Complexity

Collaborative Big Data Management and Analytics in Complex Systems with Edge 2021

Lead Guest Editor: Xuyun Zhang
Guest Editors: Shancang Li, Yuan Yuan, and Wanchun Dou

Collaborative Big Data Management and Analytics in Complex Systems with Edge 2021

Collaborative Big Data Management and Analytics in Complex Systems with Edge 2021

Lead Guest Editor: Xuyun Zhang
Guest Editors: Shancang Li, Yuan Yuan, and Wanchun Dou

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Complexity
Volume 2023, Article ID 7111548, 19 pages
https://doi.org/10.1155/2023/7111548

Research Article

PSO with Mixed Strategy for Global Optimization

Jinwei Pang,[1] **Xiaohui Li**,[2] **and Shuang Han**[3]

[1]*School of Computer and Control Engineering, Yantai University, Yantai, China*
[2]*Harbin Vocational and Technical College, Harbin, China*
[3]*Department of Computer Science and Technology, Harbin Engineering University, Harbin, China*

Correspondence should be addressed to Jinwei Pang; pangjinwei789@qq.com

Received 9 December 2021; Revised 31 December 2022; Accepted 19 June 2023; Published 6 September 2023

Academic Editor: Alejandro F. Villaverde

Particle swarm optimization (PSO) is an evolutionary algorithm for solving global optimization problems. PSO has a fast convergence speed and does not require the optimization function to be differentiable and continuous. In recent two decades, a lot of researches have been working on improving the performance of PSO, and numerous PSO variants have been presented. According to a recent theory, no optimization algorithm can perform better than any other algorithm on all types of optimization problems. Thus, PSO with mixed strategies might be more efficient than pure strategy algorithms. A mixed strategy PSO algorithm (MSPSO) which integrates five different PSO variants was proposed. In MSPSO, an adaptive selection strategy is used to adjust the probability of selecting different variants according to the rate of the fitness value change between offspring generated by each variant and the personal best position of particles to guide the selection probabilities of variants. The rate of the fitness value change is a more effective indicator of good strategies than the number of previous successes and failures of each variant. In order to improve the exploitation ability of MSPSO, a Nelder–Mead variant method is proposed. The combination of these two methods further improves the performance of MSPSO. The proposed algorithm is tested on CEC 2014 benchmark suites with 10 and 30 variables and CEC 2010 with 1000 variables and is also conducted to solve the hydrothermal scheduling problem. Experimental results demonstrate that the solution accuracy of the proposed algorithm is overall better than that of comparative algorithms.

1. Introduction

In recent years, optimization algorithms are applied more and more widely in various fields [1]. One of the most famous ones is PSO. In 1995, Kennedy and Eberhart developed particle swarm optimization (PSO) [2]. PSO as a global optimization method is an important tool to solve difficult optimization problems without a good problem-specific approach efficiently. Its original inspiration comes from birds flocking behaviours. In PSO, each individual in the population is a particle. A particle represents a potential solution in solution space. Particles scan the search area and converge to the optimum by flying in the space and adjusting its flying velocity based on its personal best historical experience and the best solution in the population. PSO is a robust stochastic optimization algorithm that is easy to implement. Its parameter settings are negligible. On account of its simple realization and high efficiency, PSO has been successfully applied to various real-world problems such as wireless sensor networks [3], feature selection [4], traffic control [5], road identification [6], task allocation [7], and crowd user selection [8].

Recently, various improvements of PSO are proposed to enhance these comprehensive performances. In this research, according to a recent theory [9], this study presented a mixed strategy PSO (MSPSO). In the theory, the hardest problem to one evolutionary algorithm might be the easiest for another algorithm and vice versa. Thus, the mixed strategy PSO algorithms might be more efficient than pure strategy PSO algorithms. Just as a company wants to run well, it needs talents who are good at management, good at marketing, and good at purchasing to work together. If a company employs talents who are good at management for all work, the company will not operate well because employees who are not good at what they do spend more time and get worse results. Inspired by this theory, MSPSO integrates five different PSO variants and adopts a new probability update strategy according to the proportion of differences in fitness values. According to our experimental verification, the probability of those variants is guided by the rate of change of fitness value between offspring generated by each variant and the personal best positions of the particles. According to the rank of the rate of change, the variants are assigned the different probabilities. Using the rate of fitness change to guide the selection probabilities of variants could increase the probability of selecting excellent variants than using the number of previous successes and failures of each variant. In addition, in order to enhance the exploitation ability of MSPSO, a local search method inspired by the Nelder–Mead method is proposed.

In summary, we have made the following contributions:

(1) We propose a cooperative strategy to integrate multiple PSO operators, and the integrated algorithm can achieve better generalization capability

(2) We add a local search operator to the integrated algorithm, so that the algorithm can further obtain better performance

(3) We also demonstrate the performance of MSPSO on benchmark suites and real-world problem instances

The rest of the study is organized as follows. First, related methods are reviewed in Section 2. Second, Section 3 introduces the MSPSO. Third, Section 4 evaluates the proposed MSPSO and gives the results of the experiments. Finally, the conclusion of the study is shown in Section 5.

2. Related Work

2.1. Canonical PSO. In the optimization process, the velocity vector V_i for the ith particle in the population is updated using (1) given in [2] iteratively through the guidance of $p\text{best}_i$ and $g\text{best}$.

$$v_{id} = v_{id} + c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right) + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right). \tag{1}$$

Acceleration parameters c_1 and c_2 are usually set to 2.0. r_{1id} and r_{2id} are two random numbers within [0,1] for the dth dimension of the ith particle.

To avoid the premature convergence, the authors in [10] introduced an inertia weight ω to update the flying velocities of particles. The particle velocity is adjusted through the following formula:

$$v_{id} = \omega * v_{id} + c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right) + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right). \tag{2}$$

In (2), ω commonly decreases linearly from 0.9 to 0.4 with generations to balance exploration capability and exploitation capability. A large value of ω enhances the exploration capability, whereas a small value of ω encourages the capability of convergence during the search process.

2.2. PSO Variants. To enhance the performance of PSO on global optimization problems, a lot of researches have been working on improving PSO algorithms, and numerous PSO variants have been presented. Designing new strategies, new techniques and topological structures of PSO are an important research trend. Various topologies have been suggested. In PSO, the trajectory of particles is adjusted by their own personal best positions and the best position in the population. However, this may cause premature convergence when solving multimodal functions. Because the best particle in the population is the best solution for the whole population, it could be a local optimum for a multimodal function and is far away from the global optimum. The authors in [11] proposed a social learning PSO (SL-PSO) which introduced social learning into PSO. The advantage of social learning was that individuals could learn from others without paying for their own trials and mistakes. In SL-PSO, each particle was updated based on any better particles in the current population. Furthermore, to reduce parameter settings, SL-PSO proposed a dimension-dependent parameter control method. Compared with other optimization algorithms, SL-PSO could be implemented easily, be computed efficiently, and require no complicated adjustment of the control parameters. In order to accelerate convergence speed and improve exploitation ability, the authors in [12] proposed prey-predator PSO (PP-PSO). PP-PSO achieved this goal by deleting or transforming "slothful particles" which were the particles with low velocities. It was hard for these slothful particles to find the global optimum, and this reduced the convergence speed. Furthermore, in order to enhance population diversity, PP-PSO designed a proportional-integral control parameter to control the population to fluctuate within a relatively stable range during the iterative process. The above-mentioned PSO variant algorithm mainly improves the classical PSO algorithm from the perspective of designing a new information-sharing mode between particles and building a new particle search model.

However, when solving some complex optimization problems, PSO and its variants are still prone to premature convergence in the search process. Furthermore, when trapped in the local optimum, it is difficult for particles to get rid of this region. Therefore, in the past decades, researchers have also tried to solve this problem by proposing various improvement strategies based on existing PSO algorithms.

Another popular modification is to combine PSO with other mathematical methods or evolutionary computation techniques. The authors in [13] integrated a PSO algorithm with the sine cosine algorithm (SCA) and the Lévy flight approach, to overcome the shortcoming that PSO tended to fall into a local optimum. The solution in SCA was updated by sine and cosine functions to ensure the exploitation and exploration capabilities. In addition, SCA used Lévy distribution which was a more effective search to produce a random walk in the search space. The combination of SCA, Lévy flight, and PSO enhanced the exploration capability of the original PSO and prevented being trapped in the local minimum. In addition, the hybridization of PSO with GAs has also been presented in [14, 15]. A hybrid PSO with bat algorithm (BA) has been proposed in [16] for numerical optimization problems. A communicating strategy provided information flow between the population of PSO and the population of BA. In this work, several best individuals in BA replaced the worst individuals in PSO after fixed iterations, and on the contrary, the finest particles of PSO replaced the poorer individuals of BA.

Multipopulation strategy and ensemble optimizer are also effective methods to optimize the performance of PSO. In order to avoid the phenomena of "oscillation" and "two steps forward, one step back" in PSO, the authors in [17] proposed a two-swarm learning PSO algorithm called TSLPSO. The algorithm hybridized two different learning strategies which were dimensional learning strategy (DLS) and comprehensive learning strategy, respectively. One of the swarms used DLS to construct the learning exemplars. DLS used the information of the best particle in the population for the local search of the particles. However, in order to guide the global search, the other swarm used the comprehensive learning strategy to construct the learning exemplars. In [18], Xu et al. constructed a DMS-PSO-CLS algorithm that combined the dynamic multiswarm particle swarm optimizer (DMS-PSO) and a new cooperative learning strategy (CLS). In the CLS subpopulation, in order to learn more excellent examples, the two poor particles updated their dimensions with the better particle which was selected from two random subswarms using a tournament selection strategy. By using this method, particles could search the global optimum more easily. The simulation results showed that the performance of the DMS-PSO-CLS algorithm was superior compared with other comparison PSO variants. The above algorithms also have their limitations. For example, some hybrid algorithm frameworks require more computing resources to execute the iterative process of different algorithms, while most multipopulation strategies cannot perform fine local search at the later stage of the search process, so it is difficult to obtain the final search results with high accuracy.

2.3. Complementary Strategy Theorem. The authors in [19] proposed a complementary strategy theorem. According to this theorem, mixed strategy evolutionary algorithms might outperform pure strategy evolutionary algorithms. One advantage was that the overall performance of mixed strategy evolutionary algorithms might be the same as the best performance of pure strategy evolutionary algorithms.

Theorem 1. *If a pure strategy evolutionary algorithm PS_2 is better than another pure strategy evolutionary algorithm PS_1, then for any initial population P, the expected hitting time of mixed strategy evolutionary algorithms MS derived from PS_1 and PS_2 satisfies that $m_{MS}(P) \geq m_{PS2}(P)$ and $m_{MS}(P) > m_{PS2}(P)$ for some state P.*

Theorem 2. *If a pure strategy evolutionary algorithm PS_2 is equivalent to another pure strategy evolutionary algorithm PS_1, then for any initial population P, the expected hitting time of mixed strategy evolutionary algorithm MS derived from PS_1 and PS_2 satisfies that $m_{MS}(P) = m_{PS2}(P)$.*

Theorem 3. *If a pure strategy evolutionary algorithm PS_2 complements with another pure strategy evolutionary algorithm PS_1, then there exists a mixed strategy evolutionary algorithm MS derived from PS_1 and PS_2, and its expected hitting time satisfies that $m_{MS}(P) \leq m_{PS2}(P)$ for any initial population P and $m_{MS}(P) < m_{PS2}(P)$ for some initial population P.*

Theorem 4 (complementary strategy theorem). *The condition that a pure strategy evolutionary algorithm PS_2 is complementary to another pure strategy evolutionary algorithm PS_1 is sufficient and necessary if there exists a mixed strategy evolutionary algorithm MS derived from them such that $m_{MS}(P) \leq m_{PS2}(P)$ for any initial population P and $m_{MS}(P) < m_{PS2}(P)$ for some initial population P.*

The complementary strategy theorem can be interpreted intuitively as follows:

(1) If one pure strategy evolutionary algorithm is better than another pure strategy evolutionary algorithm, then the design of a mixed strategy evolutionary algorithm with the same performance as the better pure strategy evolutionary algorithm is impossible. So mixed strategy evolutionary algorithms do not usually outperform pure strategy evolutionary algorithms that they derived from.

(2) If one pure strategy evolutionary algorithm is complementary to another, then the design of a mixed strategy evolutionary algorithm better than both pure strategy evolutionary algorithms is possible. However, this does not mean all mixed strategy evolutionary algorithms will outperform pure strategy evolutionary algorithms that they derived from.

(3) The following principle should be followed when a better-mixed strategy evolutionary algorithm is designed: if a pure strategy evolutionary algorithm has a better performance than another at a state, then

the mixed strategy evolutionary algorithm should apply the pure strategy with a higher probability at that state.

3. Mixed Strategy PSO

3.1. PSO Strategies. MSPSO hybridizes PSO [10], MCLPSO [20], LIPS [21], HPSO-TVAC [22], and FDR-PSO [23] with an adaptive selection strategy. Velocity update formulae of these four PSO variants except PSO are given as follows.

3.1.1. MCLPSO. We proposed a modified CLPSO (MCLPSO) in [20] a few years ago. Compared with CLPSO, MCLPSO could improve the convergence ability while maintaining the population diversity. Furthermore, MCLPSO has a better balance of exploration and exploitation than CLPSO. The updating equation of MCLPSO for a particle velocity is given as follows:

$$\begin{aligned}
&if\ \text{rand} < \alpha * \left(1 - \frac{g-1}{\text{Max\_Gen}}\right),\\
&\quad v_{id} = \omega * \left(v_{id} - \text{mean}v_d\right) + c * r_{id} * \left(p\text{best}_{fi(d)d} - x_{id}\right),\\
&\quad\quad \text{else}\\
&\quad v_{id} = \omega * v_{id} + c * r_{id} * \left(p\text{best}_{fi(d)d} - x_{id}\right),
\end{aligned} \tag{3}$$

where $g = [1, 2, \ldots, \text{Max\_Gen}]$ is the current generation number, α is an adjustment coefficient between 0 and 1, Max _Gen is the maximum number of generations, rand is a random number within range [0, 1], meanv_d is the dth dimension of the average value of velocities in the whole population, and $p\text{best}_{fi(d)d}$ represents the dth dimension of the best position of the particle located in a list of particles selected randomly from the whole population, and the rest of the parameters have the same meanings as those in (2).

Using the above equation, the velocity of MCLPSO is fast with a high probability in the early stage of the search. Conversely, in the later stage, the velocity is slow with a high probability for better exploitation.

3.1.2. LIPS. LIPS used the best experiences of adjacent particles rather than the global best experience of the population to guide the particles to the optimum [21]. This algorithm adopted the personal best position of neighbor particles measured by Euclidean distance to adjust the particle velocity. The formula is given as follows:

$$\begin{aligned}
v_{id} &= 0.7298 * \left(v_{id} + \phi * \left(pn_{id} - x_{id}\right)\right),\\
pn_i &= \frac{\sum_{k=1}^{n\text{size}} \left(\phi_k * n\text{best}_k\right)/n\text{size}}{\sum_{k=1}^{n\text{size}} \phi_k},\\
\phi_k &\sim U(0, 4.1/n\text{size}),
\end{aligned} \tag{4}$$

where $n\text{best}_k$ is the best position of the kth neighbor particle of the ith particle, ϕ_k is a random number that obeys uniform distribution within [0, 4.1/$n$size], and $n$size is the number of neighbor particles.

3.1.3. HPSO-TVAC. In order to make the particles converge quickly to the global optimum, the authors in [22] designed a new formula for calculating velocity without using the previous velocity. The formula is given as follows:

$$v_{id} = c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right) + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right), \tag{5}$$

where c_1, r_{1id}, c_2, and r_{2id} have the same meanings as those in (1).

3.1.4. FDR-PSO. In order to avoid the premature convergence, the authors in [23] added the position of neighbor particle in the formula of particle velocity. The formula is given as follows:

$$\begin{aligned}
v_{id} = {} & \omega * v_{id} + c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right)\\
& + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right) + c_3 * r_{3id} * \left(n\text{best}_{id} - x_{id}\right),
\end{aligned} \tag{6}$$

where $n\text{best}_{\text{id}}$ is the dth dimension of the best experience of the neighbor of the ith particle which minimizes the fineness-distance ratio (FDR), and the rest of the parameters have the same meanings as those in (2). The formula of fineness-distance ratio for a minimization problem is given as follows:

$$\text{FDR} = \frac{\text{Cost}\left(P_i\right) - \text{Cost}\left(X_i\right)}{\left|P_{id} - X_{id}\right|}, \tag{7}$$

where P_i denotes the best experience of other particles in the population except the ith particle.

3.2. Local Search. A local search method inspired from the Nelder–Mead method is used in MSPSO, in order to improve its exploitation ability. The Nelder–Mead method is a numerical algorithm that adapts to local landscapes [24]. It makes down-hill search using a simplex instead of derivatives. Introducing the Nelder–Mead method into MSPSO can further improve the performance of MSPSO.

In our work, we make a modification of the Nelder–Mead method, in order to reduce time consumption. The number of testing points is set to 3, rather than $n + 1$ (the dimension). The following is the detail of the method. Given 3 test points x_1,x_2, x_3, a Nelder–Mead variant is given in Algorithm 1. In Line 1, three individuals are sorted in the order of the function value from low to high. In Line 2, various x_o is the centre of triangle $\triangle x_1x_2x_3$. Lines 3–22 are used to implement the Nelder–Mead process.

3.3. Improved Adaptive Probability Adjustment Method. In some ensemble evolutionary algorithms, the selection probability of different variants is adjusted based on the number of previous successes and failures of each variant at a fixed iteration interval. However, the number of successes

```
Input: population $P$ with three individuals.
Sort the three points in the order: $f(x_1) \le f(x_2) \le f(x_3)$.
Calculate $x_o$ as follows:
        $x_o = 1/3\sum_{k=1}^{3} x_k$
[Reflection] Compute the reflected point $x_r = x_o + \alpha(x_o - x_3)$. Where $\alpha$ is a reflection coefficient. Its standard value is $\alpha = 1$.
if $f(x_1) \le f(x_r) < f(x_3)$, then
    $x_3' =$ the reflected point $x_r$.
else if $f(x_r) < f(x_1)$ then
    [Expansion] Compute the expanded point $x_e = x_o + \gamma(x_r - x_o)$. Where $\gamma$ is an expansion coefficient. Its standard value is $\gamma = 2$.
    if $f(x_e) < f(x_r)$ then
        $x_3' =$ the expanded point $x_e$
    else
        $x_3' =$ the reflected point $x_r$
    end if
else
    [Contraction] Compute the contracted point $x_c = x_o + \rho(x_3 - x_o)$. Where $\rho$ is a contraction coefficient. Its standard values is $\rho = 1/2$.
    if $f(x_c) < f(x_3)$ then
        $x_3' =$ the contracted point $x_c$
    else
        for $i = 2, 3$ do
            [Shrink] $x_i' = x_1 + \sigma(x_i - x_1)$. Where $\sigma$ is a shrink coefficient. Its standard value is $\sigma = 1/2$.
        end for
    end if
end if
Output: population $P = \{x_1', x_2', x_3'\}$.
```

Algorithm 1: Nelder–Mead variant.

and failures is not a perfect indicator of a good strategy because it cannot measure the degree of improvement of successful offspring generated by each variant. Thus, we use the rate of change of fitness value between offspring generated by each variant and the personal best position to adjust the selection probabilities of variants. The adaptive probability adjustment method is given as follows.

Step 5. Initialize the probability p_k of the kth PSO variant to $1/K$ and the change rate Cr_k of the kth variant as 0. In our work, K is equal to 5.

Step 6. Generate a random number rand_{pk}. If $0 \le \text{rand}_{pk} < p_1$, choose the first variant to generate an offspring. If $\sum_{i=1}^{k-1} p_i \le \text{rand}_{pk} < \sum_{i=1}^{k} p_i$, $1 < k < K$, choose the kth variant to generate an offspring. If $\sum_{i=1}^{k} p_i \le \text{rand}_{pk} \le 1$, choose the Kth variant to generate an offspring.

Step 7. If the kth variant is selected to generate an offspring for a particle, then the change rate is recorded as follows:

$$Cr_k = \frac{Cr_k - \left(f_o - f_p\right)}{f_p}, \tag{8}$$

where f_o and f_p are the fitness of the offspring and the fitness of the personal best historical position of the particle, respectively. When the fitness of the offspring is larger than the fitness of *pbest*, i.e., the offspring is worse than the parent, the change rate is reduced. Otherwise, the change rate is increased.

Step 8. After lp generations, update the probability p_k of each PSO variant, and set Cr_k to 0. The probability p_k is updated by the following steps:

(1) Sort the K strategies in descending order based on Cr_k. Get a new sequence K'.

(2) Assign probabilities to the strategies according to their ranking, $p_{K'} = [0.4, 0.3, 0.15, 0.12, 0.03]$, i.e., the probability of the strategy with the largest Cr is set to 0.4.

In this study, we use the change rate rather than the difference between the fitness of the offspring and the fitness of *pbest* to guide the adjustment of the probability. Because the difference cannot reflect the merits and demerits of each strategy especially when the fitness values of a strategy and *pbest* are large, however, the fitness values of another strategy and *pbest* are small. In this situation, the strategy with a small difference in fitness value may be better than the strategy with a large difference. Furthermore, we use a fixed probability distribution to assign a significantly larger selection probability to a good strategy and a significantly smaller selection probability to a bad strategy.

3.4. Framework of MSPSO. MSPSO integrates PSO, MCLPSO, LIPS, HPSO-TVAC, and FDR-PSO together. It adopts two subpopulations in the early stage and a whole population in the later stage of the search process. In the early stage, MSPSO adopts a subpopulation that implements MCLPSO and a subpopulation that implements ensemble

PSO. In the later stage, the whole population implements ensemble PSO. Furthermore, the Nelder–Mead method variant is used in this stage to improve the exploitation ability of MSPSO. In this way, the population diversity and convergence ability of the algorithm can be improved in the early stage. Its convergence ability can be improved in the later stage. The pseudo-code of MSPSO is given in Algorithm 2. In lines 1-2, population and parameters are initialized. Lines 3–10 give the steps of the early stage of MSPSO. In this stage, the whole population is divided into two subpopulations, one of which is composed of μ_1 individuals and the other is composed of $\mu - \mu_1$ individuals. Lines 5–7 indicate that MCLPSO is implemented in the first subpopulation, while lines 8–10 indicate that the ensemble PSO is implemented in the second subpopulation. Lines 11–16 give the steps of the late stage of MSPSO. In this stage, the whole population implements the ensemble PSO, and then, the Nelder–Mead method variant is implemented. In lines 17–22, the best fitness value of the current population is obtained, and the best fitness value of the algorithm is updated if necessary. The framework figure is given in Figure 1.

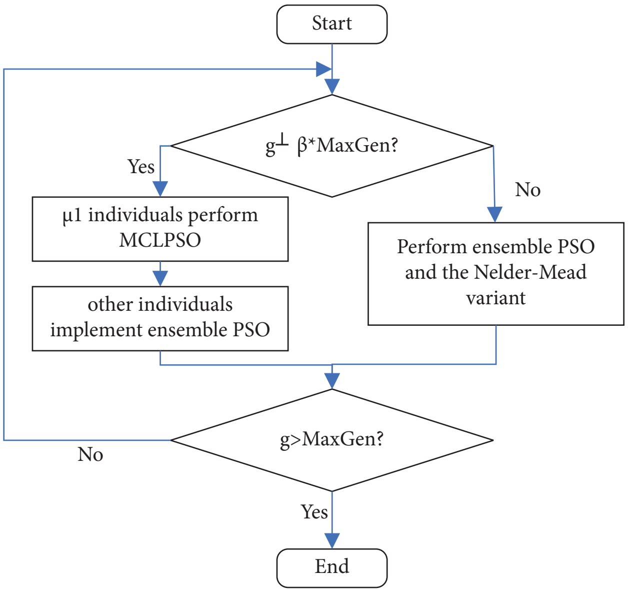

Start
g⊥ β*MaxGen?
Yes
No
μ1 individuals perform MCLPSO
other individuals implement ensemble PSO
Perform ensemble PSO and the Nelder-Mead variant
g>MaxGen?
No
Yes
End

Figure 1: Framework figure of MSPSO.

4. Experiments and Results

4.1. Benchmark Functions and Comparative Algorithms. CEC 2014 [25] benchmark functions are used to evaluate the performance of MSPSO. Benchmark problems in CEC 2014 are developed with several novel features such as novel basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on. In CEC 2014 benchmark suite, F1–F3 are unimodal functions, F4–F16 are simple multimodal functions, F17–F22 are hybrid functions, and F23–F30 are composition functions. In order to evaluate the mean and standard deviation of solution errors, we take thirty independent runs for each algorithm on each problem in 10 and 30 dimensions. For 10 dimensions, each run lasts up to 100,000 function evaluations (FES). For 30 dimensions, it is up to 300,000 FES per run.

The performance of MSPSO is compared with eight other PSO variants, which are PSO [10], CLPSO [26], LIPS [21], HPSO-TVAC [22], FDR-PSO [23], EPSO [27], OSC-PSO [28], and A-PSO [29]. All the selected peer algorithms are proposed in the last decade. EPSO is an ensemble PSO. OSC-PSO drives particles into oscillatory trajectories. A-PSO introduces the nonlinear dynamic acceleration coefficients, logistic map, and a modified particle position update approach in PSO. In order to verify the effectiveness of all the improved strategies proposed by us, we compare MSPSO with the original CLPSO algorithm. The parameter settings of these algorithms are listed in Table 1. The parameter settings of MSPSO in different dimensions are given in Table 2. The high dimension of the function is more complex compared to the low dimension of the function, so we use a larger population size for the high dimension of the function to maintain the diversity of the population. Parameters limit$_{gp}$, α, and β can affect the population diversity and convergence of MSPSO. The larger the limit$_{gp}$, the greater the probability of updating particles in MCLPSO based on the global optimal position. Also, the larger the α, the greater the probability of updating particle velocities in MCLPSO based on the mean velocity of the population. The smaller the β, the more times ensemble PSO is used in the whole population. The benchmark problems with 30 variables in CEC 2014 are more complex than those with 10 variables. So, we use different parameter settings in MSPSO for different dimensions.

Experimental results in the CEC 2014 suite with 10 and 30 dimensions are reported in Tables 3 and 4, respectively. The error is an absolute value of the difference between the best value for 30 runs and the actual optimal value of a specific objective function.

The nonparametric statistical test has become an important method to compare a group of evolutionary algorithms recently [30]. In this study, the Wilcoxon signed-rank test is employed to estimate MSPSO and other PSO variants with the significance level of 5%. For each algorithm, Tables 3 and 4 show the number of best/2nd best/worst ranking, the number of average ranking, and the number of +/=/− in the last three rows, respectively. The algorithms are ranked according to the mean error of each algorithm. Symbol "+," "=," and "−" indicate that MSPSO is significantly better than, similar to, and worse than the compared PSO variant, respectively.

The simulation results on 30 functions with 10 variables in CEC 2014 are shown in Table 3. The results show that MSPSO outperformed the other eight algorithms on functions F2, F3, F5, F6, F13, F17, F21, F24, F25, F27, and F30. For function F8 and F26, the mean error of MSPSO was equal to other optimal algorithms. Specifically, for unimodal functions, compared with other algorithms, MSPSO generally outperformed other algorithms. It is superior or equal to the other eight algorithms on all functions except F1. But it ranked third on functions F1. For simple multimodal functions, MSPSO shows the best performance on four

```
Input: fitness function $f(x)$, dimension $n$, population size $\mu$, subpopulation size $\mu_1$, maximum number of generation Max_Gen,
MCLPSO limit value limit$_{gp1}$*and* limit$_{gp2}$, MCLPSO adjustment coefficient $\alpha_1$ and $\alpha_2$, iterative parameter $\beta$, interval iteration $lp$.
Generate an initial population $P$ consisting of $\mu$ individuals at random.
Set $lp = 10$, $Cr_k = 0$, $p_k = 1/5$.
for $g = 1, 2, \cdots$, Max_Gen do
  if $g \leq \beta *$ Max_Gen
  for $i = 1, 2, \cdots, \mu_1$ do
    Perform MCLPSO to generate offsprings.
  end for
  for $i = \mu_1 + 1, \mu_1 + 2, \cdots, \mu$ do
      Using improved adaptive probability adjustment method to generate offsprings and update the selection probability.
    end for
  else
    for $i = 1, 2, \cdots, \mu$ do
      Using improved adaptive probability adjustment method to generate offsprings and update the selection probability.
    end for
    Update the best three individuals using Algorithm 1;
  end if
  Obtain the fitness value $f_{\text{best}}$ of the optimal in the population
  if $f_{\min} > f_{\text{best}}$ then
    $f_{\min} = f_{\text{best}}$
    $I_{\min} = x_{\min}$
  end if
end for
Output: the best fitness value $f_{\min}$.
```

Algorithm 2: MSPSO.

Table 1: Parameter setting for 8 algorithms.

Methods	ω	Contraction coefficient	Acceleration parameters	Neighbourhood size
PSO	0.9–0.2	—	$c1 = 2$, $c2 = 2$	—
FDR-PSO	0.9–0.2	0.729	$c1 = 1$, $c2 = 1$, $c3 = 2$	—
HPSO-TVAC	—	—	$c1 = 2.5–0.5$ $c2 = 0.5–2.5$	—
LIPS	—	0.729	$c = 2$	3
FIPS	—	0.7298	c1 $= c2 = 2.05$	—
CLPSO	0.9–0.2	—	$c = 1.49445$	—

Table 2: Parameter setting for MSPSO.

Parameters	μ_1	μ	limit$_{gp1}$	limit$_{gp2}$	α_1	α_2	β	lp
CEC 2014 10D	9	20	0.5	0.5	1	0.9	0.9	10
CEC 2014 30D	15	40	0.5	0.5	1	0.9	0.9	10

functions and a moderate performance compared with other comparative algorithms on other functions. With regard to hybrid functions, the performance of MSPSO ranked within the top 4 on all functions. As for composition functions, MSPSO shows the best performance on five functions. And it ranked within the top 2 on all functions except F28. For other algorithms, EPSO and CLPSO perform well, ranking second and third, respectively. PSO, FDR-PSO, HPSO-TVAC, and OSC-PSO perform moderately. All of these algorithms win on no more than 3 functions. LIPS and A-PSO also win on a few functions, but their average ranking is the lowest. To sum up, first, compared with the other eight competitors, MSPSO indicates the best overall performance on all 30 functions in CEC 2014 with 10 variables in terms of the number of the best and average ranking. Second, MSPSO is significantly different from other algorithms in most of the functions.

The simulation results on 30 functions with 30 variables in CEC 2014 are shown in Table 4. The results show that MSPSO outperformed the other eight algorithms on functions F2, F3, F4, F5, F6, F7, F11, F15, F17, F20, and F27. For function F26, the mean error of MSPSO was equal to other optimal algorithms. Specifically, for unimodal functions, MSPSO generally performs better than most of the other algorithms. Also, it is superior or equal to the other eight algorithms on all functions except F1. Furthermore, for simple multimodal functions, MSPSO exhibits a better performance. Also, it is superior to EPSO in eight functions. Regarding hybrid functions, MSPSO shows a better performance compared with other comparative PSO variants. Also, it ranked within the top 3 on all functions. As for

Table 3: Computational result of benchmark functions in CEC 2014 with 10 variables.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F1	Mean	1.96E+04	**5.37E+02**	3.95E+04	1.25E+05	3.60E+04	2.79E+03	3.84E+06	7.86E+03	6.59E+03
	Std.	1.33E+04	4.83E+02	3.92E+04	2.08E+05	3.29E+04	2.48E+03	2.16E+06	6.56E+03	4.78E+03
	Rank	5 (+)	1 (−)	7 (+)	8 (+)	6 (+)	2 (−)	9 (+)	4 (+)	3
F2	Mean	3.06E+03	9.91E+02	2.51E+02	2.29E+03	4.40E+02	4.57E+02	3.84E+08	2.04E+02	**6.21E+00**
	Std.	1.77E+03	4.93E+02	3.10E+02	2.26E+03	7.18E+02	1.83E+02	4.63E+07	2.21E+02	8.55E+00
	Rank	8 (+)	6 (+)	3 (+)	7 (+)	4 (+)	5 (+)	9 (+)	2 (+)	1
F3	Mean	6.67E+02	2.26E+02	5.22E+02	7.37E+02	1.86E+02	3.04E+02	2.34E+03	2.69E+01	**2.57E+00**
	Std.	4.30E+02	1.74E+02	8.57E+02	7.51E+02	7.21E+01	2.61E+02	3.12E+02	4.42E+01	1.20E+01
	Rank	7 (+)	4 (+)	6 (+)	8 (+)	3 (+)	5 (+)	9 (+)	2 (=)	1
F4	Mean	3.74E+01	1.16E+01	1.38E+00	3.16E+01	**3.00E−02**	2.42E+01	4.22E+01	5.58E−02	1.23E+00
	Std.	2.81E−01	2.01E+01	2.98E−01	2.74E+01	5.61E−03	2.10E+01	1.10E+00	5.58E−02	1.43E+00
	Rank	8 (+)	5 (+)	4 (+)	7 (+)	1 (−)	6 (+)	9 (+)	2 (−)	3
F5	Mean	2.02E+01	2.02E+01	2.00E+01	2.00E+01	2.01E+01	2.02E+01	2.04E+01	2.01E+01	**1.33E+01**
	Std.	8.16E−02	1.42E−01	3.96E−06	1.30E−03	1.36E−02	4.26E−02	2.63E−02	4.18E−02	1.15E+01
	Rank	6 (+)	6 (+)	2 (−)	2 (−)	4 (+)	6 (+)	9 (+)	4 (+)	1
F6	Mean	1.30E+00	2.29E+00	2.24E+00	1.61E+00	1.05E−01	1.95E+00	6.10E+00	**9.10E−02**	2.50E−01
	Std.	1.11E+00	2.63E+00	8.34E−01	7.65E−02	2.75E−02	1.60E+00	1.39E+00	2.45E−02	2.47E−01
	Rank	4 (+)	8 (+)	7 (+)	5 (+)	2(−)	6 (+)	9 (+)	1 (−)	3
F7	Mean	8.61E−02	1.51E−01	3.40E−01	**1.15E−02**	5.08E−02	1.68E−01	8.48E+00	6.27E−02	5.90E−02
	Std.	3.49E−02	1.14E−01	6.29E−02	1.24E−02	1.65E−02	7.28E−02	1.42E+00	3.94E−02	1.61E−02
	Rank	5 (+)	6 (+)	8 (+)	1 (−)	2 (−)	7 (+)	9 (+)	4 (+)	3
F8	Mean	3.32E+00	3.65E+00	3.32E−01	9.62E+00	**0.00E+00**	5.97E+00	3.12E+01	**0.00E+00**	**0.00E+00**
	Std.	2.07E+00	2.50E+00	5.74E−01	2.07E+00	0.00E+00	3.45E+00	4.75E+00	0.00E+00	0.00E+00
	Rank	5 (+)	6 (+)	4 (+)	8 (+)	1 (=)	7 (+)	9 (+)	1 (=)	1
F9	Mean	5.97E+00	1.03E+01	1.56E+01	1.39E+01	4.04E+00	9.95E+00	4.21E+01	4.76E+00	**3.31E+00**
	Std.	9.95E−01	4.60E+00	1.10E+01	5.26E+00	9.93E−01	6.52E+00	9.11E−01	3.72E−01	1.14E+00
	Rank	4 (+)	6 (+)	8 (+)	7 (+)	2 (+)	5 (+)	9 (+)	3 (+)	1
F10	Mean	6.24E+01	1.31E+02	2.08E+02	4.99E+02	**0.00E+00**	2.68E+02	7.66E+02	2.32E−01	1.24E−01
	Std.	7.38E+01	1.23E+02	5.85E+01	1.38E+02	0.00E+00	1.24E+01	2.06E+02	7.09E−02	1.05E−12
	Rank	4 (+)	5 (+)	6 (+)	8 (+)	1 (−)	7 (+)	9 (+)	3 (=)	2
F11	Mean	2.86E+02	5.05E+02	1.49E+02	5.36E+02	**9.18E+01**	7.72E+02	7.40E+02	9.60E+01	1.80E+02
	Std.	1.18E+02	1.37E+02	2.34E+02	1.58E+02	6.63E+01	9.76E+01	9.10E+01	1.34E+02	1.18E+02
	Rank	5 (+)	6 (+)	3 (=)	7 (+)	1 (−)	9 (+)	8 (+)	2 (−)	4
F12	Mean	1.21E−01	1.02E−01	2.08E−01	**7.06E−02**	1.70E−01	4.34E−01	1.03E+00	2.14E−01	7.70E−02
	Std.	1.33E−02	6.71E−02	1.08E−01	6.77E−02	1.54E−02	7.11E−02	2.00E−01	1.17E−01	6.42E−02
	Rank	4 (+)	3 (+)	6 (+)	1 (−)	5 (+)	8 (+)	9 (+)	7 (+)	2
F13	Mean	1.80E−01	9.31E−02	4.14E−01	9.39E−02	1.24E−01	1.71E−01	5.39E−01	9.94E−02	**5.92E−02**
	Std.	1.03E−01	2.51E−02	2.35E−01	3.28E−02	2.78E−02	2.27E−02	3.97E−02	1.48E−02	3.89E−02
	Rank	7 (+)	2 (+)	8 (+)	3 (+)	5 (+)	6 (+)	9 (+)	4 (+)	1

Table 3: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F14	Mean	1.22E − 01	1.06E − 01	2.43E − 01	3.77E − 01	1.51E − 01	2.11E − 01	6.32E − 01	**8.31E − 02**	8.86E − 02
	Std.	2.79E − 02	7.31E − 02	9.40E − 02	9.76E − 02	6.60E − 02	1.37E − 01	1.75E − 01	7.21E − 02	6.66E − 02
	Rank	4 (+)	2 (=)	7 (+)	8 (+)	5 (+)	6 (+)	9 (+)	1 (−)	2
F15	Mean	6.96E − 01	9.91E − 01	2.14E + 00	6.20E − 01	**5.85E − 01**	8.12E − 01	6.07E + 00	7.13E − 01	7.01E − 01
	Std.	1.19E − 01	1.79E − 01	8.23E − 01	3.34E − 01	6.62E − 02	2.77E − 01	1.56E − 01	3.53E − 01	1.37E − 01
	Rank	3 (−)	7 (+)	8 (+)	2 (−)	1 (−)	6 (+)	9 (+)	5 (+)	4
F16	Mean	1.95E + 00	2.72E + 00	2.57E + 00	2.85E + 00	1.84E + 00	1.93E + 00	3.22E + 00	**1.42E + 00**	1.59E + 00
	Std.	2.79E − 01	4.14E − 01	4.70E − 01	4.06E − 01	1.76E − 01	4.50E − 01	1.56E − 01	2.08E − 01	4.22E − 01
	Rank	5 (+)	7 (+)	6 (+)	8 (+)	3 (+)	4 (+)	9 (+)	1 (−)	2
F17	Mean	3.66E + 03	2.97E + 03	2.01E + 03	4.29E + 04	6.04E + 03	4.29E + 03	8.11E + 03	2.86E + 03	**1.18E + 03**
	Std.	2.87E + 03	2.96E + 03	1.91E + 03	1.92E + 04	4.31E + 03	1.75E + 03	1.68E + 03	3.31E + 03	1.03E + 03
	Rank	5 (+)	4 (+)	2 (+)	9 (+)	7 (+)	6 (+)	8 (+)	3 (+)	1
F18	Mean	1.02E + 02	1.96E + 03	5.24E + 03	3.26E + 03	**3.09E + 01**	3.24E + 03	9.19E + 03	6.08E + 02	5.06E + 02
	Std.	6.78E + 01	1.63E + 03	5.39E + 03	1.63E + 03	2.39E + 01	3.87E + 03	1.21E + 03	4.26E + 02	4.35E + 03
	Rank	2 (−)	5 (+)	8 (+)	7 (+)	1 (−)	6 (+)	9 (+)	4 (+)	3
F19	Mean	1.07E + 00	1.51E + 00	1.71E + 00	2.66E + 00	4.34E − 01	2.71E + 00	3.96E + 00	**1.09E − 01**	5.32E − 01
	Std.	8.56E − 01	8.67E − 01	1.20E + 00	1.08E + 00	5.39E − 01	1.59E + 00	4.41E − 01	6.57E − 02	4.44E − 01
	Rank	4 (+)	5 (+)	6 (+)	7 (+)	2 (−)	8 (+)	9 (+)	1 (−)	3
F20	Mean	7.26E + 01	3.10E + 03	1.75E + 03	8.52E + 03	3.16E + 01	4.60E + 01	8.79E + 02	**1.38E + 01**	3.59E + 01
	Std.	2.01E + 01	4.25E + 03	2.49E + 03	1.27E + 04	4.11E + 01	1.81E + 01	7.50E + 01	4.36E + 00	1.62E + 01
	Rank	5 (+)	8 (+)	7 (+)	9 (+)	2 (−)	4 (+)	6 (+)	1 (−)	3
F21	Mean	1.05E + 02	7.95E + 01	1.84E + 03	3.17E + 03	1.30E + 02	9.02E + 01	5.67E + 03	3.08E + 01	**2.87E + 01**
	Std.	7.52E + 01	1.36E + 02	1.22E + 03	4.48E + 03	1.12E + 02	3.28E + 01	4.18E + 03	3.67E + 01	3.56E + 01
	Rank	5 (+)	3 (+)	7 (+)	8 (+)	6 (+)	4 (+)	9 (+)	2 (+)	1
F22	Mean	4.86E + 01	5.48E + 01	1.27E + 02	8.51E + 01	**3.69E − 01**	2.09E + 01	5.27E + 01	1.72E + 00	4.86E − 01
	Std.	6.67E + 01	7.74E + 01	4.40E + 01	1.08E + 02	4.55E − 01	4.77E − 01	8.41E + 00	1.19E + 00	3.30E − 01
	Rank	5 (+)	7 (+)	9 (+)	8 (+)	1 (−)	4 (+)	6 (+)	3 (+)	2
F23	Mean	3.29E + 02	3.29E + 02	3.29E + 02	3.29E + 02	3.29E + 02	3.29E + 02	**2.92E + 02**	3.29E + 02	3.29E + 02
	Std.	0.00E + 00	2.64E − 13	1.89E − 12	2.08E − 12	2.64E − 13	2.64E − 13	7.94E + 01	2.64E − 13	2.64E − 13
	Rank	2 (=)	2 (=)	2 (=)	2 (=)	2 (=)	2 (=)	1 (−)	2 (=)	2
F24	Mean	1.22E + 02	1.18E + 02	1.55E + 02	1.39E + 02	1.13E + 02	1.23E + 02	1.82E + 02	1.11E + 02	**1.10E + 02**
	Std.	1.96E + 00	5.08E + 00	4.32E + 01	2.19E + 01	3.32E + 00	3.42E + 00	3.17E + 01	1.00E + 01	1.43E + 00
	Rank	5 (+)	4 (+)	8 (+)	7 (+)	3 (+)	6 (+)	9 (+)	2 (+)	1
F25	Mean	1.78E + 02	1.48E + 02	1.83E + 02	1.75E + 02	1.82E + 02	1.76E + 02	2.00E + 02	1.54E + 02	**1.27E + 02**
	Std.	3.76E + 01	4.11E + 01	2.92E + 01	3.96E + 01	4.17E + 00	4.59E + 01	0.00E + 00	4.04E + 01	4.63E + 00
	Rank	6 (+)	2 (+)	8 (+)	4 (+)	7 (+)	5 (+)	9 (+)	3 (+)	1
F26	Mean	**1.00E + 02**	1.33E + 02	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**
	Std.	2.13E − 02	5.77E + 01	1.32E − 01	5.34E − 02	3.45E − 02	1.15E − 02	7.89E − 02	1.99E − 02	2.93E − 02
	Rank	1 (+)	9 (+)	1 (+)	1 (=)	1 (+)	1 (+)	1 (+)	1 (+)	1

Table 3: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F27	Mean	3.41E + 02	2.36E + 02	2.75E + 02	3.59E + 02	1.13E + 02	3.73E + 00	1.47E + 02	1.12E + 02	**2.87E + 00**
	Std.	5.15E + 01	2.08E + 02	2.35E + 02	3.60E + 01	1.86E + 02	1.51E + 00	2.24E + 02	1.90E + 02	1.85E + 00
	Rank	9 (+)	6 (+)	7 (+)	8 (+)	4 (+)	2 (+)	5 (+)	3 (+)	1
F28	Mean	4.59E + 02	5.24E + 02	**3.85E + 02**	5.44E + 02	**3.85E + 02**	4.09E + 02	3.96E + 02	3.92E + 02	3.88E + 02
	Std.	7.56E + 01	1.03E + 02	1.37E + 02	5.80E + 01	4.73E + 01	7.60E + 01	2.97E + 00	3.44E + 01	2.28E + 01
	Rank	7 (+)	8 (+)	1 (−)	9 (+)	1 (−)	6 (+)	5 (+)	4 (+)	3
F29	Mean	7.14E + 02	3.41E + 02	**2.20E + 02**	8.06E + 02	2.83E + 02	6.48E + 02	6.83E + 05	2.51E + 02	2.68E + 02
	Std.	6.66E + 02	6.05E + 01	1.87E + 01	3.82E + 02	4.96E + 01	8.79E + 01	1.18E + 06	8.97E + 00	1.00E + 01
	Rank	7 (+)	5 (+)	1 (−)	8 (+)	4 (+)	6 (+)	9 (+)	3 (+)	2
F30	Mean	6.17E + 02	1.39E + 03	9.52E + 02	1.52E + 03	6.10E + 02	7.84E + 02	9.24E + 02	6.87E + 02	**6.01E + 02**
	Std.	7.36E + 01	3.32E + 02	9.20E + 02	8.71E + 01	1.14E + 02	1.14E + 02	3.29E + 02	7.65E + 01	3.35E + 01
	Rank	3 (+)	8 (+)	7 (+)	9 (+)	2 (+)	5 (+)	6 (+)	4 (+)	1
Best/2nd best/worst ranking		1/2/0	1/4/0	3/3/0	3/3/4	9/7/0	1/3/1	2/0/21	7/7/0	13/7/0
Avg. ranking		5	5.2	5.57	6.2	2.97	5.33	7.83	2.73	1.98
+/=/−		27/1/2	27/2/1	24/3/3	24/2/4	17/2/11	28/1/1	29/0/1	19/4/7	
Algorithms		PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO

Bold values represent the best results of each function.

Table 4: Computational result of benchmark functions in CEC 2014 with 30 variables.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F1	Mean	8.17E+06	9.82E+04	4.47E+05	2.78E+07	5.33E+06	1.53E+07	1.35E+08	**7.82E+04**	1.36E+05
	Std.	7.53E+06	5.80E+04	3.46E+05	3.62E+07	2.23E+06	3.78E+06	3.40E+07	7.51E+03	1.00E+05
	Rank	6 (+)	2 (−)	4 (+)	8 (+)	5 (+)	7 (+)	9 (+)	1 (−)	3
F2	Mean	1.80E+01	4.43E+02	2.37E+03	4.99E+03	1.72E+02	6.98E+01	9.61E+09	**2.86E−01**	4.73E+00
	Std.	3.00E+01	7.63E+02	1.40E+03	8.23E+03	1.82E+02	1.08E+02	2.11E+09	3.09E−01	7.48E+00
	Rank	3 (+)	6 (+)	7 (+)	8 (+)	5 (+)	4 (+)	9 (+)	1 (−)	2
F3	Mean	3.89E+01	4.14E+01	3.83E+02	8.42E+03	1.67E+02	3.70E+01	3.18E+04	8.36E−02	**6.07E−02**
	Std.	3.88E+01	6.44E+01	1.79E+02	1.11E+04	4.40E+01	1.29E+01	3.34E+03	7.31E−02	6.59E−02
	Rank	4 (+)	5 (+)	7 (+)	8 (+)	6 (+)	3 (+)	9 (+)	2 (+)	1
F4	Mean	1.56E+02	2.24E+01	3.37E+01	2.64E+02	6.36E+01	1.53E+02	6.14E+02	4.93E−01	**3.61E−01**
	Std.	5.06E+01	3.76E+01	3.64E+01	1.43E+02	3.18E+01	2.26E+01	1.01E+02	7.10E−02	1.94E−01
	Rank	7 (+)	3 (+)	4 (+)	8 (+)	5 (+)	6 (+)	9 (+)	2 (+)	1
F5	Mean	2.09E+01	2.04E+01	**2.00E+01**	**2.00E+01**	2.04E+01	2.08E+01	2.09E+01	2.03E+01	**2.00E+01**
	Std.	1.24E−01	3.92E−01	8.20E−06	1.91E−04	6.69E−02	1.20E−01	9.54E−02	1.32E−02	1.25E−05
	Rank	8 (+)	5 (+)	1 (−)	1 (−)	5 (+)	7 (+)	8 (+)	4 (+)	1
F6	Mean	1.28E+01	7.56E+00	1.94E+01	1.74E+01	1.23E+01	1.87E+01	2.89E+01	9.00E+00	**6.89E+00**
	Std.	3.46E+00	2.86E+00	2.27E+00	3.22E+00	7.72E−01	3.57E+00	1.07E+00	2.94E−01	1.74E+00
	Rank	5 (+)	2 (+)	8 (+)	6 (+)	4 (+)	7 (+)	9 (+)	3 (+)	1
F7	Mean	2.21E−02	1.97E−02	9.01E−03	3.29E−03	4.98E−07	1.23E−02	8.47E+01	**9.79E−09**	1.54E−07
	Std.	1.95E−02	1.91E−02	1.56E−02	5.69E−03	2.09E−07	2.46E−03	8.05E+00	1.70E−08	2.68E−07
	Rank	8 (+)	7 (+)	5 (+)	4 (+)	3 (+)	6 (+)	9 (+)	1 (−)	2
F8	Mean	2.35E+01	3.71E+01	9.62E+00	5.64E+01	**1.14E−13**	6.90E+01	2.23E+02	2.27E−13	3.33E−12
	Std.	2.07E+00	1.38E+01	3.20E+00	1.65E+01	0.00E+00	3.51E+01	7.48E+00	0.00E+00	2.36E−12
	Rank	5 (+)	6 (+)	4 (+)	7 (+)	1 (−)	8 (+)	9 (+)	2 (−)	3
F9	Mean	7.10E+01	5.84E+01	9.72E+01	6.57E+01	4.78E+01	9.75E+01	2.55E+02	4.90E+01	**3.97E+01**
	Std.	4.49E+00	1.96E+01	6.53E+01	1.43E+01	5.63E+00	1.15E+01	7.61E+00	6.55E+00	3.58E+00
	Rank	6 (+)	4 (+)	7 (+)	5 (+)	2 (+)	8 (+)	9 (+)	3 (+)	1
F10	Mean	5.60E+02	7.87E+02	3.67E+02	1.94E+03	**7.11E−01**	1.67E+03	5.53E+03	5.84E+00	4.00E+00
	Std.	3.28E+02	2.87E+02	2.39E+02	2.99E+02	2.14E−01	5.97E+02	5.05E+02	2.84E+00	6.51E−01
	Rank	5 (+)	6 (+)	4 (+)	8 (+)	1 (−)	7 (+)	9 (+)	3 (+)	2
F11	Mean	2.48E+03	2.82E+03	3.08E+03	2.69E+03	2.22E+03	3.47E+03	6.93E+03	**2.02E+03**	2.35E+03
	Std.	7.18E+02	8.01E+02	6.64E+02	3.29E+02	2.92E+02	6.89E+02	9.08E+01	1.78E+02	1.25E+02
	Rank	4 (+)	6 (+)	7 (+)	5 (+)	2 (−)	8 (+)	9 (+)	1 (−)	3
F12	Mean	1.66E+00	7.75E−01	2.59E−01	1.55E−01	4.27E−01	8.86E−01	2.38E+00	1.94E−01	**1.40E−01**
	Std.	4.46E−01	6.84E−01	8.74E−02	5.77E−02	1.36E−01	6.91E−01	1.62E−01	1.94E−01	2.57E−02
	Rank	8 (+)	6 (+)	4 (+)	2 (=)	5 (+)	7 (+)	9 (+)	3 (+)	1
F13	Mean	5.09E−01	2.67E−01	4.12E−01	2.62E−01	3.34E−01	4.12E−01	2.08E+00	2.75E−01	**1.81E−01**
	Std.	1.05E−01	8.41E−02	7.90E−02	2.27E−02	3.69E−02	2.87E−02	2.91E−01	3.02E−02	6.19E−02
	Rank	8 (+)	3 (−)	6 (+)	2 (+)	5 (+)	6 (+)	9 (+)	4 (+)	1

Table 4: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F14	Mean	2.91E − 01	2.25E − 01	3.07E − 01	2.67E − 01	2.51E − 01	3.21E − 01	2.54E + 01	**2.19E − 01**	2.36E − 01
	Std.	4.78E − 02	2.41E − 02	4.75E − 02	1.47E − 02	9.50E − 03	4.43E − 02	2.63E + 00	3.79E − 02	1.55E − 02
	Rank	6 (+)	2 (−)	7 (+)	5 (+)	4 (+)	8 (+)	9 (+)	1 (−)	3
F15	Mean	7.08E + 00	4.61E + 00	3.64E + 01	7.49E + 00	7.43E + 00	1.12E + 01	1.50E + 03	4.39E + 00	**2.67E + 00**
	Std.	4.15E + 00	9.58E − 01	8.33E + 00	3.07E + 00	1.09E + 00	7.51E − 01	9.12E + 02	7.15E − 01	6.67E − 01
	Rank	4 (+)	3 (+)	8 (+)	6 (+)	5 (+)	7 (+)	9 (+)	2 (+)	1
F16	Mean	1.10E + 01	9.92E + 00	1.03E + 01	1.17E + 01	1.03E + 01	1.07E + 01	1.27E + 01	9.84E + 00	**8.95E + 00**
	Std.	1.23E + 00	1.10E − 01	5.82E − 01	2.26E − 01	4.98E − 02	1.37E + 00	1.04E − 01	3.80E − 01	7.44E − 01
	Rank	7 (+)	3(+)	4 (+)	8 (+)	4 (+)	6 (+)	9 (+)	2 (+)	1
F17	Mean	7.03E + 05	3.82E + 04	9.13E + 04	2.50E + 05	6.83E + 05	8.62E + 04	4.10E + 06	4.08E + 04	**2.05E + 04**
	Std.	3.10E + 05	1.81E + 04	7.73E + 04	9.41E + 04	1.41E + 05	6.02E + 04	3.89E + 06	2.18E + 04	2.41E + 03
	Rank	8 (+)	2 (+)	5 (+)	6 (+)	7 (+)	4 (+)	9 (+)	3 (+)	1
F18	Mean	2.22E + 02	3.07E + 03	2.41E + 02	5.83E + 02	**8.79E + 01**	1.90E + 03	1.06E + 08	2.71E + 02	1.01E + 02
	Std.	1.18E + 02	1.60E + 03	1.16E + 02	4.58E + 02	1.71E + 01	5.34E + 02	3.87E + 07	1.55E + 02	9.15E + 01
	Rank	3 (+)	8 (+)	4 (+)	6 (+)	1 (−)	7 (+)	9 (+)	5 (+)	2
F19	Mean	8.06E + 00	5.22E + 00	1.65E + 01	3.29E + 01	6.97E + 00	1.23E + 01	7.17E + 01	7.30E + 00	**4.68E + 00**
	Std.	3.38E + 00	1.74E + 00	2.48E + 00	4.20E + 01	6.06E − 01	1.92E + 00	1.20E + 01	1.63E + 00	7.77E − 01
	Rank	5 (+)	2 (+)	7 (+)	8 (+)	3 (+)	6 (+)	9 (+)	4 (+)	1
F20	Mean	3.10E + 02	5.79E + 02	1.76E + 03	8.36E + 03	2.05E + 03	1.14E + 03	1.52E + 04	2.22E + 02	**1.63E + 02**
	Std.	3.58E + 01	2.53E + 02	9.85E + 02	4.39E + 03	5.18E + 02	9.09E + 02	7.81E + 03	1.14E + 02	7.36E + 01
	Rank	3 (+)	4 (+)	6 (+)	8 (+)	7 (+)	5 (+)	9 (+)	2 (+)	1
F21	Mean	3.35E + 04	2.77E + 04	1.54E + 04	3.10E + 05	3.30E + 04	**1.25E + 04**	1.25E + 06	1.58E + 04	1.49E + 04
	Std.	1.47E + 04	9.11E + 03	1.21E + 04	3.96E + 05	1.06E + 04	1.26E + 04	2.35E + 05	1.24E + 04	1.13E + 04
	Rank	7 (+)	5 (+)	3 (=)	8 (+)	6 (+)	1 (−)	9 (+)	4 (+)	2
F22	Mean	3.55E + 02	2.82E + 02	4.55E + 02	3.20E + 02	**7.27E + 01**	5.36E + 02	7.18E + 02	2.40E + 02	1.54E + 02
	Std.	2.50E + 02	1.26E + 02	1.47E + 02	1.09E + 02	6.20E + 01	8.25E + 01	1.34E + 02	1.32E + 02	1.80E + 01
	Rank	6 (+)	4 (+)	7 (+)	5 (+)	1 (−)	8 (+)	9 (+)	3 (+)	2
F23	Mean	3.16E + 02	3.15E + 02	3.14E + 02	3.24E + 02	3.15E + 02	3.16E + 02	**3.08E + 02**	3.15E + 02	3.15E + 02
	Std.	1.25E − 01	2.36E − 12	2.30E − 11	8.89E + 00	2.01E − 09	2.20E − 01	9.45E + 01	2.24E − 12	9.46E − 13
	Rank	7 (+)	3 (=)	2 (−)	9 (+)	3 (=)	7 (+)	1 (−)	3 (=)	3
F24	Mean	2.34E + 02	2.24E + 02	2.30E + 02	2.39E + 02	2.24E + 02	2.25E + 02	**2.00E + 02**	2.26E + 02	2.25E + 02
	Std.	1.27E + 01	1.91E + 00	1.49E + 00	7.90E + 00	3.46E − 01	8.14E − 01	3.52E − 07	3.21E − 01	7.88E − 01
	Rank	8 (+)	2 (−)	7 (+)	9 (+)	2 (−)	4 (=)	1 (−)	6 (=)	5
F25	Mean	2.08E + 02	2.09E + 02	2.18E + 02	2.14E + 02	2.08E + 02	2.15E + 02	**2.00E + 02**	2.07E + 02	2.05E + 02
	Std.	1.91E + 00	3.98E − 01	4.81E + 00	1.16E + 00	9.54E − 01	4.31E + 00	0.00E + 00	9.08E − 01	1.29E + 00
	Rank	4 (+)	6 (+)	9 (+)	7 (+)	4 (+)	8 (+)	1 (−)	3 (+)	2
F26	Mean	1.01E + 02	1.67E + 02	1.34E + 02	1.34E + 02	**1.00E + 02**	**1.00E + 02**	1.67E + 02	**1.00E + 02**	**1.00E + 02**
	Std.	8.25E − 02	5.76E + 01	5.74E + 01	5.76E + 01	1.05E − 01	5.60E − 02	5.66E + 01	1.45E − 01	4.39E − 03
	Rank	5 (+)	8 (+)	6 (+)	6 (+)	1 (+)	1 (+)	8 (+)	1 (+)	1

Table 4: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F27	Mean	6.72*E* + 02	4.89*E* + 02	6.74*E* + 02	5.30*E* + 02	4.11*E* + 02	8.68*E* + 02	9.09*E* + 02	**4.03E + 02**	**4.03E + 02**
	Std.	4.43*E* + 01	9.15*E* + 01	2.36*E* + 02	2.13*E* + 02	1.32*E* + 00	8.05*E* + 01	3.24*E* + 02	1.56*E* + 00	1.87*E* + 00
	Rank	6 (+)	4 (+)	7 (+)	5 (+)	3 (+)	8 (+)	9 (+)	1 (=)	1
F28	Mean	1.00*E* + 03	1.34*E* + 03	1.34*E* + 03	1.62*E* + 03	**8.67E + 02**	1.38*E* + 03	1.12*E* + 03	9.79*E* + 02	8.71*E* + 02
	Std.	6.73*E* + 01	7.40*E* + 02	7.96*E* + 02	1.60*E* + 02	3.56*E* + 01	3.24*E* + 02	8.97*E* + 01	2.11*E* + 01	4.58*E* + 01
	Rank	4 (+)	6 (+)	6 (+)	9 (+)	1 (−)	8 (+)	5 (+)	3 (+)	2
F29	Mean	8.15*E* + 06	1.15*E* + 03	**2.16E + 02**	1.27*E* + 03	1.01*E* + 03	5.40*E* + 06	3.65*E* + 05	1.07*E* + 03	9.81*E* + 02
	Std.	1.41*E* + 07	1.87*E* + 02	2.92*E* + 00	9.06*E* + 01	1.11*E* + 02	9.35*E* + 06	2.14*E* + 05	1.13*E* + 02	5.81*E* + 01
	Rank	9 (+)	5 (+)	1 (−)	6 (+)	3 (+)	8 (+)	7 (+)	4 (+)	2
F30	Mean	2.24*E* + 03	1.74*E* + 03	**8.95E + 02**	3.35*E* + 04	2.61*E* + 03	6.61*E* + 03	3.21*E* + 04	2.20*E* + 03	1.98*E* + 03
	Std.	2.34*E* + 02	6.64*E* + 02	4.21*E* + 02	2.02*E* + 04	2.20*E* + 02	3.92*E* + 03	1.49*E* + 03	2.22*E* + 02	5.36*E* + 02
	Rank	5 (+)	2 (+)	1 (−)	9 (+)	6 (+)	7 (+)	8 (+)	4 (+)	3
Best/2nd best/worst ranking		0/0/0	1/8/0	3/1/1	1/2/4	6/3/0	2/0/0	3/0/22	7/6/0	14/9/0
Avg. ranking		5.8	4.33	5.27	6.4	3.67	6.23	7.9	2.7	1.83
+/=/−		30/0/0	25/1/4	25/1/4	28/1/1	22/1/7	28/1/1	27/0/3	21/3/6	
Algorithms		PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO

Bold values represent the best results of each function.

composition functions, MSPSO shows the best performance on two functions and a moderate performance compared with other comparative algorithms on other functions. For other algorithms, EPSO and CLPSO also perform well, ranking second and third, respectively. The rest of these algorithms win on no more than 3 functions. The average ranking of A-PSO is the lowest. In summary, first, MSPSO also has the best overall performance compared with the other eight competitors on all functions in CEC 2014 with 30 variables. Second, MSPSO is also significantly different from other algorithms in most of the functions.

From Tables 3 and 4, we see that the performance of two ensemble PSO algorithms (EPSO and MSPSO) is better than other types of PSO algorithms. This result can be explained by the theory of easy and hard fitness functions [17]. According to that theory, the hardest problem to one evolutionary algorithm could be the easiest to another algorithm. Thus, given an ensemble of different PSO algorithms, a hard problem might be solved easily by one of them. Of course, if a problem is hard (or easy) to all of them, using an ensemble does not bring too much improvement.

The convergent speed is evaluated in Figures 2–8. From Figure 2, we can see that MSPSO obtains a better performance than other PSO variants. The convergent speed of MSPSO is not fast, but its optimization accuracy is higher than other competitors in many functions. This is because, in the early stage of the search, different particle generation strategies may interfere with the direction in which particles quickly find good positions. However, in the later stage of the search, different particle generation strategies increase the chances of particles finding good positions.

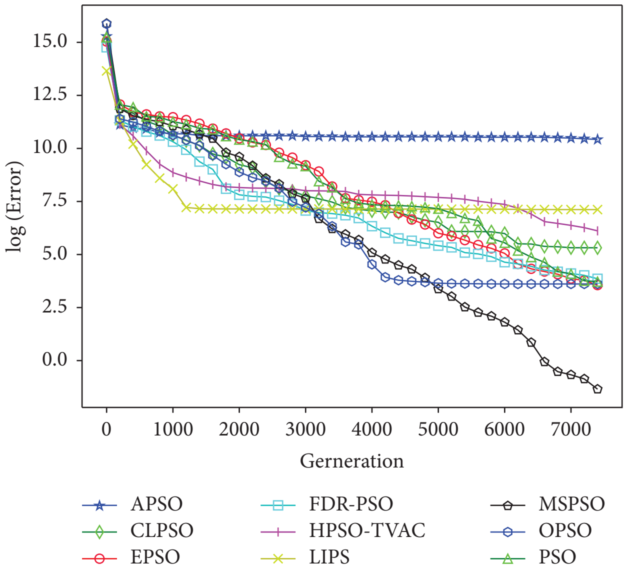


Figure 2: The results on F3 with 30 variables.

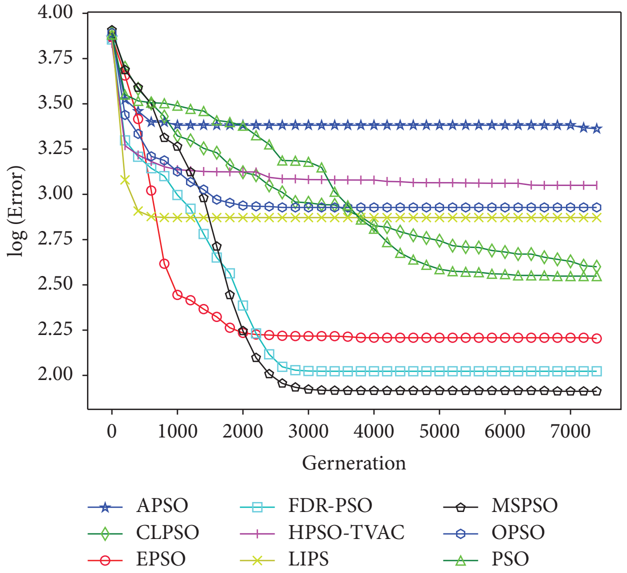


Figure 3: The results on F6 with 30 variables.

4.2. Application to the Hydrothermal Scheduling Problem. The hydrothermal scheduling problem [31] is a complex optimization problem from the real world. Its main objective is to schedule the power generations of the thermal and hydro units in the system to meet the load demands, under the premise of satisfying the constraints of the hydraulic systems and the power system networks. In order to evaluate the performance of hydrothermal scheduling problem in dealing with real-world problems, we apply MSPSO to solving this problem. In the hydrothermal scheduling problem, decision variables are nonlinearly related to the major operation problem of hydrothermal systems. The objective of the problem is to minimize the fuel cost of thermal units for 24 hours with four hydro units in the system, and the dimension of the problem is 96.

In order to meet load requirements during the scheduling period, the total fuel cost of the thermal system operation is expressed by F. The objective function is given as follows:

$$\text{Minimize } F = \sum_{i=1}^{M} f_i\left(P_{Ti}\right). \tag{9}$$

In the previous formula, P_{Ti} is the power generation of an equivalent thermal unit at ith interval, and f_i represents the cost function corresponding to P_{Ti}. M is the total number of intervals considered for the short-term planning. The cost function f_i is expressed as follows:

$$f_i\left(P_{Ti}\right) = a_i P_{Ti}^2 + b_i P_{Ti} + c_i + \left|e_i \sin\left(f_i\left(P_{Ti}^{\min} - P_{Ti}\right)\right)\right|. \tag{10}$$

MSPSO is compared with five other algorithms in three hydrothermal scheduling instances, which are CoBiDE [32], TLBO [33], ALC-PSO [34], DNS-PSO [35], and EPSO [27]. CoBiDE incorporates the covariance matrix learning and the bimodal distribution parameter setting into DE. TLBO designs an optimization mechanism inspired by the effect of the influence of a teacher on learners. TLBO divides the optimization process into "Teacher Phase" and "Learner Phase." ALC-PSO transplants the aging mechanism to PSO

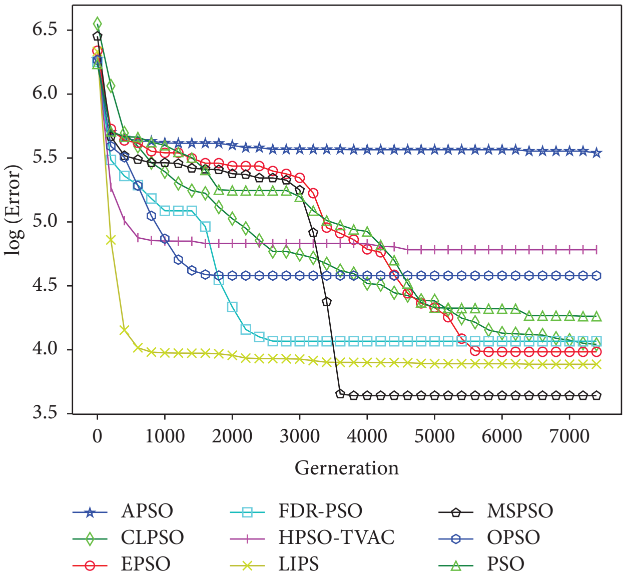


Figure 4: The results on F9 with 30 variables.

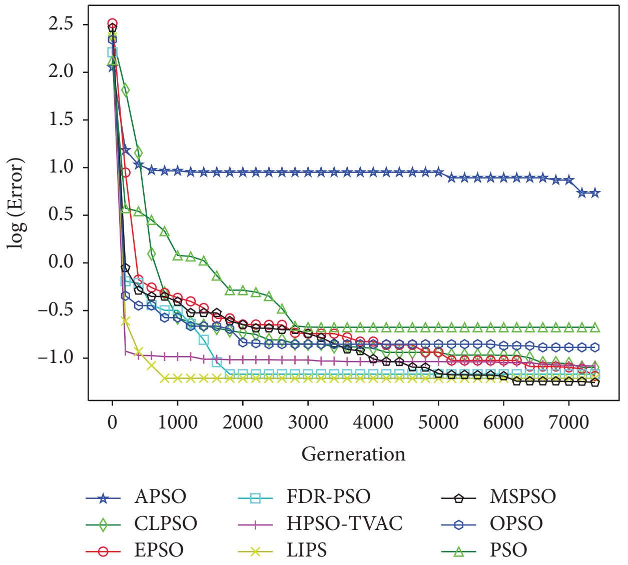


Figure 5: The results on F13 with 30 variables.

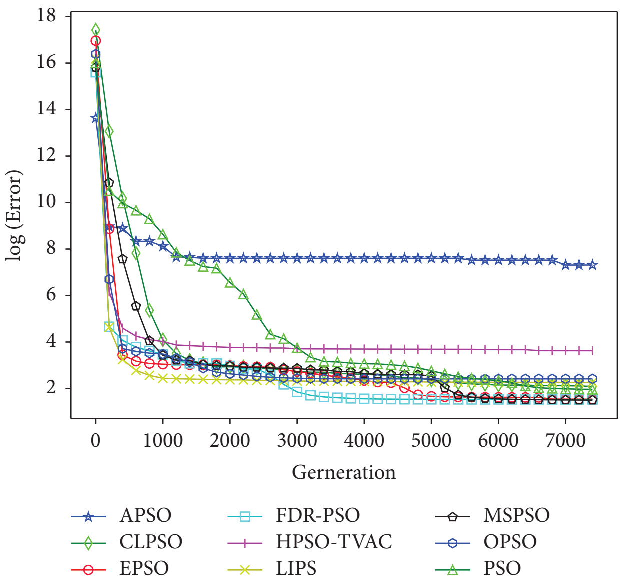


Figure 6: The results on F15 with 30 variables.

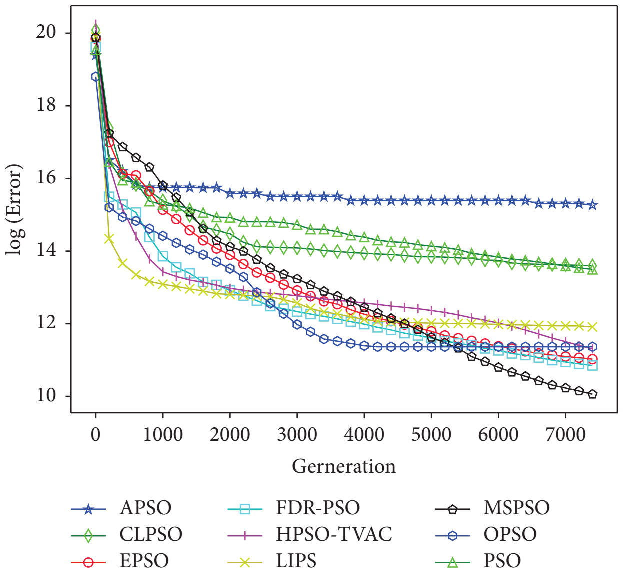


Figure 7: The results on F17 with 30 variables.

to overcome the problem of premature convergence. DNS-PSO employs a diversity-enhancing mechanism and neighborhood search strategies in PSO to achieve a trade-off between exploration and exploitation abilities.

The computational results of hydrothermal scheduling instances are shown in Table 5. From the table, MSPSO outperformed the other five comparative algorithms in two instances. This means that MSPSO is a good alternative algorithm for solving the hydrothermal scheduling problem.

4.3. The IEEE CEC 2010 Standard Test Functions Set. In order to further analyze the performance of MSPSO to solve the large-scale global optimization problem, CEC 2010 [36] is employed in experiments. The performance of MSPSO is compared with PSO [2], grey wolf optimizer (GWO) [37], standard sine cosine algorithm (SCA) [38], and slap swarm algorithm (SSA) [39]. All experiments are tested 30 times in 1000 dimensions. The mean and standard deviation of all algorithms are shown in Table 6. The average rank and rank are also recorded in the last two rows of Table 6. From Table 6, it shows that MSPSO has outperformance than other comparative algorithms to solve the large-scale global optimization problems. For most CEC 2010 functions, the MSPSO improves the accuracy by some orders of magnitudes. Therefore, the experimental results demonstrate that MSPSO has a good performance in solving the large-scale optimization problems.

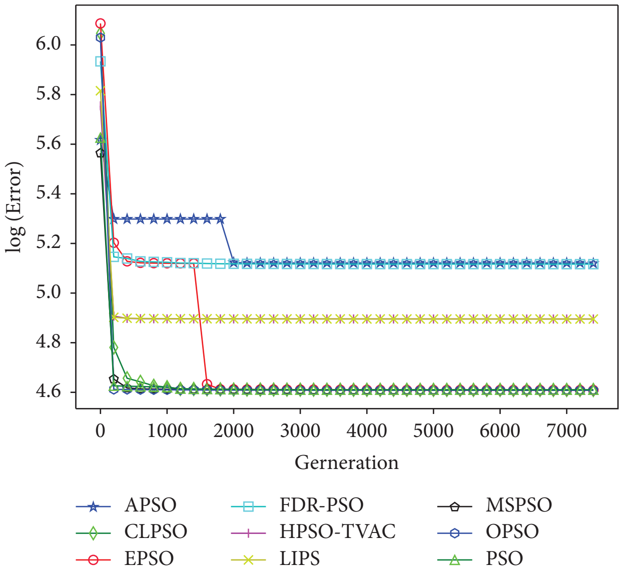


Figure 8: The results on F26 with 30 variables.

Table 5: Results of the hydrothermal scheduling instances.

Problems	Criteria	CoBiDE	TLBO	ALC-PSO	DNS-PSO	EPSO	MSPSO
Instance 1	Mean	$1.10E+06$	$1.28E+06$	$1.06E+06$	$9.77E+05$	$9.42E+05$	$9.38E+05$
	Std	$7.90E+04$	$8.49E+05$	$1.69E+05$	$8.38E+04$	$2.72E+03$	$2.93E+03$
Instance 2	Mean	$1.73E+06$	$1.54E+06$	$1.51E+06$	$1.41E+06$	$1.15E+06$	$1.27E+06$
	Std	$1.32E+05$	$7.93E+05$	$3.27E+05$	$1.90E+05$	$1.71E+05$	$1.11E+05$
Instance 3	Mean	$1.09E+06$	$1.28E+06$	$1.20E+06$	$9.78E+05$	$9.46E+05$	$9.45E+05$
	Std	$7.02E+04$	$8.49E+05$	$4.45E+05$	$8.57E+04$	$3.66E+03$	$6.32E+03$

Table 6: Computational result of CEC 2010 with 1000 variables.

Functions	Criteria	PSO	GWO	SCA	SSA	MSPSO
f1	Mean	$4.32E+10$	$5.69E+10$	$1.57E+11$	$6.16E+10$	$1.59E+10$
	Std	$6.39E+09$	$5.44E+09$	$1.08E+10$	$5.31E+09$	$1.26E+09$
f2	Mean	$1.71E+04$	$1.35E+04$	$1.77E+04$	$1.46E+04$	$8.30E+03$
	Std	$3.81E+02$	$1.64E+02$	$4.48E+02$	$2.38E+02$	$2.95E+02$
f3	Mean	$1.64E+01$	$9.37E+00$	$1.61E+01$	$1.53E+01$	$1.96E+01$
	Std	$3.72E-01$	$5.56E-02$	$1.43E+00$	$1.041E-01$	$3.88E-02$
f4	Mean	$6.34E+13$	$1.32E+14$	$1.70E+15$	$7.78E+13$	$6.16E+12$
	Std	$2.77E+13$	$6.96E+13$	$2.56E+14$	$1.19E+13$	$6.96E+11$
f5	Mean	$5.87E+08$	$2.51E+08$	$6.30E+08$	$3.94E+08$	$4.22E+08$
	Std	$8.68E+07$	$5.12E+07$	$1.84E+07$	$6.93E+07$	$2.84E+07$
f6	Mean	$1.99E+07$	$1.28E+07$	$1.95E+07$	$1.31E+07$	$1.97E+07$
	Std	$3.60E+05$	$1.92E+06$	$4.09E+05$	$7.28E+06$	$5.76E+04$
f7	Mean	$2.70E+11$	$3.91E+10$	$1.35E+11$	$3.85E+10$	$3.43E+08$
	Std	$1.51E+11$	$6.54E+09$	$1.42E+10$	$9.09E+09$	$1.82E+08$
f8	Mean	$4.92E+14$	$1.39E+15$	$2.19E+16$	$1.12E+11$	$2.17E+08$
	Std	$9.30E+14$	$1.76E+15$	$5.00E+15$	$2.11E+11$	$1.76E+08$
f9	Mean	$7.71E+10$	$5.31E+10$	$1.78E+11$	$6.99E+10$	$6.33E+08$
	Std	$1.25E+10$	$6.91E+09$	$1.49E+10$	$7.14E+09$	$1.35E+08$
f10	Mean	$1.77E+04$	$1.34E+04$	$1.78E+04$	$1.50E+04$	$8.86E+03$
	Std	$3.82E+02$	$2.56E+02$	$4.50E+02$	$1.93E+02$	$2.64E+02$

Table 6: Continued.

Functions	Criteria	PSO	GWO	SCA	SSA	MSPSO
f11	Mean	2.30E + 02	2.23E + 02	1.38E + 03	2.26E + 02	2.16E + 02
	Std	2.92E − 01	1.75E + 00	2.43E + 00	5.89E − 01	9.10E − 02
f12	Mean	6.10E + 06	3.72E + 06	1.98E + 07	6.45E + 06	7.48E + 05
	Std	9.86E + 05	1.82E + 05	2.17E + 06	4.43E + 05	5.44E + 05
f13	Mean	1.07E + 11	2.94E + 11	9.48E + 11	3.66E + 11	5.77E + 05
	Std	1.20E + 10	2.65E + 10	1.71E + 11	3.17E + 10	2.80E + 04
f14	Mean	8.97E + 10	4.38E + 10	1.75E + 11	7.61E + 10	1.66E + 09
	Std	9.52E + 09	4.19E + 09	8.56E + 09	5.94E + 09	1.36E + 07
f15	Mean	1.75E + 04	1.30E + 04	1.85E + 04	1.50E + 04	1.03E + 04
	Std	3.33E + 02	3.28E + 02	8.81E + 01	2.87E + 02	1.80E + 02
f16	Mean	4.20E + 02	4.13E + 02	2.23E + 03	4.14E + 02	3.93E + 02
	Std	1.05E + 00	2.52E + 00	5.40E + 00	0.54E + 00	5.33E − 01
f17	Mean	1.26E + 07	4.80E + 06	3.63E + 07	9.87E + 06	1.91E + 06
	Std	2.68E + 06	5.46E + 05	3.39E + 06	8.82E + 05	3.15E + 04
f18	Mean	5.90E + 11	8.38E + 11	1.17E + 12	1.27E + 12	2.78E + 06
	Std	5.56E + 10	2.84E + 10	3.58E + 09	5.23E + 10	9.23E + 05
f19	Mean	6.85E + 07	1.12E + 07	9.06E + 07	2.41E + 07	4.97E + 06
	Std	2.69E + 07	1.18E + 06	1.52E + 07	2.53E + 06	1.51E + 06
f20	Mean	6.94E + 11	1.01E + 12	8.86E + 10	1.50E + 12	2.80E + 06
	Std	5.88E + 10	5.11E + 10	5.58E + 09	5.69E + 10	3.15E + 05
Avg. ranking		3.55	2.35	4.55	3.10	1.45
Rank		4	2	5	3	1

5. Conclusions

The paper proposes a mixed-strategy PSO algorithm called MSPSO. MSPSO uses the rate of fitness change which measures the degree of improvement of successful offspring generated by each variant to guide selection probabilities of variants. Compared with previous PSO algorithms which use the number of previous successes and failures of each variant to adjust selection probabilities, MSPSO can increase the probability of selecting excellent variants. Furthermore, the proposed Nelder–Mead variant method is introduced in MSPSO to improve the exploitation ability. The proposed algorithm is tested on CEC 2014 benchmark suites with 10 and 30 variables. Experimental results demonstrate that MSPSO has a better overall performance than the other eight PSO algorithms on all problems in terms of the solution accuracy. MSPSO is also applied to three instances of the hydrothermal scheduling problem. Computational results show that the MSPSO algorithm also has a good performance in dealing with this real-world optimization problem. MSPSO is further tested on CEC 2010 with 1000 variables. The experimental results show that MSPSO has a good performance in solving large-scale optimization problems.

Our work shows a promising direction for designing efficient mixed strategy PSO algorithms; that is, the rate of fitness change guides the selection probabilities of variants. Thus, using the rate of fitness change to design other mixed strategy evolutionary algorithms will be left for testing as a future work.

Data Availability

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

Disclosure

The short version of this study has been accepted by the 2021 IEEE International Conference on Space-Air-Ground Computing (SAGC 2021). We expanded that conference paper in this study.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200, the National Natural Science Foundation of China under Grant Nos. 61872105 and 62072136, and the Educational Science Planning of Heilongjiang Province under Grant Nos. ZJB1421113 and GJB1421251.

References

[1] X. Yang, S. Zheng, T. Zhou, Y. Liu, and X. Che, "Optimized relinearization algorithm of the multikey homomorphic encryption scheme," *Tsinghua Science and Technology*, vol. 27, no. 3, pp. 642–652, 2022.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November 1995.

[3] H. Wu, J. Liu, Z. Dong, and Y. Liu, "A hybrid mobile node localization algorithm based on adaptive MCB-PSO approach in wireless sensor networks," *Wireless Communications and*

Mobile Computing, vol. 2020, Article ID 3845407, 17 pages, 2020.

[4] Y. Wang, W. Peng, C. Qiu, J. Jiang, and S. R. Xia, "Fractional-order Darwinian PSO-based feature selection for media-adventitia border detection in intravascular ultrasound images," *Ultrasonics*, vol. 92, pp. 1–7, 2019.

[5] N. Sridevi, V. Nagarajan, and K. Sakthidasan, "Efficient traffic control and lifetime maximization in mobile ad hoc network by using PSO–BAT optimization," *Wireless Networks*, vol. 27, pp. 1–10, 2019.

[6] Y. Makhlouf and A. Daamouche, "Automatic generation of adaptive structuring elements for road identification in VHR images," *Expert Systems with Applications*, vol. 119, pp. 342–349, 2019.

[7] Y. Wang, Z. Cai, Z.-H. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1033–1046, 2020.

[8] Z. Lu, Y. Wang, X. Tong, C. Mu, C. Yu, and Y. Li, "Data-driven many-objective Crowd worker selection for mobile crowdsourcing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 531–540, 2023.

[9] J. He, T. Chen, and X. Yao, "On the easiest and hardest fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 295–305, 2015.

[10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 69–73, Anchorage, AK, USA, May 1998.

[11] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, pp. 43–60, 2015.

[12] H. Zhang, M. Yuan, Y. Liang, and Q. Liao, "A novswarm optimization based on prey–predator relationship," *Applied Soft Computing*, vol. 68, pp. 202–218, 2018.

[13] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Lévy flight for solving optimization problems," *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.

[14] H. Garg, "A hybrid pso-ga algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292–305, 2016.

[15] Q. Zhang, R. M. Ogren, and S. C. Kong, "A comparative study of biodiesel engine performance optimization using enhanced hybrid pso-ga and basic ga," *Applied Energy*, vol. 165, pp. 676–684, 2016.

[16] T.-S. Pan, T. K. Dao, and S. C. Chu, "Hybrid particle swarm optimization with bat algorithm," *Genetic and evolutionary computing*, pp. 37–47, Springer, Cham, Switzerland, 2015.

[17] G. Xu, Q. Cui, X. Shi et al., "Particle swarm optimization based on dimensional learning strategy," *Swarm and Evolutionary Computation*, vol. 45, pp. 33–51, 2019.

[18] X. Xu, Y. Tang, J. Li, C. Hua, and X. Guan, "Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy," *Applied Soft Computing*, vol. 29, pp. 169–183, 2015.

[19] J. He, W. Hou, and H. Dong, "Mixed strategy may outperform pure strategy: an initial study," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 562–569, Cancun, Mexico, June 2013.

[20] J. Pang, H. Dong, J. He, and R. Ding, "A modified comprehensive learning particle swarm optimizer," *International Journal of Performability Engineering*, vol. 15, no. 9, pp. 2553–2562, 2019.

[21] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 387–402, 2013.

[22] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[23] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," *Proceedings of Swarm Intelligence Symposium SIS*, vol. 03, pp. 174–181, 2003.

[24] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[25] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," *Computational Intelligence Laboratory*, Zhengzhou University, Zhengzhou, China, 2013.

[26] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[27] N. Lynn and P. N. Suganthan, "Ensemble particle swarm optimizer," *Applied Soft Computing*, vol. 55, pp. 533–548, 2017.

[28] H. Shi, S. Liu, H. Wu et al., "Oscillatory particle swarm optimizer," *Applied Soft Computing*, vol. 73, pp. 316–327, 2018.

[29] K. Chen, F. Zhou, Y. Wang, and L. Yin, "An ameliorated particle swarm optimizer for solving numerical optimization problems," *Applied Soft Computing*, vol. 73, pp. 482–496, 2018.

[30] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[31] S. Das and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*, Jadavpur University, Kolkata, Inida, 2010.

[32] Y. Wang, H. X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing*, vol. 18, pp. 232–247, 2014.

[33] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[34] W. N. Chen, J. Zhang, Y. Lin et al., "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.

[35] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J. S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.

[36] T. Ke, X. Li, and P. N. Suganthan, "Benchmark functions for the cec' 2010 special session and competition on large-scale global optimization," *Nature Inspired Computation & Applications Laboratory*, vol. 17, 2009.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[38] S. Mirjalili, “SCA: a sine cosine algorithm for solving optimization problems,” *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[39] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems,” *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

Hindawi
Complexity
Volume 2022, Article ID 4901746, 7 pages
https://doi.org/10.1155/2022/4901746

Research Article

Few-Shot Segmentation via Capturing Interclass and Intraclass Cues Using Class Activation Map

Yan Zhao,[1] **Ganyun Lv**,[1] **and Gongyi Hong**[2]

[1] *School of Electric Power Engineering, Nanjing Institute of Technology, Nanjing 211167, China*
[2] *Nari Group Corporation, Nanjing 211000, China*

Correspondence should be addressed to Yan Zhao; zy_njit@163.com

Received 10 December 2021; Revised 23 March 2022; Accepted 22 April 2022; Published 4 July 2022

Academic Editor: Xuyun Zhang

Few-shot segmentation is a challenging task due to the limited class cues provided by a few of annotations. Discovering more class cues from known and unknown classes is the essential to few-shot segmentation. Existing method generates class cues mainly from common cues intra new classes where the similarity between support images and query images is measured to locate the foreground regions. However, the support images are not sufficient enough to measure the similarity since one or a few of support mask cannot describe the object of new class with large variations. In this paper, we capture the class cues by considering all images in the unknown classes, i.e., not only the support images but also the query images are used to capture the foreground regions. Moreover, the class-level labels in the known classes are also considered to capture the discriminative feature of new classes. The two aspects are achieved by class activation map which is used as attention map to improve the feature extraction. A new few-shot segmentation based on mask transferring and class activation map is proposed, and a new class activation map based on feature clustering is proposed to refine the class activation map. The proposed method is validated on Pascal Voc dataset. Experimental results demonstrate the effectiveness of the proposed method with larger mIoU values.

1. Introduction

Image segmentation [1] aims to segment object regions from images, which is fundamental to many computer vision tasks [2]. Based on the deep learning-based method [3–7], the existing segmentation models can segment object well when sufficient annotations are given [8]. However, the existing segmentation methods still have two drawbacks. Firstly, the annotation generation is time consuming. The number of annotations is usually so small that it is hard to train the segmentation models from a few of annotations. The other is that the segmentation models work badly on new classes, i.e., the segmentation models only recognize the objects in the training dataset and cannot segment regions of classes unknown.

To solve the drawbacks, few-shot segmentation [9–14] is proposed. Given a set of images of new classes, with a few of annotations (support images), the aim of few-shot segmentation is to segment region of query images efficiently. However, the intuitive method of refining the segmentation model by a few of annotations is proved to be ineffective. Few-shot segmentation faces the challenges of discovering object cues from limited annotations. To this end, researchers have proposed many methods to enhance few-shot segmentation [15–17]. These methods can be summarized to provide segmentation cues from existing annotations of known classes where the annotations are sufficient to train the model. Therefore, the class-agnostic guided model that transfers segmentation cues from support mask to query mask can be trained firstly and is then used in reference stage to locate the foreground regions in query image directly. Several strategies such as mask transferring and prototype feature are used. The few-shot segmentation has been improved obviously.

Meanwhile, few-shot segmentation still faces the lack of object priors although many existing annotation datasets are used. Two reasons caused such challenge. Firstly, there are large variation interclasses, which make the knowledge transferring between known class and new class very hard. Secondly, there is large variation intraclass. Therefore, a few of annotations cannot describe all the types of classes and leads to bad guidance. In other words, the foreground priors are still limited by current few-shot segmentation manner.

In this paper, we propose a new few-shot segmentation method that considers two aspects, namely, interclass cue and intraclass cue to capture more sufficient segmentation cues from known and unknown classes. The first one captures the semantic relationships between the existing classes and unknown classes and is used to capture the discriminative cue through comparing existing classes and unknown classes. The second one captures the common cues intraclasses, that is, the common features shared by the query and support images are captured to locate the object. The two aspects are achieved by class activation maps (CAMs). A classification model considering only class-level labels is first built. Then, class activation map is extracted based on the feedback analysis. Afterwards, since the discriminative regions are usually small, we expand the discriminative region using the feature clustering method guided by support masks. Finally, the CAM is introduced into the few-shot segmentation mask as an attention map to enhance the query image segmentation.

The contributions of the proposed method are listed as follows:

(1) A new few-shot segmentation method based on the segmentation cues interclass and intraclass is proposed

(2) Class activation map is used to capture the segmentation cues, and a new attention module is proposed to add the class activation map in to the few-shot segmentation network

(3) An extension method based on the clustering method is proposed to enlarging class activation map

2. Related Work

Few-shot segmentation aims to segment regions of new classes with a few annotated images given, which is a fundamental task in computer computing [18, 19]. The few-shot segmentation task is always formulated as an information guidance model, where the common knowledge that can be used in segmentation task is learned in the support branch and transferred between the support branch and the query branch. There are two key components in existing few-shot segmentation methods, of which the first component is a class prior extraction module in the support branch, and the second component is a guidance network to transfer the extracted knowledge between branches.

As for class prior extraction, multiple types of class prior have been proposed and can be further categorized into the weight-based methods and prototype-based methods. The weight-based methods consider the weight of a classifier as the class prior. The most representative work in the weight-based methods is OSLSM [10], which leverages a conditional branch to generate parameters for query branch.

The current state-of-the-art methods are prototype-based methods. The prototype-based methods can be further divided into the global prototype, the fusion of global and local prototype, and the prototype of background. The global prototype-based methods consider the converted deep features from the support branch into the class prototype, e.g., PANet [20] and CANet [21] learn a class-specific global prototype with a masked average pooling operation.

The second type of prototype-based methods takes the global and local prototypes into consideration simultaneously and has the ability to extract features with more semantic knowledge. The most representative methods are PPNet [22] and PMMs [22], where the first method decomposes the holistic class representation into a set of part-aware prototypes with k-means and the second correlates the diverse image regions with multiple prototypes to enforce the prototype-based representation with the aid of the EM algorithm.

The third type of prototype-based methods employs the background prototype to enhance the semantic knowledge of foreground. The most representative methods are MLCNet [23] and SCNet [24], where the first method introduces a mining branch that exploits latent novel classes via transferable subclusters and the second method generates self-contrastive background prototypes directly from the query image, enabling the construction of complete sample pairs to form a complementary and auxiliary segmentation task.

As for the design of guidance network between support branch and query branch, multiple types of guidance module have been proposed and can be further categorized into the feature-level guidance network and the parameter-level guidance network. The feature-level guidance conducts similarity propagation based on the extracted features by diverse branches. The representative methods include PFENet [25], which generates the prior mask based on the cosine similarity between features, and then employ the feature enrichment module to propagate this similarity in multiple resolutions. LTM [26] proposed a nonparametric and class-agnostic transformation method, where the relationship of the local features is calculated in a high-dimension metric embedding space based on cosine distance, and then are mapped from the low-level local relationships to high-level semantic cues with the generalized inverse matrix of the annotation matrix. The parameter-level guidance network considers the model parameter of the last specific layer as the class prior and uses the parameter transformation from support branch to query branch to achieve the guidance. The most representative work is CWT [27], where the guidance is conducted at the classification layer only; it proposed a Classifier Weight Transformer to dynamically adapt the support-set trained classifier's weights to each query image in an inductive way.

3. The Proposed Method

3.1. The Pipeline of the Proposed Method. The pipeline of the proposed method is shown in Figure 1, where the proposed method consists of four steps: the classification step, the

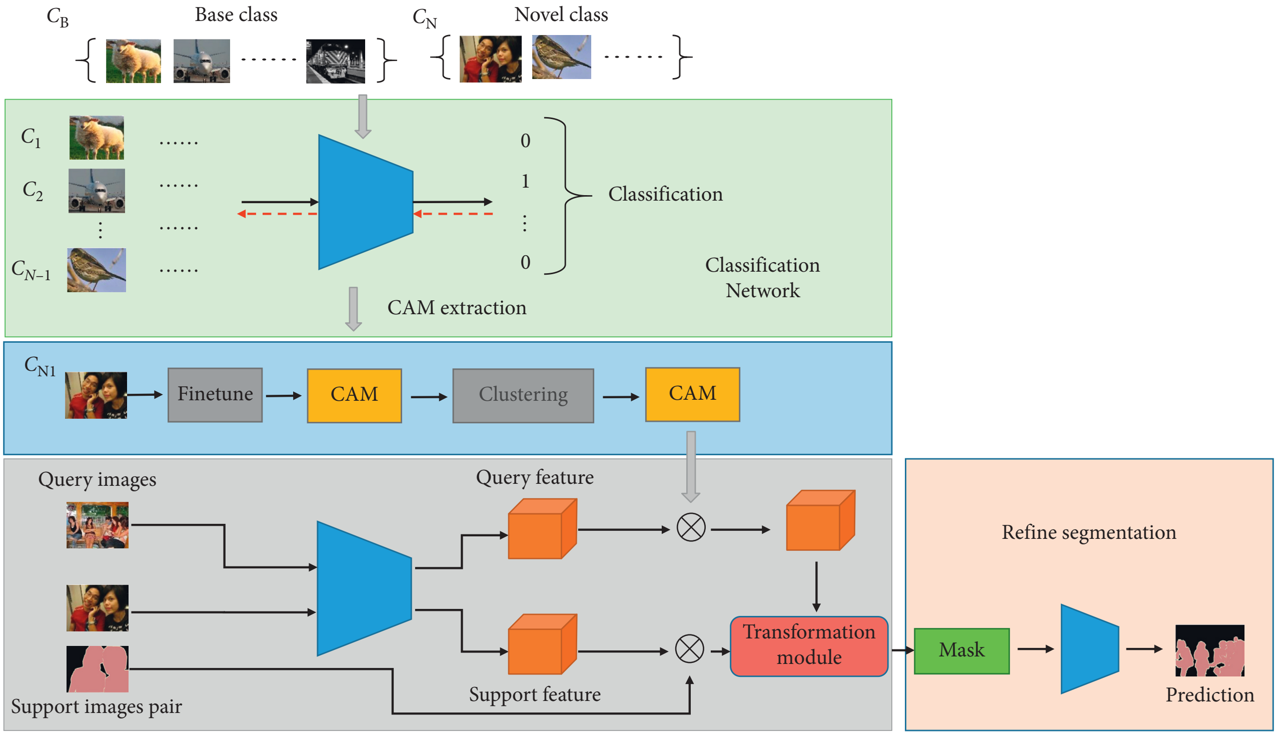


FIGURE 1: The pipeline of the proposed method.

CAM generation step, the mask generation step, and the mask refinement step. The classification step is to train a classification network by considering all the existing classes and the new classes based on image-level labels only and output the class activation map that represents the discriminative regions of the unknown classes via gradient feedback forward. Then, since the initial CAM is usually very small, the CAM generation step expands the CAM using the clustering strategy. Afterwards, the mask generation step generates the segmentation mask in terms of soft values based on mask transferring strategy where the CAM generated in the second step is used as attention map to enhance the features of the query image. Finally, the mask refinement step is to improve the segmentation mask based on classical segmentation framework. We next detail the four steps.

3.2. Classification Step. The aim of the classification step is to train a classification model by considering the known classes and new classes and extracts the discriminative regions of new classes that distinct the new class from the existing classes. Therefore, the rough location of new classes can be obtained in the query image.

Specifically, a training dataset $\{C_B, C_N\}$ consisting of the existing classes and the new classes is constructed firstly. Here, $C_B = \{C_1, \ldots, C_{N-1}\}$ is composed of existing classes with number $N-1$. $C_N = \{C_{N1}, \ldots C_{Nk}\}$ is the image set of new classes. Based on all classes, a classification network is trained, and the classification map C_0 is extracted using Grad-CAM methods.

Meanwhile, the regions are usually small due to the fact that the rough image-level labels cannot obtain the whole region of the object but a small area. The next step expands the highlighted region using feature clustering.

3.3. CAM Generation Step. The CAM generation step expands the class activation maps based on the idea that the regions located by the initial step can be treated as the class center, and the rest pixels similar with the region highlighted can be treated as the object regions. Therefore, we use the clustering method to obtain the similar pixels.

Specifically, the CAM generation step consists of three substeps: pixel clustering, cluster selection, and CAM generation. In the first step, the K-means clustering [28] is used to cluster the pixels into n_g clusters based on the deep features obtained in the classification step. For each cluster, each pixel is given the activation value in the class activation map, and the mean value of the cluster is obtained through averaging the activation values. The mean value represents the important for the pixel to the class, and the mean value is used as the activation value for all the pixels in the cluster. Thus, a new class activation map M is obtained.

3.4. Mask Generation Step. Mask generation step segments foreground regions of query image based on the class activation map M. Here, a few-shot segmentation network based on transferring is used, and the class activation map M is embedded into the network to enhance the guidance.

3.4.1. Few-Shot Segmentation Network. The few-shot segmentation network is constructed by the method in [26], of which the idea is to obtain the query mask M_q (with size $n \times n$) based on the relationships as follows:

$$M_q * M_s^T = R, \tag{1}$$

where M_q and M_s are the query mask and support mask, respectively, and the two masks are reshaped into column vector. R is the matrix product of M_q and M_s, with size $n^2 \times n^2$. It is seen that value one in R means that the values in M_q and M_s are all value one. Otherwise, the value in R is zero.

Once R is known, the query mask can be obtained by

$$M_q = R * \left(M_s^T\right)^{-1}. \tag{2}$$

Thus, the few-shot segmentation problem changes to obtain the Matrix product R, which can be estimated by the feature similarity of the pixels in the support and query images, i.e., the foreground pixels have the similar features, and have similarity distance of value one. Otherwise, the distance is value zero.

Based on the formulation above, the few-shot segmentation network can be constructed as a two-branch based network, with a guidance model by formula (2). The network is shown in Figure 1.

Specifically, given a support image I_s with support mask M_s and query image I_q, a two-branch based network is used to extract the pixel features. One is the support branch that extracts the features of support image F_s, and the other is the query branch that extracts the features of query image F_q. Then, the similarity matrix M_{sq} of F_s and F_q is calculated via calculating the discrete cosine distance, where

$$M_{sq}(i, j) = d\left(F_s\{i\}, F_q\{j\}\right), \tag{3}$$

where $M_{sq}(i, j)$ is the value at location (i, j). $F_s\{i\}$ and $F_q\{j\}$ are the i th feature and j th feature in F_s and F_q. d is the discrete cosine distance. Therefore, M_{sq} refers to the similarity of pixels, which is similar with the similarity relationships of masks, and can be used to estimate the matrix product R.

Then, M_{sq} is used to estimate the matrix product R and is used to obtain the query mask via (2).

Note that estimating R using the pixel feature is challenging. Thus, we use the support mask to filter the foreground regions via element-wise production.

3.4.2. Feature Enhancement via CAM. Different from the few-shot segmentation method in [26], we introduce the class activation map which carries the discriminative cues through all classes to enhance the features of query image. Specifically, as shown in Figure 1, the query image is sent into the classification network to form the initial classification map. Then, the clustering algorithm is used to refine the class activation map. The class activation map is then used as attention map to refine the deep features of query image, and the refined features guide the segmentation of query image.

3.5. Mask Refinement Step. The output of few-shot segmentation branch is the soft mask of query image. To obtain the binary mask, a threshold can be used to obtain the hard mask from the soft mask. However, the results are sensitive to the selection of threshold. Therefore, a segmentation mask is used to segment the final mask from the soft mask, where the soft mask is used as foreground probability map, and the segmentation network is performed to obtain the final hard segmentation mask. We use the method in [8] to implement the mask refinement.

4. Experimental Results

4.1. Dataset. We next verify the proposed method based on the Pascal Voc dataset which consists of 20 classes. Similar with the existing few-shot segmentation method, the 20 classes are split into two class set. One is training set that trains the few-shot segmentation network. The other is the test set that validates the segmentation quality of the network. To fully validate the few-shot segmentation model, four splits are used. The details are found in Table 1.

4.2. Implementation Details. We implement our method on Titan-XP GPU. Pytorch is used to realize our method. The network is optimized by Adam optimizer with the initial learning rate $1e-4$. Several backbones such as VGG16, ResNet50, and ResNet 101 are used for sufficient evaluation. The pretrained backbone network based on ImageNet [29] is used for training.

4.3. Subjective Results. We first display some subjective results in Figure 2, where the input image, the prediction results, and the ground truth results are displayed. It is seen that the prediction results are similar with the ground truth results, which demonstrates the fact that our method can segment these new classes of images successfully.

4.4. Objective Results. We objectively evaluate the proposed method by mIoU and FB-IoU values that are usually used for few-shot segmentation evaluation. The results are shown in Table 2, where 1-shot and 5-shot mean the few-shot segmentation with one and five support annotations, respectively. Three backbones such as VGG16, ResNet 50, and ResNet 101 are considered. We can see that ResNet 101 obtains the best results due to the deeper layers in the networks. The results by ResNet 50 are better than VGG 16, which is also caused by the deeper network that captures more semantic features.

4.5. Comparison with Existing Methods. We also compare our method with the existing state-of-the-art methods. The comparison methods are displayed in Table 2. It is seen that our method outperforms these comparison methods, which demonstrate the effectiveness of the proposed method, especially for the comparison with the method in [26], our method can be considered as a improvement of the method [26]. It is seen that our method is better than the method in

Table 1: The detailed splitting of Pascal Voc 2012 dataset.

Subdataset	Corresponding classes
PASCAL-5^0	Aeroplane, bicycle, bird, boat, bottle
PASCAL-5^1	Bus, car, cat, chair, cow
PASCAL-5^2	Dining table, dog, horse, motorbike, person
PASCAL-5^3	Potted plant, sheep, sofa, train, tv/monitor

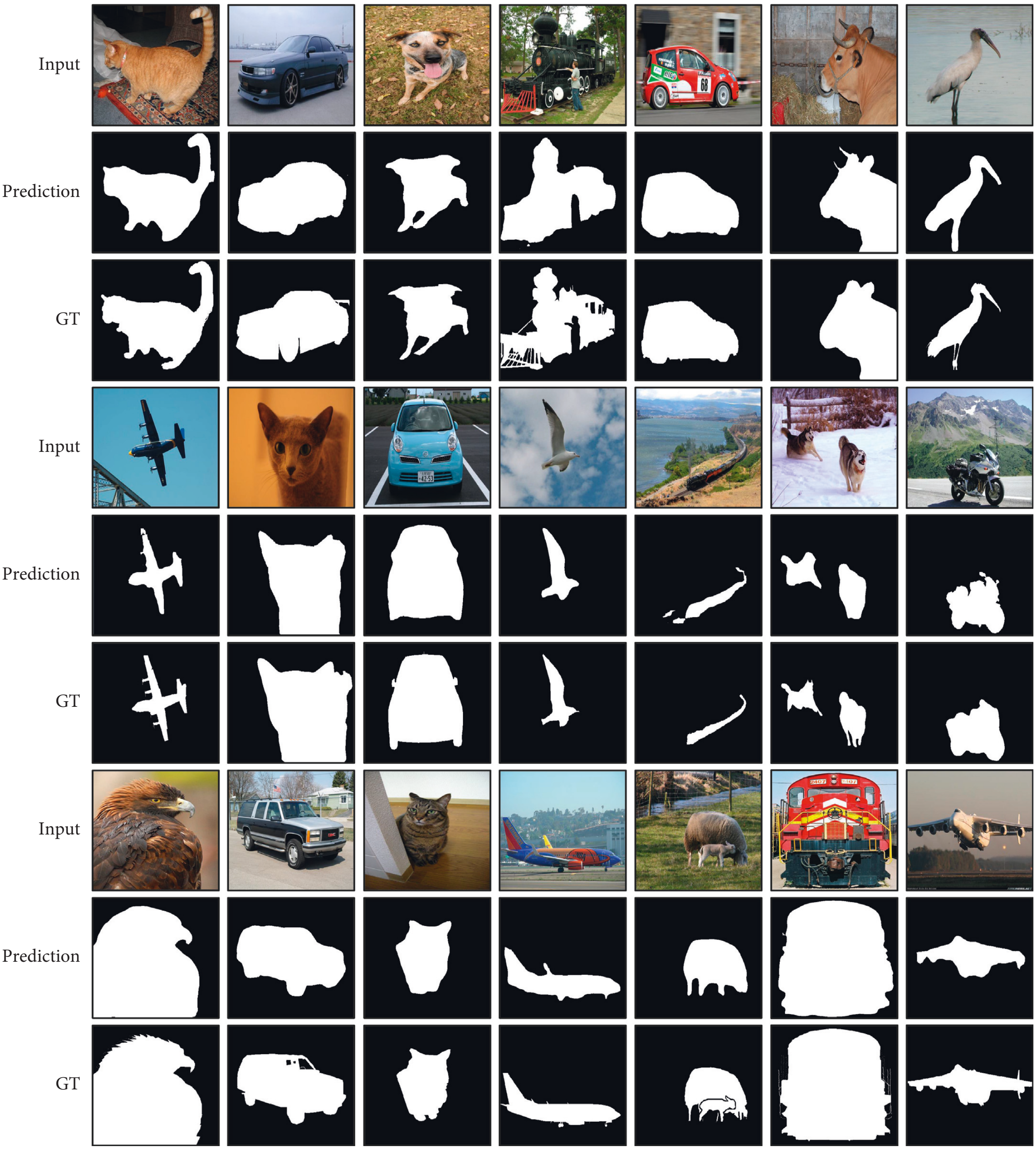


Figure 2: The segmentation results by the proposed method.

[26] which demonstrates the effectiveness of our strategy that introduces the class activation map to capture both the cues interclass and intraclass.

4.6. The Ablation Study. We next show the ablation results. The initial CAM and improved CAM are considered for the ablation study. The backbone ResNet 50 is used. The results

Table 2: Comparison with SOTA on the PASCAL-5^i dataset.

Backbone	Method	1-shot		5-shot	
		mIoU	FB-IoU	mIoU	FB-IoU
VGG 16	OSLSM [10]	40.8	61.3	43.9	61.5
	Co-FCN [30]	41.0	60.1	41.4	60.2
	SG-one [11]	46.3	63.1	47.1	65.9
	PANet [20]	48.1	66.5	55.7	70.7
	FWB [31]	51.9	—	55.1	—
	RPMM [22]	53.0	—	54.0	—
	Ours	**56.1**	**71.9**	**58.1**	**73.2**
ResNet 50	A-MCG	—	61.2	—	62.2
	CANet [21]	55.4	66.2	57.1	69.6
	PGNet [13]	56.0	66.9	58.5	70.5
	CRNet [32]	55.7	66.8	58.8	71.5
	RPMM [22]	56.3	—	57.3	—
	LTM [26]	57	—	60.6	—
	Ours	**58.3**	**73.7**	**60.9**	**74.4**
ResNet 101	FWB [31]	56.19	—	59.92	—
	DAN [33]	58.2	71.9	**60.5**	72.3
	LTM [26]	60	74	61.5	74.5
	Ours	**60.4**	**74.3**	**61.8**	**75.0**

Table 3: The ablation study on the PASCAL-5^i dataset and ResNet 50 backbone.

CAM	Our CAM	1-shot (mIoU)	5-shot (mIoU)
		57	60.6
✓		57.4	60.7
	✓	58.3	60.9

are shown in Table 3, where mIoU values are shown. It is seen that original CAM can also lead to the improvement. Meanwhile, our improved CAM can enhance the results further, which demonstrates that fact that clustering strategy is a useful method to enhance CAM regions.

5. Discussion

The existing few-shot segmentation methods usually focus on the learning class-agnostic model, which is based on the level interclasses only. Such class-agnostic model can lead to good generalization on new classes, which however also lacks the class cues of new classes. Based on the existing class-agnostic model, we try to add new segmentation cues through the discriminative cues interclass and the common cues intraclasses, which is the level of both interclass and intraclass. Therefore, better segmentation results can be obtained by our method.

It is seen that our method is based on the method in [26] (LTM), which proposed a few-shot segmentation method via estimating the relationship matrix of masks that is an interesting idea. However, our method is different from LTM [26]. Firstly, our main contribution is using the class activation map to capture the segmentation cues interclass and intraclass, which is not considered in [26]. Secondly, an attention module is added in LTM, which can add the CAM segmentation cues to enhance the segmentation. Therefore, our method can be considered as an extension to LTM [26] with better segmentation results."

6. Conclusion

This paper proposed a new few-shot segmentation method that uses the class activation map to enhance the generation of object priors by considering the common cues intraclass and the discriminative cues interclass. The proposed network consists of four steps: classification step, CAM generation step, mask generation step, and mask refinement step, which are used to generate the initial CAM via class classification, to generate the CAM via feature clustering, to generate the segmentation mask, and to refine the segmentation mask, respectively. The proposed method is validated on Pascal Voc dataset. The experimental results demonstrate that the consideration of common cues intraclass and the discriminative cues interclasses can enhance the few-shot segmentation in terms of large IoU values.

Data Availability

The datasets used for validation are available from https://host.robots.ox.ac.uk/pascal/VOC/. The detailed results are listed in the paper. More results can be found from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was partially supported by The National Natural Science Foundation of China (51577086); Jiangsu Six Talent Peaks (TD-XNY004); and Jiangsu Major Scientific Research Projects in Colleges and Universities (19KJA510012).

References

[1] 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)*, vol. 39, no. 4, pp. 3431–3440, Boston, MA, USA, June 2015.

[2] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in internet of vehicles: a survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representation (ICLR)*, San Diego, CA, USA, April 2015.

[5] Z. Hu, X. Xu, Y. Zhang et al., "Cloud-edge Cooperation for Meteorological Radar Big Data: A Review of Data Quality Control," *Complex & Intelligent Systems*, pp. 1–15, 2021.

[6] Q. Wang, C. Yuan, and Y. Liu, "Learning deep conditional neural network for image segmentation," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1839–1852, 2019.

[7] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for iov in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[9] Y. Liu, N. Liu, Q. Cao, X. Yao, J. Han, and L. Shao, "Learning non-target knowledge for few-shot semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, Louisiana, May 2022.

[10] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proceedings of the British Machine Vision Conference 2017, BMVC*, p. 167, London, UK, September 2017.

[11] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "Sg-one: similarity guidance network for one-shot semantic segmentation," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3855–3865, 2020.

[12] N. Dong and E. Xing, "Few-shot semantic segmentation with prototype learning," in *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 79–91, Newcastle, UK, September 2018.

[13] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, "Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9587–9595, Seoul, Korea (South), November 2019.

[14] Y. Yang, F. Meng, H. Li, K. N. Ngan, and Q. Wu, "A new few-shot segmentation network based on class representation," in *Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, IEEE, Sydney, NSW, Australia, December 2019.

[15] S. Zhang, T. Wu, S. Wu, and G. Guo, "Catrans: context and affinity transformer for few-shot segmentation," in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Vienna, Austria, July 2022.

[16] S. Gairola, M. Hemani, A. Chopra, and B. Krishnamurthy, "Simpropnet: Improved Similarity Propagation for Few-Shot Image Segmentation," 2020, https://arxiv.org/abs/2004.15014.

[17] J. Liu, Y. Bao, G. Xie, H. Xiong, J. Sonke, and E. Gavves, "Dynamic prototype convolution network for few-shot semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, April 2022.

[18] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[19] B. Shen, X. Xu, L. Qi, X. Zhang, and G. Srivastava, "Dynamic server placement in edge computing toward internet of vehicles," *Computer Communications*, vol. 178, pp. 114–123, 2021.

[20] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: few-shot image semantic segmentation with prototype alignment," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9197–9206, Seoul, Korea, October 2019.

[21] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "Canet: class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5217–5226, Long Beach, CA, USA, June 2019.

[22] Y. Liu, X. Zhang, S. Zhang, and X. He, "Part-aware prototype network for few-shot semantic segmentation," in *Proceedings of the Computer Vision – ECCV 2020, European Conference on Computer Vision*, pp. 142–158, Springer, Glasgow, UK, August 2020.

[23] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8721–8730, Montreal, BC, Canada, October 2021.

[24] J. Chen, B.-B. Gao, Z. Lu, J.-H. Xue, C. Wang, and Q. Liao, "Scnet: Enhancing Few-Shot Semantic Segmentation by Self-Contrastive Background Prototypes," 2021, https://arxiv.org/abs/2104.09216.

[25] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior Guided Feature Enrichment Network for Few-Shot Segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, 2020.

[26] Y. Yang, F. Meng, H. Li, Q. Wu, X. Xu, and S. Chen, "A new local transformation module for few-shot segmentation," in *Proceedings of the International Conference on Multimedia Modeling*, pp. 76–87, Springer, Daejeon, South Korea, January 2020.

[27] Z. Lu, S. He, X. Zhu, L. Zhang, Y.-Z. Song, and T. Xiang, "Simpler is better: few-shot semantic segmentation with classifier weight transformer," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8741–8750, Montreal, BC, Canada, October 2021.

[28] J. A. Hartigan and M. A. Wong, "Algorithm as 136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, p. 100, 1979.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Kai Li, and L. Li Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami, FL, USA, June 2009.

[30] K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine, "Conditional networks for few-shot semantic segmentation," in *Proceedings of the International Conference on Learning Representation workshop (ICLRW)*, Vancouver, BC, Canada, April 2018.

[31] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 622–631, Seoul, South Korea, September 2019.

[32] W. Liu, C. Zhang, G. Lin, and F. Liu, "Crnet: cross-reference networks for few-shot segmentation," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4165–4173, Seattle, WA, USA, June 2020.

[33] H. Wang, X. Zhang, Y. Hu, Y. Yang, X. Cao, and X. Zhen, "Few-shot semantic segmentation with democratic attention networks," in *Proceedings of the Computer Vision – ECCV 2020, Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 730–746, Springer, Glasgow, UK, August 2020.

Hindawi
Complexity
Volume 2022, Article ID 9031900, 15 pages
https://doi.org/10.1155/2022/9031900

Research Article

Stock Price Prediction Based on Natural Language Processing[1]

Xiaobin Tang,[1] Nuo Lei,[1] Manru Dong,[1] and Dan Ma[2]

[1] *School of Statistics, University of International Business and Economics, Beijing 100029, China*
[2] *School of Statistics, Southwestern University of Finance and Economics, Chengdu 610071, Sichuan, China*

Correspondence should be addressed to Dan Ma; 219020208012@smail.swufe.edu.cn

Received 1 November 2021; Revised 13 January 2022; Accepted 25 February 2022; Published 6 May 2022

Academic Editor: Atila Bueno

The keywords used in traditional stock price prediction are mainly based on literature and experience. This study designs a new text mining method for keywords augmentation based on natural language processing models including Bidirectional Encoder Representation from Transformers (BERT) and Neural Contextualized Representation for Chinese Language Understanding (NEZHA) natural language processing models. The BERT vectorization and the NEZHA keyword discrimination models extend the seed keywords from two dimensions of similarity and importance, respectively, thus constructing the keyword thesaurus for stock price prediction. Furthermore, the predictive ability of seed words and our generated words are compared by the LSTM model, taking the CSI 300 as an example. The result shows that, compared with seed keywords, the search indexes of extracted words have higher correlations with CSI 300 and can improve its forecasting performance. Therefore, the keywords augmentation model designed in this study is helpful to provide references for other variable expansion in financial time series forecasting.

1. Introduction

The stock market is a barometer of the macroeconomy, which reflects many investors' expectations on the market for future economic conditions. With China's financial market's continuous reform and gradual opening, the stock market plays an increasingly important role in the national economy. Since the stock market has important functions such as resource allocation, economic adjustment, and price discovery, and is closely related to CPI, interest rate, and other indicators, the stock market index has an important reference value for the government's macroeconomic policy and the central bank's monetary policy; therefore, it has always been the focus of academic and industrial research.

The research on stock market price prediction has a long history. Although Fama [1] has developed the efficient market hypothesis, indicating that under ideal conditions, information in the past has been fully reflected in the share price, thus stock price can only be affected by newly emerged information. But due to its harsh assumption, the theory is always challenged by other researchers. In the market, fundamental analysis, technical analysis, quantitative analysis and other methods still occupy a place in active investment. With the rise of behavioral finance, people gradually realize that irrational behavior in the market is widespread. For example, psychological characteristics such as the herd effect make a piece of news in the market likely to lead to drastic fluctuations in the stock market; therefore, it is possible to analyze network public opinion data by statistical methods and then predict the stock market price. With our proposed keyword augmentation strategy based on Bidirectional Encoder Representation from Transformers (BERT) and Neural Contextualized Representation for Chinese Language Understanding (NEZHA), financial institutions, for example, can acquire more timely time series by web search index and improve their risk management strategy to address evolving market fluctuation.

The structure of this study is as follows: Section 2 introduces the development of natural language processing and stock-related literature. Section 3 introduces the basic model and algorithm used in this study. Section 4 introduces the framework of stock prediction method designed in this study. Section 5 is the experimental research on the prediction of CSI 300 stock index through empirical research, and Section 6 gives the conclusion.

2. Related Work

The prediction of the stock price trend has always been studied by scholars. The existing research model of stock prediction is mainly reflected in two aspects. On the one hand, traditional econometric models are used, such as regression model and ARIMA under the framework of least squares, because of a series of constraints and nonlinear data that cannot be well dealt with, and the performance effect of the model is limited [2–4]. On the other hand, machine learning and deep learning models should be improved and used. The predictors are common features of stock data (open and volume, etc.) to establish a stable and high-precision prediction model [5–7]. In terms of data types of prediction targets, stock prediction can be divided into classified predictions based on the rise and fall of stocks [8–10] and regression predictions based on stock time series data [11–13]. The difference lies in whether the data types of prediction targets are discrete or continuous, and this study belongs to the latter type.

Scholars have made remarkable achievements in stock price prediction. Still, the common feature of existing literature is to improve prediction methods to improve prediction accuracy, and there are the following deficiencies in feature selection: (1) Although predictors are widely used, the selection of predictors mostly relies on literature and empirical intuition, and there is no relatively scientific measurement standard. Because the selection of keywords is affected by subjective factors to some extent, it is inevitable to miss important keywords due to the limited selection range. However, if the keyword index set as a predictive variable is selected improperly, it will affect the accuracy of stock price prediction to a great extent. (2). The former Natural Language Processing (NLP) vectorization technique is not sufficient in semantic recognition and understanding, which is easy to cause information loss, thus leading to the deterioration of the quality of the vocabulary expansion of predictive variables. For example, the average of word vector ignores the importance of word order and semantics, resulting in information loss. The vectorized Word2Vec model, which maps words to fixed vectors, cannot take context into account in terms of word association and lacks generalization representation ability.

NLP aims to understand and dig out the connotation of human text language by computer. It is an efficient way to analyze a large amount of network text data. From statistical language models to deep learning language models, the models' ability to represent natural language texts is constantly improving and even exceeds human representation in some areas. The statistical language model mainly extracts keywords based on word frequency and subject word distribution [14–17]. With the development of computer's computing power, the deep learning language model based on large-scale neural networks has been realized. Compared with the traditional statistical language model, it has a stronger text mining ability. The BERT model proposed by Google improves the static representation of the Word2vec algorithm [18], integrates the advantages of ELMo model and GPT model to distinguish polysemic words and parallel pretraining [19, 20], and conducts pretraining through an in-depth bi-directional transformer structure. Then the BERT model can realize the word representation integrating context semantics [21]. Based on the BERT model, the NEZHA model (Wei et al., 2019) [22] adopted Whole Word Masking (WWM) and other technologies to improve Chinese text features and achieved the SOTA effect in a number of Chinese natural language tasks. Existing literature shows that BERT shows strong semantic recognition ability from different perspectives in text classification, machine translation, q&A, and other tasks; therefore, this study adopts the BERT and NEZHA models to realize the seed keyword expansion task [23–25].

For predicting missing data, Kong et al. proposed a novel multitype health data privacy-aware prediction approach based on locality-sensitive hashing [26]. With the advent of the era of big data, the emergence of search engines provides more and more quantitative data for network public opinion analysis. Among them, the keyword web search index is widely used in the research of stock price prediction due to its features of intuitive data form, fast update speed, and strong timeliness. The current research mainly innovates on the forecasting method based on the web search index [27–30], which also provides ideas for the research of this study.

With the continuous improvement and development of deep learning technology in machine learning, LSTM can automatically search nonlinear features and complex patterns in data, and it shows excellent predictive performance in practical application research. For example, in the study of portfolio application, Fischer and Krauss (2018) compared with other prediction models, the portfolio constructed based on LSTM can obtain better investment performance [31]. Li Bin et al. (2019) constructed a stock return prediction model in fundamental quantitative investment by using cyclic neural network and long- and short-term memory networks and other technologies, and the results show that the LSTM model is significantly superior to the traditional linear algorithm in identifying the complex relationship between anomaly factor prediction and excess return [32]. Liu et al. showed that LSTM could capture the relationship between historical climate data, which has good practicability for predicting greenhouse climate [33]. Mehtab, Baek et al.'s research also shows that the deep learning LSTM model has outstanding performance in stock prediction [34, 35].

Based on the above analysis, the following research methods are presented in this study. First, based on the seed-word database summarized in the existing literature, crawler technology, and search engine are adopted to capture the web text related to the stock price as the text database, and a large number of keywords are obtained after word segmentation. Second, the BERT model is used to represent the word vectorization and calculate the word similarity to conduct preliminary screening, and then the potential predictive variable keywords are extended. Then, the NEZHA model with better performance under the Mindspore framework is selected to finetune the keyword data set and obtain the importance of words in combination with the context to screen out the predictive word variables and further expand the predictive variable keywords with higher quality. Finally, this study uses a machine learning LSTM prediction model to empirically test the set of predicted variables obtained and compares and analyzes the prediction effect of the model before and after the expansion of the set of variables.

3. Model and Algorithm

3.1. JIEBA Word Segmentation Algorithm. JIEBA word segmentation algorithm is an efficient sentence segmentation algorithm for Chinese. Compared with English, there is no obvious separation mark between Chinese words; so word segmentation algorithms are particularly important in Chinese semantic analysis. The word segmentation principle of the JIEBA word segmentation algorithm mainly includes the following three parts [36].

3.1.1. Generate All Possible DAG in the Sentence Based on the Prefix Dictionary. The JIEBA algorithm uses the data structure of Trie to store more than 300,000 common Chinese words. The prefix tree saves a large number of words in a tree-like path, concatenating words starting from the root node. Compared with the traditional hash table, it has the advantages of high efficiency and fast speed in the task of searching Chinese words.

According to the above prefix dictionary, the JIEBA algorithm abstracts all possible segmentation of a Chinese sentence into a directed acyclic graph (DAG) and records the word frequency of the training sample in Trie to further determine the most likely segmentation combination.

3.1.2. Use DP to Find the Most Probable Path and Segmentation Based on Word Frequency. In all DAGs, dynamic programming (DP) can be used to find the maximum probability path based on the word frequency in the sample. Set $\text{Path} = (\text{node}_1, \text{node}_2, \ldots, \text{node}_n)$. The goal of our programming is

$$\max \sum_i \text{weight}(\text{node}_i). \tag{1}$$

Where node_i represents each node where we possibly separate the sentence. $\text{weight}(\text{node}_i)$ represents the probability, which is represented by the frequency of the word in the corpus, of the from another node to the present node. We link these nodes together to make sure we get the most possible segmentation of the sentence. Let the route with the greatest probability be $P_{\max}$. In practice, we finds the most possible path in reverse. For node_x, there are of nodes behind such as $\text{node}_i, \text{node}_j, \text{node}_n$. Assume that the maximum split routes to reach the previous node are within $P_{\max i}$, $P_{\max j}$, $P_{\max n}$, etc. We can get the state transition equation in DP:

$$P_{\max x} = \max\left(P_{\max i}, P_{\max j}, \ldots P_{\max n}\right) + \text{weight}(\text{node}_x). \tag{2}$$

By solving this DP problem, we can find the path with maximum probability.

3.1.3. Use HMM and Viterbi Algorithm to Infer Uncollected Words. Suppose there are four hidden states of BEMS for each Chinese character in a Chinese vocabulary, namely B-Beging, E-End, M-Middle, and S-Single. The JIEBA algorithm uses Hidden Markov Model (HMM) to infer the hidden state chain of unlisted words. The conversion probability of the hidden Markov chain at each position has been stored in the above prefix dictionary, and the target sentence has provided a visible state chain. Therefore, the Viterbi algorithm is used to solve the hidden state chain of uncollected words to achieve the purpose of word segmentation.

3.2. NEZHA. The original BERT model was developed by Google. Although it has achieved good training results in English and other texts, it is mainly pretrained for English texts and not optimized for Chinese texts; therefore, there is still a lot of room for improvement. Huawei Noah's Ark Laboratory has developed a model focusing on NEural contextualiZed representation for Chinese lAnguage understanding, which is referred to as NEZHA for short [22].

Compared with the original BERT model, the NEZHA model mainly improved the following four aspects: (1) Using functional relative positional encoding that is conducive to the model's understanding of the sequence relationship in the text. (2) In the pretrained MLM task, the WWM skill is used, combined with JIEBA word segmentation. If a Chinese character is covered, other Chinese characters that belong to the same word as the Chinese character in the sentence will also be covered. Although the improvement increases the difficulty of model pretraining, it helps the model to better understand the information on the word dimension of Chinese text. (3) Using the mixed-precision training method, the data are reduced from FP32-bit to FP16-bit in the gradient calculation process, thereby reducing the volume of model parameters and speeding up the training. (4) Use Layer-wise Adaptive Moments optimizer for Batching (LAMB) training optimizer to optimize model training, shorten training time, adaptively adjust the learning rate when the batch size is large, and maintain the accuracy of gradient update. Therefore, this article uses the BERT model to initially select the matched derived keywords and then employ the NEZHA model to extract keywords from the related stock price text captured on the network.

Since NEZHA is an improved model based on BERT, we first introduce the BERT model structure based on the research of the Devlin et al [21]. Bidirectional Encoder Representations from Transformer (BERT) is a bidirectional representation encoder based on Transformer. Compared with the traditional RNN-based natural language processing model, BERT has the following advantages: (1) Using the encoder from Transformer as the model's basic structure, parallel training can be carried out, thereby improving the overall training speed of the model. (2) Compared with other generative models that also use the Transformer structure for pretraining (such as OpenAI GPT), the BERT model uses bidirectional representation for pretraining to better understand the context information token-level tasks.

The BERT model broke the records of many text understanding tasks, which is inseparable from the structure of the BERT model. The NEZHA model and the BERT model have almost the same model structure, both using the encoder part of the Transformer structure to process the input text through the stacked multihead self-attention mechanism and the fully connected network. In the Transformer structure, the embedding feature of the input text is the vector sum of the three vectors, including token embedding, segment embedding, and positional embedding. The NEZHA and BERT models have the same performance in word embedding and segment embedding. However, in terms of position embedding coding, NEZHA encodes the absolute position of BERT and improves it to functional relative position coding, which is conducive to the model's understanding of the sequential relationship in the text.

The encoder part of Transformer contains six layers, and each layer includes two sublayers, namely Multi-Head Self-Attention and Feed-Forward Network (FFN). There is a residual connection mechanism and a layer normalization mechanism between each sublayer to prevent gradient dispersion and explosion.

The self-attention mechanism is the key of NEZHA and BERT models for mining text semantics. By calculating the attention score to weight the original embedding, the attention mechanism can allow the language model to learn the dependencies between texts from a distance. At the same time, a multihead attention mechanism is formed by stacking multiple attention modules. The model can extract relevant information from different representation subspaces at different positions. Aiming at the keyword expansion demand in the stock price prediction problem, this mechanism can effectively learn the deep semantics of the keywords in the original text except for the position information, and then extract high-quality keywords related to stock prediction. The specific principle of the attention mechanism is as follows: First, the model multiplies the original embedding matrix by the corresponding weight matrix to construct three feature matrices of query (Q), key (K), and value (V). Assuming that the embedding matrix of the original text is X, and the corresponding weights to be trained are W_Q, W_K, W_V, the calculation formula of the above matrix is

$$(Q, K, V) = \left(W_Q, W_K, W_V\right)X. \tag{3}$$

Then, the weights are calculated through the query matrix and the key matrix, and normalized with the softmax function, it is weighted with the value matrix V. The specific calculation steps are as follows: First, the matrix Q and the K matrix are multiplied by dot product to calculate the initial attention weight matrix QK^T. In order to prevent the gradient dispersion problem of the Softmax function caused by the excessive value, the initial weight is further scaled to obtain $(QK^T/\sqrt{d_k})$, and then the Softmax function is used to normalize the weight. Finally, the weighted calculation is performed on the value matrix. The overall calculation formula is as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{4}$$

The multihead attention mechanism stacked at the same time can extract text information from multiple subspaces in parallel, so the multiple attention results are spliced and then multiplied by the training matrix W^O. The overall calculation formula of the multihead attention mechanism is as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right)W^o, \tag{5}$$

where the single attention mechanism $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. *Concat* function represents the splicing of multiple attention heads. The dimensions of each parameter matrix to be trained are $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $W_i^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

The next fully connected FFN will further refine the calculation results of the multihead self-attention mechanism layer. It contains two linear transformations and an intermediate ReLU activation function. The specific form is as follows:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2. \tag{6}$$

The NEZHA and BERT models have added residual connection and normalization processing common in deep networks between the abovementioned multihead self-attention layer and feed-forward neural network layer. They can be used in multilayered to improve the performance of the network; therefore, the output of each sublayer is processed as follows:

$$\text{output} = \text{LayerNorm}(x + \text{Sublayer}(x)). \tag{7}$$

The dimension of the output result is d_{model}; therefore, the basic structure of NEZHA implemented in our experiment is constructed in this study (see Figure 1). In this structure, we especially modify andutilize segment embedding so that the model better distinguishes our input ofkeywords and sentences.

The functional relative positional encoding adopted by the NEZHA model Wei et al. [22] mainly improves the calculation of the self-attention mechanism so that the attention score can take into account the relative positional relationship between the two tokens. Let the sequence of network text input for crawling stocks be $x = (x_1, x_2, \ldots, x_n)$, the output sequence value be $z = (z_1, z_2, \ldots, z_n)$, where $x_i \in \mathbb{R}^{d_x}, z_i \in \mathbb{R}^{d_z}, W^K, W^Q, W^V$ are defined as above. Then the output value is calculated as follows:

$$z_i = \sum_{j=1}^{n} \alpha_{ij}\left(x_j W^V + a_{ij}^V\right), \tag{8}$$

where α_{ij} is the attention score calculated first by scaling the dot product of query matrix Q and key matrix K between position i and position j, and then by the processing of $Softmax$:

$$\begin{aligned} \alpha_{ij} &= \frac{\exp e_{ij}}{\sum_k \exp e_{ik}}, \\ e_{ij} &= \frac{\left(x_i W^Q\right)\left(x_j W^K + a_{ij}^K\right)^T}{\sqrt{d_z}}. \end{aligned} \tag{9}$$

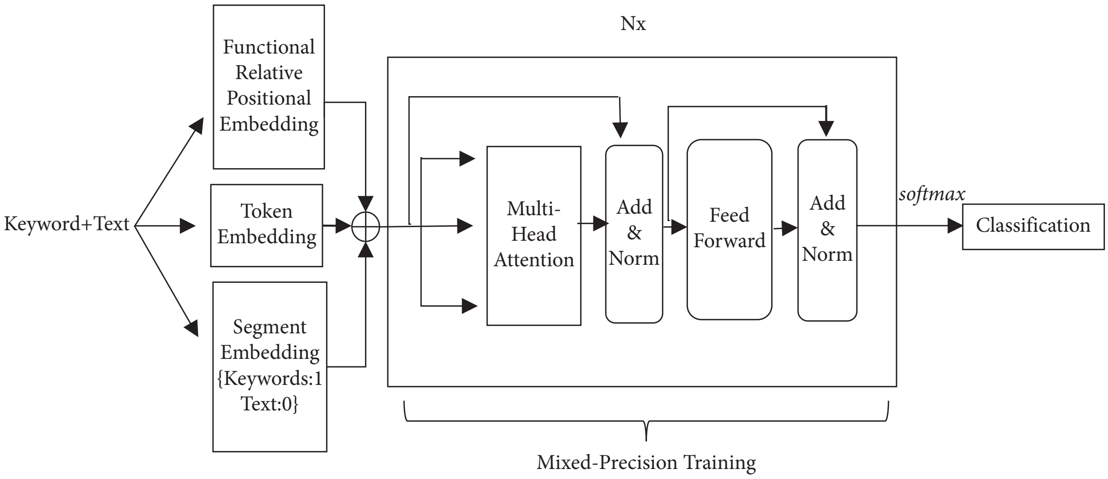


Figure 1: Basic structure of NEZHA model in our experiment. As the core of the model, it can deeply recognize and understand the semantic meaning of the text.

In formula (9), a_{ij}^{K} and a_{ij}^{V} represent the value of functional relative positional encoding. As for the case where the dimension of a_{ij} is $2k$ or $2k+1$, the calculation are as follows:

$$\begin{aligned} a_{ij}[2k] &= \sin\left(\frac{(j-i)}{10000^{(2k/d_z)}}\right), \\ a_{ij}[2k+1] &= \cos\left(\frac{(j-i)}{10000^{(2k/d_z)}}\right). \end{aligned} \tag{10}$$

Under this positional coding rule, the trigonometric function will have different wavelengths in different dimensions, which would help the model learn the information contained in the relative position of the tokens in different dimensions, thus helping to improve the model's performance in downstream tasks.

3.3. LSTM.

3.3. LSTM. LSTM is short for Long Short-term Memory. It is mainly improved based on the original RNN in its hidden layer. By introducing Input Gate, Forget Gate, and Output Gate, LSTM can effectively solve the problem that the RNN network cannot capture the long-distance dependence in the long-distance sequence as discussed by Hochreiter and Schmidhuber [37]. This study uses NEZHA model to obtain the keyword and the LSTM model to predict the stock price sequence. LSTM can mine the dependence between the keywords' web search index and the stock price compared with traditional linear models.

The input gate, forget gate, and output gate play different roles in a cell of the LSTM model. Suppose the cell state value at the previous moment is C_{t-1}, the output result of the LSTM at the previous moment is h_{t-1}, and the network input value at the current moment is X_t. The forgetting gate is responsible for controlling the degree to which the state C_{t-1} of the previous period is retained, generating the forgetting threshold vector f_t, and the input gate is responsible for controlling the size of the current network input value X_t, and generating the input threshold vector i_t. The two works together generate the current cell state C_t. After that, the output gate is responsible for outputting the current LSTM output result h_t, with its output threshold vector o_t. Based on Hochreiter and Schmidhuber [37], the specific formulas are as follows:

$$\begin{aligned} f_t &= \sigma\left(W_f \times [h_{t-1}, X_t] + b_f\right), \\ i_t &= \sigma\left(W_i \times [h_{t-1}, X_t] + b_i\right), \\ o_t &= \sigma\left(W_o \times [h_{t-1}, X_t] + b_o\right), \end{aligned} \tag{11}$$

where W_f, W_i, W_o represent the weight matrix of the forget gate, input gate, and output gate, respectively. b_f, b_i, b_o are the bias matrix. $\sigma(\cdot)$ represents the *Sigmoi d* function.

In the process of calculating the current cell state value C_t, first calculate the intermediate variable $\widetilde{C}_t$ through the activation function $\tanh(\cdot)$ through the current input value X_t and the output value h_{t-1} of the LSTM at the previous moment, and the formula is

$$\widetilde{C}_t = \tanh\left(W_c \times [h_{t-1}, X_t] + b_c\right), \tag{12}$$

where W_c represents the weight matrix corresponding to the intermediate variable $\widetilde{C}$. b_c is the bias matrix, and $\tanh(\cdot)$ represents the tanh activation function. So the calculation formula of the cell state value C_t at time t is

$$C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t, \tag{13}$$

where $\circ$ stands for dot multiplication.

Thus, the output value h_t of the cell is calculated according to the output gate to complete the calculation inside the cell:

$$h_t = o_t \circ \tanh(C_t). \tag{14}$$

In summary, the basic cell structure of the LSTM model is summarized in Figure 2.

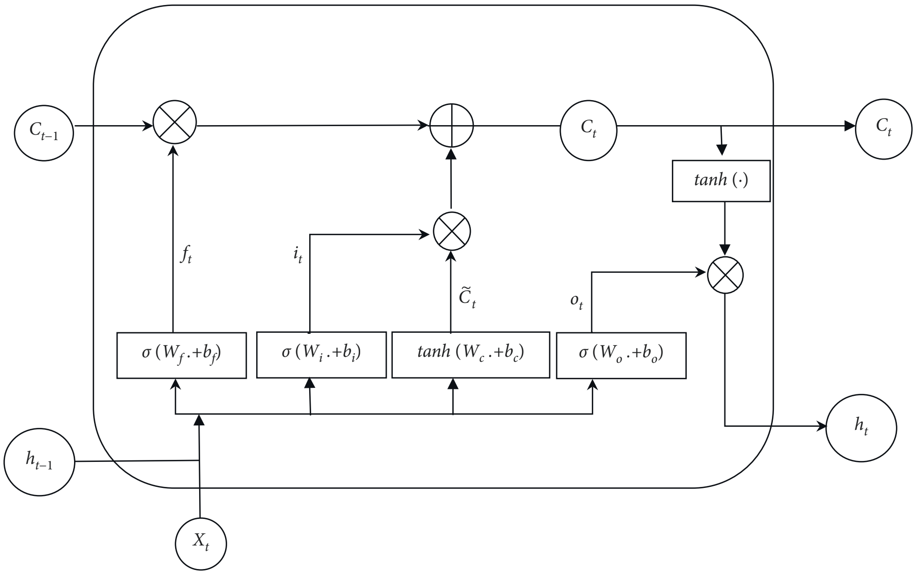


Figure 2: Basic structure of LSTM. In this context, C_t represents the cell state value at time (t); X_t represents the input value at time (t); H_t represents the cell output value at time (t) $\sigma(\cdot)$; and tanh $(\cdot)$ represent the activation function of updated information; (W) and (b) represent the weight matrix and bias of each gate control, respectively.

4. Methodology

Based on our existing seed keywords, this study first collects a large number of web texts related to stock prices through web crawlers. Second, we use JIEBA to segment the relevant texts of seed keywords, thus expanding the keyword vocabulary in terms of quantity and generating possible candidate words after removing the stop words. After that, we use the BERT model to vectorize the words and then calculate their similarity. By constructing (candidate keywords, text) pairs on the keyword data set, we apply the NEZHA model for transfer learning and further finetune it downstream, combining with the context to determine the importance of each word. Consequently, we successfully extract high-quality stock price prediction words. Finally, this study uses LSTM to predict the CSI 300 Index based on seed keywords and generated keywords, respectively. The details of our proposed algorithm are presented in Algorithm 1.

4.1. Pretraining of BERT and NEZHA. As a successful practice of transfer learning in NLP, the BERT and NEZHA models significantly reduce the difficulty of finetuning training by performing two unsupervised pretraining in a large amount of text, thereby achieving leading results in various downstream tasks. Unsupervisedpre-training methods including Masked LM (MLM)and Next Sentence Prediction (NSP) are of greatimportance in this stage [9]. The key to the keyword extraction task in this study is to infer the connection between the keyword and the sentence; therefore, it is necessary not only to dig out the meaning of the text at the word level but also to understand the logical relationship between sentences. Compared with the traditional unidirectional language model trained from left to right, the BERT and NEZHA models, as deep bidirectional network models, can predict the words covered in combination with the meaning of the context, thereby improving the model's sentence-level semantic information learning ability.

In the MLM task, 15% of the word chunks in each sentence sequence are randomly covered, marked as (MASK). The model adds a neural network at the end of the encoder as a classification layer and then uses the Softmax function to convert the output of the network into the predicted probability of each word in the vocabulary. After that, we select the word with the highest probability as the predicted result. Because in 15% of the word blocks that need to be randomly masked, the model only replaces it with (MASK) word block with 80% probability, with a random word with 10% probability, and in 10% situations, the model maintainsthe same word. This ensures that the pretraining can handle sentence without (MASK) chunks. Therefore, the probability of replacing it with a random word only accounts for 1.5% of the full text, which will not have a significant impact on the semantic understanding of the model. To be specific, the NEZHA model adopted the Whole Word Masking method here, which means the model masks not only the single Chinese character butalso other characters belonging to the same Chinese word. This skill helps the-model to better understand Chinese sentence in a more natural way and istherefore beneficial for our keywords extraction.

Compared with the MLM task, which mainly mines the token-level information inside the sentence, the NSP task focuses on understanding the logical connection of the sentence level, so it is very helpful for tasks that focus on text logic, such as question answering (QA) tasks and natural language inference (NLI). In the NSP task, the pretrained texts are sentence A and its next sentences B. Among them, sentence B has a 50% probability of matching the A sentence,

```
Input initial seed keywords from the literature
Stage 1: BERT word vector similarity selection
Initialize empty similar words vocabulary $S$
For each seed keyword $w_i$do
Collect corresponding baidu baike text
  Construct keywords vocabulary $V_i$ based on JIEBA segmentation
  Vectorize seed keywords $w_i$ and potential keywords in vocabulary $V_i$ based on BERTvec
For each keyword $v_j$ in potential keywords vocabulary $V_i$do
    Calculate cosine similarity score $\text{sim}_{ij}$ between $w_i$ and $v_j$
    IF$\text{sim}_{ij} >$ threshold then
      Add $v_j$ to similar words vocabulary $S$
    End for
  End for
Output similar words vocabulary $S$
Stage 2: NEZHA word importance selection
Initialize empty similar & important vocabulary $SI$
Collect data from CLUE data set in the form of (keywords, text)
Randomly select words from text as pseudo-keywords at a ratio of 1 : 1
Build finetune data set (Keyword/Pseudo-Keyword, text, label) as $F$
Construct training set $T$ and development set $D$ from data set $F$
Finetune BERT-TensorFlow, BERT-MindSpore, NEZHA-MindSpore in training set $T$
Select the best performing model $M$ (NEZHA-MindSpore) by precision on the development set $D$
For each keyword $v_j$ in similar words vocabulary $S$do
Calculate context importance score $I_j$ based on model $M$
Add $v_j$ and $I_j$ to similar and important vocabulary $SI$
End for
Keep words with top 100 importance scores in vocabulary $SI$
Output similar and important vocabulary $SI$
Stage 3: LSTM stock index forecast
For keyword $w_k$ in $SI$ do
For lagging term $t$ in 1 to 10 do
  Calculate lagged search index time series
End for
  Use Pearson correlation coefficient to select the most related lagged term
End for
Train LSTM to forecast CSI300 stock index on the 2215-day train data set
Calculate and compare model RMSE on the 243-day test data set
Output model RMSE
```

Algorithm 1: Experiment methodology.

which is marked as *IsNext*. In the other 50% cases, sentence B is randomly selected from the corpus and marked as *NotNext*. Since the MLM and NSP models are essentially classification tasks, the cross-entropy function is selected as the loss function; therefore, the overall loss function is obtained by adding and summing the above results. Overall, the training arrangement of textpairs, including sentences with a variety amount of lengths, enables us toprocess the logic connotation between two different pieces of texts, whichmakes it an ideal choice to select keywords from sentences.

Based on the abovementioned pretraining process, the BERT and NEZHA models have been pretrained on a large amount of corpus, thus significantly reducing the training cost of downstream tasks through this transfer learning method; therefore, this study uses the pretraining parameters from Google and Huawei. It enables the BERT model to vectorize the words and the NEZHA model to optimize the training parameters for downstream keyword discrimination.

4.2. BERT Word Vector Similarity Selection. Through a large amount of pretraining, BERT has stronger text representation capabilities as the number of network layers deepens. However, as the number of network layers increases, the output results of each layer of the network, especially the last layer, will be biased toward the pretrained objective function: the MLM task and the NSP task. Therefore, the network output of the penultimate layer is more objective and fairer and is suitable as a representative of word vectors. So in this study, we choose the penultimate network output of BERT as the word vector to represent the meaning of the word after average pooling.

The vectorization selection process uses the BERT model to vectorize the seed keywords and calculates the cosine value between the seed keywords and the candidate keywords to judge the similarity between the words and sort them by values. Then we set a certain threshold, perform preliminary screening of the candidate thesaurus according to the similarity, and keywordscorresponding to high similarity values are retained (for detailed process, see Figure 3).

4.3. NEZHA Word Importance Selection. In this study, NEZHA model is employed based on existing keywords of stock price prediction, combined with the keyword corpus material in the CLUE data set to finetune the task of identifying keywords [17]. On the one hand, we start from the seed keywords of stock forecasts, collect the Baidu Encyclopedia text corresponding to each keyword, and use the JIEBA to segment and reorganize the encyclopadia text to construct a combination of (candidate keywords, text). Thus, the candidate set of keywords is expanded in breadth. On the other hand, this study integrates the number of news corpus in CLUE, constructs (keywords/pseudo-keywords, text, tags) data sets with the same steps, and performs finetuning training through the NEZHA model to construct keywords selection model. Finally, the finetuned model is used to screen potential keywords, thus filtering the keyword set in depth. The overall tuning process is as follows (Figure 4).

In the data set for English NLP model evaluation, the GLUE data set has been widely accepted and adopted. It has become a standard test data set for evaluating the effects of many NLP models. With the rapid development of the Chinese NLP field, CLUE, a Chinese data set benchmarking similar to GLUE, came into being. The CLUE data set is called the Chinese Language Understanding Evaluation benchmark, which is the first large-scale open-source data set for NLP model benchmark testing in Chinese [38]. To extract keywords for the task of stock price prediction, this study selects the news2016zh data set in CLUE as the training data for downstream finetuning training. The original data set includes (keywords, text) pairs. Using the JIEBA word segmentation tool, this study divides the text and randomly select pseudo keywords that are different from the original keywords of the text. During this process, the ratio of the original keywords to the pseudo keywords is maintained at 1 : 1. Thus, a data set of (keyword/pseudo-keyword, text, label) is constructed for subsequent BERT/NEZHA model training and verification of the classification effect.

For the input (keyword/pseudo-keyword, text) pair, the BERT/NEZHA model encodes it in the same way as in the pretraining to serve as the input vector of the encoder and calculates the output of the numerical vector at the position of (CLS), which contains the encoding representation of the entire sentence. The model attaches a fully connected classification layer to the back end of the encoder. Suppose the parameter matrix of the fully connected layer is W and the output vector at the (CLS) position is C, then the final prediction result Prob is

$$\text{Prob} = \text{Softmax}\left(CW^T\right). \tag{15}$$

Therefore, the cross-entropy loss function is calculated and back-propagated so that all the parameters to be trained in all models are updated end-to-end.

This study builds a model based on the above structure and uses BERT and NEZHA models for training under the Tensorflow framework and the Mindspore framework, respectively. Specifically, it includes three types of models: Bert-Tensorflow, Bert-Mindspore, and NEZHA-Mindspore. The TensorFlow framework is developed and maintained by Google and is adopted by most deep learning models due to its excellent hardware compatibility and visualization ability. However, the static graph operation that Tensorflow has adopted for a long time is conducive to project deployment, but it brings great difficulties to the rapid debugging and iteration of the code. In contrast, the dynamic calculation graph used by frameworks such as Pytorch is very conducive to debugging, but it is difficult to further optimize the performance. The Mindspore framework developed by Huawei takes a different approach and adopts an automatic differentiation method based on source code conversion, which not only brings convenience to model construction but also obtains good performance through static compilation and optimization [39]. We thank MindSpore for the partial support of this work, which is a new deep learning computing framwork [40].

In terms of the hyperparameter selection of the model, most of the parameters in this article are consistent with the default situation. At the same time, to compare the classification effect of each model, the batch size and epoch on the training set, development set, and prediction set are set uniformly. Among them, the batch size of the training set is the largest batch that will not cause Our of Memory (OOM) error in the code test to accelerate model training. At the same time, the training period on the training set is set according to the recommendation of Devlin et al [21]. On the development set and prediction set, the batch size of the model is consistent with the default model with only one epoch. The selection of parameters is as Table 1.

On the training set, this study compares the classification result of different models on the development set under different frameworks so as to select the best model for classification application on the prediction set. The output results of the model on the prediction set are processed by Softmax and used as the words' score of context importance to further screen the words with predictive potential.

4.4. LSTM Stock Index Forecast. We use the LSTM model to empirically predict the stock price based on the web search index of generated word to test the interpretive and predictive ability of the generated words on the stock price. In time series forecasting, proper lag processing of the data helps to accurately describe the relationship between the explained variable and the explanatory variable, thereby improving the forecasting effect. Therefore, this article first performs a certain order of lag processing on the data, uses the Pearson correlation coefficient to screen, and selects the reliable predictor variables with strong correlation (see Figure 5).

For deep learning models such as LSTM, the selection of hyperparameters will greatly affect the model's predictive ability. The parameter setting of LSTM is referred to in the

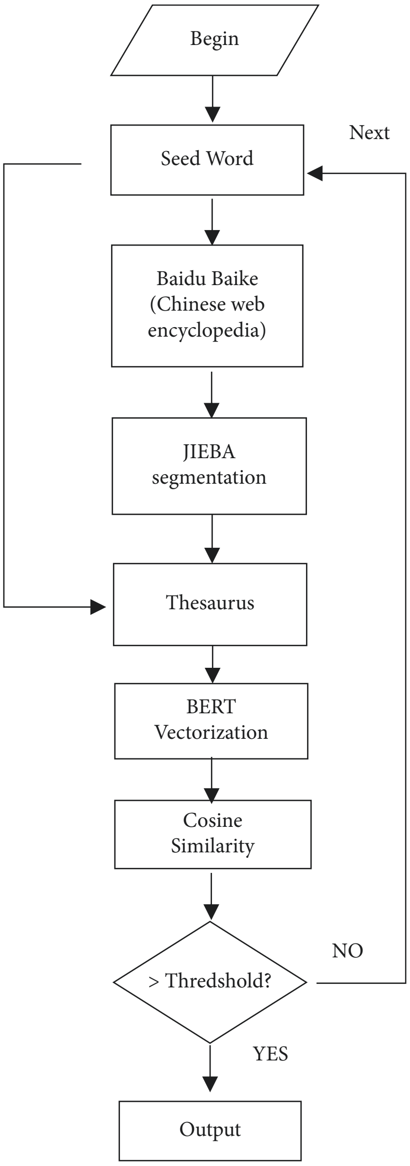


Figure 3: BERT vectorization selection process. We collected related Baike text from seed words and used BERT to perform word vectorization, which makes calculating cosine similarity possible. Finally, we select similar words above the threshold.

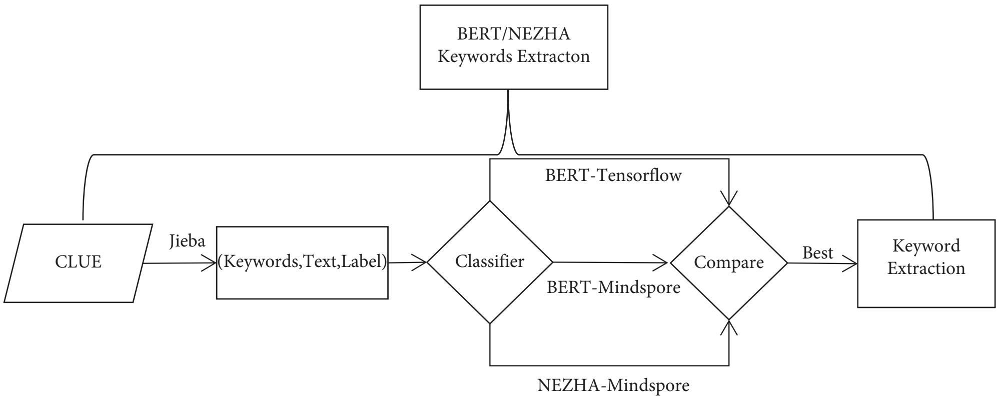


Figure 4: NEZHA importance selection process. We finetuned three different models and compared their performance to decide the best classifier and finally select the top 100 important words.

Table 1: Hyperparameters of NEZHA finetuning task.

Parameter	Train	Dev.	Test
Batch size	128	8	8
Epoch	4	1	1

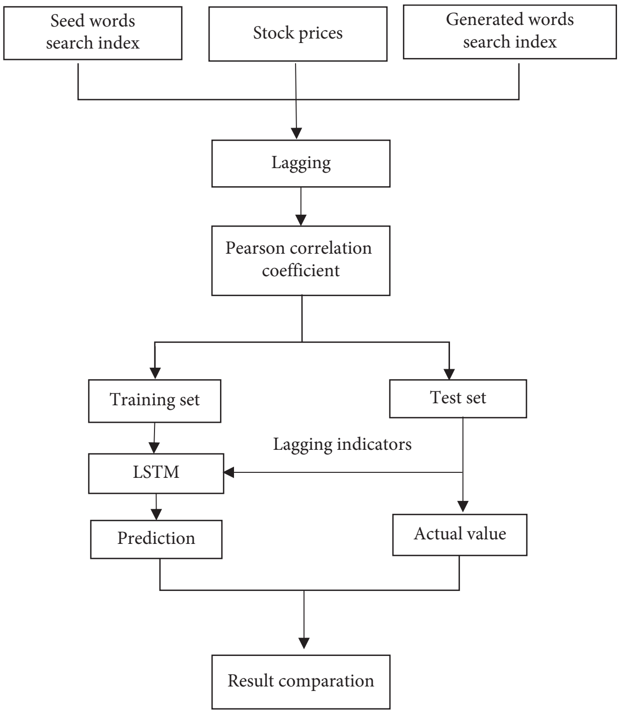


Figure 5: The LSTM Model prediction flow diagram. This figure reflects the comparison process between the predicted values and actual values of the two groups of models.

work of Tang et al. [41], in which the sliding window is set to 30 days, which means the stock price of the next trading day is predicted on the training set by learning the data of the past month. Thenumber of neuron nodes is set to 10, the total number of iterations is 500 epochs, and learning rate is 0.0006. Theoptimizer uses the Adam optimizer. The activation function of each gate is sigmoid,but the activation function of output gate adopts the tanh function, both ofwhich are the default settings of the LSTM.

5. Experiments

5.1. Experimental Data. The CSI 300 index is used as our forecast target. By referring to the existing literature and Baidu index recommendation, we select the seed keywords from the macro and micro aspects, respectively, in Table 2.

On this basis, this study uses the abovementioned vocabulary as search keywords, crawls relevant texts from Baidu Encyclopedia, and filters 19,609 long texts with a length of more than 50 words as corpus. JIEBA segmentation is performed on each text separately, and stop words are removed, thereby constructing a potential predictor variable vocabulary, with a total of 114k candidate words (under different contexts).

5.2. Similarity Selection. Based on the pretraining parameters and BERT vectorization, the potential predictor variable vocabulary related to the stock price is represented in the form of a vector through the multilayer stacked encoder mechanism.Thenthe words are screened from the perspective of similarity, and the semanticallyhighly related words are obtained. This study uses the cosine value between word vectors as a measure of words' similarity and calculates the cosine similarity for each seed keyword of stock price prediction and its corresponding candidate words. The threshold was set to 0.9, and 17,720 potential stock index prediction keywords and corresponding text context were obtained through preliminary screening. Some of the results are shown in Table 3.

5.3. Importance Selection. By calculating the similarity in the BERT vectorization model for preliminary screening, the model efficiently removes many words that have a low correlation with the seed vocabulary predicted by the stock index. Based on this, we introduce the NEZHA model to fuse the context of the candidate keywords and further filter the initial screened words through training of downstream finetune tasks, thereby carefully selecting the keywords according to their context importance.

In this stage, this study uses the news text data set in the CLUE data set. A corresponding number of pseudo keywords are randomly obtained from the text to keep the training sample balanced based on the manually labeled keywords. After that, we generate the standard data set as (text, keyword/pseudo keyword, tag (0 or 1)). In the stage of downstream finetuning, the input of the model is arranged as: [CLS] + text + [SEP] + keywords/pseudo keywords. During the training of the NEZHA model, the input is encoded by word embedding, segment embedding, and position embedding and then calculated by a multilayer encoder to generate the output vector in (CLS). Then we use the backend fully connected classification network structure and Softmax to predict the probability, representing the importance of the keyword in the text.

A total of 534,893 samples are screened in the training set, and a total of 19,609 samples are in the development set. This study trains the BERT-Tensorflow, BERT-Mindspore, and NEZHA-Mindpore models on the training set to compare the performance of the BERT model and NEZHA model in the Tensorflow framework and the Mindspore framework on the development set. Since the goal of this study is to extract keywords with high importance in the task of identifying keywords, the accuracy of the three models are compared, and the calculation formula is as follows:

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (16)$$

Among them, TP stands for True Positive, that is, the sample itself is the correct keyword, and the model judges it to be the number of correct keywords; FP stands for False Positive, the sample itself is a pseudo-keyword, and the model judges it to be the number of correct keywords. The

Table 2: Macro and micro seed thesaurus.

Aspects	Seed keywords
Macro	Financial market, bank stocks, economy, inflation, market, stock market, stocks, stock index, stock price, stock market quotations, stock market, securities, securities market, equity, a shares, A-share market, Hong Kong stocks, deposit rates, finance, GDP, CPI, bull market, rate of return, fund company, inflation rate, market conditions, rise, tax, new stocks, daily limit, bar chart, plunge, dollar, bubble, currency, policy, futures, fund company, China news, economic data, bonus, financial network, information disclosure, stock index futures, futures trading, risk management, capital, stock code, asset management, wealth, financial securities, finance, financial news, securities investment funds, Chinese stocks, exchange rates, securities networks.
Micro	Account opening, stock trading, stock recommendation, blue-chip stocks, low-priced stocks, concept stocks, banned stocks, brokerage stocks, stock introduction, stock recommendation, simulated stock trading, bank loans, stock account opening, investment and wealth management, old stockholders, investors, asset management, bankruptcy, arbitrage, financing, insider, income, shorts, speculators, retail, the main force, loans, today's stock market, today's market, restricted stocks, allotments, dark horses

Table 3: Some results of BERT similarity screening.

Seed keywords	Candidate words	Cosine similarity	Results
Financial market	Money funds	0.9212	Keep
	Bond market	0.9489	Keep
	Long-term loan	0.8497	Remove
Policy	Organization	0.9071	Keep
	System	0.9099	Keep
	Country	0.8873	Remove
Lifted stocks	Outstanding shares	0.9216	Keep
	Underweight	0.8900	Remove
	Large-cap stocks	0.8782	Remove
Bank loan	Credit	0.9213	Keep
	Working capital	0.8897	Remove
	Discount rate	0.8540	Remove

performance of the three models in the development set is shown in Table 4.

The performance of three models above verified the performanceof our experiment design. Among them, NEZHA, based on the Mindspore framework, has achieved the best performance in the development set in the keyword discrimination task. This study uses the word importance probability calculated by the NEZHA-Mindspore model as the basis for ranking. Some results of the NEZHA model are shown in Table 5.

This study ranks the abovementioned word importance, selects the top 100 generated words as candidate stock price predictors. Then we use web crawlers to obtain the corresponding Baidu search index. The time interval is set from January 1, 2011, to February 29, 2021. Some of these words were removed due to a small search volume. After deduplication, a total of 61 effective generated words and 87 effective seed words are obtained. The details are in Table 6.

5.4. Predict CSI 300 Index with LSTM. The CSI 300 Index covers the stocks of the Shanghai and Shenzhen exchange in the selection of constituent stocks, and the industry composition is consistent with the market industry distribution ratio; therefore, we choose CSI 300 Index as the object of the empirical test.

Because web search data are affected by public opinion in all aspects, some search data may have a lot of noise, which may affect the prediction ability of LSTM when predicting the CSI 300 index; therefore, this study first uses the Pearson correlation coefficient analysis method to analyze the correlation. Words with rather lower coefficients are removed with an absolute value threshold of 0.6. What is more, the lag order is set to 10. This study selects the lag term with the highest absolute value of the correlation coefficient within the 10-order lag terms of each keyword as the predictor variable. We finally determine the predictive variables by performing the above operations on the seed words and generated words, as shown in Table 7.

The predicted time interval of the CSI 300 Index is set from January 1, 2011 to March 1, 2021. The holidays with no transaction data were filtered out, and a 10-day lagging was performed to obtain a total of 2458 days of valid data. This study uses the 2215-day Baidu search index data before February 29, 2020, as the training set, and the 243-day data from March 1, 2020, to March 1, 2021, as the test set to compare forecasting ability of the seed vocabulary and the generated vocabulary. Among them, the CSI 300 stock index data come from the Wind database, and the keyword data come from the Baidu search index. After LSTM trains the CSI 300 index on the trainingsets of seed word, generates word training sets, respectively, then predicts the test set. Wedid a lot of experiments and found that the RMSE of the generated keywords is lower than the RMSE of the seed keywords in most cases, which demonstrates thestability of our prediction model. Here, we presented one of our experiment result shown in Figures 6 and 7.

Table 4: Performance of BERT/NEZHA model finetuning tasks under different frameworks.

	BERT-TensorFlow (%)	BERT-mindspore (%)	NEZHA-mindspore (%)
Precision	89.86	89.63	90.06

Table 5: Some output results of the NEZHA model.

Micro words	Prob	Macro words	Prob
Allotment payment	0.9998	Stamp duty	0.9784
Bookmaker	0.9991	Spot	0.9398
Catch up	0.9967	Risk	0.9387
Account	0.9905	RMB	0.9184
Fixed assets	0.9873	Floating exchange rate	0.9011
Short position	0.9792	Bullish candlestick	0.8975
Blue-chip stocks	0.9792	Insurance	0.8947
Sell off	0.8598	Securities law	0.7970
Settlement	0.7770	Demand deposit	0.7842
Deposit and loan	0.7455	Taxation	0.7539

Table 6: Macro and micro seed keywords and generated keywords.

Aspects	Seed keyword & generative keyword
Macro	A-share, A-share market, CPI, GDP, K-line chart, rise, China news, Chinese stocks, information disclosure, fund companies, large caps, large cap market conditions, deposit interest rates, yields, policies, new shares, plunges, futures, futures trading, exchange rate, bubble, daily limit, Hong Kong stocks, bull market, taxation, dividends, economy, economic data, US dollar, stock price, stock market, stock market quotations, stock index, stock index futures, equity, stock, stock code, stock market, securities, stock market, securities investment funds, securities networks, wealth, finance, financial news, financial networks, currency, asset management, capital, inflation, inflation rates, finance, financial markets, financial securities, bank stocks, risk management, valuation, public policy, dividends, Growth Enterprises Market, land tax, compound interest, foreign exchange, foreign exchange quotation, dalian commodity exchange, taxation, turnover tax, shenzhen stock exchange, hot money, tax burden, econometrics, surplus, middle price, blowout, interest tax, crash, liquidation, price, nasdaq, stock market crash, delisting, short-selling[2]
Micro	Main force, today's market, today's stock market, low-priced stocks, insider, brokerage stocks, arbitrage, account opening, speculators, investment and financial management, investors, income, retail accounts, concept stocks, simulated stocks, stocks, bankruptcy, shorts, old stocks, stocks introduction, stock account opening, stock recommendation, blue-chip stocks, financing, lifting ban, loans, asset management, allotment, bank loans, restricted stocks, dark horses, Shanghai Composite Index, credit, bankruptcy, borrowing, borrowing (synonym in Chinese), short selling, rebound, hold-up, bookmaker, trading volume, cost, buy at the bottom, investment, foreign exchange, shorting, cancellation, bull stock, profit, consolidation, high-quality stocks, stock reform, lifting of the ban, account, purchase of foreign exchange, capital, repayment, stock selection, allotment payment, heaveweight stock, account cancellation, ex-dividend, ex-rights, annual interest rate, popular stocks, plummeting

Table 7: Predictive variables after correlation coefficient screening.

Data type	Words	Lag	Variable	Corr. Coef.
Seed keywords	CSI 300	1	y_{t-1}	0.9979
	Inflation rate	1	inflation_{t-1}	0.6903
	Chinese news	1	news_{t-1}	−0.6836
	Policy	10	policy_{t-10}	0.6456
	Dark horse	10	dark_horse_{t-10}	0.6238
	Stock quotes	1	quote_{t-1}	0.6130
Generated keywords	CSI 300	1	y_{t-1}	0.9979
	Compound interest	1	compound_{t-1}	0.7296
	Hot money	1	money_{t-1}	0.7096
	Dividend	1	dividend_{t-1}	0.6703
	Profit	1	profit_{t-1}	0.6513
	Annual interest	2	$\text{annual\_interest}_{t-2}$	0.6218

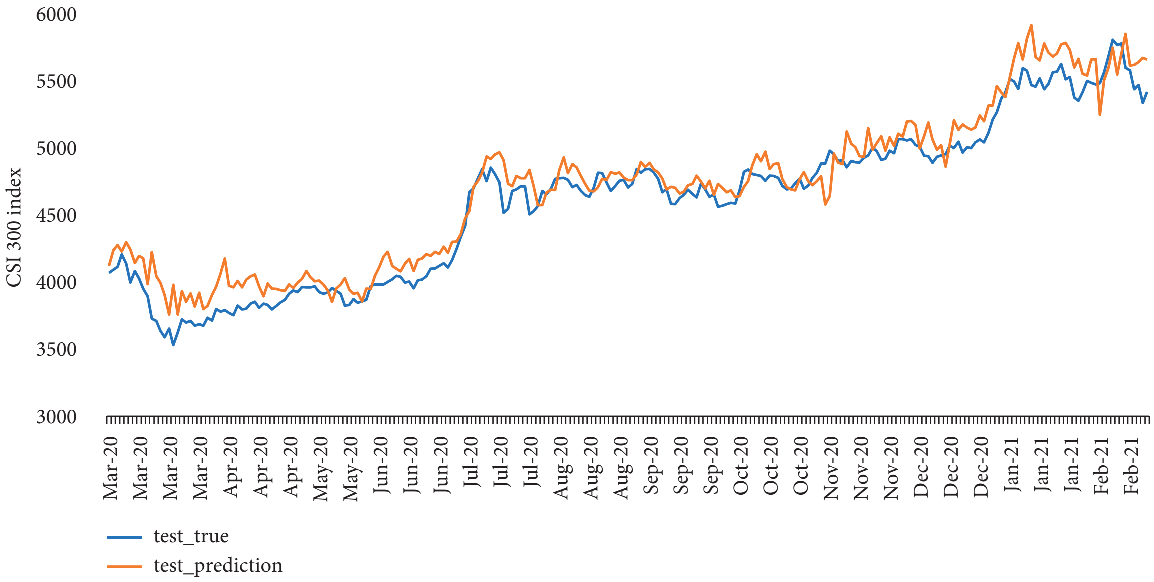


Figure 6: The trend chart of the relationship between LSTM model predicted values of seed vocabulary and CSI 300 index true values. The red and blue lines represent the true and predicted values, respectively ($n = 243$).

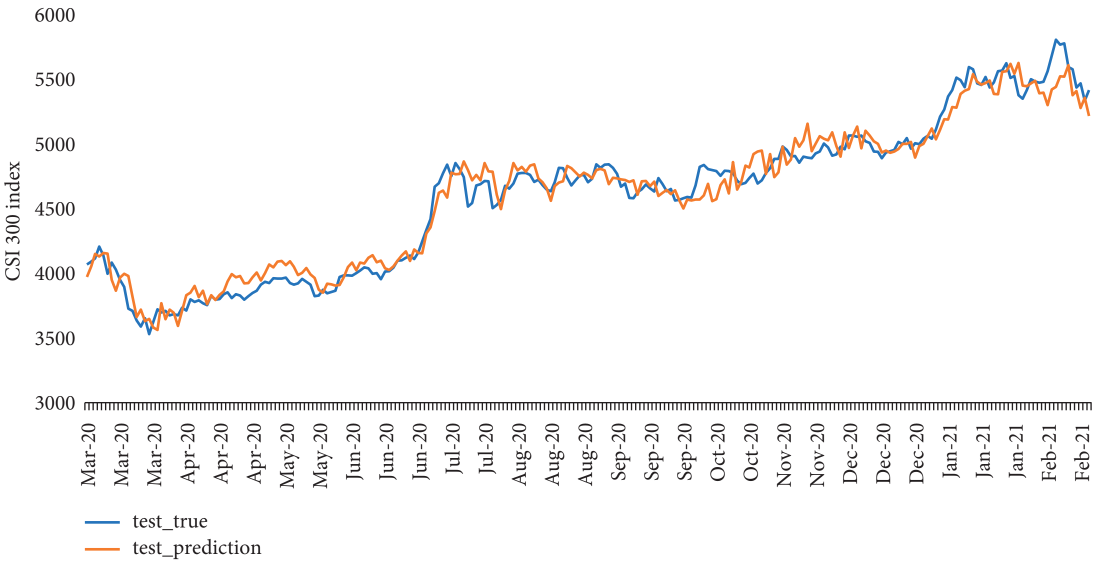


Figure 7: The trend chart of the relationship between LSTM model predicted values of Generated vocabulary and CSI 300 index true values. The red and blue lines represent the true and predicted values, respectively ($n = 243$).

Compared with the seed words of the CSI 300 Index, the same number of generated words obtained by the BERT word vector similarity filtering and NEZHA keyword selection have more stable and smooth prediction results for the CSI 300 Index. For our prediction task, this study uses the Root Mean Squared Error (RMSE) indicator as a measure of the model’s predictive ability. The smaller the RMSEmeans the better the predictive effect. The calculation formula is

$$RMSE = \sqrt{\frac{1}{m}\sum_{k=1}^{m}\left(y_k - \widehat{y}_k\right)^2}, \tag{17}$$

where y_k represents the true value, $\widehat{y}_k$ represents the predicted value, and m represents the sample size of the test set. As the result shows, in this experiment, the RMSE is 154.1831 when the lagging term of the CSI 300 index itself and the seed keywords' searchindexes are used as the predictor variable. However, the RMSE is 110.6976 whenthe lagging term of the CSI 300 index itself and the generated keywords' searchindexes are used as the predictor variable. The decrease rate is 28.20%.

Our experimental results show that, compared with the original seed keywords, the NLP text mining technology designed in this study improves the prediction accuracy and accuracy of LSTM on the Shanghai and

Shenzhen 300 stock indexes by new generated keywords with better predicting stability and better forecasting ability.

6. Conclusion

Based on BERT and NEZHA models of artificial intelligence, we optimize the text mining technology for stock price index prediction and deeply expand the keywords of higher quality predictive variables. On this basis, we use the LSTM prediction model to empirically forecast the CSI 300 stock index. The empirical results show that, based on the text information mining method of BERT model similarity and NEZHA model importance, we can screen out high-quality prediction variables with higher correlation and stronger prediction ability from network texts, thus significantly improving the prediction effect of CSI 300 stock index.

The implications are as follows: First, the artificial intelligence text mining technology based on BERT and NEZHA frontier can be better applied to stock price prediction, which not only enriches the index system of stock price prediction but also helps regulators and investors to evaluate stock price trends and control stock price risks. Second, the text mining technology can realize the keyword expansion of stock price forecast, which can provide research ideas and references for the expansion of other macro index systems. In addition, this method has strong extensibility. Future research can consider more analysis angles based on similarity and importance to achieve more high-quality keyword extension, which is also worth exploring in the following research.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

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

Acknowledgments

The authors acknowledge the helpful comments from anonymous reviewer. Xiaobin Tang was supported by CAAI-Huawei MindSpore Open Fund (no. CAAIXSJLJJ-2021-045A) and the Outstanding Young Scholars Funding Program of UIBE [No. 21JQ09]. Dan Ma was supported by the National Social Science Foundation of China (grant number no. 21&ZD149).

References

[1] E. F. Fama, "Efficient capital markets: a review of theory and empirical work." *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[2] M. Z. Asghar, F. Rahman, F. M. Kundi, and S. Ahmad, "Development of stock market trend prediction system using multiple regression," *Computational & Mathematical Organization Theory*, vol. 25, no. 3, pp. 271–301, 2019.

[3] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 106–112, Cambridge, UK, March 2014.

[4] T. Vantuch and I. Zelinka, "Evolutionary based ARIMA models for stock price forecasting," *ISCS 2014: Interdisciplinary Symposium on Complex Systems*, Springer, Cham, Manhattan, NY, USA, 2015.

[5] S. Mootha, S. Sridhar, R. Seetharaman, and S. Chitrakala, "Stock Price Prediction Using Bi-directional LSTM Based Sequence to Sequence Modeling and Multitask Learning," in *Proceedings of the 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, October 2020.

[6] S. Mehtab and J. Sen, "Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Timeseries," 2020, https://EconPapers.repec.org/RePEc:arx:papers:2001.09769.

[7] L. Dos, S. Pinheiro, and M. Dras, "Stock market prediction with deep learning: a character-based neural language model for event-based trading," in *Proceedings of the Australasian Language Technology Association Workshop 2017*, pp. 6–15, Brisbane, Australia, December 2017.

[8] X. C. Xu and K. Tian, "A new method of stock index prediction based on sentiment analysis of financial text," *Journal of Quantitative and Technical Economics*, vol. 38, no. 12, pp. 124–145, 2021.

[9] R. Kaur, Y. K. Sharma, and D. P. Bhatt, "Measuring accuracy of stock price prediction using machine learning based classifiers," *IOP Conference Series: Materials Science and Engineering*, vol. 1099, no. 1, pp. 12–49, 2021.

[10] R. Gupta and M. Chen, "Sentiment Analysis for Stock Price Prediction," in *Proceedings of the 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, August 2020.

[11] S. Mehtab and J. Sen, "Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models," 2020, https://arxiv.org/abs/2010.13891.

[12] A. Rui, "Big data business actual analysis: stock price prediction based on time series model,," *Modern Economics & Management Forum*, vol. 2, no. 2, pp. 63–71, 2021.

[13] S. Mehtab and J. Sen, "A Time Series Analysis-Based Stock Price Prediction Using Machine Learning and Deep Learning Models," 2020, https://arxiv.org/abs/2004.11697.

[14] G. Salton and C. T. Yu, "On the construction of effective vocabularies for information retrieval," *ACM Sigplan Notices*, vol. 10, no. 1, pp. 48–60, 1975.

[15] S. Deerwester and T. Landauer, G. Furnas, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[17] R. Mihalcea and P. Tarau, "Textrank: bringing order into text," in *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pp. 404–411, Stroudsburg, PA, USA, July 2004.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[19] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, https://arxiv.org/abs/1802.05365.

[20] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018, https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

[21] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[22] J. Wei, X. Ren, X. Li et al., "NEZHA: neural contextualized representation for Chinese language understanding," 2019, https://arxiv.org/abs/1909.00204.

[23] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" *Lecture Notes in Computer Science*, Springer, Cham, vol. 11856, Manhattan, NY, USA, 2019.

[24] J. Zhu, Y. Xia, L. Wu et al., "Incorporating BERT into Neural Machine Translation," 2020, https://arxiv.org/abs/2002.06823#:~:text=The%20recently%20proposed%20BERT%20has,(NMT)%20lacks%20enough%20exploration.

[25] C. Qu, L. Yang, M. H. Qiu, B. Croft, Y. Zhang, and M. Iyyer, "BERT with History Answer Embedding for Conversational Question Answering," pp. 1133–1136, 2019, https://arxiv.org/abs/1905.05412.

[26] L. Kong, L. Wang, W. Gong, C. Yan, Y. Duan, and L. Qi, "LSH-aware multitype health data prediction with privacy preservation in edge environment," *World Wide Web*, 2021.

[27] T. Preis, H. S. Moat, and H. E. Stanley, "Quantifying trading behavior in financial markets using Google Trends," *Scientific Reports*, vol. 3, no. 1, pp. 1684–1686, 2013.

[28] B. Weng, M. A. Ahmed, and F. M. Megahed, "Stock market one-day ahead movement prediction using disparate data sources," *Expert Systems with Applications*, vol. 79, pp. 153–163, 2017.

[29] H. Hu, L. Tang, S. Zhang, and H. Wang, "Predicting the direction of stock markets using optimized neural networks with Google Trends," *Neurocomputing*, vol. 285, pp. 188–195, 2018.

[30] Y. Liu, G. Peng, L. Hu, J. Dong, and Q. Zhang, "Using Google Trends and Baidu index to Analyze the Impacts of Disaster Events on Company Stock Prices," *Industrial Management & Data Systems*, vol. 120, 2019.

[31] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[32] B. Li, X. Y. Shao, and Y. Y. Li, "Research on fundamental quantitative investment driven by machine learning," *China Industrial Economics*, vol. 8, pp. 61–79, 2019.

[33] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2022.

[34] J. Sen, S. Mehtab, and A. Dutta, "Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models," 2021, https://arxiv.org/abs/2009.10819.

[35] Y. Baek and H. Y. Kim, "ModAugNet: a new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Systems with Applications*, vol. 113, pp. 457–480, 2018.

[36] J. Y. Sun, "Jieba" Chinese Text Segmentation," 2012, https://github.com/fxsjy/jieba.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] L. Xu, X. Zhang, and Q. Dong, "CLUECorpus2020: a large-scale Chinese corpus for pre-trainingLanguage model," 2020, https://arxiv.org/abs/2003.01355.

[39] Y. Fan, "Research on the next-generation deep learning framework," *Big Data Research*, vol. 6, no. 4, pp. 69–80, 2020.

[40] Mindspore, 2020, https://www.mindspore.cn/.

[41] X. Tang, M. Dong, and R. Zhang, "Research on the prediction of consumer confidence index based on machine learning LSTM&US model," *Statistical Research*, vol. 37, no. 7, pp. 104–115, 2020.

Hindawi
Complexity
Volume 2022, Article ID 6860915, 9 pages
https://doi.org/10.1155/2022/6860915

Research Article

Quick Compression and Transmission of Meteorological Big Data in Complicated Visualization Systems

He-Ping Yang,[1] Ying-Rui Sun,[1] Nan Chen,[1] Xiao-Wei Jiang,[1] Jing-Hua Chen,[1] Ming Yang,[2] Qi Wang,[1] Zi-Mo Huo,[1] and Ming-Nong Feng[1]

[1]*National Meteorological Data Center, Beijing, China*
[2]*Zhejiang Meteorological Information and Network Center, Zhejiang Meteorological Bureau, Hangzhou, China*

Correspondence should be addressed to Ying-Rui Sun; sunyr@cma.gov.cn

Received 24 November 2021; Revised 12 January 2022; Accepted 12 March 2022; Published 5 May 2022

Academic Editor: Ning Cai

The sizes of individual data files have steadily increased along with rising demand for customized services, leading to issues such as low efficiency of web-based geographical information system (WebGIS)-based data compression, transmission, and rendering for rich Internet applications (RIAs) in complicated visualization systems. In this article, a WebGIS-based technical solution for the efficient transmission and visualization of meteorological big data is proposed. Based on open-source technology such as HTML5 and Mapbox GL, the proposed scheme considers distributed data compression and transmission on the server side as well as distributed requests and page rendering on the browser side. A high-low 8-bit compression method is developed for compressing a 100 megabyte (MB) file into a megabyte-scale file, with a compression ratio of approximately 90%, and the recovered data are accurate to two decimal places. Another part of the scheme combines pyramid tile cutting, concurrent domain name request processing, and texture rendering. Experimental results indicate that with this scheme, grid files of up to 100 MB can be transferred and displayed in milliseconds, and multiterminal service applications can be supported by building a grid data visualization mode for big data and technology centers, which may serve as a reference for other industries.

1. Introduction

Currently, the development of information collection and storage technology has ushered in the era of big data in various industries, as the amounts of data being recorded, processed, and analyzed have exploded. In particular, meteorological data are among the most important types of data encountered in people's daily lives, playing an essential role in understanding the environment, natural resources, the economy, and other aspects of life [1]. To address society's need for refined meteorological data, grid data products for observations and predictions based on radar data, satellite data, and station observations have been extensively utilized [2]. On a global scale, the available meteorological grid data mainly include numerical forecasting products from the European Centre for Medium-Range Weather Forecasts (ECMWF) [3], the National Centers for Environmental Prediction (NCEP) Global Forecast System (GFS) model [4], the Global Regional Assimilation and PreEdiction System Global Forecast System (GRAPES_GFS) model of the China Meteorological Association (CMA) [5], and the real-time High-Resolution CMA Land Data Assimilation System (HRCLDAS) [6]. The most common storage formats for such grid data products are General Regularly-distributed Information in Binary form (GRIB) and Network Common Data Form (NetCDF). The former is a file format that was designed by the World Meteorological Organization (WMO) for storing and transmitting meteorological grid data, such as the outputs of numerical weather prediction models; this format is concise enough to be widely used in meteorology to store historical and forecast weather data [7]. The latter is an array-oriented and network sharing-based data description and coding standard proposed by scientists of the Unidata project at the University Corporation for Atmospheric Research (UCAR) [8].

Files in these commonly utilized grid data formats can only be opened by professional applications (apps) and can be used to obtain values at specific locations or to analyze the spatial distributions of variables such as temperature or rainfall [9]. For meteorological big data, which tend to have strong geographical spatial characteristics and location correlations, geographical information system (GIS) technology is usually combined with tools for visual expression and application provided by common services [10]. Panoply from the National Aeronautics and Space Administration (NASA) Goddard Institute for Space Studies, which is a viewing tool rather than a data extraction tool that supports multiple formats such as GRIB, requires the Java Runtime Environment [11]. MeteoInfo (i.e., MeteoInfoMap and MeteoInfoLab), which was developed by the Chinese Academy of Meteorological Sciences, is an integrated framework for both GIS applications and scientific computation environments that is utilized by the meteorological community to visualize and analyze spatial and meteorological big data in multiple data formats [12]. As the centerpiece of National Weather Service operations in China, the Meteorological Information Comprehensive Analysis and Process System (MICAPS) is a complicated computer system that combines meteorological, satellite, and radar data into one workstation and allows graphical and alphanumeric weather data in GRIB format to be read, analyze, combined, and manipulated [13].

All of the above methods have been used to display local meteorological files. However, with the emergence and popularization of cloud storage, new types of applications have arisen in which computing resources are no longer localized but rather distributed, heterogeneous, and dynamic. The Grid Analysis and Display System (GrADS) is a widely used drawing software tool in meteorology. It has two main functions, namely, data processing and image display, and plays a role in the meteorological research community similar to that of the World Wide Web in facilitating information exchange over the Internet. GrADS has unique characteristics, mainly for scientific research and business personnel involved in atmospheric and marine research. With its powerful data analysis capabilities, flexible environment setup, wide range of mapping types and variety of map projection methods, GrADS has greatly aided meteorological research [14]. The Integrated Data Viewer (IDV) from Unidata/UCAR is a Java-based software platform for analyzing and visualizing geoscience data [15]. The IDV combines the abilities to display and analyze satellite imagery, gridded data (such as numerical weather prediction model outputs), GIS data, and other data in a single interface [16]. It has been integrated with common scientific data servers, including Unidata's THematic Realtime Environmental Data Distributed Services (THREDDS) Data Server (TDS) [17], as data sources to enable easy access to a large number of real-time and archival datasets. The IDV is the main tool used in the computer laboratory portion of various meteorological courses at colleges and universities.

These localized applications can read files directly from a local disk, efficiently download them to a local disk, or integrate them with common scientific data servers. However, the local installation process has relatively high system requirements and is therefore not suitable for public services.

As mobile terminal apps such as Wireless Application Protocol (WAP) browsers and WeChat have been developed for IOS and Android operating systems, rich Internet applications (RIAs), which are web-based applications designed to deliver the same features and functions normally associated with desktop applications, have become essential platforms that can run in web browsers without installation. One of the earliest attempts to make RIAs accessible was to use the World Wide Web Consortium (W3C) standard for Accessible Rich Internet Applications (ARIA) [18]. This technology has been used to visualize meteorological big data with web-based GIS (WebGIS) tools. Rain Viewer, an all-in-one weather radar and rain forecast app for predicting storm tracks, is available for 90 countries and offers the most comprehensive weather radar coverage on the market, displaying a single map with data from 1000+ Doppler radars with the option of viewing information about each radar on the map. To provide this functionality, all requests to Rain Viewer are routed through an online web service that overlays the Rain Viewer data on a map tile with 256 or 512 pixels centered on the user's current location and then resizes the image to match the screen size of the device [19]. In the WebGIS service OpenStreetMap, the layer overlay is displayed within milliseconds by using leaflet technology and considering the aging of single-slice requests. Thus, this service cannot meet the requirements of real-time interactions, such as changing the color range or filtering by value. Moreover, the layer is virtualized, and the user experience can be poor if the maximum image resolution is exceeded after the image is enlarged. Advancements in vector tile technology have offered solutions to the above problems, in which vector tile layers are saved as compressed files in the Protocolbuffer Binary Format (PBF) file format [20]. Such a compressed file, which contains vector map data in one or more layers, can be rendered and styled based on the style of each layer. The data in a vector tile include geographic features in the forms of points, lines, and polygons [21]. For weather radar, this solution goes beyond simply resampling the data and aims to generate vector contours based on the raw radar data. With the data in the form of vector polygons, the AerisWeather Mapping Platform (AMP) can render radar data at any zoom level without reducing the resolution or quality. This also allows the user to control how much smoothing is applied to the radar data, enabling clean and smooth radar imaging at the city and neighborhood levels [22].

Meteorological big data vary spatially and temporally, and any dynamic vector slicing scheme must include data preparation, slicing, and front-end visualization, with high-performance requirements for the server hosting the spatial database (PostGIS) [23]. Hence, complex visualization technology for meteorological big data is gradually developing in the direction of data file compression prior to transmission, followed by foreground decoding. The Null school designed a global visual display system (Nullschool.net) [24] for ECMWF forecast data, which converts a map from the Natural Earth dataset into the TopoJSON format to serve as the base map, utilizes the EPAK format to transform and compress the grid data, and applies Node.js to rapidly render and display the foreground [25]. Since the launch of this system, the Tokyo Meteorological Bureau and other institutions have developed the Tokyo Wind Map based on this technical framework, which has been extensively promoted. Although this framework makes full use of the advantages of rapid visualization on a web terminal, there are still incompatibility problems on mobile terminals for base maps with coastline contour resolutions of 50 km or 110 km in the JSON format; hence, it is difficult to use this framework for fine-scale service applications. Lytvyn et al. [26] and Prastika et al. [27] have implemented multichannel support applications such as PC browser, WAP, and mobile applications based on the OpenStreetMap online map by integrating slices with data compression and transmitting only single slices within 30 kB for faster transmission and visualization. For data visualization based on the browser/server (B/S) architecture, the minimum resolution of meteorological grid data reaches more than 9 km worldwide, and the file size for a single transmission is between 700 kB and 2 MB. A single request can reach a second-level response, and dynamic rendering allows updating, which satisfies the requirements for large-scale services. However, various disaster prevention and emergency mitigation support applications need more refined grid data services. In 2021, the CMA released the HRCLDAS product [28], which covers East Asia with a resolution of 1 km. The grid size is 7000 $*$ 4500, and the data volume of a single file reaches 106 MB. The main applications for visualizing such data superimpose a transparent image directly on top of the map and reduce the zoom or resolution layer by layer, resulting in a loss of eigenvalues (e.g., the maximum and minimum values) and thus affecting the front-end rendering results. Therefore, it is extremely difficult to balance efficiency and data accuracy in page rendering based on the B/S architecture.

In conclusion, visualization systems based on the B/S architecture are the most user-friendly option; however, their compression and transmission methods have become increasingly complicated as the demand for refined services and individual file sizes have increased. Compression is a highly efficient method for files smaller than 10 MB, but the visualization process achieves better transmission and rendering pressure when the file size is approximately 100 MB or larger, and 100 MB is a common file size for refined meteorological services. Therefore, there is a need to design a fast transmission and display scheme for meteorological grid data on PCs, mobile browsers, WAP apps, WeChat applets, and other platforms based on an open-source WebGIS service platform.

In this article, we propose a customized scheme for use in complicated visualization systems for meteorological big data. At the back end, a high-low 8-bit compression algorithm is adopted, and customized slice transmission is required to ensure high network transmission efficiency. Based on the HTML5 and Vue frameworks, the front-end uses Mapbox GL [29, 30] technology to satisfy the demands of dynamic meteorological big data visualization services while considering compression, slicing, display, and other factors. The proposed scheme offers a mid-platform support mode for visualizing meteorological grid data that have been published in the meteorological visualization column of the China Meteorological Data Service Center, via a mobile app, or on WeChat. This scheme provides a fast and convenient solution for rapidly visualizing and rendering grid data and can serve as a reference for grid data visualization applications in other industries.

2. Methodology

The proposed system includes big data processing, transmission, and page rendering and involves technologies such as data analysis, compression, browser transmission, page data restoration and splicing, as well as WebGL [31] rendering. The detailed design is shown in Figure 1.

Data processing and compression: Red-green-blue (RGB) channels are used to compress and store the data in high-low 8-bit PNG files to maintain data accuracy to the fullest extent possible during the transmission process.

Data slicing and transmission: Based on pyramid slicing technology, slicing is performed with a specified minimum scaling resolution, and distributed multithreading is used during transmission to increase the timeliness of transmitting the data from the server to the page terminal.

Data visualization rendering: The browser obtains slices and performs slice stitching, while the Mapbox GL component based on WebGL technology realizes fast dynamic rendering.

2.1. Data Processing and Compression. First, the source GRIB file is transformed into a float array by PYGRID [32] in Python and is generally stored as 4-byte data with a maximum legend display resolution of 0.1. The data can be retained to 1 significant digit before being stored, that is, the original value O is multiplied by 10 and rounded to obtain the integer C. The image is saved as an RGB compressed image with 2 bytes in the G and B channels in the range of [−32768, 32767]. To minimize data loss, the image is stored in the PNG format by adopting an LZ77-derived algorithm for file compression, thus resulting in a small data volume, a high compression ratio, and no data loss.

$$O = \begin{bmatrix} I_{11} & \cdots & I_{1W} \\ \vdots & \ddots & \vdots \\ I_{H1} & \cdots & I_{HW} \end{bmatrix}; \qquad (1)$$
$$C = \mathrm{round}(O \times 10).$$

The high 8-bit and low 8-bit values of the converted data C are stored in the G and B channels, respectively. The specific operations are shown as follows:

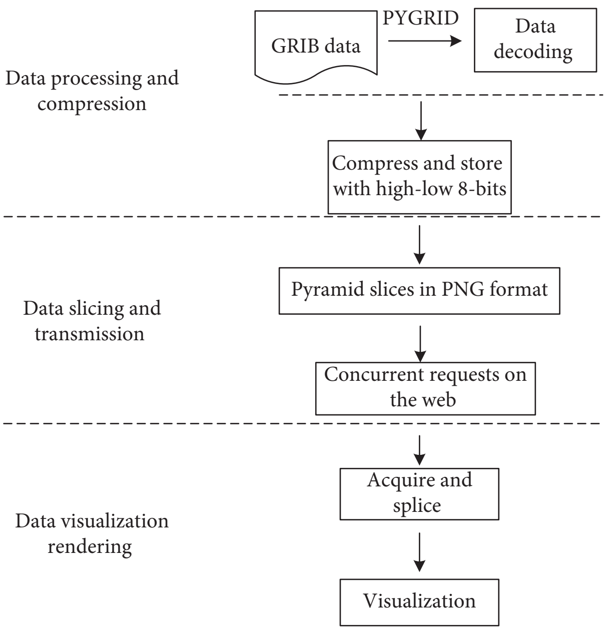

GRIB data
PYGRID
Data decoding
Data processing and compression
Compress and store with high-low 8-bits
Pyramid slices in PNG format
Data slicing and transmission
Concurrent requests on the web
Acquire and splice
Data visualization rendering
Visualization

FIGURE 1: Design of the system.

$$
\begin{aligned}
U &= \operatorname{trunc}\left(\frac{C}{256}\right);\\
D &= C\%256,\\
G_{-N} &= \begin{bmatrix} [R_{11}, G_{11}, B_{11}] & \cdots & [R_{1W}, G_{1W}, B_{1W}] \\ \vdots & \vdots & \vdots \\ [R_{H1}, G_{H1}, B_{H1}] & \cdots & [R_{HW}, G_{HW}, B_{HW}] \end{bmatrix},\\
G_{-N}&[:\ ][:\ ][:, :, 1] = U;\\
G_{-N}&[:\ ][:\ ][:, :, 2] = D.
\end{aligned}
\tag{2}
$$

where U and D represent the intermediate values, while G_N denotes the newly created PNG image used to store the corresponding compressed values.

To improve the efficiency of page data transmission, the data are processed in accordance with the image pyramid model. The compressed PNG file is scaled down using the image pyramid approach with bicubic interpolation. The tilemap pyramid model is a multiresolution hierarchical model. The resolution decreases from the bottom to the top of the tile pyramid; however, the geographical range of the representation remains constant. The image is first scaled and then filled with squares in accordance with the tilemap pyramid model.

2.2. Data Slicing and Transmission. The original image serves as layer 0 of the pyramid and is scaled by 2×, 4×, 8×, and 16× via bicubic interpolation [33]. In numerical analysis, bicubic interpolation is the most commonly applied interpolation method in two-dimensional space.

It is assumed that if the source image G has a size of $M \times \mathrm{N}$ and the scaled target image g has a size of $m \times n$, the coordinates of g on G can be calculated using formula (3).

$$
\begin{aligned}
x\prime &= x \times \frac{M}{m},\\
G\_N[:\ ][:\ ][:, :, 1] &= U;\\
G\_N[:\ ][:\ ][:, :, 2] &= D.
\end{aligned}
\tag{3}
$$

As shown in Figure 2, (x', y') denotes the location of a point P' in the original image, which corresponds to the position $g(x, y)$ in the compressed image; the value at this point is obtained by interpolating from the pixel values at the 16 neighborhood points (P00, . . ., P33). If the position of P11 is (x, y), then the position of P' can be expressed as $(x + u, y + v)$, where u and v represent the fractional parts of the pixel coordinates.

Once the influence weights of the 16 neighborhood points relative to point $P^{'}$ have been calculated, the value of $P^{'}$ can be obtained and mapped to the scaled image g. The basic function for bicubic interpolation is shown in Formula (2), where $a = -0.5$:

$$
W(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^3 + 1, & \text{for } |x| \le 1,\\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & \text{for } 1 < |x| < 2,\\ 0, & \text{otherwise.} \end{cases}
\tag{4}
$$

When the rows and columns are separated, the distance between the pixel value to be calculated and the known pixel value P00 in Figure 2 can be expressed as $(1 + u, 1 + v)$; hence, the abscissa-coordinate weight of the P00 pair is $W(1 + U)$ and the ordinate weight is $W(1 + V)$, yielding a corresponding value contribution of $\mathrm{Pix}_{00} \times W(1 + u) \times W(1 + v)$. The contributions from the other 15 points can be calculated similarly. Finally, the pixel value at the point in the map scaled to the image G can be calculated using formula (5).

$$
G(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} \mathrm{Pix}_{ij} \times W(i) \times W(j). \tag{5}
$$

Notably, a browser can process only a limited number of concurrent requests for the same domain name, which restricts the number of simultaneous requests that can be served during page rendering, resulting in requests queuing or timing out. This process occurs on GIS service websites such as Google and Baidu Maps, which add subdomains and domain dashes to increase the number of concurrent requests that can be served [34]. However, given the increased difficulty of DNS resolution for an excessive number of domain names, the concurrency of each secondary domain name should be limited to 2–4.

The scheme proposed here, which is based on a B/S service framework, uses an Nginx server, which is a lightweight Web server/reverse proxy server, and an e-mail (IMAP/POP3) proxy server distributed under a BSD-like

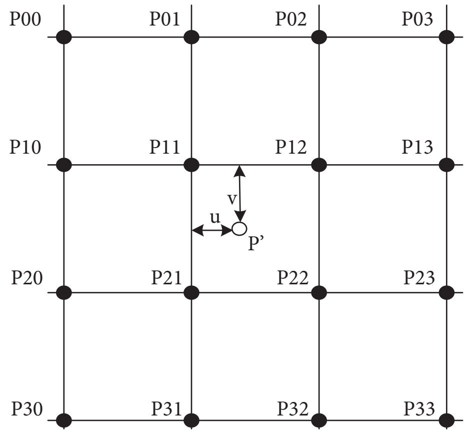


Figure 2: Bicubic interpolation diagram.

protocol, as proxies, thus occupying less memory and enabling high concurrency. Moreover, Nginx supports the gzip compression function, which can be used to compress the website CSS, JS, XML, and HTML files during transmission, thus boosting the access speed and optimizing Nginx performance.

2.3. Data Visualization Rendering. During the rendering process, the number of slices should be adjusted to account for different screen resolutions and map magnifications. The coordinates of the map are translated into screen coordinates to realize the transformation between the longitude and latitude coordinates of the map and the screen coordinates. Once the scale problem has been resolved, the spatial information of the latitude and longitude ranges on the earth corresponding to the GRIB file is transformed to the range of the screen. Based on the size of each layer of the image pyramid in the GRIB file, the pyramid slice that offers the best rendering is finally selected (Figure 3).

The distance between any two points expressed in terms of longitude and latitude can be calculated using the following flow formula:

$$\text{Distance}\left((\text{lons}, \text{lats}), (\text{lone}, \text{late})\right) = R * \cos^{-1}\begin{bmatrix} \cos(\text{lats}) * \cos(\text{late}) * \cos(\text{lons} - \text{lone}) \\ +\sin(\text{lats}) * \sin(\text{late}) \end{bmatrix}, \tag{6}$$

where R is the radius of the earth, which is approximately equal to 6371.0 km.

The corresponding screen distance can be calculated from the Mapbox scale.

$$\text{screen}D = \text{Distance}\left((\text{lons}, \text{lats}), (\text{lone}, \text{late})\right) * \text{scale}. \tag{7}$$

Finally, the layer number of the selected slices can be determined based on the diagonal width of each layer image in the pyramid.

Here, orgSZ denotes the diagonal image size in the compressed resource file (Figure 4). The result of dividing "orgSZ" by "screenD" indicates how many times larger the visible range on the screen is than the size of the compressed file (layer 0). Because consecutive layers differ in size by a factor of 2, the indicated layer is determined by rounding up the value to a power of 2.

$$\text{Pyramid layer} = \sqrt[2]{\frac{\text{org}SZ}{\text{screen}D}}. \tag{8}$$

Thereafter, the latitude and longitude ranges on the screen are obtained, namely, the upper left (lu_x, lu_y) and lower right corner (rd_x, rd_y), and the tile index is obtained in accordance with the flow progress.

$$\begin{aligned} \text{dis}x &= \text{tile}_x - \text{lu}_x, \\ \text{col}_{\text{start}} &= \text{ceil}\left(\frac{\text{dis}x}{\text{interval}}\right), \quad \text{if dis}x > 0 \text{ else} = 0, \\ \text{col}_{\text{end}} &= \text{col}_{\text{start}} + \text{ceil}\left(\frac{rd_x - lu_x}{\text{interval}}\right), \end{aligned} \tag{9}$$

where tile_x is the upper left corner of the data in the x direction, dis$x$ denotes the distance between the upper left corner of the screen and the upper left corner of the data in the x direction, $\text{col}_{\text{start}}$ represents the index of the starting tile column, col_{end} represents the index of the ending tile row, interval denotes the length and width of a tile, and ceil denotes the operation of rounding up. Similarly, the row and column indices of the starting and ending tiles can be obtained. Because the tile coordinate range is greater than the screen coordinate range, all tiles need to be offset. Starting with the upper left corner of the screen, the position offsets in the CSS file can be obtained by calculating the difference between the pixel coordinates of the upper left corner of each tile and those of the upper left corner of the screen [35].

Based on the Vue development framework, the proposed scheme comprehensively considers the spatiotemporal attributes of meteorological big data and the demands of fast rendering, utilizes Mapbox to support geographical information services, and applies WebGL high-performance front-end rendering technology for data visualization. Vue is a progressive and high-performance JavaScript framework for front-end page display that uses view layer rendering as its core. Vue utilizes a component mechanism, a routing mechanism, and a state management mechanism to quickly realize front-end high-frequency Document Object Model (DOM) operations and efficient page interactions. Due to the use of the NodeJS service and the utilization of Node Package Manager (NPM) to install the Vue command-line interface (Vue CLI), the framework can be built quickly. Mapbox, which has corresponding GIS engines for different platforms (e.g., PC and mobile), is an efficient WebGIS development framework. As a Mapbox component [36],

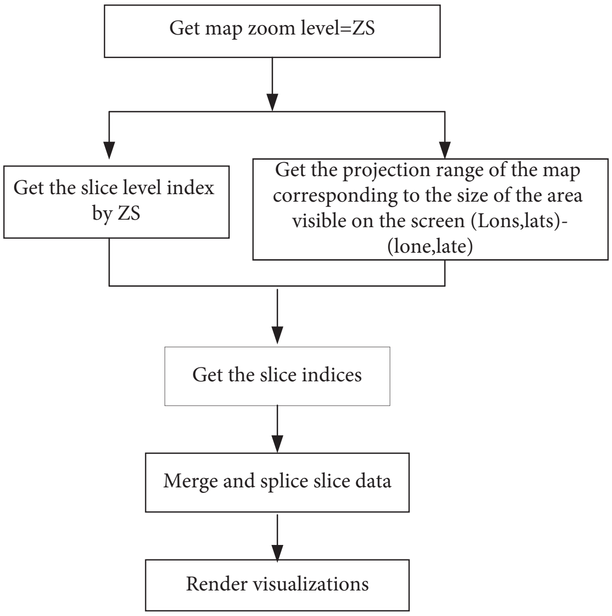

Get map zoom level=ZS
Get the slice level index by ZS
Get the projection range of the map corresponding to the size of the area visible on the screen (Lons,lats)-(lone,late)
Get the slice indices
Merge and splice slice data
Render visualizations

Figure 3: The rendering process.

Mapbox GL has been used for HTML5 web development. Mapbox GL is a JavaScript library that can render a large number of map elements while allowing for smooth interactions and animation.

3. Experiments and Analysis

To examine the performance and effectiveness of the whole proposed process, high-low 8-bit compression was developed in Python 3.7, and the grid data (GRIB2) were decoded into arrays by PYGRID. The web page was developed in Visual Studio XCODE, and the web server was deployed using Nginx 1.16.0. The progress server and web server were 64-bit Linux servers, and the CPU for the experiment was an eight-core Intel Core i5 @ 2.30 GHz with 16 GB of memory.

3.1. Data Processing. The data used in the experiments were obtained from the CMA Multisource Precipitation Analysis System (CMPAS) [37], which are available through the China Meteorological Data Service Center (http://data.cma.cn). The data include latitude and longitude ranges of 70–140° E and 15–60° N. The experiments included 24 hourly precipitation fusion products with a resolution of 1 km $*$ 1 km from 00 : 00 to 23 : 00 on July 20, 2021; these data were chosen as an example to evaluate and compare the data processing with the foreground display. The data included 7000 latitude points and 4500 longitude points, with over 300 million data points in a single file, and the single file size of the hourly precipitation fusion product was 101.3 MB.

Before image slicing, the image in each layer was transformed into a square. The longest side length was taken as the side length to create a new square canvas. The upper left corner of the original image was overlapped with the upper left corner of the square, and the remainder of the square was filled in white. Thus, layer 0 ($7000 * 7000$), layer 1 ($3500 * 3500$), layer 2 ($1750 * 1750$), layer 3 ($875 * 875$), and layer 4 ($437 * 437$) were obtained.

Table 1 shows the minimum and maximum compression ratios of the five data layers, which reached 30 and 95, respectively, after PNG compression and conversion. The average file size of the 24 experimental files was calculated. The degree value equivalent to the pixel interval corresponding to each scale ratio in the five layers of the pyramid was determined based on the scaling coefficient. The pixel size in layer 4 was taken as the size of a single tile to slice the data from the other four layers. The maximum size of a single-slice file was less than 90.7 kB, sufficient to guarantee fast transmission. When the sizes of the slice files were compared, it was found that the slice files tended to be larger in areas with rainfall due to the data distribution in these locations.

3.2. Transmission Efficiency. In the experiments, the time it took for all slices in the compressed file (layer 0) to be transmitted from the server to the browser was a thousand times faster than the time it took for the original file to be transmitted when 256 concurrent channels were used. The compressed PNG transmission efficiency is compared between the full-size image and its slices (Figure 5).

The maximum amount of data that could be requested by the browser includes the map and data slices. In these experiments, the actual maximum amount requested was less than 60 because the visible slices were determined according to the range of the map. Sixty-four concurrent channels with eight subdomains were opened for the requests.

3.3. Rendering Efficiency. Currently, there are three main online WebGIS rendering methods for processing GRIB, NetCDF, and other file formats: GeoJSON processing, binary compression, and grayscale compression. Table 2 lists

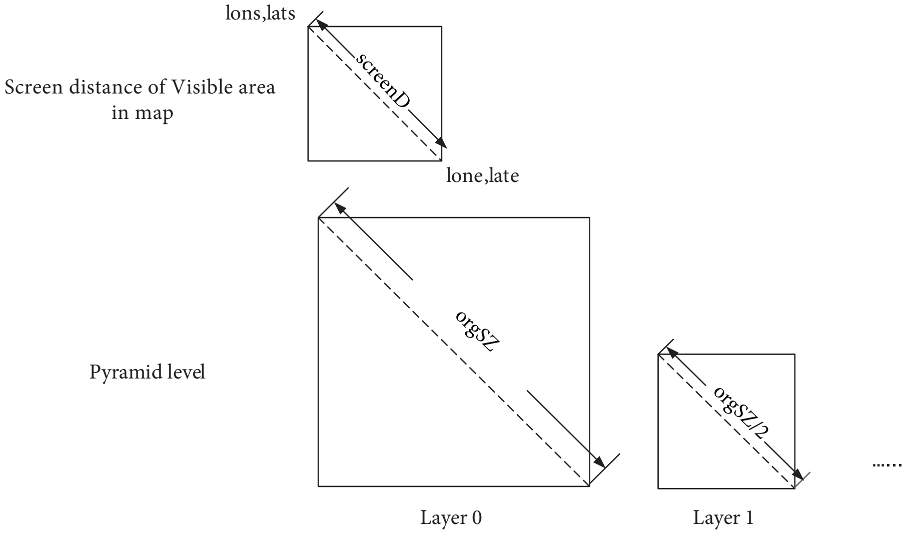


Figure 4: The area relation between the visible screen range and the diagonal width of each layer.

Table 1: Slices after compression.

Layer	Resolution	Compression to PNG (kB)	Range of the single-slice file size (kB)	Number of slices	Distance between adjacent grid points
0	7000 × 7000	1611.9	[0.6, 79.3]	256	0.01° (1 km)
1	3500 × 3500	543.0	[0.6, 57.3]	64	0.02° (2 km)
2	1750 × 1750	174.3	[0.9, 35.8]	16	0.04° (4 km)
3	875 × 875	56.4	[9.5, 20.4]	4	0.08° (8 km)
4	437 × 437	17.6	[17.6, 17.6]	1	0.16° (16 km)

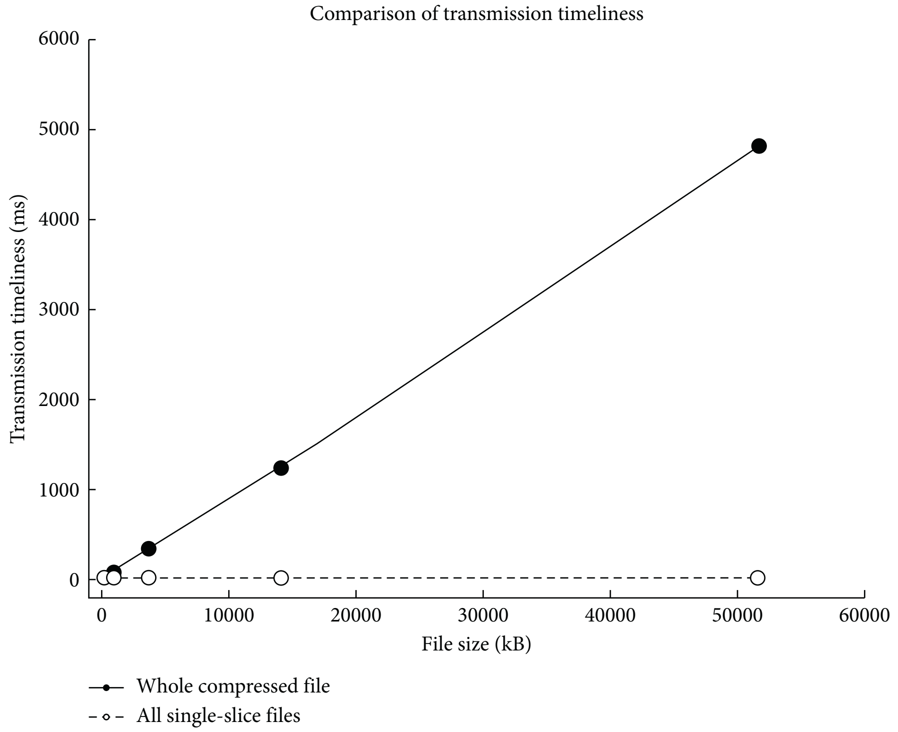


Figure 5: Comparison of transmission timeliness between the whole compression file and the slices.

Table 2: Comparison of the various treatment results and the rendering efficiency.

Treatment	Result (MB)	Rendering time
GeoJSON transformation	253	2.5 s
Binary compression	76	485 ms
Grayscale image compression	89	287 ms
High-low 8-bit compression	1.3	145 ms

the rendering efficiencies of the above methods and the method in this article for a 120 MB file. The same WebGL and WebGIS rendering technology was used for all processes. The results indicate that the rendering efficiency of the high-low 8-bit compression method reduced the time consumption to the order of milliseconds, thus making this the optimal method.

4. Conclusion and Future Work

Due to climate and weather phenomena such as global climate change and the frequent occurrence of extreme weather, the demands for meteorological services have been increasing in relation to various social activities and industries. Accordingly, there is a need for methods of visualizing the spatiotemporal characteristics of meteorological big data for disaster prevention and mitigation for various social activities and industries. Based on the Vue architecture of HTML5, the scheme proposed in this article can be used to quickly visualize grid data of approximately 100 MB in size with PC and mobile browsers as carriers. The multiterminal rendering of technical data is accomplished by combining Python, Node.js, HTML5, Mapbox, and other technologies. The proposed process can support efficient WebGIS rendering of various kinds of large grid data files, thus providing a solution for quickly visualizing industrial data and spatial big data after fusion and improving the efficiency of installation-free visualization of grid big data in browsers.

Data compression: Various compression algorithms, including binary compression, grayscale map compression, and high-low 8-bit compression, were compared in terms of the compression ratio and the loss ratio. High-low 8-bit compression was selected because it enables the visual display of meteorological values accumulated over periods such as years, months, and days. However, the use of this compression algorithm is limited due to the large scale of the accumulated values and the need for visualization accuracy up to three significant digits after the decimal point.

Slicing: Pyramid slicing met the efficiency requirements of this study. The visualization efficiency could be further improved by adopting other algorithms, such as the quadtree algorithm, based on specific requirements.

Transmission and rendering: In this study, the image rendering data were separated from the original data. The data of layer 0 were original, while the data in the other layers were rendered on the web page, thus preserving the eigenvalues in the visualization, which is preferable to grid pumping and visualization. Furthermore, the rendering method based on the WebGL technical framework fully uses the capabilities of the browser, reducing the pressure on the server and taking advantage of combining cloud computing with sliding windows.

Data Availability

The data were saved as Grib2 format, which can be downloaded from http://image.data.cma.cn/test/Z_SURF_C_BABJ_P_CMPA_RT_CHN_0P01_HOR-PRE-20210720.rar, and the datafile can be decoded by the software of Panoply (https://www.giss.nasa.gov/tools/panoply/).

Conflicts of Interest

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

Authors' Contributions

He-Ping Yang and Ying-Rui Sun conceptualized the study and developed the original research program; Nan Chen and Jing-Hua Chen analyzed the data; Qi Wang and Ming Yang handled the data; He-Ping Yang and Xiao-Wei Jiang wrote the original draft; Zi-Mo Huo and Ming-Nong Feng compared the results of the different methods. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2016YFA0600301); a Major Project of Zhejiang Province, China (Grant 2021C02036); and the Key Research and Development Project of China (Grant 2018YFC1505601). The authors are thankful to Profs. Hui Gao, Qingyun Zhang, and Xia Xu for helpful discussions.

References

[1] M. B. G. Martín, "Weather, climate and tourism a geographical perspective," *Annals of Tourism Research*, vol. 32, no. 3, pp. 571–591, 2005.

[2] L. Bright and D. Maier, *Deriving and Managing Data Products in an Environmental Observation and Forecasting System," the 2005 CIDR Conference*, pp. 162–173, 2005.

[3] T. N. Palmer, J. Barkmeijer, R. Buizza, and T. Petroliagis, "The ECMWF ensemble prediction system," *Meteorological Applications*, vol. 4, no. 4, pp. 301–304, 1997.

[4] Y. Ota, J. C. Derber, E. Kalnay et al., "Ensemble-based observation impact estimates using the NCEP GFS," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 65, no. 1, p. 20038, 2013.

[5] R. Zhang and X. Shen, "On the development of the GRAPES——A new generation of the national operational NWP system in China," *Science Bulletin*, vol. 53, no. 22, pp. 3429–3432, 2008.

[6] J He, K Yang, W Tang et al., "The first high-resolution meteorological forcing dataset for land process studies over China," *Scientific Data*, vol. 7, no. 1, pp. 25–11, 2020.

[7] S. E. Haupt and B. Kosović, "Variable generation power forecasting as a big data problem," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 2, pp. 725–732, 2016.

[8] C. S. Zender, "Analysis of self-describing gridded geoscience data with netCDF Operators (NCO)," *Environmental Modelling & Software*, vol. 23, no. 10-11, pp. 1338–1342, 2008.

[9] P. Berrisford, R. Brugge, L. Steenman-Clark, and D. Li, "The UGAMP use and diagnosis of the ECMWF meteorological analyses," *Systems Analysis Modelling Simulation*, vol. 42, no. 11, pp. 1615–1621, 2002.

[10] M. Ninyerola, X. Pons, and J. M. Roure, "A methodological approach of climatological modelling of air temperature and precipitation through GIS techniques," *International Journal of Climatology*, vol. 20, no. 14, pp. 1823–1841, 2000.

[11] R. B. Schmunk, *Panoply netCDF, HDF and GRIB Data Viewer*, NASA Goddard Institute for Space Studies, 2017.

[12] Y. Q. Wang, "MeteoInfo: GIS software for meteorological data visualization and analysis," *Meteorological Applications*, vol. 21, no. 2, pp. 360–368, 2014.

[13] H Wu, W Zheng, B Luo et al., "Decision making meteorological services system based on geographic information system," *International Joint Conference on Computational Sciences and Optimization*, vol. 2, pp. 53–55, 2009.

[14] F. Berman, A. Chien, K. Cooper et al., "The GrADS project: software support for high-level grid application development," *International Journal of High Performance Computing Applications*, vol. 15, no. 4, pp. 327–344, 2001.

[15] D. Murray, J. McWhirter, Y. Ho et al., "The IDV at 5: new features and future plans," *Proceedings of 25th Conference on International Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, vol. 7, no. 2, 2009.

[16] R. Nogueira and E. M. Cutrim, "Applications of "integrated data viewer" (IDV) in the classroom," *Advances in Geosciences*, vol. 8, pp. 63–67, 2006.

[17] J. Gaigalas, L Di, and Z. Sun, "Advanced cyberinfrastructure to enable search of big climate datasets in THREDDS," *ISPRS International Journal of Geo-Information*, vol. 8, no. 11, p. 494, 2019.

[18] M. Linaje, J. C. Preciado, and F. Sanchez-Figueroa, "Engineering rich internet application user interfaces over legacy web models," *IEEE internet computing*, vol. 11, no. 6, pp. 53–59, 2007.

[19] S. Popelka, A. Vondrakova, and P. Hujnakova, "Eye-tracking evaluation of weather web maps," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, p. 256, 2019.

[20] S. Popić, D. Pezer, B. Mrazovac et al., "Performance evaluation of using protocol buffers in the internet of things communication," in *International Conference on Smart Systems and Technologies (SST)*, pp. 261–265, 2016.

[21] X. Shang, *A Study on Efficient Vector Mapping with Vector Tiles Based on Cloud Server Architecture*, Graduate Studies, 2015.

[22] N. Yap, M. Gong, R. K. Naha, and A. Mahanti, "Machine learning-based modelling for museum visitations prediction," in *Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–7, Shenzhen, China, 2020.

[23] L. S. Hsu and O. Obe, *PostGIS in Action*, Manning Publications Co., 2015.

[24] B. N. Hill, *An Analysis of the Factors that Influence the Sargassum Migratory Loop*, Diss., 2016.

[25] K. S. Yarygin, B. A. Kovarsky, T. S. Bibikova, D. S. Melnikov, A. V. Tyakht, and D. G. Alexeev, "ResistoMap-online visualization of human gut microbiota antibiotic resistome," *Bioinformatics*, vol. 33, no. 14, pp. 2205-2206, 2017.

[26] V. Lytvyn, O. Pashchetnyk, O. Klymovych et al., "Assessment of the hydro-meteorological conditions impact on the combat troops operations preparation and conduct in the geo-information subsystem of the automated battlefield management system," *CEUR Workshop Proceedings*, vol. 1, pp. 1063–1076, 2021.

[27] R. S. Prastika, A. N. Afandi, and D. Prihanto, "Mitigation of the alternative energy for the wind farm center considering temperature and wind speed," *MATEC Web of Conferences*, vol. 204, p. 04013, 2018.

[28] Y. Jiang, S. Han, C. Shi, T. Gao, H. Zhen, and X. Liu, "Evaluation of HRCLDAS and ERA5 datasets for near-surface wind over hainan island and south China sea," *Atmosphere*, vol. 12, no. 6, p. 766, 2021.

[29] B. Kastanakis, *Mapbox Cookbook*, Packt Publishing Ltd, 2016.

[30] G. Qiu and J. Chen, *Web-based 3D Map Visualization Using WebGL," 2018 13th IEEE Conference On Industrial Electronics And Applications (ICIEA)*, pp. 759–763, 2018.

[31] T. Parisi, *WebGL: Up and Running*, O'Reilly Media, Inc., 2012.

[32] M. Samir, M. Azab, M. R. M. Rizk, and N. Sadek, "PYGRID: a software development and assessment framework for grid-aware software defined networking," *International Journal of Network Management*, vol. 28, no. 5, p. e2033, 2018.

[33] S. Gao and V. Gruev, "Bilinear and bicubic interpolation methods for division of focal plane polarimeters," *Optics Express*, vol. 19, no. 27, pp. 26161–26173, 2011.

[34] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen, "Soundsquatting: uncovering the use of homophones in domain squatting," *Lecture Notes in Computer Science*, vol. 8783, pp. 291–308, 2014.

[35] W. Busch, B. T. Moore, B. Martsberger et al., "A microfluidic device and computational platform for high-throughput live imaging of gene expression," *Nature Methods*, vol. 9, no. 11, pp. 1101–1106, 2012.

[36] P. Halliday, *Vue. Js 2 Design Patterns and Best Practices: Build enterprise-ready, Modular Vue. Js Applications with Vuex and Nuxt*, Packt Publishing Ltd, 2018.

[37] Y. Wang, K. Dai, Z. Zong et al., "Quantitative precipitation forecasting using multi-model blending with supplemental grid points: experiments and prospects in China," *Journal of Meteorological Research*, vol. 35, no. 3, pp. 521–536, 2021.

Hindawi
Complexity
Volume 2022, Article ID 9504055, 9 pages
https://doi.org/10.1155/2022/9504055

Research Article

Disease Identification of Lentinus Edodes Sticks Based on Deep Learning Model

Dawei Zu,[1] **Feng Zhang**,[1] **Qiulan Wu**,[1] **Wenyan Wang**,[1] **Zimeng Yang**,[1] **and Zhengpeng Hu**[2]

[1]*School of Information Science & Engineering, Shandong Agricultural University, Taian 271018, China*
[2]*Shandong Qihe Bio-Technology Limited Company, Zibo 255100, China*

Correspondence should be addressed to Qiulan Wu; zxylsg@sdau.edu.cn

Received 10 October 2021; Revised 10 December 2021; Accepted 10 February 2022; Published 4 March 2022

Academic Editor: Atila Bueno

Lentinus edodes sticks are susceptible to mold infection during the culture process, and manual identification of infected sticks is heavy, untimely, and inaccurate. Aiming to solve this problem, this paper proposes a method for identifying infected Lentinus edodes sticks based on improved ResNeXt-50(32 × 4d) deep transfer learning. First, a dataset of Lentinus edodes stick diseases was constructed. Second, based on the ResNeXt-50(32 × 4d) model and the pretraining weight of the ImageNet dataset, the influence of pretraining weight parameters on recognition accuracy was studied. Finally, six fine-tuning strategies of the fully connected layer were designed to modify the fully connected layer of ResNeXt-50(32 × 4d). The experimental results show that the recognition accuracy of the method proposed in this paper can reach 94.27%, which is higher than the Vgg16, GoogLeNet, ResNet50, and MobileNet v2 models by 8.47%, 6.49%, 4.68%, and 9.38%, respectively, and the F1-score can reach 0.9422. The improved method proposed in this paper can reduce the calculation pressure and overfitting problem of the model, improve the accuracy of the model in the identification of Lentinus edodes stick mold diseases, and provide an effective solution for the selection of diseased sticks.

1. Introduction

As an important carrier for the production of Lentinus edodes, Lentinus edodes sticks are often infected by mold diseases [1], which results in large economic losses. Currently, the selection of diseased sticks is still at the level of empirical management, which requires an inspector to manually extract the Lentinus edodes sticks and judge whether they are diseased. This traditional method has some problems, such as sticks being missed by inspectors and untimely selection of diseased Lentinus edodes sticks, which can easily lead to mold diffusion. At the same time, research on the automatic identification of Lentinus edodes stick diseases has been very rare, and there is a lack of specific identification models. Therefore, it is necessary to collect and process Lentinus edodes stick disease images during inoculation, precultivation, cultivation, cold storage, and other steps, to research identification technology of Lentinus edodes stick diseases, and to achieve accurate identification and judgment of Lentinus edodes stick diseases. It is of great significance to reduce the spread of Lentinus edodes stick diseases, improve the yield and quality of Lentinus edodes, drive the large-scale development of the Lentinus edodes industry, and improve economic benefits.

Since the large-scale development of deep learning [2–4], an increasing number of researchers have introduced deep learning into the field of crop disease image detection [5–10]. Compared with the traditional image recognition method, this new nondestructive testing technology avoids the complex image data preprocessing process by inputting the image directly into the network. Deep learning uses the method of automatic feature extraction to combine low-level features into high-level abstract visual features. It can quickly and nondestructively identify crop diseases within the visible light range without using hyperspectral imaging technology.

It has higher accuracy, faster detection speed, and better stability.

At present, deep learning research in the field of agricultural disease identification has become a hot spot for the application of deep learning. Fan Xiangpeng et al. [11] optimized the convolutional neural network, trained, and tested corn disease images under complex backgrounds, and the recognition rate reached 97.10%. Mohanty S P et al. [12] used AlexNet and GoogLeNet to classify and recognize 54306 plant disease images in the PlantVillage dataset, and the model accuracy was up to 99.35%. Yang Sen et al. [13] used VGG-16 as the feature extractor of the Faster R–CNN model through the method of deep transfer learning, made the clustering method to build a composite dictionary for the mixed features of color and SIFT, and achieved a recognition accuracy of 90.83% of diseased potato leaves. Although deep learning has achieved quite good results in the field of crop disease identification, relevant literature on deep learning in the identification of Lentinus edodes stick diseases has not been found in previous studies.

To solve the above problems, this paper proposes a method based on the ResNeXt-50(32 × 4d) deep transfer learning method for Lentinus edodes sticks disease identification. The main contributions are as follows: (1) this paper takes the lead in applying the deep learning model to the identification of Lentinus edodes sticks infection, which makes up for the gap of domestic deep learning in the disease identification of Lentinus edodes sticks. (2) For the Lentinus edodes stick disease dataset, the fully connected layer of ResNeXt-50(32 × 4d) model is redesigned to improve the recognition accuracy. (3) The disease identification method of Lentinus edodes sticks studied in this paper can be extended to the disease identification of other bagged edible fungi.

2. Materials and Methods

2.1. Lentinus Edodes Sticks Diseases Dataset. Shandong Qihe Biotechnology Limited Company produces approximately 700 thousand Lentinus edodes sticks every year, which are infected by diseases such as Aspergillus flavus, Trichoderm viride, and Neurospora, resulting in a direct economic loss of 9 million yuan. In the Qihe biological intelligence factory, the images of Lentinus edodes sticks infected by mold in the culture shed were collected manually, and they were divided into Aspergillus flavus diseased sticks, Trichoderm viride diseased sticks, Neurospora diseased sticks, and normal Lentinus edodes sticks based on the type of mold disease (see Figure 1).

In this paper, 942 images of Aspergillus flavus diseased sticks, 893 images of Trichoderm viride diseased sticks, 664 images of Neurospora diseased sticks, and 1179 images of normal Lentinus edodes sticks were collected, for a total of 3678 images. Because the amount of image data of Lentinus edodes stick diseases is relatively low, this study uses image enhancement methods [14] such as random rotation and horizontal flip to increase the diversity of the samples and builds a Lentinus edodes stick disease dataset.

2.2. ResNeXt-50(32 × 4d) Network. The traditional method to improve the accuracy of model recognition is to deepen or widen the network. However, with the increase in the number of hyperparameters (such as channel number and filter size), the difficulty of network design and computational overhead will increase. The ResNeXt-50(32 × 4d) network [15] combines the stacking strategy of the ResNet network [16] and the packet convolutional strategy of the inception network [17]. It uses a residual block with the same topology to stack in parallel instead of the original three-layer convolutional residual block of ResNet. Compared with the ResNet network, the ResNeXt-50(32 × 4d) network can not only improve the accuracy without increasing the complexity of parameters but also reduce the number of hyperparameters to achieve a better classification effect.

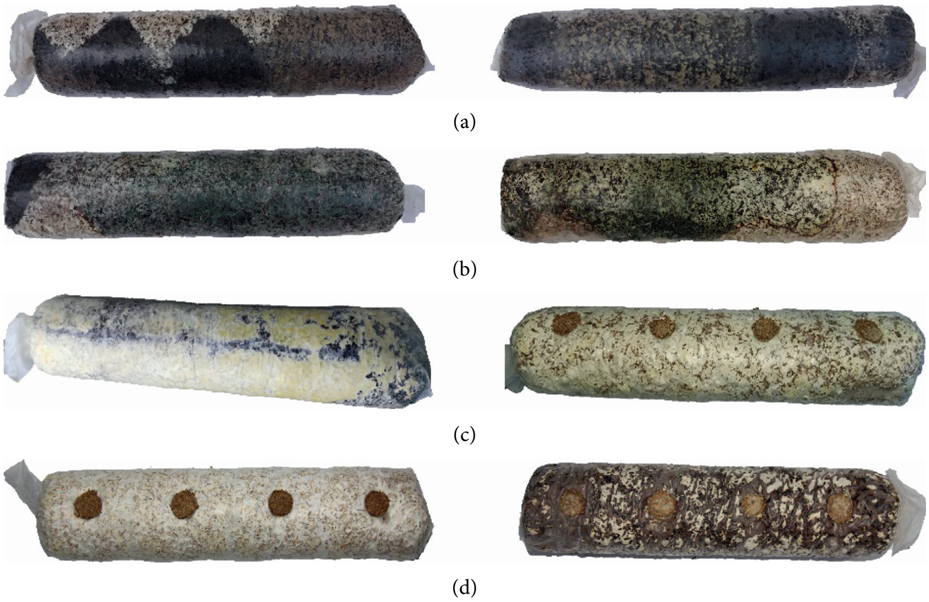
ResNeXt network is composed of a series of residual blocks, and each residual block has the same topological structure [18]. The residual block of ResNeXt-50(32 × 4d) network conv2 is takenn as an example (see Figure 2), the ResNeXt residual block is divided into 32 groups for the image feature matrix with 256 input channels. For each grouping, first, the image feature matrix is reduced by 4 convolutional kernels with 256 channels and 1×1 in size. Second, it is convolved by 4 convolutional kernels with 4 channels and 3×3 in size. Then, it uses 256 channels of 4 and a size of 1×1 convolutional kernel to increase the dimensionality of the output. Finally, the output image feature matrix of each group is added, and then, the image feature matrix is added with 256 input channels to obtain the final output image matrix.

The ResNeXt residual block implements the splitting-transforming-merging strategy, which is expressed as

$$F(x) = x + \sum_{i=1}^{C} T_{\mathrm{i}}(x). \tag{1}$$

Here, T_i has the same topological structure, and C represents the number of groups of each ResNeXt residual block and $C = 32$.

ResNeXt-50(32 × 4d) network structure is shown in Figure 3. Conv2, conv3, conv4, and conv5 are composed of 3, 4, 6, and 3 residual blocks, respectively. The design of residual blocks follows two rules: (1) if a characteristic diagram of the same size is generated, these groups share the same hyperparameters (convolutional kernel size and number of channels); (2) when the size of the feature map is down-sampled twice, the number of channels in the feature map needs to be doubled. For example, when the number of channels in the residual block of conv2 is 256, it is divided into 32 groups, and the number of channels in each group is 4; when the number of channels in the residual block of conv3 is 512, it is divided into 32 groups, and the number of channels in each group is 8. By analogy, the number of channels gradually doubles.


(a)

(b)

(c)

(d)

Figure 1: Examples of diseased Lentinus edodes sticks and normal Lentinus edodes stick images. (a) Aspergillus flavus diseased sticks. (b) Trichoderm viride diseased sticks. (c) Neurospora diseased sticks. (d) Normal Lentinus edodes sticks.

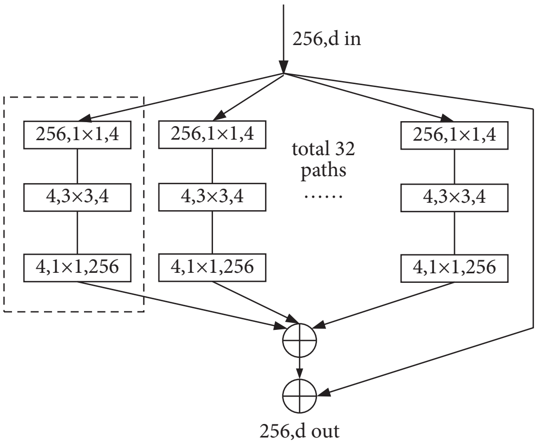


Figure 2: ResNeXt block.

After the feature calculation of the residual neural network, the fully connected layer flattens the incoming feature vectors into one-dimensional vectors and then uses these feature vectors as input to calculate the probability value of each sample category.

stag	output	ResNeXt–50 (32x4d)
conv1	112x112	7x7,64,stride 2
conv2	56x56	3x3 max pool,stride 2
		[1x1,128 3x3,128, C = 32 1x1,256] x3
conv3	28x28	[1x1,256 3x3,256, C = 32 1x1,512] x4
conv4	14x14	[1x1,512 3x3,512, C = 32 1x1,1024] x6
conv5	7x7	[1x1,1024 3x3,1024, C = 32 1x1,2048] x3
	1x1	global average pool 4–d fc,softmax

Figure 3: Structure of ResNeXt-50(32 × 4d) network.

2.3. Transfer Learning. With the rapid development of image recognition technology, the demand for labeled image data is growing. However, labeling image data is a repetitive and cumbersome task. At present, although there are high-precision image datasets and application scenes, it is time-consuming to establish a new model for each scene, and there are not enough labeled image data. In recent years, with the establishment of large datasets such as ImageNet [19], there have been an increasing number of publicly available annotated image data. As the largest image recognition task database in the world, there are more than 14 million labeled images in the ImageNet dataset, among which there are a large number of plant disease images.

Based on these plant disease image data, multiple deep neural network models have been trained, and the complete training parameters and model weights have been saved.

In 2014, Yosinski [20] and others took the lead in exploring the transitivity of deep neural networks and reached three main conclusions as follows:

(1) The first few layers of the neural network learn the basic features of the image, and the trained parameters based on these features have a good recognition effect.

(2) The result of fine-tuning the deep transfer network is better than that of the initial training.

(3) Fine-tuning can overcome the differences between data.

In this study, ResNeXt-50(32 × 4d) model is used for transfer learning [21], in which the pretraining weight is trained on the ImageNet data set. Because the weight trained by ImageNet image data has a strong ability to express the underlying features during transfer learning and can well deal with the same type of image recognition tasks, therefore, based on transfer learning, the trained model weights are used, and the model is fine-tuned, which can not only improve the robustness and generalization of the model but also save training time by not training the network from scratch [22–24].

3. Results and Discussion

3.1. Evaluation Criteria. In order to evaluate the effect of model recognition, this paper uses Accuracy and F1-score in the confusion matrix [25] as evaluation indicators. The value of the F1-score depends on the calculation of Precision and Recall, and the calculation rule of Macro-F1 is used. The calculation formula is as follows. Among them, TP represents the number of positive samples predicted to be positive, FP represents the number of negative samples predicted to be positive, TN represents the number of negative samples predicted to be negative, and FN represents the number of positive samples predicted to be negative.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}. \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{4}$$

$$F1 - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

3.2. Influence of Pretraining Weight Parameters on Accuracy. To reduce the calculation pressure and overfitting problem of the model [26, 27], the transfer learning pretraining weight is introduced. The pretraining weight retains a large amount of parameter information trained on the ImageNet dataset. In this section, the influence of pretraining weight parameters on Accuracy is studied.

The experimental running environment is Windows 10 and Python 3.7. The open-source deep learning framework PyTorch is used as the development environment. An Nvidia GTX1070Ti GPU is used to accelerate the training in the training process. In order to improve the generalization ability of the model in the process of image recognition, the collected Lentinus edodes sticks disease dataset is preprocessed: The RandomResizedCrop function is used to uniformly adjust the size of the picture to the size 224 × 224 required by the ResNeXt-50(32 × 4d) model; image enhancement techniques such as random rotation and horizontal flip are used to increase the diversity of Lentinus edodes sticks disease images and expand the data set. The ToTensor function is used to convert the image into the tensor format acceptable to the model and normalize it to between [0.0, 1.0]. The Normalize function is used to standardize the image. After standardization, the data are more in line with the distribution law of data centralization, which can increase the generalization ability of the model.

The Lentinus edodes stick disease dataset is divided into a training set and a test set at a ratio of 9:1. Then, the transfer learning method is adopted, and the load_state_dict function is used to load the pretraining weight resnext-50(32 × 4d).pth corresponding to the ResNeXt-50(32 × 4d) model and transfers its network parameters to the collected dataset of Lentinus edodes stick diseases. To study the influence of transfer learning pretraining weight parameters on Accuracy, the following six groups of comparative experiments were designed.

(1) The Accuracy of the ResNeXt-50(32 × 4d) network was only 72.89% without using the pretraining weight of transfer learning and using the Lentinus edodes stick disease dataset to train the ResNeXt-50(32 × 4d) network from scratch.

(2) Transfer learning was used to pretrain the weights, but none of the weight parameters of the layers were frozen. The Lentinus edodes stick disease dataset is used to retrain the weights of all layers, and the Accuracy was 91.39%.

(3) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer and layer1 were frozen. The Lentinus edodes stick disease dataset was used to retrain layer2, layer3, layer4, and the fully connected layer. The Accuracy was 90.62%.

(4) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer, layer1, and layer2 were frozen. The Lentinus edodes stick disease dataset was used to retrain layer3, layer4, and the fully connected layer. The Accuracy was 85.52%.

(5) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer, layer1, layer2, and layer3 were frozen. The Lentinus edodes stick disease dataset was used to retrain layer4 and the fully connected layer. The Accuracy was 84.37%.

(6) The transfer learning pretraining weight was used, and all parameters of the pretraining weight convolutional layer, layer1, layer2, layer3, and layer4 were frozen. The fully connected layer was retrained with the Lentinus edodes stick disease dataset, and the Accuracy was 76.55%.

The comparison between experiments (2)-(6) and experiment (1) shows that for the Lentinus edodes stick disease dataset, the pretraining weight parameter had a significant effect on the improvement of the model recognition accuracy. This is due to the use of the large image dataset ImageNet for transfer learning pretraining weight. ImageNet provides a large number of images, which enables the model to learn more features and better fit the parameters. Therefore, the model obtained better initialization network parameters during transfer learning and reduced the possibility of overfitting. This also shows that the transfer learning ability of the pretraining weight obtained with sufficient training data in the target domain is stronger than that of directly training small sample data.

In an experiment (2), the transfer learning pretraining model was used, but none of the weight parameters of the layers were frozen, and then, the weights of all layers were retrained. This transfer learning method achieved the highest Accuracy of 91.39%. This shows that on the premise of using the transfer learning pretraining weight, training the whole model with the Lentinus edodes stick disease dataset will quickly improve the learning ability of the model. Because the training process from beginning to end gradually refines the underlying features of the original input, the feature expression ability between layers is stronger, and the abstract features of the image can be better integrated.

In experiments (3)–(6), the more frozen layers there were, the lower the accuracy of the model, and the higher the overfitting ratio of the model. The reason for this situation is that the more frozen layers there are, the fewer the parameters that can be trained in the model, the weaker the calculation and feature extraction capabilities of the model on the images of Lentinus edodes stick diseases, the weaker the mutual influence ability of the shared features between layers, and the original bottom features cannot be relearned when they propagate layer by layer. This leads to the gradual decline of the feature transfer ability of the top layer. That is, the model only transfers high-level features but cannot realize the gradual abstraction, characterization, and extraction of features from the bottom to the high level, and so the recognition rate of the model will gradually decline in the end.

On the basis of experiment (2), the fully connected layer fine-tuning experiment of the model is carried out.

3.3. Fine-Tuning Strategy of Model Fully Connected Layer. To improve the accuracy of the model in the identification of Lentinus edodes stick diseases, six fine-tuning strategies of the fully connected layer were designed to modify the fully connected layer of ResNeXt-50(32 × 4d).

Based on the influence experiment of pretraining weight parameters, the hyperparameters of the feature extraction layer are modified to adapt to the training of the Lentinus edodes stick disease dataset. The design of the hyperparameters uses a grid search algorithm [28] to select the best combination of parameters. After experiments, the best hyperparameters of ResNeXt-50(32 × 4d) in this experiment are shown in Table 1.

Before entering the fully connected layer, the image feature matrix will pass through the global pooling layer and then use the Flatten function to flatten the dimension, and the multidimensional output will become one-dimensional. At this time, the number of nodes is 2048. To improve the classification performance of the model, the usual approach is to increase the depth of the model, increase the number of model parameters, or increase the samples of the training dataset. However, simply increasing these values will cause the model to become overfit and will reduce training accuracy. For the fine-tuning model, the classification performance can be improved by adding the fully connected layer and setting the number of neuron nodes in the fully connected layer. The increase in the number of neuron nodes and layers will enable the model to learn more information from the Lentinus edodes stick disease dataset. However, this will also increase the computational complexity and even lead to network degradation and loss of feature extraction information [29]. Based on this, 7 groups of comparative experiments are designed, including six fine-tuning methods of the fully connected layer and the ResNeXt-50(32 × 4d) original fully connected layer.

(1) FC0 : ResNeXt-50(32 × 4d) model original fully connected layer. The fully connected layer was redesigned to contain 1 layer. The number of nodes was the classification number 4.

(2) FC1 (2048-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st and 2nd layers was 2048 and the classification number 4.

(3) FC2 (2048-1024-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st, 2nd, and 3rd layers was 2048, 1024, and the classification number 4, respectively.

(4) FC3 (2048-512-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st, 2nd, and 3rd layers was 2048, 512, and the classification number 4, respectively.

(5) FC4 (2048-256-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers. The number of nodes in the 1st, 2nd, and 3rd layers was 2048, 256, and the classification number 4, respectively.

Table 1: The hyperparameters settings.

Hyperparameters	Value
Learning rate	0.001
Batch size	64
Epochs	150
Optimization	Adam
Dropout	0.5
Activation function	Relu

Table 2: Classification results of the ResNeXt-50(32 × 4d) model under different fine-tuning strategies.

Fine-tuning method	Accuracy (%)	Precision	Recall	F1-score
FC0	91.39	0.8611	0.8891	0.8749
FC1	92.39	0.9074	0.8938	0.9005
FC2	90.58	0.9318	0.9041	0.9177
FC3(the best)	94.27	0.9432	0.9412	0.9422
FC4	91.78	0.8879	0.9130	0.9003
FC5	91.99	0.8959	0.9180	0.9068
FC6	92.59	0.9357	0.9032	0.9192

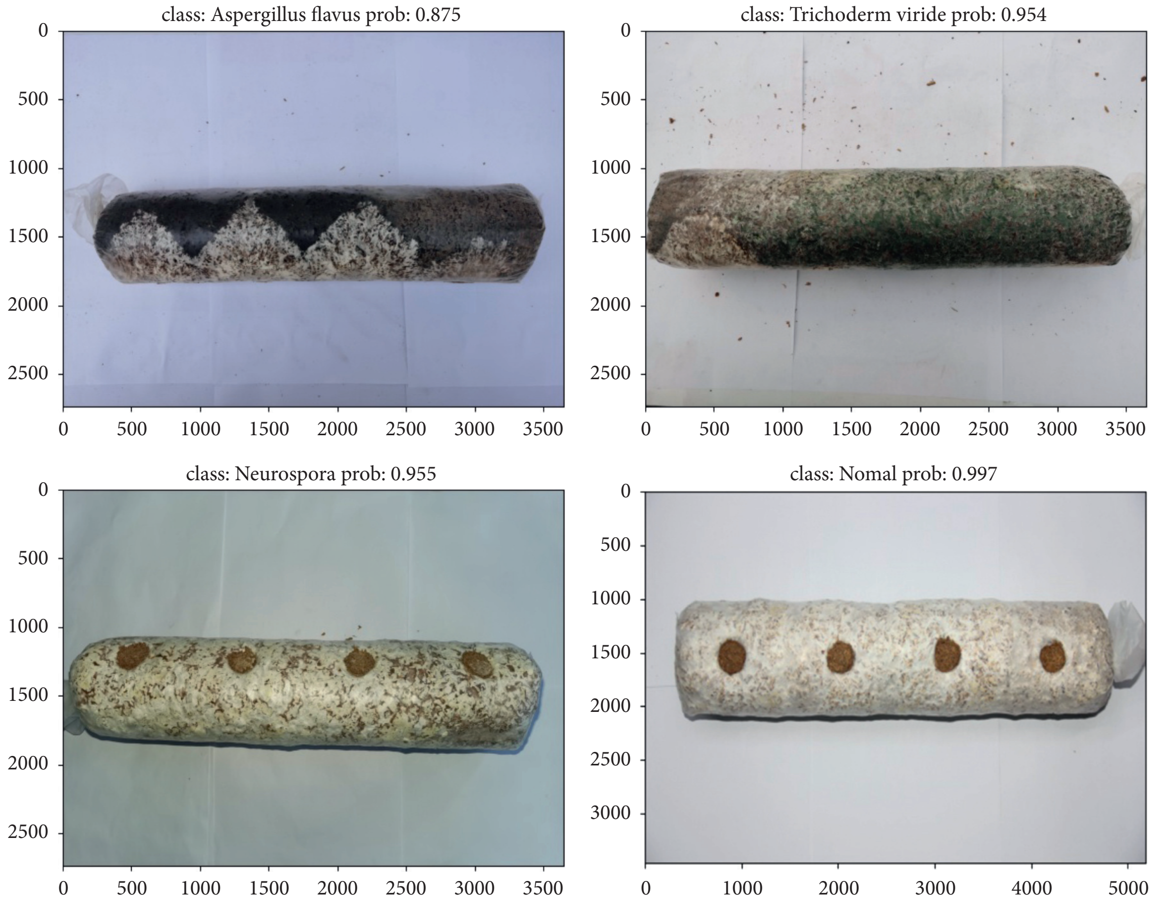

class: Aspergillus flavus prob: 0.875
class: Trichoderm viride prob: 0.954
class: Neurospora prob: 0.955
class: Nomal prob: 0.997
0
500
1000
1500
2000
2500
3000
3500
4000
5000

Figure 4: Part of the prediction results.

(6) FC5 (2048-1024-512-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 4 layers. The number of nodes in the 1st, 2nd, 3rd, and 4th layers was 2048, 1024, 512, and the classification number 4, respectively.

(7) FC6 (2048-1024-256-4): The fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 4 layers. The number of nodes in the 1st, 2nd, 3rd, and 4th layers was 2048, 1024, 256, and the classification number 4, respectively.

Table 3: Classification confusion matrix.

Actual label	Forecast label			
	Normal	Aspergillus flavus disease	Trichoderm viride disease	Neurospora disease
Normal	112	2	1	3
Aspergillus flavus disease	2	88	4	0
Trichoderm viride disease	1	3	85	0
Neurospora disease	3	1	1	61

Table 4: Classification results of different models for Lentinus edodes stick diseases.

Model	Accuracy (%)	Precision	Recall	F1-score
The model proposed in this paper	94.27	0.9432	0.9412	0.9422
VGG16	85.80	0.8742	0.8571	0.8656
GoogLeNet	87.78	0.8505	0.8863	0.8680
ResNet50	89.59	0.9378	0.8728	0.9041
MobileNet v2	84.89	0.8564	0.8466	0.8515

The number of nodes in the first layer of the fully connected layer is the one-dimensional vector obtained after global pooling and flattening the dimension with the Flatten function, while the number of nodes in the last layer is the number of output categories. The number of nodes in the middle layer is set to an exponential multiple of 2, and a large value is set to improve the calculation efficiency. At the same time, the BatchNorm1d function is used to accelerate the convergence of the neural network and improve the stability in the training process. The feature mapping is transformed into a nonlinear map by the ReLU activation function, which makes up for the deficiency of linear operation and improves the classification ability of the model. The comparison results of six fully connected layer fine-tuning strategies are shown in Table 2.

It can be seen from the table that the Accuracy of the model was improved by fine-tuning methods FC1, FC3, and FC5, which indicates that the model fine-tuning studied in this paper is effective. Among the fine-tuning methods, FC3 had the best effect. The Accuracy in the Lentinus edodes stick disease test set reached 94.27%, which was 2.88% higher than that of the original fully connected layer of the model. Part of the prediction results is shown in Figure 4.

When the number of images in the test set is 367, the confusion matrix obtained by ResNeXt-50(32 × 4d) based on FC3 method model fine-tuning is shown in Table 3.

3.4. Comparison and Analysis of Algorithms. To reflect the effectiveness of the research model in this paper, VGG16, GoogLeNet, ResNet50, and MobileNet v2 deep learning models are selected to conduct comparative experiments on the self-built Lentinus edodes stick disease dataset. The experimental results are shown in Table 4. It can be seen from the table that the Accuracy of the model studied in this paper has reached 94.27% and the F1-score value has reached 0.9422, which is the best for the recognition of Lentinus edodes stick diseases.

4. Conclusions

In this paper, the ResNeXt-50(32 × 4d) model based on deep transfer learning is designed and improved. It is used for the automatic identification of Lentinus edodes stick diseases. First, based on the pretraining weight of the ResNeXt-50(32 × 4d) model and ImageNet dataset, the influence of pretraining weight parameters on recognition accuracy is studied, and it is proven that the pretraining weight parameters have a significant effect on the improvement of the model's recognition accuracy. At the same time, without freezing the pretraining weight parameters, using the Lentinus edodes stick disease dataset to retrain the weights of all layers of ResNeXt-50(32 × 4d) can better initialize the network parameters and reduce the calculation pressure and overfitting problems of the model. Second, to improve the accuracy of the model, the fully connected layer of the ResNeXt-50(32 × 4d) model was redesigned to contain 3 layers, and the number of nodes in the 1st, 2nd, and 3rd layers was 2048, 512, and classification number 4, respectively.

In this paper, there are still deficiencies in the construction of data sets, and there is a lack of disease images of Lentinus edodes sticks in the actual culture environment. Therefore, in the next step, it is planned to add image acquisition equipment to the Lentinus edodes sticks pricking machine. When the Lentinus edodes sticks pricking machine pulls out the Lentinus edodes sticks from the shelf and rotates, the image acquisition equipment can shoot the Lentinus edodes sticks images 360°, so as to complete the collection of Lentinus edodes sticks disease images in the actual environment. In addition, the author will continue to study the compression algorithm [30] for Lentinus edodes stick disease identification and optimize the network structure to limit the number of computing and storage resources needed to run the deep neural network on mobile or embedded devices. The recognition test results will be analyzed from multiple evaluation dimensions, such as Recognition Speed, Accuracy, F1-score AUC, and ROC.

Data Availability

The data presented in this study are available on request from the corresponding author due to restrictions on privacy.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the Major Scientific and Technological Innovation Project in Shandong Province (project no. 2022CXGC010609) and Smart Qihe Biological Innovation Project.

References

[1] X. L. Zhu, L. J. Zhang, J. L. Zhang, Q. Tan, and X. D. Shang, "Identification of common molds in factory cultivation of Lentinus edodes and screening of fungistats," *Acta Agriculturae Shanghai*, vol. 2016, no. 3, pp. 83–90, 2016.

[2] F. L. Kang, J. Li, T. Liu, X. Tong, and W. B. Yu, "Application technology of image recognition for various crop diseases and insect pests:a review," *Jiangsu Agricultural Sciences*, vol. 48, no. 22, pp. 22–27, 2020.

[3] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2021.

[4] S. K. Patnaik, C. N. Babu, and M. Bhave, "Intelligent and adaptive web data extraction system using convolutional and long short-term memory deep learning networks," *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 279–297, 2021.

[5] Z. Y. Zhao, H. Yang, Z. W. Hu, and H. P. Yu, "Identification model of pests on yuluxiang pear leaves based on TACNN," *Computer Engineering and Applications*, vol. 57, no. 9, pp. 176–181, 2021.

[6] J. X. Hou, R. Li, H. X. Deng, and H. F. Li, "Leaf disease identification of fusion channel information attention network," *Computer Engineering and Applications*, vol. 56, no. 23, pp. 124–128, 2020.

[7] Z. Wang, S. W. Zhang, and B. P. Zhao, "Crop diseases leaf segmentation method based on cascade convolutional neural network," *Computer Engineering and Applications*, vol. 56, no. 15, pp. 242–250, 2020.

[8] G. Feng, L. B. Kong, M. M. Shi, M. H. He, and Y. X. He, "Crop pest recognition based on inception and residual combined network," *Journal of Guangdong University of Technology*, vol. 37, no. 3, pp. 17–22, 2020.

[9] F. K. Xiong, L. Lu, T. R. Cao, and j. Li, "Crop leaf diseases recognition: a generative adversarial network based approach," *Computer and Modernization*, vol. 303, no. 11, pp. 43–50, 2020.

[10] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[11] X. P. Fan, J. P. Zhou, Y. Xu, and X. Peng, "Corn disease recognition under complicated background based on improved convolutional neural network," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 52, no. 3, pp. 210–217, 2021.

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

[13] S. Yang, Q. Feng, J. H. Zhang, S. Wei, and G. P. Wang, "Identification method for potato disease based on deep learning and composite dictionary," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 51, no. 7, pp. 22–29, 2020.

[14] X. S. Feng, Y. Shen, and D. Q. Wang, "A survey on the development of image data augmentation," *Computer Science and Application*, vol. 11, no. 2, pp. 370–382, 2021.

[15] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks,"-vol. 2017, pp. 5987–5995, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017, IEEE, Honolulu, Hawaii, July 2017.

[16] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 770–778, Las Vegas, NV, USA, June 2016.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 1818–2826, Las Vegas, NV, USA, June 2016.

[18] F. Wang, M. Zhu, M. L. Wang et al., "6G-enabled short-term forecasting for large-scale traffic flow in massive IoT based on time-aware locality-sensitive hashing," *IEEE Internet of Things Journal*, vol. 991 page, 2020.

[19] W. Liu, X. Chen, J. Ran et al., "LaeNet: a novel lightweight multitask CNN for automatically extracting lake area and shoreline from remote sensing images," *Remote Sensing*, vol. 13, no. 1, pp. 56–62, 2020.

[20] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," *Eprint Arxiv*, vol. 27, pp. 3320–3328, 2014.

[21] W. Wang, Z. Wang, Z. Zhou et al., "Anomaly detection of industrial control systems based on transfer learning," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 821–832, 2021.

[22] J. M. Li, Y. Qu, and Y. H. Pei, "Pedestrian re-identification based on fine-tuned pre-trained convolutional neural network model," *Computer Engineering and Applications*, vol. 54, no. 20, pp. 219–222, 2018.

[23] L. Kong, L. Wang, W. Gong, C. Yan, Y. Duan, and L. Qi, "LSH-aware multitype health data prediction with privacy preservation in edge environment," *World Wide Web*, 2021.

[24] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[25] C. S. Hong, "Confusion plot for the confusion matrix," *Journal of the Korean Data and Information Science Society*, vol. 32, no. 2, pp. 427–437, 2021.

[26] Y. W. Xu, L. Y. Qi, W. C. Dou, and J. G. Yu, "Privacy-preserving and scalable service recommendation based on SimHash in A distributed cloud environment," *Complexity*, vol. 2017, Article ID 3437854, 9 pages, 2017.

[27] Y. Liu, Z. Song, X. Xu et al., "Bidirectional GRU networks-based next POI category prediction for healthcare," *International Journal of Intelligent Systems*, 2021.

[28] L. L. Sun, H. B. Fang, X. X. Zhu, L. M. Hu, and L. W. Qi, “Stock prediction using XGBoost model based on grid search optimization,” *Journal of Fujian Normal University (Philosophy and Social Sciences Edition)*, vol. 38, no. 2, pp. 97–101, 2021.

[29] D. Wei, H. Ning, F. Shi et al., “Dataflow management in the internet of things: sensing, control, and security,” *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 918–930, 2021.

[30] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and S. Yu, “Compatibility-Aware web API recommendation for mashup creation via textual description mining,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 20, pp. 1–19, 2021.

Hindawi
Complexity
Volume 2022, Article ID 4354198, 13 pages
https://doi.org/10.1155/2022/4354198

Research Article

Intensive Cold-Air Invasion Detection and Classification with Deep Learning in Complicated Meteorological Systems

Ming Yang,[1] Hao Ma ,[2] Bomin Chen,[3] and Guangtao Dong[3]

[1]*Zhejiang Meteorological Information and Network Center, Hangzhou 310052, China*
[2]*Zhejiang Climate Center, Hangzhou 310052, China*
[3]*Shanghai Climate Center, Shanghai 200030, China*

Correspondence should be addressed to Hao Ma; mahao20032003@aliyun.com

Received 21 September 2021; Revised 19 December 2021; Accepted 4 January 2022; Published 2 February 2022

Academic Editor: Xuyun Zhang

Faster R-CNN architecture is used to solve the problems of moving path uncertainty, changeable coverage, and high complexity in cold-air induced large-scale intensive temperature-reduction (ITR) detection and classification, since those problems usually lead to path identification biases as well as low accuracy and generalization ability of recognition algorithm. In this paper, an improved recognition method of national ITR (NITR) path in China based on faster R-CNN in complicated meteorological systems is proposed. Firstly, quality control of the original dataset of strong cooling processes is carried out by means of data filtering. Then, according to the NITR standard and the characteristics of NITR, the NITR dataset in China is established by the intensive temperature-reduction areas located through spatial transformation. Meanwhile, considering that the selection of regularization parameters of Softmax classification method will cause the problem of probability calculation, support vector machine (SVM) is used for path classification to enhance the confidence of classification. Finally, the improved faster R-CNN model is used to identify, classify, and locate the path of NITR events. The experimental results show that, compared to other models, the improved faster R-CNN algorithm greatly improves the performance of NITR's path recognition, especially for the mixed NITR paths and single NITR paths. Therefore, the improved faster R-CNN model has fast calculation speed, high recognition accuracy, good robustness, and generalization ability of NITR path recognition.

1. Introduction

China, located in the east of Eurasian continent and adjacent to the Northwest Pacific, is significantly influenced by the prominent Asia monsoon system originating from the thermal gradient between ocean (the Pacific and Indian Ocean) and land (the Asia continent) [1]. Chinese climate usually exhibits multiscale variability, from diurnal to decadal [2], due to the complicated interactions among various atmospheric circulation systems including the western Pacific subtropical High, South Asia High, mid-latitude high level jet, blocking High, and typhoon and the multisource modulation including ENSO [3], the Indian Ocean sea surface temperature [4], snow cover over the Tibetan-Plateau [5], and sea ice in the polar regions [6].

In winter, China is frequently impacted by the cold-air (CA) processes especially for the northeast and northwest region, which may cause huge economic loss and serious health threat. In January of 2008, most areas of southern China suffered an extreme cold spell accompanied by severe precipitation and snowfall [7, 8], which brought grave traffic and energy pressure. Unfortunately, such cold wave happened in the Spring Festival travel season and thus many people had to stay in railway stations or airports for several days and could not come back home.

Under the background of global warming, subtropical extreme cold events keep increasing rather than decreasing because of the weakened westerly jet associated with the lessened temperature gradient between polar and tropical regions [9, 10], and therefore CA events become a hot topic in current climate research [11]. Generally speaking, CA studies can be divided into three fields: case study, synoptic dynamics study, and climate dynamics study. Case analyses mainly focus on temporal-spatial characteristics and associated physical mechanisms in certain extreme CA event [12–14]. In the viewpoint of synoptic dynamics, the intensity, persistency, and spatial coverage of CA events depend on complicated and nonlinear interaction among different circulation systems [15–17]. With climatic diagnostic analysis and numerical sensitivity experiments, the climatological background of CA events and the role of critical external forcings can be understood [18–21]. Also, from the perspective of spatial impact, the CA events can be separated into national and regional processes, and the former can cover larger areas and have stronger influences on economic and social development. Due to frequent occurrence of large-scale CA events after 2000, meteorologists' interest in national CA processes is fuelled in recent years.

Although CA researches have made great progress in the past decades, especially the investigation on the role of subseasonal processes in triggering strong cold wave which has gone deep, how to identify the routine of CA invasion and associated intensive temperature-reduction (ITR) remains unclear. In fact, the track of most moving synoptic systems is hard to detect with simple statistical method. Even for some single entity systems such as typhoon, determining the exact path is also very difficult [22]. As for the march of CA and ITR, considering that each air particle has its own path, the composite pathway of CA actually reflects the statistical characteristic of all particles, which is invisible and arduous to calculate. In traditional CA and ITR monitoring operation, the routine is usually and subjectively judged by forecasters [23], which is obviously not precise. To improve monitoring and diagnosis accuracy of CA events and large-scale ITR, an objective identification method for CA and ITR path is urgently needed. In fact, the trajectory of ITR does reflect the influence of CA and has intimate association with meteorological-disaster prevention, and large-scale ITR especially national ITR (NITR) usually causes extremely serious damage, so, in this paper, we mainly focus on objective recognition for NITR path.

In the past ten years, artificial intelligence (AI) technology has made great progress in the fields of computer vision [24], language processing [25, 26], machine translation [27], medical imaging [28], robotics [29], and biological information control [30], especially for medical diagnoses [31]. For example, it performs well in terms of unmanned driving [32] and has higher recognition accuracy than the human brain for image and voice recognition. As the core method of AI, machine learning is the main method to implement artificial intelligence. Machine learning is a collection of various algorithms that allow computers to learn automatically. It helps computers analyze large sets of sample data, obtain rules, and then use these rules to classify or predict new data. Therefore, it has triggered a historic revolution in many fields [33, 34]. In common meteorological research, low-temperature forecasting involves the combination of numerical prediction products and statistical theory [35–37]. The rise of artificial intelligence facilitates applying deep learning technology to the forecasting of meteorological elements and improving the accuracy of the forecast core research problems. As an extension of machine learning algorithms, deep learning is currently and mostly used and limited to image recognition technology in meteorology. How to better to apply it to intelligent seamless grid weather forecasting is an urgent problem. Compared with the establishment of forecast equations point by point in the past, deep learning can directly establish the forecast of the entire element field, which can not only correct the forecast results in the numerical model but also consider the continuity of the spatial distribution of the elements. It has a very considerable advantage in developing objective forecasting techniques for grid points and is also more in line with forecasters' forecasting ideas.

Convolutional deep learning neural networks models, including convolutional neural networks (CNNs) [38], recurrent neural networks (RNNs) [39], deep neural networks (DNNs), and gated recurrent units (GRUs) [40], are mainly used to extract and recognize image features of the meteorological element field. Long short-term memory (LSTM) [41] networks are particularly suitable for predicting and analyzing big data time series and are continuously improving for meteorology.

To detect the exact track of NITR events and make reasonable classification, faster R-CNN target detection architecture is used to solve the problems of moving path uncertainty, changeable coverage, and high complexity. On this basis, an improved recognition method based on faster R-CNN and SVM is proposed. This algorithm adopts SVM for NITR's path classification to enhance the confidence of classification. Finally, the improved faster R-CNN model is used to identify, classify, and locate the path of NITR events. The experimental results show that, compared to the original algorithm, the improved faster R-CNN algorithm greatly improves the performance of path identification, especially for the mixed directions and incomplete development scenarios. In general, the amended faster R-CNN algorithm has fast calculation speed, high recognition accuracy, good robustness, and generalization ability of the practical application of NITR pathway detection.

The remainder of this paper is organized as follows. Section 2 describes data processing, the overall architecture, and the methods for the Faster R-CNN and SVM model of intensive NITR recognition and classification, including the faster R-CNN network, network training, recognition, and classification method. Section 3 presents the experimental environment and method used to evaluate and analyze the performance of the improved faster R-CNN and SVM model. Section 4 concludes the paper.

2. Methodology

In this section, we describe our model in detail. First, we introduce data used in this paper and the division of source regions of the NITR events. Next, we present overall structure of the faster-RCNN model. Finally, the improved faster-RCNN model for intensive NITR recognition and classification is provided.

2.1. Data and Methods. Based on the daily dataset of China's surface observational temperature (including mean temperature, maximum temperature, and minimum temperature over 1995 national conventional stations) from January 1^{st}, 1961 to December 31^{st}, 2018, provided by the National Meteorological Information Center, the surface observational data of 1995 stations were processed by simple time series investigation, neighbour interpolation, and outlier detection analysis methods, and different station datasets of intensive temperature-reduction processes were generated. As shown in Figure 1, there is a comparison of original data and revised data of national conventional station dataset from 1961 to 2018. There are invalid values and missing measured data in Figures 1(a), 1(c), and 1(e), which are corrected by simple time series investigation, neighbour interpolation, and outlier detection analysis methods. The revised results shown in Figures 1(b), 1(d), and 1(f) are used as the original dataset of this paper.

According to the results of massive studies associated with large-scale ITR over China induced by heavy cold-air processes, NITR events are mainly originated from dense cold-air invasion (CAI) from three source regions: northwest, North, and northeast of China. Although all the southward movement of CA is related to the negative phase of the Arctic Oscillation, each path is dominated by separate circulation system. The northwest pathway is usually controlled by the Siberia High, and the northeast routine is linked to activity of the northeast cold vortex and Okhotsk High. The existence of the north path can be attributed to interaction/competition between the Siberia High and northeast cold vortex. Therefore, three source regions of CAI can be preassigned, i.e., the northwest region (73°E-95°E), the north region (95°E-115°E), and the northeast region (115° E-135° E) (see Figure 2). Considering that there may exist compound pathways, actually we have seven types of NITR routines: the single northwest (NW), north (N), northeast (NE) path and the composite northwest + north (NW + N), northeast + north (NE + N), northwest + northeast (NW + NE), and northwest + north + northeast (NW + N + NE) path.

2.2. National Intensive Temperature-Reduction Recognition

2.2.1. Faster R-CNN Network. Convolutional neural network (CNN) has been widely used in many fields, such as target detection and speech recognition. Besides, the region based convolutional neural network (R-CNN), which was proposed by Ross Girshick in 2014 [42], also performs well and gets rapid development. R-CNN is a classic algorithm and basic method for image recognition using region recommendation, and, on the basis of R-CNN, two new technologies, the Fast R-CNN and faster R-CNN algorithms, are further proposed and improved.

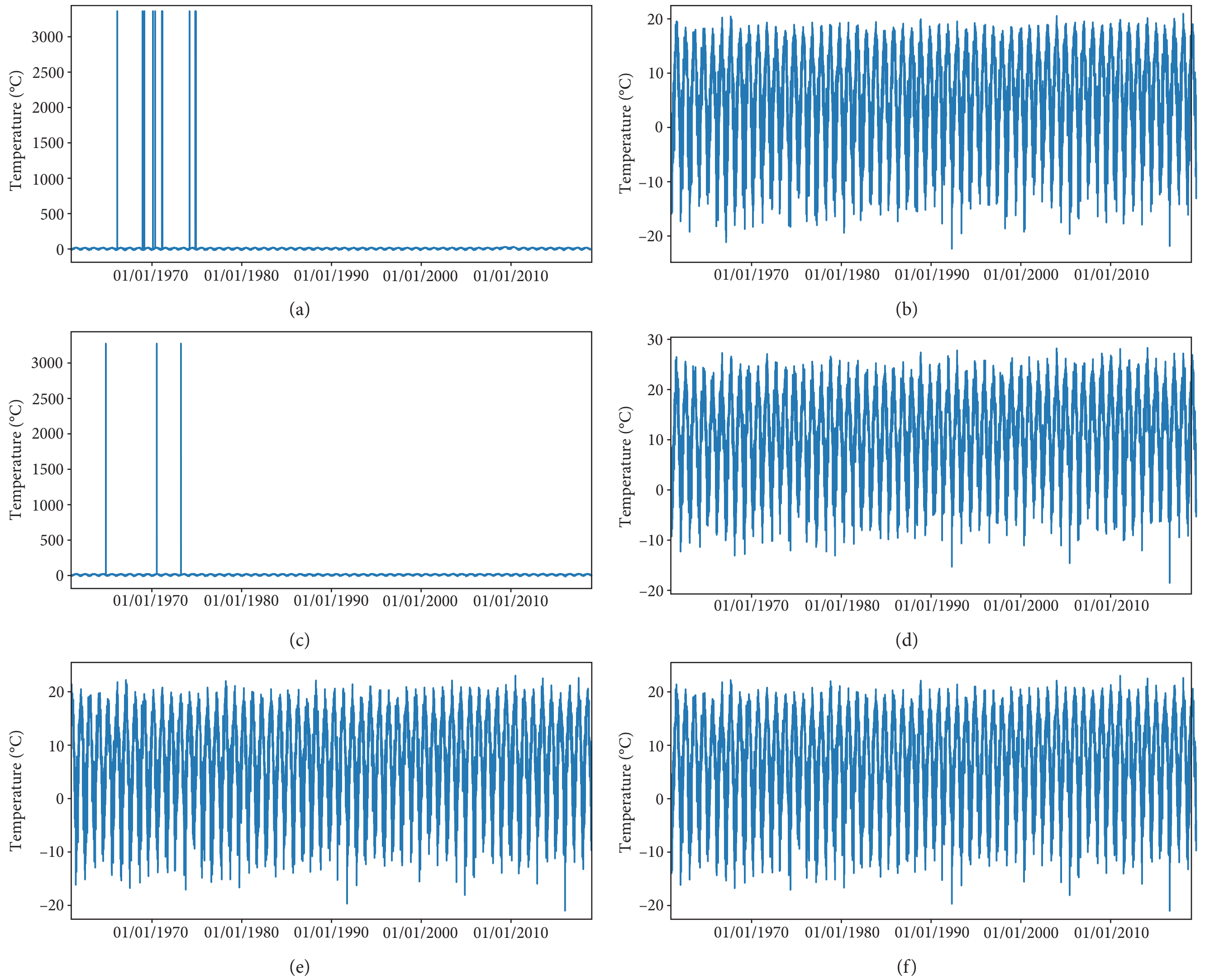
In general, R-CNN algorithm can be divided into four steps: (1) candidate region generation, (2) feature extraction, (3) category judgment, and (4) location refinement. Firstly, a large number of candidate regions are generated by visual method, and then the high-dimensional feature vectors of these regions are formed by convolution operation with CNN method. Subsequently, these feature vectors are sent to some classifiers, such as simple logical regression and Softmax regression. After calculating the overlap degree IOU of the object score and bounding box of the candidate regions, the candidate box is refined to realize object recognition and location.

Compared with the traditional target detection algorithm which uses sliding window to judge all possible regions in turn, the R-CNN algorithm extracts a series of candidate regions which are more likely to be objects in advance and then extracts features only on these candidate regions (using CNN) for judgment, which effectively reduces the calculation of subsequent feature vectors and can better deal with the scale problem. The implementation of CNN adopts GPU parallel computing, which improves the computing speed and efficiency. In addition, the regression step of the peripheral box improves the accuracy of target location.

2.2.2. National Intensive Temperature-Reduction Recognition. Although R-CNN has become a typical algorithm in the field of image recognition, the bottleneck of the algorithm is that it needs to take long time to generate region suggestions in the first step. Aiming at this defect, faster-RCNN came into being. As for the new algorithm, an RPN is proposed, which is a network based on full convolution. It can simultaneously predict the target area box and target score of each position of the input image, aiming at efficiently generating high-quality area suggestion box. Its appearance replaces the previous methods such as selective search and edge boxes. It shares the convolution characteristics of the whole image with the detection network, so that the detection of region recommendation is almost time-consuming. Therefore, the faster R-CNN is used for NITR recognition in this paper, and, moreover, Support Vector Machine (SVM) model is adopted to classify the type of NITR.

(1) Network Training. Faster R-CNN algorithm includes RPN and faster CNN detection network. In this paper, ZFNet [43] is pretrained to initialize the detection network of RPN and faster R-CNN. The typical structure of ZFNet is shown in Figure 3. The pretraining process of this method in this paper is as follows.

(1) *Pretraining CNN.* The typical structure of ZFNet consists of five Convolutional Layers and two Fully Connected (FC) Layers. A pooling layer is added behind convolution layers, and the filter size and convolution step size of each layer are slightly different. The last Convolution Layer 5 (Layer 5) of ZFNet outputs 256 channel feature maps, and the Full Connection Layer 6 (Layer 6) concatenates all the features in 256 channels to generate a single channel high-dimensional feature vector with 4096 dimensions. Different types of images have great differences in deep features. The classifier is used for the feature vectors output by Layer 5, Layer 6, and Layer 7, which can output image recognition results.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 1: Comparison of original data and revised data of national conventional station dataset from 1961 to 2018. (a) Original minimum temperature in the last 24 hours. (b) Minimum temperature in the last 24 hours by quality control. (c) Original maximum temperature in the last 24 hours. (d) Maximum temperature in the last 24 hours by quality control. (e) Original average temperature in the last 24 hours. (f) Average temperature in the last 24 hours by quality control.

(2) *Training RPN*. The pretrained ZFNet is used to initialize the RPN, and a small Convolution Layer 6 (Layer 6) with specific function is added after the original Convolution Layer 5 (Layer 5). On this layer, the convolution operation of the feature map output by Convolution Layer 5 (Layer 5) is carried out in a sliding window way and the shapes of the sliding window were squares or rectangles and overlapping ratio is 0.5. For each position of the image, nine fixed dimensions and aspect ratio (1 : 1, 1 : 2, 1 : 2, and 1 : 2) are considered as 2 : 1. The output of Layer 6 is used as the input of two independent full connection layers, box regression layer and box classification layer, and finally multiplied by 9. The probability is that two windows belong to the target or background, and four pan zoom parameters are multiplied by 9.

(3) *Training Faster R-CNN Detection Network*. In the same way, the ZFNet is used to initialize the detection network, and the region recommendation obtained from RPN is used as the input of the detection network. The feature is extracted by five Convolution Layers, and the feature map is compressed through the corresponding pooling layer to get 256 channel feature maps. Then, the feature map is connected in series through Fully Connected Layer 6 and Fully Connected Layer 7 and finally classified by SVM. In this manner, whether there is the type of intensive temperature-reduction in the suggestion box and the associated location can be obtained. The samples are used for training and fine-tuning many times, and the layer connection weight matrix is updated in the process of error backpropagation. Finally, a detection network suitable for NITR recognition is acquired.

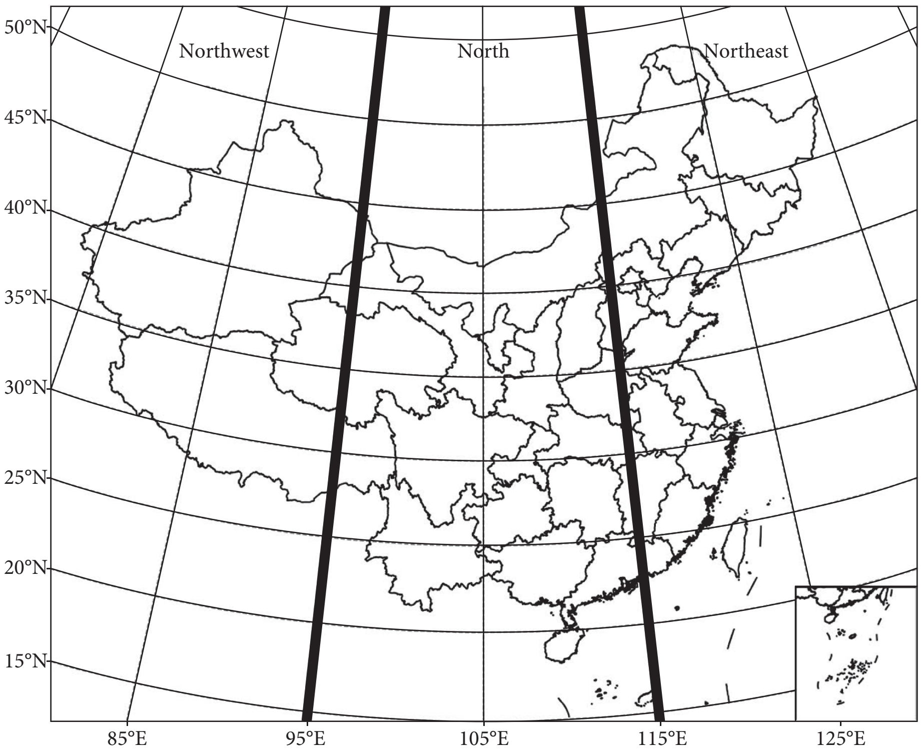


Figure 2: Division of source regions of the CAI events.

(4) *RPN and Faster R-CNN Sharing Convolution Layer*. After the above training processes, the two networks are still independent of each other, so it is necessary to share the Convolution Layer so that the features can be used for both region suggestion box generation and target detection. The specific methods are the following: (a) using ZFNet to generate RPN independently; (b) training the faster R-CNN detection network with the region suggestions and network parameters generated by RPN in (a); (c) applying the faster R-CNN detection network parameters to initialize RPN. At this time, it is necessary to pay attention to set the learning rate of convolution layers shared by RPN and faster R-CNN network to 0, that is, not to update these convolution layers, but only updating those network layers unique to RPN and retraining them. Then, RPN and faster R-CNN detection network share all the common convolution layers, which improves the region recommendation procedure and effectively reduces the run time of the algorithm.

(2) Classification Method. SVM is a machine learning method based on statistical learning theory. By seeking the minimum structural risk, the empirical risk and confidence range are minimized, so that the system can get better statistical rules even when the number of samples is small. Compared with traditional pattern recognition methods, SVM has strong generalization ability and can guarantee the global optimization. The core idea of SVM algorithm is to find an optimal classification to meet the classification requirements.

In reality, most of the classification is nonlinear, and the strong cooling path recognition in this paper is also nonlinear. At this time, the nonlinear problem can be transformed into a linear problem in a high-dimensional space through space transformation, and the optimal classification surface or the optimal generalized classification surface can be obtained in the transformed high-dimensional space. The kernel function is used to map the linear nonseparable low dimensional space to the linear separable high-dimensional space. The common kernel functions are the Polynomial function, the Radial Basis function (RBF), and the Sigmoid function. In this paper, RBF is used in the NITR pathway recognition algorithm, which can be expressed as

$$K(x, y) = \exp\left(-\frac{x - y^2}{2\sigma^2}\right), \tag{1}$$

where σ is the kernel parameter. x and y are the vector.

NITR pathway recognition is a multiclassification issue. Given a set of training samples, it is necessary to divide those raw data into seven categories, namely, NW (marked as 1), N (marked as 2), NE (marked as 3), NW + N (marked as 4), NE + N (marked as 5), NW + NE (marked as 6), and NW + N + NE (marked as 7), so we need totally 7 SVM classifiers. In practice, SVM can be trained and used for classification through the following steps: (1) the first is feature extraction of classified images; (2) a simple linear method is used to normalize the feature vector to prevent large data fluctuation from dominating data perturbation and small data fluctuation from being ignored; (3) the RBF

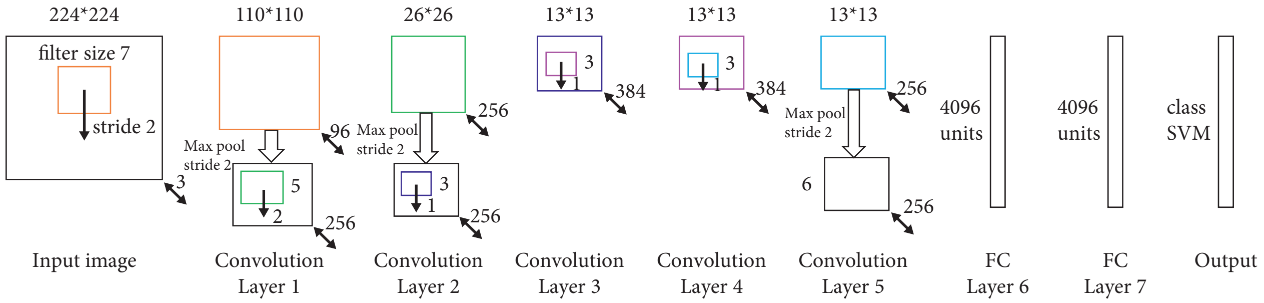


Figure 3: The typical structure of ZFnet.

kernel is used to select the kernel function; (4) the cross validation method was used to select parameter C; (5) the optimal parameters are used to train the training set to obtain the SVM classification model; (6) the trained SVM model is used to classify and predict the output eigenvectors, and the output eigenvector matrix is dot-multiplied with the SVM weight matrix to get the score of the recommendation box in the region, that is, the NITR path type in the recommendation box in the region.

(3) Recognition and Classification Method. From the above network training process and SVM classification method, we can see that the two networks using faster R-CNN for recognition share convolution layer. Therefore, the whole recognition process only needs to complete a series of convolution operations, which is able to effectively realize recognition and solve the long time-consuming problem of regional recommendation. In addition, SVM is used as the final classifier to minimize the empirical risk and confidence range, which can get better statistical rules for the number of samples is small. The structure of faster R-CNN and SVM model is demonstrated in Figure 4. First of all, the structure of proposed model with RPN has been implemented on the available dataset to extract the features on the convolution layer. And, second, the feature map from convolution layer enters RPN and generates a large number of regional suggestion boxes on the feature map. It should be noted that, for each position of the feature map, nine candidate windows with fixed scale and aspect ratio are considered. Thirdly, nonmaximum suppression was applied to the RPN-generated regional suggestion boxes, and 200 boxes with higher scores were retained. Fourthly, the faster R-CNN recognition network extracts feature vectors from the image in the region suggestion boxes, inputting them into the full connection layer, and then inputs them into the SVM classifier to calculate the score of each region suggestion box. Finally, the faster R-CNN recognition network refines the region suggestion box by regression.

3. Experiments and Analysis

To examine the performance and effectiveness of NITR identification based on the faster R-CNN model, the deep learning experiments are constructed by using Python 3.7. The CPU of the experiment is an Intel Core i5 @ 2.30 GHz with 8 GB of memory, and the operation system is 64-bit Windows 10. The proposed hybrid faster R-CNN model has default parameter settings; the number in the 1st convolution layer is 64; the filter size in the 1st convolution layer is 3; the pooling size is 2; the dropout rate is 0.46.

The experimental process of the faster R-CNN model includes data acquisition, data preprocessing, feature importance assessment, model training, model testing, and model evaluation. Data preprocessing includes data normalization, training set partition, test set construction, and time series construction. After data preprocessing, the training data are used to generate the model that is to adjust the network weight through the optimization function to minimize the loss function of the model until the number of iterations reaches the set value. Then, the training model is applied to the test set data, and the performance of the model is measured by the average precision (AP), the mean average precision (mAP), and other evaluation indicators.

3.1. Dataset. How to describe climatic characteristics of the NITR events including cooling amplitude and related coverage is an important issue in NITR path identification. In a national standard published by the China Meteorological Administration, the change of daily minimum temperature is chosen to reflect the intensity of heavy CA or cold wave processes. Thus, the time series of daily minimum temperature over 1995 national conventional stations is selected here as the original dataset.

Table 1 shows the classification of stational ITR by reference to Chinese national standard (GBT 20484-2017). Based on this standard, the linear interpolation method has been chosen to make grid analysis and pictures for national station data; 497 NITR processes from 1961 to 2018 are generated, with a total of 3434 target images. The image size is 800 pixels × 800 pixels, and the storage format is JPG.

After strict selection, there are totally 2800 exactly suitable images, marked with LabelImg tool, and the location of NITR is recorded. According to the preassigned types of NITR paths, the dataset is divided into seven different kinds of NITR processes: NW, N, NE, NW + N, NE + N, NW + NE, and NW + N + NE, as shown in Figure 5. In accordance with the ratio of 8 : 1:1, the dataset is divided into training set (80%), verification set (10%), and test set (10%). These three datasets are independent and mutually exclusive, which are used for training, parameter optimization, and performance evaluation of target detection model, respectively.

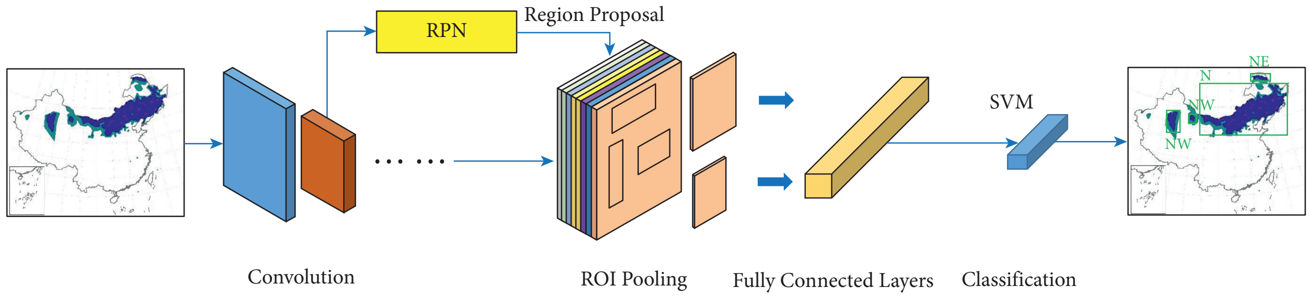


Figure 4: The architecture of the proposed Faster R-CNN and SVM model.

3.2. Model Training. In order to improve the training speed and convergence performance of target model. Firstly, the ImageNet dataset is preprocessed to convert the training set and verification set data into TFRecord format. Then, the training is started based on the TFRecord data file. During the training procedure, critical parameters are settled as follows: the batch size is 64; the image size is scaled to 224 pixels × 224 pixels; the training cycle is 85; the number of iterations in each cycle is 10000 and the total number of iterations is 850000; the momentum factor is 0.9; the weight attenuation coefficient is 0.0001; the initial learning rate is 0.01. The learning rate is attenuated by using the segmentation constant, and the final learning rate is faded to 0.00001.

Finally, the random gradient descent method is used to deploy the target detection model. In the process of network training, there are altogether 2240 images in the training set; the momentum factor is 0.9; the weight attenuation system is 0.0005; the initial learning rate is 0.0001; the attenuation rate is 0.9 and the total number of iterations is 6000.

3.3. Evaluation. To objectively evaluate the generalization ability of the NITR path type recognition model, the AP and mAP criteria are used as measurements of derivation between observed and predicted values. In the application scenario of this paper, NITR events are set as positive samples and the corresponding backgrounds are negative samples. The ratio between the number of strong cooling paths correctly detected by the model and the entire number of predicted strong cooling tracks is defined as precision (P), which is used to measure the recognition ability of positive samples. Recall (R) is defined as the ratio of the amount of correctly identified data of a certain type of strong cooling pathway in the test set data to the total number of such strong cooling pathways, which is used to measure the coverage of positive samples.

The average accuracy is related to the accuracy and recall. It is the integral of the accuracy recall curve and the coordinate axis, which is used to measure the recognition effect of the model. The larger the value is, the better the recognition effect of strong cooling path is. The average value is able to reflect mean accuracy of multicategory strong cooling path identification. Similarly, the larger the value is, the higher the accuracy of model realization is.

(1) AP can more intuitively show the classifier performance, which is defined in the following equation:

$$\mathrm{AP} = \int_0^1 p(r)dr, \tag{2}$$

where $p(r)$ is a function of precision as a function of r. The area between the function curve and the coordinate axis is the average accuracy.

(2) The calculation formulas of accuracy and recall are as follows:

$$P = \frac{T_P}{T_P + F_P}, \quad R = \frac{T_P}{T_P + F_N}, \tag{3}$$

where P is the accuracy rate; R is the recall rate. T_P is the number of truly positive samples, and such samples as positive members in observation are also determined to be positive samples by recognition model, so the prediction is correct. F_P is the number of falsely positive samples, and such samples as actually negative members in observation are judging as positive samples by model, so the prediction is wrong. F_N is the number of falsely negative cases, and such cases are positive ones in fact but judging as negative samples by model, and thus these samples are mistakenly omitted.

(3) mAP is defined in the following equation:

$$\mathrm{mAP} = \sum_{i=1}^{n} AP_i, \tag{4}$$

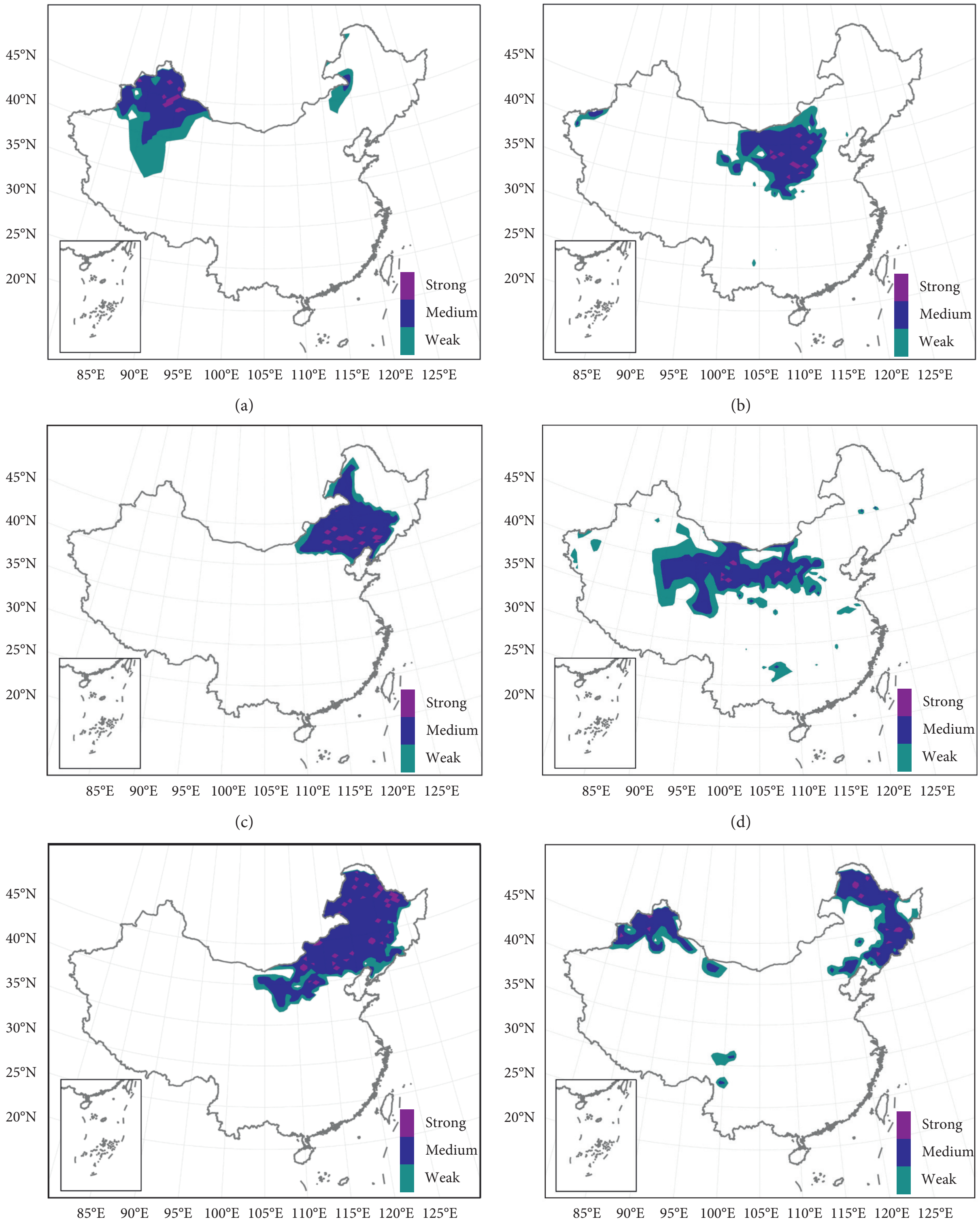
where mAP represents the average precision and n is the number of targets to be detected. There are 7 detection targets in this paper.

3.4. Performance and Analysis. Faster R-CNN and R-FCN models are trained with the same training set samples, and the performance of the trained models are compared with our new model proposed in this paper.

Table 2 shows performance comparison of the three models. In terms of accuracy, the average accuracy of our

Table 1: Classification of stational ITR in China.

Grade	Division index
Weak	The daily minimum temperature drops by less than 6°C within 48 hours
Medium strong	The daily minimum temperature drops by more than or equal to 6°C but less than 8°C within 48 hours; or the daily minimum temperature drops by more than or equal to 8°C within 48 hours but fails to reduce the daily minimum temperature to 8°C or below
Strong	The daily minimum temperature drops by more than or equal to 8°C within 48 hours, and the daily minimum temperature also reduces to 8°C or below



(a)

(b)

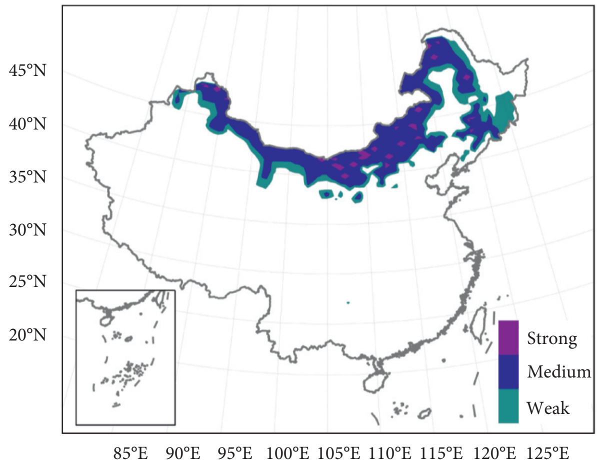
(c)

(d)

(e)

(f)

Figure 5: Continued.



(g)

Figure 5: The kinds of different NITR pathways. (a) is the NW path, (b) is the N path, (c) is the NE path, (d) is the NW + N path, (e) is the NE + N path, (f) is the NW + NE path, and (g) is the NW + N + NE path.

Table 2: Comparison of performance of different models.

Model	mAP (%)	Speed (*f*/*s*)	Storage space (MB)
Our model	86.5	11.1	85
Faster R-CNN	84.8	3.3	112
R-FCN	85.1	6.5	91

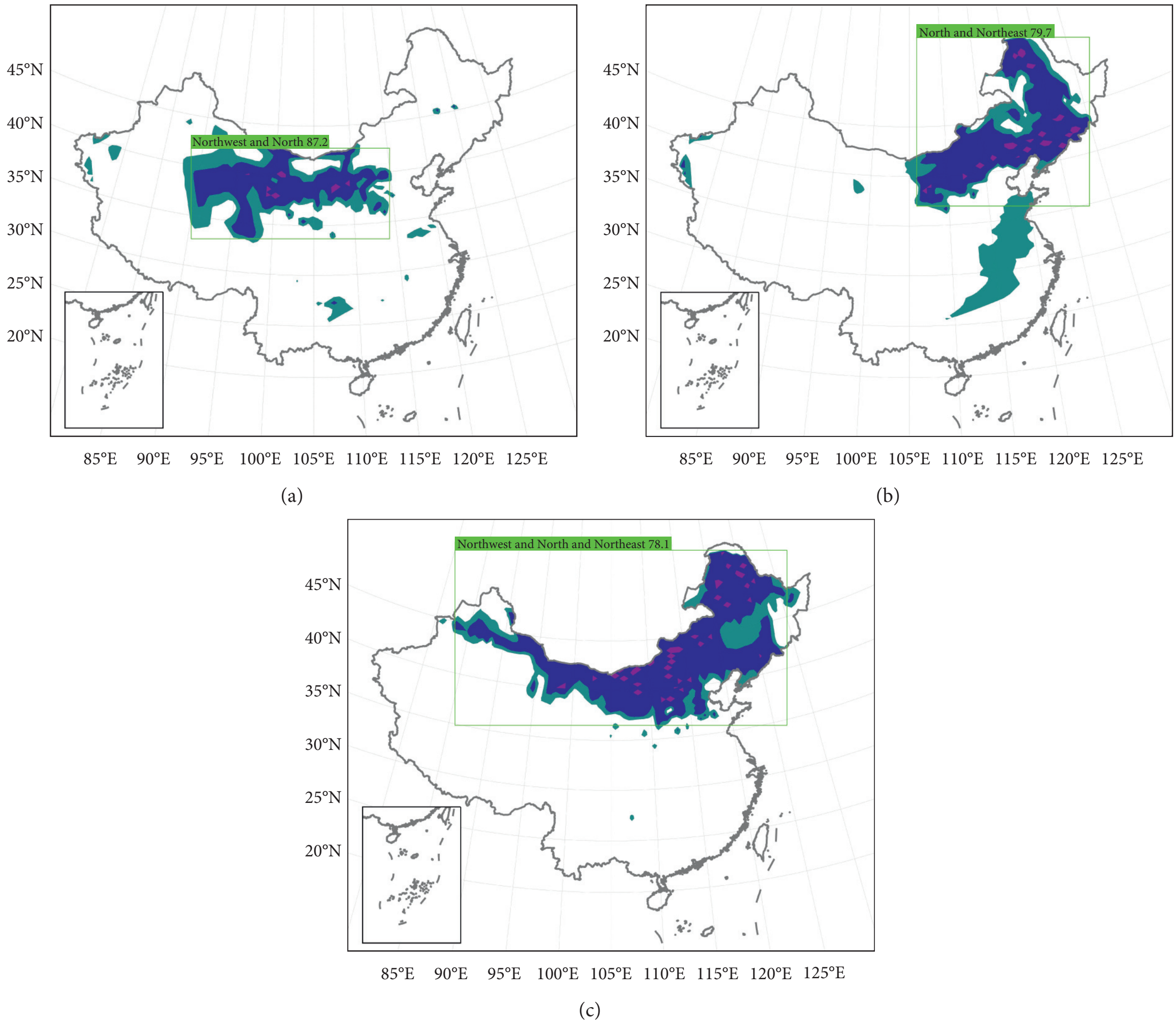
Table 3: Evaluation results of different models in mixed and single NITR paths recognition.

Model	mAP		
	Mixed paths	Single paths	Average
Our model	86.3	87.6	86.95
Faster R-CNN	82.2	84.5	83.35
R-FCN	83.6	85.1	84.35

model for NITR paths is 1.7% and 1.4% higher than that of faster R-CNN and R-FCN, respectively, and has strong ability of feature extraction and accurate location regression. In terms of real-time efficiency, the number of image frames processed per second is used as the speed measurement index. Due to the lightweight design of feature extraction network and detection head in our model, the network complexity is reduced, and the reasoning speed is 3.4 times and 1.7 times higher than that of faster R-CNN and R-FCN, respectively. In terms of network scale, storage space of our model is 85 MB, less than that of faster R-CNN (112 MB) and R-FCN (91 MB). The experimental results show that our new model has significant advantages over faster R-CNN and R-FCN in average accuracy, detection speed, and network scale.

In many cases of the NITR events, there appear mixed NITR paths and some single NITR paths, which bring difficulties to the identification of severe cooling paths. In order to distinguish multiple intense cooling routes in different directions and the model recognition accuracy in the case of incomplete development, 160 samples are randomly selected for each fine condition and input into faster R-CNN, R-FCN, and our model, respectively, for recognition test. The results are shown in Table 3, the average precision of our model for recognizing mixed NITR paths and single NITR paths in different directions is 86.3% and 87.6%, respectively, and the average value in two cases is 86.95%. All the three indices are higher than those of the faster R-CNN and R-FCN models. This is because our model uses convolution kernels of different sizes for operation, which has strong multiscale feature extraction ability. FPN unit fuses different scale feature information to strengthen the expression ability of target characteristics. Under such circumstance, various strong cooling paths in different directions and incomplete development processes can be effectively identified even if the semantic information is lost on the feature map.

Figures 6 and 7 show the recognition effect of our models on single and mixed NITR paths under medium strong cooling conditions. Figure 6 shows an example of single NITR path and Figure 7 shows the performance of all models on mixed NITR paths. It can be clearly seen that our model is able to accurately detect the types of NITR paths in different environments.

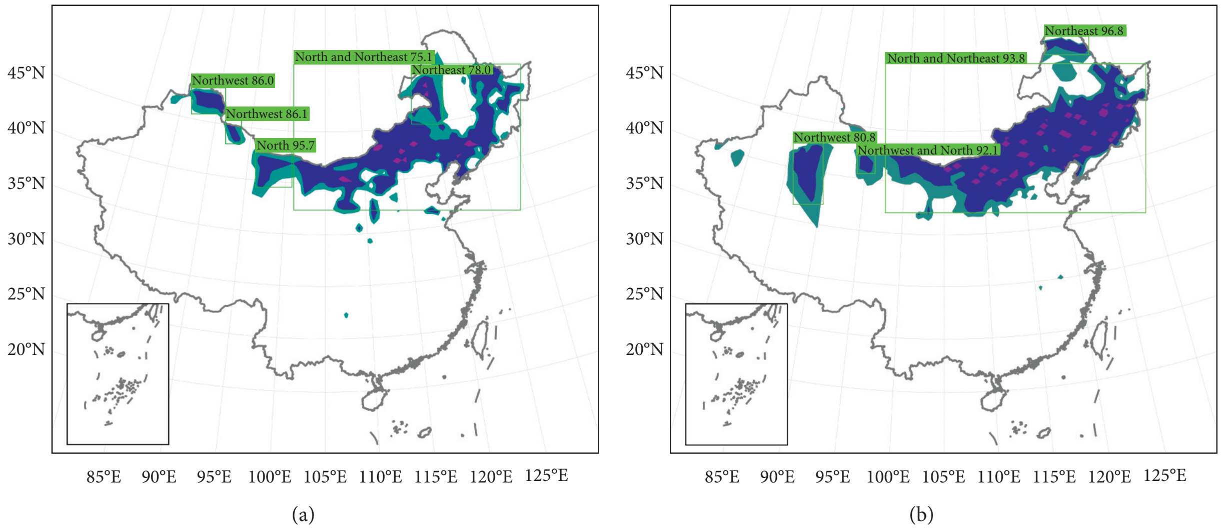

Northwest and North 87.2
North and Northeast 79.7
Northwest and North and Northeast 78.1
45°N
40°N
35°N
30°N
25°N
20°N
85°E 90°E 95°E 100°E 105°E 110°E 115°E 120°E 125°E

(a)

(b)

(c)

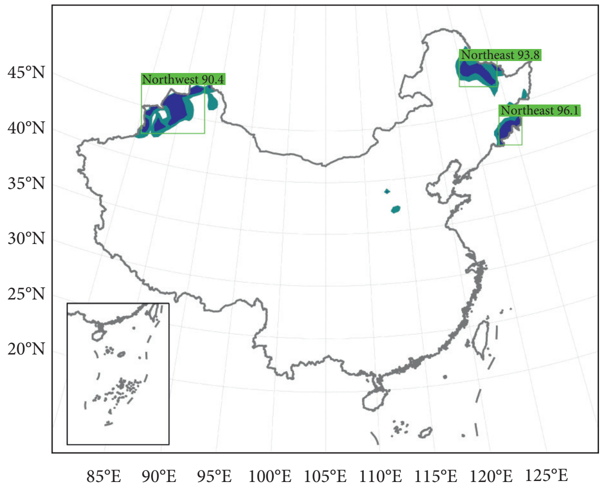
FIGURE 6: The effect of different single NITR paths. (a) is the NW + N path, (b) is the NE + N path, and (c) is the NW + N + NE path.



(a)

(b)

FIGURE 7: Continued.



(c)

Figure 7: The effect of three models on mixed NITR paths.

4. Conclusion and Future Work

With the development of deep learning technology, an improved recognition and classification method of national ITR path in China based on the faster R-CNN and SVM Model in complicated meteorological systems has fast calculation speed and high recognition accuracy. The method proposed in the paper improves the recognition performance of NITR paths. First, quality control of the original dataset of strong cooling processes is carried out by means of data filtering. Then, based on the Chinese national standard (GBT 20484-2017), the linear interpolation method has been chosen to make grid analysis and pictures for national station data; 497 NITR processes from 1961 to 2018 are generated. Meanwhile, the regularization parameters of Softmax classification method will cause approximate results of probability calculation, so SVM is used for path classification, which can obtain better results when the number of samples is small, ensure the global optimization, and improve the reliability of classification.

The experimental results show that, compared with other models, the storage space of the faster R-CNN and SVM Model is 85 MB and the recognition speed is 11.1f/s, which effectively reduces the network scale and significantly improves the recognition speed. In addition, the mAP of new model is 86.5%, 1.7%, and 1.4% higher than that of faster R-CNN and R-FCN, respectively. At the same time, it has good generalization performance for mixed paths and single NITR paths. Therefore, the improved faster R-CNN model is new method in the meteorological application of NITR path recognition.

In the future, with the development of deep learning technology and cloud computing, we will study the methods of migrating the model computing tasks in this paper to edge devices, including mobile edge computing [44], privacy aware deployment of machine learning applications [45], and dynamic resource allocation [46]. Then, we try to use the idle computing power of edge devices to share the computing pressure of cloud servers and improve computing efficiency.

Data Availability

The data and models used during the study are available from the corresponding author by request.

Conflicts of Interest

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

Authors' Contributions

Ming Yang and Hao Ma conceptualized the study. Hao Ma performed data analysis. Ming Yang and Hao Ma wrote the original draft. All the authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was jointly supported by the National Key Research and Development Program of China (Grant no. 2018YFC1505601), Zhejiang Province Basic Public Welfare Program (Grant no. LGF19D050001), Key Research and Development Program of Zhejiang Province (Grant no. 2021C02036), and Key Program of Zhejiang Meteorological Service (Grant nos. 2020ZD14, 2021YYZX09, and 2018ZD03).

References

[1] Y. Ding and J. C. L. Chan, "The East Asian summer monsoon: an overview," *Meteorology and Atmospheric Physics*, vol. 89, no. 1-4, pp. 117–142, 2005.

[2] H. Weng, K.-M. Lau, and Y. Xue, "Multi-scale summer rainfall variability over China and its long-term link to global sea surface temperature variability," *Journal of the Meteorological Society of Japan. Ser. II*, vol. 77, no. 4, pp. 845–857, 1999.

[3] B. Wang, R. Wu, and X. Fu, "Pacific-east asian teleconnection: how does ENSO affect east asian climate?" *Journal of Climate*, vol. 13, no. 9, pp. 1517–1536, 2000.

[4] S.-P. Xie, Y. Kosaka, Y. Du, K. Hu, J. S. Chowdary, and G. Huang, "Indo-western Pacific ocean capacitor and coherent climate anomalies in post-ENSO summer: a review," *Advances in Atmospheric Sciences*, vol. 33, no. 4, pp. 411–432, 2016.

[5] A. Duan, G. Wu, Y. Liu, Y. Ma, and P. Zhao, "Weather and climate effects of the Tibetan Plateau," *Advances in Atmospheric Sciences*, vol. 29, no. 5, pp. 978–992, 2012.

[6] P. Zhao, X. Zhang, X. Zhou, M. Ikeda, and Y. Yin, "The sea ice extent anomaly in the North Pacific and its impact on the East Asian summer monsoon rainfall," *Journal of Climate*, vol. 17, no. 17, pp. 3434–3447, 2004.

[7] L. Wang, G. Gao, Q. Zhang et al., "Analysis of the severe cold surge, ice-snow and frozen disasters in South China during January 2008: I. Climatic features and its impact," *Meteorological Monthly*, vol. 34, no. 4, pp. 95–100, 2008.

[8] Q. Lu, W. Zhang, P. Zhang et al., "Monitoring the 2008 cold surge and frozen disasters snowstorm in South China based on regional ATOVS data assimilation," *Science China Earth Sciences*, vol. 53, no. 8, pp. 1216–1228, 2010.

[9] X. Zhang, Y. Fu, and Z. Han, "Extreme cold events from East Asia to North America in winter 2020/21: Comparisons, causes, and future implications," *Advances in Atmospheric Sciences*, vol. 41, no. 20210229-1, p. 1, 2021.

[10] F. Zheng, Y. Yuan, and Y. Ding, "The 2020/21 extremely cold winter in China influenced by synergistic effect of La Niña and warm Arctic," *Advances in Atmospheric Sciences*, vol. 38, no. AAS-2021-0033, p. 1, 2021.

[11] J.-B. Peng, C. Bueh, and Z.-W. Xie, "Extensive cold-precipitation-freezing events in southern China and their circulation characteristics," *Advances in Atmospheric Sciences*, vol. 38, no. 1, pp. 81–97, 2021.

[12] C.-C. Hong and T. Li, "The extreme cold anomaly over southeast Asia in february 2008: roles of ISO and ENSO," *Journal of Climate*, vol. 22, no. 13, pp. 3786–3801, 2009.

[13] C. Qian, J. Wang, S. Dong et al., "Human influence on the record-breaking cold event in January of 2016 in Eastern China," *Bulletin of the American Meteorological Society*, vol. 99, no. 1, pp. S118–S122, 2017.

[14] G. Dai, C. Li, Z. Han, D. Luo, and Y. Yao, "The nature and predictability of the East Asian extreme cold events of 2020/21," *Advances in Atmospheric Sciences*, vol. 38, no. 5, pp. 1–10, 2020.

[15] G. Müller, M. Gan, E. Piva, and V. P. Silveira, "Energetics of wave propagation leading to cold event in tropical latitudes of South America," *Climate Dynamics*, vol. 45, no. 1, pp. 1–20, 2015.

[16] D. Luo, Y. Yao, and S. B. Feldstein, "Regime transition of the north atlantic oscillation and the extreme cold event over europe in january-february 2012," *Monthly Weather Review*, vol. 142, no. 12, pp. 4735–4757, 2014.

[17] J. Yamaguchi, Y. Kanno, G. Chen, and T. Iwasaki, "Cold air mass analysis of the record-breaking cold surge event over East Asia in January 2016," *Journal of the Meteorological Society of Japan. Ser. II*, vol. 97, no. 1, pp. 275–293, 2019.

[18] D. C. Barber, A. Dyke, C. Hillaire-Marcel et al., "Forcing of the cold event of 8,200 years ago by catastrophic drainage of Laurentide lakes," *Nature*, vol. 400, no. 6742, pp. 344–348, 1999.

[19] J. Inoue, M. E. Hori, and K. Takaya, "The role of Barents Sea ice in the wintertime cyclone track and emergence of a warm-Arctic cold-Siberian anomaly," *Journal of Climate*, vol. 25, no. 7, pp. 2561–2568, 2012.

[20] T. C. Peterson, R. R. Heim, R. Hirsch et al., "Monitoring and understanding changes in heat waves, cold waves, floods, and droughts in the United States: state of knowledge," *Bulletin of the American Meteorological Society*, vol. 94, no. 6, pp. 821–834, 2013.

[21] L. Li, W. Ni, Y. Li, D. Guo, and H. Gao, "Impacts of sea surface temperature and atmospheric teleconnection patterns in the northern mid-latitudes on winter extremely cold events in North China," *Advances in Meteorology*, vol. 2021, pp. 1–15, 2021.

[22] H. Yang, L. Wu, and T. Xie, "Comparisons of four methods for tropical cyclone center detection in a high-resolution simulation," *Journal of the Meteorological Society of Japan. Ser. II*, vol. 98, no. 2, pp. 379–393, 2020.

[23] S. Li, Q. Sun, Y. Yao, and L. I. A. N. Yi, "Definition of extreme low-temperature events over northeastern China in summer and the related cold air path," *Scientia Geographica Sinica*, vol. 34, no. 2, pp. 249–256, 2014.

[24] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: a brief review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, 2018.

[25] C. Cardie, "Embedded machine learning systems for natural language processing: a general framework," *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, vol. 1040, pp. 315–328, 1996.

[26] K. Gyeongmin, L. Chanhee, J. Jaechoon, and L. Heuiseok, "Automatic extraction of named entities of cyber threats using a deep Bi-LSTM-CRF network," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 2, 2020.

[27] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: an overview," in *Proceedings of the 2017 International Conference on Computer Communications and Electronics (Comptelix)*, July 2017.

[28] M. Rodríguez and J. Apolinar, "Computer vision of the foot sole based on laser metrology and algorithms of artificial intelligence," *Optical Engineering*, vol. 48, no. 12, p. 123604, 2009.

[29] N. Sünderhauf, O. Brock, W. Scheirer et al., "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[30] L. Peng, M. Peng, B. Liao, G. Huang, W. Li, and D. Xie, "The advances and challenges of deep learning application in biological big data processing," *Current Bioinformatics*, vol. 13, no. 4, pp. 352–359, 2018.

[31] M. Bakator and D. Radosav, "Deep learning and medical diagnosis: a review of literature," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 47, 2018.

[32] J. Woo, J. Park, C. Yu, and N. Kim, "Dynamic model identification of unmanned surface vehicles using deep learning network," *Applied Ocean Research*, vol. 78, pp. 123–133, 2018.

[33] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] C.-J. Huang and P.-H. Kuo, "A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities," *Sensors*, vol. 18, no. 7, p. 2220, 2018.

[35] L. Zhang and X. F. Zhi, "Multimodel consensus forecasting of low temperature and icy weather over central and Southern

China in early 2008," *Journal of Tropical Meteorology*, vol. 21, no. 1, pp. 67–75, 2015.

[36] T. N. Krishnamurti, V. Kumar, A. Simon, A. Bhardwaj, T. Ghosh, and R. Ross, "A review of multimodel superensemble forecasting for weather, seasonal climate, and hurricanes," *Reviews of Geophysics*, vol. 54, no. 2, pp. 336–377, 2016.

[37] L. Y. Ji, X. F. Zhi, C. Simmer, S. P. Zhu, and Y. Ji, "Multimodel ensemble forecast of precipitation based on an object-based diagnostic evaluation," *Monthly Weather Review*, vol. 148, no. 6, 2020.

[38] L. ., C. Yann, B. Leon, B. Yoshua, and H. Patrick, "Gradient-based learning applied to document recognition," *IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[39] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14)*, AAAI, Québec City, Québec, Canada, September 2014.

[40] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," NIPS, in *Proceedings of the NIPS 2014 Workshop on Deep Learning*, December 2014.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Santiago, Chile, Dec. 2015.

[43] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Computer Vision - ECCV 2014*, vol. 8689, pp. 818–833, 2014.

[44] X. Xu, R. Mo, X. Yin et al., "PDM: privacy-aware deployment of machine-learning applications for industrial cyber-physical cloud systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5819–5828, 2021.

[45] X. Xu, Q. Huang, H. Zhu et al., "Secure Service offloading for internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[46] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

Hindawi
Complexity
Volume 2021, Article ID 2227459, 10 pages
https://doi.org/10.1155/2021/2227459

Review Article

Trusted Service Evaluation for Mobile Edge Users: Challenges and Reviews

Tingting Shao[1] **Xuan Yang**[2] **Fan Wang**[3] **Chao Yan**[3] **and Ashish Kr. Luhach**[4]

[1]*Medical Information Engineering School, Jining Medical University, Jining, China*
[2]*Weifang Key Laboratory of Blockchain on Agricultural Vegetables, Weifang University of Science and Technology, Weifang, China*
[3]*School of Computer Science, Qufu Normal University, Jining, China*
[4]*Department of Electrical and Communications Engineering, The PNG University of Technology, Lae, Papua New Guinea*

Correspondence should be addressed to Ashish Kr. Luhach; ashish.kumar@pnguot.ac.pg

Received 7 August 2021; Revised 8 September 2021; Accepted 8 December 2021; Published 21 December 2021

Academic Editor: Anirban Chakraborti

With the increasing growth of web services shared in various mobile edge platforms, it becomes necessary to evaluate all the candidates based on their quality of services to reduce the users' service selection cost. However, the service quality data released by service providers cannot be simply deemed as trusted due to various subjective or objective reasons, which further produce a series of serious trust-aware service evaluation problems, including service quality data sparsity and lack of feedback incentive. In view of this, we summarize the challenging issues existing in the current research field of trusted mobile edge service evaluation. Afterward, we review the current research status of the trusted service evaluation in the mobile edge environment and discuss one of the typical application scenarios based on trusted service evaluation, that is, recommender systems, as well as their diverse categories. We believe this research could be helpful in assisting a mobile edge platform to build a trusted reputation system for various smart applications hosted in the mobile edge platform.

1. Introduction

The credibility of network-structure software or web service is vital for building a highly trusted mobile edge computing platform (i.e., edge computing in mobile devices) [1]. Due to the inherent openness and dynamic nature of the mobile edge environment, the running process of web services in a mobile edge platform is often affected by many uncertain factors, which greatly reduces the credibility of service running quality [2]. Therefore, to ensure the normal operation of the mobile edge-based web services or business processes, it is urgent and necessary to study a credibility-guaranteed mechanism for web services. At present, both the academia and industry areas have conducted preliminary explorations and research on the topic of "trusted web services" [3–7] and proposed a series of important research topics, such as trusted selection of web services, trusted combination of collaborative services, and trusted replacement of abnormal services.

Web services selection is the first step for users to invoke Web services and then construct complex mobile edge applications. (Here, the scope of "service" is very wide and comprehensive. Every item that can be provisioned to users could be regarded as a service, e.g., movie, news, blog, commercial products). With the increasing success of service computing technologies in e-Economy [8], e-Science [9], e-Government [10], and other fields, more and more web services are emerging with the same functions in the mobile edge environment. Therefore, when choosing web services, users should not only consider their application requirements in terms of functions but also pay more attention to the nonfunctional quality performance of web services, that is, QoS (quality of service), such as response time, throughput rate. Through objective measurement and evaluation of the QoS natures of each dimension of web services, users can select a web service with the best quality that meets their functional requirements from many similarly available candidate web services to participate in their mobile edge-based business execution process.

However, due to the dynamic and unpredictable services running environment and the business competition from the false propaganda and malicious deception, the service QoS data released by service providers are not always truthful [11–14]. This untrusted QoS data will interfere with the normal service selection process of the user and cause users to make the wrong decision and judgment (such as no credible sensor service QoS cause the failure of fire warning). They will destroy the fair and reasonable competition order between service subjects. Therefore, finding more authentic and reliable sources of QoS data to replace untrusted ones published by service providers is crucial for mobile edge users.

In this paper, we focus on the problems and challenges existing in the field of trusted service evaluation in mobile edge computing. Concretely, the remainder of this paper is structured as follows: in Section 2, we summarize the current search challenges and problems in the trusted service evaluation based on historical service quality records. Afterward, in Section 3, we review the current research literature from two aspects: subjective user rating and objective QoS records. In Section 4, we discuss one of the future application patterns of trusted service quality information, that is, recommender systems in mobile edge computing. Finally, in Section 5, we conclude the paper and analyze the improvement directions in future work.

2. Research Challenges

In a mobile edge computing environment, users tend to leave a record after invoking the web service, such as subjective user rating (e.g., common rating of "1 star" to "5 stars") or objective QoS records (the quality information of the web service in the execution of this invocation, e.g., a web service's response time is 2 seconds). The invocation record more truly reflects the quality of web services in the past; thus, it became one of the most credible bases for measuring the true quality of the web service in the mobile edge environment. At present, academia widely uses the historical invocation record of web services to evaluate the quality of service and select web services to overcome the defect of the unreliability of QoS data published by the service provider in traditional methods [15–18]. However, this method of "web service selection based on historical invocation records in mobile edge environment" still faces many trust problems that need to be solved.

2.1. Incentives and Preprocessing of Sparse User Ratings. First of all, due to the lack of an effective incentive mechanism, users are not highly motivated to make ratings after invoking web services. As a result, user ratings of web services are sparse in the mobile edge environment [19, 20], which greatly reduces the feasibility and accuracy of evaluating the quality of web services through user ratings of web services. Secondly, to ensure the authenticity of user ratings, malicious ratings of bad users (such as deliberate fraud and malicious collusion between service providers and users) should be identified and punished. However, when a user rating is very sparse, the effect of the traditional malicious rating recognition method based on statistical thinking is not good [21, 22]. Moreover, by doping from the subjective preference of the user, the web service user rating is not the unbiased estimator of the quality of the service, so you need to identify and reversely correct subjective preference in user ratings. However, the traditional, preferred rating recognition method based on a statistical idea requires a large number of known user rating data, which is not suitable for very sparse user ratings.

Generally, we regard the situations where feedback is very sparse as cold-start problems, which often render trusted service evaluation infeasible. As inherent ills in the mobile edge environment, many researchers devote their attention to alleviating cold-start problems for better service selection. Wang et al. [23] incorporate user trust into service evaluation and combine trust relationships with rating records to achieve robust service selection. Wang et al. [24] employed a metalearning embedding ensemble (ML2E) algorithm to perform a more accurate evaluation for new services. However, the above studies do not fundamentally solve the cold-start problems, which need to be further studied in the future.

2.2. Protection and Evaluation of QoS Records. Firstly, the QoS records generated by the user after invoking the web service are also a kind of private data. Therefore, for privacy protection, users are not willing to disclose the monitored QoS record [25, 26], which intensifies the sparse QoS record in a mobile edge environment and reduces the feasibility and accuracy of evaluating the quality of web services through the QoS record of web services. Secondly, some QoS natures of web services are not completely independent but correlated with each other [27, 28]. However, the existing web services evaluation methods (such as the commonly used weighted method) do not consider such attribute correlation, thus reducing the accuracy of the evaluation results of quality of service. Moreover, some web services (such as Mobile edge service) have a longer running cycle (such as running for a week), and their quality of service constantly fluctuates during the running cycle. Therefore, some QoS records for this type of web service are not simply fixed values (quality points) but a quality curve that fluctuates over time [29]. However, the existing web services evaluation methods do not consider this special form of QoS record. Therefore, it is easy to cause the one-sidedness and incompleteness of service quality evaluation, thus reducing the accuracy of service evaluation results.

2.3. Weight Allocation of Historical Invocation Records. There are probably multiple historical invocation records for web services that are used more frequently [30] (i.e., multiple user ratings, or multiple QoS records, or a combination of user ratings and QoS records). For the multiple invocation records (such as ratings and rating scores, etc.), the context information (such as invocation time and network environment) are probably distinctive. Thus, multiple invocation records for the same web service are not exactly the same for evaluating and forecasting the web service quality. In addition, for different candidate web services, the number of invocation records (i.e., user ratings or QoS records) also varies. To a certain extent, this will also affect the degree of trust of mobile edge users in web service quality. Therefore, treating all invocation records of all candidate web services equally will result in inaccurate web service selection results.

To sum up, in the mobile edge computing environment, with the increasingly intensified competition of web services and the constantly changing service operating environment, QoS data published by service providers may not be true and reliable. Therefore, predicting the future quality of a web service based on its historical invocation record is one of the effective ways to implement trusted service selection. However, due to the sparsity of user ratings, the diversity of QoS records (e.g., diverse privacy requirements, diverse attribute associations, and diverse record forms), and the difference of invocation records, currently "the selection of web services based on historical invocation records in mobile edge environment" still faces many trust problems that need to be solved urgently. Accordingly, we carry out the research of "trusted service selection based on historical invocation records in mobile edge environment" based on the previous achievements. The ultimate research goal is to provide real and reliable service quality reference data for mobile edge users' web services selection and then provide necessary theoretical and technical support for the development and maintenance of highly reliable network software platform when the QoS data released by the service provider is not credible.

3. Research Review

At present, the academic community has made an active exploration and research on the topic of "web service selection based on historical invocation records under mobile edge environment", and has gained many phased scientific research achievements. The following summarizes the existing research results from two aspects: subjective user rating and objective QoS records (topic distribution and temporal distribution of mentioned literature are shown in Figures 1 and 2).

3.1. Subjective User Rating. It is summarized from four perspectives (briefly introduced in Table 1): feedback incentive of user rating, identification and punishment of malicious rating, identification and correction of preference rating, and weight allocation of user rating.

3.1.1. Feedback Incentive of User Rating. In order to fundamentally solve the sparsity of user rating, an effective incentive mechanism should be designed to improve the enthusiasm of users for feedback rating. Li et al. [16] calculated the "recommendation trust" of each user, and users with high recommendation trust were given priority to get high attention to encourage users to give positive and credible feedback ratings. According to the previous rating data of users, Yu et al. [32] calculated the credibility of their ratings. Users with high credibility will be given priority to get high-quality service recommendations to improve the enthusiasm of users' ratings. However, the incentive basis of the above incentive mechanism is relatively single, and there is no treatment method for repeated rating. Therefore, the incentive effect is relatively limited, which cannot effectively reduce the sparsity of user rating under the mobile edge environment.

3.1.2. Identification and Punishment of Malicious Rating. Malicious rating from bad users will bring great damage and interference to the trust system of the mobile edge platform. Malik et al. identified possible malicious ratings [21] by comparing the rating differences between a single user and a group of users for the same web service. He used a method of analyzing the distribution of a large number of user ratings to find the possible false and malicious rating. Wang et al. detected malicious rating [31] by comparing the normal feedback level and the average step of sampling feedback level. Other malicious rating recognition methods include the recognition method based on pattern analysis [33] and recognition method based on the user registration information [34]. However, the mentioning identification methods aiming at malicious rating mainly depend on a lot of user rating data. Therefore, when a user rating is very sparse, the effect is not satisfactory. In addition, Web services run differently at different times and in different environments, so different users may have different ratings for the same web service. However, the above methods cannot distinguish such normal differences and are easy to "misjudge" the real user ratings.

What is more, in order to encourage users to give real feedback, it is necessary to give a punishment to the malicious ratings from bad users. Witkowski et al. designed a punishment method [35] based on service price. Zhang et al. punished bad users who provided malicious ratings [36] by reducing the trust and attention of bad users. The above punishment methods consider the "benefits" and "risks" of malicious rating by bad users, respectively. However, the punishment basis is relatively single and cannot adapt to the complex web service trust system.

3.1.3. Identification and Correction of Preference Ratings. Identifying the implicit subjective preference in user ratings is conducive to an objective and accurate quality assessment of web services. According to the user's sensitive degree to the quality of service, Li et al. found that users can be divided into tolerant and rigid users. They discussed the rating rules of two types of users: one is positive and the other is negative [16]. Malik and Bouguettaya revealed the positive user rating distribution which is described as to J-shape [21].

3.1.4. Through the Statistical Analysis. In order to minimize the negative impact of the preference rating to evaluate service, we need to do the reverse correction to the preference rating that is identified. Based on user feedback between the ranking

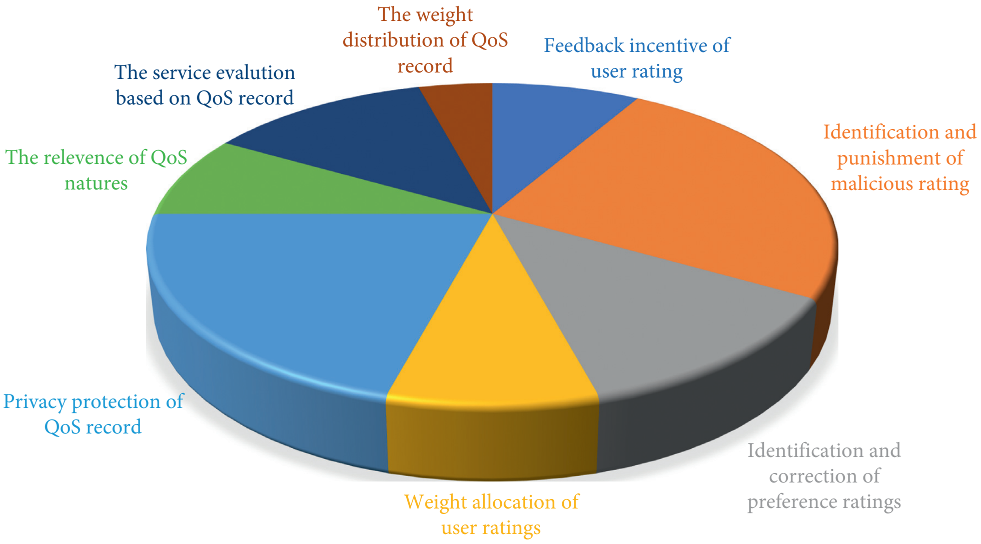


Figure 1: Topic distribution of literature about subjective user rating and objective QoS records.

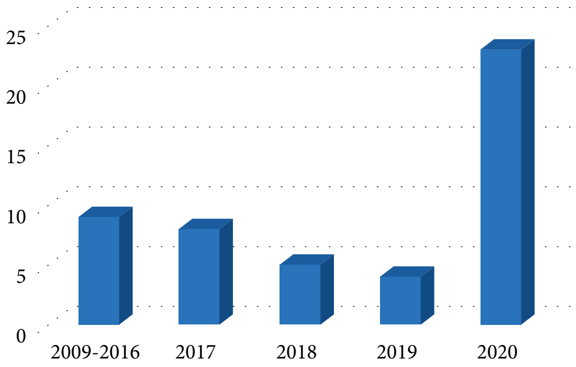


Figure 2: Temporal distribution of recent literature in this study.

Table 1: Research protocol of subjective user rating.

Category	Title	Authors	Year	Publisher
Feedback incentive of user rating	Community-Diversified Influence Maximization in Social Networks [16]	Li et al.	2020	Information Systems
	Hybrid Attacks on Model-Based Social Recommender Systems [32]	Yu et al.	2017	Physica A: Statistical Mechanics & Its Applications
Identification and punishment of malicious rating	Community-Diversified Influence Maximization in Social Networks [16]	Li et al.	2020	Information Systems
	Rater Credibility Assessment in Web Services Interactions [21]	Malik et al.	2009	World Wide Web Journal
	Shilling Attack Detection in Recommender Systems via Selecting Patterns Analysis [33]	Li et al.	2016	IEICE Transactions on Information and System
	Identifying Fake Feedback for Effective Trust Management in Cloud Environments [34]	Noor et al.	2013	LNCS
	A. Krause. Incentive-Compatible Forecasting Competitions [35]	Witkowski et al.	2018	Thirty-Second AAAI Conference on Artificial Intelligence
	Study on the Trust Evaluation Approach Based on Cloud Model [36]	Zhang et al.	2013	Chinese Journal of Computers

Table 1: Continued.

Category	Title	Authors	Year	Publisher
Identification and correction of preference ratings	Community-Diversified Influence Maximization in Social Networks [16]	Li et al.	2020	Information Systems
	Rater Credibility Assessment in Web Services Interactions [21]	Malik et al.	2009	World Wide Web Journal
	Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation System [31]	Wang et al.	2015	IEEE Transactions on Services Computing
Weight allocation of user ratings	A Time-Aware Dynamic Service Quality Prediction Approach for Services [7]	Jin et al.	2020	Tsinghua Science and Technology
	An Attention-Based Category-Aware GRU Model for Next POI Recommendation [15]	Liu et al.	2021	International Journal of Intelligent Systems

and user context information, Wang et al. calculated the current user's virtual rating and the conflict of virtual rating and the actual rating and then did reverse correction to users with large conflicts [31]. However, the above methods for the identification and correction of preference rating mainly depend on many user rating data. When a user rating is very sparse, the effect will not be beautiful.

3.1.5. Weight Allocation of User Ratings. To accurately assess the true quality of a web service, it is necessary to assign different weights to its user ratings. Jin et al. [7] studied the correlation between the rating time and the rating weight. Hu et al. used the user rating score size as the design basis of the weighted rating to weaken the negative impact on objective evaluation from the positive user ratings.

In addition, the credibility of user rating will also affect the contribution from user ratings to service quality evaluation. Liu et al. analyzed the correlation between user's credibility and rating weight [15]. However, the above literature focused more on qualitative analysis of various factors affecting rating weight, lack of quantitative theoretical analysis and data support, which results in inability to effectively support the quality assessment of web services based on multirating weighted aggregation.

3.2. Objective QoS Record. It is summarized from the privacy protection of QoS record (briefly introduced in Table 2), QoS relevance, service evaluation based on QoS record, and weight distribution of QoS record.

3.2.1. Privacy Protection of QoS Record. In order to protect the privacy information in QoS records, Razaque et al. [37] incorporated QoS privacy into the contract of SLA (service protection of QoS privacy data through classified privacy.

3.2.2. Contracts. Zhang et al. [41], Wang et al. [43], and Khazbak et al. [44] discussed the possible privacy leakage issues in various domains. For example, the authors introduced the privacy exposure problems and challenges existing in current various sharing economy services, including biking location privacy. In other words, when people are enjoying the convenient services provisioned by biking rental enterprises, they are often confronted with hidden and unsecure privacy issues because the sensors and GPS modules embedded in bikes will monitor and collect the real-time user location information at any time and any place. Moreover, Meng et al. [38] and Wang et al. [39, 40] alleviated the privacy leakage issues of QoS records transmission on the distributed computing platforms. However, all the above literature only studied the privacy protection of QoS records from a higher level and perspective and lacked specific solutions. Therefore, the effect of privacy protection is relatively limited and cannot effectively eliminate users' worries about QoS privacy leakage.

3.2.3. The Relevance of QoS Natures. Part of the QoS natures of Web services is not completely independent but related. Luo and others modeled the relationship among the QoS attributes through the service-related model BSCM [27]. They analyzed the reverse relationship between different QoS attributes. Zhong et al. used the TOPSIS method in multiobjective optimization to make a dimension reduction aimed at multidimensional and associated QoS attributes of a web service [28]. To a certain extent, it has weakened the relevance of QoS natures' negative influence on the service quality evaluation. However, the above literature focused on modeling and qualitative description of QoS natures correlation, lacking quantitative correlation calculation. Many QoS records are needed to support the calculation of attribute correlation, but the scope of application is narrow. In addition, the above literature did not discuss the nonlinear correlation between different QoS natures of web services, which further reduced its application scope.

3.2.4. The Service Evaluation Based on QoS Record. After the QoS records generated after the web service are invocated, trusted web services can be evaluated, selected, and combined. Zhang et al. used the historical records of web services to predict the future preferences of users and make appropriate recommendation decisions [41]. Zhong et al. [28] used the QoS records of web services to select the best web service through multiquality evaluation. Malik et al. [21] determined the quality of the web service credibility by comparing the web service QoS record with its promise of SLA quality level. However, the above literature all assumed that the web service QoS record is a

Table 2: Research protocol of objective user rating.

Category	Title	Authors	Publication year	Publisher
Privacy protection of QoS record	Privacy Preserving Model: A New Scheme for Auditing Cloud Stakeholders [37]	Razaque et al.	2017	Journal of Cloud Computing
	Amplified LSH-Based Recommender Systems with Privacy Protection [25]	Chi et al.	2020	Concurrency and Computation: Practice and Experience
	Security-Aware Dynamic Scheduling for Real-Time Optimization in Cloud-Based Industrial Applications [38]	Meng et al.	2021	IEEE Transactions on Industrial Informatics
	An Optimization and Auction Based Incentive Mechanism to Maximize Social Welfare for Mobile Crowdsourcing [39]	Wang et al.	2019	IEEE Transactions on Computational Social Systems
	A Novel Hybrid Method to Analyze Security Vulnerabilities in Android Applications [40]	Tang et al.	2020	Tsinghua Science and Technology
The relevance of QoS natures	Business Correlation-Aware Modeling and Services Selection in Business Service Ecosystem [27]	Luo et al.	2013	International Journal of Computer Integrated Manufacturing
	Multi-Dimensional Quality-Driven Service Recommendation with Privacy-Preservation in Mobile Edge Environment [28]	Zhong et al.	2020	Computer Communications
The service evaluation based on QoS record	Deep Sequential Model for Anchor Recommendation on Live Streaming Platforms [41]	Zhang et al.	2021	Big Data Mining and Analytics
	Multi-Dimensional Quality-Driven Service Recommendation with Privacy-Preservation in Mobile Edge Environment [28]	Zhong et al.	2020	Computer Communications
	Rater Credibility Assessment in Web Services Interactions [21]	Malik et al.	2009	World Wide Web Journal
The weight distribution of QoS record	How Textual Quality of Online Reviews Affect Classification Performance: A Case of Deep Learning Sentiment Analysis [42]	Li et al.	2020	Neural Computing and Applications

simple fixed value (i.e., the quality point), the diversity of QoS record is not considered under the mobile edge environment (i.e., quality, quality curve) and their integration problem. Therefore, it is likely to cause the partial and incomplete problem of service quality evaluation, thus reducing the accuracy of the service evaluation results.

3.2.5. The Weight Distribution of QoS Record. In order to evaluate the real quality of a web service more accurately, different weights should be assigned to each QoS record of a web service. Li et al. analyzed the correlation between QoS record time and the weight [42]. At present, there are few studies on this aspect, which cannot effectively support web service quality assessment based on the weighted aggregation of multi-QoS records. Other similar work can be found in [10, 45–47], where the multiple-dimensional weighting issue is studied in various ways.

4. Future Directions: Service Quality-Driven Recommendation

Through utilizing the existing and known web service quality data (including objective QoS records and subjective user ratings), we can perform personalized service recommendation for prospective users. This section discusses one of the research directions using service quality information: recommender systems. Generally, the research field can be divided into the following six categories.

4.1. Content-Based Recommender Systems. Service content is mainly about the details of "what" the service executed by users involves or discusses. For example, a user watches a movie named "Roman Holiday" whose actress is Audrey Hepburn and the movie genre is Love. Then, according to content-based recommendation theory, in the future, the user would be recommended the movies whose actress is Audrey Hepburn and whose genre is Love. In other words, content-based recommendation only considers the information contained in the content of the services ever executed by users, without involving other people.

The advantages of content-based recommendation theory are that it does not need to consider the information of other people. Instead, it only needs to know about the historically executed service information of a target user himself. Therefore, if the historical service execution data are rare or sparse, a content-based recommendation is a promising solution to alleviate the sparse data or cold-start recommendation issues.

4.2. Knowledge-Based Recommender Systems. Knowledge is often a key information source in various computing-intensive smart or intelligent applications, such as gambling, chess, and mathematical reasoning. Likewise, in recommender systems, knowledge also plays an essential role in outputting a group of high-quality recommended items. For example, if TV says it will be rainy today, then you would be recommended to take an umbrella when you intend to go outside, as there is an obvious knowledge between rain and an umbrella. Besides the obvious knowledge mentioned above, there is also various knowledge that is hidden and implicit. For example, if Alice took a taxi to the hospital at 2 : 00 a.m., there is an implicit knowledge that Alice was very sick. Such implicit knowledge also contributes much to improving user satisfaction when obvious knowledge is absent from the decision-making process.

Typically, a knowledge graph (KG) provides a promising way for service recommendation and draws attention to researchers in the field of knowledge-based recommender systems. Zhang et al. [48] modeled the collaborative filtering problem as a knowledge graph for link prediction and recommendation. However, this research does not consider privacy protection. In light of this, Yu et al. [49] employed the Laplacian noise to optimize recommendation process based on KG. However, the above literature only leveraged a single relationship to construct the KG framework, which is difficult to cover multiple relationships in practice. To address this issue, Shi et al. [50] put forward a multidimensional knowledge graph framework to recommend personalized learning paths for E-learners. In summary, the advantage of knowledge-based recommendation is that it is precise and accurate as the knowledge can capture the users' preferences well, while the disadvantage is that the knowledge is often not easy to capture as sometimes it is hidden in data and implicit enough.

4.3. Association Rule-Based Recommender Systems. Association rule is the valuable information contained in the correlation among different data dimensions. Association rule implies the hidden knowledge extracted from big data and can be used to make directional information reduction. For example, we can infer or predict the future pork price in a certain time period through analyzing the user reviews, blogging, and sum-ups recorded on the Web because there is an association rule between the pork price and the web information. This way, through association rules, we can reduce the heavy burden on frequent economic statistics activities.

The advantages of the association rules-based recommendations are that the association rules are mined from big data and can accurately reflect the correlation relationships among different things involved. The disadvantages are that association rules are often difficult to mine and obtain, especially when the available data for mining are sparse.

4.4. Collaborative Recommender Systems. The collaborative recommendation is one of the most understandable recommendation manners and has been widely employed in various industrial fields. The basic idea of collaborative recommendation is through similarity calculation. For example, if Alice is a similar friend to Tom, then we can recommend the things liked by Alice to Tom, vice versa, which is the basic idea of user collaboration-based recommender systems. Another example is that if a user likes Coca-Cola, then we can recommend Pepsi Cola to them as Coca-Cola and Pepsi Cola are similar drinks to some extent, which is the basic idea of item collaboration-based recommender systems. Thus, through calculating various similarity values, we can make corresponding recommendations to users.

The advantage of this recommendation way is that it is easy to interpret and can be applied to various fields. The disadvantage is that it fails in delivering a quick response as the similarity calculation is often computation-intensive and time-consuming. Therefore, it is not very suitable for the big data application environment where quick responses are needed.

4.5. Demography-Based Recommender Systems. Demography contains a variety of useful information that depicts users' profiles, such as the users' age, salary, sex, education degree, and working positions. These pieces of individual information collectively constitute the personalized profile of a user and, therefore, can predict the user preferences well. For example, a professor in a university is apt to buy the tools associated with education; a rich man often buys some luxury goods.

The rationale behind demography-based recommender systems is easy to interpret, which is the main advantage of this recommendation way. On the contrary, the disadvantage is that it cannot capture the user preferences accurately and dynamically, as user preferences are often variable with time and not fixed at all. Therefore, it is often inappropriate to use only the demography information for a successful recommender system.

4.6. Hybrid Recommender Systems. If a recommender system combines more than one recommendation technique, it is called a hybrid recommender system. Generally, any successful recommender system is hybrid, such as Amazon, Alibaba, and eBay, as hybrid recommender systems can integrate the advantages of all the involved recommendation manners. As a result, hybrid recommender systems can bring better user experiences and satisfaction.

5. Conclusions

Trusted service evaluation based on historical service quality records is crucial for a mobile edge platform to build a dependable service reputation system. However, due to dynamic service execution context and malicious commercial competition, the QoS data released by service providers are often not trusted, especially for a newcomer of a mobile edge platform. Given this drawback, we review the current literature of the trusted service evaluation in mobile edge computing and analyze the challenging issues existing in the field. As a promising extension, we discuss one of the killer applications of trusted service evaluation: recommender systems. We believe this research could be helpful in assisting a mobile edge platform build a trusted reputation system so as to assist the successful deployment of various service-related smart applications.

In the mobile edge computing environment, data collaboration among different mobile devices or edge terminals is inevitable [51–59]. Therefore, how to secure user privacy (including privacy measurement) while guaranteeing other conflicting performances in service evaluation is an open research issue that calls for intensive study. In addition, computational overload is normal in the big data environment [60–66]. Therefore, how to effectively offload the heavy computational tasks or jobs in peak time still requires challenging efforts.

Data Availability

This study is a review article, so no data are available.

Conflicts of Interest

The authors declare no conflicts of interest regarding the submission, and the manuscript has not been submitted to other journals or conferences for consideration.

References

[1] M. R. Khosravi and S. Samadi, "BL-ALM: a Blind Scalable edge-Guided Reconstruction filter for smart environmental monitoring through green IoMT-UAV networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 727–736, 2021.

[2] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-Sanitization for Preventing sensitive information inference Attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[3] M. R. Khosravi and S. Samadi, "Reliable data aggregation in Internet of ViSAR Vehicles using Chained Dual-Phase Adaptive Interpolation and data embedding," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2603–2610, 2020.

[4] W. Zhang, Z. Li, and X. Chen, "Quality-aware user Recruitment based on Federated learning in mobile Crowd sensing," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 869–877, 2021.

[5] Z. Xue and H. Wang, "Effective Density-based Clustering algorithms for incomplete data," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 183–194, 2021.

[6] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple Parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[7] Y. Jin, W. Guo, and Y. Zhang, "A time-aware dynamic service quality prediction Approach for services," *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 227–238, 2020.

[8] E. Jiang, L. Wang, and J. Wang, "Decomposition-based multi-objective optimization for Energy-aware distributed hybrid Flow Shop Scheduling with Multiprocessor tasks," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 646–663, 2021.

[9] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data Logger for Weather monitoring using Arduino-based Wireless sensor networks with Remote Graphical application and Alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[10] R. Kumari, S. Kumar, R. C. Poonia et al., "Analysis and predictions of Spread, Recovery, and Death caused by COVID-19 in India," *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 65–75, 2021.

[11] F. Wang, H. Zhu, G. Srivastava, S. Li, M. R. Khosravi, and L. Qi, "Robust collaborative filtering recommendation with user-item-trust records," *IEEE Transactions on Computational Social Systems*, pp. 1–11, 2021.

[12] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A Blockchain-enabled Deduplicatable data Auditing mechanism for network Storage services," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1421–1432, 2021.

[13] L. Zhang, J. Liu, F. Shang, G. Li, J. Zhao, and Y. Zhang, "Robust Segmentation method for Noisy Images based on an Unsupervised Denosing filter," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 736–748, 2021.

[14] Y. Xu, Ju Ren, Y. Zhang, C. Zhang, Bo Shen, and Y. Zhang, "Blockchain Empowered Arbitrable data Auditing Scheme for network Storage as a service," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2020.

[15] Y. Liu, A. Pei, F. Wang et al., "An attention-based Category-aware GRU model for Next POI recommendation," *International Journal of Intelligent Systems*, vol. 36, 2021.

[16] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Information Systems*, vol. 92, pp. 1–12, 2020.

[17] Q. Liu, G. Wang, X. Liu, T. Peng, and J. Wu, "Achieving reliable and secure services in cloud computing environments," *Computers & Electrical Engineering*, vol. 59, pp. 153–164, 2017.

[18] X. Liu, Q. Liu, T. Peng, and J. Wu, "Dynamic access policy in cloud-based personal health record (PHR) systems," *Information Sciences*, vol. 379, no. 2, pp. 62–81, 2017.

[19] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized Travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[20] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data Fusion and prediction with spatial-temporal context for smart City industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, 2020.

[21] Z. Malik and A. Bouguettaya, "Rater credibility assessment in web services Interactions," *World Wide Web Journal*, vol. 12, no. 1, pp. 3–25, 2009.

[22] T. Cai, J. Li, S. M. Ajmal, R. Li, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[23] F. Wang, W. Zhong, X. Xu, W. Rafique, Z. Zhou, and L. Qi, "Privacy-aware cold-start recommendation based on collaborative filtering and Enhanced trust," in *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Sydney, NSW, Australia, October 2020.

[24] H. Wang and Y. Zhao, "ML2E: meta-learning embedding ensemble for cold-start recommendation," *IEEE Access*, vol. 8, Article ID 165757, 2020.

[25] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified lsh-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, 2020.

[26] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A Survey of data Partitioning and sampling methods to support big data analysis," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 85–101, 2020.

[27] Y. Luo, Y. Fan, and H. Wang, "Business correlation-aware modeling and services selection in business service Ecosystem," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 8, pp. 772–785, 2013.

[28] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-Driven service recommendation with privacy-Preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[29] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware Cross-platform service recommendation based on Enhanced Locality-sensitive Hashing," *IEEE Transactions on Network Science and Engineering*, vol. 8, 2020.

[30] J. Hu, Yi Pan, T. Li, and Y. Yang, "TW-Co-MFC: two-level weighted collaborative Fuzzy Clustering based on Maximum Entropy for multi-view data," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 185–198, 2021.

[31] S. Wang, Z. Zheng, Z. Wu, R. Lyu, and F. Yang, "Reputation measurement and malicious feedback rating Prevention in web service recommendation system," *IEEE Transactions on Services Computing*, vol. 8, pp. 755–767, 2015.

[32] J. Yu, M. Gao, W. Rong, W. Li, Q. Xiong, and J. Wen, "Hybrid Attacks on model-based social recommender systems," *Physica A: Statistical Mechanics and its Applications*, vol. 483, pp. 171–181, 2017.

[33] W. Li, M. Gao, H. Li, J. Zeng, Q. Xiong, and S. Hirokawa, "Shilling Attack detection in recommender systems via selecting patterns analysis," *IEICE Transactions on Information and System*, vol. E99-D, no. 10, pp. 2600–2611, 2016.

[34] T. H. Noor, Q. Z. Sheng, A. Abdullah, J. Law, H. Anne, and H. Ngu, "Identifying Fake feedback for effective trust Management in cloud environments," *Lecture Notes in Computer Science*, Springer, vol. 7759, pp. 47–58, , Berlin, Heidelberg, 2013.

[35] J. Witkowski, R. Freeman, J. W. Vaughan, and D. M. Pennock, "Andreas Krause. Incentive-Compatible Forecasting Competitions," in *Proceedings of the Thirty-Second Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, February 2018.

[36] S. Zhang and C. Xu, "Study on the trust evaluation Approach based on cloud model," *Chinese Journal of Computers*, vol. 36, no. 2, pp. 422–431, 2013.

[37] A. Razaque and S. S. Rizvi, "Privacy preserving model: a new scheme for auditing cloud stakeholders," *Journal of Cloud Computing*, 2017.

[38] S. Meng, W. Huang, X. Yin et al., "Security-aware dynamic Scheduling for real-time optimization in cloud-based industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4219–4228, 2021.

[39] Y. Wang, Z. Cai, Z.-H. Zhan, Y. Gong, and X. Tong, "An optimization and Auction based incentive mechanism to Maximize social Welfare for mobile Crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 414–429, 2019.

[40] J. Tang, R. Li, K. Wang, X. Gu, and Z. Xu, "A Novel hybrid method to analyze Security Vulnerabilities in Android applications," *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 589–603, 2020.

[41] S. Zhang, H. Liu, J. He, S. Han, and X. Du, "Deep Sequential model for Anchor recommendation on live Streaming platforms," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 173–182, 2021.

[42] Li Lin, T.-T. Goh, and D. Jin, "How Textual quality of online reviews affect Classification performance: a Case of Deep learning Sentiment analysis," *Neural Computing & Applications*, vol. 32, pp. 4387–4415, 2020.

[43] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-Preserving for mobile Crowdsourcing systems," *Computer Network*, vol. 135, pp. 32–43, 2018.

[44] Y. Khazbak, J. Fan, S. Zhu, and G. Cao, "Preserving personalized location privacy in Ride-Hailing service," *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 743–757, 2020.

[45] M. R. Khosravi and S. Samadi, "Data Compression in Visar sensor networks using non-linear adaptive weighting," *EURASIP Journal on Wireless Communications and Networking*, vol. 264, 2019.

[46] Y. Wang, Z. Cai, Z.-H. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian Equilibrium-based multi-objective optimization for task allocation in mobile Crowdsourcing," *IEEE Transactions on Computational Social Systems*, 2020.

[47] M. R. Khosravi, "ACI: A bar Chart Index for non-linear Visualization of data embedding and aggregation Capacity in IoMT multi-source Compression," *Wireless Networks*, 2021.

[48] Y. Zhang, J. Wang, and J. Luo, "Knowledge graph embedding based collaborative filtering," *IEEE Access*, vol. 8, Article ID 134553, 2020.

[49] B. Yu, C. Zhou, C. Zhang, G. Wang, and Y. Fan, "A privacy-Preserving multi-Task framework for knowledge graph Enhanced recommendation," *IEEE Access*, vol. 8, Article ID 115717, 2020.

[50] D. Shi, T. Wang, H. Xing, and H. Xu, "A learning path recommendation model based on A multidimensional knowledge graph framework for E-learning," *Knowledge-Based Systems*, vol. 195, Article ID 105618, 2020.

[51] Y. Li, H. Ma, L. Wang, S. Mao, and G. Wang, "Optimized content Caching and user association for edge computing in Densely deployed Heterogeneous networks," *IEEE Transactions on Mobile Computing*, 2020.

[52] M. R. Khosravi, H. Basri, H. Rostami, and S. Samadi, "Distributed random Cooperation for VBF-based Routing in high-Speed Dense Underwater Acoustic sensor networks," *The Journal of Supercomputing*, vol. 74, no. 11, pp. 6184–6200, 2018.

[53] Y. Li, S. Xia, M. Zheng, B. Cao, and Q. Liu, "Lyapunov optimization based Trade-Off policy for mobile cloud offloading in Heterogeneous Wireless networks," *IEEE Transactions on Cloud Computing*, 2019.

[54] C. Hu, W. Fan, E. Zen et al., "A Digital Twin-Assisted real-time Traffic data prediction method for 5G-enabled Internet of Vehicles," *IEEE Transactions on Industrial Informatics*, 2021.

[55] X. Zhou, L. Yue, and W. Liang, "CNN-RNN based intelligent recommendation for online Medical Pre-Diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2020.

[56] M. R. Khosravi and S. Samadi, "Efficient payload communications for iot-enabled visar vehicles using discrete cosine transform-based quasi-sparse bit injection," *EURASIP Journal on Wireless Communications and Networking*, 2019.

[57] J. Cai, Z. Huang, L. Liao, J. Luo, and W.-Xi Liu, "APPM: Adaptive Parallel processing mechanism for service function Chains," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1540–1555, 2021.

[58] J. Luo, J. Li, L. Jiao, and J. Cai, "On the effective Parallelization and Near-Optimal deployment of service function Chains," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1238–1255, 2021.

[59] Y. Li, Z. Zhang, S. Xia, and H. H. Chen, "A Load-Balanced Re-embedding Scheme for Wireless network Virtualization," *IEEE Transactions on Vehicular Technology*, vol. 70, 2021.

[60] X. Hu, S. Peng, B. Guo, and P. Xu, “Accurate AM-FM Signal Demodulation and Separation using Nonparametric Regularization method,” *Signal Processing*, vol. 186, pp. 108–131, 2021.

[61] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, “Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, 2020.

[62] X. Xu, Z. Fang, J. Zhang et al., “Edge content Caching with Deep Spatiotemporal Residual network for IoV in smart City,” *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[63] X. Hu, S. Peng, and W. L. Hwang, “EMD Revisited: a new understanding of the Envelope and Resolving the Mode-Mixing problem in AM-FM Signals,” *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1075–1086, 2012.

[64] X. Zhou, X. Xu, W. Liang et al., “Intelligent Small Object detection based on Digital Twinning for smart Manufacturing in industrial CPS,” *IEEE Transactions on Industrial Informatics*, 2021.

[65] Y. Li, S. Xia, Q. Yang, G. Wang, and W. Zhang, “Lifetime-priority-driven Resource allocation for WNV-based Internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4514–4525, 2021.

[66] X. Zhou, W. Liang, K. I. K. Wang, H. Wang, T. Y. Laurence, and Q. Jin, “Deep learning Enhanced Human activity recognition for Internet of Healthcare things,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

Hindawi
Complexity
Volume 2021, Article ID 4799921, 14 pages
https://doi.org/10.1155/2021/4799921

Research Article

Collaborative Big Data Management and Analytics in Complex Systems with Edge 2021 eaCamera: A Case Study on AI-Based Complex Attention Analysis with Edge System

Chaopeng Guo, Peimeng Zhu, Feng Li, and Jie Song

Northeastern University, No. 195, Chuangxin Road, Hunnan District, Shenyang 110169, Liaoning, China

Correspondence should be addressed to Jie Song; songjie@mail.neu.edu.cn

Received 23 September 2021; Revised 24 October 2021; Accepted 2 November 2021; Published 27 November 2021

Academic Editor: Xuyun Zhang

As an extension of cloud computing, edge computing makes up for the deficiency of cloud computing to a certain extent. Edge computing reduces unnecessary data transmission and makes a significant contribution to the real-time and security of the system due to its characteristics that are closer to the terminal equipment. In this paper, we study the problem of attention detection. Attentional concentration during some specific tasks plays a vital role, which indicates the effectiveness and performance of human beings. Evaluation of attentional concentration status is essential in many fields. However, it is hard to define the behavior features related to the variety of tasks and behaviors. To solve this problem, we propose an intelligent edge system for attention concentration analysis, eaCamera, to recognize attentional concentration behaviors of students at the edge. To make objective measurements and save the label cost, eaCamera utilizes AI approaches to find the concentration behaviors based on a behavior analysis model with two perspectives, namely, individual perspective and group perspective. Individual perspective indicates personal behavior changes in time dimension while group perspective indicates the changes of the behavior within a group behavior manner. To evaluate the proposed system, a case study is done within a primary school to evaluate student's performance in the classroom and offer teaching advice for teachers.

1. Introduction

At present, an attention concentration analysis system is used to track an individual's attention state to obtain its attention duration. Cognitive research shows that it is vital for success in any field of skilled performance [1]. Improving the period of attentional concentration is essential in lots of fields and scenarios.

However, the concentration status cannot be observed directly [2]. Researchers, especially in cognitive computing and computer vision, try to capture the concentration status-related features like head poses, eye movements, emotions, behaviors, and visual attention to estimate concentration status. In an aspect of different elements, two kinds of attention definitions have been adopted [3]:

(i) Object-driven: attention is the process of attending to objects. Attention is object-based, so attention can also be defined as how much you focus on an object.

(ii) Task-driven: the human brain is always focused on the task at hand, which is related to the high-level mental information in the human mind. When a human is doing a task, it is a kind of high-level information in the mind, guiding human attention [5].

Visual attention is a common approach based on object-driven attention that highlights the essential regions of an image where human observers would allocate their attention at first glance. The comprehensive method to represent visual attention is by eye fixation saliency object [6–8] or saliency map [9, 10]. In these methods, the saliency object or map is usually obtained by the eye-tracking equipment that records the eye fixations of the observer looking at the image.

To drive the research of attention analysis, we implement an edge intelligence system for attention analysis. The edging machine collects data, processes the data through the countermeasure instance detection method [11], analyzes the attentional concentration, and generates a report. The data are regularly transmitted to the cloud server to avoid data loss.

Attention detection cannot avoid the processing of the massive video. If the video is transmitted to the cloud for processing each time, it will consume a lot of time and energy in data transmission and processing. Meanwhile, it introduces nonnegligible delay. In addition, our study uses cutting-edge image processing methods for attention detection. Recognizing attention through images includes multiple tasks, such as face recognition, attention recognition, and so on. Moreover, these tasks are carried out at the edge devices. The edge devices themselves are limited by computing resources, so it must be implemented by using lightweight techniques with certain accuracy. In the intelligent edge system implemented in this research, the edge device with limited computing resources acts as the agent of the cloud to process the video. Therefore, this study promotes the research of edge-supported complex systems to a certain extent.

In the attention concentration detection part of the intelligent edge system, the attentional concentration is defined based on task-driven attention presented by task-level features. The task-level features are related to the task type and its environment. Since there a variety of tasks and different environments, it is hard to describe or find all the task-related attentional behaviors. According to the research, facial expression can be recognized according to the detection of facial feature points [12].

In recent years, information technology has been more and more widely used in the education industry. With the thriving of AI technology, its applications in education have been increasing, with a promising potential to provide customized learning, offer dynamic assessments, and facilitate meaningful interactions in online, mobile, or blended learning experiences [13]. Online education services such as Google Classroom, Zoom, and Microsoft Team also emerge one after another [14]. In turn, the research on the education industry is conducive to the design of extensible neuromorphic complete hardware primitives and the corresponding chips [15]. Therefore, we focus on the enclosed multiple people space: educational environment. In such environments, we have the following observations:

(i) Individual behavior will change over time, which indicates that the concentration status changes as well. For example, when the teacher asks students to read their textbook, there is a student who read the textbook at the beginning but looked at the window later. By comparing the behavior changes from reading the textbook to looking at the window, the concentration changes of the individual can be captured.

(ii) Group behavior indicates the attentional concentration status in general. In the enclosed space, especially like a classroom in the educational environment or a factor in the industry, most people follow the instrument and concentrate on finishing their tasks, which indicates that most people would have the same behavior according to the instructions. Therefore, the outlier behavior can be captured, and the people are in abstracted status.

According to the above observations, a study case, eaCamera (A Case Study on AI-based Complex Attention Analysis with Edge System), is introduced in this paper. EaCamera focuses on the educational scenario and enclosed multiple people space. It is deployed in a primary school classroom to obtain the concentration duration in the lecture time for students. At first, eaCamera accepts the raw video of the lecture. Later, the video goes through the compute vision pipeline and behavior analysis pipeline to analyze the students' concentration status in the classroom on edge machines. In the end, eaCamera provides static reports to teachers, which can be used to downstream teaching tasks, for example, analyzing teaching performance and improving teaching approaches. In eaCamera, we propose a novel attentional concentration analysis model, which captures task-related attentional behaviors from two perspectives, namely, individual and group perspectives. The analyzed model is an unsupervised learning process, which detects attentional behaviors automatically.

The rest of the paper is organized as follows: Section-related work summarizes related works that inspire this work; Section system architecture describes the system architecture of eaCamera and explains its mechanism; Section attentional concentration analysis proposes an attention control analysis module, which is the critical component within eaCamera; Section case result describes the study case in which we deploy eaCamera within a local primary school to evaluate the proposed approach; The Conclusion section summarizes the work.

2. Related Work

Many attentional concentration types of research within the computer science area focus on the object-driven attention model, and a few focus on the task-driven model. The attention concentration analysis system is undoubtedly a waste of time and resources to transmit data to the cloud for analysis, and the intelligent edge system can solve this problem. Therefore, we introduce the basis of the edge intelligence system: edge computing firstly. Then, we review visual-based attention works. Later, we review joint attention approaches, which inspire our group behavior model. In the end, some related deep learning models, considered fundamental approaches of this study, are introduced.

2.1. Edge Computing. Edge computing refers to an open platform integrating network, computing, storage, and application core capabilities on the side close to the object or data source to provide the nearest end services. Although the security design of Internet of things application based on edge computing is still in its infancy, there are many security solutions of edge layer Internet of things recently, which makes edge computing have great application value [16]. Compared with the system architecture of cloud computing, edge computing architecture is closer to the edge side of users and terminal devices. Compared with the distance from the user to the cloud side, the distance from the user to the edge machine is negligible [17]. Therefore, users can get a faster response. At the same time, edge computing also has a cooperation mechanism [18], which solves the privacy and trust problems in large data-driven complex systems to a certain extent. Parallel optimization algorithms and joint optimization algorithm based on reinforcement learning can also be used in edge machines to reduce task execution delay and control additional resource consumption [19, 20].

Due to the geographical dispersion of cloud data centers, the storage and processing needs of billions of geographically distributed sensors are often not met. The result is network congestion and high-latency delivery in the service, which may cause a reduction in quality of service (QoS). Typically, edge computing is made up of traditional network elements such as routers, switches, proxy servers, and base stations (BSs). It can be placed closer to IoT sensors. These components are provided with a variety of computing, storage, networking, and other functions that can support the execution of service applications [21].

At present, there are many attention detection models. Still, due to the lack of advanced hardware resources, these prediction models cannot be applied to the analysis of daily life tasks. Edge intelligent system has high application value in this problem. EaCamera captures and processes the video at the edge side, analyzes the attention concentration status, and sends the generated data to the cloud side for display. This avoids a large amount of video data transmission and saves a lot of resources and time consumption.

2.2. Visual Attention.

The typical workflow of detecting saliency-based visual attention is predicting a saliency object/map, then minimizing the loss, which stands for the difference between the prediction and the ground truth. To predict a saliency map or object, single-stream networks are used to extract the feature map in early works. However, a single-stream network is unable to extract multiple-scale cues. Thus, Xun Huang proposes to use multiple stream networks [22]. Due to the importance of feature map extraction in visual problems, many works have been done to study how to extract features. Bo Du and Wei Xiong propose models to extract features in an unsupervised framework and achieve excellent performance [23, 24]. Recently, a study proposed that the first layers of the network capture macroinformation and the latter layers capture detailed information [25]. A novel architecture is proposed to extract features by combining different layers [26, 27]. A detailed survey for saliency object detection can be found in work [28, 29].

While saliency-based visual attention study about the attention of an outside human image, in many actual circumstances, we need to infer the attention of a human from the third-person view of inside images. In a typical scenario, we need to figure out the attentional concentration object inside an image or video.

Hyeonggyu Park proposes to infer human attention by using their eye movement patterns [30]. Participants were asked to view pictures while operating under different intentions, and a classic support vector machine algorithm was used to infer participants' attention. However, this method revealed low classification accuracy due to significant inter-individual variance and psychological factors underlying intentions. Thus, using eye movement alone is not sufficient to infer human attention.

Ping Wei proposes a probabilistic method to infer the third-person view human attention based on the latent intention by jointly modeling attention, intentions, and interactions [31]. It models the attention inference as a joint optimization with latent intentions. In this paper, an EA-based approach is adopted to learn latent intentions and model parameters. Given a video with human skeletons, a joint-state dynamic programming algorithm is utilized to infer the attention direction.

The above methods define attention as the process of attending to objects. Zhixiong Nan proposes a model to infer task-driven, inside-image human attention [3]. It defines human attention as attentional objects that coincide with the task the human is doing and suggests that a human finishes the task by doing several sub-tasks in a certain temporal order with a task in mind [32]. This literal uses a model that integrates the low-level visible human pose cue with the high-level invisible task encoding information to infer human attention inside a VR video.

In this paper, eaCamera takes advantage of visual attention approaches to implement the attentional concentration analysis system. We mainly follow the task-driven branch to evaluate the students' concentration status based on high-level behavior features [33]. However, in the implementation, we do not label the concentrated behavior and abstract behavior. A concentration analysis model with two perspectives is proposed to recognize the attentional concentration status automatically.

2.3. Joint Attention.

Joint attention is a behavior in which two people focus on an object or event to interact with each other. It is a form of early social and communicative behavior [34]. Joint attention involves sharing a shared focus on something (such as other people, objects, concepts, or events) with someone else. It requires the ability to gain, maintain, and shift attention. For example, a parent and child may both look at a toy they are playing with or observe a train passing by. Joint attention (also known as "shared attention") may be gained by using eye contact, gestures (e.g., pointing using the index finger), and vocalizations, including spoken words (e.g. "look over there").

In cognitive computing and artificial intelligence, joint-attention-related approaches are used in building human-robot interaction systems. Wallace Lawson introduced joint attention mechanism in robotic systems based on the assumption that when robots have joint attention with a human collaborator [35], they are perceived as more competent and more socially interactive [36]. They proposed a joint attention estimator that creates many possible candidates for joint attention and chooses the most likely object based on the human teammate's hand cues. The human attentional objects are found based on visual attention approaches.

Some kinds of literature use joint attention mechanisms to implement multi-perspective video analysis systems. Zhaohui Yang challenged cross-view video co-analysis and delivered a novel learning-based method, in which "joint attention" is used as core notion, indicating the shared attention regions that link the corresponding views [37].

EaCamera takes advantage of joint attention to evaluate students' concentration status, namely, the group perspective within the attentional concentration model. When we cannot determine the concentration status based on individual behavior changes, the status is determined according to the group manner, based on joint attention theory.

2.4. Deep Learning Models. With the development of computer vision and deep learning, more and more deep learning models are proposed to interpret images and videos. The primary tasks of these deep learning models include classification, object detection, segmentation, action recognition, etc. Object detection and facial landmark detection are two fundamental tasks within eaCamera to capture the behavior features of the student.

Object Detection, one of the most fundamental and challenging problems in computer vision, seeks to locate object instances from many predefined categories in natural images [38]. In recent years, plenty of deep learning models have been proposed to cope with the task, Fast-RCNN [39], Yolo [40], RetinaNet [41], CornerNet [42], and so on. In eaCamera, we choose yolov3 to implement the object detection module since its simplicity and high performance.

Facial Landmark Detection predicts the locations of the fiducial facial landmark points around facial components and facial contour to capture the rigid and nonrigid facial deformations due to head movements and facial expressions [43]. Lots of works have been done on this problem to detect facial key points automatically. DAN [44] is used to implement the facial landmark detection module. EaCamera uses facial key points to estimate the head pose and emotions, which are keys to determine the students' behavior.

3. System Architecture

EaCamera is an attentional concentration analysis system deployed in a primary school to automatically obtain the concentration duration of students in the lecture time and generate analysis reports for each student and provide advice to teachers for following education improvement. This section illustrates the system architecture and data processing pipeline to present eaCamera in the general view.

The architecture of eaCamera follows the systematic architecture method of cloud edge combination (see Figure 1), in which the masked parts are the main modules of eaCamera.

The basic hardware of the edge side includes the surveillance camera responsible for recording video and the host hardware for video processing: CPU, memory, and network. The system at the edge side is a Linux operating system, which is configured with a video processor module for processing video into images, a CV module for processing images, and a behavior analysis module for analyzing students' attention concentration. At the same time, the edge side has a data sending module to communicate with the cloud, which is responsible for sending the time series data of students' attention concentration state to the cloud side.

The basic hardware of the cloud side is host hardware: CPU, network, and database module responsible for long-term storage and query of data. The system at the cloud side is also a Linux operating system, which is configured with a data receiving module to receive data of the edge side. At the same time, the module is also responsible for generating an attention analysis report from the data transmitted from the edge side or the data in the database for display by the front-end module. Users can flexibly operate the front-end module according to the working mode of eaCamera to view the attention analysis report they want to see.

There are two working modes in eaCamera, namely, active mode and passive mode. In the active mode, the edge side will scan the local storage. If an unprocessed video is found, the video will be sent to the video processor module, CV module, and behavior analysis module to generate concentration status data and sent to the cloud side to create a report in real time. The video will be preferentially saved on the local storage at the edge side in the passive mode. When the user calls the process command on the front-end page of the cloud, the required videos will be processed to generate concentration status data, which will be sent to the cloud to create a report. Active mode provides fast result access because the video is processed in real time and sent to the cloud side. But, it consumes more resources. Passive mode saves resources and money, but it cannot provide an online query service, and users have to wait until the report is generated.

In both modes, the pipeline of processing a video is the same (see Figure 2). The pipeline mainly includes a video processor, computer vision module, behavior analysis module, and front-end. The functionality and process procedure for each module is shown below.

(1) *Video Processor.* A long-term service at the edge side, responsible for streaming video into images. The video reader uses a parameter, sampler frequency, to control the fps of the output images stream to improve the processing speed. We take 12 fps as the default setting.

(2) *Computer Vision Module.* There are three functions within the CV module: face detector, id assigner, and landmark detector. The face detector draws a bound box for each student in all images generated by the video reader. Id assigner gives an id to each student in all images and maintains the assignments. In the

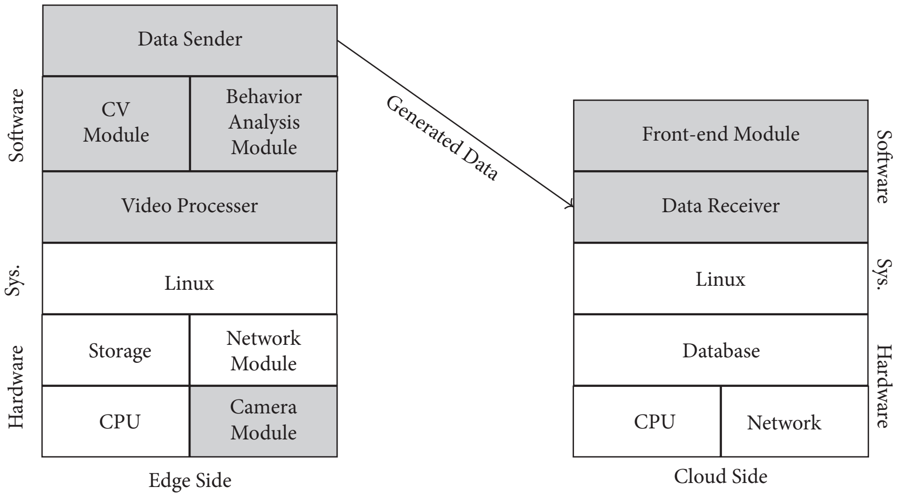


Figure 1: System architecture (Masked parts are main module of eaCamera).

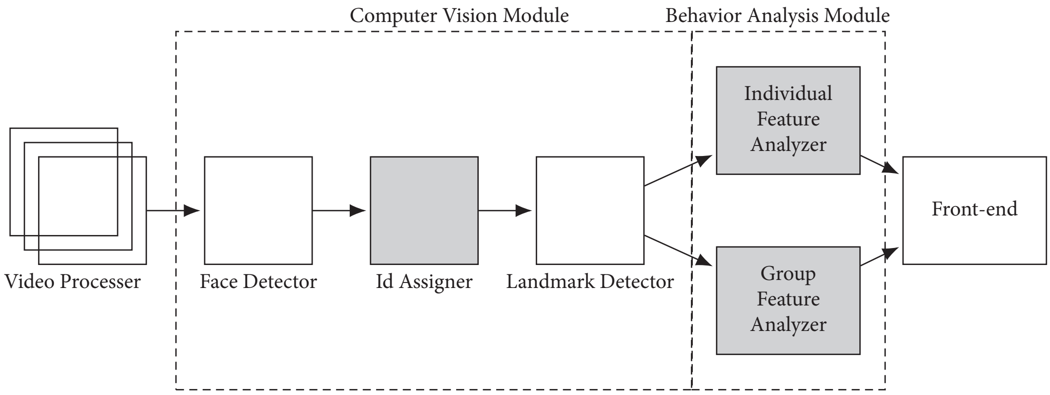


Figure 2: Data processing pipeline of eaCamera.

end, the landmark detector generates facial key points according to the bounded areas.

(3) *Behavior Analysis Module.* The responsibility of the behavior analysis module recognizes the attentional concentration behaviors. The module includes two components, namely, individual feature analysis and gourd feature analyzer, which extract attentional features from two perspectives. All features are combined later to detect the attentional concentration status of each student.

(4) *Front-End.* A web-based user application where the user gives a query about the classroom or student name to obtain the attentional concentration analysis reports. Besides, suppose the passive working mode is set. In that case, users can schedule analysis tasks according to their query conditions in the front-end, and the system notifies users when the reports are generated.

In Figure 2, the marked components Id Assigner, Individual Feature Analyzer, and Group Feature Analyzer are key modules of eaCamera. The novel models and algorithms are proposed in this paper to analyze student's attentional concentration status effectively.

4. Attentional Concentration Analysis

Attentional concentration analysis describes human concentration status at a certain time. Let us consider the concentration status sequence along with the time. We can explain the student concentration status during lecture time, evaluate their learning performance, and propose advice for improving teaching methods to teachers. This section focuses on the id assignment algorithm, behavior capture model, and attentional concentration analysis model used in eaCamera to build up the attentional concentration analysis system.

4.1. Id Assignment Algorithm. If we combine the functions of Face Detector and Id Assigner, it is an object tracking system [45], in which we try to track all the people in a video and assign a unique id to each of them. To be noticed, when we process a video, we process a sequence of images, and it is hard to keep the same id for the same person in all images, which is defined as an id switching problem. To conquer the id assignment task, two strategies can be used, face recognition and image similarity approach. In face recognition, the features of each person need to be collected in advance.

When the images are processed, we recognize who they are, and the id is related to their personal information, like name and gender. In the image similarity approaches, the chopped images from different video frames are compared. If the similarity of two images is higher than a user-defined threshold, we say two chopped images are assigned the same id. Deep neural networks can obtain the similarity and the features of the image. However, two approaches are unsuitable for eaCamera for the following reasons: (1) Face recognition needs to collect personal facial information, which causes privacy issues and needs a lot of labor; (2) Image similarity is time-consuming since the feature extraction neural network and the similarity comparison neural network need to be trained in advance and chopped images need to be labeled as well.

We apply a location-based id assignment strategy based on the assumption that students' seats are relatively fixed in the classroom. Figure 3 shows an example of 9 id assignments, and Algorithm 1 shows the id assignment algorithm. The id assignment process is shown as follows:

(1). For all the images from a video, the photo with a maximized number of bounding boxes is found (Line 1 in Algorithm 1).

(2) The bounding boxes are reshaped to the maximum size of the bounded box within the image (Line 2 in Algorithm 1), and the reshaped bounded boxes are denoted as standard boxes with unique ids. In Figure 3, the dashed rectangles are bounded boxes generated from the face detector, and the solid lines are standard boxes. The id of a standard box is set to (x, y), which is a pair of coordinates within the image.

(3) For the rest of the images, each bounded box is assigned with the id of the closest standard bounded box. The similarity is measured according to the intersection-over-union (IoU) function shown as equation (1), in which $\mathrm{A}_{\mathrm{box}}$ and $\mathrm{A}_{\mathrm{standard}}$ indicate the area of the bounded box and standard box, respectively.

$$\mathrm{IoU}\left(B_{\mathrm{Box}}, B_{\mathrm{standard}}\right) = \frac{\left(A_{\mathrm{box}} \cap A_{\mathrm{standard}}\right)}{\left(A_{\mathrm{box}} \cup A_{\mathrm{standard}}\right)}. \quad (1)$$

4.2. Behavior Feature Model. To evaluate students' attentional concentration status, eaCamera extracts the attentional features based on student's behavior. To be noticed, students sit down in the classroom most of the lecture time. Therefore, general behaviors like sitting, writing, and reading are unsuitable for attentional concentration analysis. Besides, general behaviors are continuous action, and the start and end times are estimated within a video by action localization techniques [46]. However, the abstracted behaviors might be transient, and action localization techniques might fail in this scenario.

EaCamera defines the student behavior features based on their emotions and head poses. Figure 4 shows an output example of the landmark detector. To be noticed, the example is a chopped facial image according to Id Assignment Algorithm. The landmark detector gives 68 key points for each facial image. We only consider six key points to describe behavior features. Six key points present the left eye, the right eye, the nose, the left cheek, the right cheek, and the chin, respectively, and the points are denoted as a set $\mathbf{P} = \{(x_{el}, y_{el}), (x_{\mathrm{er}}, y_{er}), (x_n, y_n), (x_{cl}, y_{cl}), (x_{cr}, y_{cr}), (x_c, y_c)\}$.The behavior features are extracted according to Algorithm 2.

In Algorithm 2, the features are given by a distance vector $\mathbf{V} = \{d_{el}, d_{er}, d_{cl}, d_{cr}, d_c\}$, where d_{el} presents the distance between the left eye point and the nose point. The distance is computed by Euclid distance shown as equation (2). We use distance-based features rather than vision-based features for the following reasons:

(1). It is efficient to obtain features for all facial images. There are more than 50 students in the classroom, and we need to collect behavior features for each of them. Therefore, the behavior feature model needs to be efficient and straightforward.

(2) The distance-based behavior features describe the head poses and emotions. The head pose is essential in the classroom to describe the concentration status since most of the time, students focus on the teacher and blackboard. Our eaCamera is installed in the front of the classroom. Therefore, the distances are symmetrical if the student is looking at the front. However, if the distances are unsymmetrical, we say the student is looking at the left or right sides.

(3) Since we consider the distances between the eyes and the nose and the distances between the cheeks and the nose, the behaviors like lowering one's head and raising one's head can be found.

$$d_t = \sqrt{\left(x_t - x_n\right)^2 + \left(y_t - y_n\right)^2}, \quad t \in \{el, er, cl, cr, c\}. \quad (2)$$

4.3. Attentional Concentration Analysis Model. The behavior feature model can find the head pose changes and the emotion changes. However, it is not sufficient to automatically analyze the attentional concentration status because we need to point out the concentration and abstracted behavior manually. Namely, we have to label the data. Therefore, we propose an attentional concentration analysis model to recognize the concentration-related behaviors. In eaCamera, the attentional concentration is described in two perspectives, namely, individual perspective and group perspective based on the following assumption:

4.3.1. Individual Perspective. Students' attentional concentration status changes during lecture time. However, primary students pay attention to the lecture content most of the time. Therefore, we try to observe the changes in students' behavior features, and the attentional concentration status can be captured from an individual perspective. Besides, primary students are always concentrated at the beginning of the lecture or the teacher's instruction. Figure 5 shows an example of individual attentional

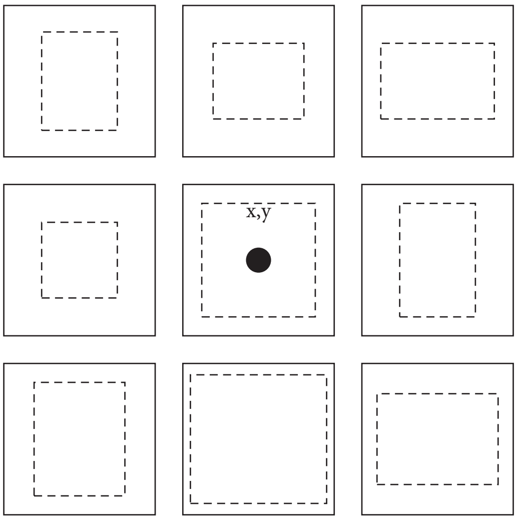


Figure 3: Id assignment strategy.

```
Input: $L_{\text{Box}}$: The list of bounded box lists. Each bounded box list includes a list of four coordinates ($L_{\text{Corrdinate}}$) for a video frame.
The bounded box is a list of coordinates.
Output: The list of id for all bounded box $L_{\text{ID}}$
frameIndex, maxNumberOfBox ← FindFrameWithMaxNumberBoxes ($L_{\text{Box}}$)
$L_{\text{StandardBoxes}}$ ← GenerateStandardBoxes ($L_{\text{Box}}$, frameIndex, maxNumberOfBox)
$L_{\text{ID}}$ ← {}
for $L_{\text{Corrdinate}}$ in $L_{\text{Box}}$ do
    idAssignment ← MatchStandardBox (StrandardBox, $L_{\text{Corrdinate}}$)
    Append ($L_{\text{ID}}$, idAssignment)
end for
return $E_n$;
```

Algorithm 1: Id Assignment Algorithm.

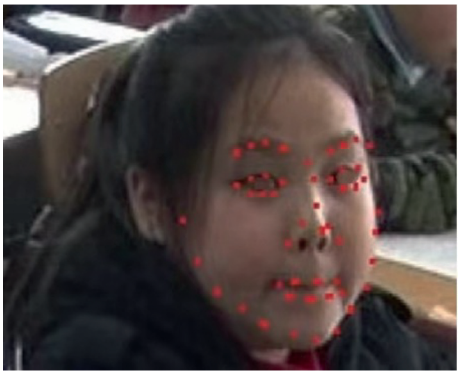


Figure 4: An output example of the landmark detector.

```
**Input:** P: The list of key points. Each key point includes a list of two coordinates
**Output:** V: A feature vector of the facial image
{$(x_{el}, y_{el}), (x_{er}, y_{er}), (x_n, y_n), (x_{cl}, y_{cl}), (x_{cr}, y_{cr}), (x_c, y_c)$} ← ExtractKeyPoints(P)
V ← {}
**for** $x, y$ in {$(x_{el}, y_{el}), (x_{er}, y_{er})(x_{cl}, y_{cl}), (x_{cr}, y_{cr}), (x_c, y_c)$} **do**
    d ← DistanceBetween $((x, y), (x_n, y_n))$
    Append (V, d)
**end for**
**return** V;
```

Algorithm 2: Behavior feature generation algorithm.

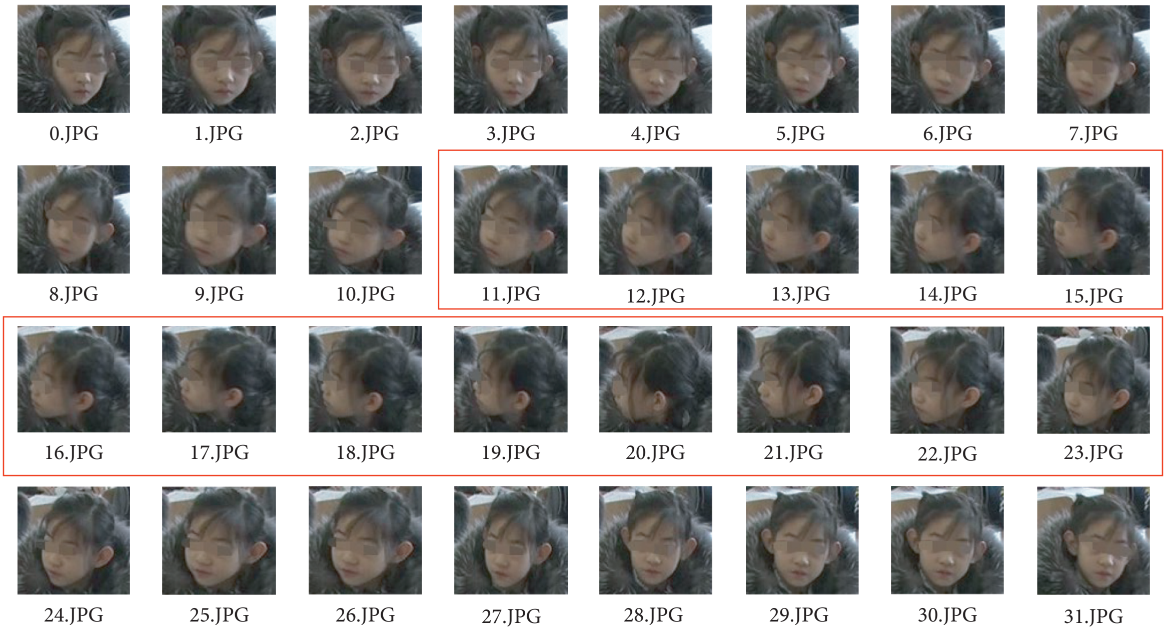


Figure 5: Example of individual attentional concentration changes.

concentration changes, in which the abstracted behavior is recognized by finding the outliers within behavior features in all frames.

4.3.2. Group Perspective. Teaching is a group activity. When most students pay attention to the lecture content, they have the same action pattern, which indicates students' attentional concentration status. Since most of the students focus on the lecture content in primary school, we use this character to recognize the outliers who are nonconcentrated. Figure 6 shows an example of group attentional concentration status comparison. In the figure, the teacher was demonstrating the lecture content. A few students did no't look at the teacher but focused on the desk or the textbook, which can be recognized as abstracted status since they are the outliers within all students.

According to the lecture content, the object, which students pay attention to, changes during the lecture time. The objective could be the teacher, the blackboard, the textbook, and so on. The individual perspective model only captures the behavior changes over time for a student, and it cannot capture the behavior changes caused by lecture content changes. Therefore, the group perspective model determines the attentional concentration status if the behavior feature change happens for a single student.

In the individual perspective model, the first step is to obtain the attentional concentration feature baseline. We assume that the primary student is concentrated at the beginning of the lecture or at the beginning of the teacher's instruction. Equation (3) shows a baseline feature matrix for a student, which consists of 1440 behavior feature vectors. The baseline feature matrix is extracted from the first 2 minutes of the class, which includes 120 seconds in total and 12 frames for each second. According to Chinese classroom habits, teachers will say something important in the first two minutes. We can think that the students' attention state is

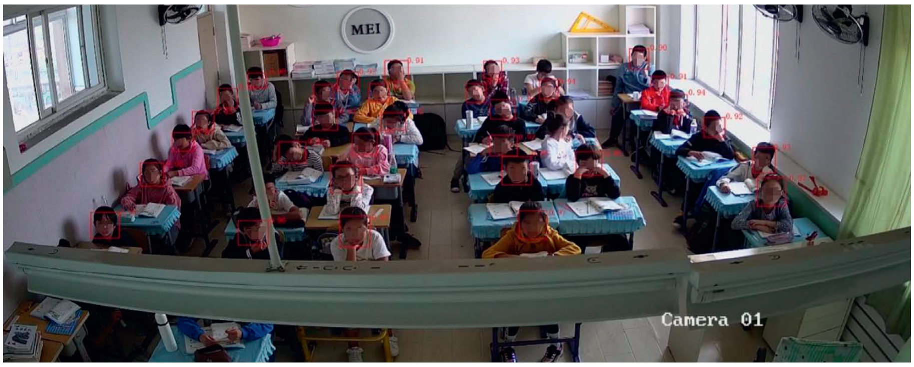


Figure 6: Example of group attentional concentration status comparison.

the most concentrated in the first 2 minutes of class, and it is the most accurate to compare the students' state in the first 2 minutes with the follow-up. At the same time, according to our calculation method, 1440 eigenvectors will be generated from the data in the first 2 minutes, and this amount of data has no calculation pressure on the edge machine.

$$\mathrm{M} = \begin{bmatrix} d_{el}^{1} & d_{er}^{1} & d_{cl}^{1} & d_{cr}^{1} & d_{c}^{1} \\ d_{el}^{2} & d_{er}^{2} & d_{cl}^{2} & d_{cr}^{2} & d_{c}^{2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{el}^{1440} & d_{er}^{1440} & d_{cl}^{1440} & d_{cr}^{1440} & d_{c}^{1440} \end{bmatrix}. \tag{3}$$

A baseline vector is obtained in the following steps: (1) The frequency diagram is done for each row within the feature matrix. (2) Choose the group of data with maximum frequency. (3) Compute the mean value of the group data as a part of the baseline. Figure 7 shows an example of frequency diagram for $[d_{el}^{1}, d_{el}^{2}, \ldots, d_{el}^{1440}]$. The class interval is set to 0.5. The majority distances are between 1.0 cm and 1.5 cm. Let us say the mean value is 1.2 cm. Then, we set $d_{el}^{*} = 1.2$ cm. Using the same strategy, we can obtain a standard baseline vector to present attentional concentration status of a student, denoted as $\mathbf{V}^{*} = \{d_{el}^{*}, d_{er}^{*}, d_{cl}^{*}, d_{cr}^{*}, d_{c}^{*}\}$.

In a short period, the teaching object remains the same. To evaluate the attentional concentration status, we only compare the current student behavior feature V with the attentional concentration baseline $\mathbf{V}^{*}$. A distance, Offset, between two features is obtained according to equation (4). When $\mathrm{Offset}(\mathbf{V}, \mathbf{V}^{*}) < \varepsilon$, we say the students are concentrated, in which ε is the pre-defined parameter present in concentration tolerance.

$$\mathrm{Offset}(V, V^{*}) = \sqrt{\sum_{t \in el,er,cl,cr,c} (d_t - d_t^{*})}. \tag{4}$$

The object that students pay attention to changes according to the teaching content. It causes attentional concentration baseline changes, which makes the individual perspective model fail. The group perspective model is introduced to solve this problem. In the individual perspective model, if the offset between the current behavior feature and the standard feature is higher than ε, the student might be in the abstracted status or still in concentration status with another attentional object. Therefore, we observe the others to decide whether the student is still concentrated.

In the individual perspective model, when an abstracted status is detected, namely, $\mathrm{Offset}(\mathbf{V}, \mathbf{V}^{*}) \geq \varepsilon$, then we use the group perspective model to detect the attentional object change. In the group perspective model, the evaluation of a student's attentional concentration status is done in the following steps:

(1). For all students, we compute the offset matrix G shown as equation (5), in which each row presents offsets for a student within 2 seconds (12 frames for each second).

(2) Use ε as a threshold to compute the binarization matrix of G, denoted as G_b.

(3) Summarize each column of G_b, denoted as group indicator, which indicates the number of students whose attentional concentration is changed.

(4) Since most of the primary students are concentrated in the class, if more than half of the students' attentional concentration status changes, we say the attentional object changes currently. In this case, the concentration standard baselines for all students are recomputed.

(5) Otherwise, we say the student is in abstracted status.

$$G = \begin{bmatrix} \mathrm{Offset}\left(V_1^{*}, V_1^{i}\right) & \mathrm{Offset}\left(V_1^{*}, V_1^{i+1}\right) & \cdots & \mathrm{Offset}\left(V_1^{*}, V_1^{i+24}\right) \\ \mathrm{Offset}\left(V_2^{*}, V_2^{i}\right) & \mathrm{Offset}\left(V_2^{*}, V_2^{i+1}\right) & \cdots & \mathrm{Offset}\left(V_2^{*}, V_2^{i+24}\right) \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{Offset}\left(V_n^{*}, V_n^{i}\right) & \mathrm{Offset}\left(V_n^{*}, V_n^{i+1}\right) & \cdots & \mathrm{Offset}\left(V_n^{*}, V_n^{i+24}\right) \end{bmatrix}. \tag{5}$$

5. Case Result

EaCamera is deployed in a primary school in Liaoning province, China. In the school, each classroom contains 40 students, and the lecture time is usually 40 minutes. In eaCamera, we choose state-of-art deep learning models to implement face detectors and landmark detectors. The face detector is implemented based on YoloV3 [47] and ArcFace [48], and the landmark detector is implemented based on DAN [44]. In this section, we first briefly introduce the

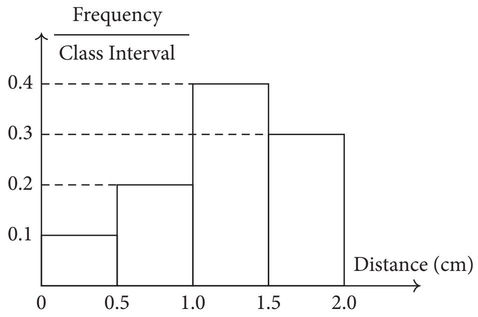


Figure 7: Example of frequency diagram.

structure of the network we use, and then we show the attentional concentration analysis result for each component.

As the calculation of eaCamera is mainly carried out at the edge side, we have selected several lightweight network implementations on the premise of ensuring accuracy.

(i) YoloV3: Yolo3 adopts a network structure called darknet-53 (including 53 convolution layers). It draws lessons from the practice of residual network, sets up fast links between some layers, forms a deeper network level and multi-scale detection, which improves the detection effect of mean Average Precision(mAP) and small objects [47].

(ii) ArcFace: ArcFace is a new loss function for face recognition based on additive angular margin loss. Its focus is to directly maximize the classification boundary in the angle space [48].

(iii) DAN: DAN is based on AlexNet network, which explores the adaptation relationship between source and target. As a representative method of deep transfer learning, it makes full use of the transferable characteristics of deep network, and then introduces the maximum mean discrepancy (MMD) in statistical learning, which has achieved good accuracy [44].

Figure 8 shows the result of the face detector. The red box is the bounded box for each student's face, and the number is the model's confidence for detecting facial objects. Figure 9 shows the result of the landmark detector and key point filter. To be noticed, after the face detector, the id assigner gives a unique id for each bounded box. However, the face detector might fail to generate a bounded box in some students. In this case, the id assigner ignores these ids, which indicates that the student is assigned with an abstraction label in this frame.

Figures 10 and 11 show examples of concentrated behaviors and abstracted behaviors. In Figure 10, student A follows the teacher's instruction. When the teacher required students to read the textbook, student A stared down her head and paid attention to it. When the teacher asked to look at the blackboard, student A was raising her head as well. In Figure 11, student B's behaviors are adverse. Therefore, we say she was abstracted.

In the front-end module, eaCamera generates statistic reports of the chosen lecture. To simplify the description, we only show the result of 18 students, whose id is assigned from numbers 1 to 18. To be noticed, a student with a lower id indicates that his position is more close to the blackboard and the teacher.

Figures 12 and 13 show the concentrated duration report and the corresponding statistic report. As can be seen from Figure 12, except for individual examples, the attention concentration time decreases as the ID increases. This shows that the closer the students are to the blackboard, the more focused they are in class, which is consistent with the reality. The closer the students are to the teacher, they may be more afraid to be distracted. In Figure 13, two-thirds of the students' attention concentration time is 30–40 minutes. The time of a class is 45 minutes, that is, most primary school students' attention is focused on the class time, which is consistent with our hypothesis.

EaCamera can give analysis reports according to user requirements or user-defined scripts. Users can generate corresponding reports for viewing according to their own needs. For example, users can view the analysis report in a certain day or even in a customized time period, the attention concentration analysis report of all people in a class, or the average attention duration of each class in different courses. Through different choices, users can carry out different target analysis. Figure 14 shows the compression of mean concentration duration between different lectures, namely, English, Chinese Literature, and Math. The result shows that students are likely pay more attention to Math considering the difficulty of the course and its importance. Using different analysis reports, teachers may change or improve their teaching approaches to get better performance.

6. Discussion

In this section, we compare eaCamera with other attention-related studies to analyze the advantages and disadvantages of eaCamera.

Traditionally, the attention concentration status of students is collected manually by human observers. With the development of science and technology, more and more attention detection methods have been proposed:

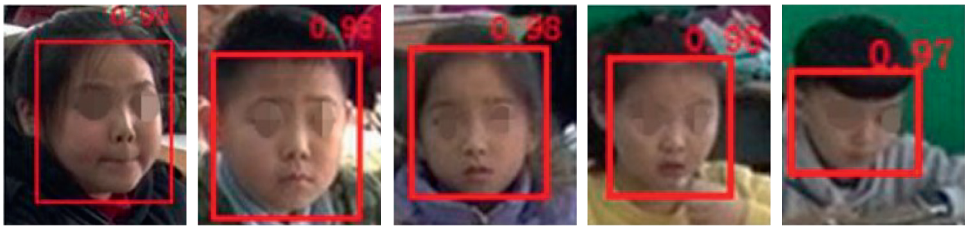


Figure 8: Result of face detector.

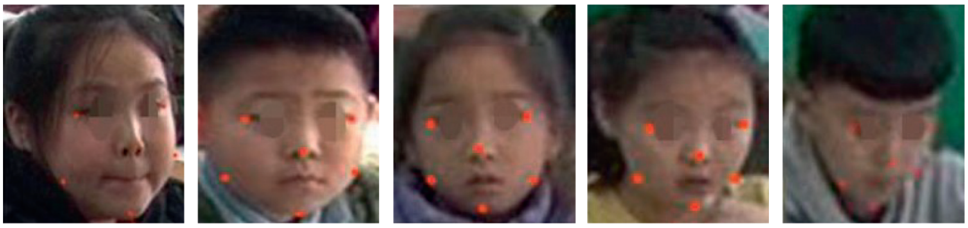


Figure 9: Result of key point filter.

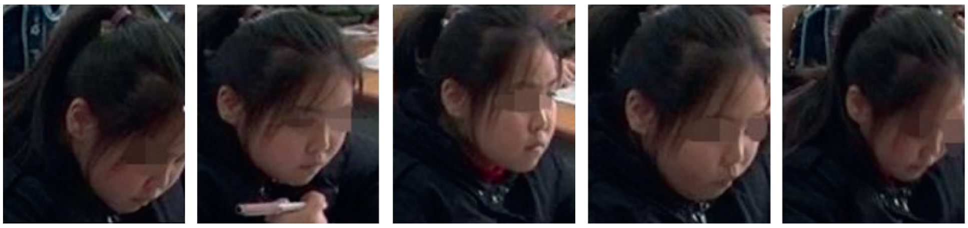


Figure 10: Example of student A concentrated status.

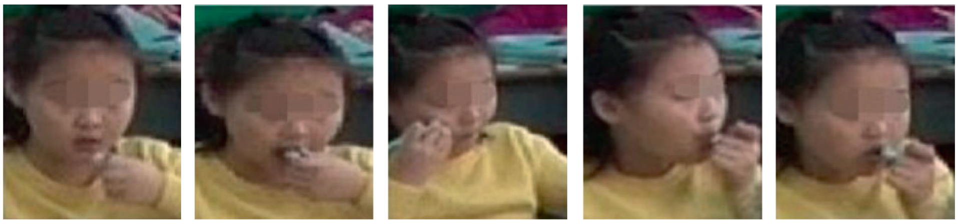


Figure 11: Example of student B abstracted status.

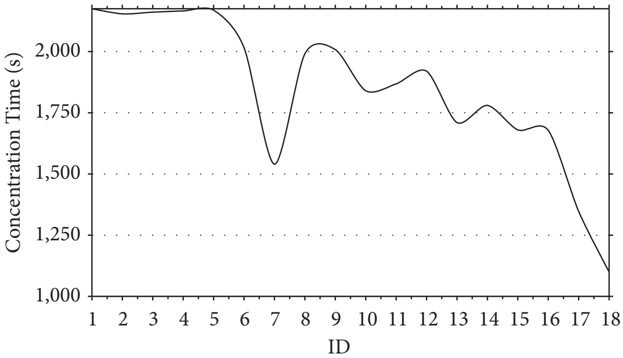


Figure 12: Concentrated duration report for all students.

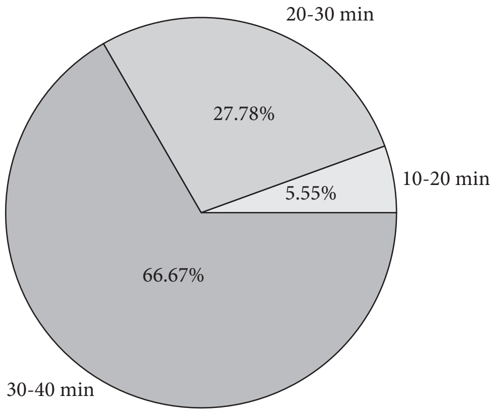


Figure 13: Concentration duration statistic report.

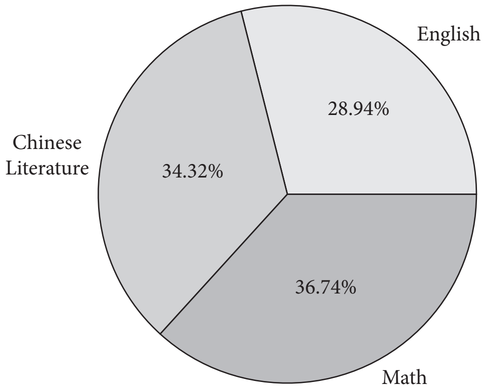


Figure 14: Mean concentration duration statistic report in aspects of different lectures.

Table 1: Comparison of eaCamera with other studies.

System	Function				Cost		Method	
	Attention analysis	Real-time warning	Generate report	Long-time storage	Expensive	Cheap	Tradition	New
Study 1	√					√	√	
Study 2	√	√			√			√
Study 3	√	√			√			√
eaCamera	√		√	√		√		√

(i) Study 1: Sujan Poudyal uses image processing technology to extract features from the student data captured from the monitoring system, and uses three data mining methods (SVM, decision tree, and KNN) to classify students' attention patterns [49].

(ii) Study 2: Xin Zhang proposed that wearable devices can be used to analyze students' attention state in class. The system integrates head movement module, pen movement module, and visual focus module to accurately analyze students' attention level in class [50].

(iii) Study 3: Shimeng Peng has developed an attention perception system (AAS) based on electroencephalography (EEG) signals to accurately identify students' attention level, which has high application potential and can provide online teachers with timely early warning of low attention level feedback in an e-learning environment [51].

We compared eaCamera with the above three studies in terms of function, cost, and method of use (see Table 1). Study 1 uses traditional methods such as data mining to analyze the data captured from the camera. Its function is relatively simple. It can only analyze the attention state of students, but it is not real time, and does not support the generation of reports and long-term storage of data; Study 2 and Study 3, respectively, use wearable devices and brain wave analysis devices to monitor students' attention state in class in real time, which can provide real-time alarm. The method is novel, but compared with cameras, the cost of their devices is very high. EaCamera adopts the method of edge calculation, the main functions are set at the edge side, and the hardware only needs cameras. It uses novel methods such as CV to realize the function of attention detection. At the same time, due to the cloud side, our research can realize the long-term storage of data, so as to facilitate the long-term comparative analysis and customize the analysis report. It can be seen that eaCamera uses novel methods to complete more comprehensive functions at a lower cost. However, eaCamera does not support the real-time warning function, which is also the further work of eaCamera.

7. Conclusion

We propose a novel edge intelligence system for attention analysis, eaCamera, in this case study, which is deployed in a primary school in China to evaluate students' attentional concentration status during lecture time. EaCamera gathers videos of lectures and utilizes deep learning models to extract behavior features of students. Later, the features are processed by an attentional concentration analysis model proposed to capture students' concentrated and abstracted behaviors automatically.

In eaCamera, we mainly focus on three modules. (1) Id assignment algorithm is used to assign a unique id to students within the same lecture. The assigned id keeps the same id for the same person within a video to indicate each student's identity. (2) Behavior feature model is adopted to generate the behavior feature vectors for each student in a video frame. A behavior feature is a vector that contains distances between facial key points. (3) Attentional concentration analysis model is proposed to capture students' attentional concentration status. Based on the above three modules, we implement an edge intelligent system for attention concentration analysis, which not only provides a certain reference value for the research of attention concentration detection but also promotes the application of science and technology in the educational environment. At the same time, the system also makes a certain contribution to the research of edge intelligent complex system.

EaCamera can be used in enclosed multi-people spaces to analyze users' attentional states according to two perspectives, namely, individual perspective and group perspective. Individual perspective captures the behavior feature changes over time, and group perspective captures the behavior feature outlier within a group of people. EaCamera relies on the assumption that the majority of people within a group are concentrated. Therefore, eaCamera cannot work within some environments that do not have the above assumption. Besides, eaCamera replies on computer vision technique, namely, its anti-interference ability is limited. If the faces are blocked, then eaCemear would not be able to produce the correct analyses. In further work, we want to expand the usage scenario of eaCamera, for example, diver notification system, factory monitoring system, office monitoring system, driver monitoring system, and hospital operation monitoring system. In order to improve the existing shortcomings of eaCamera, we can expand the recognition scope of eaCamera in the future, recognize the state of attentional concentration according to human behavior, and make a certain contribution to pedestrian behavior recognition and other scenarios in automatic driving. In the current implementation, the facial-based behavior feature model is adopted considering lecture and primary school characters. A novel behavior feature model is needed to capture the behavior features in a different scenario in further work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Moran, "Concentration: attention and performance," *The Oxford Handbook of Sport and Performance Psychology*, Oxford University Press, Oxford, UK, pp. 117–130, 2012.

[2] Y. Wang, Y. Shi, J. Du, Y. Lin, and Q. Wang, "A CNN-based personalized system for attention detection in wayfinding tasks," *Advanced Engineering Informatics*, vol. 46, Article ID 101180, 2020.

[3] Z. Nan, T. Shu, R. Gong et al., "Learning to infer human attention in daily activities," *Pattern Recognition*, vol. 103, Article ID 107314, 2020.

[4] B. J. Scholl, "Objects and attention: the state of the art," *Cognition*, vol. 80, no. 1-2, pp. 1–46, 2001.

[5] A. Seemann, *New Developments in Psychology, Philosophy of Mind, and Social Neuroscience*, MIT Press, Cambridge, MA, USA, 2011.

[6] T. Liu, Z. Yuan, J. Sun et al., "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 353–67, 2011.

[7] J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, and X. Zhu, "Deep learning towards mobile applications," in *Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems*, pp. 1385–1393, IEEE, Vienna, Austria, July 2018.

[8] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-Aware video object segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 20–33, 2018.

[9] A. Borji, M. N. Ahmadabadi, B. N. Araabi, and M. Hamidi, "Online learning of task-driven object-based visual attention control," *Image and Vision Computing*, vol. 28, no. 7, pp. 1130–1145, 2010.

[10] W. Wang, J. Shen, J. Xie, M. M. Cheng, H. Ling, and A. Borji, "Revisiting video saliency prediction in the deep learning era," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 220–237, 2021.

[11] K. Han, Y. Li, and B. Xia, "A cascade model-aware generative adversarial example detection method," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 800–812, 2021.

[12] T. Qian, "Emotion recognition system based on distributed edge computing," *Computer Science*, vol. 48, pp. 638–643, 2021.

[13] K. Zhang and A. B. Aslan, "AI technologies for education: recent research & future directions," *Computers and Education: Artificial Intelligence*, vol. 2, Article ID 100025, 2021.

[14] A. Agarwal, S. Sharma, V. Kumar, and M. Kaur, "Effect of E-learning on public health and environment during COVID-19 lockdown," *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 104–115, 2021.

[15] Y. Zhang, P. Qu, and W. Zheng, "Towards "general purpose" brain-inspired computing system," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 664–673, 2021.

[16] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 195–202, 2020.

[17] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, pp. 155–172, 2021.

[18] L. Yuan, Q. He, S. Tan et al., "CoopEdge: a decentralized blockchain-based platform for cooperative edge computing," in *Proceedings of the Web Conference 2021*, pp. 2245–2257, ACM, Ljubljana Slovenia, April 2021.

[19] J. Luo, J. Li, L. Jiao, and J. Cai, "On the effective parallelization and near-optimal deployment of service function chains," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1238–1255, 2021.

[20] J. Cai, Z. Huang, L. Liao, J. Luo, and W.-X. Liu, "APPM: adaptive parallel processing mechanism for service function chains," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1540–1555, 2021.

[21] X. Zheng, M. Li, and J. Guo, "Task scheduling using edge computing system in smart city," *International Journal of Communication Systems*, vol. 34, no. 6, 2021.

[22] X. Huang, C. Shen, X. Boix, and Q. Zhao, "SALICON: reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proceedings of the International Conference on Computer Vision*, pp. 1–9, IEEE, Santiago, Chile, December 2015.

[23] B. Du, Y. Wang, C. Wu, and L. Zhang, "Unsupervised scene change detection via latent dirichlet allocation and multivariate alteration detection," *IEEE Journal of Selected Topics in*

Applied Earth Observations and Remote Sensing, vol. 11, no. 12, pp. 4676–4689, 2018.

[24] W. Xiong, L. Zhang, B. Du, and D. Tao, “Combining local and global_ Rich and robust feature pooling for visual recognition,” *Pattern Recognition*, vol. 62, pp. 225–235, 2017.

[25] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: delving deep into convolutional nets,” in *British Machine Vision Conference*, M. F. Valstar, A. P. French, and T. P. Pridmore, Eds., BMVA Press, University of Nottingham, Nottingham, UK, pp. 1–11, 2014.

[26] M. Kümmerer, L. Theis, and M. Bethge, “Deep gaze I: boosting saliency prediction with feature maps trained on ImageNet,” in *Proceedings of the International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., pp. 1–12, San Diego, CA, USA, May 2015.

[27] W. Wang and J. Shen, “Deep visual attention prediction,” *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2368–2378, 2018.

[28] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: an in-depth survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2021.

[29] C. Baoyuan, L. Yitong, and S. Kun, “Research on object detection method based on FF-yolo for complex scenes,” *IEEE Access*, vol. 9, pp. 127950–127960, 2021.

[30] H. Park, S. Lee, M. Lee, M.-S. Chang, and H.-W. Kwak, “Using eye movement data to infer human behavioral intentions,” *Computers in Human Behavior*, vol. 63, pp. 796–804, 2016.

[31] P. Wei, D. Xie, N. Zheng, and S.-C. Zhu, “Inferring human attention by learning latent intentions,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1297–1303, Melbourne, Australia, August 2017.

[32] X. Zhang, N. Mlynaryk, S. Japee, and L. G. Ungerleider, “Attentional selection of multiple objects in the human visual system,” *NeuroImage*, vol. 163, pp. 231–243, 2017.

[33] Y. Xu, X. Hong, Q. He, G. Zhao, and M. Pietikäinen, “A task-driven eye tracking dataset for visual attention analysis,” in *Advanced Concepts for Intelligent Vision Systems*, S. Battiato, J. Blanc-Talon, G. Gallo, W. Philips, D. Popescu, and P. Scheunders, Eds., Springer International Publishing, Berlin, Germany, pp. 637–648, 2015.

[34] P. Mundy, “A review of joint attention and social-cognitive brain systems in typical development and autism spectrum disorder,” *European Journal of Neuroscience*, vol. 47, no. 6, pp. 497–514, 2018.

[35] W. Lawson, A. M. Harrison, E. S. Vorm, and J. G. Trafton, “Joint attention estimator,” in *Proceedings of the Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 331–333, ACM, Cambridge UK, March 2020.

[36] C.-M. Huang and A. L. Thomaz, “Effects of responding to, initiating and ensuring joint attention in human-robot interaction,” in *Proceedings of the 2011 RO-MAN*, pp. 65–71, IEEE, Atlanta, GA, USA, July 2011.

[37] Z. Yang, Y. Wang, X. Chen et al., “CARS: continuous evolution for efficient neural architecture search,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1826–1835, IEEE, Seattle, WA, USA, June 2020.

[38] L. Liu, W. Ouyang, X. Wang et al., “Deep learning for generic object detection: a survey,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[39] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, IEEE Computer Society, Santiago, Chile, December 2015.

[40] J. Redmon and A. Farhadi, *YOLOv3: An Incremental Improvement*, https://arxiv.org/abs/1804.02767, 2018.

[41] X. Li, H. Zhao, and L. Zhang, *Recurrent RetinaNet: A Video Object Detection Model Based on Focal Loss*, L. Cheng, A. C. S. Leung, and S. Ozawa, Eds., , Springer International Publishing, Berlin, Germany, 2018pp. 499–508, Neural Information Processing.

[42] H. Law and J. Deng, “CornerNet: detecting objects as paired keypoints,” *International Journal of Computer Vision*, vol. 128, no. 3, pp. 642–656, 2020.

[43] M. Bodini, “A review of facial landmark extraction in 2D images and videos using deep learning,” *Big Data and Cognitive Computing*, vol. 3, no. 1, p. 14, 2019.

[44] M. Kowalski, J. Naruniec, and T. Trzcinski, “Deep alignment network: a convolutional neural network for robust face alignment,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2034–2043, IEEE, Honolulu, HI, USA, July 2017.

[45] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, “Multiple object tracking: a literature review,” *Artificial Intelligence*, vol. 293, p. 103448, 2021.

[46] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao, “3C-Net: Category count and center loss for weakly-supervised action localization,” in *Proceedings of the International Conference on Computer Vision*, pp. 8678–8686, OpenReview.net, Seoul, Republic of Korea, October 2019.

[47] J. Qi, C. Wang, L. Cheng, S. Jiang, X. Zhang, and H. Jing, “YOLOFKP: dense face detection based on YOLOv3 key point network,” in *Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition*, pp. 187–191, ACM, Xiamen China, October 2020.

[48] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, IEEE, Long Beach, CA, USA, June 2019.

[49] S. Poudyal, M. J. Mohammadi-Aragh, and J. E. Ball, “Data mining approach for determining student attention pattern,” in *Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE)*, pp. 1–8, IEEE, Uppsala, Sweden, October 2020.

[50] X. Zhang, C.-W. Wu, P. Fournier-Viger, L.-D. Van, and Y.-C. Tseng, “Analyzing students’ attention in class using wearable devices,” in *Proceedings of the 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–9, IEEE, Macau, China, June 2017.

[51] C.-M. Chen, J.-Y. Wang, and C.-M. Yu, “Assessing the attention levels of students by using a novel attention aware system based on brainwave signals: novel attention aware system based on brainwave signals,” *British Journal of Educational Technology*, vol. 48, no. 2, pp. 348–369, 2017.

Hindawi
Complexity
Volume 2021, Article ID 8554351, 16 pages
https://doi.org/10.1155/2021/8554351

Research Article

A Reputation Value-Based Task-Sharing Strategy in Opportunistic Complex Social Networks

Jia Wu,[1,2] Fangfang Gou,[1,2] Wangping Xiong,[1] and Xian Zhou[1]

[1]*School of Computer, Jiangxi University of Chinese Medicine, NanChang 330004, JiangXi, China*
[2]*School of Computer Science and Engineering, Central South University, Changsha 410083, China*

Correspondence should be addressed to Wangping Xiong; 20030730@jxutcm.edu.cn and Xian Zhou; 20030731@jxutcm.edu.cn

Received 10 October 2021; Accepted 5 November 2021; Published 26 November 2021

Academic Editor: Xuyun Zhang

As the Internet of Things (IoT) smart mobile devices explode in complex opportunistic social networks, the amount of data in complex networks is increasing. Large amounts of data cause high latency, high energy consumption, and low-reliability issues when dealing with computationally intensive and latency-sensitive emerging mobile applications. Therefore, we propose a task-sharing strategy that comprehensively considers delay, energy consumption, and terminal reputation value (DERV) for this context. The model consists of a task-sharing decision model that integrates latency and energy consumption, and a reputation value-based model for the allocation of the computational resource game. The two submodels apply an improved particle swarm algorithm and a Lagrange multiplier, respectively. Mobile nodes in the complex social network are given the opportunity to make decisions so that they can choose to share computationally intensive, latency-sensitive computing tasks to base stations with greater computing power in the same network. At the same time, to prevent malicious competition from end nodes, the base station decides the allocation of computing resources based on a database of reputation values provided by a trusted authority. The simulation results show that the proposed strategy can meet the service requirements of low delay, low power consumption, and high reliability for emerging intelligent applications. It effectively realizes the overall optimized allocation of computation sharing resources and promotes the stable transmission of massive data in complex networks.

1. Introduction

In recent years, with the deep integration and development of IoT technologies and industries, various revolutionary mobile devices have penetrated into infrastructure, life services, national defense, and military, giving rise to new IoT smart applications such as smart home, driverless, augmented reality/virtual reality (AR/VR), and face recognition [1]. These applications generate large amounts of data, and at the same time, they are computationally intensive and time-sensitive. In particular, some new applications based on big data and artificial intelligence have high requirements for low-latency transmission of large-capacity data, making mobile devices face huge challenges in terms of computing resources and computing capabilities.

In complex opportunistic social networks, both mobile smart terminal devices and servers can be considered as social nodes. These social nodes can communicate and share computing and storage resources [2–4]. However, frequent communication between these nodes leads to the surge in traffic and high multivariate of data-type, which poses a challenge to network management. When processing computation-intensive and latency-sensitive services, mobile terminals can share the computation task to servers with greater computing power at base stations, instead of performing computation by themselves. Then, the servers can provide the communication, storage, and computation resources needed to process these functions and services through networks. Meanwhile, through the management and mining of the hidden knowledge in the massive data, the end-users can obtain high-quality, low-latency, low-consumption, and highly personalized services [5, 6].

However, in the process of sharing a large number of computing tasks from the terminal device to the server, how to allocate the server's computing resources to ensure the service performance of smart mobile terminals is an urgent problem to solve [7]. In addition, wireless sensor networks have some inherent characteristics, such as limited node energy, storage space, and computational processing capabilities. These characteristics can record nodes movement, so the bad behavior nodes [8, 9] are prone to exist in the network. Bad behavior nodes are further divided into selfish behavior nodes and malicious attack behavior nodes. The selfish behavior node uses the network resources as much as possible, and it does not make any/makes very little contribution to the network. The sole purpose of malicious nodes is to attack other nodes in the network or the entire peer-to-peer network. Malicious nodes forge false hotspot resources and provide malicious resources to other nodes in the network to download or forward them to achieve the purpose of invading, controlling nodes, or even destroying the entire network. Malicious nodes are the direct source of security problems in peer-to-peer networks. Mass data have higher requirements for data transmission on complex networks. How to exclude malicious nodes from a large number of nodes is a big challenge.

These problems can be solved in two steps: firstly, to solve the decision problem of whether and when to share the tasks under latency and energy consumption demand. Secondly, to solve the problem of reliable sharing allocation arising from malicious competition for computing resources. By using the cooperation between nodes, the reputation value of each node is updated and the malicious node is finally derived. Therefore, we try to allocate server computing resources based on the reputation value of smart mobile terminals to prevent malicious behaviors and realize the optimal allocation of overall resources from the perspective of the benefits of a larger smart mobile terminal, not just from the perspective of the economic benefits of a larger service provider.

To solve the above problems, this paper proposes an advanced task-sharing model (DERV), which consists of a task-sharing decision model that takes into account the delay and energy consumption requirements of new IoT applications, and a resource game allocation model that allocates server computing resources based on reputation values. Among them, the shared decision model adopts an improved PSO algorithm to realize multi-task sharing among multiple mobile terminals, which meets the low latency and low energy consumption requirements of new applications of the Internet of Things. The resource allocation model aims to prevent unreasonable resource allocation caused by malicious terminal competition, and it is mainly realized by the Lagrangian multiplier method. In this way, computing resources are allocated reasonably and the overall utility is maximized. The DERV model realizes the stable transmission of massive data in complex social networks.

The contributions to this paper are listed as follows:

(1) We focus on providing low-latency, low-energy, and highly reliable service quality guarantees for time-delay and energy-sensitive, computing-intensive smart mobile terminals in the big data environment. We first propose a network model composed of users, MEC servers, and trusted authorities. Then, we formulate the optimal configuration of the overall resources as how to allocate the computing resources of the MEC server. Finally, we transform the optimization problem as offloading decision and offloading allocation subproblems.

(2) For offloading decision issues, this paper proposes a task-sharing decision model for multiple smart mobile terminals in a complex opportunistic social network environment. It considers time delay and energy consumption comprehensively. The purpose is to achieve an optimal task-sharing solution with low latency and low energy consumption.

(3) For offloading allocation issue, a dynamic task-sharing allocation game based on the reputation value of smart mobile terminals is proposed, and it is achieved by the Lagrange multiplier method.

The rest of the paper is arranged as follows: The second part introduces the related research of data transmission routing algorithms in opportunistic social networks, the third part introduces related theoretical concepts and algorithm models, and the fourth part verifies the performance of the model on various standards through simulation experiments. At the end of the paper, we discussed and summarized the full text.

2. Related Work

Since the task-sharing strategy promises to solve the performance bottlenecks faced by mobile smart terminals in opportunistic complex social networks when dealing with novel smart applications such as computation-intensive and latency-sensitive applications, it has received widespread attention from scholars and has gradually become a research hotspot. Therefore, in this section, we will give a brief introduction to the task-sharing study of whether and how much mobile smart terminals share computational tasks and how to allocate the macro base station computational resources.

2.1. Task-Sharing Model. In terms of task sharing, there are a large number of researches' results on whether mobile smart terminals share tasks to macro base stations. Task-sharing decision schemes often use delay and energy consumption as benchmarks. Different environments and systems have different requirements for task-sharing decisions, and some applications require a good balance between latency and energy consumption. Paper [10] presents the Lyapunov optimization-based dynamic computation offloading algorithm (LODCO), a dynamic computational task-sharing algorithm based on Lyapunov's optimization theory. It optimizes task-sharing decisions in terms of both task running delay and task running failure, minimizing task processing delay, and ensuring the success rate of the data transfer process, but ignoring influencing factors such as energy consumption and cost. Paper [11] adopts an artificial fish swarm algorithm to design a task-sharing strategy for energy consumption optimization under delay constraints. The strategy takes full account of the link conditions in the task data transmission network and effectively reduces the energy consumption of the device, but the disadvantage is that the algorithm is too complex. The paper [12] designs an energy-efficient computational task-sharing scheme, which is based on deep learning, to solve the problem of selective sharing of mobile application components. Experimental results show that the solution has high accuracy and low energy consumption, but the computationally intensive task remote interaction brings about high latency problems. The influential factors considered in the above paper in the task-sharing decision modeling process are not comprehensive enough, which affects the rational execution of task-sharing decisions.

The paper [13] considers a multi-user MEC system in which a single MEC server can handle the computational tasks shared by multiple user devices over wireless channels. This solution has a significant reduction in the sum of delay and energy consumption. Paper [14] presents a Lyapunov optimization theory combined with an adaptive e-learning approach to the problem of optimal sharing of trade-offs between response latency and energy costs in IoT scenarios. Paper [15] optimizes multi-user mobile edge computing task-sharing systems, constructs Markov decision problems with the long-run average overhead of delay and power consumption as optimization goals, and solves them using convex optimization theory. Although the above paper treats the delay and energy consumption as important components in the computational task-sharing process, the method of reducing energy consumption by constraining the delay makes the task-sharing strategy lack generality. We noticed that there are many excellent energy-saving strategies in the Internet of Things (IoT) paradigm. The energy consumption models proposed by scholars have all been proved to be effective in important IoT applications such as organizational collaboration, staff track, and logistics positioning. Many strategies emphasize the realization of potential benefits in terms of energy and cost, and have been implemented on real test beds [16–18]. Therefore, based on summarizing previous studies, this paper introduces delay and energy demand coefficients to consider more comprehensively the delay and energy consumption of intelligent terminal node computing task sharing.

2.2. Resource Allocation Model. In terms of resource allocation, the current research work focuses on the design of an allocation strategy for the problem of how to allocate computing resources for macro base stations. Paper [19] enabled MEC's LTE-V network using a deep Q-Learning approach and proposed an optimization goal of maximizing the utility of the sharing system under a given delay constraint. The results show that the scheme can share vehicle tasks with optimal utility while also satisfying the reliability and wait time constraints but ignores the important resource allocation aspect of the task-sharing problem. Huang et al. [20] proposed a computing task plan based on mobile user security and cost awareness in a mobile edge computing environment. Its goal is to minimize the total cost under the constraint of risk probability and provide security and cost efficiency for mobile users. However, it ignores the actual needs of the end nodes, which is not conducive to the overall optimization of resource allocation. The paper [21] studies the sharing strategies for computationally intensive tasks (data processing tasks and blockchain mining tasks) in blockchain scenarios, addressing the failure of traditional task-sharing strategies (e.g., auction and game theory strategies) to adjust the sharing strategies to changes in the environment, but the efficiency and reliability of the average resource allocation scheme are difficult to meet the quality of service (QoS) needs of users.

The above paper has achieved some results in the study of resource allocation in opportunistic complex networks, but it ignores the problem of smart mobile terminal nodes competing for resources, which makes it difficult to achieve a reasonable and reliable allocation of computing resources in opportunistic social IoT systems. If the malicious resource competition behavior of terminal nodes is regulated and the overall optimal allocation of computational task-sharing resources is achieved, low latency, low energy consumption, and highly reliable QoS guarantees can be provided for experimental and energy-sensitive terminal nodes [22]. We have noticed that trust models of wireless sensor networks (WSNs) security have flourished due to the day-to-day attack challenges, which are most popular for the Internet of things [23]. Many strategies have introduced a reputation mechanism to solve this problem, and achieved good results [24, 25]. For example, Han et al. [24] supplemented the fence-sitter group to the existing rumor dissemination model, and then proposed a novel SIFR rumor dissemination model, which effectively realizes the security monitoring of rumors dissemination in the network. Therefore, this paper innovatively introduces smart mobile terminal node reputation values to allocate computational resources in macro base station servers to effectively achieve the optimal allocation of computational resources [26].

To summarize, this paper builds a model (DERV) to address the shortcomings of task-sharing decision-making and resource allocation methods in complex social networks of things. The model takes into account the time delay and energy consumption of computational tasks and allocates computational resources based on the reputation value, which effectively achieves overall optimization, and rational and reliable allocation of computational resources.

3. System Model Design

3.1. Network Model. The MEC server allocates computing resources to perform the different computational task-sharing processes for users in the environment of opportunistic social networks of things based on different computational task requirements and reputation values of smart terminals [27]. The mobile edge intelligent computing network model based on reputation value consists of users holding smart mobile terminals (VR/AR, smart cars, video game consoles, PC monitors, drones, smart home, etc.), MEC servers, and trusted institutions. These devices generate massive amounts of data, posing a challenge to network management. The network model is shown in Figure 1. And, a list of all acronyms used in the paper is shown in Table 1.

3.1.1. Users. The users who need to perform computation-intensive computing tasks in this paper have the computation, GPS, and wireless communication modules. The computing module performs computational tasks, the GPS module acquires location information in real-time, and the wireless communication module enables data transmission. When a user needs to run emerging applications such as augmented reality, image processing, etc., and it is difficult for the computation module to complete the corresponding computation and storage tasks, the computation task-sharing request is initiated by the wireless communication module and the computation task is sent to the specified MEC server.

3.1.2. MEC Servers. Distributed MEC servers are deployed in the vicinity of users and are responsible for allocating computation resources to perform different computation shares and returning data to users after completing computation tasks. Users have different reputation requirements for computing task sharing. In the case of partial sharing, when more computing tasks are shared, a small base station is set up to store the queued waiting tasks, and the stored tasks are uploaded to the macro base station when the macro base station is idle. In this paper, we use the full sharing decision, so we do not consider the case of deploying a small base station, but only the macro base station of the MEC server, and each user chooses a different task-sharing method according to different requirements.

3.1.3. Trusted Authority. A trusted authority is responsible for recording events and updating the reputation value database. All users must register their legal identity with the trustworthy authority, and users with legal identity have valid reputation value and obtain the public and private key pairs and certificates required for secure communication. The trusted authority updates the reputation value in real-time through the user's behavior records at different times and provides the service provider with access to the whole network reputation value database.

3.2. Mathematical Model. The nodal users connected to the complex social networks of things have different computational task-sharing needs, and in addition to performing local computation, they can also share the computational task to the macro base station where the MEC server is deployed [30]. In a multi-user participation scenario, whether to choose to share and how to allocate resources effectively after task sharing are issues that need to be addressed [31]. To solve these problems, this paper proposes an intelligent computing task-sharing method that jointly considers delay and energy consumption. We select an appropriate task-sharing strategy according to the performance benchmarks of different smart mobile terminals held by users and the needs of smart mobile terminals to realize low-latency, low-energy-consumption computing task-sharing decision-making. In addition, to achieve the overall optimal allocation of computing resources, this paper proposes a reliable computing resource allocation model to realize a bargaining game based on user reputation values and maximize user benefits [32].

3.2.1. Shared Decision-Making Models. In this paper, a single cellular network model with multiple users and the deployment of MEC servers for macro base stations is constructed. Assuming $M = \{1, 2, \ldots, m\}$ denotes m users holding different kinds of smart mobile terminals. Each user has one computation-intensive or latency-sensitive computation task. The computational task owned by user i is denoted as

$$U_i = \{A_i, B_i, T_i^{\max}\}, \quad i \in M. \tag{1}$$

In this equation, U_i indicates the computational task owned by user i. A_i indicates the data size of the computational task. B_i indicates the number of CPU cycles required to complete the computational task, and $T_i^{\max}$ indicates the maximum latency that user i can tolerate completing the computational task.

According to the different needs of smart mobile terminal users, delay-sensitive smart mobile terminal users have a higher demand for time delay, while energy-sensitive smart mobile terminal users have a higher demand for energy-saving due to their power shortage, so a trade-off mechanism is introduced, and the time demand coefficient ∂_i^t and energy demand coefficient ∂_i^e indicate the bias degree of the user's demand for delay sensitivity and energy-saving, respectively. The computational tasks for each user can either be shared with the MEC server or performed locally. This paper introduces the decision mechanism $d_{i,j} = \{0, 1\}$, where $j = \{1, 2\}$ indicates the decision mode, $j = 1$ indicates that the user chooses the local computational model to perform the computational task, and $j = 2$ indicates that the user chooses to share the computational model to the macro base station to perform the computational task. In this paper, we construct a task-sharing decision model with the goal of global system consumption involving multiple smart mobile devices, as shown in equation (2).

$$\min \sum_{i=1}^{m} \left\{ \partial_i^t \left(d_{i,1} F_1 + d_{i,2} F_2 \right) + \partial_i^e \left(d_{i,1} G_1 + d_{i,2} G_2 \right) \right\}. \tag{2}$$

In this equation, F_1 and F_2 denote the total latency of local and shared to macro base station calculations, respectively, G_1 and G_2 denote the total energy consumption of local and shared to macro base station calculations, respectively. The task-sharing decision process is shown in Figure 2.

(1) Local Computational Model. The local computing power of different smart mobile devices held by users varies, and the computational delay $T_{i(z)}^{P}$ and computational energy

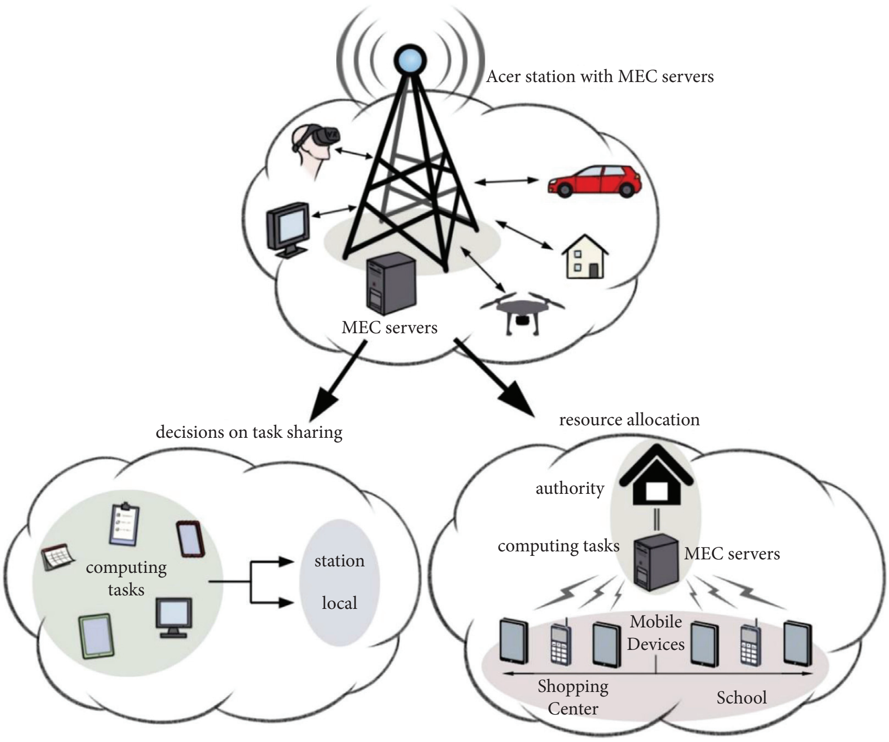


Figure 1: This diagram depicts the network model on which this paper is based.

Table 1: List of all acronyms used in the paper.

Acronym	Meaning
AR	Augmented reality
VR	Virtual reality
MILECR	Message importance based low energy consumption routing algorithm [28]
FCNS	Fuzzy routing-forwarding algorithm exploiting comprehensive node similarity [29]
DERV	The model we proposed in this paper
LODCO	The Lyapunov optimization-based dynamic computation offloading algorithm

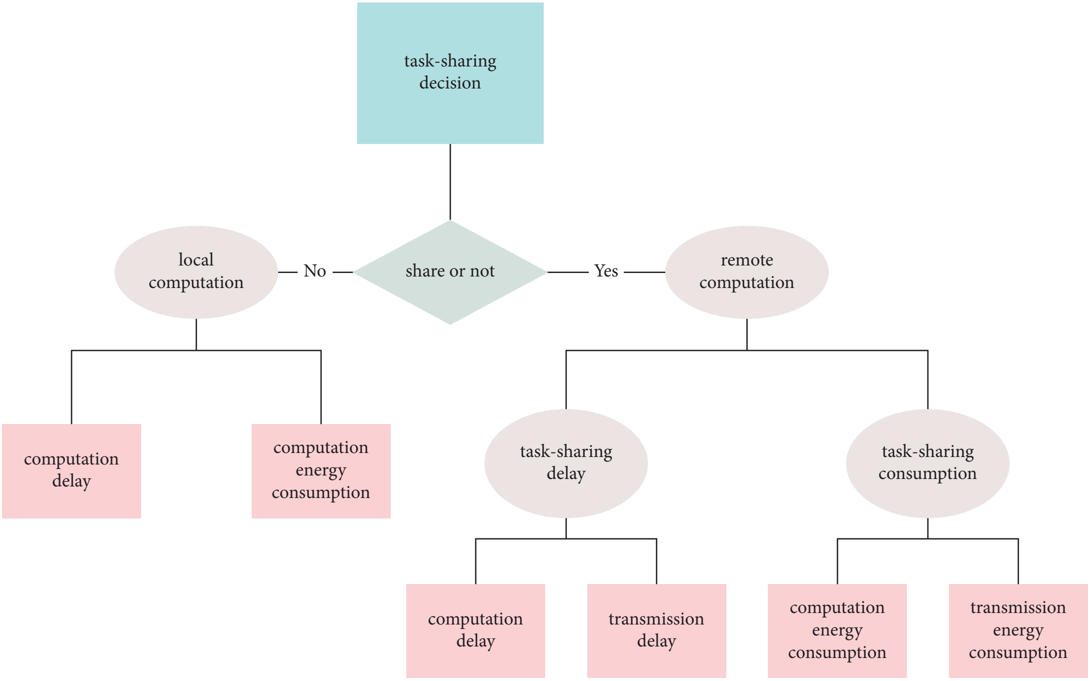


Figure 2: This diagram depicts the task-sharing decision-making process.

consumption $N^P_{i(z)}$ of the smart mobile device i when computing is

$$T^P_{i(z)} = \frac{B_i}{J^P_i}, \tag{3}$$

$$N^P_{i(z)} = \frac{B_i}{K^P_i}. \tag{4}$$

In this equation, J^P_i denotes the local computing power of the smart mobile terminal held by user I, and K^P_i denotes the energy consumption of this terminal i in a single local CPU cycle. For the local computing mode, since there is no other form of time consumption and energy consumption, equations (3) and (4) denote the total time delay and total energy consumption of user i's local computing, respectively.

(2) The Task-Sharing Model. The MEC model constructed in this paper is a heterogeneous network with orthogonal frequency division multiplexing, where the channels between users accessing the same base station are orthogonal to each other, and there is only interference between accessing macro base stations. Therefore, the entire task-sharing computation process includes transmission and computation delays, and energy consumption includes transmission and execution energy consumption.

(1) Task Sharing Delay

We denote the data rate of user i choosing to access server h as

$$s^M_{i\longrightarrow h} = H_{i\longrightarrow h}\left(1 + \frac{P^h_i Q^h_i}{X^h_i + z_0}\right). \tag{5}$$

In this equation, H denotes the actual data transfer rate at which user i sends an uplink computation request to server h. P^h_i denotes the power between user i and server h. Q^h_i denotes the gain between terminal i and server h. X^h_i denotes the interference that exists between other users accessing the server and user i. z_0 denotes the background noise power.

The transmission latency for user i to share the computational task directly to the MEC server is

$$T^P_{i\longrightarrow h} = \frac{A_i}{S^M_{i\longrightarrow h}}. \tag{6}$$

The calculated delay in performing the completed tasks is

$$T^P_{i(z)} = \frac{B_i}{R^h_i}. \tag{7}$$

In this equation, R^h_i denotes the computational capacity of user i located at server h. Since there is no further form of delay in the entire computational task-sharing process, the total delay for user i to choose to share the computational task directly to the server is

$$T^P_i = \frac{A_i}{S^M_{i\longrightarrow h}} + \frac{B_i}{R^h_i}. \tag{8}$$

Substituting the decision mechanism $d_{i,j}$ into the delay of the server shown in equation (6) yields

$$T^P_{i\longrightarrow h} = \frac{A_i}{H_{i\longrightarrow h}\left(1 + \left(P^h_i Q^h_i / \sum_{l=1,l\neq i}^{M} d_{l,2} P^h_l Q^h_l + z_0\right)\right)}. \tag{9}$$

In this equation, l denotes the user accessing the server, P^h_l denotes the transmission power of l determined by server h based on some power control algorithm, and Q^h_i denotes the channel gain between user i and server h.

(2) Task-Sharing Energy Consumption

The energy consumption of user i directly sharing the computing task to the MEC server consists of both transmission energy $E^P_{i\longrightarrow b}$ and execution energy $E^P_{i(z)}$, as shown in equations (10) and (11).

$$E^P_{i\longrightarrow b} = \frac{P^h_i A_i}{T^P_{i\longrightarrow h}}, \tag{10}$$

$$E^P_{i(z)} = B_i L^h_i. \tag{11}$$

In this equation, L^h_i represents the energy consumption of user i over a single CPU cycle of the macro base station.

Since there is no further form of energy consumption in the entire computational sharing process, the total energy consumption E^P_i that user i chooses to share the computational task directly to the server can be expressed as follows:

$$E^P_i = \frac{P^h_i A_i}{T^P_{i\longrightarrow h}} + B_i L^h_i. \tag{12}$$

(3) Task-sharing Calculation

The MEC task-sharing computational model for joint consideration of latency and energy consumption is

$$\begin{aligned}\min\sum_{i=1}^{M}\Bigg\{&\partial_i^t\left[d_{i,1}\frac{B_i}{R_i^L}+d_{i,2}\left(\frac{A_i}{H_{i\longrightarrow h}\left(1+\left(P_i^hQ_i^h/\sum_{l=1,l\neq 1}^{M}d_{l,2}P_l^hQ_l^h+z_0\right)\right)}+\frac{B_i}{R_i^h}\right)\right]\\&+\partial_i^e\left[d_{i,1}B_iL_i^L+\ d_{i,2}\left(\frac{P_i^hA_i}{H_{i\longrightarrow h}\left(1+\left(P_i^hQ_i^h/\sum_{l=1,l\neq 1}^{M}d_{l,2}P_l^hQ_l^h+z_0\right)\right)}+B_iL_i^h\right)\right]\Bigg\},\end{aligned}\tag{13}$$

$$\text{s.t. }T_{i(z)}^L\leq T_i^{\max},\tag{14}$$

$$T_i^N\leq T_i^{\max},\tag{15}$$

$$\partial_i^T+\partial_i^E=1,\ldots,\partial_i^T,\quad \partial_i^E\in[0,1],\tag{16}$$

$$d_{i,1}+d_{i,2}=1,\ldots,d_{i,1},\quad d_{i,2}\in[0,1].\tag{17}$$

In this equation, the time delay $T_{i(z)}^L$ of the local execution method and the time delay T_i^N of the direct sharing of the computational task to the MEC server are given in the constraints, both of which are less than the maximum delay demand $T_i^{\max}$. The time demand factor ∂_i^T and the energy demand factor ∂_i^E in the trade-off mechanism take values in the range of $[0,1]$ and their sum is 1. The decision mechanism $d_{i,1}=1$, $d_{i,2}=0$ means that user i chooses the local computational model to compute, $d_{i,1}=0$, $d_{i,2}=1$ means that user i chooses the task-sharing computational model to compute to the server.

3.2.2. Resource Allocation Model. Considering the problem of irrational resource allocation due to malicious competition caused by irregular behavior, this paper attempts to introduce a bargaining game model based on reputation value to achieve an overall optimal allocation of computational resources [33].

Assume that the MEC server computes resources to meet all demands, a MEC server receives compute share tasks from M users in one work cycle, and the MEC server has the compute capacity to perform K CPU cycles per second. We assume that E_i represents the computational resources allocated by the MEC server for the corresponding user i. Each user competes for the computational resources on the MEC server through a bargaining game process and gives greater priority to the one with the higher reputation value, then the resource allocation model [34] is

$$\begin{aligned}&\max\prod\left(E_i-E_{i(\min)}\right)^{\theta_i}\\&\text{s.t.}\quad\sum_{i=1}^{M}E_i\leq E.\end{aligned}\tag{18}$$

In this equation, $E_{i(\min)}$ represents the minimum computational resource allocated to user i by the MEC server, and θ_i represents the user's authoritative decision factor for the MEC server's computational resource, which is related to the user's current reputation value G_i, as shown in equation (19).

$$\theta_i=\frac{G_i}{\sum_{i=1}^{M}G_i}.\tag{19}$$

We adopt the Verifiable Caching Interaction Digest schema (VCID) proposed in Ref. [35] to get reputation value G_i. This mechanism is suitable for new applications generated by intelligent terminals in the Internet of Things environment, and has good convergence and scalability. It assesses credibility based on hosts' behavior. The calculation process is

$$G_i=\alpha T(i,O)+(1-\alpha)T(O).\tag{20}$$

In this equation, $T(i,O)$ represents the direct credibility, its value depends on previous interaction experience. $T(O)$ represents the indirect credibility, and its value is based on the reputation of the authority O; i and O can be gotten from the credit institutions. α is the factor used to regulate specific gravity. When G_i is less than a certain threshold, i will be rejected for authorization.

The resource allocation model can also be modeled as a benefit function for different reputation values, where the higher the reputation value, the larger the computational resources allocated. In order to maximize the benefits, this paper adopts the Lagrange multiplier method to solve the task-sharing resource allocation game model based on the reputation value in the MEC environment. Taking the logarithm of the resource allocation and introducing a Lagrange factor of τ, the Lagrange function is constructed, as shown in equation (20).

$$T=\left(\sum_{i=1}^{M}\theta_i\ln\left(E_i-E_{i(\min)}\right)\right)+\tau\left(\sum_{i=1}^{M}E_i-E\right).\tag{21}$$

4. Model Solution

The computational task-sharing decision model developed in Section 3 includes a task-sharing decision model and a resource allocation model, corresponding to the task-sharing decision problem and the resource allocation problem, respectively. The task-sharing decision problem is nondeterministic polynomials (NP) in terms of mathematical modeling and needs to apply to multi-competition models when considering the search for the optimal mobile edge computing task-sharing scheme. Therefore, the PSO algorithm with high search accuracy and fast convergence speed is considered to solve the computational task-sharing model proposed in this paper for joint consideration of delay and energy consumption to achieve low delay and low energy consumption for computational task-sharing decisions [35]. At the same time, the amount of resources available to each server in the resource allocation problem is bounded, and the definition domain is within the computing resources of the base server. According to the theory of large values of functions with bounded definition domains, if large values of resource allocation functions exist, it can only be obtained at the point of maximum value within the definition domain or the boundary point of the definition domain, and the Lagrange multiplier method is an effective way to determine whether the function has a point of maximum value or not [36]. Therefore, to magnify the benefit function for smart mobile terminals with different reputation values, this paper uses the Lagrange multiplier method [37] to solve the task-sharing resource game theory allocation model to achieve a rational and reliable optimal resource allocation.

4.1. Task Sharing Decision Solving based on Improved PSO Algorithm

4.1.1. PSO Algorithm. The PSO algorithm is a population-based intelligent stochastic optimization algorithm based on the collaborative search for food by a flock of birds. The PSO algorithm randomly generates M particles to form a population U in an N-dimensional search space, $U = \{L_1, L_2, \ldots, L_M\}$, each particle represents a potential solution, the position of the kth particle is denoted as a vector $L_k = \{l_{k1}, l_{k2}, \ldots, l_{kN}\}$, the velocity is denoted as a vector $V_k = \{v_{k1}, v_{k2}, \ldots, v_{kN}\}$, the kth particle search to the most available position during the search process is denoted as $L_{best_k} = \{l_{best_{k1}}, l_{best_{k2}}, \ldots, l_{best_{kN}}\}$, and the optimal position searched by the population of particles during the global search process is denoted as $G_{best} = \{g_{best_1}, g_{best_2}, \ldots, g_{best_N}\}$, the kth particle's position and velocity updates are calculated as

$$\begin{aligned} v_{kd}(t+1) = \omega v_{kd}(t) + x_1 \mathrm{rand}()\left[l_{best_{kd}}(t) - l_{kd}(t)\right] \\ + x_2 \mathrm{rand}()\left[g_{best_d}(t) - l_{kd}(t)\right], \end{aligned} \tag{22}$$

$$l_{kd}(t+1) = l_{kd}(t) + v_{kd}(t+1), \tag{23}$$

$$1 \le k \le M, 1 \le d \le N. \tag{24}$$

In this equation, x_1 and x_2 are acceleration factors, indicating the extent to which particles are affected by individual values and social cognition, respectively. rand indicates the random number in $[0, 1]$. ω indicates the inertia weight, is nonnegative, and used to adjust the search scope of the solution space. t indicates the number of iterations.

4.1.2. Improved PSO Algorithm. In this paper, the proposed MEC task-sharing model solves the multi-participant NP problem, where the system consumption generated by n end-users choosing different task-sharing methods is different, but considering that the consumption values are not too different, the total value of system consumption generated by different combinations of task sharing methods does not change much and is more stable. The PSO algorithm with fast convergence is used to solve the problem. It is prone to cause the consumption of the system in high-dimensional states with multiple users involved to fall into a local optimum, so the algorithm is improved in terms of economic and exploratory capabilities, respectively. In terms of economy, in the optimization process, the inertia weight in the model is adaptively and dynamically adjusted to improve the local and global optimization capabilities of the algorithm to obtain better solution quality. In terms of exploration ability, from the perspective of mutation, the acceleration factor in the particle swarm is optimized to improve the exploration ability of the algorithm in the solution space.

(1) Inertia Weighting Improvement. The MEC offloading decision-making model in this paper is oriented to the multi-task decision-making process of multiple smart mobile terminals. The PSO algorithm tends to fall into the precocious convergence of locally optimal solutions when solving high dimensional functions of system consumption under multitasking. Using a single adjustment method where the inertial weights are kept constant or linearly decreasing, it is difficult to guarantee that every dimension of the particle tends to be optimal at the same time, making the probability of simultaneously searching for an optimal solution in every dimension of the system consumption function under multiple participation very small. Therefore, in this paper, an adaptive nonlinear dynamic approach is used to find optimality, and a cosine function is introduced based on previous studies, as shown in equation (24).

$$\omega = \frac{\omega_{\max} + \omega_{\min}}{2} \cos\left(\frac{\pi t}{T_{\max}}\right) + \frac{\omega_{\max} - \omega_{\min}}{2}. \tag{25}$$

In this equation, $\omega_{\max}$ indicates the maximum value of the inertia weight, which is generally taken as 0.7, $\omega_{\min}$ indicates the very small value of the weight, which is generally taken as 0.1, and $T_{\max}$ indicates the maximum number of iterations. Then, the value of $(t/T_{\max})$ is between 0 and 1. And, because the cosine function is monotonically decreasing over the interval $[0, \pi]$, ω is going to increase as t increases.

The weight of inertia controls the influence of the historical position of particles on the current search state, and maintains the balance between global search and local search, which can effectively achieve adaptive nonlinear adjustment of its value and improve the efficiency and intelligence of the algorithm. In the early stage of the algorithm, to increase the global search, the inertia weights should have a larger value; in the late algorithm, they should maintain a reasonable convergence rate to increase the local search capability, so the inertia weights should be kept smaller.

(2) Improvement of the Acceleration Factor. Since particle position is influenced by both individual and population extremes, the cognitive part and social part could greatly affect the direction and velocity of particle convergence. The acceleration factor x_1 in the PSO algorithm usually takes a value of 0.43 and x_2 usually takes a value of 0.4. But, in this paper's MEC task-sharing decision-oriented model, when the number of iterations varies, there is a bias about the leading position between the cognitive part and social part. When the number of iterations is small, the cognitive part represented by x_1 plays a dominant role, and the social part represented by x_2 plays a secondary role. When the number of iterations is large, the accumulation of social knowledge continues to increase, just the opposite. This paper faces dynamic multi-local computation problems and server computation selection problems in the MEC task-sharing decision. To further improve the PSO algorithm's ability to explore the solution space, we improved the acceleration factor from a variation perspective. The dynamic acceleration factor is used instead of the static acceleration factor as shown in equations (25) and (26).

$$x_1 = \frac{\alpha}{t}, \tag{26}$$

$$x_2 = \beta t^2. \tag{27}$$

In this equation, t indicates the number of iterations. α indicates the individual cognitive impact factor, which takes values ranging from 149 to 280 in this scenario. β indicates the social cognitive impact factor, which takes values ranging from 0.00013 to 0.000205 in this scenario.

4.1.3. Task Sharing Decision Method based on Improved PSO Algorithm. The mapping of the solution space of the improved PSO algorithm to the task-sharing decision problem is shown in Table 2.

The task-sharing decision-solving process based on the improved PSO algorithm is shown in Algorithm 1.

Algorithm 1 The task sharing decision-solving process based on the improved PSO algorithm.

Step 1. Initialize: Spatial dimension $N = 50$ and population size $M = 100$ were determined; inertial weight extremes $\omega_{\max}$ and $\omega_{\min}$ were set to 0.7 and 0.1, respectively; and initial position and initial velocity were randomly generated within the search space.

Step 2. Find the adaptation value. Calculate the total consumption of the system based on the adaptation function shown in equation (13).

Step 3. Find the individual extreme value L_{best_k} and the population extreme value G_{best}. Find the minimal value L_{best_k} of system consumption resulting from choosing different offloading methods and the optimal value G_{best} of system consumption for all different combinations of offloading methods.

Table 2: The table describes the solution space of the improved PSO algorithm with respect to the task sharing decision problem.

Improved PSO algorithm	Task-sharing decision-making issues
Spatial dimension	Quantity
Stocks	Pooling of different task-sharing decisions
Particle location	Different task-sharing decisions
Fitness value	Total system consumption

Step 4. Update the particle position and velocity. Update the velocity and position according to equations (21) to (26). Each particle shares the same distance from the current system eigenvalue. Particle k selects the nearest consumption target to the system; the other particles will vectorize their positions relative to particle k and their preferred positions away from the small system energy target.

Step 5. Update the individual polar value L_{best_k} and the group polar value G_{best}. Compare the system consumption value of each particle to L_{best_k}. Replace L_{best_k} with the current position if the current consumption value is smaller than L_{best_k}. Then, compare the minimum system consumption position in L_{best_k} with G_{best}. Replace G_{best} if the system consumption value in L_{best_k} is smaller than G_{best}.

Step 6. Determine the termination condition. The termination condition is judged by the number of iterations of the algorithm. If the termination condition is satisfied, exit the loop and return the superior search result G_{best}. If the termination condition is not satisfied, repeat steps 2 to 5 until the termination condition is satisfied.

Step 7. Outputs the optimal solution, which is the small value consumed by the system.

Since the resource allocation problem is NP-hard, we use a heuristic algorithm to solve it. It can be seen from the pseudo-code of the algorithm that the running time depends on the number of mobile devices participating in the allocation and the number of iterations set. The time complexity is $O(n^2)$, but since n here is generally not a large number, the running time is acceptable.

4.2. Resource Allocation Solution based on Lagrange Multiplier Method. The resource allocation model is aimed at the benefit function of smart mobile terminals with different reputation values, and the amount of resources available to each smart mobile terminal is not greater than the computing resources of the MEC server. To maximize the benefits of MEC resources, the maximum value point of the benefit function is only obtained within the bounded range or boundary point of the computing resource. We know that the Lagrangian multiplier method is an effective method to determine whether the benefit function has the maximum value of computing resources. So, this paper adopts the Lagrangian multiplier method to solve the

```
Input: acceleration factors $x_1, x_2$, inertia weight $\omega$, iterations $t_0$, task-sharing decision model mentioned above
Output: the minimum of total system consumption of task-sharing decision model
Begin
$N = 50$, $M = 100$//spatial dimension, population size
$\omega\mathrm{min}_{\max}$, $t = 0$
$fitness(x)$←Equation (13)//take total system consumption as the fitness function
For each particle $k$
    Initialize velocity $V_k$ and position $L_k$ for particle $k$ randomly
    evaluate particle $k$ and set $L_{best_k} = L_k$
End for
While $t \le t_0$
For $k = 1$ to 100
    Update the velocity and position of particle $i$ according to equations (21), (22), (25), and (26)
    Evaluate particle i
    if $fitness(L_k) < fitness(L_{best_k})$
    $L_{best_k} \leftarrow L_k$
    if $fitness(L_{best_k}) < fitness(G_{best})$
    $G_{best} \leftarrow L_{best_k}$
End for
End while
print $G_{best}$
Stop
```

Algorithm 1: Task-sharing decision model using improved PSO algorithm.

abovementioned unloading resource game allocation model based on reputation value. Take the logarithm of the resource allocation model and introduce the Lagrangian factor τ to construct the Lagrangian function, as shown in equation (20).

Using the one-time bias derivative of T for τ as 0, the one-time bias derivative can determine the locally optimal feasible solution, yielding equation (27).

$$\tau = \frac{-1}{E - \sum_{i=1}^{M} E_{i(\min)}}. \tag{28}$$

The quadratic partial derivative of t is then performed to obtain equation (28).

$$\frac{\partial^2 T}{\partial \tau^2} = \frac{1}{\tau^2}. \tag{29}$$

Since the quadratic partial derivative of τ is greater than zero, it can be determined that there is a most-valued solution to the benefit function.

Substitute equation (27) into equation (20) to obtain equation (29).

$$E_i = E_{i(\min)} + \theta_i \left(E - \sum_{i=1}^{M} E_{i(\min)} \right). \tag{30}$$

Ultimately, the MEC server allocates computational resources according to the results of equation (29), performs different computational task-sharing tasks, and reports the results of the computation back to the end-user.

The process of solving the resource allocation based on the Lagrange multiplier method is shown in Algorithm 2.

Algorithm 2. The process of solving the resource allocation based on the Lagrange multiplier method.

Step 1. Taking the logarithm of the target function for users with different reputation values and introducing the Lagrange factor τ, then construct the Lagrange function T

Step 2. Perform a first partial derivation of the constructed benefit function T about the computational resource requirements E_i of each MEC server to obtain a feasible solution for the local superiority of the computational resources. Then, perform a second partial derivation, and if the second partial derivation is not equal to 0 then prove that the benefit function has a valued solution.

Step 3. Make the value of the bias derivative of the benefit function T with respect to the computational resource requirement E_i for each MEC server equal to 0, and solve for the stationary point $E_i = E_{i(\min)}$.

Step 4. Obtain a bias derivative of the benefit function T with respect to the Lagrange factor τ once.

$$\frac{\partial T}{\partial \tau} = -\frac{1}{\tau} + \sum_{i=1}^{M} E_{i(\min)}. \tag{31}$$

Step 5. From the formula (28), it can be seen that the benefit function exists as the most valuable solution. Then, make the value of the benefit function T on the Lagrange factor τ of the first partial derivative equal to 0, the solution to the value of τ as shown in formula (27).

$$E_i = E_{i(\min)} + \frac{G_i}{\sum_{i=1}^{M} G_i} \left(E - \sum_{i=1}^{M} E_{i(\min)} \right). \tag{32}$$

Algorithm 2 can be described in Algorithm 2. It can be seen that the time complexity of Algorithm 2 is $O(n)$.

```
Input: the reputation value $G_i$, the minimum computational resource $E_{i(\min)}$ of every device $i$, the total computational resource of the MEC server $E$
Output: the deserved computational resource $E_i$ of every device i
Begin
For $i = 1$ to M
    $\theta_i = (G_i/\sum_{i=1}^{M} G_i)$//Calculate the authoritative decision factors
End for
$T = (\sum_{i=1}^{M} \theta_i \ln(E_i - E_{i(\min)})) + \tau(\sum_{i=1}^{M} E_i - E)$//Construct Lagrangian function
For each device $i$ requesting for resource
Calculate the partial derivative of $T$ with respect to $E_i$
If $(\partial^2 T/\partial \tau^2) > 0$
    For $i = 1$ to M
        $E_i = E_{i(\min)} + (G_i/\sum_{i=1}^{M} G_i)(E - \sum_{i=1}^{M} E_{i(\min)})$
    End for
End if
Print $E_i\,(i = 1 \text{ to } M)$
Stop
```

Algorithm 2: Resource allocation based on the Lagrange multiplier method.

5. Experimental Simulation

In this paper, we use ONE, an open-source simulation tool, to evaluate the performance of DERV. We compare the packet transfer ratio, average end-to-end delay [38], routing overhead [39], and energy consumption with the FCNS, SCR, MILECR, and epidemic algorithm [40] to verify the advantages of DERV in terms of system overhead. In addition, we considered the effects of three different schemes on the overall energy consumption of the model [41], namely, low energy task sharing, randomly assigned task-sharing [42], and task sharing with joint delay and energy consumption.

5.1. Parameter Settings. To follow the characteristics of massive data transfer in a short period in a 5G environment, this paper uses real datasets downloaded from CRAWDAD to drive node activities, and datasets from Infocom 5, Infocom 6, Cambridge, and Intel are selected for simulation [43]. The specific simulation environment parameter settings are shown in Table 3.

The number of nodes set up in this experiment is 50, the node computing power is randomly assigned to 500–1000 MHz/s, and the computing power of the server is assigned to 5 GHz/s. The specific physical model parameter settings are shown in Table 4.

5.2. Metrics. Metrics are used to measure the performance of algorithms in opportunistic complex social networks. Four are selected in this paper, they are packet transfer ratio, average end-to-end delay, routing overhead, and overall energy consumption. Here is a brief introduction of what they mean.

(1) Packet transfer ratio: the ratio of the number of packets received at the destination to the number of packets in recent years, with the deep integration ts sent from the source.

(2) Average end-to-end delay: the average delay in data transmission between two nodes.

(3) Routing overhead: the total size of routing packets sent for maintenance and also for route discovery.

(4) Energy consumption: ALL the energy used in data transmission between two nodes.

5.3. Analysis of Results. As the simulation time increases, the clustered bar graphs in Figures 3(a)–3(d) show the packet transfer ratios of the DERV, FCNS, SCR, MILECR, and epidemic algorithms. When the simulation time is short, the performance advantage of the DERV algorithm is not as pronounced as the other four algorithms, but as the simulation time increases, the success rate of the DERV algorithm is significantly higher than that of the other four algorithms. This is because our solution fully considers the problem of nodes maliciously competing for computational resources, and introduces the measure of reputation, where nodes with a higher reputation are allocated more computational resources, thus achieving an overall optimal allocation of computational resources.

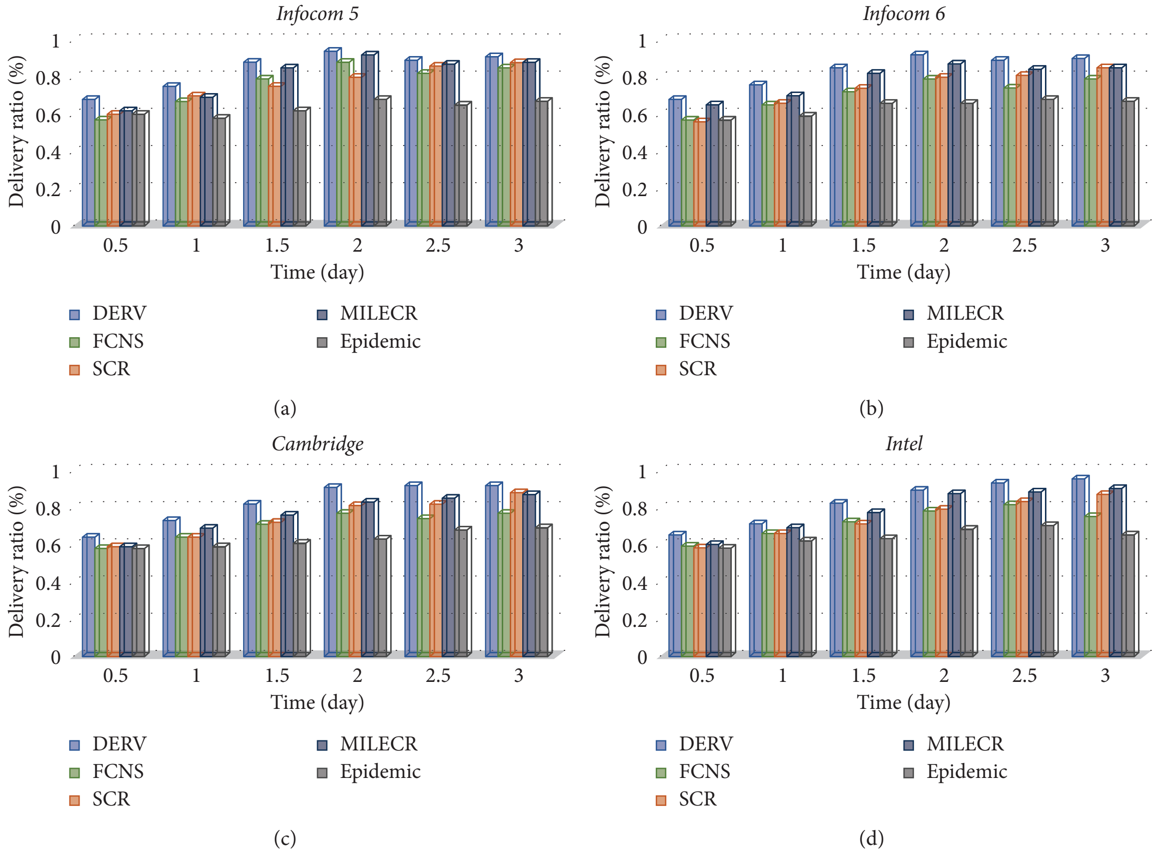
Figures 4(a)–4(d) show the comparison results of the average end-to-end delay for the DERV, FCNS, SCR, MILECR, and epidemic algorithms as the simulation time increases. Among the comparison schemes, the DERV strategy has the lowest average end-to-end latency and this advantage becomes more pronounced with the increase in the simulation time. This is because the DERV algorithm introduces delay and energy consumption trade-off factors, enables full sharing of multitasking computational resources across multiple smart mobile devices, and optimizes the routing strategy for data transfer more efficiently than the other schemes. In contrast, regarding the epidemic algorithm, a large number of copies of information are generated during data transfer, which will lead to an increase in forwarding delay. In addition, SCR and MILECR algorithms use the strategy of neighbor node cooperative transmission,

Table 3: The parameter settings for a specific simulation environment.

Dataset	Infocom5	Infocom6	Cambridge	Intel
Device	iMote	iMote	iMote	iMote
Duration (day)	3.5	4	11.5	4
Number of experimental devices	41	98	52	9
Number of internal contacts iMote	22459	170601	10873	1364
Number of nodes	41	98	52	9
Buffer size(M)	5	5	5	5
TTL	60 min	60 min	2 days	0.5 day

Table 4: Specific physical model parameter settings.

Parameters	Value
Node's computing power (GHz)	[0.5,1.0]
Node's reputation value	[0,10]
Time requirement factor	[0,1]
Acer station's computing power (GHz)	10
Power (mW)	100
Gain between nodes and station	10^{-6}
Background noise power (dBm)	−100
The actual bandwidth of the uplink calculation request (kHz)	15
Calculating task data maximum (kB)	5000
Individual CPU power consumption of the station (W)	5
Total CPU required for the task (mc)	1000



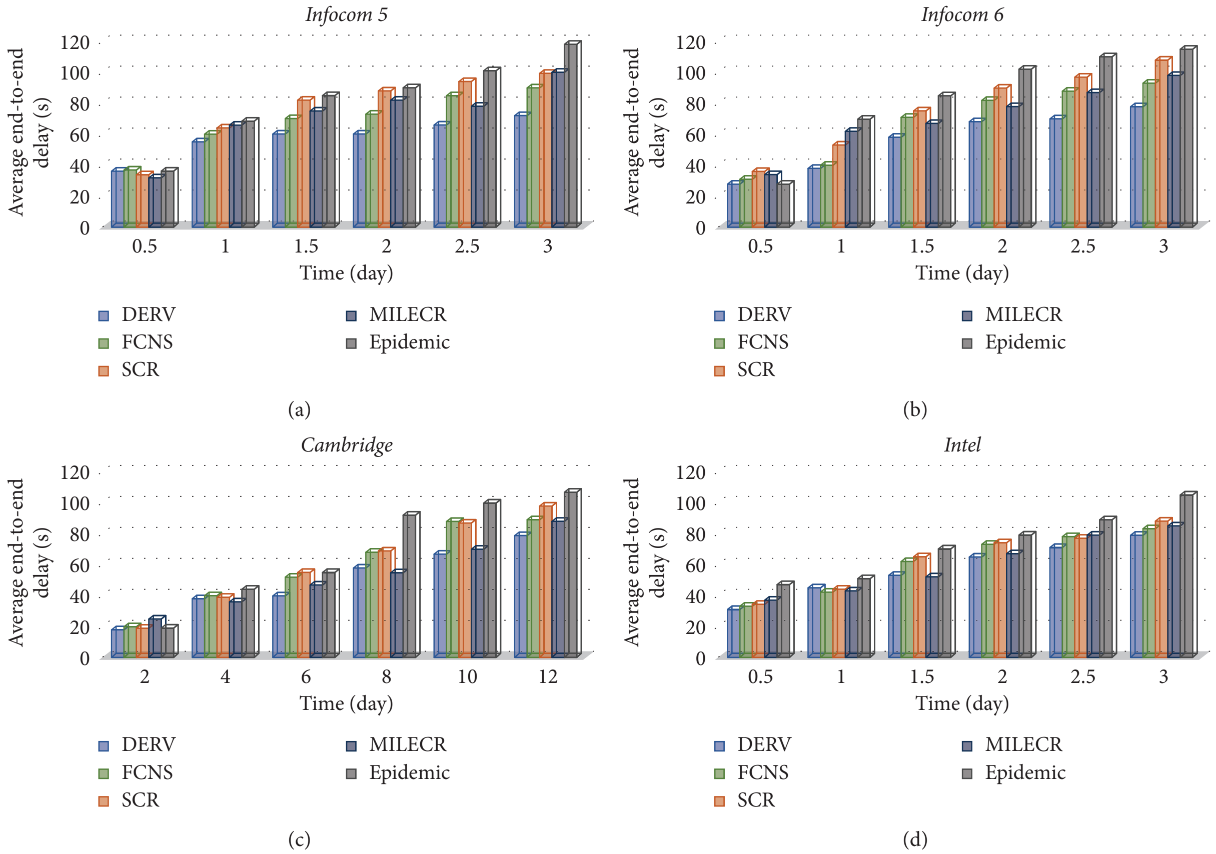
(a)

(b)

(c)

(d)

Figure 3: (a)–(d) The packet transfer ratios of the DERV, FCNS, SCR, MILECR, and epidemic algorithms on datasets Infocom5, Infocom6, Cambridge, Intel.



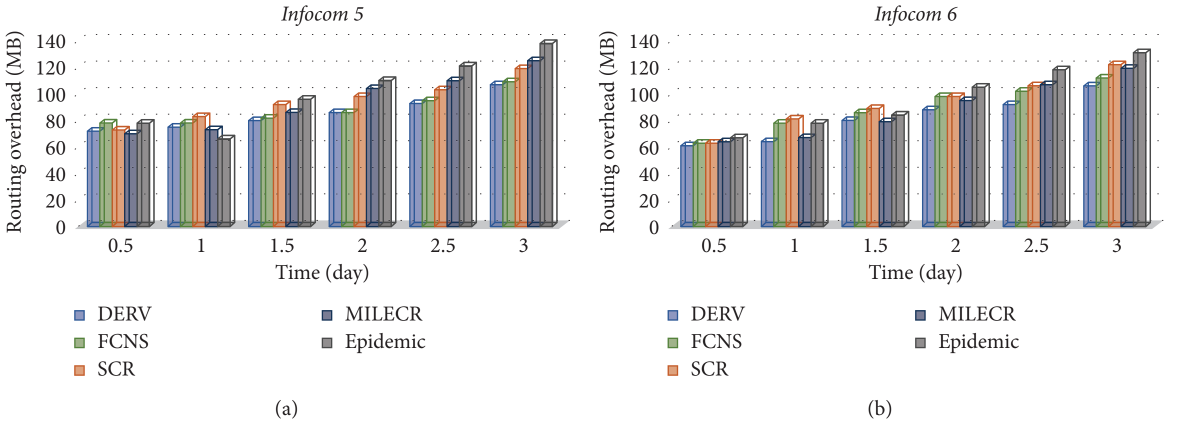
(a)

(b)

(c)

(d)

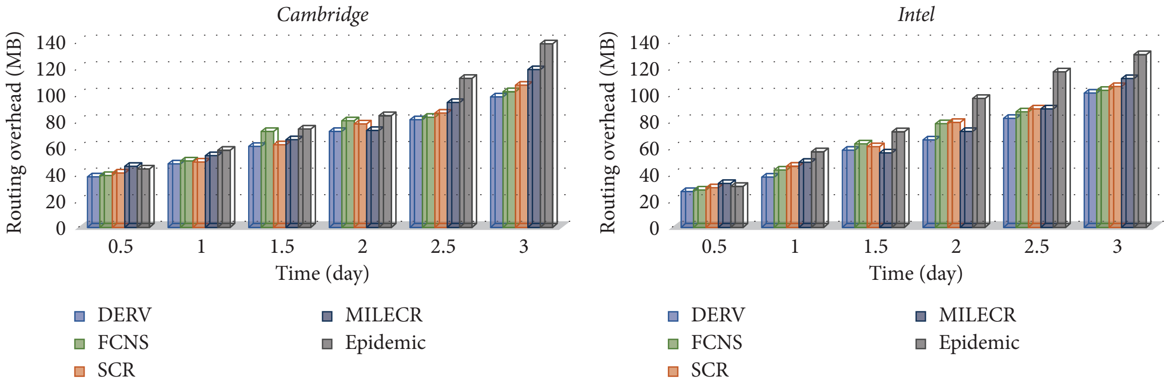
Figure 4: (a–d) The average end-to-end delay for the DERV, FCNS, SCR, MILECR, and epidemic algorithms on datasets Infocom5, Infocom6, Cambridge, Intel.



(a)

(b)

Figure 5: Continued.



(c)

(d)

Figure 5: (a, b) The routing overhead of the DERV, FCNS, SCR, MILECR, and epidemic algorithms on datasets Infocom5, Infocom6, Cambridge, Intel.

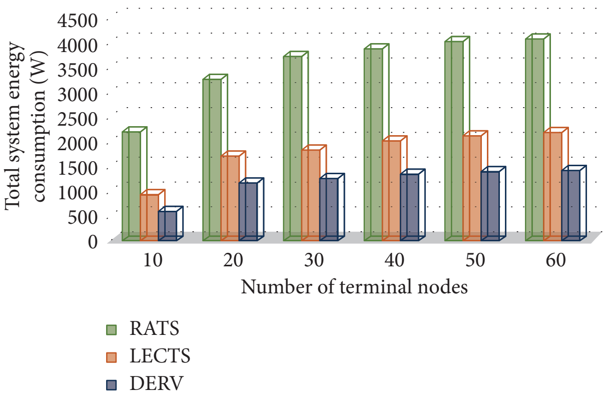


Figure 6: The overall energy consumption of three different schemes.

which reduces the impact of node cache on message transmission, but still has high latency when the experiment is longer. FCNS algorithm is more affected by node cache and has poorer performance when the node cache space is smaller. In summary, the DERV algorithm is the best method to improve the average end-to-end delay performance in high-speed communication scenarios compared to other algorithms.

Figures 5(a) and 5(b) show the comparison results of the routing overhead of the DERV, FCNS, SCR, MILECR, and epidemic algorithms when the simulation time increases. DERV can better predict the next-hop node compared to other models. By sending messages to nodes that satisfy the reputation value in the communication domain, the routing cost of sending messages to other noncooperating nodes can be effectively reduced. In addition, nodes do not need to use a computational model for continuous computation and decision-making during message transmission. This can reduce the time and routing resource costs. In the case of the epidemic algorithm, a large number of redundant message copies require time and computational resources, and the routing overhead is significantly higher than that of other algorithms. For SCR and MILECR algorithms, the cooperative mechanism facilitates the rational allocation of computational resources, so the cost of these two algorithms is at an intermediate level. The FCNS algorithm takes into account the mobile similarity of nodes, but does not fully consider the transmission preferences of nodes, so its performance is poorer than the DERV algorithm. Compared to the results, DERV outperforms the other four models in terms of routing overhead.

Figure 6 shows a comparison of the overall energy consumption of three different schemes, namely, low energy task sharing (LETS), randomly assigned task sharing (RATS), and task sharing with joint time delay and energy consumption in this paper. As shown in the figure, as the number of end nodes increases, the total energy consumption of the system under all three schemes increases, while the total energy consumption of the scheme proposed in this paper is always at the lowest level.

6. Conclusion

This paper proposes a task-sharing model (DERV) based on reputation value, which solves the problems of high latency, high energy consumption, and low reliability faced by computing sharing in emerging mobile applications in the big data environment. DERV divides the computing sharing task in the opportunistic complex IoT environment into two processes: shared decision-making and resource allocation. The experimental results show that the computing task-sharing model proposed in this paper can meet the service requirements of low latency, low energy consumption, and high reliability in emerging intelligent applications, and can effectively realize the overall optimized configuration of computing shared resources. Among them, the main innovations and contributions of this article are as follows:

(1) This article focuses on providing low-latency, low-energy consumption, and high-reliability service quality guarantees for time-delay and energy-sensitive computing-intensive smart mobile terminals in a big data environment. We propose a network

model composed of users, MEC servers, and trusted institutions.

(2) To realize the multitasking computational task-sharing scheme for multiple smart mobile terminals, we propose a task-sharing decision model that takes into account both delay and energy consumption. The model uses an improved PSO algorithm to achieve an optimal computing task-sharing scheme with low latency and low energy consumption.

(3) Aiming at the problem of unreasonable resource allocation caused by malicious competition for resources due to irregular behavior in the IoT system, this paper proposes a bargaining game model based on reputation value, which mainly uses the Lagrangian multiplier method to realize the calculation Reliable allocation of resources.

In the future, with the increase in the computing power of mobile devices in opportunistic complex social networks, the DERV model proposed in this paper can be applied to the transmission environment of 5G and big data networks. We will collect larger real datasets in social scenarios and explore ways to improve information transmission performance.

Data Availability

The data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by The National Natural Science Foundation of China (61762051 and 82160955).

References

[1] S. N. Moosavi and V. Pourahmadi, "Opportunistic multiple access (OMA) for crowdsensing networks with sparse activation," *Trans Emerging Tel Tech*, vol. 30, Article ID e3559, 2019.

[2] J. Wu, Z. Chen, and M. Zhao, "An efficient data packet iteration and transmission algorithm in opportunistic social networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 3141–3153, 2020.

[3] J. Wu, Z. Chen, and M. Zhao, "Community recombination and duplication node traverse algorithm in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 13, pp. 940–947, 2020.

[4] J. Lorincz, A. Capone, and J. Wu, "Greener, energy-efficient and sustainable networks: state-of-the-art and new trends," *Sensors*, vol. 19, no. 22, Article ID 4864, 2019.

[5] G. Yu and J. Wu, "Content caching based on mobility prediction and joint user Prefetch in Mobile edge networks," *Peer-to-Peer Networking and Applications*, vol. 13, pp. 1839–1852, 2020.

[6] J. LUO, W. U. Jia, and Y. WU, "Advanced data delivery strategy base on multi-perceived community with IoT in social complex networks," *Complexity*, vol. 2020, Article ID 3576542, 20 pages, 2020.

[7] Y. A. N. G. Weiyu, W. U. Jia, and J. LUO, "Effective date transmission and control base on social communication in social opportunistic complex networks," *Complexity*, vol. 2020, Article ID 3721579, 13 pages, 2020.

[8] M. S. Abdalzaher and O. Muta, "A game-theoretic approach for enhancing security and data trustworthiness in IoT applications," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11250–11261, 2020.

[9] W. U. Jia, C. H. A. N. G. Liu, and Y. U. Genghua, "Effective data decision-making and transmission system based on mobile health for chronic diseases management in the elderly," *IEEE Systems Journal*, 2020.

[10] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[11] H. Zhang, J. Guo, L. Yang, X. Li, and H. Ji, "Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC," in *Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 115–120, Atlanta, GA, USA, May 2017.

[12] Y. Lu, L. Chang, and J. Luo, "Routing algorithm based on user adaptive data transmission scheme in opportunistic social networks," *Electronics*, vol. 10, Article ID 1138, 2021.

[13] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep reinforcement learning based computation offloading and resource allocation for MEC," in *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, April 2018.

[14] Y. Nan, W. Li, W. Bao et al., "Adaptive energy-aware computation offloading for cloud of things systems," *IEEE Access*, vol. 5, pp. 23947–23957, 2017.

[15] Y. L. Teng, W. Liu, W. P. Ouyang, L. I. Kun, and M. Song, "Queue-aware joint optimization of offloading and transmission in wireless mobile edge computing systems," *Journal of Beijing University of Posts and Telecommunications*, vol. 42, 2019.

[16] H.-H. Chang, W.-Y. Chiu, H. Sun, and C.-M. Chen, "User-centric multiobjective approach to privacy preservation and energy cost minimization in smart home," *IEEE Systems Journal*, vol. 13, no. 1, pp. 1030–1041, 2019.

[17] D. Waters, A. Donnellan, and J. Fox, "An adaptable internet of things network infrastructure implemented for a smart building system," in *Proceedings of the 2021 32nd Irish Signals and Systems Conference (ISSC)*, pp. 1–7, Athlone, Ireland, June 2021.

[18] K. Mamo, J. I. Nieto, R. Buenrostro, and M. Z. Ali, "Spectrum based power management for congested iot networks," *Sensors*, vol. 21, Article ID 14, 2021.

[19] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7635–7647, 2019.

[20] B. Huang, Y. Li, Z. Li et al., "Security and cost-aware computation offloading via deep reinforcement learning in mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 3816237, 20 pages, 2019.

[21] Y. Dong, L. Chang, J. Luo, and W. Jia, "A routing query algorithm based on time-varying relationship group in opportunistic social networks," *Electronics*, vol. 10, Article ID 1595, 2021.

[22] L. I. Xiaoli and W. U. Jia, "Node-oriented secure data transmission algorithm based on IoT system in social networks," *IEEE Communications Letters*, vol. 24, 2020.

[23] Z. Fang, L. Chang, J. Luo, and W. Jia, "A data transmission algorithm based on triangle link structure prediction in opportunistic social networks," *Electronics*, vol. 10, Article ID 1128, 2021.

[24] Q. Han, H. Wen, J. Wu, and M. Ren, "Rumor spreading and security monitoring in complex networks," in *Computational Social Networks. CSoNet 2015*, M. Thai, N. Nguyen, and H. Shen, Eds., Springer, Berlin, Germany, Springer, 2015 Lecture Notes in Computer Science.

[25] Y. Deng, F. Gou, and J. Wu, "Hybrid data transmission scheme based on source node centrality and community reconstruction in opportunistic social network," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1–13, 2021.

[26] J. Wu, S. Yin, Y. Xiao, and G. Yu, "Effective data selection and management method based on dynamic regulation in opportunistic social networks," *Electronics*, vol. 9, Article ID 1271, 2020.

[27] Y. Xiao and J. Wu, "Data transmission and management based on node communication in opportunistic social networks," *Symmetry*, vol. 12, no. 8, Article ID 1288, 2020.

[28] Y. I. N. Sheng, W. U. Jia, and Y. U. Genghua, "Low energy consumption routing algorithm based on message importance in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1–14, 2020.

[29] W. U. Jia, Q. U. Jingge, and Y. U. Genghua, "Behavior prediction based on interest characteristic and user communication in opportunistic social networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 1006–1018, 2021.

[30] J. Wu, W. Zou, and H. Long, "Effective path prediction and data transmission in opportunistic social networks," *IET Communications*, vol. 15, pp. 2202–2211, 2021.

[31] Y. A. N. G. Weiyu, J. LUO, and W. U. Jia, "Application of information transmission control strategy based on incremental community division in IoT platform," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21968–21978, 2021.

[32] F. Gou and J. Wu, "Triad link prediction method based on the evolutionary analysis with IoT in opportunistic social networks," *Computer Communications*, vol. 181C, pp. 143–155, 2022.

[33] W. Lin, X. Zhang, L. Qi, W. Li, and S. Nepal, "Location-aware service recommendations with privacy-preservation in the internet of things," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 99, pp. 1–9, 2020.

[34] S. Li, K. K. R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "IoT forensics: amazon echo as a use case," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6487–6497, 2019.

[35] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, "A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems," *Future Generation Computer Systems*, vol. 105, pp. 789–799, 2020.

[36] Y. Yuan, X. Zhang, and J. Tang, "Guest editorial special issue on privacy and security in computational intelligence," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, pp. 590–592, 2020.

[37] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, vol. 23, 2020.

[38] X. Xu, X. Zhang, H. Gao, Y. Xue, and W. Dou, "BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, 2019.

[39] W. Tang, C. Wu, L. Qi, X. Zhang, X. Xu, and W. Dou, "A WiFi-aware method for mobile data offloading with deadline constraints," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 7, pp. 1–15, Article ID e5318, 2021.

[40] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, and Q. Sun, "Distributed consensus algorithm for events detection in cyber-physical systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2299–2308, 2019.

[41] Y. Yuan and W. Banzhaf, "Making better use of repair templates in automated program repair: a multi-objective approach," *Evolution in Action: Past, Present and Future*, Springer, Berlin, Germany, 2020.

[42] X. Zhao, W. Dou, X. Yin, H. Wang, and L. Qi, "Edge computing-enabled deep learning for real-time video optimization in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 99, p. 1, 2020.

[43] W. Dou, W. Tang, B. Liu, X. Xu, and Q. Ni, "Blockchain-based mobility-aware offloading mechanism for fog computing services," *Computer Communications*, vol. 164, no. 6, pp. 261–273, 2020.

Hindawi
Complexity
Volume 2021, Article ID 1055604, 14 pages
https://doi.org/10.1155/2021/1055604

Research Article

A Garbage Detection and Classification Method Based on Visual Scene Understanding in the Home Environment

Yuezhong Wu,[1,2] **Xuehao Shen**,[1] **Qiang Liu**,[2,3] **Falong Xiao**,[1] **and Changyun Li**[2,3]

[1]*College of Railway Transportation, Hunan University of Technology, Zhuzhou 412007, China*
[2]*College of Computer Science, Hunan University of Technology, Zhuzhou 412007, China*
[3]*Intelligent Information Perception and Processing Technology Hunan Province Key Laboratory, Zhuzhou, China*

Correspondence should be addressed to Qiang Liu; liuqiang@hut.edu.cn and Changyun Li; lichangyun@hut.edu.cn

Received 20 August 2021; Revised 14 September 2021; Accepted 2 November 2021; Published 26 November 2021

Academic Editor: Xuyun Zhang

Garbage classification is a social issue related to people's livelihood and sustainable development, so letting service robots autonomously perform intelligent garbage classification has important research significance. Aiming at the problems of complex systems with data source and cloud service center data transmission delay and untimely response, at the same time, in order to realize the perception, storage, and analysis of massive multisource heterogeneous data, a garbage detection and classification method based on visual scene understanding is proposed. This method uses knowledge graphs to store and model items in the scene in the form of images, videos, texts, and other multimodal forms. The ESA attention mechanism is added to the backbone network part of the YOLOv5 network, aiming to improve the feature extraction ability of the network, combining with the built multimodal knowledge graph to form the YOLOv5-Attention-KG model, and deploying it to the service robot to perform real-time perception on the items in the scene. Finally, collaborative training is carried out on the cloud server side and deployed to the edge device side to reason and analyze the data in real time. The test results show that, compared with the original YOLOv5 model, the detection and classification accuracy of the proposed model is higher, and the real-time performance can also meet the actual use requirements. The model proposed in this paper can realize the intelligent decision-making of garbage classification for big data in the scene in a complex system and has certain conditions for promotion and landing.

1. Introduction

In recent years, as the global garbage production has shown a cliff-like growth, my country has also introduced a series of policies. The latest revision of the "Law of the People's Republic of China on the Prevention and Control of Environmental Pollution by Fixed Wastes" in 2020 requires that local people's governments at or above the county level should speed up the establishment of a domestic waste management system for classified release, recycling, transportation, and treatment. At this stage, the garbage classification is mainly concentrated in fixed places in the outdoor public environment. There are problems such as high labor intensity, low sorting efficiency, and poor working environment. In fact, the garbage classification in the home environment can really solve the problem from the source. However, because the people's awareness of classification is not strong, the classification is troublesome, and there are many types of garbage; people seldom actually throw garbage in categories. In recent years, home service robots have attracted widespread attention. Among them, sweeping robots are the first products to realize industrialization and have entered the consumer market widely. Although the sweeping robots currently on the market have basic functions such as path planning [1, 2], automatic charging, and automatic obstacle avoidance, their intelligence is still not high. Although a simple path planning function is added to the cleaning process, the cleaning process is blind. No matter whether there is garbage in the working path that needs to be processed, the cleaning action will be performed, and the work efficiency is low. In addition, it does not have the ability to distinguish whether items are garbage or not, nor does it have the ability to treat garbage by category. In fact, according to the shape, material, and other attributes of the item itself, as well as the relationship with other items, such as its location, you can further determine whether it is garbage, improve its intelligence, and avoid waste of resources; and different types of garbage should be sorted by category to meet environmental protection requirements.

In order to solve the above problems, a feasible solution is to perform intelligent garbage classification tasks on the home service robot. On the one hand, the home service robot is equipped with visual sensors to enable it to obtain visual perception capabilities [3]; on the other hand, research on effective perception detection algorithms aims to achieve the purpose of visual scene understanding and ultimately guide home service robots to autonomously perform intelligent garbage classification, improve work efficiency, and reduce energy consumption. At present, there have not been public reports about the work carried out on the autonomous garbage detection and classification of household service robots. Therefore, the realization of garbage classification and detection algorithms on household service robots has certain practical significance. However, only using the detection and classification model can only realize the identification and positioning of the garbage, and the degree of intelligence is not high. To make the robot achieve the ability of cognition and discrimination of objects in the home environment like humans, for example, humans can understand what they see, the items in the scene can be associated and imagined based on these items, not only relying on the appearance and geometric characteristics of the items, but also relying on the guidance and reasoning of the high-level prior knowledge of the items.

If you want service robots to have the ability to recognize and discriminate objects in the scene like humans, perhaps visual scene understanding can be competent. Visual scene understanding needs to understand not only the information of each entity object in the image, but also the relationship between the entity objects. Visual scene understanding, called image semantic description, is a hot issue combining machine vision and natural language processing [4–6]. Home environment information has the characteristics of diversity, semantics, and relevance. Intelligent decision-making for garbage classification based on big data of items in the scene is the key issue studied in this paper. In order to achieve the intelligent decision of whether items are garbage in the home environment, this paper proposes a garbage detection and classification method based on visual scene understanding. The main contributions of this paper are as follows: first, the construction of the scene multimodal knowledge graph. Aiming at the problem of rich and diverse semantics of items in the home environment, which is difficult to model, the knowledge graph is used to uniformly represent and store the input multimodal information; the second is to propose a garbage classification and detection model YOLOv5-Attention-KG based on visual scene understanding. Combining the improved YOLOv5m detection algorithm with the knowledge graph and deploying it to the edge device home service robot, the system has the ability to associate similar to people, which is the key to improving the system's intelligent garbage classification.

The subsequent chapters of this paper are arranged as follows. Section 2 starts with object detection, including traditional methods and deep learning, and then leads to knowledge graphs and finally mentions edge computing as a demonstration application of the system; the interrelationship between them and how to integrate them into the method proposed in this paper is shown in Section 3; Section 4 discusses the relevant analysis and verification of the experimental results of the model proposed in this paper; Section 5 summarizes the research work of this paper and the prospects for the next research work.

2. Related Work

Most of the traditional object detection algorithms are based on manual design and extraction of features, combined with the construction of classifiers, which have disadvantages such as relatively complex models and poor robustness. Deep learning is an important breakthrough in the field of artificial intelligence in the past decade. Since the multilayer convolutional neural network can autonomously extract and filter the features of different layers, compared with the traditional target detection method, the detection effect is more accurate and the generalization ability is stronger. At present, the target detection methods based on deep learning are divided into two types: one-stage and two-stage [7, 8]. Typical two-stage methods include region-based convolutional neural network (RCNN) method [9], Fast RCNN method [10], Faster RCNN method [11], region-based complete convolutional network (R-FCN) method [12], and other improvement methods [13]. A typical representative of the one-stage method is the YOLOv1 algorithm proposed by Redmon et al. [14]. Since YOLOv1 directly fits the position coordinates and confidence level, there are obvious defects. On the basis of YOLOv1, Redmon et al. proposed YOLOv2 [15], which uses the new basic network structure DarkNet-19, to delete the full connection layer and the last pooled layer, and uses the anchor frame to predict the boundary box. YOLOv3 [16] is the last version of the YOLO method proposed by Redmon et al. Two years later, Alexey et al. proposed the YOLOv4 [17] method. In order to improve the model's ability to detect accuracy and small targets, they proposed a better basic network, DarkNet-53, and used some techniques to improve performance. Another type of single-stage detection method is represented by SSD [18]. In recent years, many researchers have used deep learning technology in the research of garbage classification. Yang et al. [19] created a garbage classification data set and established a garbage classification model using support vector machines and convolutional neural networks. Mao et al. [20] used genetic algorithms to optimize the parameters of the fully connected layer of the DenseNet121 network and trained a garbage classification model with a classification accuracy of 99.60%. Literature [21] uses a self-encoding network to reconstruct the garbage classification data set and uses CNN to automatically extract features from the data set. Zhang et al. [22] used the Faster RCNN algorithm to detect

681 street pictures with 9 categories of garbage targets, and the detection mAP was 0.82, but there was a problem of unbalanced categories. Seredkin et al. [23] used Faster RCNN network to perform garbage classification which has high accuracy and effectively realized garbage identification. Chen et al. [24] used the Faster RCNN algorithm to detect 199 garbage targets on the pipeline and obtained a system missed identification rate of 3% and a false identification rate of 9%. Abeywickrama et al. [25] regarded garbage classification as image classification, used support vector machines and convolutional neural networks to identify and classify 6 types of garbage, and achieved a recognition result with a recognition accuracy of 83%. Mikami et al. [26] produced a dataset consisting of 2561 garbage images and designed a GarbNet model with an accuracy rate of 87.69%. With the increasing demand for garbage detection and classification of mobile edge devices, most of the hardware used in these scenarios is edge devices with weak computing power, and some larger detection networks are difficult to deploy. YOLOv5 launched by Ultralytics in 2020 has the advantages of small size, fast speed, and high precision and is suitable for deployment on edge devices. Therefore, this paper uses YOLOv5 as the basic network. In addition, since the above studies are based on the premise that the object is garbage, it mainly relies on a large amount of labeled data to fit a large number of parameters for prediction and lacks the guidance of prior knowledge, so the degree of system intelligence needs to be further improved. Therefore, this paper intends to add a knowledge graph on the basis of the YOLOv5 algorithm to further enhance the intelligent level of the system.

The knowledge graph aims to describe the entities, attributes, and their relationships that exist in the real world. It is generally expressed in the form of triples, so it is an effective method to use the knowledge graph to store and represent the attribute information and interrelated information of the item itself. The multimodal knowledge graph enriches the information types in the knowledge graph by combining the semantic information in the triples and the image feature information in the image and improves the information density and is widely used in question answering systems [27], search and recommendation systems [28, 29], and other fields. Literature [30] uses YOLO9000 as the object recognition module, which can recognize 9000 object categories after training, and uses external knowledge graph to obtain background knowledge related to the object. Marino et al. [31] studied the application of structured prior knowledge in the form of knowledge graph in image classification. Liu et al. [32] proposed a collection of three knowledge graphs of MMKG (Multimodal Knowledge Graphs), including the digital features and images of all entities and the overall alignment between the knowledge graphs. Chen et al. [33] proposed an expression learning framework for knowledge embedding. The framework first builds a knowledge graph based on statistical "category-attribute" related information; then it uses a graph network to spread node information on the graph to learn its knowledge expression; finally it designs a gated network to embed the knowledge expression into the image feature learning process and guides the learning of the attributes associated with the feature. Jiang et al. [34] proposed a hybrid knowledge routing module to improve model performance. In order to solve the traditional methods that ignore the correlation between the training set and the test set category, Wang et al. [35] proposed the use of category semantic expression and knowledge graph to guide the information dissemination between categories and applied it to zero-sample learning. Chen et al. [36] introduced statistical target objects and the possible coexistence of prior knowledge to constrain the relationship prediction space, aiming at improving the accuracy of the model in less sample categories. Wang et al. [37] introduced the prior knowledge of the association between the characters in the scene and the surrounding objects and performed explicit reasoning based on knowledge. Wu et al. [38] proposed a visual question-and-answer method, which constructs a textual representation of the semantic content of an image and merges it with the textual information from the knowledge base, aiming at having a deeper understanding of the scene. Lu et al. [39] combined visual features and prior knowledge of language models to determine visual relationships and realized the detection of multiple visual relationships in a picture. For object attributes such as shape and color, Sun et al. [40] proposed a method to automatically extract visual concepts using similar text and visual collections.

In order to test the effectiveness of the method proposed in this paper, consider deploying the model to edge devices for experimental verification. To perform big data analysis and management in complex systems, edge computing, as a new paradigm, can sink cloud computing functions and services to network edge devices and provide real-time data analysis and intelligent processing nearby, thereby effectively solving the problems of network congestion and network delay caused by the transmission and processing of massive data. Different from the large-scale data processing center in cloud computing, the communication, computing, storage, and other resources of edge devices in mobile edge computing are relatively limited [41]. On the one hand, when the task demands of end users increase sharply, a large number of end users need to offload tasks to edge devices, which is prone to problems such as excessive task load and increased processing delay, resulting in the lack of timeliness of task processing; on the other hand, there is an unbalanced load distribution among devices, and it is prone to the problem that some edge devices are overloaded with tasks and other edge device resources are idle. To effectively cope with the above problems, multiple edge devices can coordinate to perform computing tasks to achieve load balancing among edge devices while ensuring the service requirements of end users. Therefore, multiedge device collaboration has become an inevitable trend. The latest research work considers the collaboration of multiple edge devices to perform computing tasks together. Literature [42, 43] uses matching strategies to formulate task offloading strategies among multiple end users and multiple edge devices. Literature [44] studies the problem of task offloading in dense deployment scenarios of edge devices. Through the alliance game theory among multiple edge devices, a cooperative alliance is formed to jointly perform the computing tasks of the end user.

Literature [45, 46] implements task offloading between edge devices through a distributed game method, with the goal of minimizing the overall execution delay of the task.

3. The Design of Garbage Sorting Model

3.1. System Architecture. This paper designs a complex system for garbage detection and classification based on visual scene understanding. The overall architecture of the system is shown in Figure 1. First of all, through the knowledge graph, the unified representation and storage of the multimodal item knowledge in the home environment is used to form a priori knowledge base; among them, the YOLOv5m-Attention detection algorithm recognizes and locates the two modalities of the item image and video in the scene to obtain the item entity category and location information and combines the prior knowledge base to form a visual scene understanding model YOLOv5m-Attention-KG (see Figure 2); secondly, cloud computing is used as a computing back-end to form collaborative computing with edge devices. Finally, home service robots are used as edge computing devices for experimental verification, supporting real-time data processing and analysis, and completing the task of garbage classification.

3.2. The Key Technology and Algorithm. In order to autonomously realize intelligent garbage classification on home service robots, this paper proposes a YOLOv5m-Attention-KG visual scene understanding model. The structure of the model is shown in Figure 2. First, according to the different modalities of the items in the home environment, different model processing is adopted, and the YOLOv5m-Attention detection algorithm is used to process the two modalities of video and image; use BLSTM-LCRF and PCNN-BLSTM-Attention proposed by Wang et al. [47] to extract entities and relationships from text modalities. The open source structured data collected from the Internet and the entity relationship extracted above form a knowledge triple. The knowledge graph finally constitutes a unified characterization and storage of the semantic description information, attribute information, and spatial location information of the items in the scene. The open source structured data collected from the Internet completes the extraction of item attributes and relationship information; then it forms a knowledge triple with the entity relationships extracted above; the final knowledge graph can uniformly represent and store the semantic description, attributes, and related information of the items in the scene. Secondly, when detecting and classifying items in the home environment, the YOLOv5m-Attention detection algorithm will perform real-time detection to obtain its location and category information and query the entity information with high semantic similarity to the category information in the knowledge graph, based on the returned attributes and related information to determine whether the item is garbage and what kind of garbage it is, and make further intelligent decisions.

3.2.1. Multimodal Knowledge Graph. With the continuous popularization of the Internet technology, information from different sources such as text, images, video, and audio jointly portrays the same or related content, presents complex, multilevel semantic relationships, and forms multimodal information. As shown in Figure 3, the multimodal knowledge graph is divided into three parts: information representation, knowledge processing, and knowledge update. Entity extraction is generally to automatically extract a list of entities from a multimodal sample. At present, there is no special study on the extraction method of multimodal attribute extraction. Generally, attributes are regarded as a kind of entity concept, and the same method is used as entity extraction. Relations in multimodal samples are divided into simultaneous relations and hierarchical relations. Generally, when extracting relations, the idea that general concepts appear more frequently than specific concepts is used to extract by calculating the statistical relationship between the text and image features of the entity. Knowledge inference of multimodal samples can use label propagation based on multimodal features. For example, Fang et al. [48] use similarity matrix and image similarity matrix for label propagation; factor graphs can also be used for derivation and learning. Because every step of the construction process of the multimodal knowledge graph requires all multimodal samples, if new samples are added, a comprehensive update is required. However, there are currently no more relevant papers on the multimodal knowledge graph. The knowledge graph contains a large amount of factual knowledge, which is generally represented by triples: $(h, r, t)h$ represents the head entity, t represents the tail entity, and r represents the relationship between the two entities. The input multimodal information knowledge is modeled as a collection of triples. In the knowledge graph, nodes are used to represent entities and edges are used to represent attributes or relationships. Thus, the entities and relationships in the real indoor scene can be formed into a huge picture of the semantic network. Figure 4 is a case of knowledge graph. For the same entity as a drink bottle, due to the integrity of the shape, the attributes of the material information, and the relationship with other entities, it can be judged whether it is recyclable garbage.

3.2.2. Improved YOLOv5m-Attention Algorithm Design

3.2.2.1. Network Structure. YOLOv5 is divided into 4 models, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, according to the depth of the network and the width of the feature map. In this paper, considering the accuracy and speed, the YOLOv5m network is selected as the model for item detection and classification. YOLOv5m still uses the overall layout of v3 and v4 and divides the entire network structure into four parts: Input, Backbone, Neck, and Output. The difference from the original network is that the ESA attention mechanism is added after the Cross Stage Partial Networks (CSPNet), as shown in the highlighted module in Figure 5. Input terminal: adaptively zoom the picture, adopt Mosaic data enhancement method, enrich the data, improve the recognition ability of small objects, and automatically calculate the best anchor frame value of the data set.

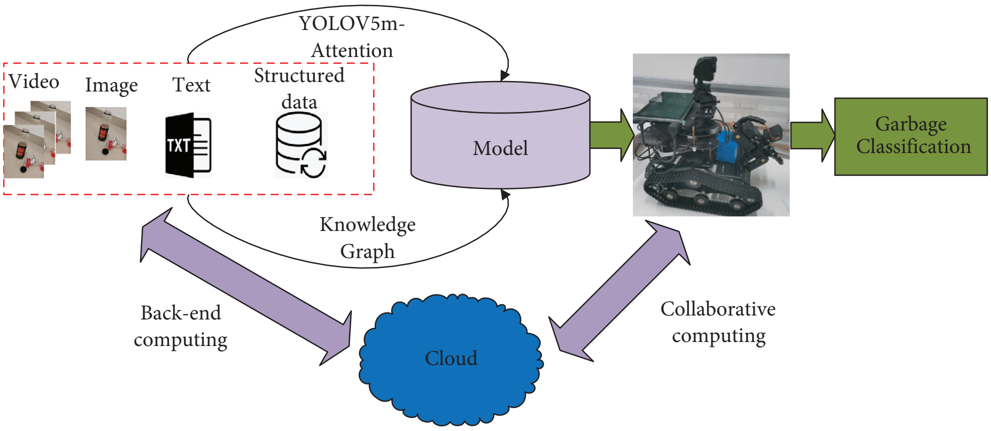


Figure 1: Overall architecture diagram.

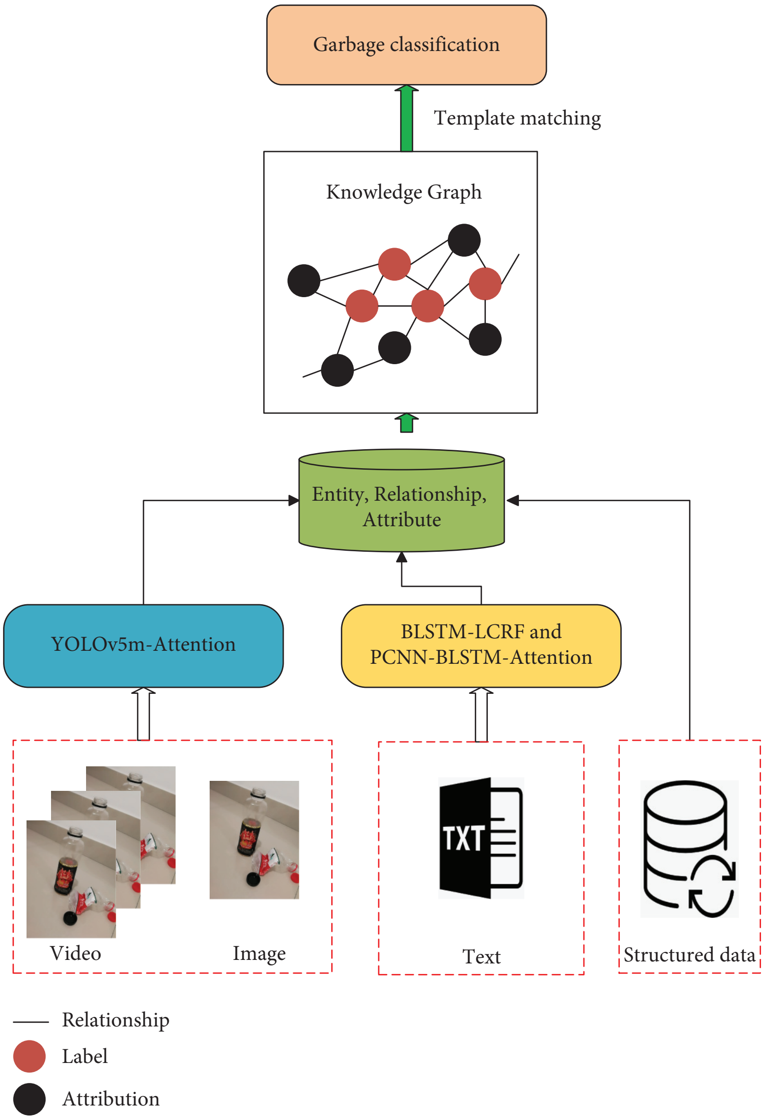


Figure 2: Visual scene understanding model.

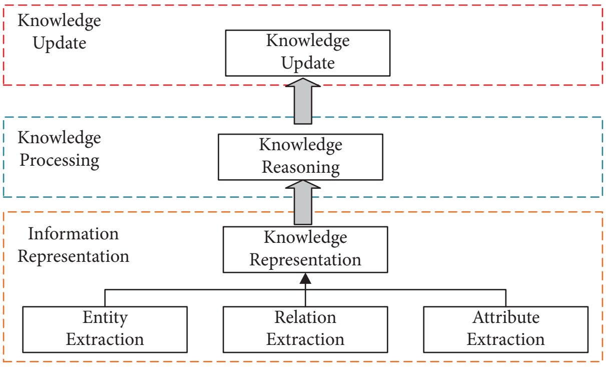


Figure 3: Construction of multimodal knowledge graph.

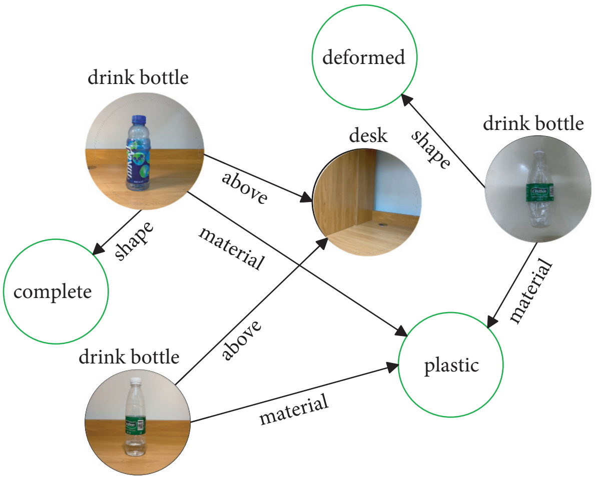


Figure 4: Knowledge graph case.

Backbone contains the Focus structure and improved CSPNet. The Focus structure includes 4 slice operations and 1 convolution operation with 32 convolution kernels, turning the original $608 \times 608 \times 3$ image into a $304 \times 304 \times 32$ feature map. CSPNet imitates the idea of dense cross-layer connection of Densenet, performs partial cross-layer fusion, and uses the feature information of different layers to obtain a richer feature map. In the figure, $n = 1$ or 2, X takes 1 or 3, which represents X residual components Res unit, a total of $n * X$ residual components Res unit. The ESA module (see Figure 6) calculates the weight information of the feature map on the channel position and spatial position and makes the network focus on the feature regions that are beneficial to classification according to the weight distribution and suppresses the background and other secondary information. Neck contains Path Aggregation Network (PANet) and Space Pyramid Pooling (SPP) modules. PANet aggregates high-level feature information with the output features of different layers of CSP modules from top to bottom and then aggregates shallow features through a bottom-up path aggregation structure, thereby fully fusing image features of different layers. The SPP module first uses 4 cores of different sizes to perform the maximum pooling operation and then performs tensor splicing. Output layer: In this paper, between GIOU Loss [49] and CIOU Loss [50], CIOU Loss with a slightly better effect is finally selected as the loss function of the prediction box regression. Because CIOU Loss considers the scale information of the bounding box aspect ratio compared to GIOU Loss and measures it from the three angles of overlap area, center point distance, and aspect ratio, it makes the prediction box regression better.

Drawing lessons from the ideas of CBAM [51] and ECA attention mechanism [52], the ESA attention block first obtains the channel and spatial attention weight maps according to the input feature map of the model; then it, respectively, multiplies it with the original feature map to obtain the space and channel feature maps with weights; finally, the channel and space feature maps are added in parallel to obtain a feature map with attention weights. The ESA attention structure is shown in Figure 6.

The difference from the CBAM attention mechanism is that the channel attention CAM in the ESA attention mechanism borrows the ECA attention mechanism. After global average pooling of the input features, it does not change the dimension of the channel and uses the size k Fast one-dimensional convolution to capture the local cross-channel feature information of each channel, replacing the multilayer MLP block in the channel attention mechanism in CBAM, avoiding the problem of reduced attention to the channel caused by dimensionality

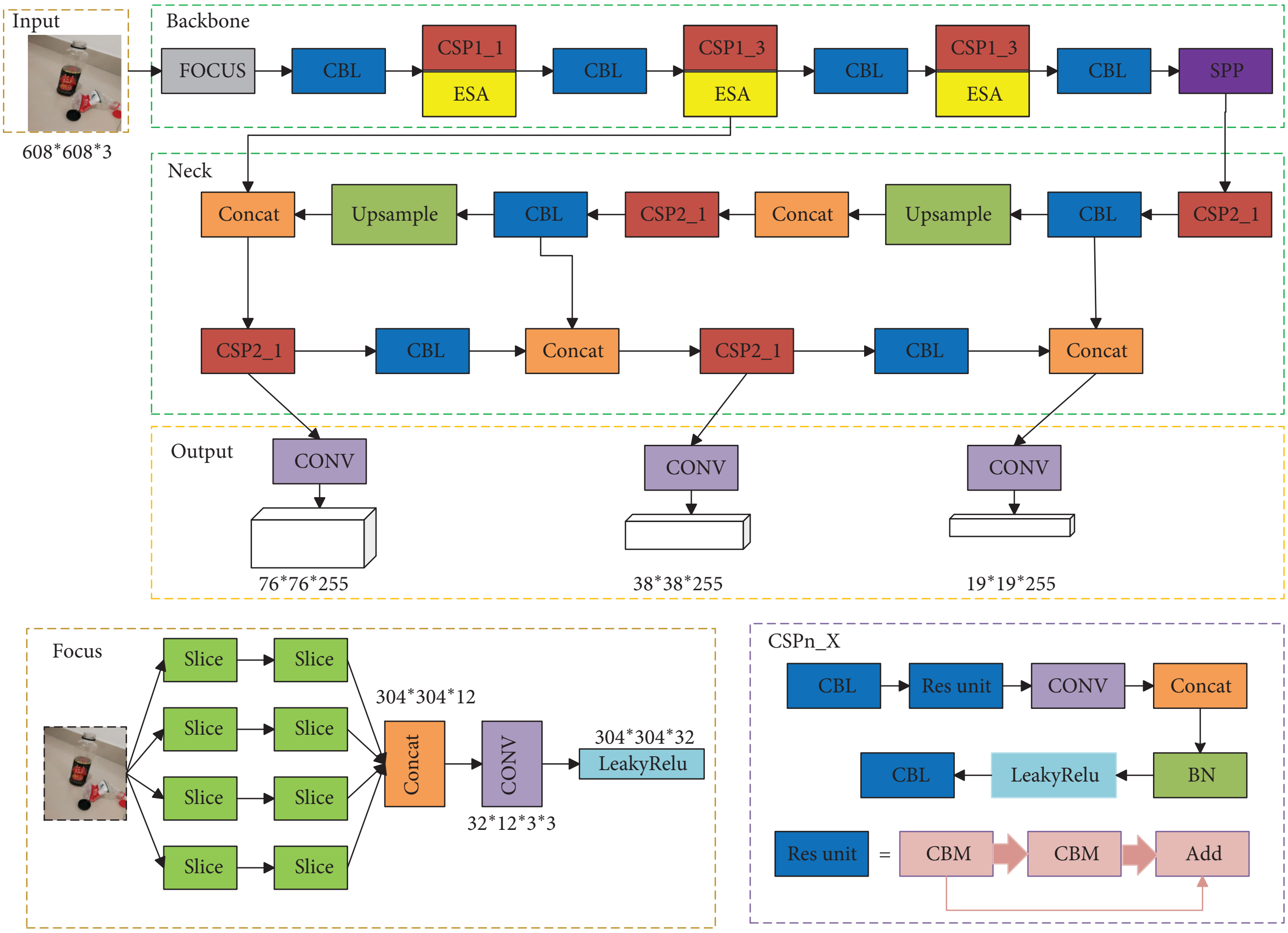


Figure 5: YOLOv5m-Attention network structure diagram.

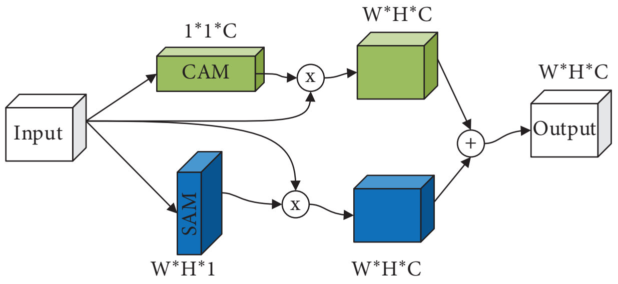


Figure 6: ESA attention structure.

reduction in MLP, and at the same time significantly reducing the complexity of the model. The spatial attention SAM in the ESA attention mechanism performs global average pooling and maximum pooling operations on the pixel values of the same position on the input feature map at the spatial position and obtains two spatial attention weights, respectively. They are merged into a 2-channel feature map in the channel dimension. Then use the convolutional layer composed of $3 * 3$ convolution kernel to compress the channel to 1, get the feature map size $W * H * 1$, and finally activate it through the Sigmoid function to obtain the spatial attention block.

3.2.2.2 Loss function. The model loss function is composed of classification loss, localization loss, and object confidence loss. YOLOv5 uses binary cross entropy loss to calculate the loss of category probability and object confidence score. Through experiments, the loss function CIOU_Loss is shown in formula (1).

```
Input: Self-made object image data set and markup files
Initialization parameters: epoch, learning rate, batch size, input size, network model configuration yaml file, IOU, yolov5m.pt pre-training weights
Image preprocessing: image brightness, image contrast, image saturation, Mosaic
(1) Prepare data, make data set, and divide training set and validation set.
(2) Load data configuration information and initialization parameters, input data, and preprocess it.
(3) Load the network model, and perform feature extraction and object positioning and classification on the input image.
(4) As the number of epochs increases, use SGD to update and optimize each set of parameters in the network.
(5) If the current epoch is not the last round, the MAP of the current model is calculated in the validation set. If the calculated model
    performance is better, the best model is updated and stored.
(6) After training the set number of epochs, obtain the trained optimal performance model and the most recently trained model.
Output: The best-performing detection model in this training.
```

Algorithm 1: Model training algorithm.

$$\text{CIOU\_Loss} = 1 - \left\{ \text{IOU} - \frac{d_1^2}{d_2^2} - \beta\alpha \right\}. \tag{1}$$

Among $\beta = (\alpha/(1 - IOU) + \alpha)$, $\alpha = (4/\pi^2) = (\tan^{-1}(W^{gt}/h^{gt}) - \tan^{-1}(W/h))^2$, d_1 represents the Euclidean distance between the two center points of the prediction box and the object box, and $d_2$represents the diagonal distance of the smallest bounding rectangle. (W^{gt}/h^{gt}) and (W/h), respectively, represent the respective aspect ratios of the object frame and the prediction frame.

3.2.2.3 Network training. The overall training process of the YOLOv5m network (Algorithm 1):

Some network parameter descriptions are shown in Table 1.

Table 1: Network parameter description table.

Parameter name	Parameter values
Learning rate	0.001
Momentum	0.9
Decay	0.0005
Batch size	64

4. Experiment

4.1. Experimental Configuration. The experiment in this paper is built on the Windows environment. CUDA is a general parallel computing architecture launched by NVIDIA. CUDNN is a GPU acceleration library for deep neural networks. The data is trained through the cooperation of the two. The experimental configuration is shown in Table 2.

4.2. Data Collection. The data set used in this paper has a total of 15,000 domestic garbage pictures, most of which come from the data set in the garbage classification competition held by Ali Yun Tianchi and some pictures of domestic garbage collected by the author. The data set can be divided into four categories in general, namely, recyclable trash, food trash, harmful trash, and other trash. Each category contains multiple objects. Among them are recyclable trash: power bank, bag, wash supplies, plastic toy, plastic utensils, plastic hangers, glassware, metalware, courier bags, plug wire, old clothes, ring-pull can, pillow, plush toy, shoes, cutting board, carton, wine bottle, metal food can, ironware, wok, edible oil drum, drink bottle, and paper books; harmful trash: dry battery, Unguentum, and expired drugs; other trash: disposable snack box, stained plastic, but, toothpick, flowerpot, chinaware, chopsticks, and stained paper; 10% of each category selects a total of 1500 images as the validation set, and the remaining 13,500 images are used as the training set. Use the LabelImage tool to label the training set, and generate the corresponding xml file for training. Figure 7 shows a visual display of the data set. Figure 7 shows a visual display of the data set. The left picture is the label distribution map of the data set. The sample distribution of various items can be clearly seen. There are many samples of small and large targets; the right picture is the distribution of data correlation maps.

4.3. Experimental Indicators. The evaluation criteria of the results of this experiment are mainly Precision (P), Recall (R), Mean Average Precision (MAP), and detection speed FPS. Among them, Precision represents the ratio of the real samples in the recognized positive samples.

P represents the ratio of the total number of predicted correct positive samples to the total number of actual positive samples in the prediction data set, as shown in formula (2):

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2}$$

where R represents the probability that the correct category in the sample is predicted to be correct, as shown in formula (3):

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

The MAP is determined by the precision rate P and the recall rate R. The curve with R as the horizontal axis and P as the vertical axis is referred to as the PR curve. The area under the PR curve is recorded as the AP value, as shown in formula (4), and the average of the average accuracy of all object categories is the MAP value, as shown in formula (5):

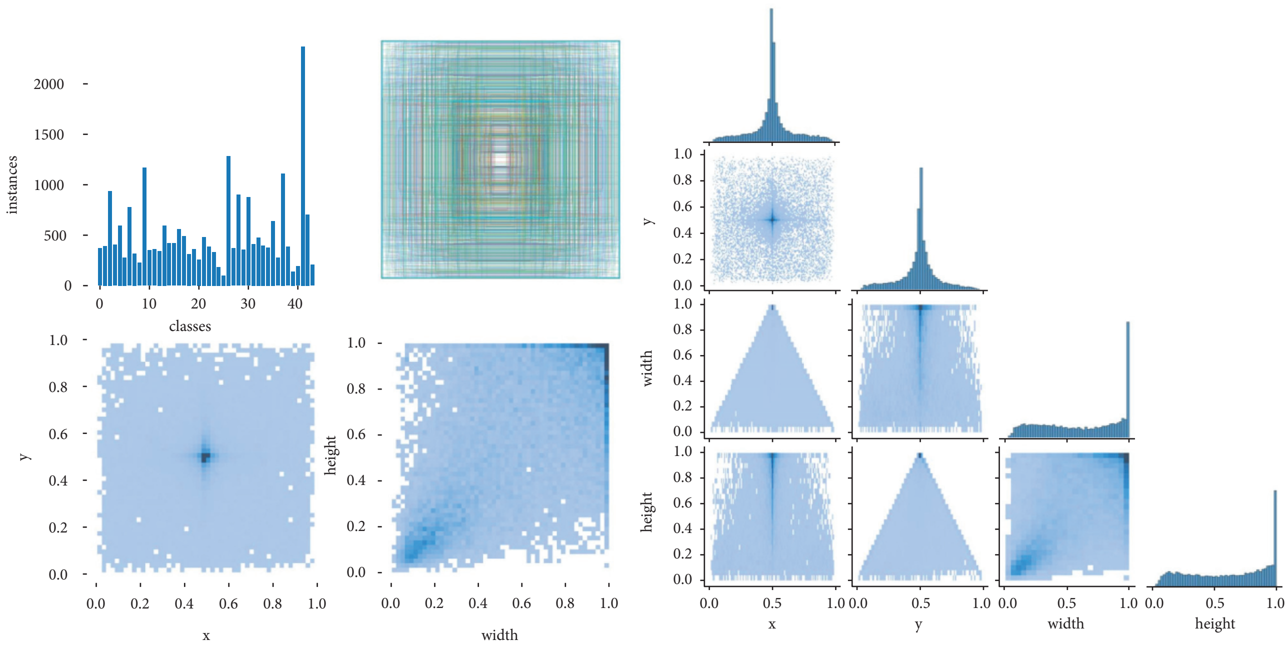


Figure 7: Data set visualization.

$$\mathrm{AP} = \int_0^1 P \mathrm{d}R, \tag{4}$$

$$\mathrm{MAP} = \frac{\sum_{i=1}^{N} \mathrm{AP}_i}{N}. \tag{5}$$

Among them, TP in formulae (2) and (3) indicates that the correct class is predicted as the number of correct categories, FP indicates that the negative category is predicted as the number of correct categories, and FN indicates that the correct category is predicted as the number of negative categories; in formula (5) N refers to the total number of detection object categories.

Table 2: Experiment environment configuration.

Project	Experimental environment
System	Windows
Programming environment	Pycharm
GPU	NVIDIATITAN RTX
Memory	24 GB
Pytorch version	Pytorch1.6
Python version	Python3.6
CUDA version	CUDA10.1
CUDNN version	CUDNN7.6
Data bases	Neo4j4.2.2
Java runtime environment	JDK15.0.1

4.4. Experimental Results and Analysis. The network parameters were trained and verified with different algorithms in accordance with Table 2 above, and the MAP and FPS were calculated as shown in Table 3. Figure 8 shows part of the evaluation indicators of the improved model in turn. The upper left is the training and verification loss function curve. You can intuitively see that, after 100 epochs, the loss reaches the lowest value and tends to balance; the upper right is the confusion matrix; the lower left is the PR curve; the bottom right is the $F1$ value curve.

From the results in Table 3, compared with the original YOLOv5 algorithm, the average accuracy rate of YOLOv5m-Attention-KG is increased by about 0.4% when the detection speed is equivalent. It also shows that the algorithm has a lower cost of additional propagation time in exchange for detection accuracy. Figure 9 is a partial visual comparison result of the original YOLOv5 and YOLOv5m-Attention-KG algorithms, where Figures 9(a)–9(k) are the detection results of the original YOLOv5 algorithm and Figures 9(b)–9(l) are the corresponding detection results of the YOLOv5m-Attention-KG algorithm. With improved algorithm from the comparison of Figures 9(a)–9(g) and 9(b)–9(h), it is obvious that the accuracy rate has improved; Figures 9(i)–9(k) and Figures 9(j)–9(l) show YOLOv5m-Attention-KG algorithm to increase the missed detection rate, and the accuracy rate has been improved.

Figure 10 shows the application of garbage classification. It is the same as the entity label of the drink bottle, because it has different attributes and the position relationship with other entity labels can make different decisions. The two entities of the drink bottle and the desk can obtain their entity labels through the recognition algorithm, and the entity label is used as a keyword to query in the neo4j graph database, and an intelligent decision will be made as to whether it is garbage. For example, the drink bottle in Figure 10(a) is placed on the floor, and because its shape is deformed and the material is plastic, it can be concluded that it is recyclable garbage, while the drink bottle in Figure 10(b) is placed on the desk and its shape is complete, so it cannot be determined to be garbage.

Table 3: Parameters of detection results of each algorithm.

Algorithm	MAP (%)	FPS
YOLOv5	72.8	30
YOLOv5m-Attention-KG	73.2	28

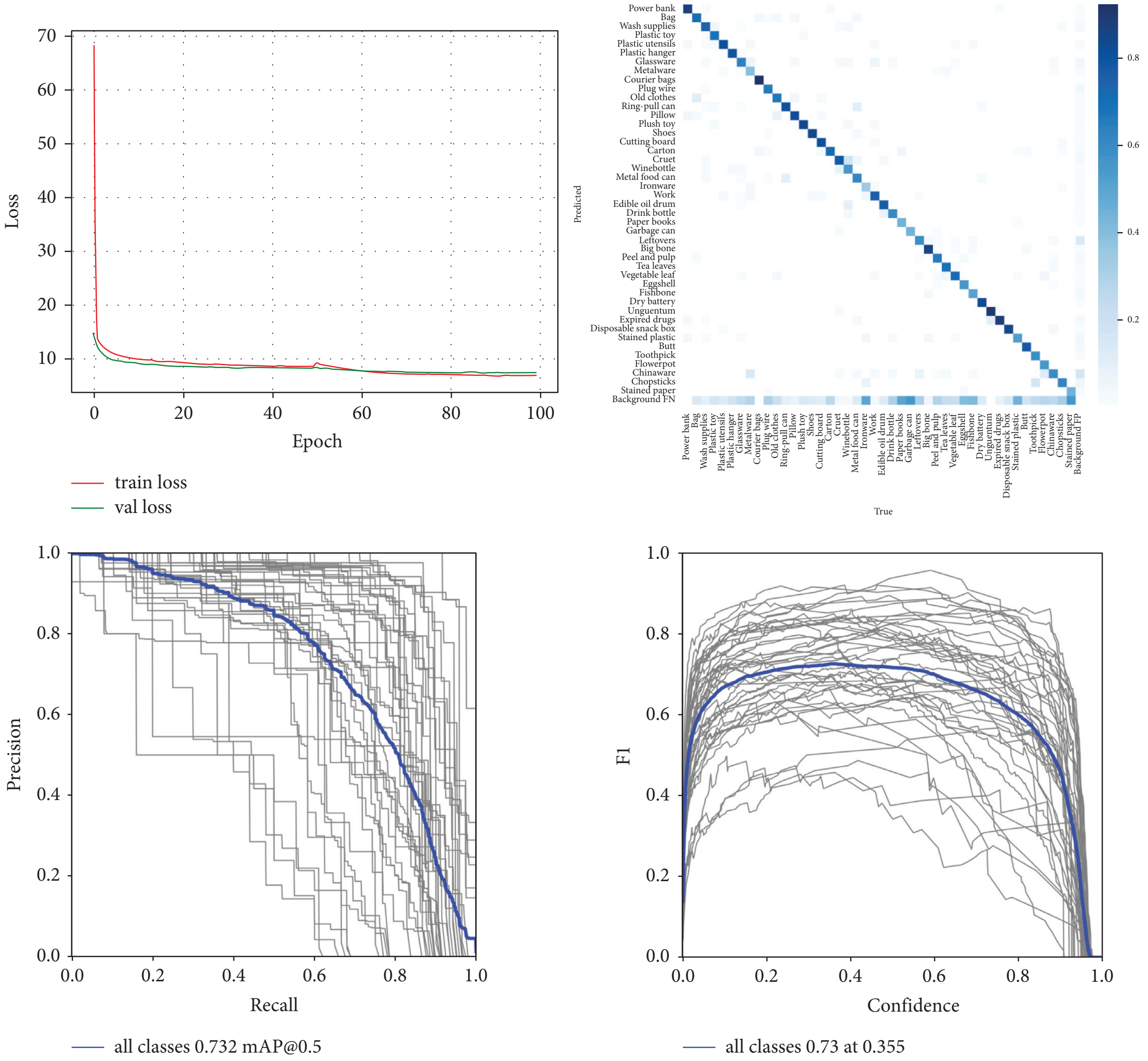
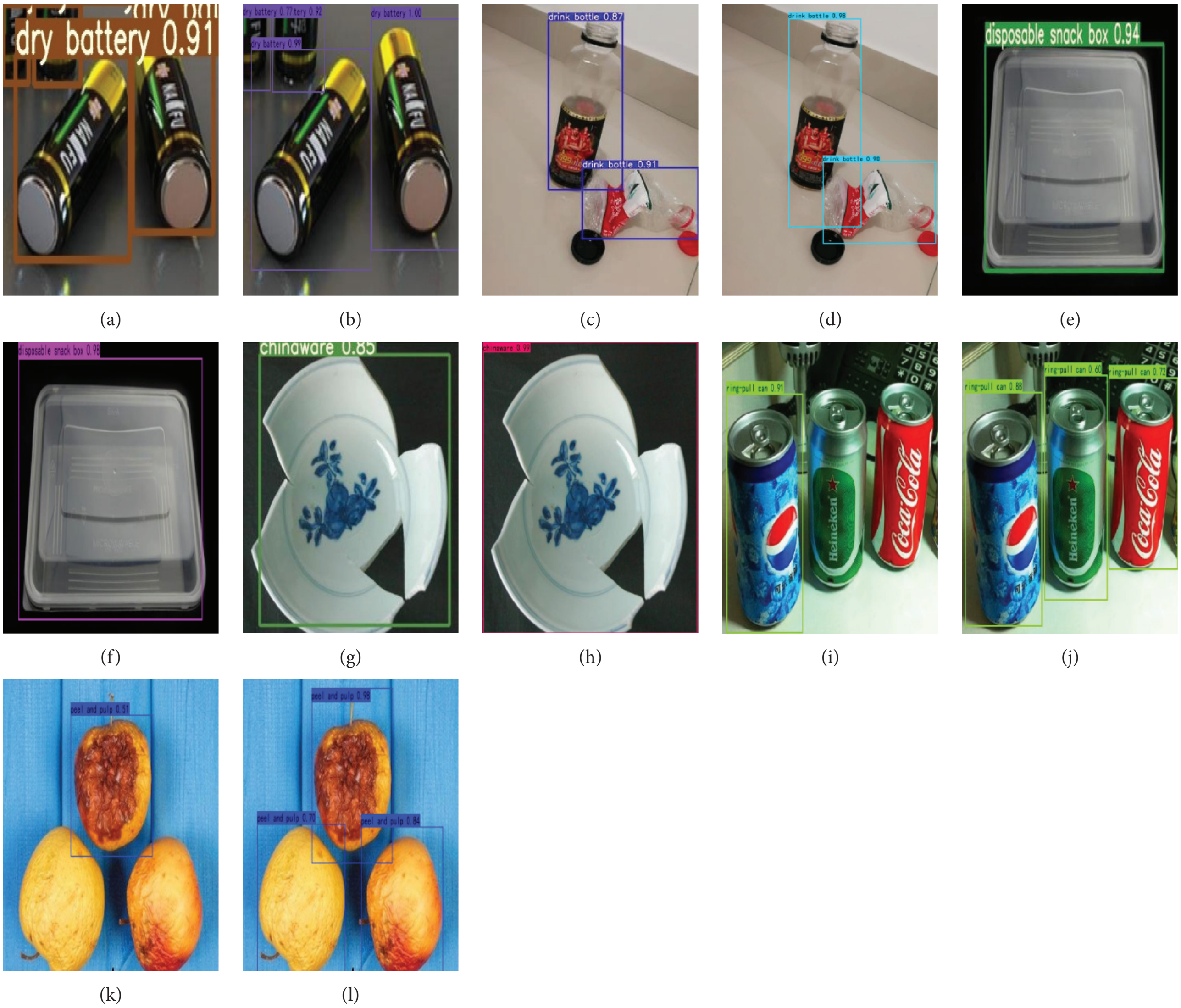


Figure 8: Part of the evaluation index.

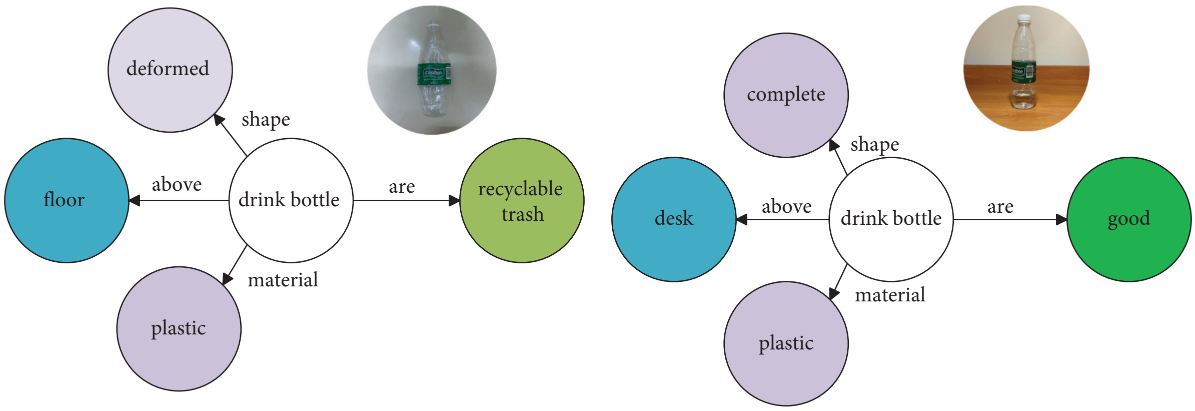


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l)

Figure 9: Comparison result graphs.



(a) (b)

Figure 10: Garbage classification.

5. Conclusions and Future Work

In order to autonomously complete intelligent garbage classification tasks on edge devices, this paper proposes a garbage detection and classification method based on visual scene understanding. Different from the existing method, the perceptual detection under the premise that the item is artificially defaulted to be garbage, this method uses knowledge graphs and visual algorithms to realize intelligent decision-making of items in the scene. Potential future research directions: First, the extraction of the attributes of the items in the scene and the associated information of other items requires further in-depth research; the second is that the system is now only real-time perception of two modal items of image and video, and it can go deep into voice modal in the future, through intelligent interaction with people, to improve the degree of intelligence of edge devices.

Data Availability

The data used to support the findings of this study are not applicable because the data interface cannot provide external access temporarily.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the National Key R&D Program of China (Grant nos. 2019YFE0122600, 2018YFB1700200, and 2019QY1604), National Natural Science Foundation of China (Grant no. U1836217), Hunan Provincial Key Research and Development Project of China (Grant no. 2019GK2133), Natural Science Foundation of Hunan Province (Grant nos. 2021JJ50050, 2021JJ50058, and 2020JJ6089), Scientific Research Project of Hunan Provincial Department of Education (Grant no. 19B147), Key Project of the Department of Education in Hunan Province (Grant no. 19A133), Scientific Research Project of China Packaging Federation (Grant no. 17ZBLWT001KT010), Open Platform Innovation Foundation of Hunan Provincial Education Department (Grant no. 20K046), and Special Fund Support Project for the Construction of Innovative Provinces in Hunan (2019GK4009).

References

[1] L. K. Zhou, H. Z. Liu, and Y. Li, "Summary for the key technologies and research status of the cleaning robot," *Mechanical Science and Technology for Aerospace Engineering*, vol. 33, no. 5, pp. 635–642, 2014.

[2] D. Q. Zhu and M. Z. Yan, "Survey on technology of mobile robot path planning," *Control and Decision*, vol. 25, no. 7, pp. 961–967, 2010.

[3] J. Y. Yang, L. Ma, and D. C. Bai, "Robot vision environmental perception method based on hybrid features," *Journal of Image and Graphics*, vol. 17, no. 1, pp. 114–122, 2012.

[4] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural baby talk," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7219–7228, Salt Lake City, UT, USA, June 2018.

[5] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: adaptive attention via a visual sentinel for image captioning," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3242–3250, Honolulu, HI, USA, July 2017.

[6] Z. Ren, X. Wang, N. Zhang, X. Lv, and L. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1151–1159, Honolulu, HI, USA, July 2017.

[7] Y. Zhang, C. Song, and D. Zhang, "Deep learning-based object detection improvement for tomato disease," *IEEE Access*, vol. 8, pp. 56607–56614, 2020.

[8] M. Algabri, H. Mathkour, M. A. Bencherif, M. Alsulaiman, and M. A. Mekhtiche, "Towards deep object detection techniques for phoneme recognition," *IEEE Access*, vol. 8, pp. 54663–54680, 2020.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[10] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.

[11] 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, 2016.

[12] J. Dai, Y. Li, and K. He, "R-FCN: object detection via region-based fully convolutional networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 379–387, Barcelona, Spain, December 2016.

[13] K. N. R. S. V. Prasad, K. B. D'souza, and V. K. Bhargava, "A downscaled faster-RCNN framework for signal detection and time-frequency localization in wideband RF systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4847–4862, 2020.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[15] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, Honolulu, HI, USA, July 2017.

[16] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[17] A. Bochkovskiy, C. Y. Wang, and H. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: single shot multiBox detector," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, October 2016.

[19] M. Y. Yang and G. Thung, "Classification of trash for recyclability status," *Computer Science*, pp. 1–6, 2016.

[20] W. L. Mao, W. C. Chen, C. T. Wang, and Y. H. Lin, "Recycling waste classification using optimized convolutional neural

network," *Resources, Conservation and Recycling*, vol. 164, 2021.

[21] M. Toaar, B. Ergen, and Z. Cmert, "Waste classification using AutoEncoder network with integrated feature selection method in convolutional neural network models," *Measurement*, vol. 153, 2019.

[22] P. Zhang, Q. Zhao, J. Gao, W. Li, and J. Lu, "Urban street cleanliness assessment using mobile edge computing and deep learning," *IEEE Access*, vol. 7, pp. 63550–63563, 2019.

[23] A. Seredkin, M. Tokarev, and I. Plohih, "Development of a method of detection and classification of waste objects on a conveyor for a robotic sorting system," *Journal of Physics: Conference Series*, vol. 1359, 2019.

[24] C. Zhihong, Z. Hebin, W. Yanbo, L. Binyan, and L. Yu, "A vision-based robotic grasping system using deep learning for garbage sorting," in *Proceedings of the 2017 36th Chinese Control Conference (CCC)*, pp. 11223–11226, Dalian, China, July 2017.

[25] T. Abeywickrama, M. A. Cheema, and D. Taniar, "k-nearest neighbors on road networks," *Proceedings of the VLDB Endowment*, vol. 9, no. 6, pp. 492–503, 2016.

[26] K. Mikami, Y. Chen, J. Nakazawa, Y. Iida, Y. Kishimoto, and Y. Oya, "Deep counter: using deep learning to count garbage bags," in *Proceedings of the 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 1–10, Hakodate, Japan, August 2018.

[27] L. Bauer, Y. Wang, and M. Bansal, "Commonsense for generative multi-hop question answering tasks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4220–4230, Brussels, Belgium, 2018.

[28] H. W. Wang, M. Zhao, X. Xing, W. J. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proceedings of the World Wide Web Conference*, San Francisco, CA, USA, March 2019.

[29] Y. Z. Wu, Q. Liu, R. Chen, C. Y. Li, and Z. R. Peng, "A group recommendation system of network document resource based on knowledge graph and LSTM in edge computing," *Security and Communication Networks*, vol. 2020, Article ID 8843803, 11 pages, 2020.

[30] Y. Zhou, Y. W. Sun, and V. Honavar, "Improving image captioning by leveraging knowledgegraphs," in *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 283–293, Waikoloa Village, HI, USA, January 2019.

[31] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: using knowledge graphs for image classification," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20–28, Honolulu, HI, USA, July 2017.

[32] Y. Liu, H. Li, and A. Garcia-Duran, "MMKG: multi-modal knowledge graphs," 2019, https://arxiv.org/abs/1903.05485.

[33] T. Chen, L. Lin, R. Chen, Y. Wu, and X. Luo, "Knowledge-embedded representation learning for fine-grained image recognition," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI*, Stockholm, Sweden, July 2018.

[34] C. Jiang, H. Xu, X. Liang, and L. Lin, "Hybrid knowledge routed modules for large-scale object detection," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1552–1563, Montréal, Canada, December 2018.

[35] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6857–6866, Salt Lake City, UT, USA, June 2018.

[36] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6156–6164, Long Beach, CA, USA, June 2019.

[37] Z. Wang, T. Chen, J. Ren, W. Yu, H. Cheng, and L. Lin, "Deep reasoning with knowledge graph for social relationship understanding," 2018, https://arxiv.org/abs/1807.00504.

[38] Q. Wu, P. Wang, C. Shen, A. Dick, and A. Van Den Hengel, "Ask me anything: free-form visual question answering based on knowledge from external sources," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4622–4630, Las Vegas, NV, USA, June 2016.

[39] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *Proceedings of the 14th European Conference on Computer Vision*, pp. 852–869, Amsterdam, The Netherlands, October 2016.

[40] C. Sun, C. Gan, and R. Nevatia, "Automatic concept discovery from parallel text and visual corpora," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2596–2604, Santiago, Chile, December 2015.

[41] Q. Zhang, S. Sun, M. Liu, Z. Li, and Z. Zhang, "Online joint optimization mechanism of task offloading and service caching for multi-edge device collaboration," *Journal of Computer Research and Development*, vol. 58, no. 6, pp. 1318–1339, 2021.

[42] Q. V. Pham, T. LeAnh, H. Nguyen, and C. Hong, "Decentralized computation offloading and resource allocation in heterogeneous metworks with mobile edge computing," 2018, https://arxiv.org/abs/1803.00683.

[43] X. H. Shen, Y. Z. Wu, S. H. Chen, and X. M. Luo, "An intelligent garbage sorting system based on edge computing and visual understanding of social internet of vehicles," *Mobile Information Systems*, vol. 2021, Article ID 5231092, 12 pages, 2021.

[44] L. Chen and J. Xu, "Socially trusted collaborative edge computing in ultra dense networks," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, San Jose, CA, USA, October 2017.

[45] L. Chen, J. Xu, and S. Zhou, "Computation peer offloading in mobile edge computing with energy budgets," in *Proceedings of the 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, December 2017.

[46] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1619–1632, 2018.

[47] H. Wang, W. Q. Zhu, Y. Z. Wu, P. J. He, and L. J. Wan, "Named entity recognition based on equipment and fault field of CNC machine tools," *Journal of Engineering Science*, vol. 42, no. 4, pp. 476–482, 2020.

[48] Q. Fang, C. Xu, J. Sang, M. S. Hossain, and A. Ghoneim, "Folksonomy-based visual ontology construction and its applications," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 702–713, 2016.

[49] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, Long Beach, CA, USA, June 2019.

[50] Z. Zheng, P. Wang, W. Liu, and D. Ren, "Distance-IoU loss: faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12993–13000, 2020.

[51] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Proceedings of the ECCV*, Munich, Germany, September 2018.

[52] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: efficient channel attention for deep convolutional neural networks," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11531–11539, Seattle, WA, USA, June 2020.

Hindawi
Complexity
Volume 2021, Article ID 7937922, 7 pages
https://doi.org/10.1155/2021/7937922

Research Article

Cooperative Cloud-Edge Feature Extraction Architecture for Mobile Image Retrieval

Chao He [1] and Gang Ma [2]

[1] *State Key Laboratory of Network and Switching Technology Institute, Beijing University of Posts and Telecommunications, Beijing, China*
[2] *School of Mathematics and Statistics, Shandong University of Technology, Zibo, Shandong, China*

Correspondence should be addressed to Chao He; chaohe@bupt.edu.cn

Received 8 September 2021; Revised 14 October 2021; Accepted 18 October 2021; Published 1 November 2021

Academic Editor: Xuyun Zhang

Mobile image retrieval greatly facilitates our lives and works by providing various retrieval services. The existing mobile image retrieval scheme is based on mobile cloud-edge computing architecture. That is, user equipment captures images and uploads the captured image data to the edge server. After preprocessing these captured image data and extracting features from these image data, the edge server uploads the extracted features to the cloud server. However, the feature extraction on the cloud server is noncooperative with the feature extraction on the edge server which cannot extract features effectively and has a lower image retrieval accuracy. For this, we propose a collaborative cloud-edge feature extraction architecture for mobile image retrieval. The cloud server generates the projection matrix from the image data set with a feature extraction algorithm, and the edge server extracts the feature from the uploaded image with the projection matrix. That is, the cloud server guides the edge server to perform feature extraction. This architecture can effectively extract the image data on the edge server, reduce network load, and save bandwidth. The experimental results indicate that this scheme can upload few features to get high retrieval accuracy and reduce the feature matching time by about 69.5% with similar retrieval accuracy.

1. Introduction

Mobile image retrieval plays an important role in processes such as identification of crop diseases and insect pests, the protection of pedestrians in autonomous vehicles, suspect identification, and medical services [1–6]. It has penetrated into all aspects of people's lives. Feature extraction and feature matching are two important factors in image retrieval tasks. Feature matching is the most time-consuming, and feature extraction affects the matching time and retrieval results. As a result of the limited computing and storage resources of user equipment, many mobile image retrieval tasks are based on mobile cloud computing architecture [7–11]. For instance, Shelly et al. [12] proposed a cloud computing-based iris retrieval solution based on the Hadoop framework and proved that the use of cloud servers can effectively accelerate retrieval tasks. Hassan et al. [13] proposed a face retrieval method based on mobile cloud computing architecture. In this framework, the mobile device performs a lightweight task and the cloud server runs the computationally intensive tasks. In these solutions, the user equipment performs a lightweight task and the cloud server runs the computationally intensive tasks with powerful computing and storage resources. The above methods all use the powerful computing and storage resources of cloud servers. That is, mobile users first use their user equipment to capture images and then upload the captured image data to the cloud server for further processing. After obtaining the uploaded image data, the cloud server processes it and gets the result and sends it to the mobile users. To reduce the network traffic, many studies [14, 15] have also preprocessed these captured image data, and even extracted features from these image data, and only uploaded the extracted features to the cloud server. Kan et al. [14]

presented an automatic classification method for medicinal plant leaves instead of manual classification, which can greatly improve the accuracy and speed of retrieval. This method preprocesses the image of the leaves of medicinal plants, extracts the shape and texture features, and then classifies the leaves of the medicinal plants through a support vector machine (SVM) classifier. Mannan et al. [15] proposed an enhanced cloud-based biometric identification method for identifying individuals on campus. By using the computing and storage resources of the cloud server, the system's computing and storage burdens are reduced. This method uses the local binary pattern (LBP) algorithm to extract features of encrypted images to protect personal privacy. At the same time, it uses the principal component analysis algorithm to reduce transmission delay and calculation time.

Mobile image retrieval greatly facilitates our lives by providing various retrieval services. However, as a result of the long distance between mobile users and the cloud server, it is difficult to provide a fast response to massive mobile image retrieval tasks, and with limited coverage and spectrum resources, mobile users are usually far away from cloud servers, leading to network delays and interruptions, which will bring a poor user experience. Since fast response is a great demand for mobile image retrieval tasks, a new computing paradigm multiaccess edge computing (MEC) [16–20] architecture has emerged as a promising solution to address the long response time of mobile cloud architecture by providing computing and storage resources at the edge of the network. In the MEC environment, feature extraction is mainly performed on edge servers and the time-consuming operation is performed on the cloud server. In recent years, many related studies have been proposed [21, 22]. For instance, Soyata et al. [21] presented a hybrid mobile-cloudlet-cloud computing architecture that uses the cloudlet as a lightweight server and applies an optimal task-partitioning method for distributing computing load among cloud servers. Hu et al. [22] proposed a face retrieval framework based on fog computing architecture. In the proposed framework, fog nodes perform face detection, image preprocessing, feature extraction, and face identifier from the raw image transmitted by the client with a local binary pattern algorithm. Then, the cloud server performs face matching and identity information acquisition after receiving the identifier from fog nodes. Results show that the proposed framework can reduce bandwidth consumption and response time.

However, in most of the existing methods, the feature extraction on the edge server is separated from the cloud servers. This leads to the lack of effectiveness of the extracted features, which affects the performance of image retrieval tasks. Sharma et al. [23] presented a coordinated architecture for edge and cloud computing that can analyse big data effectively in the Internet of Things networks. The key point is that the cloud server utilizes the network knowledge and historical information to guide the edge server to provide various customized services. To this end, we aim to study mobile image retrieval in the MEC environment. And, we propose a cloud-edge collaboration feature extraction solution for mobile image retrieval in MEC. In the proposed framework, the effective features are extracted through the collaboration of the cloud server and edge servers, rather than through the cloud server or the edge servers alone. Meanwhile, we store the extracted features and results on the edge server to respond to the same image retrieval service faster.

The rest of this paper is organized as follows: Section 2 presents the system framework. Section 3 presents the experimental results and analysis. Finally, we conclude this paper in Section 4.

2. Design of the Proposed Architecture

2.1. Problem Statement. In this paper, we study the problem of image retrieval in the MEC environment, which is described as follows: As illustrated in Figure 1, the system architecture of MEC consists of three layers of components: user equipment, edge servers, and cloud servers. User equipment communicates with edge servers through a network gateway, and the edge servers connect to cloud servers via the Internet backbone. A large amount of labelled image data is stored on cloud servers. Mobile users first use their user equipment to capture images and preprocess them and then upload the preprocessed image data to edge servers. After receiving the preprocessed images, the edge servers use a feature extraction algorithm to extract effective features, store the features, and then upload the extracted features to the cloud server for further processing. After receiving the extracted features, the cloud server processes them and gets the results and finally returns the results to the edge servers and user equipment. The symbols used in this paper are summarized in Table 1.

2.2. Detailed Design of the Proposed Architecture. The architecture of the proposed framework consists of three layers of components: user equipment, edge servers, and cloud servers. User equipment refers to some devices with limited computing and storage resources, such as smartphones, laptops, and Apple watches with a mobile broadband adapter. Edge servers are usually a group of servers that are deployed at the edge of the network, such as microservers. Cloud servers are usually Alicloud servers, Amazon Web Service (AWS) Cloud servers, and Microsoft Azure Cloud servers.

In order to verify the feasibility and effectiveness of the architecture, we implemented a prototype system that uses a Karhunen–Loève transform (KLT) [24, 25] algorithm for feature extraction. In practical scenarios, our framework can flexibly select algorithms according to application scenarios, including but not limited to KLT algorithms. The proposed framework can be divided into offline stages and online stages. In the offline stage, we get the projection matrix A with a KLT algorithm from the image data set on the cloud server. We assume that x_i, $i = 1, \ldots, m$, is the i-th image in the image data set. The number of features of the image is d, and the number of class labels is K. The calculation process of KLT is as follows:

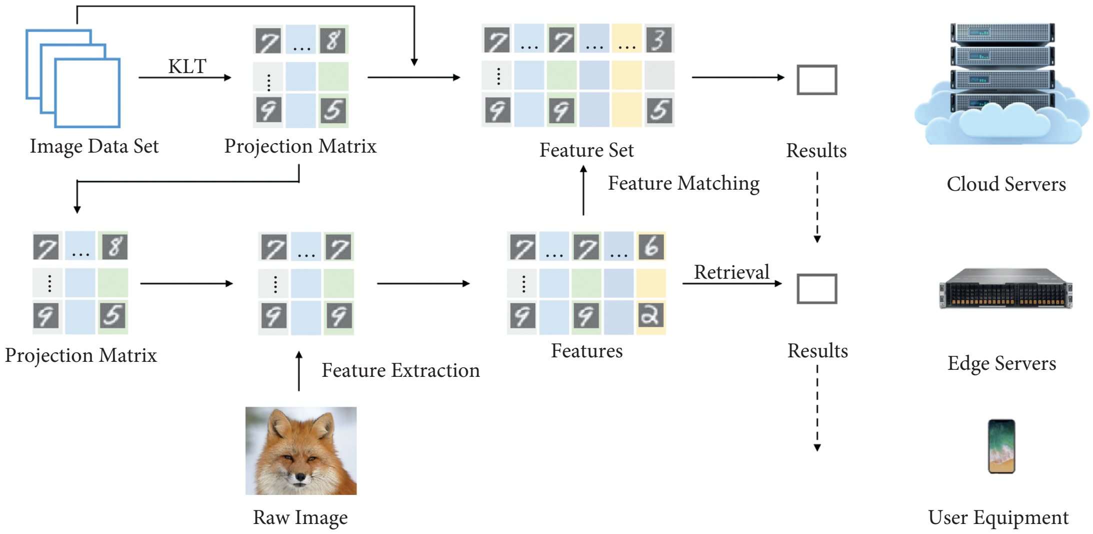


FIGURE 1: The detailed design of the proposed framework.

TABLE 1: Frequently used symbols.

Symbols	Descriptions
X	The pixel matrix of an image data set
x_i	The i-th image in the image data set
μ	The mean of the image
$x_i(j)$	The j-th feature of the i-th image
σ_j	The standard deviation of the j-th feature
C	The covariance matrix for the features in the image data set
A	The projection matrix
$\mathrm{d}(x_i, v_i)$	The Euclidean distance between two vectors x_i and v_i
R	The accuracy rate of the feature matching
$\varphi(a_i, b_i)$	Step function, equals 1 if $a_i = b_i$ and equals 0 otherwise
K	K-nearest neighbor

(1) Standardize the data set on the cloud server. Most machine learning and optimization algorithms perform better when all the features are along the same scale. To do this, a standardization approach can be implemented. Sample x_i can become the standardized feature by using the following calculation:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{1}$$

where μ is the mean of the sample

$$x_i = x_i - \mu. \tag{2}$$

Then, the sample x_i has zero mean:

$$\sigma_j = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(x_i(j)\right)^2}, \tag{3}$$

where σ_j is the standard deviation of the corresponding feature, $x_i(j)$, $j = 1, \ldots, d$, is the j-th feature, and finally, the standardized feature $x_i(j)$ is as follows:

$$x_i(j) = \frac{x_i(j)}{\sigma_j}. \tag{4}$$

(2) Calculate the covariance matrix for the features in the data set. The covariance matrix C is as follows:

$$\begin{aligned} \mathrm{C} &= \frac{1}{m}\sum_{i=1}^{m} x_i (x_i)^T, \\ &= \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,m} \end{pmatrix}, \\ &= \begin{pmatrix} \sigma_1^2 & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & \sigma_2^2 & \cdots & c_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & \sigma_m^2 \end{pmatrix}. \end{aligned} \tag{5}$$

(3) Calculate the eigenvalues and eigenvectors for the covariance matrix.

(4) Sort eigenvalues and their corresponding eigenvectors.

(5) Form the projection matrix A by selecting the corresponding eigenvectors of the t largest eigenvalues.

$$\mathrm{A} = [p_1, \ldots, p_t]. \tag{6}$$

The feature set on the cloud server can also be calculated by $X \cdot A$. Then, the projection matrix A is transmitted to the edge server for feature extraction from the raw image uploaded by the user equipment. Matrix A gets the whole information of the image data set on the cloud server; therefore, the feature extraction from the raw image on the edge server is effective and has fewer features.

We use the K-nearest neighbor (KNN) [26, 27] algorithm to match features. KNN assumes the similarity between the new sample and available cases and puts the new case into the category that is most similar to the available categories. The processing flow of KNN is that we first choose the number of *K*, where *K* represents the number of neighbors. Then, we measure the distance of the *K*-nearest neighbors of the test data. Followed by that, we count the number of neighbors of each category. Finally, we assign the test data to the category with the most neighbors. Note that we use the most used Euclidean distance to calculate the distance between samples. Given two samples x_i and v_i, their Euclidean distance can be written as

$$\mathrm{d}(x_i, v_i) = \sqrt{\sum_{j=1}^{d}\left(x_{ij} - v_{ij}\right)^2}. \tag{7}$$

We also store the results of feature extraction and feature matching for saving computing resources when there are the same image retrieval requests. We assume that for the image x_i, a_i and b_i are the results of the feature matching. To quantify the performance of retrieval, the accuracy rate is defined as follows:

$$\mathrm{R} = \frac{1}{m}\sum_{\mathrm{i}}^{m}\varphi(a_i, b_i), \tag{8}$$

where $\varphi(a_i, b_i)$ equals 1 if $a_i = b_i$ and equals 0 otherwise. In the online stage, mobile users use their user equipment to capture images and upload the captured image data to the edge server. After receiving the image data, the edge server first performs object detection, using object segmentation algorithms to preprocess it. Then, the edge server uses the projection Matrix A to extract features from the preprocessed image data and uploads the extracted feature data to the cloud server for feature matching. Note that KNN is used to test the effectiveness of the extracted features. For convenience, we set K = 1.

The detailed cooperative cloud-edge process is given in Algorithm 1.

The novelty of our scheme is that the proposed framework uses the collaboration between the cloud servers and edge servers to extract efficient features to reduce feature matching time and get similar retrieval accuracy with fewer features, thus improving the user experience of mobile image retrieval applications.

3. Experiment

In this section, we use six data sets: ORL, YALE, UMIST, MNIST, COIL20, and LEAVES [28, 29] to verify our proposed architecture. Table 2 lists the details used in the experiment, and all the data sets are divided randomly into the image data set on the cloud server and multiple raw images on the edge server uploaded by user equipment mentioned in Section 2. Hence, the raw image is the image that has been preprocessed in our experiment. The MNIST data set is rescaled to 28×28 pixels, and the other image data sets are rescaled to 32×32 pixels.

We evaluate the proposed framework in terms of accuracy and feature matching time. Figure 2 shows the accuracy of the raw image and our method using different training samples. In Figure 2(a), in the beginning, the accuracy of using the original image method is higher than using our method. When the number of training images reaches 280, our method begins to outperform the original image method. At first, in Figure 2(b), the accuracy of our proposed method is not as good as the original image method, but when the number of training images increases to 83, the accuracy of the original image method starts to decline relative to our method. In Figure 2(c), the accuracy changes of the original image method and our method are similar to those of Figure 2(a) and Figure 2(c), but the accuracy of the two is more similar. Then, in Figure 2(d) and Figure 2(e), the accuracy of the original image method is better than of our method and the accuracy difference between the two is even higher than that of other data sets. In Figure 2(f), the accuracy of our method has always been better than that of the original image method, and after the number of training images is set to greater than 130, the accuracy of our method improves faster. In most cases, the accuracy of our method is similar to that of using raw images. Note that our method uses only a few features to participate in the retrieval tasks.

Our method achieves accuracy similar to that of using all features, which proves the effectiveness of the features extracted by our method. In addition, with the increase in training images, the accuracy of directly using the raw images and our method has increased, which proves the importance of collecting enough training images.

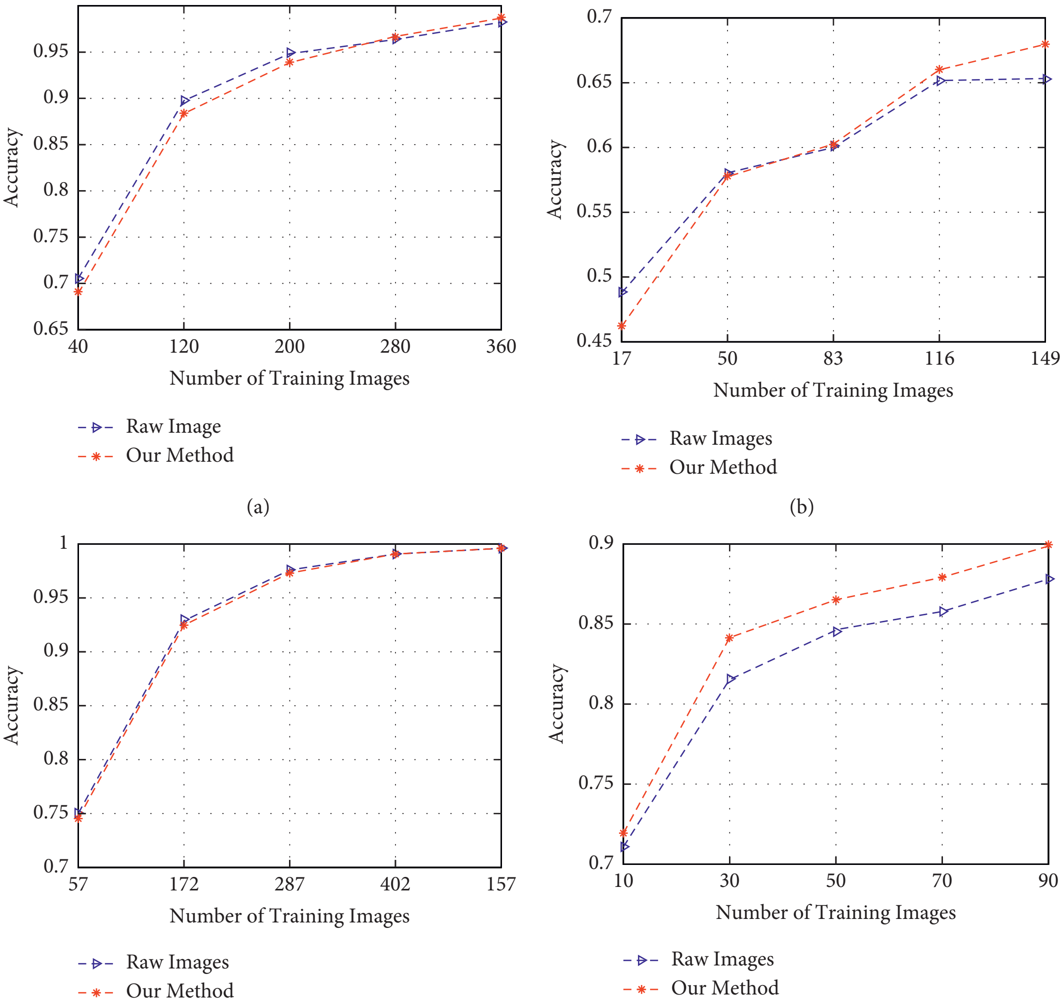
Our method has the least feature matching time. Figure 3 shows that the matching time of raw images and the matching time of the extracted features have similar accuracy. We observe that compared to using raw images, using the extracted features can save a lot of matching time. For instance, compared with using raw images, on the ORL data set, our method reduces the feature matching time by about 69.5% in the case of similar retrieval accuracy. This is because our method extracts a small number of features. Therefore, with the same number of images, fewer features result in less feature matching time. This also shows that using our method can save a lot of feature matching time. Moreover, on the COIL20 data set, the matching time between our method and the original image method has the largest change, and on the YALE data set, we have the smallest change in matching time between the original image method and our method. The size of matching time changes of other data sets is in the order of MNIST data set, UMIST data set, ORL data set, and LEAVES data set.

```
(i) Input: image data set, raw image
(ii) Output: result of image retrieval
(iii) Cloud gets the projection Matrix A with a KLT algorithm for the image data set storing on the cloud server
(iv) Cloud sends the projection Matrix A to the edge server
(v) Edge server gets the raw image uploading from the user equipment
(vi) Edge server preprocesses the raw image and gets the pixel Matrix $\mathbf{X}$, then extracts features by $\mathbf{X} \cdot \mathbf{A}$
(vii) Edge server sends the features of raw image to the cloud server for feature matching;
(viii) Cloud gets the result of image retrieval with a KNN algorithm
(ix) Cloud sends the result to the edge server, and the user equipment gets the image retrieval result sending by the edge server
```

Algorithm 1: Cooperative cloud-edge process.

Table 2: Description of benchmark data sets.

Data sets	Images	Dimensions	Classes
ORL	400	1024	40
YALE	165	1024	15
UMIST	564	1024	20
MNIST	70000	256	10
COIL20	1440	1024	20
LEAVES	186	1024	3



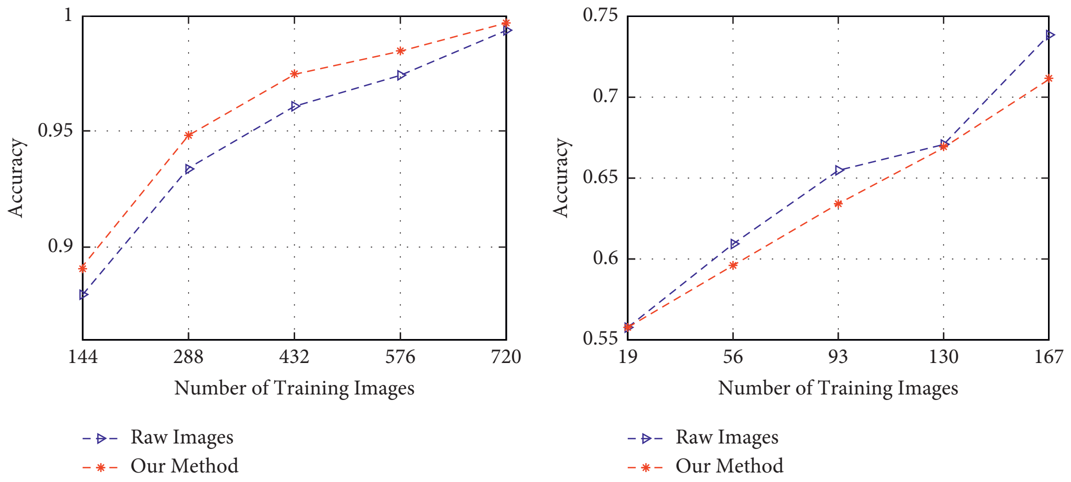
(a)

(b)

(c)

(d)

Figure 2: Continued.



(e)

(f)

FIGURE 2: Accuracy vs number of training images on (a) ORL, (b) YALE, (c) UMIST, (d) MNIST, (e) COIL20, and (f) LEAVES data sets.

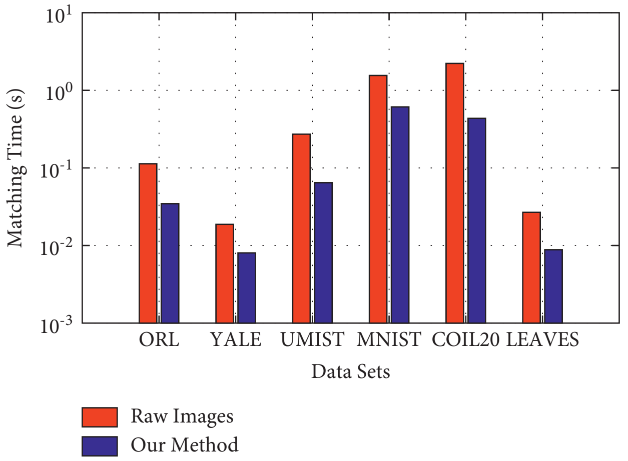


FIGURE 3: Matching time of the raw image vs matching time of our method.

4. Conclusions

In this article, the advantages of our architecture are the collaboration between cloud servers and edge servers. The cloud server performs the computing-intensive part of the image retrieval task, that is, projection matrix generation and feature matching. The edge server performs the lightweight part of the task, i.e., extracting features from the raw image with the projection matrix. Although the edge server can also perform feature extraction on the original image uploaded by user equipment, using our method can make the feature extraction of the original image more effective. The experiment also proved this, and the experiment shows that feature extraction is more effective as the number of images in the data set increases. Therefore, the projection matrix generated from the entire data set on the cloud server can better guide the original image feature extraction work on the edge server and make feature extraction more effective. Also, the accuracy of image retrieval and the matching time verify the effectiveness of our proposed architecture. Moreover, our proposed framework uses the KLT algorithm to extract features. In future work, the efficiency of the adopted algorithm for different data sets in different scenarios might prove important. This is an issue for future research to explore.

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 regarding the publication of this paper.

Acknowledgments

The authors would like to thank professorate senior engineer Hui Xu and Dr. Zhimi Cheng for their valuable comments on the paper. This research and publication were supported by Shandong Provincial Natural Science Foundation of China under Grant no. ZR2019MA012.

References

[1] X. Sun, S. Mu, Y. Xu, Z. Cao, and T. Su, “Image retrieval of tea leaf diseases based on convolutional neural network,” in *Proceedings of the 2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, Jinan, China, December 2018.

[2] H. Wang, B. Wang, B. Liu, X. Meng, and G. Yang, “Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle,” *Robotics and Autonomous Systems*, vol. 88, pp. 71–78, 2017.

[3] W.-J. Chang, L.-B. Chen, C.-H. Hsu, C.-P. Lin, and T.-C. Yang, “A deep learning-based intelligent medicine retrieval system for chronic patients,” *IEEE Access*, vol. 7, pp. 44 441–444 458, 2019.

[4] V. D. A. Kumar, V. D. A. Kumar, S. Malathi, K. Vengatesan, and M. Ramakrishnan, "Facial recognition system for suspect identification using a surveillance camera," *Pattern Recognition and Image Analysis*, vol. 28, no. 3, pp. 410–420, 2018.

[5] K. K. Singh and A. Singh, "Diagnosis of COVID-19 from chest X-ray images using wavelets-based depthwise convolution network," *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 84–93, 2021.

[6] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[7] J. Yang, N. Xiong, A. V. Vasilakos et al., "A fingerprint recognition scheme based on assembling invariant moments for cloud computing communications," *IEEE Systems Journal*, vol. 5, no. 4, pp. 574–583, 2011.

[8] P. Duan, W. Wang, W. Zhang, F. Gong, P. Zhang, and Y. Rao, "Food image retrieval using pervasive cloud computing," in *Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. IEEE*, pp. 1631–1637, Beijing, China, December 2013.

[9] W. Zhang, X. Chen, and J. Jiang, "A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2021.

[10] D. Kim, J. Son, D. Seo, Y. Kim, H. Kim, and J. T. Seo, "A novel transparent and auditable fog-assisted cloud storage with compensation mechanism," *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 28–43, 2020.

[11] X. Tan, J. Zhang, Y. Zhang, Z. Qin, Y. Ding, and X. Wang, "A PUF-based and cloud-assisted lightweight Authentication for multi-hop body area network," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 36–47, 2021.

[12] Shelly and R. Nallanthighal, "Iris retrieval on hadoop: aA biometrics system implementation on cloud computing," in *Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems. IEEE*, pp. 482–485, Beijing, China, September 2011.

[13] G. Hassan and K. Elgazzar, "The case of face recogni- tion on mobile devices," in *Proceedings of the 2016 IEEE wireless communications and networking conference. IEEE*, pp. 1–6, Doha, Qatar, April 2016.

[14] H. X. Kan, L. Jin, and F. L. Zhou, "Classification of medicinal plant leaf image based on multi-feature extraction," *Pattern Recognition and Image Analysis*, vol. 27, no. 3, pp. 581–587, 2017.

[15] J. Ma and Y. Yuan, "Dimension reduction of image deep feature using pca," *Journal of Visual Communication and Image Representation*, vol. 63, Article ID 102578, 2019.

[16] Q.-V. Pham, F. Fang, V. N. Ha et al., "A survey of multi-access edge computing in 5g and beyond: fundamentals, technology integration, and state-of-the-art," *IEEEAccess*, vol. 8, pp. 116974–117017, 2019.

[17] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[18] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2021.

[19] T. Zhao, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[20] Z. Xue and H. Wang, "Effective density-based clustering algorithms for incomplete data," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 183–194, 2021.

[21] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: real-time face retrieval using a mobile-cloudlet-cloud acceleration architecture," in *Proceedings of the 2012 IEEE symposium on computers and communications (ISCC). IEEE*, Cappadocia, Turkey, July 2012.

[22] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing based face identification and resolution scheme in internet of things," *IEEE transactions on industrial informatics*, vol. 13, no. 4, pp. 1910–1920, 2016.

[23] S. K. Sharma and X. Wang, "Live data analytics with collaborative edge and cloud processing in wireless iot networks," *IEEE Access*, vol. 5, pp. 4621–4635, 2017.

[24] K. Koutroumbas, *Sergios Theodoridis, Pattern Retrieval*, pp. 327–334, Academic Pressvol, Amsterdam, Netherlands, 2008.

[25] F. Kherif and A. Latypova, "Principal component analysis," *Machine Learning*, Elsevier, Amsterdam, Netherlands, pp. 209–225, 2020.

[26] G. Amato and F. Falchi, "Knn based image classification relying on local feature similarity," in *Proceedings of the Third International Conference on Similarity Searchand Applications*, pp. 101–108, Istanbul Turkey, September 2010.

[27] Z. Abu-Aisheh, R. Raveaux, and J.-Y. Ramel, "Efficient k-nearest neighbors search in graph space," *Pattern Recognition Letters*, vol. 134, pp. 77–86, 2020.

[28] M. Karasuyama and H. Mamitsuka, "Manifold-basedsimilarity adaptation for label propagation," *Advances in Neural Information Processing Systems*, vol. 26, pp. 1547–1555, 2013.

[29] D. Xu, W. Cheng, B. Zong et al., "Deep co-clustering," in *Proceedings of the 2019 SIAM International Confer-ence on Data Mining. SIAM*, pp. 414–422, Houston, TX, USA, March 2019.