde{M}/M} f[g_i(\mathbf{x})] = \frac{M}{\widetilde{M}} \sum_{i=1}^{\widetilde{M}/M} \widetilde{f}_i(\mathbf{x}), \tag{8}$$

where $\widetilde{f}_i$ can be seen as a series of heterogeneous weak learners that focus on different aspects of training samples. Assuming that all the samples are independent and identically distributed (*i.i.d*), the data augmentation in the test stage can be interpreted as an implicit model ensemble through transforming $\widetilde{\mathbf{x}}$ back to $\mathbf{x}$. It can reduce the bias and variance of the convergent network, thus reducing the risk of "overfitting" problem on the training set while increasing the classification accuracy on the test set.

By reducing the reconstruction error between original sample and augmented samples, we can obtain the updated parameters of deep CNNs. We have observed the parameter distribution of a series of networks $[\widetilde{f}_1, \widetilde{f}_2, \ldots, \widetilde{f}_{\widetilde{M}/M}]$ in Figure 3. It can be seen that the parameter distribution of some networks is obviously different from that of other models. Therefore, these models can be viewed as focusing on different features of the image.

4. Experimental Results and Analysis

4.1. Experimental Setup

4.1.1. Experimental Datasets and Image Preprocessing. Two benchmarks CIFAR-10 and CIFAR-100 represent coarse-grained and fine-grained natural image classification tasks, respectively, which are used to evaluate the effectiveness of full stage data augmentation frameworks under different difficulties. CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset [28]. The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. CIFAR-100 is just like CIFAR-10, except it is a fine-grained version and has 100 classes containing 600 images each. There are 500 training images and 100 testing images in each class. Some examples of images in the two datasets are shown in Figure 4.

Input images of CIFAR-10 and CIFAR-100 datasets [28] are preprocessed in the following manner. Each original image is first color-normalized and then zero-padded to be 40×40 pixels. As for data augmentation at both training and testing stages, all samples are cropped to be 32×32 pixels and followed by a random horizontal flip with the 50% probability during both training and testing stages. The sample size is expanded ten times by considering model stability. Moreover, each image subtracts its own three-channel (R/G/B) mean value to speed up the convergence of deep CNN model.

4.1.2. Network Architectures. Two specially designed deep CNNs are constructed to complete the image classification, as shown in Figure 5. The network trained on CIFAR-100 uses a deeper and broader structure than network trained on CIFAR-10, because finer-grained data require a larger capacity for the model to characterize. Batch normalization layer [38] is added between each convolutional layer and the activation function. Fully connected layers that usually appear in traditional networks are replaced by global average pooling layer [39] to alleviate the "overfitting" problem, except for the last fully connected layer with softmax function used to output the category probability. All the weights in the network are set according to MSRA method [25].

4.1.3. Hyperparameters Setting. Network hyperparameters including initial learning rate, batch size, dropout rate, momentum, weight decay rate, and Leaky-ReLU hyperparameter a are set to 0.01, 512, 0.5, 0.9, 0.0005, and 5, respectively. Nine-tenths of the samples in a batch come from data augmentation. As training iterations, the learning rate is decreased in an exponential form with a decay rate of 0.9.

4.1.4. Experimental Platform. All the training and testing procedures of deep CNNs are carried out under the Caffe deep learning framework [40], based on the workstation consisting of an Intel Core i7-8700k CPU, a NVIDIA GeForce GTX 1080 GPU, 16 gigabytes of memory, and 1 terabyte of storage. The hardware platform and framework only affect the training efficiency rather than the actual classification performance of deep learning model.

4.2. Comparison of Classification Results. In this section, we report the experimental results and discuss possible reasons behind some phenomena. To prove the validity of proposed full stage data augmentation method, fivefold cross validation results are computed for final evaluation and comparison. Furthermore, the classification results of two datasets are presented separately in terms of the fineness degree of object categories.

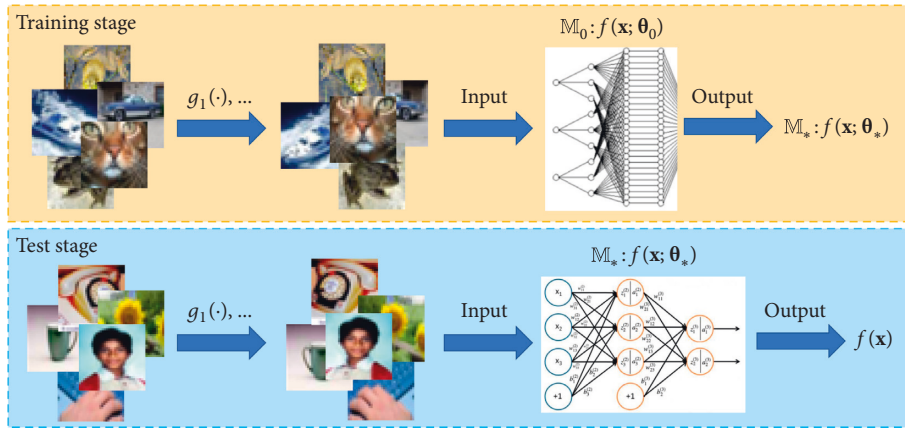


Figure 2: The whole full stage data augmentation framework.

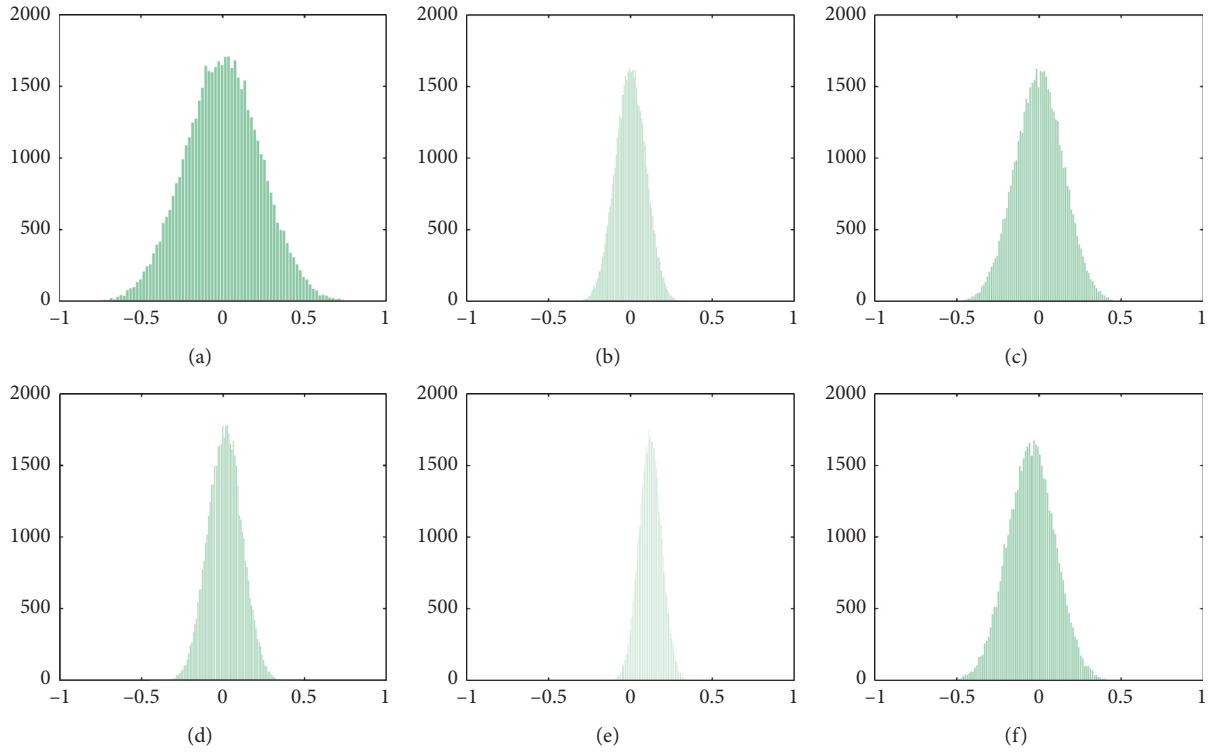


Figure 3: Parameter distribution of a series of deep CNNs by projecting the augmented images back into the input space. The horizontal axis represents the normalized network parameters.

4.2.1. Coarse-Grained Image Classification Results. We first report the baseline classification results before and after using full stage data augmentation method, as shown in Figure 6. The results show that the full stage data augmentation framework leads to a significant improvement of classification accuracy for deep CNN model. It can be clearly seen that the confusion matrix of original network is more confusing than that of the data-augmented network. In fact, the average classification accuracy of CIFAR-10 has increased from 85.7% to 93.4%. Furthermore, we also report the results of using data augmentation only during training or testing phase. Data augmentation in training phase is more effective than that in testing phase, with an accuracy increase of about 3%. However, the deep CNN also needs longer training time. In contrast, the additional reasoning costs associated with the data augmentation in the test phase can be neglected. In other words, full stage data augmentation framework improves the performance of traditional training data augmentation methods without introducing additional costs.

Then we observe the effectiveness of various data augmentation methods under the proposed full stage data augmentation framework, including translation, horizontal flip, rotation, scale transformation, and noise disturbance.

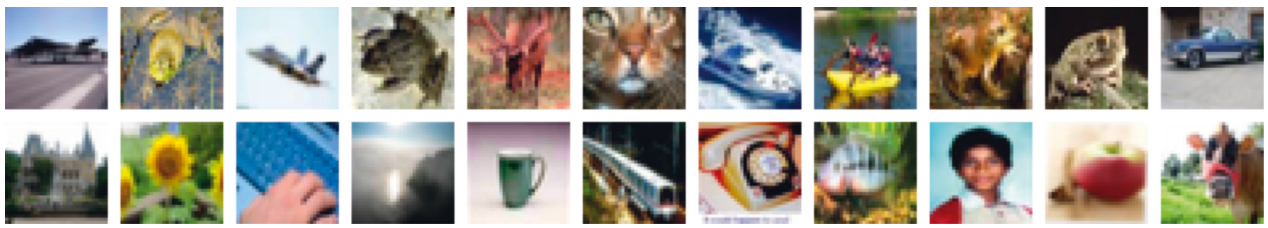


Figure 4: Some examples of images in CIFAR-10 (first row) and CIFAR-100 (second row).

CIFAR-10	CIFAR-100
32 × 32 × 3 input	32 × 32 × 3 input
3 × 3 conv, 64 3 × 3 conv, 64 Batch normalization	3 × 3 conv, 128 3 × 3 conv, 128 1 × 1 conv, 128 Batch normalization
2 × 2 max pool dropout (0.1)	2 × 2 max pool dropout (0.1)
3 × 3 conv, 128 3 × 3 conv, 128 Batch normalization	3 × 3 conv, 128 3 × 3 conv, 128 1 × 1 conv, 128 Batch normalization
2 × 2 average pool dropout (0.1)	2 × 2 average pool dropout (0.1)
3 × 3 conv, 128 3 × 3 conv, 128 Batch normalization	3 × 3 conv, 256 3 × 3 conv, 256 1 × 1 conv, 256 Batch normalization
Global average pool dropout (0.5)	Global average pool dropout (0.5)
Dense (10) Softmax	Dense (100) Softmax

Figure 5: The structure of two specially designed deep CNNs.

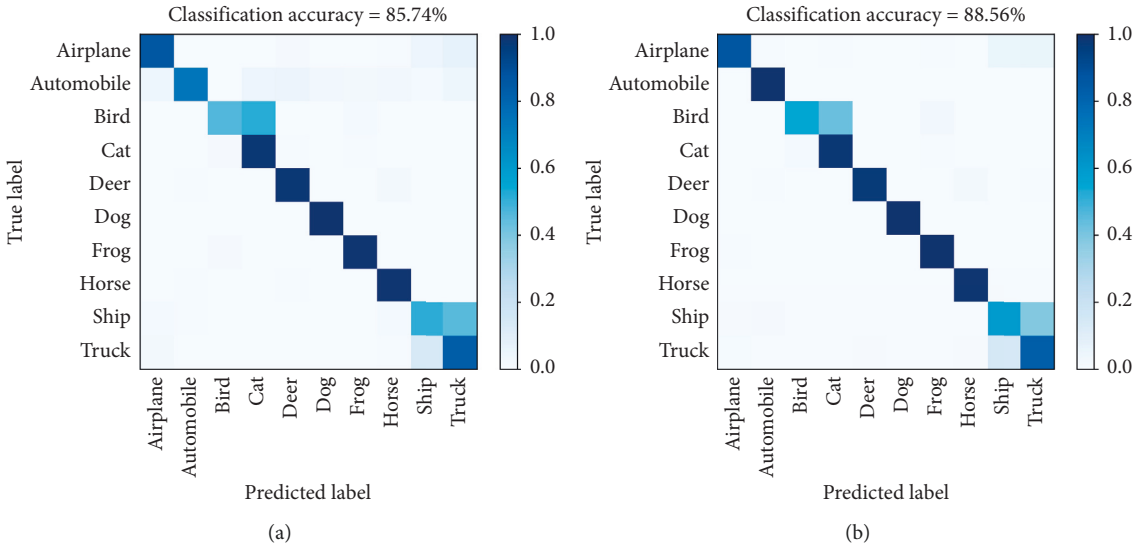


Figure 6: Continued.

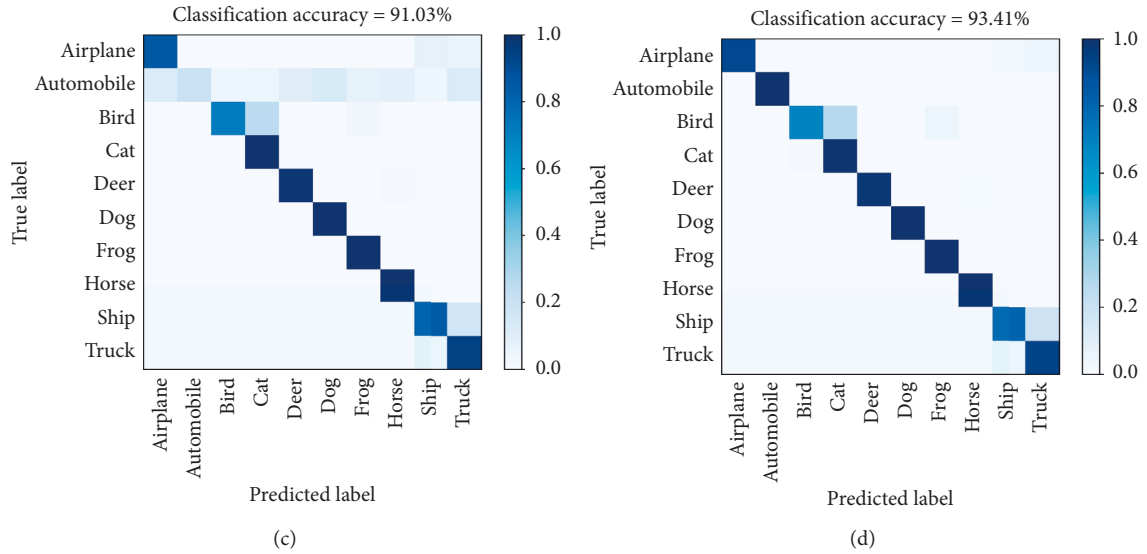


Figure 6: Classification results on CIFAR-10 before and after using full stage data augmentation method. (a)–(d) represent "no data augmentation," "augmentation in training stage," "augmentation in test stage," and "full stage data augmentation," respectively.

Translation and horizontal flip are based on the settings given above. The rotation range is from negative to positive five degrees, and the step size is 1 degree. Four Gaussian convolutional kernels with different fuzzy radii are used for the scale transformation, including 3×3, 5×5, and 7×7 pixels. The noise disturbance adds Gaussian white noises with different intensity to the original image, including 0.01, 0.05, 0.1, and 0.2. During the testing phase, the data augmentation strategy of test samples is consistent with the training samples. The experimental results are presented in Table 1. The results show that any data augmentation strategy under the full stage data augmentation framework can improve the classification performance of the deep CNN. As far as CIFAR-10 is concerned, translation and horizontal flip are the most effective means of data augmentation, while the improvement of classification accuracy caused by rotation and noise disturbance is limited. We think this may be related to the small size of the samples in CIFAR-10. The image itself is only 32×32 pixels and is quite blurred. Therefore, rotation and noise disturbance have a large impact on the image structure, resulting in limited help.

Finally, we compare state-of-the-art results brought by a series of algorithms to demonstrate the effectiveness of proposed full stage data augmentation framework, as shown in Table 2. These algorithms only adopt data augmentation strategies during the training phase. It can be seen that our proposed method has significantly improved the classification accuracy from 89.59% to 93.41%.

4.2.2. Fine-Grained Image Classification Results. Since its high similarity between different classes and the scarcity of samples in each class, fine-grained image classification is more challenging than coarse-grained classification task. Table 3 shows the experimental results on CIFAR-100 before and after using full stage data augmentation method. It can be seen that the average classification accuracy has been increased from 62% to 70%, which exceeds the improvement of CIFAR-10. On the other hand, the performance of single augmentation of training set and test set has also been improved, which increases the accuracy by 4.64% and 1.99%, respectively. In fine-grained image classification task, fewer samples in each classes caused by multiple classes make the data augmentation strategy play a greater role.

Then we also observe the effectiveness of various data augmentation methods on CIFAR-100, as given in Table 4. We find that the effect of data augmentation can only be achieved when the number of augmented samples becomes

Table 1: Classification accuracy of various data augmentation methods on CIAFR-10 under the proposed full stage data augmentation framework.

Methods	CIFAR-10 (%)
Translation	91.41
Horizontal flip	90.27
Rotation	88.78
Scale transformation	90.70
Noise disturbance	87.54

Table 2: Comparison with state-of-the-art algorithms on CIFAR-10.

Algorithms	CIFAR-10 (%)
Dropout [41]	84.40
Probout [42]	88.65
NIN + dropout [43]	89.59
Maxout + dropout [44]	88.32
Stochastic pooling [45]	84.86
Probabilistic weighted pooling [46]	88.71
Our method	93.41

Table 3: Experimental results on CIFAR-100 before and after using full stage data augmentation method.

Methods	CIFAR-100 (%)
No data augmentation	61.85
Augmentation in training stage	66.49
Augmentation in testing stage	63.84
Full stage augmentation	70.22

Table 4: Classification accuracy of various data augmentation methods on CIFAR-100 under the proposed full stage data augmentation framework.

Methods	CIFAR-100 (%)
Translation	63.73
Horizontal flip	64.11
Rotation	62.20
Scale transformation	64.83
Noise disturbance	60.47

larger than that of CIFAR-10. Moreover, fine-grained image classification is more sensitive to data augmentation strategy, and some methods may even have negative effects, such as the noise disturbance, which reduces the classification accuracy by 1.38%. This is related to the structure of dataset and the distribution condition of all samples. The distribution of samples in the dataset should be smooth; otherwise, it is easy to overlearn and cause the "overfitting" problem, which results in poor generalization on unseen test samples.

Table 5 gives the comparison results of CIFAR-100 with a series of state-of-the-art algorithms. It is worth noting that if dropout is employed improperly like in [45], the classification accuracy would decrease. Probabilistic weighted pooling [46] can also be regarded as model ensemble in the test stage, thus achieving good result (62.87%). Finally, classification accuracy has been increased from 64.32% to 69.22% by using full stage data augmentation framework.

4.3. Relationship between Data Augmentation and Network Generalization Ability. In practice, one of the obstacles to the mature application of data augmentation strategies in deep learning is that it is difficult for people to determine how many samples are efficient. In other words, the regularization intensity of data augmentation is usually uncertain. Although some scholars [47, 48] have suggested that the more samples the better, developers usually have to weigh the network performance and time cost in training and reasoning. In this part, we discuss the relationship between data augmentation and network generalization ability through extensive experiments.

We set up a series of data augmentation schemes of different sizes and observe the classification performance of the network in an attempt to mine and establish the relationship between the expanded sample size and the network generalization boundary. The experimental results of CIFAR-10 and CIFAR-100 are shown in Figure 7. The classification results of deep CNN with full stage data augmentation are always better than the baseline results on both datasets, regardless of the augmentation strength. In other words, the size of the dataset directly determines the quality of the deep learning models. On the other hand, the effect of various data augmentation methods clearly has a saturation interval. Once the augmentation strength exceeds this threshold, the performance of the network on the test set no longer grows and tends to be stable. In this case, we believe that the data itself or the network structure itself has become an "information bottleneck," which hinders the further improvement of classification accuracy. At this point, the direction of improvement should be considered from data sources with higher quality and more advanced network structures.

Table 5: Comparison with state-of-the-art algorithms on CIFAR-100.

Algorithms	CIFAR-100 (%)
Probout [42]	61.86
NIN + dropout [43]	64.32
Maxout + dropout [44]	61.43
Stochastic pooling [45]	57.49
Probabilistic weighted pooling [46]	62.87
Our method	70.22

Then we visualize the convolutional kernels in the first layer of deep CNN trained on CIFAR-10/100, as shown in Figure 8. All of them are ordered according to the value of their L1-norm. Visual spatial images can be combined by decoupled component-level convolutional kernels and mapped to different geometric spaces. These convolutional kernels reflect the organization information in the images extracted by the deep CNN, that is, the features of the object in the image. Generally speaking, the ordered convolutional kernels usually mean effective extraction of the organization information, while chaotic ones mean the "overfitting" of networks [49]. This is helpful for establishing the relationship between regularization intensity and network generalization ability and provides standards or principles to guide algorithm development or model structure improvement.

4.4. Impact on Network Optimization and Generalization. Data augmentation in training phase inevitably affects the network convergence, including the convergence speed and the final convergence position. We observe the decrease curve of the loss function of deep CNN on the training sets of CIFAR-10 and CIFAR-100 (see Figure 9) to analyze the impact of full stage data augmentation framework on network convergence. The loss of the model can reach the same level within the two epochs and eventually stabilize at 15 epochs, which means that the impact of data augmentation during the training phase on the convergence speed of the network can be ignored. On the other hand, the final loss value of the augmented model is slightly higher than that of the original model due to the regularization caused by the diversity of expanded samples. Actually, the generalization capability of deep CNNs, that is, the classification

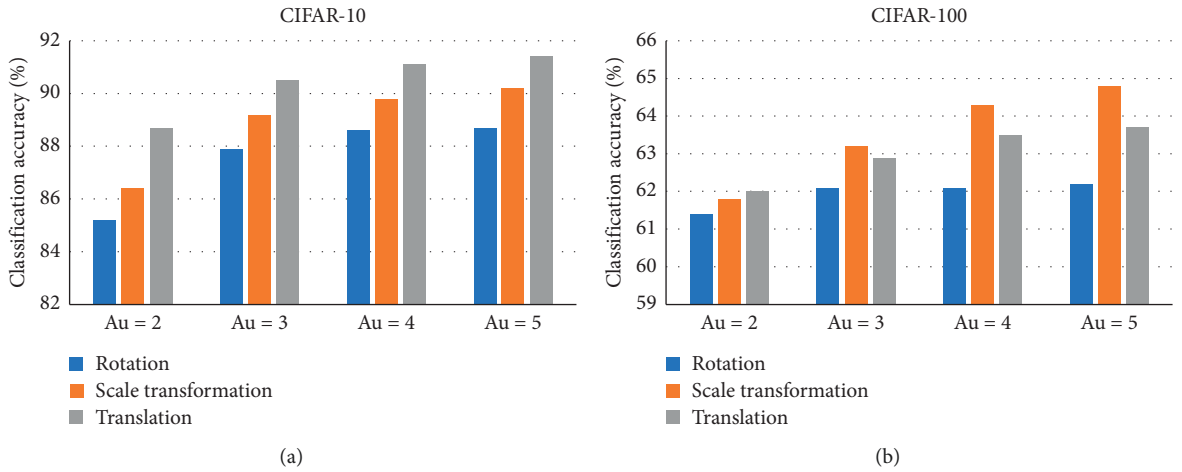


Figure 7: The relationship between the expanded sample size and network generalization ability, in which the number of images is expanded to Au times.

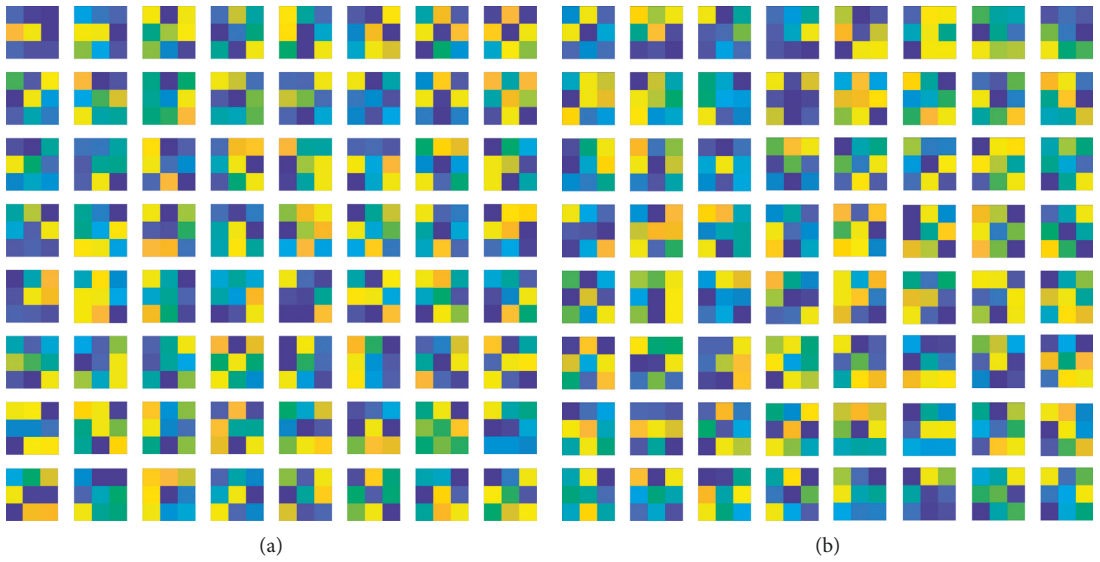


Figure 8: Visualization of convolutional kernels in the first layers of the deep CNNs trained on CIFAR-10 (a) and CIFAR-100 (b), respectively.

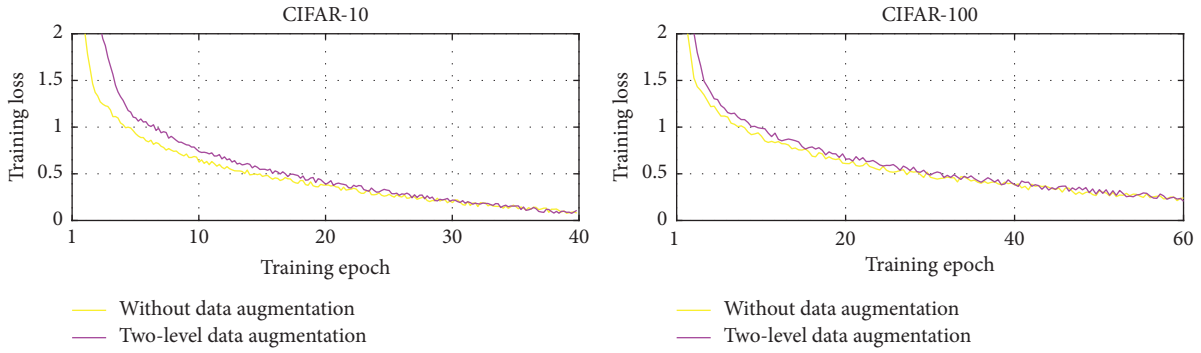


Figure 9: The optimization of the loss function of deep CNNs trained on CIFAR-10 and CIFAR-100, respectively.

performance on the test set rather than the training set, is our pursuit. Therefore, the optimization gap brought by data augmentation has no impact on the application of deep CNN in practice.

5. Conclusion

In this paper, we propose a full stage data augmentation framework to improve the accuracy of deep CNNs, which can also play the role of model ensemble without introducing additional model training costs. Simultaneous data augmentation during training and testing stages can ensure network convergence and enhance its generalization capability on unseen test samples. Furthermore, this framework is universal for any network architecture and data augmentation strategy and therefore can be applied to various deep learning based tasks. Finally, experiments about image classification on the coarse-grained dataset CIFAR-10 and fine-grained dataset CIFAR-100 demonstrate the effectiveness of the proposed framework by comparison with state-of-the-art algorithms. Through visualization of convolutional kernels, we have demonstrated that the ordered convolutional kernels usually mean effective extraction of the organization information, while chaotic ones mean the "overfitting" of networks. We have also analyzed the relationship between data augmentation and network generalization ability and observed the impact of the framework on the convergence of deep CNNs. The empirical results have shown that the data augmentation framework can improve the generalization ability of deep learning models, and it can have a negligible impact on the model's convergence.

As for future research directions, we plan to apply the proposed full stage data augmentation method to more complex CNN structures and some other machine learning related applications, such as liveness detection and gait and face recognition. We believe that it can help improve the performance of deep learning models in a series of tasks.

Data Availability

The experimental data of CIFAR-10 and CIFAR-100 used to support the findings of this study are included within the paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant 2018YFC0831503), the National Natural Science Foundation of China (Grant 61571275), and Fundamental Research Funds of Shandong University (Grant 2018JC040).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 2, pp. 84–90, 2017.

[2] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards reaching human performance in pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 973–986, 2018.

[3] J. Lu, G. Wang, and J. Zhou, "Simultaneous feature and dictionary learning for image set based face recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 4042–4054, 2017.

[4] Q. Zheng, M. Yang, J. Yang, Q. Zhang, and X. Zhang, "Improvement of generalization ability of deep CNN via implicit regularization in two-stage training process," *IEEE Access*, vol. 6, pp. 15844–15869, 2018.

[5] Q. Zheng, M. Yang, Q. Zhang, and J. Yang, "A bilinear multi-scale convolutional neural network for fine-grained object classification," *IAENG International Journal of Computer Science*, vol. 45, no. 2, pp. 340–352, 2018.

[6] Q. Zheng, X. Tian, N. Jiang, and M. Yang, "Layer-wise learning based stochastic gradient descent method for the optimization of deep convolutional neural network," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 4, pp. 5641–5654, 2019.

[7] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, J. M. Alvarez, and S. Gould, "Incorporating network built-in priors in weakly-supervised semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1382–1396, 2018.

[8] Y. Li, Y. Liu, G. Liu, D. Zhai, and M. Guo, "Weakly supervised semantic segmentation based on EM algorithm with localization clues," *Neurocomputing*, vol. 275, pp. 2574–2587, 2018.

[9] Q. Zhang, M. Liu, and S. Zhang, "Node topology effect on target tracking based on UWSNs using quantized measurements," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2323–2335, 2017.

[10] Q. Zheng, X. Tian, S. Liu et al., "Static hand gesture recognition based on Gaussian mixture model and partial differential equation," *IAENG International Journal of Computer Science*, vol. 45, no. 4, pp. 569–583, 2018.

[11] Q. Zheng, X. Tian, M. Yang, Y. Wu, and J. Su, "PAC-Bayesian framework based drop-path method for 2D discriminative convolutional network pruning," *Multidimensional Systems and Signal Processing*, 2019.

[12] J. Li, M. Yang, Y. Liu et al., "Dynamic hand gesture recognition using multi-direction 3D convolutional neural networks," *Engineering Letters*, vol. 27, no. 3, pp. 490–500, 2019.

[13] H. Zhuang, M. Yang, Z. Cui, and Q. Zheng, "A method for static hand gesture recognition based on non-negative matrix factorization and compressive sensing," *IAENG International Journal of Computer Science*, vol. 44, no. 1, pp. 52–59, 2017.

[14] D. Li, Y. Yang, Y. Song, and T. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5543–5551, Venice, Italy, 2017.

[15] Q. Zheng, M. Yang, Q. Zhang, X. Zhang, and J. Yang, "Understanding and boosting of deep convolutional neural network based on sample distribution," in *Proceedings of the IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 823–827, Chengdu, China, 2017.

[16] A. Gudigar, S. Chokkadi, and U. Raghavendra, "A review on automatic detection and recognition of traffic sign," *Multimedia Tools and Applications*, vol. 75, no. 1, pp. 333–364, 2016.

[17] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 5574–5584, Long Beach, CA, USA, 2017.

[18] Z. Zhang, C. Xu, J. Yang, Y. Tai, and L. Chen, "Deep hierarchical guidance and regularization learning for end-to-end depth estimation," *Pattern Recognition*, vol. 83, pp. 430–442, 2018.

[19] Q. Zheng, X. Tian, M. Yang, and H. Wang, "Differential learning: a powerful tool for interactive content-based image retrieval," *Engineering Letters*, vol. 27, no. 1, pp. 202–215, 2019.

[20] T. Kobayashi, "Flip-invariant motion representation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5628–5637, Venice, Italy, 2017.

[21] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[22] C. Szegedy, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, 2015.

[23] N. Srivastava, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1058–1066, Atlanta, GA, USA, 2013.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, Santiago, Chile, 2015.

[26] Q. Zhang, A. Liu, and X. Tong, "Early stopping criterion for belief propagation polar decoder based on frozen bits," *Electronics Letters*, vol. 53, no. 24, pp. 1576–1578, 2017.

[27] G. A. Carpenter and W. D. Ross, "ART-EMAP: a neural network architecture for object recognition by evidence accumulation," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 805–818, 1995.

[28] A. Krizhevsky, N. Vinod, and G. Hinton, "The CIFAR-10 dataset," 2014, http://www.cs.toronto.edu/kriz/cifar.html55.

[29] H. Inoue, "Data augmentation by pairing samples for images classification," in *Proceedings of the International Conference on Learning Representation (ICLR)*, pp. 313–322, Vancouver, BC, Canada, 2018.

[30] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 547–554, 2018.

[31] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *IEEE Access*, vol. 5, pp. 5858–5869, 2017.

[32] M. Frid-Adar, I. Diamant, E. Klang et al., "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," 2018, https://arxiv.org/abs/1712.04621.

[33] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, *Automatic Brain Tumor Segmentation Using Convolutional Neural Networks with Test-Time Augmentation*, Springer, in *Proceedings of the International MICCAI Brainlesion Workshop*, pp. 61–72, Springer, Granada, Spain, September 2018.

[34] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "DisturbLabel: regularizing CNN on the loss layer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4753–4762, Las Vegas, NV, USA, 2016.

[35] H. Proenca, J. C. Neves, T. Marques, S. Barra, and J. C. Moreno, "Joint head pose/soft label estimation for human recognition in-the-wild," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 38, no. 12, pp. 2444–2456, 2016.

[36] X. Zhang, Y. Zou, and S. Wei, "Dilated convolution neural network with leaky-ReLU for environmental sound classification," in *Proceedings of the International Conference on Digital Signal Processing (DSP)*, pp. 1–5, London, UK, 2017.

[37] X. Tian, "A electric vehicle charging station optimization model based on fully electrified forecasting method," *Engineering Letters*, vol. 27, no. 4, pp. 731–743, 2019.

[38] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 448–456, Lille, France, 2015.

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, Las Vegas, NV, USA, 2016.

[40] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, Orlando, FL, USA, 2014.

[41] G. E. Hinton, N. Srivastave, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaption of feature detectors," 2012, https://arxiv.org/abs/1207.0580.

[42] J. Springenberg and M. Riedmiller, "Improving deep neural networks with probabilistic maxout units," in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–10, Banff, Canada, 2014.

[43] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–10, Banff, Canada, 2014.

[44] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout Networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1319–1327, Atlanta, GA, USA, 2013.

[45] M. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–9, 2013, Scottsdale, AZ, USA.

[46] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.

[47] G. Weisz, P. Budzianowski, P.-H. Su, and M. Gasic, "Sample efficient deep reinforcement learning for dialogue systems with large action spaces," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2083–2097, 2018.

[48] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, "Deep learning for logo recognition," *Neurocomputing*, vol. 245, pp. 23–30, 2017.

[49] K. Knauf, D. Memmert, and U. Brefeld, "Spatio-temporal convolution kernels," *Machine Learning*, vol. 102, no. 2, pp. 247–273, 2016.