Wireless Communications and Mobile Computing

Artificial Intelligence of Things (AIoT) Technologies and Applications

Lead Guest Editor: Tien-Wen Sung
Guest Editors: Pei-Wei Tsai, Tarek Gaber, and Chao-Yang Lee

Artificial Intelligence of Things (AIoT) Technologies and Applications

Wireless Communications and Mobile Computing

Artificial Intelligence of Things (AIoT) Technologies and Applications

Lead Guest Editor: Tien-Wen Sung
Guest Editors: Pei-Wei Tsai, Tarek Gaber, and
Chao-Yang Lee

Chief Editor

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9781271, 2 pages
https://doi.org/10.1155/2021/9781271

Editorial

Artificial Intelligence of Things (AIoT) Technologies and Applications

Tien-Wen Sung,[1] **Pei-Wei Tsai**,[2] **Tarek Gaber**,[3] **and Chao-Yang Lee**[4]

[1] *Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, China*
[2] *Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, Australia*
[3] *School of Science, Engineering & Environment, University of Salford, Manchester, UK*
[4] *Department of Aeronautical Engineering, National Formosa University, Yunlin, Taiwan*

Correspondence should be addressed to Tien-Wen Sung; tienwen.sung@gmail.com

Received 19 July 2021; Accepted 19 July 2021; Published 15 August 2021

AIoT (Artificial Intelligence of Things) is a relatively new term that has recently become a hot topic that combines two of the hottest acronyms, AI (Artificial Intelligence) and IoT (Internet of Things). IoT consists of interconnected things with built-in sensors and has the potential to generate or collect a vast amount of data. Individual IoT systems can be integrated into a large-scale system for various modern applications. With that comes a lot of collected or real-time data, an intelligent and efficient data processing is essential to make effective use of the information generated from these data. The data can be analyzed and utilized with AI for problem-solving or decision-making. Without AI, IoT would have limited value. AI can multiply the value of IoT; conversely, IoT can promote the learning and intelligence of AI. However, there are many challenges while deploying AIoT in practice. For instance, machine learning is one of the key technologies to be utilized in AIoT systems. Besides, there are many other issues such as complexity, efficiency, scalability, accuracy, and robustness related to the increasing modern AIoT systems and applications.

This Special Issue is aimed at publishing original and innovative research works that focus on challenging issues in the field of AIoT technology and applications. After the review process for evaluating all submitted manuscripts, there are nine papers accepted for publication in this special issue.

The paper titled "Semantic Integration of Sensor Knowledge on Artificial Internet of Things" by Y. Huang et al. describes the problem of data meaning matching in cooperations among heterogeneous sensor-based AIoT systems. The authors propose an ontology-based approach to deal with the semantic meaning matching problem and propose a compact Particle Swarm Optimization (cPSO) algorithm to improve the quality of ontology alignment in different sensor ontologies. The experiment results indicate that the proposed approach statistically outperforms other state-of-the-art sensor ontology matching techniques.

The paper titled "Quasiconformal Mapping Kernel Machine Learning-Based Intelligent Hyperspectral Data Classification for Internet Information Retrieval" by J. Liu and Y. Qiao focuses on the proposed intelligent data classification algorithm based on machine learning. The approach of quasiconformal kernel mapping learning with Mahalanobis distance kernel functions is presented with a clear framework. The algorithm can be utilized for Internet-based hyperspectral data retrieval, which is important for many AIoT applications such as image and video-based ones. In the experiments, it achieves advantages on large training sample construction and is effective to image data classification.

The paper titled "Flower End-to-End Detection Based on YOLOv4 Using a Mobile Device" by Z. Cheng and F. Zhang proposes a flower detection approach for smart garden applications. The method of anchor-based end-to-end deep convolutional neural network with YOLO is presented. With the designed architecture of the flower detection system, the proposed method performs an improved detection speed while the accuracy is similar to that of the other methods.

The paper titled "A Node Location Method in Wireless Sensor Networks Based on a Hybrid Optimization Algorithm" by J.-S. Pan et al. deals with the sensor node positioning problem and proposes a new algorithm named WOA-QT that combines the Whale Optimization Algorithm (WOA) with QUasi-Affine Transformation Evolutionary (QUATRE) algorithm. The algorithm optimizes the received signal strength indication (RSSI) ranging and weighted centroid positioning (WCL) for improving the positioning accuracy. In the paper, 30 benchmark functions were used to evaluate the performance. The proposed method achieves a satisfactory positioning accuracy.

The paper titled "URDNet: A Unified Regression Network for GGO Detection in Lung CT Images" by W. Liu et al. describes the topic of ground-glass opacity (GGO) nodule detection in images for medical IoT systems. The authors propose an end-to-end deep convolutional neural network with a multi-input and multioutput structure. A two-stage training method is used, which includes the network backbone training and URDNet fine-tuning with pretrained weights. The LIDC-IDRI dataset is used to evaluate the performance, and the proposed method achieves a sensitivity of 90.8%. The approach could offer a useful tool in medical applications.

The paper titled "High-Dimensional Text Clustering by Dimensionality Reduction and Improved Density Peak" by Y. Sun and J. Platoš focuses on the intelligent text-based data clustering technique. The authors propose a Stacked Random Projection (SRP) method to reduce the dimensionality of high-dimensional text data. An enhanced algorithm named DPC-K-means based on density peak clustering algorithm is also proposed. Seven text datasets, including BBC News and Amazon product reviews, were used to validate the proposed approach. The experiment results indicate the method is superior to other compared algorithms.

The paper titled "Applying Robust Intelligent Algorithm and Internet of Things to Global Maximum Power Point Tracking of Solar Photovoltaic Systems" by E.-C. Chang proposes an IoT-based model and control approach to monitor solar photovoltaic (PV) systems and ensure maximum power point tracking. A neural network based on quantum particle swarm optimization (QPSO) and radial basis function (RBF) is proposed. It is used to find the maximum power of the photovoltaic array and maintain the highest PV energy conversion efficiency. The mathematical analysis and simulations show the achievement of tracking accuracy and robust adaptation.

The paper titled "Developing Deep Survival Model for Remaining Useful Life Estimation Based on Convolutional and Long Short-Term Memory Neural Networks" by C.-H. Chu et al. focuses on the remaining useful life estimation and failure probability of machines for smart manufacturing. The authors propose an integrated deep learning approach with the convolutional neural network (CNN) and long short-term memory (LSTM) network to deal with the problem. The dataset provided by NASA is utilized to evaluate the performance. The results indicate that the proposed model can capture the degradation trend of a fault under complex conditions and avoid failure with the early prediction.

The paper titled "Fuzzy Obstacle Avoidance for the Mobile System of Service Robots" by S.-P. Tseng et al. designs and implements a service robot featured by obstacle avoidance and target user tracking. The obstacle avoidance is achieved by several ultrasonic sensors and a fuzzy-based obstacle detection method. The proposed method can effectively improve the anomaly detection of sensors when facing extreme conditions. It has low computational complexity and can facilitate real-time operation in the system. The experiment was performed in a real indoor environment, and an outstanding result was achieved.

Tien-Wen Sung
Pei-Wei Tsai
Tarek Gaber
Chao-Yang Lee

Conflicts of Interest

The guest editors declare that they have no conflicts of interest regarding the publication of this special issue.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8881684, 12 pages
https://doi.org/10.1155/2021/8881684

Research Article

Path Planning for Smart Car Based on Dijkstra Algorithm and Dynamic Window Approach

Li-sang Liu,[1,2,3,4] **Jia-feng Lin**,[1] **Jin-xin Yao**,[1,3] **Dong-wei He**,[1,2] **Ji-shi Zheng**,[1,4] **Jing Huang**,[1,3] **and Peng Shi**[1,5]

[1] *School of Electronic, Electrical Engineering and Physics, Fujian University of Technology, Fuzhou 350118, China*
[2] *Fujian Key Laboratory of A.E.D, Fujian University of Technology, Fuzhou 350118, China*
[3] *National Demonstration Centre for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, Fuzhou 350118, China*
[4] *School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK*
[5] *School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, SA 5005, Australia*

Correspondence should be addressed to Li-sang Liu; 199398190@qq.com

Received 23 July 2020; Revised 29 January 2021; Accepted 3 February 2021; Published 15 February 2021

Academic Editor: Pei-Wei Tsai

Path planning and obstacle avoidance are essential for autonomous driving cars. On the base of a self-constructed smart obstacle avoidance car, which used a LeTMC-520 depth camera and Jetson controller, this paper established a map of an unknown indoor environment based on depth information via SLAM technology. The Dijkstra algorithm is used as the global path planning algorithm and the dynamic window approach (DWA) as its local path planning algorithm, which are applied to the smart car, enabling it to successfully avoid obstacles from the planned initial position and reach the designated position. The tests on the smart car prove that the system can complete the functions of environment map establishment, path planning and navigation, and obstacle avoidance.

1. Introduction

In recent years, new industries such as cloud computing, big data, data center, virtual/augmented reality (VR/AR), 5G, artificial intelligence (AI), Internet of Things (IoT), and optical fiber sensing have emerged. These developments have changed our way of life and have simplified the completion of tasks that were difficult in the past [1]. Nowadays, Mobile Robot (MR) is widely used in various fields, such as military, industrial, agricultural, and many other applications [2]. They can replace humans to complete various difficult or dangerous operations in dangerous and special environments such as aviation, underwater, and tunnels. For example, MultiModal Mall Entertainment Robot (MuMMER) [3] can offer the public with an entertaining and engaging experience in an open-air plaza. Pepper [4] is a human-like robot, who can serve as a family companion robot for elder people. Atlas robot [5], developed by Boston Dynamics, can replace people to perform special tasks in both indoor and outdoor environments. Robots are required to work in more and more complex environments nowadays, such as shopping malls, city streets, hospitals, and train stations. Thus, to coexist harmoniously with people and other agents in these highly dynamic environments is one of the key problems. For autonomous driving robots, path planning and obstacle avoidance are the most basic functions. Traditional intelligent obstacle avoidance robots generally use binocular cameras or distance detectors to avoid obstacles. These robots have poor perception of the environment and high dependence on light and can only achieve obstacle avoidance in specific environments. In view of this, the new intelligent obstacle avoidance vehicle uses a depth camera, which combines machine image recognition and machine vision technology. It can fulfill lots of SLAM functions such as indoor mapping, automatic driving, and robotic arm grabbing on the robot operating system (ROS for short) [6]. The smart obstacle avoidance vehicle can be used in safety inspection and evaluation fields, such as road quality inspection, bridge maintenance, industrial pipeline leakage, and location search. It can also replace people to complete logistics and transportation work in dangerous environments that are not suitable for humans to work. It can also perform military tasks such as surveillance, security patrol, pollutant collection, and hazardous material disposal in a more subtle manner.

Path planning is one of the key techniques for guiding the robot in dynamic environments. The goal of the path planning is to control the robot from the starting point to the target point, subject to the constraints such that the robot does not touch any obstacle throughout the process [7]. According to the target range of path planning, it can be divided into local path planning and global path planning [8]. Local path planning refers to the smart car overall unknown or partially unknown environmental data, and the environmental information is collected by the sensors in real time to determine the distribution of local obstacles, and the optimal path from the starting point to the target point is selected, so it is also called dynamic planning. Global path planning is also known as static planning, which refers to the smart car knowing all the geographic data and planning corresponding feasible paths based on this environmental information, also known as static planning.

At present, local path planning mainly includes artificial potential field method and dynamic window approach. The artificial potential field method is an abstract artificial force field, which assumes that a virtual gravitational field is formed between the target point and the car, and the gravitational force is inversely proportional to the distance between the car and the target point; a virtual repulsion field is formed between the obstacle and the car; the repulsive force is inversely proportional to the distance between the car and the obstacle; the smart car is regarded as a particle. The path optimization can be achieved by using the form of gravitation and repulsion functions. The planned path has fast convergence speed, smooth path, good real-time algorithm, and small amount of calculation. However, when facing with a complex dynamic environment, there will be a situation where the combined force is zero and the car stops moving during the movement of the smart car, so the design of the gravitational field is the key to the success of the algorithm. The dynamic window approach is to combine the sampled multiple sets of speed with the motion constraints of speed and acceleration, then evaluate the simulated multiple sets of trajectories through an evaluation function, and select the optimal speed corresponding to the trajectory as the drive of the car. Compared with the artificial potential field method, this approach has the characteristics of fast planning, safe and reliable, and good real-time performance [8–10].

A lot of researches have been done on global path planning, and many efficient algorithms have been proposed. These algorithms can be divided into three categories, Graph Search Algorithms, Random Sampling Algorithms, and Intelligent Bionic Algorithms. The Graph Search Algorithms mainly include the Dijkstra algorithm, A∗ algorithm, BFS algorithm (breadth-first search), and DFS algorithm (depth-first search). The Dijkstra algorithm and A∗ algorithms are widely implemented in the ROS (robot operating system) [11]. Although these methods are improved by reducing the number of searching grids through a heuristic estimation, the planning efficiency is inevitably low when the environment is complex, however. Random Sampling Algorithms include BIT (batch informed trees), RABIT (regionally accelerated batch informed trees), RRT (rapidly exploring random tree), and Risk-DTRRT (risk-based dual-tree rapidly exploring random tree). These algorithms are more efficient and widely used in dynamic high-dimensional environments. Intelligent Bionic Algorithms mainly include GA (genetic algorithm), ACO (ant colony algorithm), ABC (artificial bee colony algorithm), and PSO (particle swarm optimization algorithm). These methods simulate the evolutionary and biomimetic insects' behaviors. Wang et al. [12] proposed the optimization of the genetic algorithm-particle swarm optimization algorithm (OGA-PSO) to speed up the calculation and solve the shortcoming of locally optimal. Liu et al. [13] raised a combined algorithm with artificial potential field and geometric local optimization method, aiming to search a globally optimal path. Mac et al. [14] optimized constrained multiobjective PSO with an accelerated update methodology, which can shorten and smoothen the optimal global robot path.

This paper designs and implements the autonomous path planning of the smart car. The rest of this paper is organized as follows. In Section 2, we create a two-dimensional map of an unknown indoor environment by using the SLAM method based on depth information to simulate laser scanning data. The Dijkstra global path planning algorithm and DWA (dynamic window algorithm) local path planning algorithm are introduced in Section 3. Then, in Sections 4 and 5, we test our combined algorithm in a simulation environment and on practical self-built smart cars. The results show that the car can successfully avoid obstacles from the planned initial position and reach the designated position. Section 6 comes with our conclusion that both the algorithm and the system are proved to be effective and feasible.

2. Building a Map Based on Depth Information

SLAM technology is a very important part in the realization of intelligent obstacle avoidance vehicle technology. It is a technology for the intelligent vehicle to rely on various sensors to obtain information about the surrounding environment and establish a map of the environment, while using the created environment map to achieve reliable positioning. It mainly solves the problems such as the navigation of the vehicle in an unknown environment. This paper uses the SLAM method based on depth information to simulate laser scanning data to create a two-dimensional map of an unknown indoor environment [15].

First, obtain the color map and depth map of the indoor scene through the depth camera, and establish a geometric model of depth data, and then implement the function of simulating 2D laser scanning according to the geometric model of depth data, and create a 2D map of the surrounding environment.

Figure 1 is a schematic diagram of the geometric model of depth data. The pixel value of each pixel in the depth image of the environment collected by the depth camera has a corresponding depth value. This depth value is the vertical distance depth from the plane of the object to the plane of the depth camera, not the straight-line distance R between the camera and the object. Therefore, only through the corresponding geometric transformation can obtain the distance information of the object.

The method of geometric transformation is shown in

$$\begin{cases} D = \text{depth}[i][j], \\ R = \dfrac{D}{\sin\theta}, \end{cases} \quad (1)$$

where the depth value corresponding to the pixel in the i th row and jth column of the depth image is expressed as $\text{depth}[i][j]$ and θ represents the angle in the geometric model coordinates of the coordinate of the pixel in the depth image [9].

The principle of converting depth map to laser data is shown in Figure 2. In the schematic diagram of the depth map to laser data, for any depth image point $m(u, v, z)$ in the image (which is shown as red block in Figure 2), the steps to convert each pixel data $m(u, v, z)$ into laser data $M(x, y, z)$ are as follows:

(1) Convert the point of the depth image into a point under the depth camera coordinate system

(2) Calculate the angle ∠AOC between the line AO and CO

$$\theta = \arctan\left(\frac{x}{z}\right) \quad (2)$$

(3) ∠AOC needs to be mapped into the corresponding laser data. Assuming that the minimum range and maximum range $[\alpha, \beta]$ of the laser are known, the laser beam has N equal divisions, that is, the array $\text{laser}[N]$ can be used to represent the laser data. The index n of the point M projected into the array laser can be calculated using the following formula:

$$n = \frac{\theta - \alpha}{(\beta - \alpha)/N} = N \bullet \frac{\theta - \alpha}{\beta - \alpha} \quad (3)$$

The value of $\text{laser}[n]$ represents the distance r between the point C projected on the x-axis of the point M in the depth camera coordinate system and the optical center O of the camera, that is:

$$\text{laser}[n] = r = OC = \sqrt{z^2 + x^2}. \quad (4)$$

3. Obstacle Avoidance Algorithm

Obstacle avoidance algorithm is the key to navigation, while path planning is the basic condition of obstacle avoidance algorithm. According to the target range of path planning, it can be divided into global path planning and local path planning. Global path planning refers to finding a good collision-free path among the known environment maps, and local path planning refers to planning of the intelligent car to realize its own positioning and dynamic obstacle avoidance through sensor information in a partially unknown or completely unknown environment map.

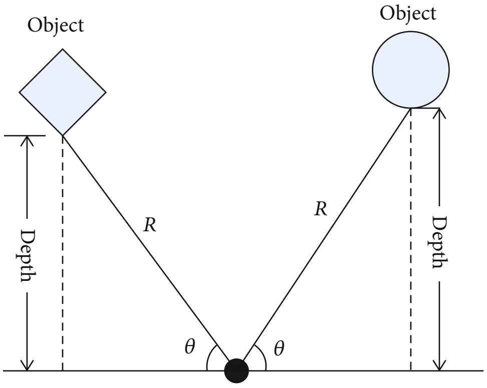


Figure 1: Schematic diagram of the geometric model of depth data.

This paper uses the Dijkstra global path planning algorithm and DWA (dynamic window algorithm) local path planning algorithm. Based on the success of global path planning, that is, global path planning first plans a roughly feasible route and then maps the global path to the local map and then uses the DWA algorithm for local path planning according to the problem of unknown obstacles that the car may face. The local path planner selects or truncates the map mapped by the global path planner based on real-time information, then samples the velocity space, scores each generated trajectory to find the optimal trajectory, and then issues velocity instructions for path planning.

3.1. Global Path Planning Algorithm. Global path planning is to make the smart car plan a path smoothly from the starting point (where it is located) to the ending point (target point). In the process of global planning, first of all, the car is required to reach the target point accurately and cannot collide with obstacles in the environment, and secondly, it is necessary to choose the path that can reach the destination as quickly as possible. The global planner designed at this time uses the Dijkstra algorithm. After specifying a starting point and an ending point in the built environment map, the Dijkstra algorithm expands radially from the starting point to the outer layer until the target point and plans the shortest path between these two points.

When using the Dijkstra algorithm to plan the shortest path, it is usually necessary to specify the starting position of the car, then introduce two sets S and U. Set S is used to record the vertices of which the shortest path has not been found, and the distance from the vertex to the starting point [16].

(1) Initially, the starting point in the map is regarded as the set S, that is, the set S contains only the starting

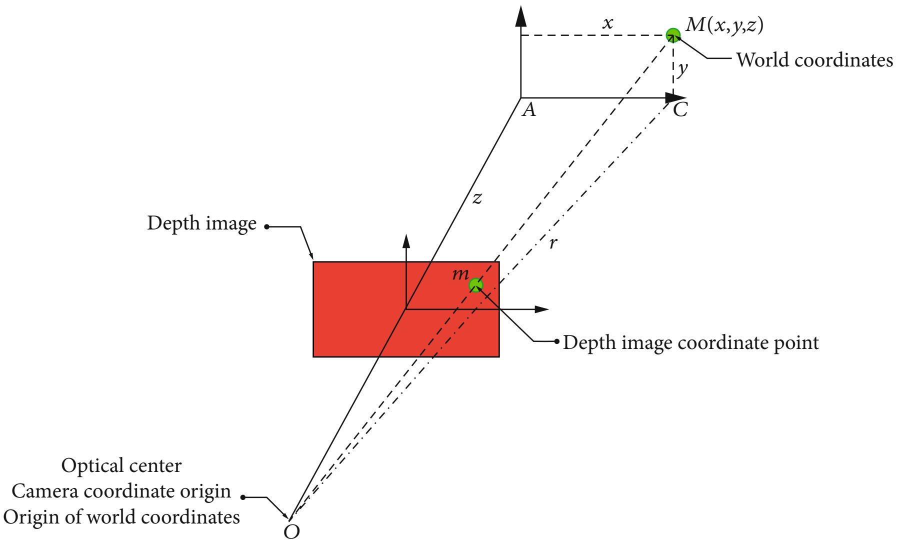


FIGURE 2: Schematic diagram of converting depth map to laser data.

point. The set U contains vertices other than the starting point

(2) According to the specified starting point, find the distance $d[i]$ from other points to the initial point. If the point is adjacent to the starting point, $d[i]$ is the edge weight (that is, the length between the two points); if the point is not adjacent to the starting point, $d[i]$ is ∞

(3) Select the smallest $d[i]$ from the set U (that is, the vertex with the shortest distance from the starting point), and add this vertex to the set S of vertices that have found the shortest path; at the same time, the vertex is removed from the set U of points that have never found the shortest path

(4) Update the distance from each vertex in the set U to the starting point

(5) Repeat steps (3) and (4) continuously until all vertices are searched; then, $d[i]$ corresponding to the target point is the shortest path length. The flowchart of the Dijkstra algorithm is shown in Figure 3

3.2. Local Path Planning Algorithm. Local path planning is to let the smart vehicle successfully complete "how to reach the destination" during the navigation process. Many uncertain factors may exist in practical car driving, for example, the local actual situation does not match the global map, temporary dynamic obstacles, and so on. It will be very dangerous if navigation was performed according to the global map. Thus, local path planning is necessary to avoid those dynamic or abrupt obstacles. Local path planning is mainly to avoid obstacles for local unknown obstacles and return to the calculated global path after avoiding obstacles. This paper uses the DWA algorithm to solve the local path planning problem.

The DWA is also called the dynamic window method, and its core idea is to transform the path planning problem into a constrained optimization problem on the velocity vector space. Multiple sets of velocities in the velocity space (v, ω), formed by the linear velocity v and the rotational velocity ω of the car, are sampled, and then, the trajectory is simulated. After obtaining multiple sets of trajectories, these trajectories are evaluated, and the optimal trajectory is selected as the actual trajectory of the car. However, there are certain constraints when sampling in velocity space. In addition, the obstacle cannot be regarded as a particle only, and the situation where the contour of the car may collide with the obstacle needs to be considered. Therefore, the dynamic window approach is added to the calculation of the expansion radius of the obstacle. The expansion radius is equal to the farthest length of the contour particle of the car and the edge of the obstacle outline. The flowchart of the DWA is shown in Figure 4. The steps of the dynamic window method are as follows:

(1) Sampling the data from the speed space to get the current state of the car

(2) For each sampling speed, calculate the movement trajectory of the car at this speed for a period

(3) Use the criteria of the evaluation function to score multiple routes and discard the infeasible routes

(4) The optimal path is selected as the actual trajectory of the car according to the scoring situation [17]

Note that the number of speed (v, ω) is infinite theoretically; thus, it is necessary to limit the speed sampling range [18]:

The car's velocity range in its own maximum and minimum speed

$$V_m = \{v \in [V_{\min}, V_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\}. \quad (5)$$

Due to the limited torque of the motor, the performance of the car will be affected by the motor drive module, so there

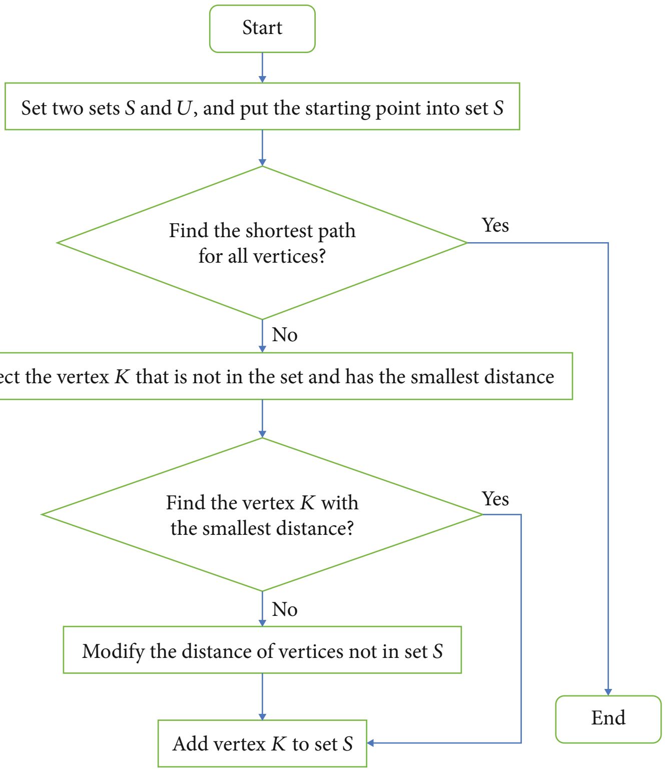

Start
Set two sets S and U, and put the starting point into set S
Find the shortest path for all vertices?
Yes
No
Select the vertex K that is not in the set and has the smallest distance
Find the vertex K with the smallest distance?
Yes
No
Modify the distance of vertices not in set S
Add vertex K to set S
End

Figure 3: Flowchart of the Dijkstra algorithm.

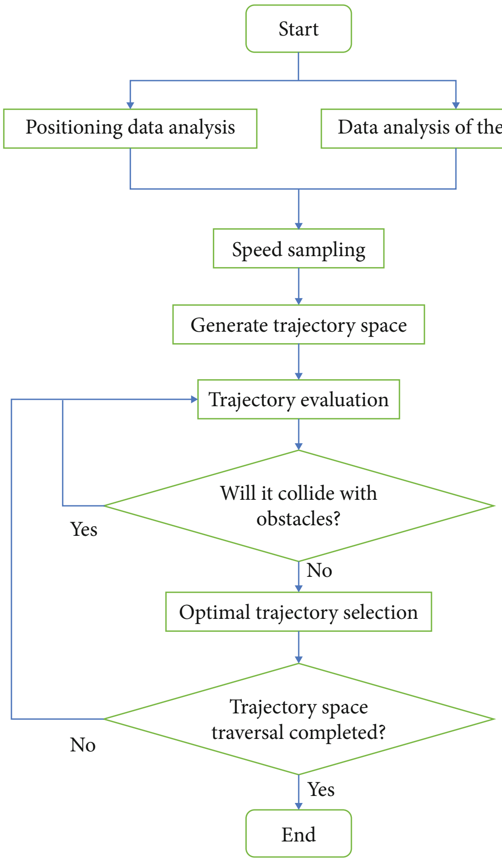

Start
Positioning data analysis
Data analysis of the car base
Speed sampling
Generate trajectory space
Trajectory evaluation
Will it collide with obstacles?
Yes
No
Optimal trajectory selection
Trajectory space traversal completed?
No
Yes
End

Figure 4: Flowchart of the DWA.

is a maximum acceleration and deceleration limit. Therefore, in the simulation cycle of the car traveling, the speed in the dynamic window is the speed that the car can actually reach:

$$V_d = \{(v, \omega) \mid v \in [V_c - V_b\Delta t, V_c + V_a\Delta t] \wedge \omega \in [w_c - w_b\Delta t, w_c + w_a\Delta t]\}. \tag{6}$$

Among them, V_c and ω_c are the current speed of the car, $V_a\Delta t$ and $w_a\Delta t$ correspond to the maximum acceleration, and $V_b\Delta t$ and $w_b\Delta t$ correspond to the maximum deceleration.

At the same time, in order to allow the car to stop before encountering an obstacle for safety consideration, the speed should have a range with the condition of maximum deceleration:

$$V_a = \left\{(v, \omega) \mid v \le \sqrt{2 \bullet \text{dist}(v, \omega) \bullet v_b} \wedge \omega \le \sqrt{2 \bullet \text{dist}(v, \omega) \bullet \omega_b}\right\}, \tag{7}$$

where $\text{dist}(v, \omega)$ represents the minimum distance between the car and the obstacle, and v_b and ω_b are the translation acceleration and rotation acceleration of the car, respectively. In the sampled speed group, there are several feasible trajectories. Therefore, each trajectory needs to be evaluated in the form of an evaluation function, so as to obtain the optimal value of the car at $k + 1$ time in the speed vector space.

The standard objective function of the DWA is

$$G(v, \omega) = \gamma \bullet \text{vel}(v, \omega) + \alpha \bullet \text{heading}(v, \omega) + \beta \bullet \text{distant}(v, \omega). \tag{8}$$

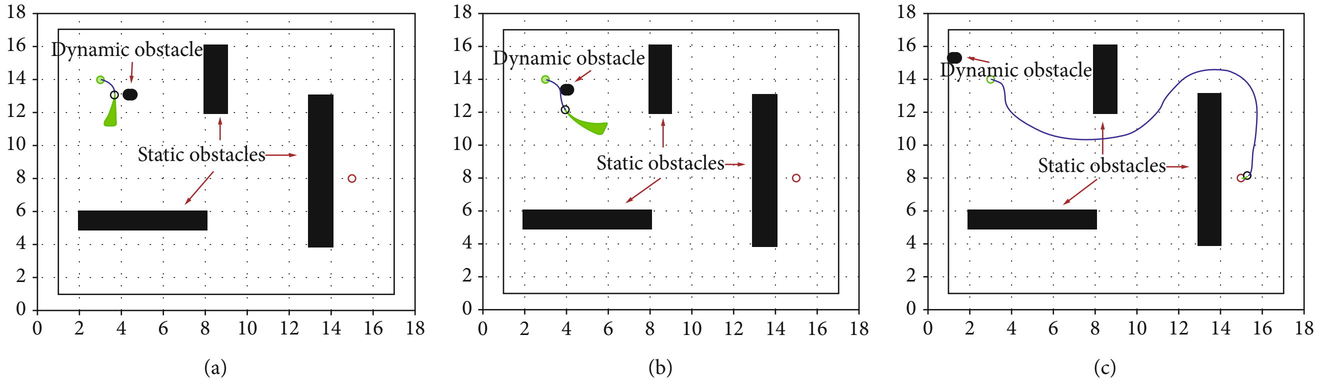


Figure 5: Simulation result of head-on situation: (a) the car meets the dynamic obstacle; (b) the car avoids the dynamic obstacle; (c) the car reaches the target position successfully.

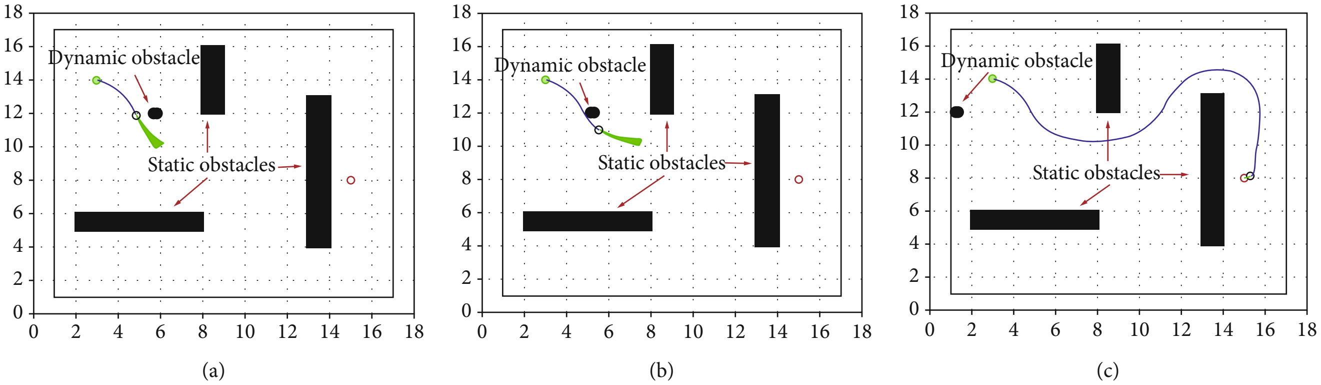


Figure 6: Simulation result of crossing situation: (a) the car meets the dynamic obstacle; (b) the car avoids the dynamic obstacle; (c) the car reaches the target position successfully.

The heading(v, ω) is a measure of progress towards the goal location. It is maximal if the car moves directly towards the target. When the angle between the car and the target point is 0 degree, the maximum value is

$$\text{heading}(v, \omega) = 1 - \frac{\theta}{\pi}. \tag{9}$$

Among them, θ represents the angle between the car and the target point.

The distant(v, ω) is the distance to the closest obstacle on the trajectory. L is the distance between the car's own position and the target point at that moment. The larger the value of the function, the farther the distance to the target point, and the less likely to be a collision; the smaller the value of the function, the closer to the obstacle, the greater the probability of a collision, which also means the higher is the car's desire to move around it.

$$\text{distant}(v, \omega) = \begin{cases} \dfrac{1}{L}, 0 \le l \le L, \\ 1, L \ll l. \end{cases} \tag{10}$$

The vel(v, ω) is the forward velocity of the car and supports fast movements.

$$\text{vel}(v, \omega) = \frac{v}{v_{\max}}, \tag{11}$$

where v is the average speed of the car and $v_{\max}$ is the maximum speed of the car in the speed space.

4. Navigation and Obstacle Avoidance Simulation

In order to verify the obstacle avoidance performance of the Dijkstra algorithm and DWA, we have established a 15 × 15 grid environment in MATLAB, as shown in Figures 5–7, and three groups of simulation experiments are carried out for navigation and avoidance of both static and dynamic obstacles. There are three different situations in each group of simulation, that is, the smart car meets dynamic obstacles in three ways, head-on, crossing, and overtaking.

The parameters in the simulation experiment are as follows: the maximum linear velocity and angular velocity of the smart car are set as 3.5 m/s and 40 rad/s; the maximum linear acceleration and angular acceleration are 0.35 m/s^2

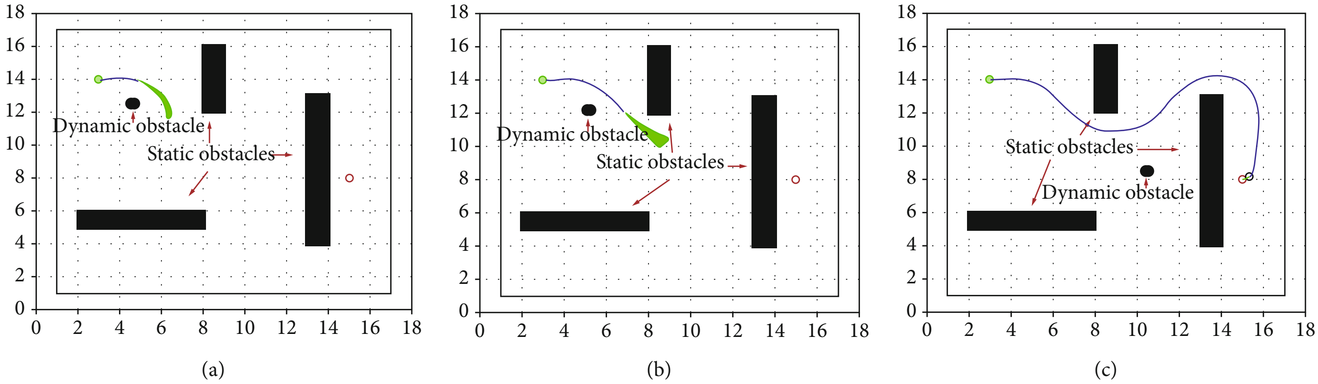


Figure 7: Simulation result of overtaking situation: (a) the car meets the dynamic obstacle; (b) the car avoids the dynamic obstacle; (c) the car reaches the target position successfully.

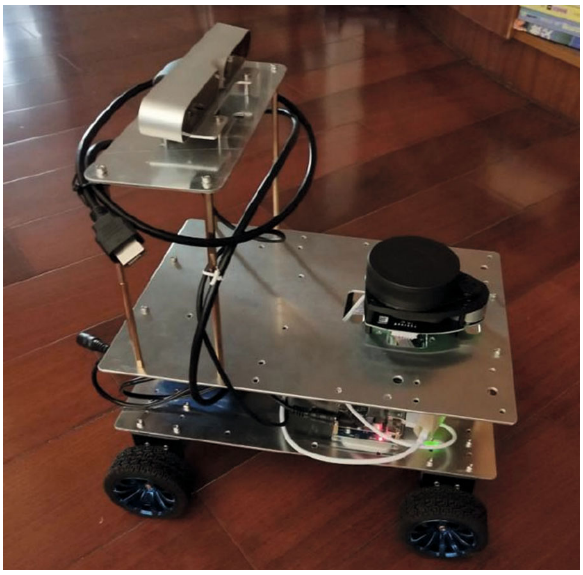


Figure 8: Smart obstacle avoidance car.

and 60 rad/s^2; the linear velocity resolution is 0.01 m/s, and the angular velocity resolution is 1 rad/s. The three weighted coefficients $\sigma\sigma$, $\beta\beta$, and $\gamma\sigma$ of the evaluation function are set as 0.05, 0.2, and 0.1. The starting position is indicated by a green dot, and its coordinates are (3,14); the target position is indicated by a red dot, and its coordinates are (15,8); the radius of dynamic obstacles is 0.3 m; the blue curves are the optimal paths obtained by the Dijkstra algorithm and DWA.

The simulation results of head-on situation are shown in Figure 5. The dynamic obstacle moves about 55° up to the left at a speed of 0.35 m/s from the initial position (6,12) and head-on to the car. When meeting the dynamic obstacle, as shown in Figures 5(a) and 5(b), the car will actively detour under the dynamic obstacle along the optimal trajectory evaluated by the DWA to avoid it. Then, the car continues to move towards the target position along the optimal trajectory. Finally, we can see from Figure 5(c) that the car reaches the target position safely and plans a smooth path.

The simulation results of crossing situation are shown in Figure 6; the dynamic obstacle moves horizontally to the left at a same speed from the initial position (7.5,12) and crosses the car. At this moment, the car will still actively detour under the dynamic obstacle along the optimal trajectory to avoid it as shown in Figures 6(a) and 6(b). Then, the car continues to move towards the target position along the optimal trajectory. Finally, we can also see from Figure 6(c) that the car reaches the target position safely and plans a smooth path, too.

The simulation results of overtaking situation are shown in Figure 7. The dynamic obstacle moves about 55° down to the right at a speed of 0.15 m/s from the initial position (4,13). When meeting the slow-moving dynamic obstacle, as shown in Figures 7(a) and 7(b), the car will actively detour to the left and surmount it to avoid it. Finally, we can still see from Figure 7(c) that the car reaches the target position safely and plans a smooth path.

We can find from the above three simulation results that the smart car can avoid dynamic obstacles well considering head-on, crossing, or overtaking situations, and the smooth final path the car planned still meets the global optimal performance.

5. Navigation and Obstacle Avoidance Tests

A self-built smart car was constructed as shown in Figure 8, mainly using Jetson Nano as the main controller, a motor with a reduction ratio of 30, a LeTMC-520 depth camera, and so on. Before the test starts, the angular velocity, linear velocity, and IMU of the car should be calibrated, and the PID parameters of the camera module and motor control module should be debugged. Various tests such as navigation and obstacle avoidance can be conducted only after they are confirmed to be correct and feasible. The working principle diagram of the car is shown in Figure 9.

5.1. Build Environment Map. When the depth camera and Rviz visual control platform are working, the grid map can be utilized and the keyboard can be used to control the car and collect environmental information. When using SLAM to model a new environment, it is necessary to manually control the rotation and forward speed of the intelligent car at medium level , so as to ensure the accuracy of created map.- After collecting all the environmental information, save the established map in time to facilitate future experiments.

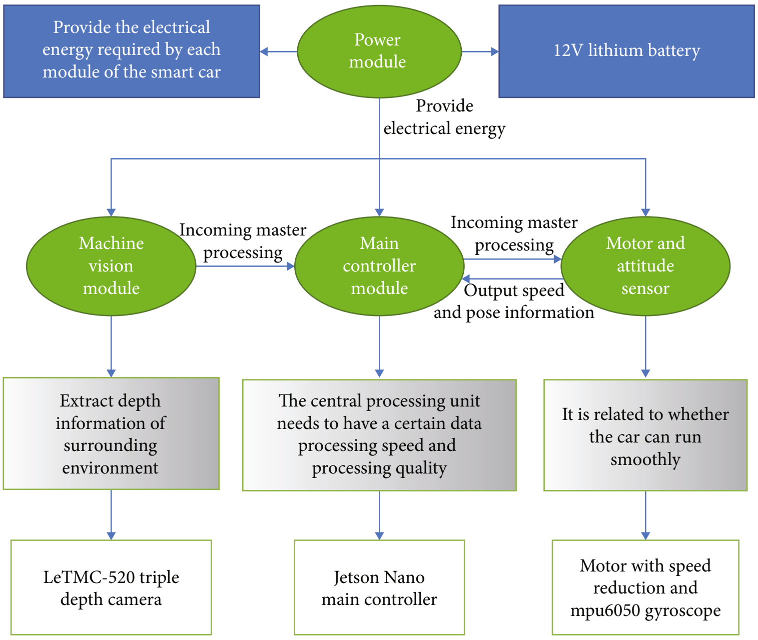
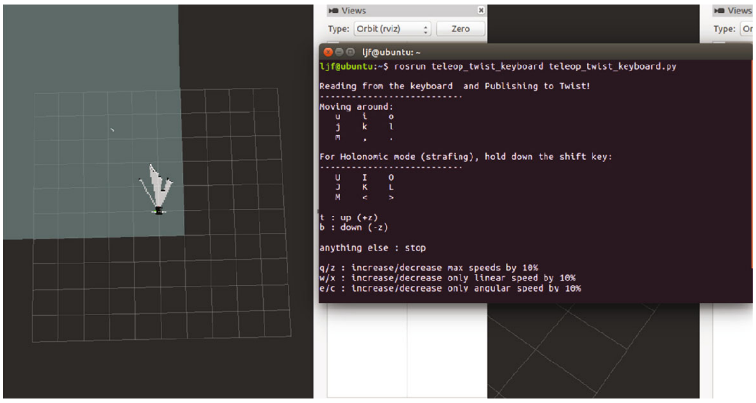

Provide the electrical energy required by each module of the smart car
Power module
12V lithium battery
Provide electrical energy
Machine vision module
Incoming master processing
Main controller module
Incoming master processing
Motor and attitude sensor
Output speed and pose information
Extract depth information of surrounding environment
The central processing unit needs to have a certain data processing speed and processing quality
It is related to whether the car can run smoothly
LeTMC-520 triple depth camera
Jetson Nano main controller
Motor with speed reduction and mpu6050 gyroscope

Figure 9: Principle of the smart car.



```
ljf@ubuntu:~$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py

Reading from the keyboard  and Publishing to Twist!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

For Holonomic mode (strafing), hold down the shift key:
---------------------------
   U    I    O
   J    K    L
   M    <    >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
```

Figure 10: The image collected by the car at the original position.

The *U*, *I*, *O*, *J*, *K*, and *L* keys in the keyboard correspond to the left, straight, right, left turn, backward, and right turn of the control car, respectively.

Figure 10 is a map image collected when the car is just in its original position, showing the orientation of the car and the range of the environment currently being collected. Figure 11 is a built environment map of barrier-free information.

5.2. Obstacle Avoidance Tests. All the obstacle avoidance studies of this paper are based on the environment map. First, specify a starting position in the map, and then, use the global path planning and local path planning to make the car bypass the obstacle to reach the specified target position.

The test of the obstacle avoidance algorithm is divided into three situations, from simple to complex. To detect whether the obstacle avoidance algorithm is suitable for a variety of occasions, here are some representative detections as follows:

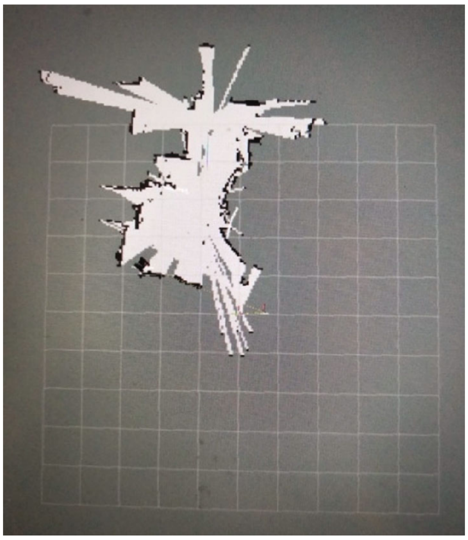


Figure 11: Environmental information of barrier-free objects.

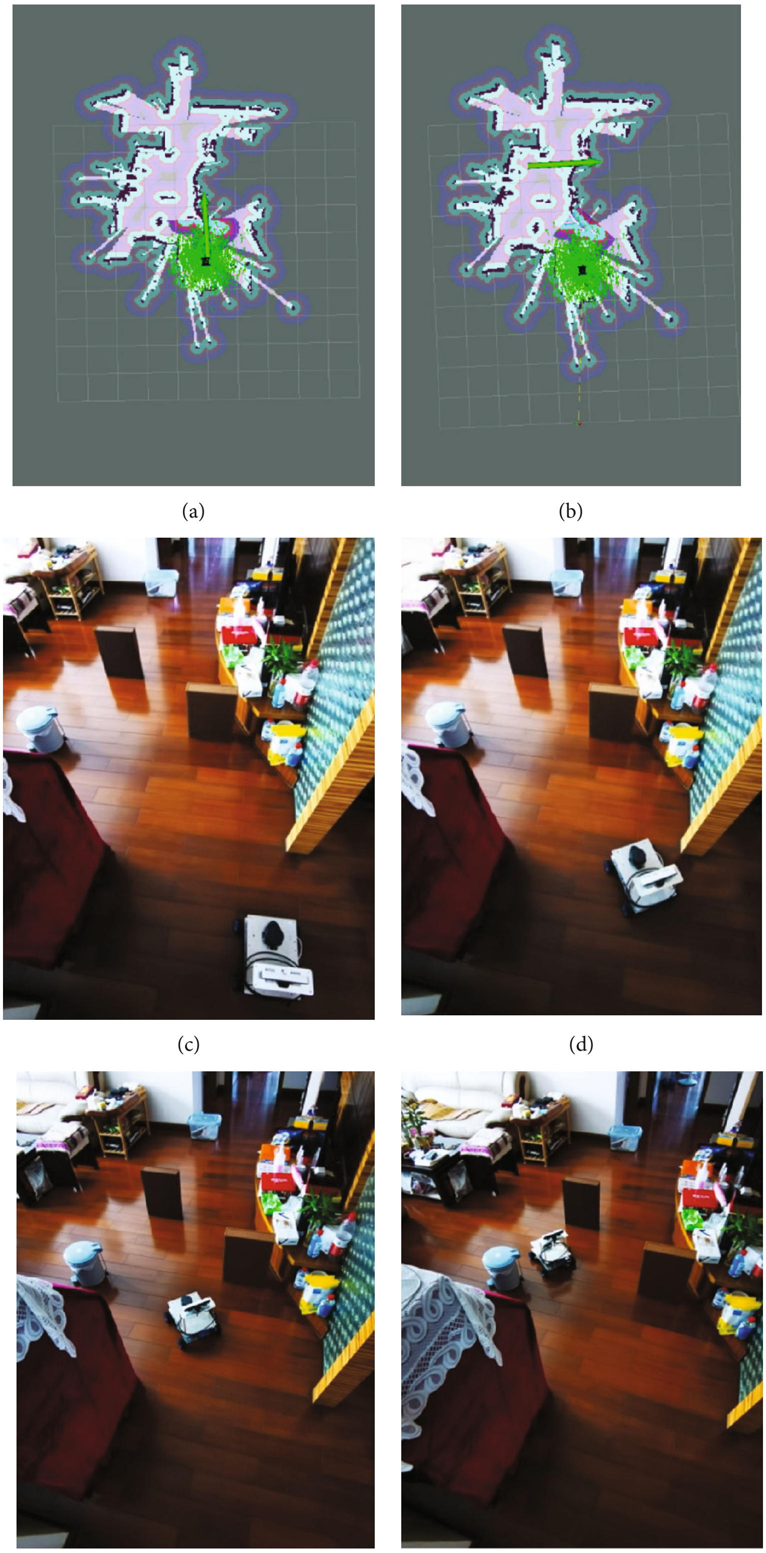
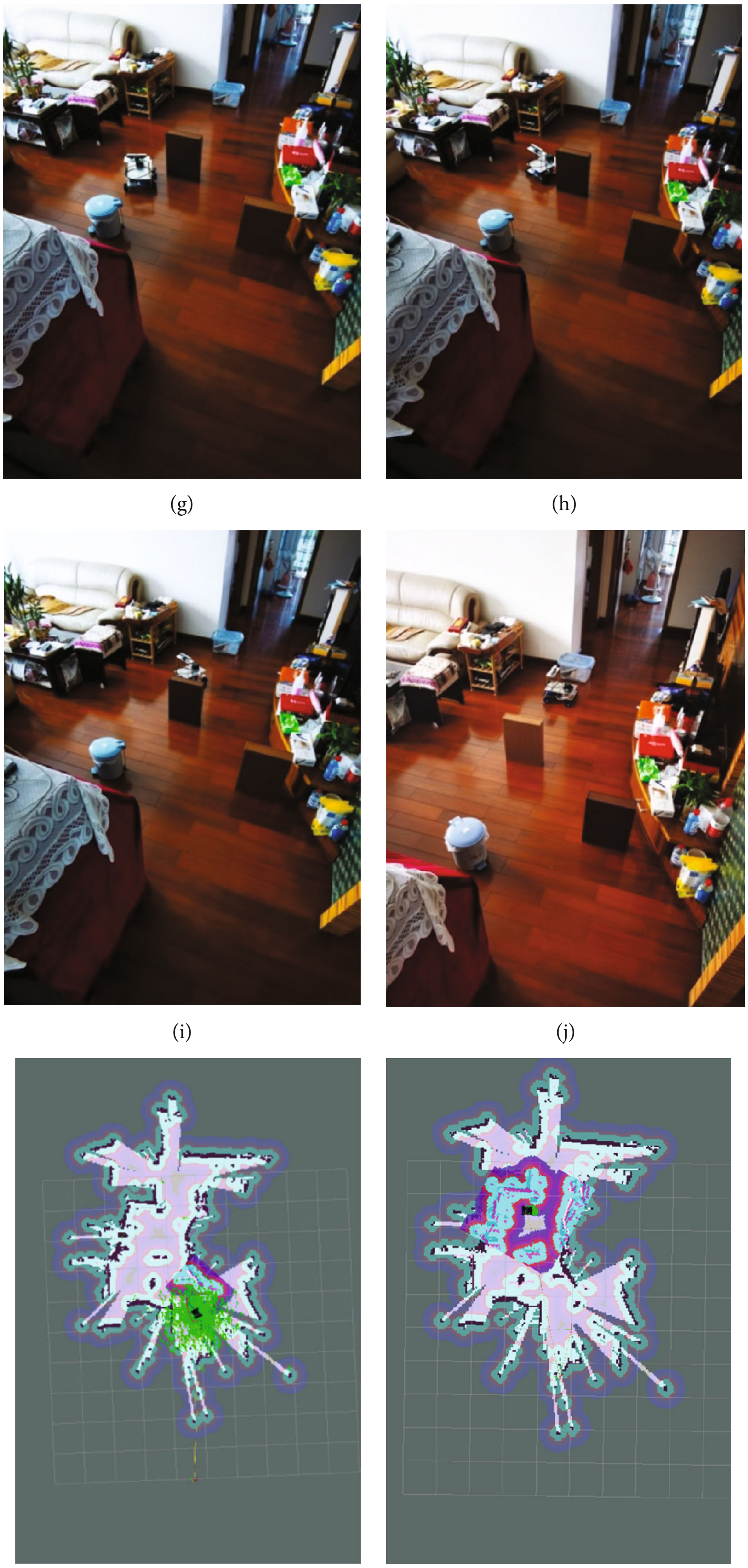

(a) (b) (c) (d) (e) (f)

Figure 12: Continued.



(g)

(h)

(i)

(j)

(k)

(l)

Figure 12: Obstacle avoidance test in a complex obstacle environment: (a) specifies the starting point; (b) specify the target point; (c) the starting position of the car; (d) bypass the first obstacle; (e) through box and trash can; (f) bypass trash can; (g) through the trash can and box; (h) adjust heading angle; (i) bypass box; (j) the car arrives at the designated location; (k) car path planning; (l) the console shows that the car has reached the destination (complete the task).

(1) In the simple obstacle environment, only one book is placed on the road as an obstacle. The test shows that the car can easily bypass obstacles from the starting position to the specified target position

(2) In a complicated environment, a book and two milk tanks are placed on the road, and which are placed in the road that are relatively close to each other enough for cars, to avoid obstacles to plan. The test also shows that the car can avoid obstacles and pass through the middle of these obstacles

(3) In a more complex environment, pianos, TV cabinets, and tea tables are used as borders on the road, and books, storage boxes, trash cans, and stools are used as obstacles. They are staggered to form an S-shaped roadblock. The specific obstacle avoidance test is shown in Figure 12. The test still shows that the car can also successfully avoid obstacles. It is worth mentioning that an appropriate inflation radius needs to be chosen carefully if there are different sizes of obstacles. In this experimental test, the value of the obstacle radius was finally set to 0.3 meters after debugging. If there are obstacles with special shapes and sizes, the obstacle radius value needs to be adjusted accordingly

The test process of the S-shaped roadblock in a complex environment is shown in Figure 12.

6. Conclusion

This paper designs and implements a smart obstacle avoidance car system, including hardware platform construction and software obstacle avoidance algorithm implementation. The main work can be summarized as follows: (1) A smart car was self-designed and self-built with a LeTMC-520 depth camera, a Jetson controller, four motors, and other modules. (2) Based on the ROS, SLAM technology was used to build a new unknown indoor environment map by converting depth map image to laser data; (3) the Dijkstra algorithm and DWA dynamic window method were used as the global path planning algorithm and local path planning algorithm, respectively. The validity and feasibility of the combined algorithm are approved by simulation results. The practical experiments were carried out and showed that the car can successfully avoid obstacles from the planned initial position and reach the designated position in a smooth planned path. The algorithm and the system are both proved to be effective and feasible.

Data Availability

Data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Initial Scientific Research Fund of FJUT (GY-Z12079 and GY-Z160124), Fujian Provincial Education Department Youth Fund (JAT170367 and JAT170369), Natural Science Foundation of Fujian Province (2019J01773, 2018J01640, and 2017J01728), and China Scholarship Council (201709360002).

References

[1] Y. Zhang and J. Xin, “Survivable deployments of optical sensor networks against multiple failures and disasters: a survey,” *Sensors*, vol. 19, no. 21, p. 4790, 2019.

[2] H. T. Nguyen and H. X. Le, “Path planning and obstacle avoidance approaches for mobile robot,” *International Journal of Computer Science*, vol. 13, no. 4, 2016.

[3] M. E. Foster, R. Alami, O. Gestranius et al., “The MuMMER project: engaging human-robot interaction in real-world public spaces,” in *Social Robotics. ICSR 2016. Lecture Notes in Computer Science, vol 9979*, A. Agah, J. J. Cabibihan, A. Howard, M. Salichs, and H. He, Eds., pp. 753–763, Springer, Cham, 2016.

[4] F. Tanaka, K. Isshiki, F. Takahashi, M. Uekusa, R. Sei, and K. Hayashi, “Pepper learns together with children: development of an educational application,” in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 270–275, Seoul, South Korea, November 2015.

[5] A. J. Bautista and S. O. Wane, “ATLAS robot: a teaching tool for autonomous agricultural mobile robotics,” in *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 264–269, Hangzhou, China, October 2018.

[6] T. Kaiqiao, *Design and Implementation in Intelligent Avoidance Obstacle Robot Based on ROS and Depth Camera Kinect*, Changchun University of Technology, 2018.

[7] K. Cai, C. Wang, J. Cheng, C. W. De Silva, and M. Q. Meng, “Mobile robot path planning in dynamic environments: a survey,” *Instrumentation*, vol. 6, no. 2, pp. 92–102, 2019.

[8] H. Shuyun, T. Shoufeng, T. Ziyuan, S. Bin, and T. Minming, “A survey of path planning methods for autonomous mobile robots,” *Software Guide*, vol. 17, no. 10, pp. 1–5, 2018.

[9] S. Xiaoru, R. Yiyue, G. Song, and C. Chaobo, “Survey on technology of mobile robot path planning,” *Computer Measurement and Control*, vol. 27, no. 4, pp. 1–5+17, 2019.

[10] B. Encyclopedia, *Path planning [EB/OL]*2017-03-06, https://baike.baidu.com/item/pathplanning/8638339#3.

[11] L. Zhi and M. Xuesong, “Navigation and control system of mobile robot based on ROS,” in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 368–372, Chongqing, China, October 2018.

[12] X. Wang, Y. Shi, D. Ding, and X. Gu, “Double global optimum genetic algorithm–particle swarm optimization-based welding robot path planning,” *Engineering Optimization*, vol. 48, no. 2, pp. 299–316, 2016.

[13] J. Liu, J. Yang, H. Liu, X. Tian, and M. Gao, “An improved ant colony algorithm for robot path planning,” *Soft Computing*, vol. 21, no. 19, pp. 5829–5839, 2017.

[14] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, “A hierarchical global path planning approach for mobile robots based

on multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 59, pp. 68–76, 2017.

[15] W. Haozuo, *Research on SLAM Algorithm Based on Depth Perception of Indoor Mobile Robot*, Lanzhou University of Technology, 2016.

[16] S. Yong, *Research of Ship Path Planning Algorithm*, Wuhan University of Technology, 2018.

[17] C. Zhuo, S. Weihua, and S. Ning, "SLAM and path planning of mobile robot in ROS framework," *Medical and Health Equipment*, vol. 38, no. 2, pp. 109–113, 2017.

[18] W. Hanyuan, "Design and improvement of autonomous path planning algorithm for smart car," *Electronic Technology and Software Engineering*, vol. 4, pp. 84-85, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8887547, 11 pages
https://doi.org/10.1155/2020/8887547

Research Article

Fuzzy Obstacle Avoidance for the Mobile System of Service Robots

Shih-Pang Tseng,[1] **Che-Wen Chen**,[2] **Ta-Wen Kuan**,[3] **Yao-Tsung Hsu**,[2] **and Jhing-Fa Wang**[2]

[1] *Software and Big Data School, Changzhou College of Information Technology, Jiangsu, China*
[2] *Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan*
[3] *School of AI, Guangdong and Taiwan, Foshan University, 528011, China*

Correspondence should be addressed to Shih-Pang Tseng; tsengshihpang@ccit.js.cn

Received 4 August 2020; Revised 9 November 2020; Accepted 17 November 2020; Published 8 December 2020

Academic Editor: Chao-Yang Lee

This study implements Fuzzy logic-based obstacle avoidance and human tracking on an omnidirectional mobile system for service robots. The mobile system could be separated and combined with the robot which can be controlled remotely and switched to go forward and avoid obstacles in an indoor environment automatically. The system is able to track and go to the user according to the user's position. The omnidirectional wheel was adapted in the power system to perform translating and spinning movements. The translating movement enables the robot to avoid obstacles faster and flexibly in paths. With the spinning movement, the robot can quickly find the direction of the object. Finally, the experiments show that the proposed system has good performance in service environments.

1. Introduction

The mechanical construction and artificial intelligence of robots have developed rapidly with the advancement of science and technology in the past decades [1]. Robots are being increasingly applied, not only in the fields of traditional industry, agriculture, and aquaculture but also in many environments such as medicine [2], medical equipment [3], education [4], home [5], and entertainment [6] where they had been widely used; robots replace manpower to save labor costs for enterprises. Factory automation robots such as handling robots and assembling robots can accelerate productivity and improve product yield. Demolition robots and exploration robots can replace people in performing tasks in hazardous environments, reducing risks such as monitoring gas poisoning or gas leakage. Additionally, people have rising expectations for quality of life. The development of service robots decreases the workload of service personnel, allowing service personnel to do more valuable things, thereby improving the service quality. Therefore, intelligent service robots have become an important development trend.

This study extends the previous research by Chen et al. [7]. The proposed system is equipped with a multifunctional human-machine interactive multimedia interface so that users can interact with the robot, and the robot can also guide users to purchase goods. The robot can adopt Kinect to track the user and follow or come to the user through the fuzzy logic control of the motor. The robot body moves the base and is separable, enabling remote control of the mobile system and avoiding obstacles along paths. The system can be controlled when it is not detached remotely. Finally, some experiments to test the design are designed. The success rate is computed in each case after 30 test runs per experiment. The outline of this study is as follows. Section 2 reviews the recent relevant work and limits the scope of this paper. Section 3 describes the system overview. Section 4 introduces the proposed composite solution. Section 5 presents the simulation of this solution. Section 6 draws conclusions and describes future work.

2. Related Work

In recent years, more and more scholars have researched mobile robots and have shown many achievements in the field, such as path planning [8, 9], obstacle avoidance system [10], navigation system [11], and tracking. As research into mobile robots is extensive, this study mainly focuses on exploring and implementing target tracking, obstacle avoidance, and fuzzy control for the proposed mobile robot.

2.1. Robot Movement. Robots can be classified by movement mode into wheel robots and foot robots. Foot robots move at slower speeds than wheel robots but can handle more complex terrains. The most common foot robots are hexapod [12], quadruped, and biped robots [5]. Wheeled robots change direction and move faster, suitable for moving in flat terrain. Chen et al. [13] and Huang and Chiang [14] adopted wheeled robots as research projects. This study adopts a wheeled robot as the base of the proposed robot system.

2.2. Target Tracking. This section reviews the relevant research of mobile platforms and mobile robots that are capable of target tracking. Li et al. [15] applied 3D cameras to estimate the target distance and establish the relationship between the control of input and the depth image, thus constructing a human tracking algorithm on a mobile robot. Zhang et al. [16] used utilized matching to track the target. The system locks the target according to the clothing color. Benli et al. [17] examined target tracking for intelligent perception in unmanned systems and focused on perspective visual and infrared bands, as well as O-D visual band tracking to approach target tracking. Feng et al. [18] proposed a target person tracking model based on ultra-wideband (UWB) technology for human tracking robots and overcome the challenge of the measurement errors with a modified hyperbolic positioning algorithm. Williams and Sukhatme [19] demonstrated a metric of observability to improve system localization. Sun et al. [20] presented a 3-DOF pedestrian trajectory prediction approach for autonomous mobile robots.

2.3. Obstacle Avoidance. The main aim of obstacle avoidance is to measure the distance from obstacles, determine the location of obstacles, and identify the shape of different obstacles. Owing to the rise in computing speed in recent years, many visual sensing methods for obstacle avoidance have been presented. Al Arabi et al. [21] proposed a method for navigation and path planning to avoid obstacles. The robot generates a 2D map using range measurement sensors and obstacle detection through radar. Mvemba et al. [22] utilized a sensor mounted on the front of a mobile robot. The servo scans and detects obstacles within the interval by estimating the distance of detected obstacles. Ziaei et al. [23] developed a global path planning method using a single overhead camera based on the principle of artificial potential fields. Chen et al. [13] presented a path tracking method for wheeled mobile robots. Their method includes path planning and controller design. The path planning employs a B-spline to generate an obstacle avoidance path, thus decreasing statistically the possibility of collision.

3. System Overview

This section introduces the design concept of robots and the important components.

3.1. Robot Mechanism Architecture. The robot is composed of DC motors, servo motors, aluminum members, plastic parts, and a control board and is controlled by a Servo Commander 16 control board. The movement method adopts four DC motors with omnidirectional wheels, which can be rotated in the same direction or moved in all directions. The joint utilizes five 25 kg servo motors and six 10.8 kg servo motors, with an Intel mini host and 10-inch IPS screen. The power supply applies a 12 V 7.2 Ah lead-acid battery for the motor and module, while the screen, Intel mini host, Kinect, and speaker power supply are powered with a UPS uninterruptible power system. The head, hands, and waist have 11 degrees of freedom. The height is about 112 cm. The system can be moved by remote control using a handheld device, and the robot can be set to move autonomously as shown in Figure 1.

3.2. System Architecture. The proposed system comprises a basic commander, RGB-D camera (Kinect), DC motors, ultrasonic sensors, and a mini PC as shown in Figure 2. The RGB-D camera transmits target tracking information while other modules such as the motor controller and sensors are connected to the basic commander.

3.3. Mobile Platform. The omnidirectional mobile platform is integrated with four omnidirectional wheels. Generally, the turning process causes cumulative deviation, causing severe body displacement after several turnings. Omnidirectional wheels enable the platform to move horizontally along the path without constant turning, thus solving the problem of error accumulation. The platform's size is 480 mm × 480 mm and the four omnidirectional wheels are 102 mm diameter and width 35 mm width. The rollers are shaped such that the overall side profile of the wheel is circular, so as to ensure the consistency and stability of the wheel in contact with the ground. The partial force of the wheel can be transformed into the perpendicular direction by rollers. The torque in an arbitrary direction is synthesized with the cooperation of speed and direction of the wheels. Figure 3 shows the omnidirectional mobile platform.

3.4. Depth Camera. Depth data extraction and color image data acquisition are similar operations, both requiring an understanding of depth data which is still necessary. Infrared light is emitted, and with the detection of infrared light reflection, the depth value of every pixel in the visual view can be calculated. In the image frame provided by the depth data stream, each pixel represents the closest distance from the object to the plane of the camera. The maximum value of the depth data is 1220 mm to 3810 mm. The skeleton tracking scheme builds the joints of the human body, which can then identify and locate different body parts, such as the hand, head, and body. When the object walks into Kinect's visual field, Kinect v1 as shown in Figure 4 can find the locations of the 20 joint points and represent the positions in (x, y, z) coordinates. Unlike the depth image spatial coordinates, these coordinate units are meters. In this way, when objects

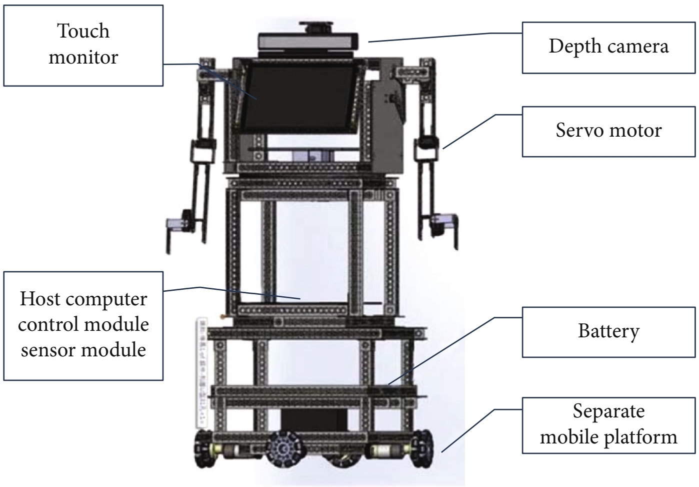


Figure 1: The prototype of the robot.

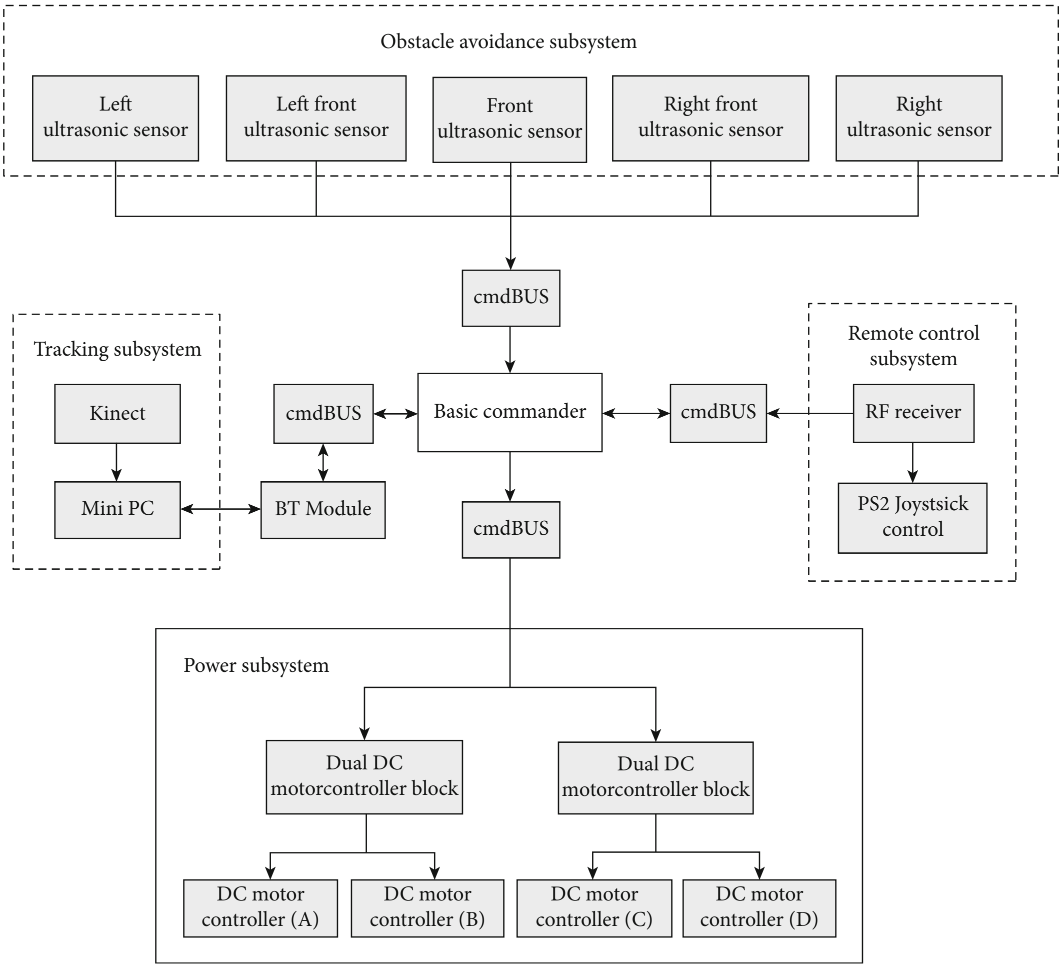


Figure 2: System architecture.

perform many complex actions in front of Kinect, the displacement between the locations of these joints is very obvious, helping the computer to understand what action objects are performing.

3.5. Ultrasonic Sensor. The system adopts a Sonar A module to receive user instructions and measures distance based on user demands under proper environmental conditions. The detection distance is 5 meters. Eight ultrasonic waves were

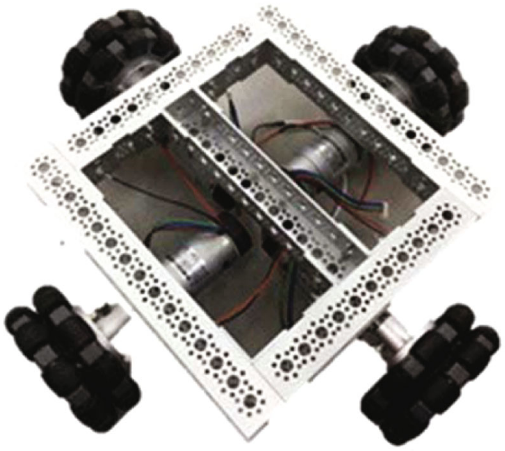


Figure 3: Omnidirectional.

Figure 4: Kinect.

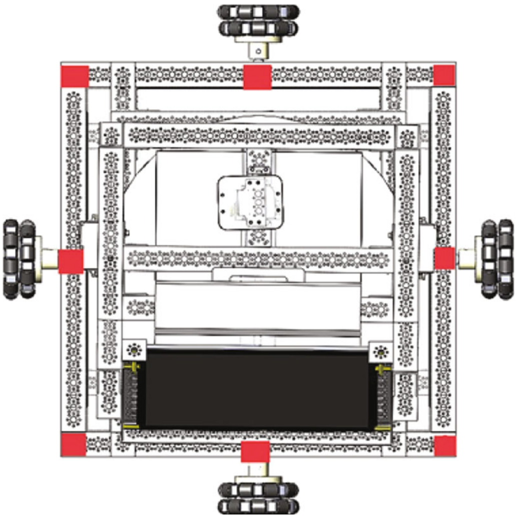


Figure 5: Ultrasonic.

set up on the mobile platform, at positions as shown in Figure 5.

4. Proposed Methods

4.1. Fuzzy-Based Obstacle Avoidance. This study proposed a new anomaly detection method of ultrasonic obstacle avoidance, which can solve the problem of the ultrasonic sensor failing to detect the distance between the obstacles due to its shape and material. Therefore, the method can effectively improve the anomaly detection ability of the ultrasonic sensor when facing extreme conditions. The method provides stable ultrasonic input values to the fuzzy controller, and with the stable input range, the method simplifies the design of the initial member function and fuzzy logic rules and therefore reduces the difficulty and the time spent in subsequent optimization.

In addition, the method has low computational complexity, enabling the entire system to be operated in real time and to integrate more flexibly with other sensing systems. In the application of the actual target tracking, the method makes the translate obstacle avoidance of mobile platforms safer and thus more practical. Therefore, the mobile platform can avoid obstacles to target tracking without turning, decreasing the risk of losing the target when avoiding obstacles. The method can enhance the ability of anomaly sensing and thus avoid the obstacle safely even if it is applied in a steering-oriented obstacle avoidance strategy. Because the ultrasonic distance measurement is triggered and calculated with the reflected wave, whether the reflected wave can truly show the presence and the distance from the obstacle is a critical issue. The distribution of the ultrasonic emission can be simulated as fan-shaped, while the ultrasonic receiver is triggered to derive the distance only if it receives enough energy. Therefore, if the obstacle cannot reflect the ultrasonic wave due to its shape, placement, or material, then the obstacle detection system will measure the distance inaccurately or even fail to detect the obstacle. The new method adopts the ground as a flat plane to reflect the ultrasonic wave stably, with a fixed difference between the actual distance and the measurement value. Several distance ranges of stable reflection zones, called the critical boundary, were chosen and tested through different types of obstacles, to see which critical boundary can determine more abnormal conditions with clear rules. The final critical boundary was chosen after multiple experiments as a fixed benchmark of anomaly detection. Figure 6 shows the side view of the critical boundary formed by the ultrasonic wave.

If the obstacle touches the critical zone, then the value of the measurement from the reflected wave of the critical boundary changes, leading to an abnormal state. This abnormal state indicates that the obstacle is too close to the mobile platform in a specific direction, and therefore, the mobile platform must move accordingly in order to turn the abnormal state back to normal, maintaining a clear critical boundary to ensure safety. In general, if the invaded obstacle is a plane with little or no incline, then the distance measured by the ultrasonic sensor falls, such that even the conventional obstacle detection method knows that an obstacle is coming

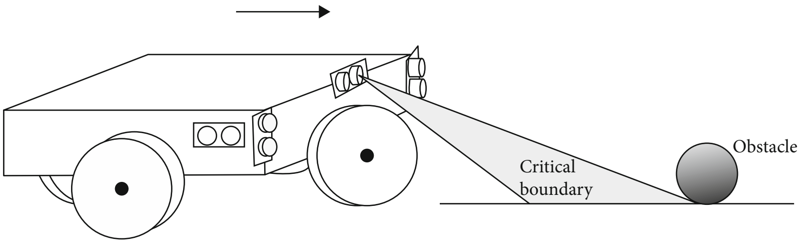


Figure 6: Boundary.

close. However, if the obstacle material absorbs ultrasound, or if its shape is cylindrical or with an inclined plane, then the value of the ultrasonic sensor measurement will be unstable or simply shows a far distance, since the reflected wave is received at different times with different energy levels or is absorbed by the obstacle. In these cases, the conventional obstacle detection method and the avoidance rules are ineffective. In the new method, even if the obstacle is cylindrical, with an inclined plane or made of a material that absorbs ultrasonic waves, it still blocks the ultrasonic reflection of the critical boundary from the ground, making the value no longer limited to a stable range. Therefore, these cases can be identified as the obstacle entering the critical boundary in a particular direction and enabling the mobile platform to take the corresponding avoidance strategy. Figure 7 illustrates the top view of the critical boundary formed by multiple ultrasonic sensors. The proposed method has much better security of a translate obstacle avoidance strategy for target tracking.

The traditional translate obstacle avoidance strategy can avoid target loss by keeping the target inside the sight view of Kinect, when the target is not too large. However, the traditional strategy can only rely on the left or right front of the ultrasonic sensor to determine whether it can move forward safely. Unfortunately, experimental results indicate that the left front or the right front of the ultrasonic sensor, when encountering an inclined plane or cylindrical obstacle, often fails to find the obstacle or measures the distance from the obstacle inaccurately, causing the sensor to hit the obstacle on the side when moving forward. Using the new detection method, the ultrasonic sensors can detect anomalies earlier and try to clean the critical boundary by translating in the opposite direction. The new method can improve the ability to sense an obstacle, whether it is applied to the translate avoidance or steering-oriented avoidance strategy. Figure 8 illustrates the comparison results of the conventional and new methods when using a translate obstacle avoidance strategy.

The inference needs to find the mass center by integrating continuously changing functions, making the calculation process inefficient. To achieve real-time operation, this study adopts the Sugeno-style inference method.

The Sugeno-style inference scheme uses only one spike of the single state function as the latter membership function of the fuzzy rule, which takes the following form:

$$\text{IF } x = A \text{ AND } y = B \text{ THEN } z = f(x, y). \tag{1}$$

All latter terms of membership functions are represented by a single state spike. The final step is to adopt the center of gravity method to calculate a clear output. The fuzzy rule adopted herein is the zero-order Sugeno fuzzy model, which is also commonly used in most systems and has this fuzzy rule:

$$\text{IF } x = A \text{ AND } y = B \text{ THEN } z = k, \tag{2}$$

where k is a constant. The output of each fuzzy rule is a constant, and all latter terms of membership functions are represented by a single state spike. The final step is to adopt the center of gravity method to compute a clear output:

$$\text{COG} = \frac{\int_a^b \mu_A(x)x dx}{\int_a^b \mu_A(x) dx} = \frac{\sum_a^b \mu_A(x)x}{\sum_a^b \mu_A(x)}. \tag{3}$$

Figures 9 and 10 show the corresponding input and output membership functions of our fuzzy logic control system:

(A) The relative distance from the target to the Kinect is adopted to decide whether to stop, spin, or follow the target:

(1) *Stop*. 0-150 cm

(2) *Spin to Find the Target*. 150-250 cm

(3) *Follow the Target*. 250-450 cm

(B) The relative absolute angle deviation from the target to the Kinect:

A1: 0-13 degrees (small-angle deviation)

A2: 10-20 degrees (medium-angle deviation)

A3: 17-30 degrees (huge-angle deviation)

(C) Four-motor rotation output:

B1: 0% power (stop following or spinning)

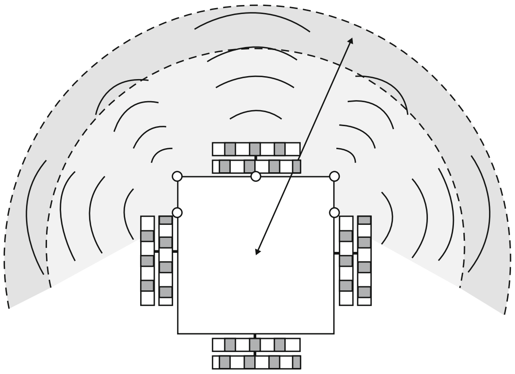


Figure 7: Top view of the critical boundary.

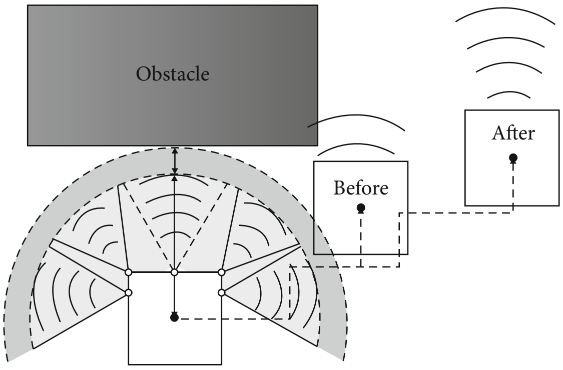


Figure 8: Comparison methods.

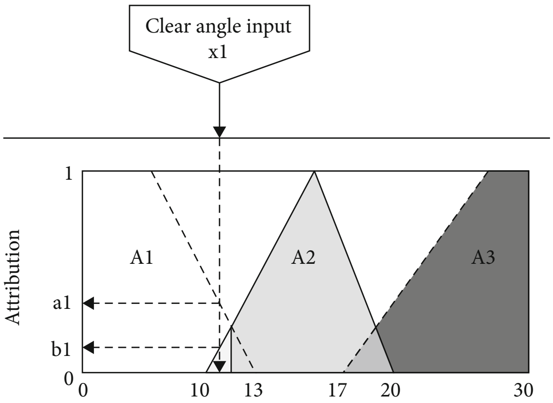


Figure 9: Input membership function.

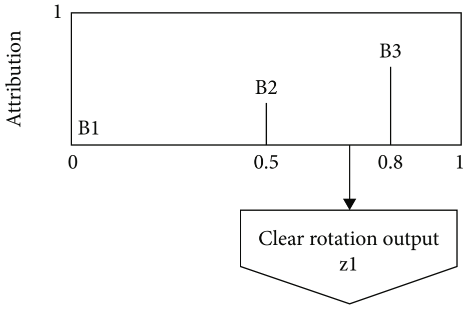


Figure 10: Output membership function.

B2: 50% power (slow following or spinning)

B3: 80% power (fast following or spinning)

(D) Fuzzy rules:

Rule 1: IF $x = \mathrm{A1}$ AND $y = \mathrm{B1}$ THEN $z = (\mathrm{C1}, \mathrm{D1})$. If the target angle deviation is small, stop turning or spinning.

Rule 2: IF $x = \mathrm{A2}$ AND $y = \mathrm{B2}$ THEN $z = (\mathrm{C1}, \mathrm{D2})$. If the target angle deviation is medium, slow turning or spinning.

Rule 3: IF $x = \mathrm{A3}$ AND $y = \mathrm{B3}$ THEN $z = (\mathrm{C2}, \mathrm{D2})$. If the target angle deviation is huge, fast turning or spinning.

4.2. Target User Tracking. The accurate position of a specific target user, whether from the front or the back of the mobile system, is computed by adopting the skeleton extraction function to calculate the distance and angle between the target users. The extracted skeleton data contain the following information:

(1) The user tracking state

(2) The user tracking ID

The fuzzy logic control system needs to receive several commands from the tracking subsystem, including the start commands of each of two functions and a cancel command for both. Generally, the user can give commands to the robot by speech, but voice recognition may also fail. Additionally, if the user forgets the commands, it causes annoying user experiences. In order to solve the problem, this study adapted gesture recognition. This is particularly suitable for our system. Figures 11 and 12 show the three chosen specific gestures.

4.3. Path Planning. The robot path planning problem is one of the most important research issues at this time. The requirements of the robot's path include avoiding the obstacles and shrinking the moving distance. The global map which contains all obstacle information is needed in some approaches, such as [24]. However, it is not practical in some situations in which obstacles may move or change directions. In this paper, we applied the elastic net-based optimizing policy on the path which originated from obstacle avoidance. The elastic net is firstly proposed for the traveling salesman problem in [25]. It is an effective method for optimizing the path. The basic idea of the elastic net is an elastic ring with several points. Each point can be attracted by the neighbors and the nearest city (in the traveling salesman problem). In our approach, the elastic net is simplified to an elastic line from the start point to the stop point, like in Figure 13. Each point can be attracted by the neighbors and repulsed by the nearest obstacle. Figure 14 shows the forces on a point of the elastic net. The point P is attracted by N1 and N2, and the attractive force is proportional to the distance. The attractive forces have a tendency to shrink the length of the robot's

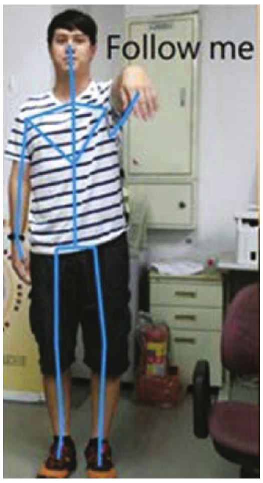


Figure 11: The gesture of "follow me."

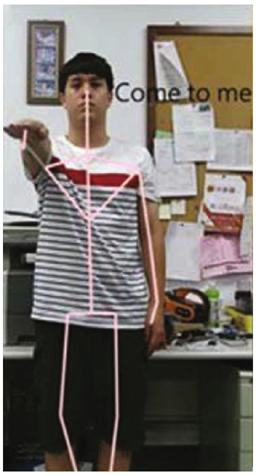


Figure 12: The gesture of "come to me."

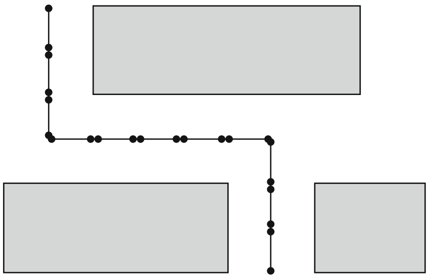


Figure 13: An elastic net for robot path planning.

path. The point P is repulsed by the nearest obstacle O1. The repulsive force is inversely proportional to the square of the distance. The repulsive force makes the path to be distant from the obstacle.

5. Experiment Results

5.1. Experiment Environment. Several experiments were designed to test the proposed system. These experiments comprise obstacle avoidance in three environments, when the mobile platform was separated from the robot, following the user, and coming to the user. The success rate of each case was calculated after each experiment 10 times.

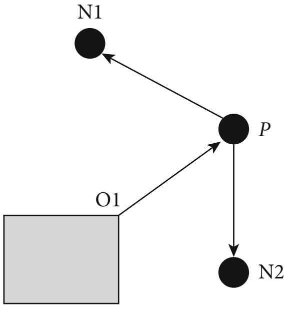


Figure 14: The forces on a point of the elastic net.

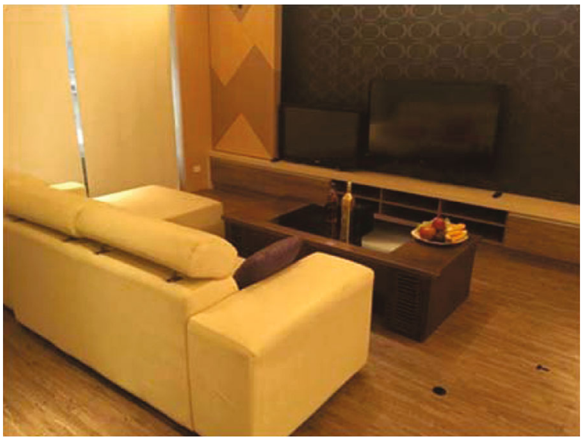


Figure 15: Living room.

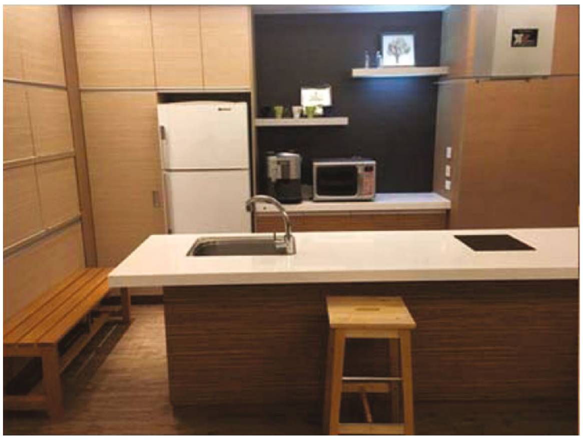


Figure 16: Kitchen.

5.1.1. Obstacle Avoidance Environment. Figures 15–17 show the three spaces in the simulated home environment. The experiments were performed in a real home environment with a living room, kitchen, and bedroom. The obstacles in the living room were a sofa, coffee table, and TV cabinet. The obstacles in the kitchen were a refrigerator and stove

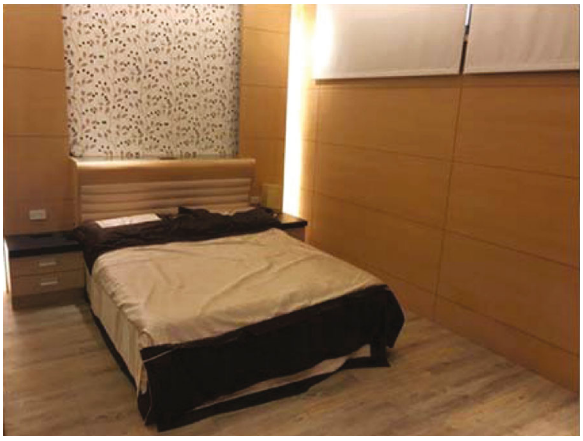


Figure 17: Bedroom.

Figure 18: Plane.

Figure 19: Cylindrical.

cabinet. The bedroom obstacles were a bed and bedside cabinet.

Figure 20: Sofa.

Figure 21: Desk.

5.1.2. Obstacle Types. In order to verify the effectiveness of robot obstacle avoidance, the experiments were designed with a variety of obstacles. Irregular shaped obstacles included the coffee table, sofa, and electric fan in the living room. Some obstacles were shaped like cylinders. Figures 18–21 show the different obstacles in each space.

5.2. Experimental Robot Mechanism. The robot was constructed according to the design. Figure 22 shows the main body, which is an aluminum alloy frame including a turntable with a height of 110–120 cm. The total weight is about 40 kg. The robot can carry an additional 5 kg and can be moved by the remote control of the handheld device. The arm part utilizes a connecting rod structure to increase the load-carrying capacity of the motor. The movement method adopts four DC motors with omnidirectional wheels of size 10.2 cm × 3.5 cm and a 6 mm hub adapter shaft, which can be rotated in situ or moved in all directions. The joint adopts five 25 kg servo motors and six 10.8 kg servo motors.

5.3. User Tracking Results. The experiments were first performed in the hall. The robot began to follow the user after it recognizes the gesture. In this experiment, the user turned at a corner to test the tracking performance of the robot. The "come to me" test was performed 30 times at different distances and failed 1 time at 10 meters, 5 times at 15 meters,

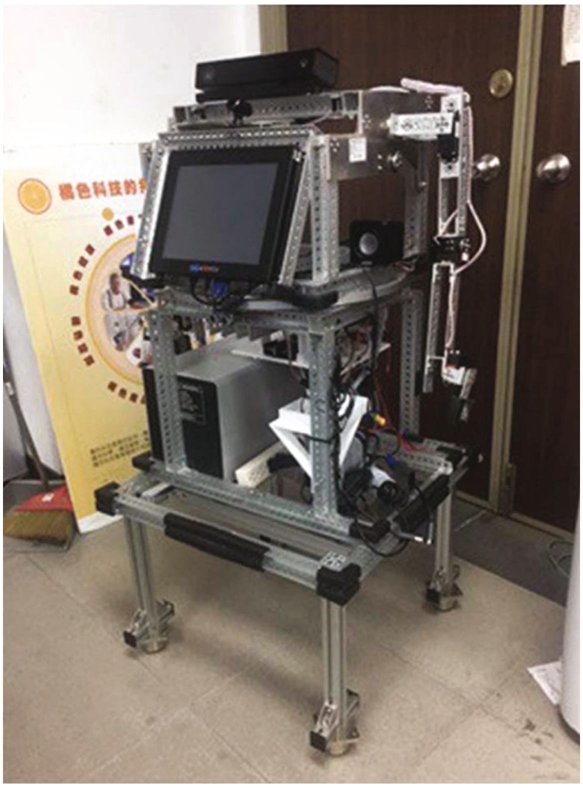


Figure 22: Real photo of a service robot.

Table 1: Result of user tracking: come to me.

Tracking length	5 m	10 m	15 m	20 m
Success	30	29	25	20
Failure	0	1	5	10
Success rate	100%	97%	84%	67%

Table 2: Result of user tracking: follow me.

Distance	2 m	3 m	5 m	7 m
Success	30	29	28	25
Failure	0	1	2	5
Success rate	100%	97%	94%	84%

and 10 times at 20 meters. The sensor could detect the user within 10 meters but failed increasingly beyond this distance. Table 1 lists the experimental results.

In the "follow me" experiment, the ability of the robot to follow has a good performance and can follow the distance of the subject by more than 5 meters. Table 2 lists the experimental results.

In the "follow me" experiment, the user gestured to the robot, and the robot began to follow the user, followed the user to turn, and continued to follow the user after turning. The experiment scenes are shown in Figures 23–26.

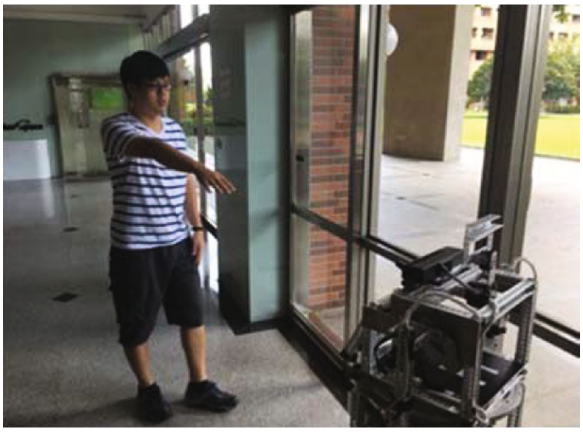


Figure 23: Gesture to the robot.

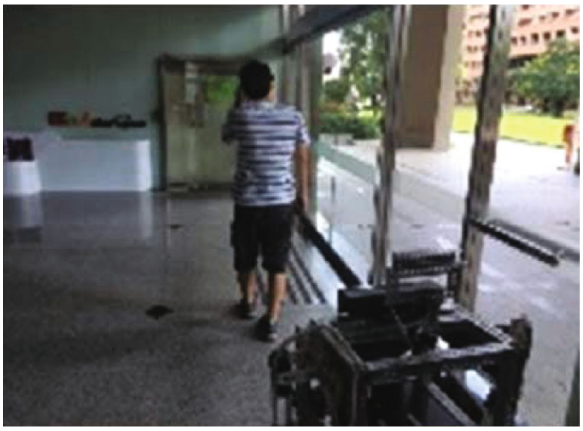


Figure 24: The robot follows the user.

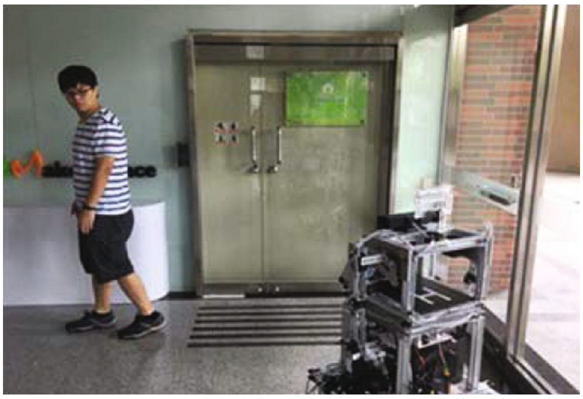


Figure 25: The robot turns to follow the user.

5.4. Obstacle Avoidance Results. The obstacle avoidance experiment was run with the living room, kitchen, and bedroom as the experimental areas. Table 3 lists the experimental results. The main test criterion is whether the robot can effectively avoid irregular shaped objects. Experimental results indicate that the robot avoided 100% of all irregular

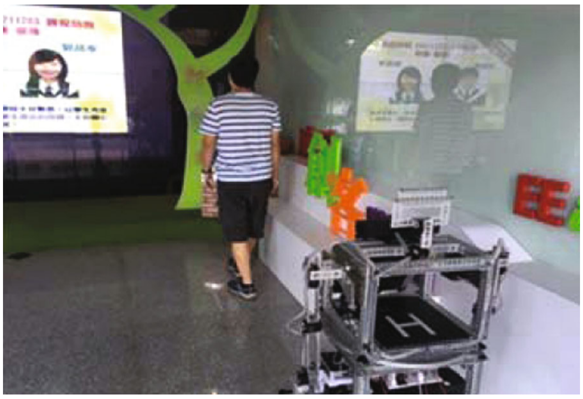


Figure 26: The robot keeps following after turning.

Table 3: Result of obstacle avoidance.

Obstacle	Parlor	Kitchen	Bedchamber
Plane	95%	95%	100%
Inclined plane	95%	95%	100%
Cylindrical	100%	100%	100%

obstacles. The experimental record was made into a show film, in addition to the actual test video in Demo Link.

6. Conclusions

This study develops a separable omnidirectional mobile system for a service robot. This system performs better than other tracking methods. The gesture recognition enables users to make a robot track themselves. Similar to the "follow" function of a tour-guide robot, the "come to me" function and gesture recognition allow users to call the robot, a feature that is especially useful in helping someone who is seriously injured. The tracking function can lock the object and lock the target if it returns. The relock ability can be strengthened in the future, by locking the same target using depth images or based on the clothing color and body sizes. In terms of remote control and obstacle avoidance, the proposed robot can patrol manually or automatically. Future work would be to strengthen the patrolling ability by making the robot start from one place to another destination and return to the origin. Additionally, traction wheels cause difficulty for a mobile platform to synchronize its steering direction with the 30 kg robot, causing real-time steering of the whole robot to fail, even with accurate motor control and calibration. Replacing the traction wheels with omnidirectional wheels in the future may improve the steering ability and speed of the robot. Conversely, to reduce the friction caused by the robot, the traction wheels could be constructed with low-friction materials such as plastic in the future. Although the mobile platform needs more friction to drag the load, high-friction materials such as rubber can be replaced to reduce the friction on the current omnidirectional wheels.

Data Availability

All used data has been included in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Steels and R. Brooks, *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*, Routledge, 2018.

[2] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism*, vol. 69, pp. S36–S40, 2017.

[3] B. F. Wu, P. Y. Chen, and C. H. Lin, "A new criterion of human comfort assessment for wheelchair robots by q-learning based accompanist tracking fuzzy controller," *International Journal of Fuzzy Systems*, vol. 18, no. 6, pp. 1039–1053, 2016.

[4] K. Siau, *Impact of Artificial Intelligence, Robotics, and Automation on Higher Education*, Twenty-third Americas Conference on Information Systems, Boston, 2017.

[5] S. Y. Chiang, C. A. Wei, and C. Y. Chen, "Real-time self-localization of a mobile robot by vision and motion system," *International Journal of Fuzzy Systems*, vol. 18, no. 6, pp. 999–1007, 2016.

[6] S. Yoshida, T. Shirokura, Y. Sugiura et al., "Robojockey: designing an entertainment experience with robots," *IEEE Computer Graphics and Applications*, vol. 36, no. 1, pp. 62–69, 2016.

[7] C. W. Chen, S. P. Tseng, Y. T. Hsu, and J. F. Wang, "Design and implementation of human following for separable omnidirectional mobile system of smart home robot," in *2017 International Conference on Orange Technologies (ICOT)*, pp. 210–213, Singapore, 2017.

[8] K. Y. Chen, C. H. Chen, C. Y. Lin, J. H. Tsai, and H. Y. Chung, "Development of optimal path planning based on ant colony and wireless sensor network localization techniques for an autonomous mobile service robot," in *2014 International Conference on Information Science, Electronics and Electrical Engineering*, vol. 2, pp. 954–959, Sapporo, Japan, 2014.

[9] I. Samejima, Y. Nihei, N. Hatao et al., "Building environmental maps of human activity for a mobile service robot at the "Miraikan" museum," in *Field and Service Robotics*, pp. 409–422, Springer, 2015.

[10] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: a review," *International Robotics & Automation Journal*, vol. 2, no. 3, article 00022, 2017.

[11] Y. Kunii, G. Kovacs, and N. Hoshi, "Mobile robot navigation in natural environments using robust object tracking," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1747–1752, Edinburgh, UK, 2017.

[12] W. J. Wang, H. G. Chou, Y. J. Chen, and R. C. Lu, "Fuzzy control strategy for a hexapod robot walking on an incline," *International Journal of Fuzzy Systems*, vol. 19, no. 6, pp. 1703–1717, 2017.

[13] W. J. Chen, B. G. Jhong, and M. Y. Chen, "Design of path planning and obstacle avoidance for a wheeled mobile robot," *International Journal of Fuzzy Systems*, vol. 18, no. 6, pp. 1080–1091, 2016.

[14] H. C. Huang and C. H. Chiang, "Backstepping holonomic tracking control of wheeled robots using an evolutionary fuzzy

system with qualified ant colony optimization," *International Journal of Fuzzy Systems*, vol. 18, no. 1, pp. 28–40, 2016.

[15] K. Li, X. Zhao, S. Sun, and M. Tan, "Robust target tracking and following for a mobile robot," *International Journal of Robotics and Automation*, vol. 33, no. 4, 2018.

[16] S. Zhang, L. Zhang, R. Gao, and C. Liu, "Mobile robot moving target detection and tracking system," in *Proceedings of The 7th International Conference on Computer Engineering and Networks — PoS(CENet2017)*, Shanghai, China, 2017.

[17] E. Benli, Y. Motai, and J. Rogers, "Human behavior-based target tracking with an omni-directional thermal camera," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, pp. 36–50, 2019.

[18] T. Feng, Y. Yu, L. Wu, Y. Bai, Z. Xiao, and Z. Lu, "A human-tracking robot using ultra wideband technology," *IEEE Access*, vol. 6, pp. 42541–42550, 2018.

[19] R. K. Williams and G. S. Sukhatme, "Observability in topology-constrained multi-robot target tracking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1795–1801, Seattle, WA, USA, 2015.

[20] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3D of pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, Brisbane, QLD, Australia, 2018.

[21] A. Al Arabi, P. Sarkar, F. Ahmed, W. R. Rafie, M. Hannan, and M. A. Amin, "2D mapping and vertex finding method for path planning in autonomous obstacle avoidance robotic system," in *2017 2nd International Conference on Control and Robotics Engineering (ICCRE)*, pp. 39–42, Bangkok, Thailand, 2017.

[22] P. K. Mvemba, B. SKGG, A. Lay-Ekuakille, and N. I. Giannoccaro, "Advanced acoustic sensing system on a mobile robot: design, construction and measurements," *IEEE Instrumentation & Measurement Magazine*, vol. 21, no. 2, pp. 4–9, 2018.

[23] Z. Ziaei, R. Oftadeh, and J. Mattila, "Global path planning with obstacle avoidance for omnidirectional mobile robot using overhead camera," in *2014 IEEE International Conference on Mechatronics and Automation*, p. 697, Tianjin, China, 2014.

[24] I. Hassani, I. Maalej, and C. Rekik, "Robot path planning with avoiding obstacles in known environment using free segments and turning points algorithm," *Mathematical Problems in Engineering*, vol. 2018, Article ID 2163278, 13 pages, 2018.

[25] R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method," *Nature*, vol. 326, no. 6114, pp. 689–691, 1987.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8814658, 12 pages
https://doi.org/10.1155/2020/8814658

Research Article

Developing Deep Survival Model for Remaining Useful Life Estimation Based on Convolutional and Long Short-Term Memory Neural Networks

Chia-Hua Chu, Chia-Jung Lee, and Hsiang-Yuan Yeh

School of Big Data Management, Soochow University, Taipei City 111, Taiwan

Correspondence should be addressed to Hsiang-Yuan Yeh; hyyeh.richard@gmail.com

Received 5 August 2020; Revised 16 October 2020; Accepted 20 November 2020; Published 7 December 2020

Academic Editor: Chao-Yang Lee

The application of mechanical equipment in manufacturing is becoming more and more complicated with technology development and adoption. In order to keep the high reliability and stability of the production line, reducing the downtime to repair and the frequency of routine maintenance is necessary. Since machine and components' degradations are inevitable, accurately estimating the remaining useful life of them is crucial. We propose an integrated deep learning approach with convolutional neural networks and long short-term memory networks to learn the latent features and estimate remaining useful life value with deep survival model based on the discrete Weibull distribution. We conduct the turbofan engine degradation simulation dataset from Commercial Modular Aero-Propulsion System Simulation dataset provided by NASA to validate our approach. The improved results have proven that our proposed model can capture the degradation trend of a fault and has superior performance under complex conditions compared with existing state-of-the-art methods. Our study provides an efficient feature extraction scheme and offers a promising prediction approach to make better maintenance strategies.

1. Introduction

With the advance of Internet of Things (IoT) technology and its applications to the industrial environments, data analytics methods can be applied to the mechanical equipment health and performance. In fact, any machine breakdown may lead to a huge loss on production yield. However, sometimes even a professional and experienced engineer cannot find where the fault is and also cannot figure out the main cause of the malfunction. In this case, the company has no choice but to suspend the production line for the thorough examination, which is certainly one of the disastrous situations for the company's business. In order to keep the high reliability and stability of the production line, reducing the downtime for fixing malfunction and the frequency of routine maintenance is necessary. The earliest maintenance technique takes place only when breakdown happens, which is called breakdown maintenance or run-to-failure maintenance. Later, companies tend to have time-based preventive maintenance. It means that engineers perform a preventive maintenance periodically, in spite of the status of the machine even though it is in healthy status. Preventive maintenance will cause a lot of cost and become a major expense of many companies. In order to save the cost, another maintenance strategy such as condition-based maintenance (CBM) is figured out to solve the situation. CBM proposes to reduce the number of unnecessary regular preventive maintenance and improves the reliability of machine by implementing maintenance only when there is an evidence that an exception occurs [1, 2]. Because it is effective in saving the cost to the companies, CBM has been more and more popular. The prognostics and health management (PHM) is one of the major tasks in CBM. The core of PHM is the estimation of the remaining useful life (RUL) of machines based on the collected information of the historical and ongoing degradation trends [2–4]. The flowchart of PHM contains five major processes as shown in Figure 1, including data acquisition, signal processing, diagnostic, prognostic, and maintenance decision. Data acquisition is the first process of PHM which is composed of sensors, data transmission, and data storage devices.

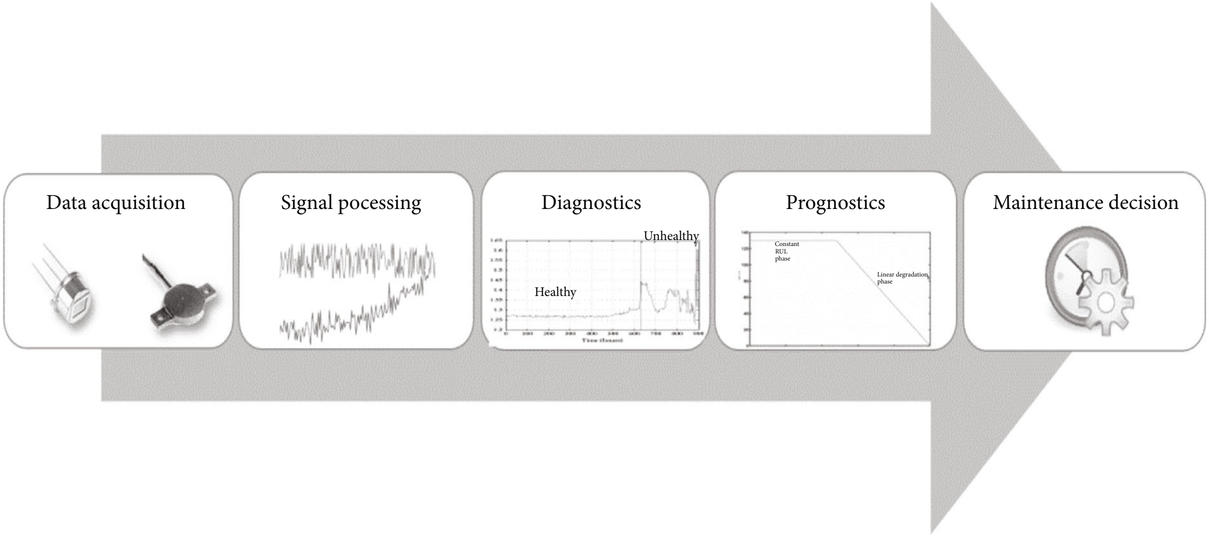


Figure 1: Flowchart of PHM processes.

Different kinds of sensors are used to collect different types of data, which are related to the health condition and able to reflect the degradation process of the monitored machine. Signal processing's task is to extract useful information from the data acquired from the previous step. Diagnostic is a process to divide the machine's whole lifetime into different health status. Prognostic is aimed at estimating the time length from current time to when it requires repair or replacement, which is also the definition of RUL. Maintenance decision is the final process in PHM and is used for analyzing the outputs from diagnostics and prognostics. If we can predict RUL, we are able to propose a strategy about scheduling maintenance, avoiding unplanned downtime, and optimizing operating efficiency and frequency to save the most cost. However, knowing that the machine and components' degradation are inevitable, the challenge of proper scheduling grows with the complexity of machines. One of the key problems in predictive maintenance is the prediction of the equipment failures should be early enough so that the proper maintenance could be scheduled before it happens. Therefore, predictive maintenance is based on the continual monitoring of the equipment in order to determine the right maintenance actions at the right times. The organization of this paper is as follows. Related works on RUL prediction is introduced in Section 2. Our proposed deep learning approach with a survival model is described in Section 3. In Section 4, the experimental results and evaluations are compared with existing state-of-the-art methods and show the effectiveness of the proposed approach. We conclude the contributions and limitations of our approach in Section 5.

2. Related Works

Generally, the methods of the RUL estimation problems can be categorized into model-based, data-driven, and hybrid approaches. Model-based prediction applies a physical model of the system for degradation [5]. This approach can be further divided into microlevel models [6] and macrolevel models [7]. Microlevel models need to consider the assumptions and simplifications in uncertain environments. A macrolevel model is constructed under different operational conditions of the physical system which includes the relationships among input variables, state variables, and system outputs. However, model-based methods require a large amount of prior knowledge. The physical models are difficult to build under many components which limit the effectiveness of the methods. The data-driven approaches detect the state of the system via large number of sensor monitoring, which are more suitable for the complex system and do not require a comprehensive understanding of the physical understanding [8]. Currently, high-dimensional data collected in real-life PHM applications makes it difficult to directly discover the trends for the prognostic algorithm. There are various operational conditions and health states in the same type of the system which may cause different degradation processes as well as unit-to-unit variability (UtUV) [9]. This situation brings difficulty to RUL estimation. Javed et al. contributed a data-driven prognostics approach based on extreme learning machine (ELM), which is able to model degrading states without assuming a homogeneous pattern [10]. Liu and Chen combined indirect health indicator (HI) and the smooth monotonic signals from sensory data and multiple Gaussian process regression (GPR) models to achieve the RUL prediction [11]. Previous works constructed a model based on Box–Cox transformation (BCT) and Monte Carlo (MC) simulation to predict the battery RUL [12]. Khelif et al. developed a procedure to estimate the RUL directly from sensor values using the support vector regression method which models the direct relationship among sensor values or health indicators [13]. However, the dramatic changes and variations of the indicators make a well-trained prediction model that may not be suitable for the practical applications. The traditional feature extraction method is hard to get high-level representations from measurements, and the poor prognostic performance may be achieved. Therefore, capturing hidden patterns from high-dimensional data efficiently is a necessary procedure in the feature extraction procedure [14]. Hybrid approaches are to complement the superiority of model-based and data-driven methods [15]. However, it still remains a challenging work to utilize the advantages and to avoid the disadvantages of both approaches.

Recently, data-driven prediction methods focused on the flexible deep learning models to capture useful information from high-dimensional data efficiently. Zhang et al. employed a multiobjective evolutionary algorithm with traditional deep belief networks (DBN) for RUL estimation in prognostics [16]. Sequence learning methods such as the hidden Markov model (HMM) were applied to capture time series information from the sensory data [17–19]. In the model, each state can only depend on the immediately previous state and the hidden states must be drawn from the discrete space. However, modeling long time dependencies may lead to high computational complexity while the set of the hidden stages grows larger. Recurrent neural network (RNN) can model time sequence data as well, and some work applied it to estimate RUL [20]. But RNN has its limitation to capture the long-term time dependencies, the gradients propagated over many hidden layers tend to either vanish or explode [21]. Long short-term memory (LSTM) network is a significant branch of RNN, can learn long-term dependencies, and avoid gradient disappearance and explosion in long sequence training. Previous studies have shown that LSTM networks can expose hidden patterns from the sequential sensor data with multiple operating conditions, fault, and degradation models [22, 23]. Some new approaches based on LSTM, such as bidirectional long short-term memory (BLSTM) network [24] and vanilla LSTM [25], were proposed. Wang et al. proposed a transfer learning algorithm based on BLSTM networks, which can be first trained on the datasets and fine-tuned the model with a different but related dataset [26]. Recent works also enhanced LSTM networks with attention mechanism and generative adversarial network (GAN) to improve the interpretability and accuracy of the deep networks [27–29]. The convolutional neural network (CNN) architecture has been proven to be effective for extracting abstract information on multichannel sequential sensor data [30–32]. Although LSTM networks enable us to build and capture long-term time dependencies, its feature extraction capabilities are marginally lower than CNN [33]. CNN can extract the spatial feature while LSTM can learn temporal features. Therefore, it is better to learn temporal features from the slow inherently long-term degradation process by combing those two structures. Recent paper proposed a deep neural network structure using both LSTM and CNN which can be combined in a serial or parallel manner to improve the accuracy of the RUL prediction of the equipment [34, 35].

Most of the prior works have focused on the RUL prediction problem which present one numeric RUL value only. However, it is nearly impossible to find an approach that can predict RUL exactly the same as the real one. If the variance is large, it is hard to have confidence on the predicted result. The RUL prediction problem is also similar to the survival analysis which is commonly used to model time-to-death events in the healthcare domain [34]. For example, the model predicts the failure will happen in 8 days with 80% probability is much better than predict 10 days until the failure. Martinsson proposed the Weibull time-to-event recurrent neural network, which is a simple framework for time series prediction of the time to the next event applicable [36]. Aggarwal et al. used the Weibull distribution assumption on the time-to-failure event with a linear hazard rate corresponding to the linear degradation model that most of the literature makes [34].

Due to the complicated environments in real-life PHM applications, monitoring sequential sensor data is subjected to the operating conditions and fault modes for the prognostic algorithm. The existing data-driven methods often rely on the sensor measurements as a whole data for training that may cause less effect and bias. To cope with this issue, there is a great potential to improve the RUL estimation by extracting latent patterns from partial information that is a necessary procedure to capture useful information from high-dimensional sequential sensor data. It is also valuable to estimate how much time is left of the equipment and the probability of a failure together. Therefore, we integrate CNN and LSTM with a deep survival model to enhance the ability of feature extraction and capture the degradation trend of a fault with a reasonable prediction horizon.

3. Materials and Methods

The overall workflow of our approach is shown in Figure 2. Different sensors may have different physical meanings and numerical ranges. In order to eliminate the influence of ranges of value, we first apply a min-max normalization method as feature scaling to adjust the range of sensor values between 0 and 1. Second, the training data and test data are prepared with a sliding time window (TW) to generate the sequential samples. Third, we use 1D temporal convolutions to learn hidden patterns in those sequential samples without any interference from the other sensor values. Forth, the extracted temporal patterns from the 1D convolution would be fed into LSTM networks to learn the long short-term time dependencies. Fifth, we use both regression and survival analyses with the discrete Weibull distribution to estimate the RUL and failure probability in the training phase. Finally, we can predict the RUL and the probability of a failure with test data in the trained model in the testing phase.

3.1. Data Preparation with Time Window. The input sequential sensor data from an engine are assumed to be a matrix $X(n) = [x^1, x^2, \cdots, x^{L_s}]$ with k sensors (measurements), where n denotes engine ID and L_s denotes the last observed cycle or the cycle that fault occurs in Figure 3. A sliding time window strategy is adopted to generate the temporal sequence data except sampled at a single time step which may conduct better feature extraction efficiently. Taking $X(n)$ as an input and extracting sequential $X_i(n)$ as Equation (1) by sliding the fixed time window (TW) with length m, this can be presented in Figure 3:

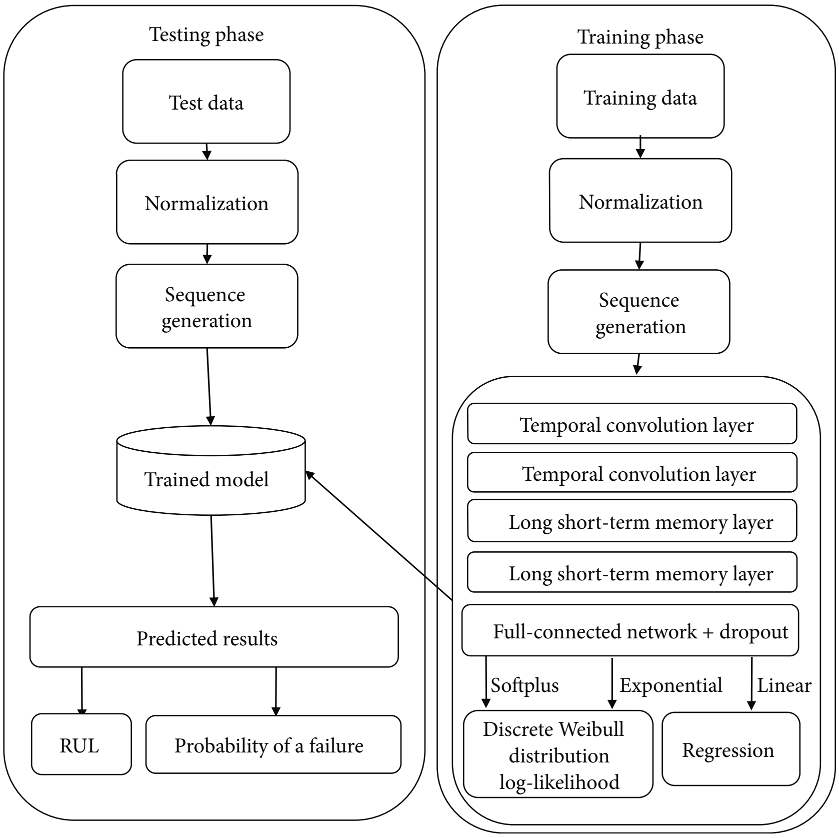


Figure 2: The workflow of our approach.

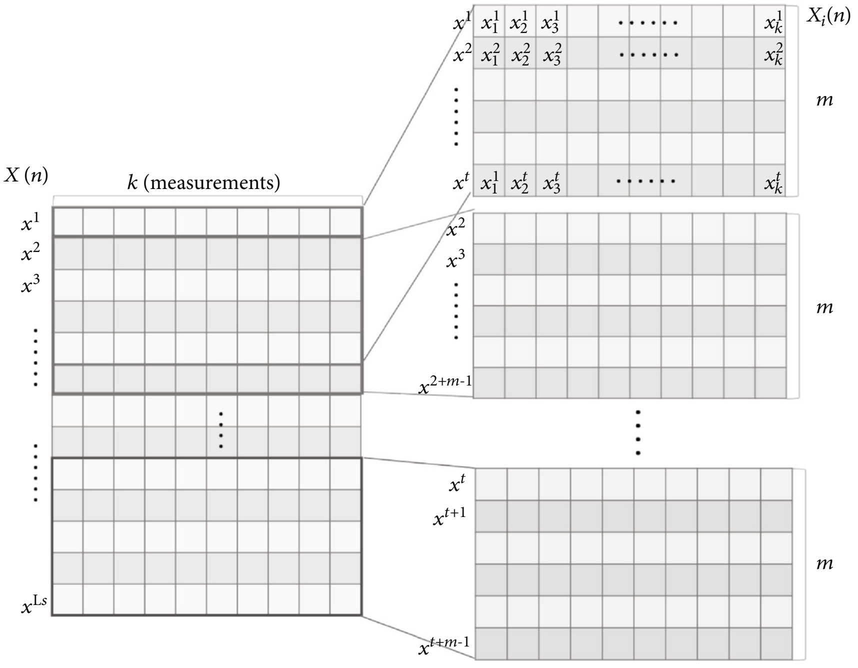

$X(n)$
k (measurements)
x^1
x^2
x^3
x^{Ls}
$X_i(n)$
x^1 x_1^1 x_2^1 x_3^1 ...... x_k^1
x^2 x_1^2 x_2^2 x_3^2 ...... x_k^2
x^t x_1^t x_2^t x_3^t ...... x_k^t
m
x^2
x^3
x^{2+m-1}
m
x^t
x^{t+1}
x^{t+m-1}
m

Figure 3: Sliding time windows.

$$X_i(n) = x^{t:t+m-1} = \left[x^t, x^{t+1}, x^{t+2}, \cdots, x^{t+m-1}\right], \quad (1)$$

where $x^t = [x_1^t, x_2^t, \cdots, x_k^t]$ represents a k-dimensional array of measurements at time (cycle) t through a TW size of m. Therefore, the size of each array extracted each time by TW is $m \times k$ (TW × numbers of sensors), and the number of arrays is L_s-m (lifetime cycles-TW).

3.2. Temporal Convolutional Layer. To deal with sequential information more effectively, a CNN layer can be used to extract abstract and high-level features before LSTM layers. The temporal convolutional layer consists of three layers, starting with 1D-convolution, 1D-max-pooling, and followed by activation function. 1D convolution represents a filtering window length and moves towards the depth across the data. We consider that the input data matrix size is $m \times k$ array in $X_i(n)$, and there are d kinds of the feature detectors with w kernel size. So, each feature detector has to move $(m - w) + 1$ times and generates the set of feature detector regions X_{dw}. Then, we add 1D-convolution weight kernel $\vec{w}$ and bias b so the convolution operation is shown as

$$C_i = f\left(X_{dw} \bigotimes \vec{\mathrm{w}} + b\right). \quad (2)$$

Where $\bigotimes$ denotes the Hadamard product (element-wise product) and f represents the nonlinear activation function, *ReLu*. Accordingly, the output feature maps of a 1D-convolution layer will be the size of $((m - w) + 1) * d$. Since we have multiple temporal convolution layers, we let $C^{(l-1)}$ and $C^{(l)}$ be the input and output of the lth layer, respectively. We denote the jth feature map of layer l as $C_j^{(l)}$ which can be computed by

$$C_j^{(l)} = f\left(\sum_i C_i^{(l-1)} \bigotimes \vec{w}_{i,j}^{(l)} + b_j^{(l)}\right). \quad (3)$$

3.3. Long Short-Term Memory. LSTM cell state relies on three control gates: input gate, forget gate, and output gate. Input gate controls the extent to which incoming data flows into the cell. Forget gate judges which data from the foregoing cell state to be taken in consideration or be ignored. Output gate decides whether the value in the cell is used to compute the output. In the LTSM layer, it performs multiple internal equations as described below as

$$f_l = \sigma\left(W_f \bullet \left[h_{l-1}, C_j^{(l)}\right] + b_f\right), \quad (4)$$

$$i_l = \sigma\left(W_i \bullet \left[h_{l-1}, C_j^{(l)}\right] + b_i\right), \quad (5)$$

$$\widetilde{Z_l} = \tanh\left(W_c \bullet \left[h_{l-1}, C_j^{(l)}\right] + b_c\right), \quad (6)$$

$$Z_l = f_l \otimes Z_{l-1} + i_l \otimes \widetilde{Z_l}, \quad (7)$$

$$o_l = \sigma\left(W_o \bullet \left[h_{l-1}, C_j^{(l)}\right] + b_o\right), \quad (8)$$

$$h_l = o_l \otimes \tanh(Z_l), \quad (9)$$

where f_l denotes the forget gate, and its main function is to neglect the data from the previous LSTM cell state. $\sigma(\,)$ is an activation function *sigmoid*. W_f is the weight matrix of the forgot gate, h_{l-1} denotes the short-term state of previous layer in the LSTM cell, the feature map of layer $C_j^{(l)}$ is the input of LSTM cell, and b_f is the bias vector of forget gate. The input gate is composed of two parts, i_l is a vector that determines which data in the short-term state h_{l-1} is used to update the new cell state. After being selected by i_l, $\widetilde{Z_l}$ will be added to the long-term cell state and tan h is an activation function. W_i and W_c denote the weight matrixes and both b_i and b_c denote the bias vectors of the input gate. Then, the forget gate and the input gate will be used to update the long-term state of the previous LSTM cell. The output gate o_l is also composed of variables where W_o denotes the weight matrixes and b_o denote the bias vectors.

Finally, the LSTM layer connects to the fully connected layer for estimating the output target RUL value. Dropout technique is a regularization technique which randomly drops the hidden nodes with a given probability during training. It forms neural networks with different architectures in parallel and then takes an ensemble of them to prevent coadaptation. In order to alleviate the overfitting problems, the dropout is used between the final LSTM layer and the first fully connected layer [37].

3.4. Loss Function. Survival analysis is also called time-to-event analysis that is a subfield of statistics for analyzing the expected time duration until one or more events happen [38]. This approach calculates the probability of the subject to 'survive' the number of days or cycles [39]. One of the most commonly used distributions in the survival analysis is the discrete Weibull distribution which can be presented as Equation (10). The time-to-failure is modeled with a random variable T giving the probability of failure time between t and $t + 1$. The probability mass function (PMF) of a discrete random variable is characterized by two parameters: alpha (α) is a scale parameter that denotes that the expected value and mode of the distribution are positioned in time, while the parameter beta (β) is an indicator of the shape as well as the variance of our prediction.

$$\Pr(t \le T \le t + 1) = e^{-(t/\alpha)^\beta} - e^{-((t+1)/\alpha)^\beta}. \quad (10)$$

We have to utilize a special log-likelihood as loss-function, called the discrete Weibull distribution log-likelihood [34]. The discrete Weibull distribution log-likelihood punishes the model for predicting high probabilities of failures occurring during the lifetime without failures for all samples. In addition, the discrete Weibull distribution log-likelihood will reward distributions that give high probabilities of the event happening at that point in time for samples where the failure time is known. The discrete Weibull distribution log-likelihood can be defined as follows:

$$d(y) = \Lambda(y+1) - \Lambda(y) = \left(\frac{y+1}{\alpha}\right)^{\beta} - \left(\frac{y}{\alpha}\right)^{\beta}, \tag{11}$$

$$\begin{aligned} \log\left(\mathfrak{L}_d\right) &= \bar{u} \bullet \log\left(e^{d(y)} - 1\right) - \Lambda(y+1) = \left(\frac{y+1}{\alpha}\right)^{\beta} - \left(\frac{y}{\alpha}\right)^{\beta} \\ &= \alpha^{-\beta}\left((y+1)^{\beta} - y^{\beta}\right), \end{aligned} \tag{12}$$

where y denotes the time-to-event value (cycle) and u denotes either a 0 or 1 machinery health event indicator. Since it is an average value, we express it by $\bar{u}$. In each training step, we apply two types of loss functions. We apply linear activation function for the output value in regression analysis approach and use mean squared errors (MSE), which measures the average squared difference between the estimated RUL values and the true RUL value as loss function. On the other hand, the activation layer of the discrete Weibull distribution is a costumed function that is set to use an exponential function for alpha and *softplus* function for beta [34]. We use the discrete Weibull distribution log-likelihood as loss function in failure probability in the survival analysis approach and estimate Weibull parameters be the outputs of the layer giving us a distribution of the training data.

3.5. Performance Evaluation. For the sake of comparability with other existing state-of-the-art methods, the same metrics are used to evaluate the performance. While using the model to predict the RUL with regression and survival analysis approaches, there is an error between the predicted RUL and the actual RUL called root mean square error (RMSE) as Equation (13). The late prediction might delay the schedule of the proper maintenance operations, and too early prediction might not be harmful but still wastes more maintenance resources. Since the key aspect is to avoid the failure, early prediction is generally more desirable than late prediction. The scoring function as Equation (14) penalizes late predictions more than early predictions to evaluate the model. In addition, we also calculate MAE as Equation (15) and R^2 (R squared) coefficient of determination which is a statistical measure of how well the predictions approximate the real data points for more comparison between the two analysis approaches. The higher R^2 value means more information about the fit of a model can be explained.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \bullet \sum_{i=1}^{N} \left(\mathrm{RUL}_{\mathrm{True}} - \mathrm{RUL}_{\mathrm{Predicted}}\right)^2}, \tag{13}$$

$$\mathrm{score} = \sum_{j=1}^{N} \begin{cases} e^{-\left(\mathrm{RUL}_{\mathrm{True}} - \mathrm{RUL}_{\mathrm{Predicted}}\right)/13} - 1, & \text{if } \left(\mathrm{RUL}_{\mathrm{True}} - \mathrm{RUL}_{\mathrm{Predicted}}\right) < 0, \\ e^{-\left(\mathrm{RUL}_{\mathrm{True}} - \mathrm{RUL}_{\mathrm{Predicted}}\right)/10} - 1, & \text{if } \left(\mathrm{RUL}_{\mathrm{True}} - \mathrm{RUL}_{\mathrm{Predicted}}\right) \geq 0, \end{cases} \tag{14}$$

$$\mathrm{MAE} = \frac{1}{N} \bullet \sum_{i=1}^{N} \left|\mathrm{RUL}_{\mathrm{True}} - \mathrm{RUL}_{\mathrm{Predicted}}\right|. \tag{15}$$

4. Results and Discussion

4.1. Datasets. We adopt the dataset from the Commercial Modular Aero-Propulsion System Simulation (CMAPSS) dataset provided by NASA [40]. The CMAPSS dataset is well known in the PHM community and has been widely used for evaluating predictive performance. The dataset consists of simulated degradation data of turbofan engines generated with the thermodynamical simulation model. This dataset includes four subdatasets with different operating conditions and fault modes. A description of four subdatasets is shown in Table 1. A fleet of engines in the FD001 dataset suffered the high-pressure compressor failure under a single operating condition. For FD002, the sample suffered the high-pressure compressor failure under six operating conditions. In the FD003 situation, the sample suffered high-pressure compressor and fan degradations under a single operating condition, while in FD004, the sample suffered under six operating conditions. There are 26 columns in each subdataset, including engine unit number, number of cycle, three operational settings, and 21 sensor measurements. Each subdataset is divided into a training set, a testing set, and actual RUL corresponding to the test data. The different subsets have different numbers of engines with various operational cycles. As the operating time increases, the engines start to degrade until a failure occurs in the training datasets, while the degradation in the test datasets ends prior to the occurrence of a failure. The purpose of the proposed algorithm is to predict the RULs of the test datasets, and the true RUL targets of the test datasets are also provided to calculate the prognostic performance.

4.2. Performance of Regression and Survival Analyses. We randomly select 80 percent of the samples from the training set to train the models and the remaining 20 percent of the instances is used as the validation set to select the parameters in the training phase. We predict the RUL from the test data using a trained model and denote the performance comparing with state-of-the-art methods in Table 2. In Table 2, the italicized numbers denote the top 3 ranked results among those methods. The first three are regression-related algorithms including the multilayer perception (MLP) [41], support vector regression (SVR) [42], and relevance vector regression (RVR) [43], and the others are deep neural network related. The deep learning methods show better performance than the traditional machine learning methods. Our proposed approach achieves the lowest RMSE values and scoring function based either on regression analysis or on the discrete Weibull distribution for the FD002 and has superior performance for scoring function in FD004. We perform significantly reduce 5.19 and 0.15 in terms of RMSE and show improvement $1.75^{*}10^{3}$ and $1.11^{*}10^{3}$ in terms of scoring function, for FD002 and FD004, respectively. The samples working in multimodal switching in FD002 and FD004 datasets are more challenging for obtaining accurate prediction results. Our proposed network structure is able to find hidden patterns, and the prediction capability of the proposed method is better than the existing RUL prediction methods under the complex conditions. According to the

Table 1: The description of the CMAPSS dataset.

Subdataset	FD001 (training/test)	FD002 (training/test)	FD003 (training/test)	FD004 (training/test)
Engines	100/100	260/259	100/100	249/248
Operating condition	1	6	1	6
Fault mode	1	1	2	2

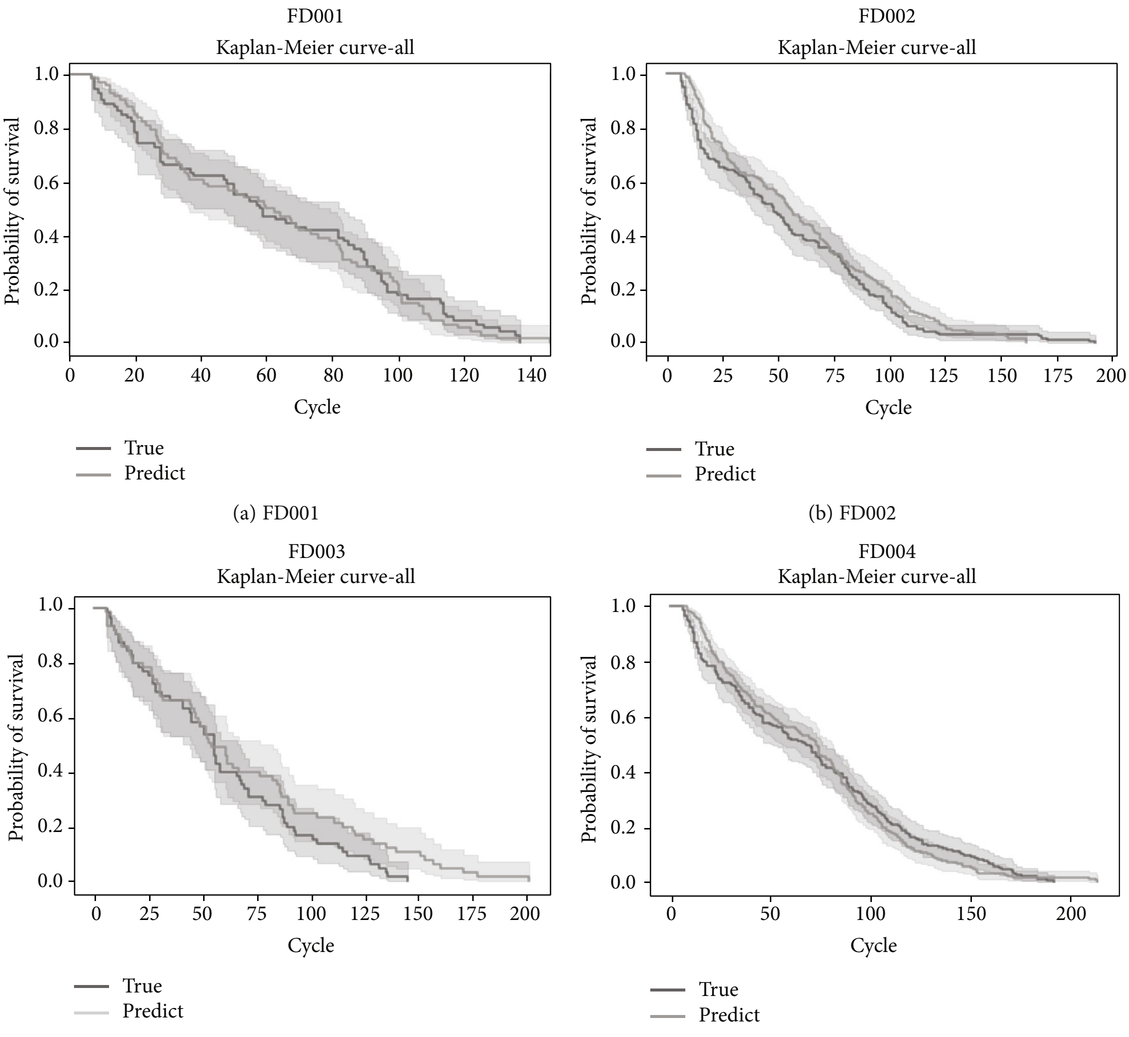
Table 2: The performance comparing our approach and state-of-the-art methods.

Methods	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
MLP [41]	37.56	1.80×10^4	80.03	7.80×10^6	37.39	1.74×10^4	77.37	5.62×10^6
SVR [42]	20.96	1.38×10^3	42.00	5.90×10^5	21.05	1.60×10^3	45.35	3.71×10^5
RVR [43]	23.80	1.50×10^3	31.30	1.74×10^4	22.37	1.43×10^3	34.34	2.65×10^4
CNN [31]	18.45	1.29×10^3	30.29	1.36×10^4	19.82	1.60×10^3	29.16	7.89×10^3
LSTM [22]	16.14	3.38×10^2	24.49	4.45×10^3	16.18	8.52×10^2	28.17	5.55×10^3
ELM [10]	17.27	5.23×10^2	37.28	4.98×10^5	18.47	5.74×10^2	30.96	1.21×10^5
DBN [15]	15.21	4.18×10^2	27.12	9.03×10^3	14.71	4.42×10^2	29.88	7.95×10^3
MODBNE [16]	15.04	3.34×10^2	25.05	5.59×10^3	*12.51*	4.22×10^2	28.66	6.56×10^3
RNN [19]	*13.44*	3.39×10^2	24.03	1.43×10^4	13.36	3.47×10^2	24.02	1.43×10^4
DCNN [32]	*12.61*	2.74×10^2	22.36	1.04×10^4	*12.64*	2.84×10^2	23.31	1.25×10^4
BiLSTM [26]	13.65	2.95×10^2	23.18	4.13×10^3	13.74	3.17×10^2	24.86	5.43×10^3
Aug+CNN+LSTM [30]	23.57	1.22×10^3	20.45	3.10×10^3	21.17	1.30×10^3	*21.03*	4.00×10^3
DAG [35]	*11.96*	2.29×10^2	*20.34*	2.73×10^3	12.46	5.35×10^2	*22.43*	3.37×10^3
CNN+LSTM w/regression	14.04	3.10×10^2	*15.15*	1.08×10^3	14.62	3.25×10^2	*21.92*	2.26×10^3
CNN+LSTM w/Weibull	13.98	2.31×10^2	*15.77*	9.79×10^2	15.55	6.72×10^2	23.05	3.67×10^4

defining of RMSE in Equation (13), the extrapolated RUL value as shown in FD001 and FD003 leads to little larger RMSE in the results. Although those subsets did not reach the best performance in RMSE, the score values are close to the best one. The experimental results also show that the integrated deep learning-related methods such as DAG and our method get better performance than previous single CNN or LSTM methods. Based on the comparison of the scoring function evaluation criteria, it can be seen that our proposed method appears in the top 3 ranked results among all four benchmark datasets, but the DAG method get three out of four in the experimental results. It shows that our proposed method can avoid failure with the early prediction.

One engine unit may have its own historical sensor data, and we apply the fixed sliding time window to scan the historical data to generate the several sequential sensor data. We estimate the RUL and its probability of failure corresponding to each sequential data and then construct the Kaplan-Meier survival curve. Figure 4 shows the predicted and the actual degradation process of the four randomly selected engines in each subdataset from the test data with Kaplan-Meier curve. We can find that the engines have 20% probability to keep running after 100 cycles, namely, the engines have 80% probability that it will have failure after operating 100 cycles in Figure 4(a). The survival probability of these engines decreases by 10% ~20% after every 20 cycles. From FD002 in Figure 4(b) and FD004 in Figure 4(d), the degradation trends have a fast decrease at the beginning and become steady when left 10% probability of availability, which is very close to failure. As for FD003 in Figure 4(c), there is a plateau in the right extrapolated tail of the Kaplan-Meier curve and it may cause the error between the predicted and true values to become bigger with cycles. This is the reason that RMSE of FD003 cannot surpass other approaches.

We describe the difference of our7 approach based on the discrete Weibull distribution and regression analysis with the linear model to predict the failure cycle. In Figures 5–8, we randomly selected four engines in each subdataset as examples and showed Kaplan-Meier curve with confidence intervals and probabilities based on the discrete Weibull distribution and RUL value based on linear regression analysis under time cycles. The results in Figures 5–8(b) have more variance and more error at the beginning of the estimation in the regression analysis approach, especially when the conditions become more complicated. It is easy to find that the prediction errors are greater in the early stage of degradation than in the late stage of degradation in the regression analysis approach. Due to the late stage with more sequential information comparing to the early stage, the predicted results can get better performance. The traditional RUL methods based on the regression analysis might lead to inconsistent



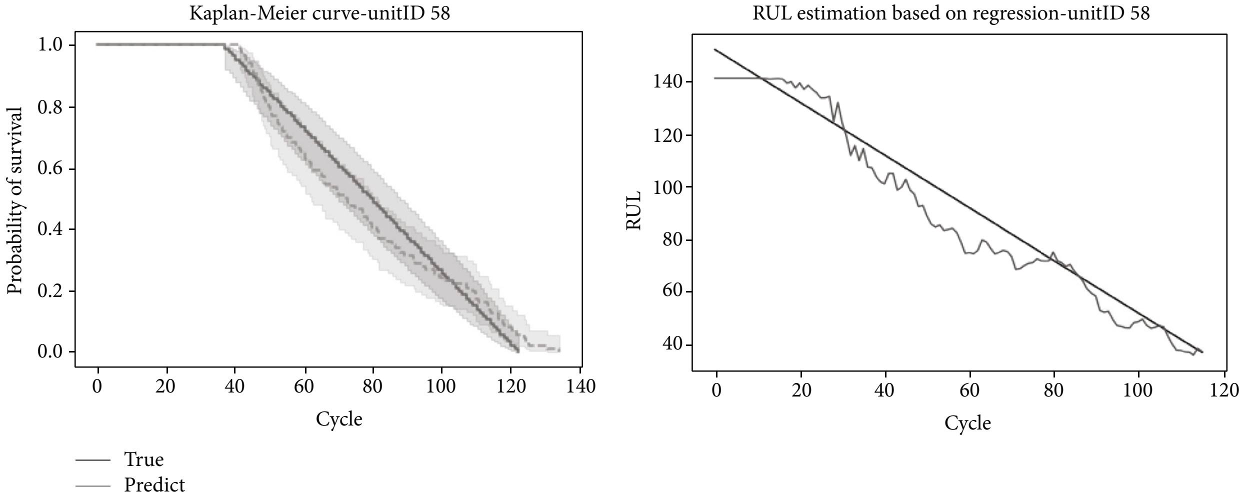
(a) FD001

(b) FD002

(c) FD003

(d) FD004

Figure 4: Results of the discrete Weibull distribution with Kaplan-Meier curve for all engines in each subdataset.



(a)

(b)

Figure 5: Result of engine ID 58 in FD001 (a) based on the Weibull distribution and (b) based on regression.

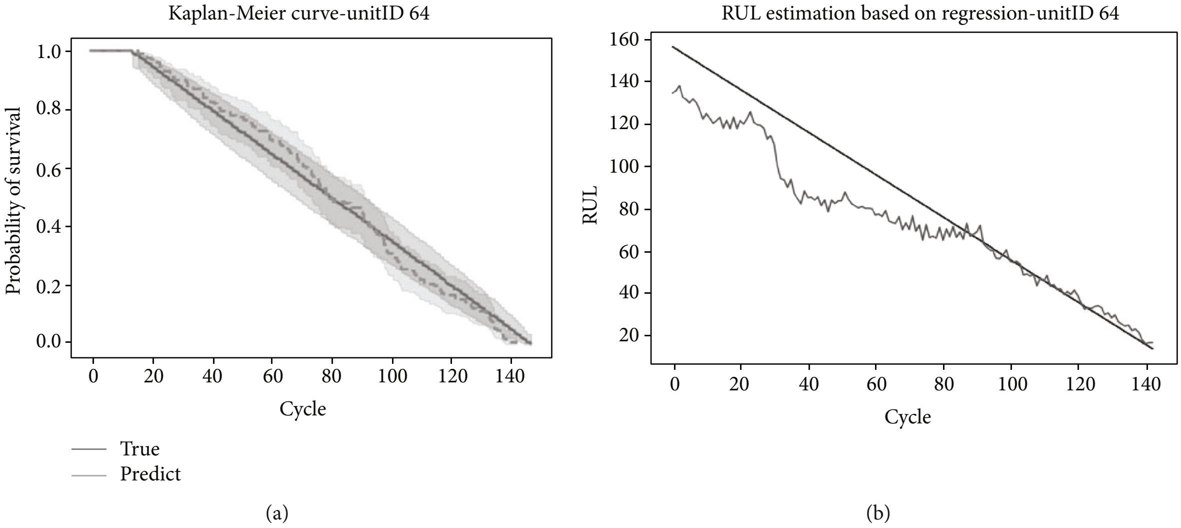


Figure 6: Result of engine ID 64 in FD002 (a) based on the Weibull distribution and (b) based on regression.

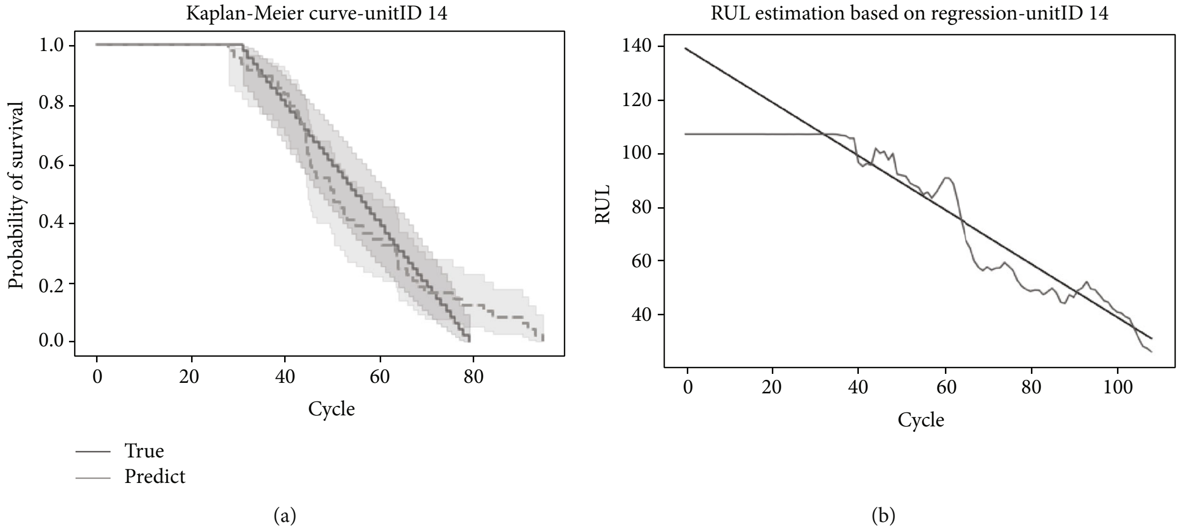


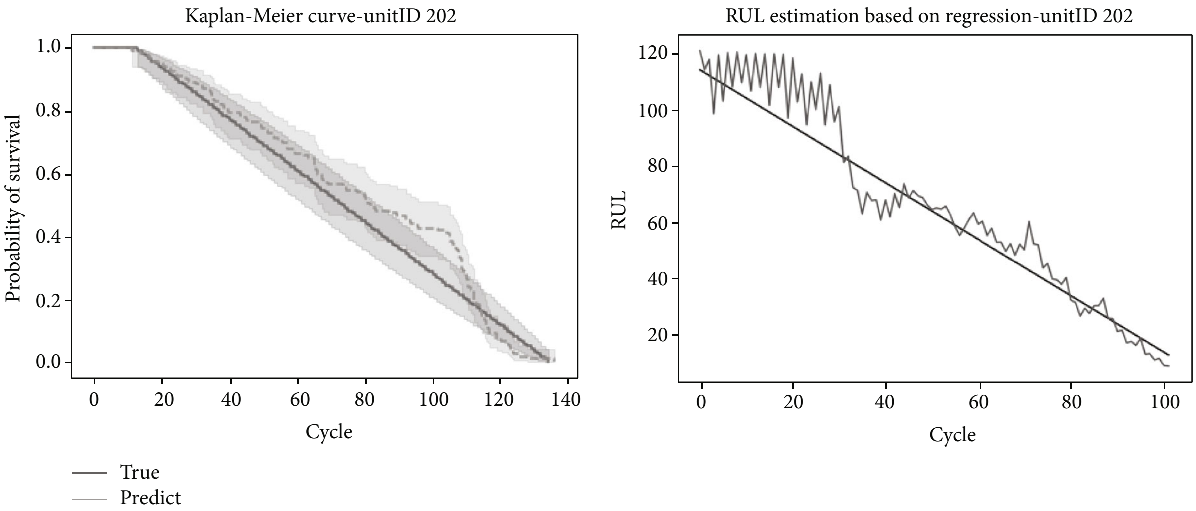
Figure 7: Result of engine ID 14 in FD003 (a) based on the Weibull distribution and (b) based on regression.

predictions that are posing a dilemma for the maintenance decision. Instead, the survival analysis based on the discrete Weibull distribution has a smoother trend with Kaplan-Meier curve in Figures 5–8(a). On the other hand, previous works based on the regression analysis only evaluate the performance at the last time step of each engine for RUL estimation and do not achieve confidence intervals for the corresponding RUL prediction. Our model based on the discrete Weibull distribution represents the probabilities of the failure and the trustworthy confidence intervals for the inherent uncertainties with the degradation process.

4.3. Performance with Different Time Windows. We apply more reference to compare the performance with different TW lengths based on two analysis approaches for each subdataset in Tables 3–6. The italicized numbers denote the best results among different TW under two analyses. As for MAE and RMSE, regression analysis approach usually has better performance than the Weibull distribution. As for R^2, it shows that the discrete Weibull distribution is much more explainable and also denotes that the degradation trends of the turbofan engines are more similar to the Weibull distribution.

5. Conclusions

With the growth of smart manufacturing in the industry, more and more data will be collected and deep learning models will be extremely applied to estimate the health statement of a machine for the maintenance strategy. The predictive maintenance can bring advantages for proposing strategies to optimize the maintenance schedule with a goal of reducing unplanned downtime, as well as needless preventive maintenance to save the most cost for the company. We propose an integrated deep learning approach with convolutional neural networks and long short-term memory networks to learn the latent features and estimate remaining useful life value with deep survival model based on the



(a)

(b)

Figure 8: Result of engine ID 202 in FD004 (a) based on the Weibull distribution and (b) based on regression.

Table 3: Results of FD001 with two analyses and different TW.

	Loss function (regression)				Loss function (Weibull)			
TW	MAE	RMSE	R^2	Score	MAE	RMSE	R^2	Score
20	17.26	23.13	0.65	3,737	20.24	28.64	0.84	12,492
30	15.06	19.06	0.73	1,065	15.08	21.50	0.91	2,324
40	12.81	17.89	0.79	994	14.14	19.91	0.91	2,164
50	11.29	14.90	0.86	431	13.14	19.23	0.91	5,861
60	*10.20*	*14.04*	*0.88*	*310*	13.69	20.36	0.90	31,427
70	10.80	14.33	0.87	254	13.98	20.01	0.90	5,888
80	10.32	14.14	0.86	198	11.63	15.45	0.93	325
90	10.36	14.15	0.87	200	*10.50*	*13.98*	*0.94*	*231*
100	9.99	13.98	0.79	190	11.17	15.02	0.91	254
110	9.91	13.81	0.77	183	11.03	15.43	0.90	389
120	9.44	13.85	0.86	161	9.79	13.73	0.93	206

Table 4: Results of FD002 with two analyses and different TW.

	Loss function (regression)				Loss function (Weibull)			
TW	MAE	RMSE	R^2	Score	MAE	RMSE	R^2	Score
20	23.85	32.58	0.50	44,982	22.88	33.89	0.81	46,493
30	24.09	33.36	0.59	41,297	21.94	30.90	0.84	29,696
40	22.22	31.75	0.63	36,752	20.86	28.92	0.86	21,125
50	18.94	27.62	0.70	12,390	18.24	25.60	0.88	7,232
60	16.04	22.42	0.81	4,696	17.31	24.60	0.89	5,494
70	13.98	20.34	0.84	3,232	16.11	21.87	0.90	3,497
80	13.43	20.03	0.80	5,478	15.19	20.12	0.90	2,293
90	*10.71*	*15.15*	*0.87*	*1,085*	13.57	17.58	0.91	1,349
100	10.37	15.32	0.84	1,476	*11.70*	*15.77*	*0.91*	*979*
110	9.96	15.13	0.84	1,414	13.53	17.97	0.88	1,701
120	9.64	14.27	0.84	790	12.69	16.39	0.88	1,810

Table 5: Results of FD003 with two analyses and different TW.

	Loss function (regression)				Loss function (Weibull)			
TW	MAE	RMSE	R^2	Score	MAE	RMSE	R^2	Score
20	14.41	20.20	0.79	2,437	33.45	49.28	0.52	4,598,400
30	11.79	17.12	0.85	1,700	24.94	37.87	0.65	139,633
40	11.01	16.02	0.83	1091	15.86	24.33	0.88	9,330
50	11.24	15.41	—	679	14.45	21.82	0.90	8,597
60	*10.80*	*14.62*	*0.86*	*325*	14.69	23.57	0.87	32,361
70	12.08	15.94	0.80	401	14.11	22.73	0.88	22,581
80	11.87	16.54	0.82	379	13.69	23.01	0.87	30,087
90	12.25	17.25	0.79	353	14.00	23.42	0.86	39,985
100	11.88	17.53	0.78	366	14.60	25.52	0.83	30,360
110	12.58	19.41	0.76	463	12.21	20.98	0.89	11,536
120	13.45	21.20	—	625	*9.81*	*15.55*	*0.93*	*672*

Table 6: Results of FD004 with two analyses and different TW.

	Loss function (regression)				Loss function (Weibull)			
TW	MAE	RMSE	R^2	Score	MAE	RMSE	R^2	Score
20	27.81	36.39	0.54	58,522	28.69	38.56	0.78	345,533
30	27.44	35.89	0.52	65,319	27.36	36.75	0.76	580,301
40	22.77	30.32	0.64	17,827	23.16	31.59	0.84	41,486
50	21.16	29.17	0.68	13,305	20.87	28.67	0.84	10,035
60	19.41	26.37	0.73	11,064	21.59	29.99	0.84	51,971
70	19.83	27.56	0.73	6,999	18.11	26.05	0.88	11,148
80	18.59	24.55	0.78	4,652	18.85	27.68	0.86	120,754
90	18.41	24.93	0.77	4,832	16.90	24.36	0.89	75,231
100	17.78	23.69	0.77	3,448	*15.71*	*23.05*	*0.89*	*36,726*
110	16.11	21.97	0.81	2,726	16.95	24.98	0.87	96,205
120	*16.29*	*21.92*	*0.82*	*2,260*	16.75	25.09	0.88	122,440

discrete Weibull distribution. Our works can not only estimate RUL but also learn the failure probability. We can provide reference for making a decision about when and how often the replacement should be implemented. In particular, our approach does well on the harder task under the complex conditions with a subtle drop on the error and scoring function compared with other existing state-of-the-art methods. The improved results have proven that our proposed model can capture the degradation trend of a fault under complex conditions and avoid failure with the early prediction. The limitation of our approach is that our model relies on the specific probability distributions corresponding to a mixture of the two-parameter discrete Weibull distributions that may not be suitable for every degradation process. There are still some available distributions that can be implemented in the survival analysis approach. The data-driven deep learning approach depends on the quality of the data and strongly requires large labeled training datasets in the supervised learning. But getting sufficient run-to-failure data for training process is very difficult, especially for new systems. For further improvements, it may be possible to use a generative adversarial network for data augmentation or generation for the future research. Since the data condition and fault mode are different between subdatasets, further optimization via transfer learning method is still necessary to improve the stability of the method and then efficiently apply to solve other problems.

Abbreviations

$X(n)$: Matrix including cycles and sensor measurements of nth engine
k: Types of sensor measurements (features)
L_s: Last observed cycle or the cycle that fault occurs
m: Time window (TW) length
x^t: k-dimensional array at cycle t of an engine
$X_i(n)$: Sequential data extracted from $X(n)$ with size $m \times k$
x_k^t: The value at cycle t and sensor measurement k in x^t
d: Kinds of the sliding windows (feature detector) in CNN
X_{dw}: Set of feature detector regions with w kernel size w in CNN
$C^{(l)}$: Convolutional lth layer
f_l: Forget gate's activation vector of the layer l in LSTM
$\widetilde{Z_l}$: Cell input activation vector of the layer l in LSTM
Z_l: Cell state vector of the layer l in LSTM.

Data Availability

The dataset used in this paper is the Turbofan Engine Degradation Simulation Dataset, provided by NASA, retrieved from https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[2] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: a systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.

[3] R. Zhao, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring: A surve," 2016, https://arxiv.org/abs/1612.07640.

[4] D. Wang, K.-L. Tsui, and Q. Miao, "Prognostics and health management: a review of vibration based bearing and gear health indicators," *IEEE Access*, vol. 6, pp. 665–676, 2018.

[5] P. C. Paris, M. P. Gomez, and W. E. Anderson, "A rational analytic theory of fatigue," *Trend Eng*, vol. 13, pp. 9–14, 1961.

[6] W. K. Yu and T. A. Harris, "A new stress-based fatigue life model for ball bearings," *Tribology Transactions*, vol. 44, no. 1, pp. 11–18, 2001.

[7] J. Luo, K. R. Pattipati, Liu Qiao, and S. Chigusa, "Model-based prognostic techniques applied to a suspension system," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 5, pp. 1156–1168, 2008.

[8] A. Mosallam, K. Medjaher, and N. Zerhouni, "Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction," *Journal of Intelligent Manufacturing*, vol. 27, no. 5, pp. 1037–1048, 2016.

[9] N. Li, Y. Lei, T. Yan, N. Li, and T. Han, "A wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 2092–2101, 2019.

[10] K. Javed, R. Gouriveau, and N. Zerhouni, "A new multivariate approach for prognostics based on extreme learning machine and fuzzy clustering," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2626–2639, 2015.

[11] J. Liu and Z. Chen, "Remaining useful life prediction of lithium-ion batteries based on health Indicator and Gaussian process regression model," *IEEE Access*, vol. 7, pp. 39474–39484, 2019.

[12] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Lithium-ion battery remaining useful life prediction with box–cox transformation and Monte Carlo simulation," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1585–1597, 2019.

[13] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2276–2285, 2017.

[14] S. Hong, E. Z. Zhou, K. Zio, and K. Hong, "Condition assessment for the performance degradation of bearing based on a combinatorial feature extraction method," *Digital Signal Processing*, vol. 27, pp. 159–166, 2014.

[15] Y. Lei, *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*, Butterworth-Heinemann, 2016.

[16] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017.

[17] S. J. Wu, N. Gebraeel, M. A. Lawley, and Y. Yih, “A neural network integrated decision support system for condition-based optimal predictive maintenance policy,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 2, pp. 226–236, 2007.

[18] P. Baruah and R. B. Chinnam, “HMMs for diagnostics and prognostics in machining processes,” *International Journal of Production Research*, vol. 43, no. 6, pp. 1275–1293, 2005.

[19] A. Malhi, R. Yan, and R. X. Gao, “Prognosis of defect propagation based on recurrent neural networks,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 703–711, 2011.

[20] F. O. Heimes, “Recurrent neural networks for remaining useful life estimation,” in *2008 International Conference on Prognostics and Health Management*, Denver, CO, USA, October 2008.

[21] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[22] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, “Long short-term memory network for remaining useful life estimation,” in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Dallas, TX, USA, June 2017.

[23] J. Zhang, P. Wang, R. Yan, and R. X. Gao, “Long short-term memory for machine remaining life prediction,” *Journal of Manufacturing Systems*, vol. 48, pp. 78–86, 2018.

[24] A. Elsheikh, S. Yacout, and M.-S. Ouali, “Bidirectional handshaking LSTM for remaining useful life prediction,” *Neurocomputing*, vol. 323, pp. 148–156, 2019.

[25] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, “Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,” *Neurocomputing*, vol. 275, pp. 167–179, 2018.

[26] J. Wang, G. Wen, S. Yang, and Y. Liu, “Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network,” in *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, pp. 1037–1042, Chongqing, China, October 2018.

[27] T. A. Shifat and H. J. Wook, “Remaining useful life estimation of BLDC motor considering voltage degradation and attention-based neural network,” *IEEE Access*, vol. 8, pp. 168414–168428, 2020.

[28] W. Bao, X. Miao, H. Wang, G. Yang, and H. Zhang, “Remaining useful life assessment of slewing bearing based on spatial-temporal sequence,” *IEEE Access*, vol. 8, pp. 9739–9750, 2020.

[29] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, “Data alignments in machinery remaining useful life prediction using deep adversarial neural networks,” *Knowledge-Based Systems*, vol. 197, article 105843, 2020.

[30] L. Jayasinghe, T. Samarasinghe, C. Yuenv, J. C. N. Low, and S. S. Ge, “Temporal convolutional memory networks for remaining useful life estimation of industrial machinery,” in *2019 IEEE International Conference on Industrial Technology (ICIT)*, pp. 915–920, Melbourne, Australia, February 2019.

[31] G. S. Babu, P. Zhao, and X. L. Li, “Deep convolutional neural network based regression approach for estimation of remaining useful life,” in *International Conference on Database Systems for Advanced Applications*, Springer, 2016.

[32] X. Li, Q. Ding, and J. Q. Sun, “Remaining useful life estimation in prognostics using deep convolution neural networks,” *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[33] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018, https://arxiv.org/abs/1803.01271.

[34] K. Aggarwal, O. Atan, A. K. Farahat, C. Zhang, K. Ristovski, and C. Gupta, “Two birds with one network: unifying failure event prediction and time-to-failure modeling,” in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, December 2018.

[35] J. Li, X. Li, and D. He, “A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction,” *IEEE Access*, vol. 7, pp. 75464–75475, 2019.

[36] E. Martinsson, *Wtte-Rnn: Weibull Time to Event Recurrent Neural Network*, Chalmers University of Technology & University of Gothenburg, 2016.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting, 7e,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[38] E. T. Lee and J. W. Wang, *Statistical Methods for Survival Data Analysis*, Wiley Series in Probability and Statistics Copyright © 2003 John Wiley & Sons, Inc., Third edition, 2003.

[39] P. Wang, Y. Li, and C. K. Reddy, “Machine learning for survival analysis: a survey,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[40] A. Saxena and K. Goebel, *Turbofan Engine Degradation Simulation Data Set*, NASA Ames Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA, USA, 2008.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[42] C. C. Chang and C. J. Lin, “LIBSVM: a library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[43] M. E. Tipping, “The relevance vector machine,” in *NIPS'99: Proceedings of the 12th International Conference on Neural Information Processing Systems*, pp. 652–658, Denver, CO, USA, November 1999.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8882482, 10 pages
https://doi.org/10.1155/2020/8882482

Research Article

Applying Robust Intelligent Algorithm and Internet of Things to Global Maximum Power Point Tracking of Solar Photovoltaic Systems

En-Chih Chang

Department of Electrical Engineering, I-Shou University, No. 1, Sec. 1, Syuecheng Rd., Dashu District Kaohsiung City 84001, Taiwan

Correspondence should be addressed to En-Chih Chang; enchihchang@isu.edu.tw

Received 4 August 2020; Revised 29 October 2020; Accepted 11 November 2020; Published 25 November 2020

Academic Editor: Chao-Yang Lee

The important dare in the solar photovoltaic (PV) system is to investigate the performance under partial shading conditions. A robust intelligent algorithm (RIA) connected with internet of things (IoT) is developed to offer the real-time monitoring of solar PV systems, thus ensuring global maximum power point tracking (MPPT). The RIA comprises a limited-time terminal sliding-mode control (LTTSMC) and a quantum particle swarm optimization- (QPSO-) radial basis function (RBF) neural network. The LTTSMC creates a quick limited-system-state convergence time and allows for singularity avoidance. However, if the system ambiguity is overrated or underrated, the tremble phenomenon or steady-state error probably occurs around the LTTSMC. The QPSO-RBF neural network is integrated into LTTSMC to handle plant parameter variations and external load perturbations, thus reducing tremble and steady-state errors. With the aggregation of the RIA and the IoT, the remote monitoring in the solar PV system yields faster convergence to nonsingular points, and it also introduces neural network method to achieve more accurate ambiguity estimation. Experimental results show the mathematical analysis and performance enhancement of a prototype algorithm-controlled solar PV system based on digital signal processing under transient and steady-state loading conditions. Because the proposed solar PV system has notable advantages over the classical terminal-sliding solar PV system in terms of tracking accuracy and robust adaptation, this paper is worthy of reference to designers of relative robust control and neural network learning algorithm.

1. Introduction

The robust intelligent algorithm for the internet of things supports the remote monitoring of real-time solar PV data, such as panel voltage, current, power, temperature, and solar radiation. In order to achieve the maximum power point tracking of a photovoltaic system, the output side of the solar cell needs to be regulated with a switching power converter with maximum power point tracking control, which allows the solar panel to output the maximum power and achieve fast and accurate tracking [1–6]. Various maximum power point tracking (MPPT) methods have been proposed in the literature, such as perturbation observation, incremental conductance, fuzzy control, and linear approximation [7–10]. However, the solar illumination and ambient temperature are closely relative to the maximum power output of the solar panel. Most of these MPPT algorithms cannot offer a strict convergence and stability analysis or have the problem with fast tracking to the maximum power point, resulting in reduced output power. Sliding-mode control (SMC) is simple and easy to design and has the robustness to parametric variations and external interference during the sliding process [11–17]; many relative SMCs have been published for the control of solar PV systems [18–20]. However, in practice, the stability, convergence, and performance of solar PV systems controlled by the law of the SMC are compromised when the system state convergence time is not finite, and the tremble and steady-state error occurs under the influence of ambiguous interferences. Recently, the limited-time terminal sliding-mode control (LTTSMC) not only has a robust design method, clear analysis of convergence, and stability, but also provides a limited-system-state convergence time and improves the dynamic quality of the PV system [21–23]. However, PV arrays are susceptible to partial shading caused by construction, trees, dust, etc. (greatly reduce the output power of the system and result in large energy losses), and therefore the output characteristics of PV arrays may change irregularly, resulting in multiple area extremes of output power. At this moment, using the previously described methodologies of maximum power tracking (such as disturbance observation method, the incremental conductance method, the fuzzy control method, and linear approximation method) would be trapped in the dilemma of tracking to local maximum power points instead of global maximum power points. Some other global optimization algorithms are proposed, such as genetic algorithm (GA), differential evolution (DE), simulated annealing (SA), and ant colony optimization (ACO) [24–27]. But they either converge prematurely to local optimization or search time is too long. Recently, based on the advanced development of artificial intelligence (AI) [28–34], the radial basis function (RBF) neural network can approximate complicated nonlinear functions by using Gaussian kernels and possess fast learning speed, leading to better transient and steady state for MPPT in PV systems. Also, particle swarm optimization (PSO) is a simple and effective method that has been applied to the actual MPPT systems [35–37]. Its improved version QPSO (quantum particle swarm optimization) can address the drawback of regular PSO which converges easily to local solutions, guaranteeing convergence to global optimal solutions [38–41]. Therefore, inspired by good algorithm design concepts and practical applications [42–45], it will be a good idea to combine the QPSO with RBF neural network. Such hybrid learning mechanism uses optimal methodology and neural network into a systematical structure to govern ambiguity, randomness, and time-varying vagueness, endowing with more accurate constraint estimates. For this reason, the LTTSMC with QPSO-RBF neural network can attenuate the tremble when the ambiguity value is overrated or reduce the steady-state error when the ambiguity value is underrated. In other words, by using the QPSO-RBF neural network and system state information, the switching gain of the LTTSMC can be adaptively adjusted to moderate the overconservation of the LTTSMC design, thereby lessening the tremble and offering global maximum power output. When subjected to partial shading and under high-ambiguity conditions, experimental results from digital implementation show that the proposed algorithm will enable the solar maximum power tracking system to enhance the performance during steady state and tracking speed during transient conditions.

2. Description of Solar PV System

In order to increase the efficiency of solar PV systems, it is necessary to have a maximum power tracking function. In particular, it should be noted that when the PV array encounters a partially occluded phenomenon, the power-voltage curve shows multiple peaks (Figure 1), and traditional maximum power tracking methods (e.g., perturbation observation, incremental conductance, and hill climbing) will fall into a local maximum power point instead of a global maximum power point. In addition, there is a voltage reference value corresponding to the maximum power point of a solar PV system. Therefore, a Zeta DC-DC converter (Figure 2) is used to adjust the solar cell voltage (for the maximum power point voltage). This is because the output voltage of Zeta DC-DC converter can be higher/lower than the input voltage and have the same polarity with continuous output current. Due to the small-ripple inductor current, it can reduce the tremble during steady-state sliding mode.

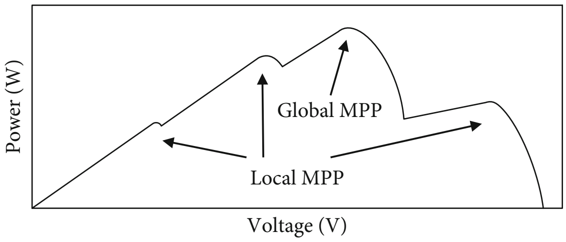


Figure 1: Power-voltage curve for partial shading.

For solar PV systems, we propose a QPSO-RBF neural network which calculates a voltage reference at the maximum power point in a solar array under partial shading conditions, and a limited-time terminal sliding-mode control achieves the v_{PV} to reach the voltage reference. The system state variables of the solar PV system are derived from the following matrix. The solar Zeta DC-DC converter circuit shown in Figure 2 uses the switch on and off states to express the matrix of the solar PV system as follows:

$$\dot{x}(t) = f(x(t)) + g(x(t))u, \tag{1}$$

where $\dot{x} = [\dot{v}_{\mathrm{PV}}\ \dot{i}_{L1}\ \dot{i}_{L2}\ \dot{v}_{c1}\ \dot{v}_{c2}]^T$, u denotes control input with duty cycle signal, $f = [\,i_{\mathrm{PV}}/C_i \quad -v_{c1}/L_1 \quad -v_{c2}/L_2 \quad i_{L1}/C_1 \quad (i_{L2} - i_o)/C_o]^T$, and $g = [-(i_{L1} + i_{L2})/C_i \quad v_{\mathrm{PV}} + v_{c1} \quad (v_{\mathrm{PV}} + v_{c1})/L_2 \quad -(i_{L2} + i_{L1})/C_1 \quad 0]^T$. It should also be noted that the parameters of the Zeta DC-DC converter need to be appropriately designed as suggested below [46–49]. In actual coupled inductor, the inductors have unequal inductance, and the ripple currents are not completely the same. For the required ripple current value, the inductance of the coupled inductor can be estimated to be half of the required inductance when two independent inductors (L_1 and L_2) exist; this can be referred to (4) and (19) in [46, 47], respectively. The coupling capacitor C_1 shown as (6) in [48] can be designed based on the ripple voltage. Also, (19) in [49] shows that the output capacitor C_o is recommended to have sufficient capacitance to keep the DC-link voltage and to provide continuous load current under high switching frequency.

The control objective in a PV system is to make v_{PV} track the reference voltage v_r. The error $e_1 = v_{\mathrm{PV}} - v_r$ can be defined as the difference of the v_{PV} and the reference voltage, at which the control rule is designed, and the purpose is to design the control rule well. Then, to convert the DC voltage generated from Zeta DC-DC converter to AC voltage, Figure 3 shows the circuit diagram of a single-phase full-bridge voltage source inverter, followed by LC filter. The full-bridge DC-AC inverter is the core of the system, which

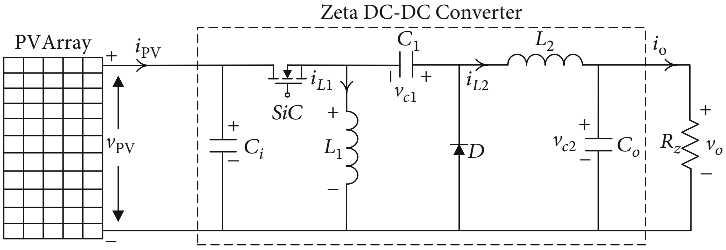


Figure 2: Solar PV powered Zeta DC-DC converter.

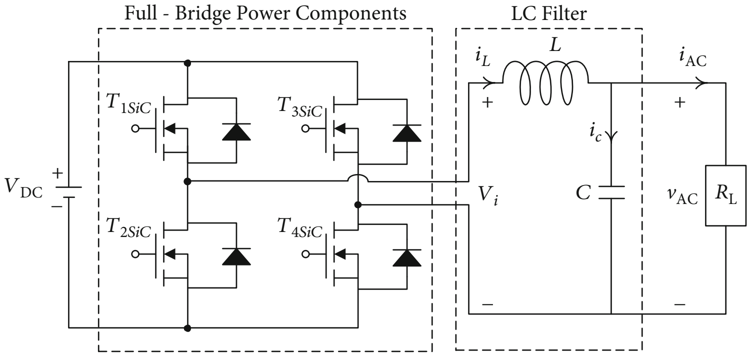


Figure 3: Block diagram of single-phase DC-AC inverter.

converts the DC input chopper into a series of pulse-width modulation according to the modulation signal. The function of second-order LC filter is to eliminate the high-frequency component of chopper output voltage v_i. We also need to pay attention to the selection and determination of the LC filter component values of the full-bridge DC-AC inverter, which can refer to the recommendations in the following literature [50–52]. (1) Selection of switching frequency [51, 52]: usually in order to reduce the size of the LC filter, the switching frequency of the insulated gate bipolar transistor switches can be picked from 3 kHz to 15 kHz, and the silicon carbide MOSFET switches can be chosen higher than 30 kHz switching frequency. (2) Selection of a factor value relative to LC low-pass filter cutoff frequency [50]: when there is a greater factor value, the switching and fundamental frequencies yield more descent and less amplification, respectively. Once the suggested modulation value is lower than 0.95, the least value of this factor can be obtained. (3) Selection of a factor value relative to the switching frequency and the inductor ripple current [52]: the inductor ripple current limit value of 20% to 40% can be recommended. The factor is chosen by the equations (8), (20), (25), and (26) of [50], thereby acquiring the L and C values. Applying KVL and KCL, the state space equation for a single-phase inverter can be expressed as $\ddot{v}_{AC} = -v_{AC}/LC - \dot{v}_{AC}/R_L C + k_{pwm} \cdot u_{inv}/LC$, where k_{pwm} stands for the equivalent gain of the inverter. Therefore, the error state equation of the inverter is obtained as $\dot{x}_{e2} = -x_{e1}/LC - x_{e2}/R_L C + k_{pwm} \cdot u_{inv}/LC - v_{AC,r}/LC - \dot{v}_{AC,r}/R_L C - \ddot{v}_{AC,r}$, where $x_{e1} = v_{\mathrm{AC}} - v_{\mathrm{AC},r}$, $\dot{x}_{e1} = x_{e2} = \dot{v}_{\mathrm{AC}} - \dot{v}_{\mathrm{AC},r}$, and $v_{\mathrm{AC},r}$ is a demanded sinusoidal reference. Figure 4 displays the structure of the whole control system, and in order to allow the error states converged to zero, the control law u_{inv} is designed via fractional proportional–integral (FPI) method expressed as $x_{e1} \cdot (k_p + k_i(d^{-\alpha}/dt^{-\alpha}))$, where k_p is proportional gain, k_i denotes integral gain, and α symbols the noninteger order of the integral term. It must be mentioned that the FOMCON toolbox of the Matlab/simulink package software offers FPI control applications, such as the identification of dynamic models in the time domain and frequency domain, and controller design. This paper employs the FOMCON toolbox to get the gain values of k_p and k_i, and then these gain values are adopted in digital implementation, thereby achieving good inverter response [53–56].

3. Control Design

The limited-time sliding function can be written as

$$s = e_1 + \frac{1}{\lambda} e_2^{q/p}, \tag{2}$$

where $e_1 = v_{\mathrm{PV}} - v_r$, $e_2 = \dot{e}_1$, $\lambda > 0$, and p and q intend by positive odd numbers ($p < q < 2p$). Afterward a sliding-mode reaching rule $\dot{s} = -\eta_1 |s|^{1-\gamma_1} \mathrm{sgm}(s) - \eta_2 |s|^{\gamma_2} \mathrm{sgm}(s)$ is recommended to be employed. Note that a sigmoid function $\mathrm{sgm}(s) = s/\tau/\sqrt{1 + (s/\tau)^2}$; here, τ is a small positive constant. The control law of the LTTSMC can be expressed as

$$u = u_{\mathrm{eq}} + u_s. \tag{3}$$

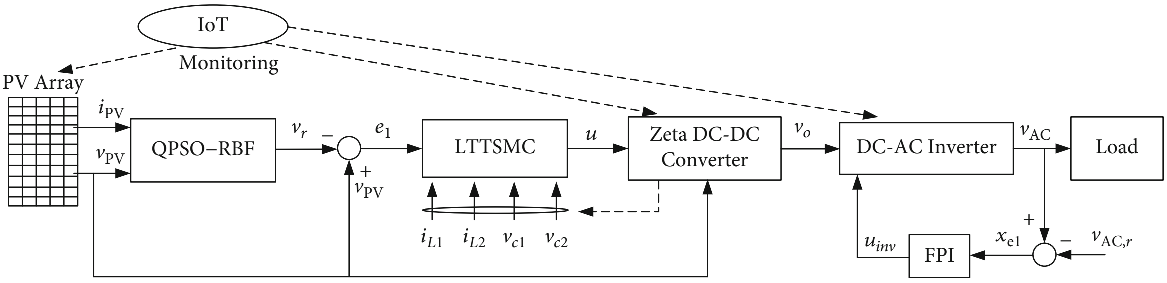


Figure 4: Block diagram of total control system.

Since u_{eq} is the equivalent control without interferences, it can be written succinctly as

$$u_{eq} = -g^{-1}\left(k_{eq}e + \lambda\frac{p}{q}e_2^{2-q/p}\right), \tag{4}$$

where e is the error state vector and k_{eq} stands for equivalent feedback gain. The index of the e_2 in the equivalent control u_{eq} is greater than zero, leading to nonsingularity.

Also,

$$u_s = -g^{-1}\left[\eta_1|s|^{1-\gamma_1}\mathrm{sgm}(s) + \eta_2|s|^{\gamma_2}\mathrm{sgm}(s)\right],\ \eta_1, \eta_2 > 0, 0 < \gamma_1 < 1, \gamma_2 > 1, \tag{5}$$

where the sliding control u_s can compensate for the repercussions of agitations. As a result, the system state will be forced to arrive $s = 0$ and converge in a limited amount of time.

Proof. Specify a Lyapunov candidate as

$$V = \frac{1}{2}s^2. \tag{6}$$

According to the dynamic system trajectory and the control law (3) and utilizing the above Lyapunov candidate, the time derivative V is given as

$$\dot{V} = s\dot{s} = s\cdot\left(\dot{e}_1 + \frac{1}{\lambda}\frac{q}{p}e_2^{q/p-1}\dot{e}_2\right) \le -s\cdot\left(\frac{1}{\lambda}\frac{q}{p}e_2^{q/p-1}\left(\eta_1|s|^{1-\gamma_1} + \eta_2|s|^{\gamma_2}\right)\mathrm{sgm}(s)\right). \tag{7}$$

On account of $e_2^{q/p-1} > 0$, the differential of V is less than or equal to zero. This deduced that the s and $\dot{s}$ of the LTTSMC in (7) are allowed to converge to the equilibrium for a limited time. In the case of $s = e_1 + \lambda^{-1}e_2^{q/p}$ and $\dot{s} = -\eta_1|s|^{1-\gamma_1}\mathrm{sgm}(s) - \eta_2|s|^{\gamma_2}\mathrm{sgm}(s)$, the system state (1) must also quickly converge to the equilibrium within a limited time. However, the tremble or steady-state error emerges from the LTTSMC. The reason for this situation is that the system load has drastic changes or extreme nonlinearity, so that the final system output cannot follow the reference sine waveform, resulting in inaccurate tracking performance. The control signal u derived from (3) is altered by the addendum of the QPSO-RBF neural network method, which depresses the tremble/steady-state error in the solar PV system. The RBF neural network is depicted as Figure 5. It is easy to grasp that there are three layers in the RBF neural network constitution, namely, the input layer, the hidden layer, and the output layer. The input layer only carries out the delivery of the information, the hidden layer possesses a perceptible region containing a set of RBFs, and the output layer neurons represent linear amalgamations of neuron activations in the hidden layer. More concretely, there are nonlinear interconnections amid the input layer and the hidden layer, but there are linear relationships amid the hidden and the output layers. Using these relationships, a mapping of low-dimensional vectors is performed, and the RBF constitution transforms the low-dimensional linearly inseparable space into a high-dimensional linearly divisible space, thereby providing an approximation ability of the uncertain nonlinear functions. In general, it is not effortless to precisely calculate the upper limit of a system subject to parameter uncertainty and external intermission, but the RBF constitution is a useful tool for solving highly complex and nonlinear problems. Thereby, this RBF neural network is used to estimate the upper limit of system uncertainty and external intermission, effectively depressing the effect of the tremble. With the application of solar PV systems, since the neurons use mathematical operations in the hidden layer and output layers, the tracking error vector e is given as input data in the input layer, and thereafter the processed input is passed through the Gaussian RBF, which maps the processed input to the output layer to obtain the output function. Note that the most effective way to discover a particular data set is to use systematic experimentation. It is often not easy to analytically calculate the number of layers or nodes used in each layer of an artificial neural network to solve the specific problem of actual prediction modeling. This is due to the fact that the number of layers and nodes in each layer must be specified

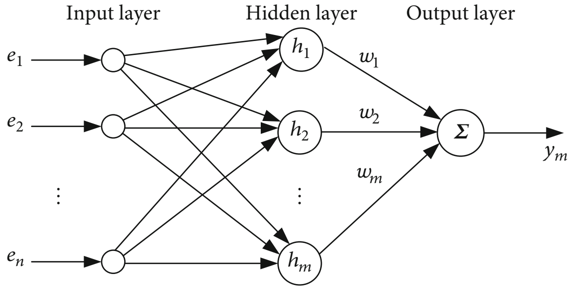


Figure 5: Constitution of RBF neural network.

as model hyperparameters. The designer may be the first person to try to solve the own particular problem using neural networks. As a result, we must use powerful testing tools and controlled experiments to discover the answer. The R programming language provides a package called caret that helps determine the numbers of layers and nodes used in RBF neural network to achieve the minimum mean square error (MSE), i.e., the best combination of parameters [57–59]. Thereby, this constitution is used to adjust the LTTSMC gain to depress tremble and improve the performance of the system, as described and demonstrated below.

We first define the upper limit of the plant parameter changes and the external load intermissions as $\bar{\Theta} > |\Theta|$, and the estimated value is symbolized by $\tilde{\bar{\Theta}} = W^T h(e_i)$; here, $W = [w_1, w_2, \cdots, w_m]^T$ stands for network weights, and h denotes the Gaussian function as follows:

$$h_j = \exp\left(-\|e - c_j\|^2 / 2\rho_j^2\right), j = 1, 2, \cdots, m, \tag{8}$$

where c_j represents the center of the hidden layer neuron, and the smoothing parameter ρ_j means the width of Gaussian function. Then, make the following suppositions throughout the proof.

Supposition I [60, 61]: if there is an optimal weight matrix W^* of the RBF constitution and the (9) is hold, then the output of the optimal neural network can be obtained.

$$\delta(e_i) = W^{*T} h(e_i) - \bar{\Theta} < \delta_n. \tag{9}$$

Supposition II [60, 61]: presuming the uncertainty upper limit of system (1) satisfies the following form:

$$\bar{\Theta} - |\Theta| > \delta_0 > \delta_n > 0. \tag{10}$$

Then, the sliding control u_s can be redescribed as

$$u_s = -\tilde{\bar{\Theta}} \operatorname{sgn}(s). \tag{11}$$

The stability of the system (1) can be guaranteed by Lyapunov function below.

$$V = \frac{1}{2} \cdot \left(s^2 + \kappa^{-1}(W^* - W)^T (W^* - W)\right), \tag{12}$$

where $\kappa > 0$.

Using the above-mentioned equivalent control and (11), the derivative of V yields

$$\begin{aligned} \dot{V} &= s\dot{s} - \kappa^{-1}(W^* - W)^T \dot{W} \\ &= -|s|\tilde{\bar{\Theta}} + s\Theta - \kappa^{-1}\left(W^{*T} - W^T\right)\dot{W}. \end{aligned} \tag{13}$$

In addition, the weight update method is designed to

$$\dot{W} = \kappa |s| h(e_i). \tag{14}$$

Table 1: Solar cell electrical parameters.

(Irradiances 1 kW/m^2, module temperature 25°C) Rated power	75 W
Rated working voltage	17 V
Rated working current	4.4 A
Open-circuit voltage	21.7 V
Short-circuit current	4.8 A
Ideality factor of the diode	1.2
Temperature coefficient of short-circuit current	4 mA/°C
P-N junction parameter	2.035

From (9), (10), and (14), the (13) becomes

$$\dot{V} = -|s| W^{*T} h(e_i) + s\Theta \leq -|s|\left(\bar{\Theta} + \delta(e_i) - |\Theta|\right) \leq -|s|(\delta_0 - \delta_n) \quad \leq 0, \tag{15}$$

where $\kappa = \delta_0 - \delta_n$. Although the (15) is kept, it is necessary to give further basic explanations about the presence of compact sets in the control of neural networks so that the approximation capability of the neural networks can be constructed. In order to limit the state of the closed-loop system, the reports of the necessary assumptions, the use of a backstepping methodology, and the suggestion for an adaptive law have been provided and proved from previous work, leading to a semiglobally consistent ultimate bounded system stabilization that always keeps the system state in a compact set (see [62] and references therein). Eventually, the solar PV system with RBF neural network and LTTSMC becomes progressively stable and then achieves finite-time convergence with zero tracking error. However, the traditional RBF neural network adopts the local information based on the parameter space to set the relative parameters, which results in the values of the parameters c_j, ρ_j, and w_j becoming locally optimal solutions. To obtain the global optimal solution, the QPSO algorithm can be used. When the number of overlaps approaches infinity, there exists a traditional PSO whose algorithm cannot converge to the global optimum with probability 1, i.e., it is not globally convergent. In addition, the velocity of a single particle has an upper limit on its search space, which is finite and cannot cover the entire feasible solution space, thus limiting the algorithm's ability to search the entire domain. Instead of requiring velocity vectors, the QPSO is a completely stochastic stacking equation that has no deterministic trajectory as the particle moves through the quantum space, and the entire stacking equation has fewer easily controllable parameters, allowing the particle to search for the global optimal value over the entire feasible solution space. The superposition representation of the QPSO is expressed as

$$\zeta_i^k = \psi^k \zeta_{p\text{best},i}^k + \left(1 - \psi^k\right) \zeta_{g\text{best},i}^k, \tag{16}$$

$$X_i^{k+1} = \zeta_i^k \pm \beta^k \left|Q^k - X_i^k\right| \ln\left(\frac{1}{\Psi^k}\right), \tag{17}$$

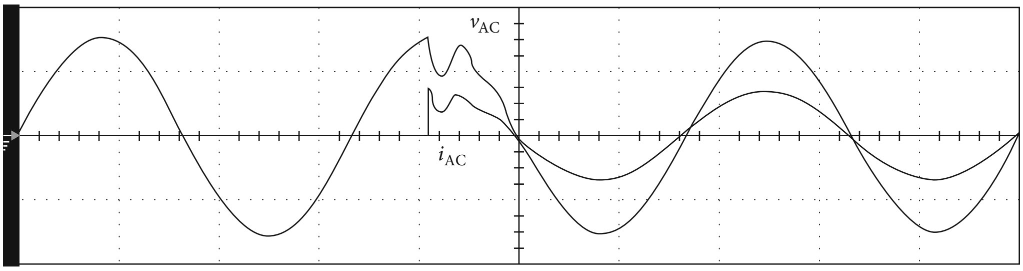


Figure 6: Experimental output waveforms of a solar PV system controlled by the classical terminal sliding-mode control under step load (vertical: 100 V/div and 5 A/div; horizontal: 5 ms/div).

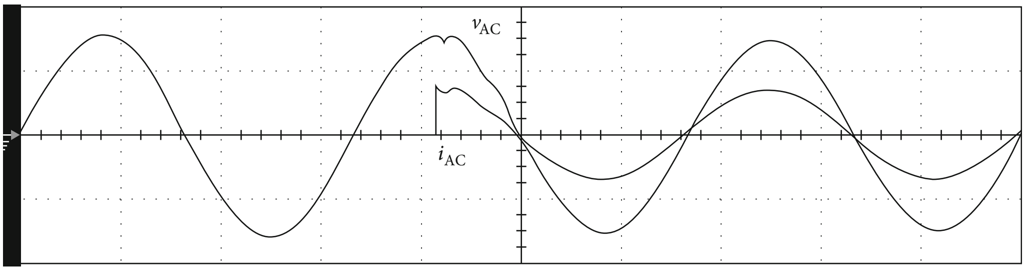


Figure 7: Experimental output waveforms of a solar PV system controlled by the proposed algorithm under step load (vertical: 100 V/div and 5 A/div; horizontal: 5 ms/div).

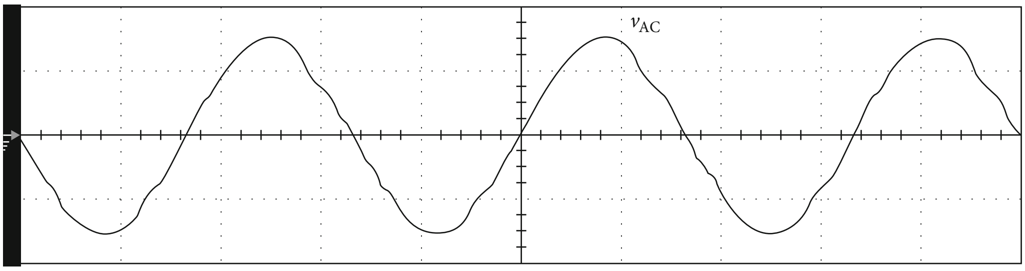


Figure 8: Experimental output waveforms of a solar PV system controlled by the classical terminal sliding-mode control under the variations of filter parameters (vertical: 100 V/div; horizontal: 5 ms/div).

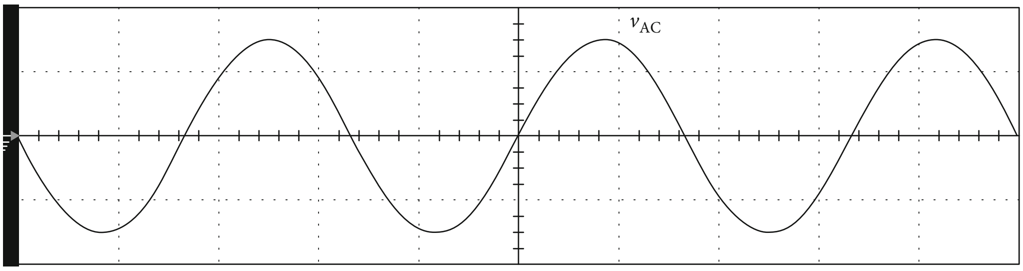


Figure 9: Experimental output waveforms of a solar PV system controlled by the proposed algorithm under the variations of filter parameters (vertical: 100 V/div; horizontal: 5 ms/div).

where $\beta^k = \beta_{\text{base}} + \mathfrak{R}$ indicates the compression/expansion factor; here, β_{base} implies the basic compression/expansion factor, and $\mathfrak{R}$ represents a random number evenly distributed over (0,1); ψ^k and Ψ^k represent random numbers including uniform distribution between 0 and 1, and Q^k denotes the average of the best locations in the region of the cluster for all particles in the search history.

4. Results and Discussion

In order to verify the effectiveness of the proposed controller, the solar module with 4 series and 2 parallel is used, and Table 1 gives relevant parameters. Based on the design of Zeta converter and considering the actual PV environment, the parameters are provided as follows: the inductance L_1 is 0.46 mH, the inductance L_2 denotes 0.5 mH, the capacitance

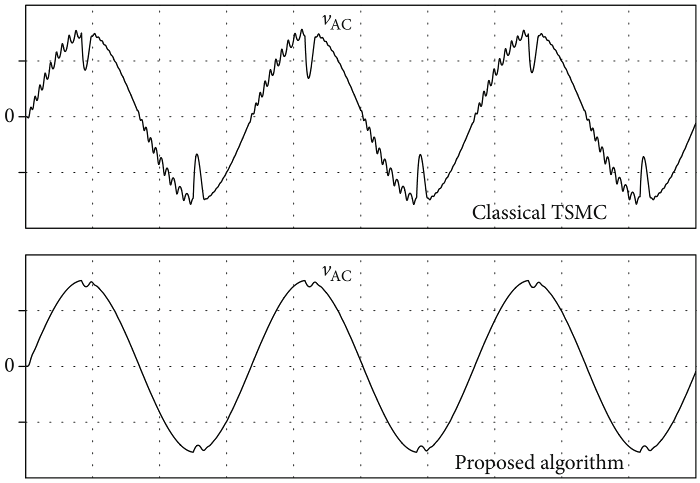


Figure 10: Simulated performance of a solar PV system controlled by the classical TSMC and the proposed algorithm under partial shading condition with five local peaks (vertical: 100 V/div; horizontal: 5 ms/div).

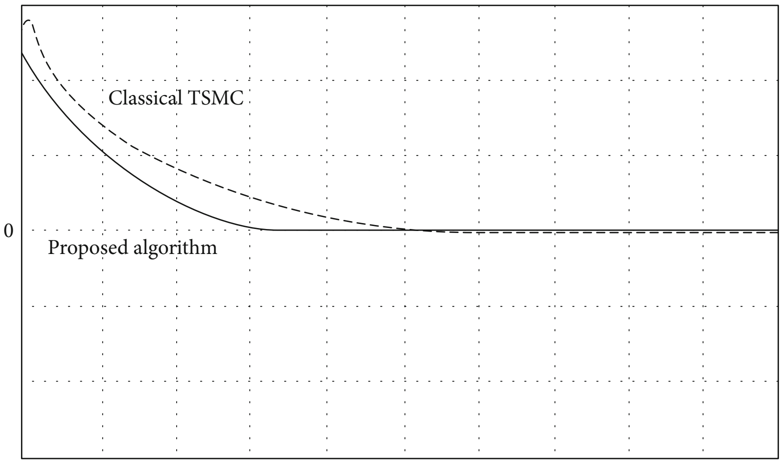


Figure 11: Tracking error comparison between the classical TSMC and the proposed algorithm (vertical: 0.15 V/div; horizontal: 0.75 ms/div).

C_1 indicates 100 μF, the capacitance C_i is 120 μF, and the capacitance C_o means 1000 μF. Also, the inductance L of the DC-AC inverter stands for 0.4 mH, the capacitance C is 10 μF, the DC-link voltage V_{DC} signifies 200 V, the AC output is 110 Vrms, $f = 60$ Hz, and the switching frequency represents 30 kHz.

Figure 6 shows the experimental output waveform of a classical PV system under partial shielding at a 90 trigger angle, changing from no load to full load. It can be seen from the figure that the transient voltage drop does not regain quickly, and the ability of the controller to compensate for the transient voltage is apparently poor. Figure 7 reports the experimental output waveform when the PV system encounters partial shielding using the proposed algorithm at 90 trigger angle, changing from no-load to full-load condition; we can observe that the transient voltage drop regains within a short period of time, and the compensation ability is better than the classical terminal sliding-mode control. Figure 8 illustrates the experimental output voltage of the PV system under partial shielding condition with classical terminal sliding-mode control when the filter parameters of the PV system are changed. The proposed output voltage of a PV system under partial shielding has depicted in Figure 9 shows when the filter parameters of the PV system are changed. Obviously, the output voltage waveform of the PV system controlled by the classical terminal sliding-mode control has more distortion, but the proposed output voltage of the PV system is very close to the sinusoidal reference voltage waveform. Considering the TRIAC-controlled load situation, the simulated output voltage waveform of a solar PV system controlled by the proposed algorithm is displayed in Figure 10 under partial shading condition with five local peaks. The proposed algorithm implies the strong performance with a slight voltage sag and rapid retrieval time. Figure 11 depicts the tracking error comparison between the classical TSMC and the proposed algorithm. The tracking accuracy and speed of the proposed algorithm are significantly better than classical TSMC. Table 2 shows the output voltage drop in step loads, and Table 3 displays the variation of the output voltage THD on the filter parameters. It is

Table 2: Output voltage drop under step load.

	Classical TSMC	Proposed algorithm
Step loads	No load to full load	No load to full load
Voltage drop	60.2 Vmax	10.2 Vmax
Improvement degree	Nearly eighty-four percent improvement in voltage drop reduction	

Table 3: Output voltage %THD under the variations of filter parameters.

	Classical TSMC	Proposed algorithm
Filter parameters	10%~200%	10%~200%
%THD	20.6%	0.015%
Improvement degree	Complete improvement the impact when faced with filter parameter changes	

important to point out that we can compare and analyze four kinds of lately used optimization algorithms, such as adaptive differential evolution (ADE), enhanced simulated annealing (ESA), improved genetic algorithm (IGA), and discrete particle swarm optimization (DPSO) [63–66]. The generation mechanism of ADE new solutions and the population search capability are similar to the DPSO. Because the best solution in the population has no impact on other solutions, there is better diversity but the mutation vector usually comes from the nonoriginal of solutions. The DPSO has fewer iterations, and its performance is much better than ESA. A lot of iterations are demanded by SA, so as to get the DPSO identical results; nevertheless, each iteration time of DPSO wastes more time than ESA. With the probability, both parents and individuals can be selected. The crossover operation can produce seed, which extracts from the parent, and its solution is similar to the parent. The IGA tends to generate solutions that can be clustered around certain fine solutions in the aggregate. By introducing different mutation operations, the diversity of IGA can be introduced into the solution, but the increment in the solution time appears nonlinearity while the population increases.

5. Conclusions

The proposed algorithm uses a QPSO-RBF neural network to find the global maximum power point of the photovoltaic array in the event of partial shielding and a limited-time terminal sliding-mode control to provide tracking control with the unique advantage of finite-time system state convergence, depressed tremble, and steady-state error under uncertain intermission conditions. The algorithm developed with the help of the IoT is used in the solar PV system to establish the maximum power output of solar panels and to maintain the highest PV energy conversion efficiency during partially shaded conditions. When comparing the output voltages of the classical terminal sliding-mode control under varying filter parameters, the total harmonic distortion (THD) rate of the proposed control is very low, and the voltage waveform is close to the required sinusoidal reference voltage. Thereby, the proposed experiments in the solar system indeed achieve good performance under partial shading conditions. More importantly, the IoT depresses the tedious task of visiting working sites with frequently recording of performance data, therefore enhancing the control of remote areas for effective and speedy troubleshooting and maintenance.

Data Availability

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

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this article.

Acknowledgments

The author gratefully acknowledges the financial support of the Ministry of Science and Technology, Taiwan, R.O.C., under project numbers MOST 109-3116-F-006-020-CC1 and MOST 107-2221-E-214-006. The author also gratefully acknowledges the financial support of the 2020 ISU Research Project, under contract number ISU-109-01-01A.

References

[1] P. Mohanty, T. Muneer, K. Tariq, and L. Mohan, *Solar Photovoltaic System Applications*, Springer, New York, 2015.

[2] B. M. Wilamowski and J. D. Irwin, *Power Electronics and Motor Drives*, CRC Press, Boca Raton, FL, USA, 2011.

[3] M. S. Mahmoud, *Microgrid: Advanced Control Methods and Renewable Energy System Integration*, Elsevier Science Ltd., Amsterdam, The Netherlands, 2016.

[4] F. Blaabjerg, *Control of Power Electronic Converters and Systems*, Academic Press, Cambridge, MA, USA, 2018.

[5] R. Yuriy, R. Sergey, C. Evgeny, and V. Pavel, *Power Electronics Basics: Operating Principles, Design, Formulas, and Applications*, CRC Press, Boca Raton, FL, USA, 2016.

[6] L. D. Branko and B. Branko, *Power Electronics: Converters and Regulators*, Springer, New York, 2015.

[7] M. Gangavarapu, “Perturb and observe MPPT algorithm implementation for PV applications,” *International Journal of Computer Science and Information Technologies*, vol. 6, no. 2, pp. 1884–1887, 2015.

[8] R. I. Putri, S. Wibowo, and M. Rifa’i, “Maximum power point tracking for photovoltaic using incremental conductance method,” *Energy Procedia*, vol. 68, pp. 22–30, 2015.

[9] B. Bendib, F. Krim, H. Belmili, M. F. Almi, and S. Boulouma, “Advanced fuzzy MPPT controller for a stand-alone PV system,” *Energy Procedia*, vol. 50, pp. 383–392, 2014.

[10] Y. Y. Yang, W. D. Yi, and K. W. Jwo, “High efficiency MPPT using piecewise linear approximation and temperature compensation,” *Advanced Materials Research*, vol. 772, pp. 658–663, 2013.

[11] G. Bartolini, L. Fridman, A. Pisano, and E. Usai, *Modern Sliding Mode Control Theory*, Springer-Verlag, Berlin, 2008.

[12] A. T. Azar and Q. M. Zhu, *Advances and Applications in Sliding Mode Control Systems*, Springer, New York, 2015.

[13] H. Sira-Ramirez, “Sliding regimes in general non-linear systems: a relative degree approach,” *International Journal of Control*, vol. 50, no. 4, pp. 1487–1506, 1989.

[14] Z. Doulgeri, “Sliding regime of a nonlinear robust controller for robot manipulators,” *IEE Proceedings-Control Theory and Applications*, vol. 146, no. 6, pp. 493–498, 1999.

[15] S. C. Tan, Y. M. Lai, and C. K. Tse, *Sliding Mode Control of Switching Power Converters: Techniques and Implementation*, CRC Press, Boca Raton, FL, USA, 2012.

[16] V. Utkin, “Variable structure systems with sliding modes,” *IEEE Transactions on Automatic Control*, vol. 22, no. 2, pp. 212–222, 1977.

[17] H. B. Zhou, J. H. Song, and S. M. Song, “Sliding Mode Guidance Law Considering Missile Dynamic Characteristics and Impact Angle Constraints,” *International Journal of Automation and Computing*, vol. 15, no. 2, pp. 218–227, 2018.

[18] L. G. Wu, P. Shi, and X. J. Su, *Sliding Mode Control of Uncertain Parameter-Switching Hybrid Systems*, Wiley, New York, 2014.

[19] Z. Wang, Y. Mao, Z. Hu, and Y. Xie, “A sliding mode control design based on the reaching law for matrix rectifiers,” *Journal of Power Electronics*, vol. 16, no. 3, pp. 1122–1130, 2016.

[20] S. Vaidyanathan and C. H. Lien, *Applications of Sliding Mode Control in Science and Engineering*, Springer, New York, 2017.

[21] Y. X. Zhao, T. Wu, and Y. Ma, “A double power reaching law of sliding mode control based on neural network,” *Mathematical Problems in Engineering*, vol. 2013, 9 pages, 2013.

[22] H. Wang, X. Zhao, and Y. Tian, “Trajectory tracking control of XY table using sliding mode adaptive control based on fast double power reaching law,” *Asian Journal of Control*, vol. 18, no. 6, pp. 2263–2271, 2016.

[23] C. Y. Dong, H. J. Wang, and W. Y. Cui, “Application of the sliding mode control approach based on double power exponential reaching law for the hydraulic servo system,” *Applied Mechanics and Materials*, vol. 741, pp. 655–658, 2015.

[24] Z. Wang, M. Chinthavali, S. L. Campbell, T. Wu, and B. Ozpineci, “A 50-kW air-cooled SiC inverter with 3-D printing enabled power module packaging structure and genetic algorithm optimized heatsinks,” *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 6256–6265, 2019.

[25] M. Ali, A. Iqbal, M. A. Anees, M. R. Khan, K. Rahman, and M. Ayyub, “Differential evolution-based pulse-width modulation technique for multiphase MC,” *IET Power Electronics*, vol. 12, no. 9, pp. 2224–2235, 2019.

[26] J. P. F. Trovão, V. D. N. Santos, P. G. Pereirinha, H. M. Jorge, and C. H. Antunes, “A simulated annealing approach for optimal power source management in a small EV,” *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 867–876, 2013.

[27] S. Yang, M. Wu, X. Yao, and J. Jiang, “Load modeling and identification based on ant colony algorithms for EV charging stations,” *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1997–2003, 2015.

[28] Q. C. Duan, M. X. Mao, P. Duan, and B. Hu, “Application of improved radial basis function neural network method in global MPPT for PV array,” in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 3260–3264, Montreal, QC, Canada, 2015.

[29] H. Hichem, B. R. Chiheb, and Z. F. Abderrahmen, “Performance improvement of a photovoltaic system with a radial basis function network based on particle swarms optimization,” in *Proc. Int. Conf. Signal, Control and Communication (SCC)*, pp. 156–162, Hammamet, Tunisia, Tunisia, 2019.

[30] B. N. Roodsari, C. J. B. Macnab, and E. P. Nowicki, “A novel adaptive controller using radial basis function neural network for the wind energy conversion system,” in *Proc. IEEE Int. Conf. Industrial Technology (ICIT)*, pp. 715–720, Toronto, ON, Canada, 2017.

[31] E. Kass, T. Eden, and N. Brown, *Analysis of Neural Data*, Springer-Verlag, New York, 2014.

[32] L. Chen, C. Liu, R. Wu, Y. He, and Y. Chai, “Finite-time stability criteria for a class of fractional-order neural networks with delay,” *Neural Computing and Applications*, vol. 27, no. 3, pp. 549–556, 2016.

[33] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Pearson Education, New Delhi, India, 2015.

[34] K. L. Du, *Neural Networks in a Softcomputing Framework*, Springer-Verlag, London, 2006.

[35] K. Ishaque and Z. Salam, “A Deterministic Particle Swarm Optimization Maximum Power Point Tracker for Photovoltaic System under Partial Shading Condition,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3195–3206, 2013.

[36] K. E. Parsopoulos and M. N. Vrahatis, *Particle Swarm Optimization and Intelligence: Advances and Applications*, Information Science Reference, Hershey, PA, USA, 2010.

[37] R. Ruiz-Cruz, E. N. Sanchez, F. Ornelas-Tellez, A. G. Loukianov, and R. G. Harley, “Particle swarm optimization for discrete-time inverse optimal control of a doubly fed induction generator,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1698–1709, 2013.

[38] Q. Niu, Z. Zhou, H. Y. Zhang, and J. Deng, “An improved quantum-behaved particle swarm optimization method for economic dispatch problems with multiple fuel options and valve-points effects,” *Energies*, vol. 5, no. 9, pp. 3655–3673, 2012.

[39] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, “Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection,” *Evolutionary Computation*, vol. 20, no. 3, pp. 349–393, 2012.

[40] C. L. Chiang, “Quantum-behaved particle swarm optimization for power economic dispatch problem of units with multiple fuel option,” *European Journal of Engineering Research and Science*, vol. 2, no. 12, 2017.

[41] X. S. Yang, *Nature-Inspired Algorithms and Applied Optimization*, Springer, New York, 2018.

[42] M. Premkumar and R. Sumithira, “Humpback whale assisted hybrid maximum power point tracking algorithm for partially shaded solar photovoltaic systems,” *Journal of Power Electronics*, vol. 18, no. 6, pp. 1805–1818, 2018.

[43] M. Premkumar and R. Sowmya, “An effective maximum power point tracker for partially shaded solar photovoltaic systems,” *Energy Reports*, vol. 5, pp. 1445–1462, 2019.

[44] M. Premkumar, A. M. Ibrahim, R. M. Kumar, and R. Sowmya, “Analysis and simulation of bio-inspired intelligent salp swarm MPPT method for the PV systems under partial shaded conditions,” *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 489–496, 2020.

[45] M. H. Parvaneh and P. G. Khorasani, “A new hybrid method based on fuzzy logic for maximum power point tracking of photovoltaic systems,” *Energy Reports*, vol. 6, pp. 1619–1632, 2020.

[46] J. Falin, “Designing DC/DC converters based on ZETA topology,” *Analog Applications Journal, Texas Instruments*

Incorporated, pp. 16–21, 2010, https://www.ti.com/lit/an/slyt372/slyt372.pdf?ts=1605727121467&ref_url=https%253A%252F%252Fwww.google.com%252F.

[47] T. M. Leandro, R. Matthias, H. F. Marcelo, S. M. Lucas, E. B. Fábio, and K. Dirk, "Modeling and design of a linear-assisted Zeta converter," in *Proc. Int. Conf. 21st European Conference on Power Electronics and Applications (EPE '19 ECCE Europe)*, pp. 1–10, Genova, Italy, Italy, 2019.

[48] P. Amit and T. Harpal, "Implementation of INC-PIMPPT and its comparison with INC MPPT by direct duty cycle control for solar photovoltaics employing Zeta converter," in *Proc. Int. Conf. Information, Communication, Instrumentation and Control (ICICIC)*, pp. 1–6, Indore, India, 2017.

[49] P. R. Babu, S. R. Prasath, and R. Kiruthika, "Simulation and performance analysis of CCM Zeta converter with PID controller," in *Proc. Int. Conf. Circuits, Power and Computing Technologies (ICCPCT-2015)*, pp. 1–7, Nagercoil, India, 2015.

[50] A. A. Ahmad, A. Abrishamifar, and M. Farzi, "A new design procedure for output LC filter of single phase inverters," in *Proc. Int. Conf. Power Electronics and Intelligent Transportation System*, pp. 86–91, Shenzhen, China, 2010.

[51] P. A. Dahono, A. Purwadi, and Qamaruzzaman, "An LC filter design method for single-phase PWM inverters," in *Proc. Int. Conf. Power Electronics and Drive Systems*, pp. 571–576, Singapore, 1995.

[52] H. S. Kim and S. K. Sul, "A novel filter design for output LC filters of PWM inverters," *Journal of Power Electronics*, vol. 11, no. 1, pp. 74–81, 2011.

[53] T. Aleksei, P. Eduard, and B. Juri, "Closed-loop identification of fractional-order models using FOMCON toolbox for MATLAB," in *Proc. Int. Conf. 14th Biennial Baltic Electronic Conference (BEC)*, pp. 213–216, Tallinn, Estonia, 2014.

[54] M. Deep, K. K. Palash, and G. Apurba, "PID controller design for an interacting tank level process with time delay using MATLAB FOMCON toolbox," in *Proc. Int. Conf. 2nd Control, Instrumentation, Energy & Communication (CIEC)*, pp. 1–5, Kolkata, India, 2016.

[55] T. Aleksei, P. Eduard, B. Juri, and F. Jevgeni, "Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB," in *Proc. IEEE Int. Conf. Computer Aided Control System Design (CACSD)*, pp. 340–345, Hyderabad, India, 2013.

[56] T. Aleksei, P. Eduard, and B. Juri, "A flexible MATLAB tool for optimal fractional-order PID controller design subject to specifications," in *Proc. Int. Conf. 31st Chinese Control*, pp. 4698–4703, Hefei, China, 2012.

[57] M. Kuhn, "Building predictive models inRUsing thecaretPackage," *Journal of Statistical Software*, vol. 28, no. 5, 2008.

[58] J. Zhou, E. Li, H. Wei, C. Li, Q. Qiao, and D. J. Armaghani, "Random forests and cubist algorithms for predicting shear strengths of rockfill materials," *Applied Sciences*, vol. 9, no. 8, p. 1621, 2019.

[59] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: an applied review," *International Journal of Remote Sensing*, vol. 39, no. 9, pp. 2784–2817, 2018.

[60] J. K. Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design*, Analysis and MATLAB Simulation, Springer-Verlag, Berlin, 2013.

[61] B. Liang, Y. Zhu, Y. Li, P. He, and W. Li, "Adaptive nonsingular fast terminal sliding mode control for braking systems with electro-mechanical actuators based on radial basis function," *Energies*, vol. 10, no. 10, p. 1637, 2017.

[62] J. Zhu, Z. Cao, T. Zhang, Y. Yang, and Y. Yi, "Sufficient condition for the existence of the compact set in the RBF neural network control," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–6, 2017.

[63] Z. Xu, G. Han, H. Zhu, L. Liu, and M. Guizani, "Adaptive DE algorithm for novel energy control framework based on edge computing in IIoT applications," *IEEE Transactions on Industrial Informatics*, pp. 1–11, 2020.

[64] Z. Huang, Z. Lin, Z. Zhu, and J. Chen, "An improved simulated annealing algorithm with excessive length penalty for fixed-outline floorplanning," *IEEE Access*, vol. 8, pp. 50911–50920, 2020.

[65] B. T. M. Anh, "Enhanced genetic algorithm for automatic generation of unit and integration test suite," in *Proc. Int. Conf. Computing and Communication Technologies (RIVF)*, pp. 1–6, Ho Chi Minh, Vietnam, 2020.

[66] T. Djemai, P. Stolf, T. Monteil, and J. M. Pierson, "A discrete particle swarm optimization approach for energy-efficient IoT services placement over fog infrastructures," in *Proc. Int. Symp. Parallel and Distributed Computing (ISPDC)*, pp. 32–40, Amsterdam, Netherlands, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8881112, 16 pages
https://doi.org/10.1155/2020/8881112

Research Article

High-Dimensional Text Clustering by Dimensionality Reduction and Improved Density Peak

Yujia Sun [1,2] and Jan Platoš [1]

[1] *Department of Computer Science, Technical University of Ostrava, 17.listopadu 2172/15, Poruba, Ostrava 70800, Czech Republic*
[2] *Institute of Network Information Security, Hebei GEO University, No. 136 East Huai'an Road, Shijiazhuang Hebei 050031, China*

Correspondence should be addressed to Yujia Sun; yujia.sun.st@vsb.cz

Received 30 May 2020; Revised 21 September 2020; Accepted 20 October 2020; Published 28 October 2020

Academic Editor: Chao-Yang Lee

This study focuses on high-dimensional text data clustering, given the inability of K-means to process high-dimensional data and the need to specify the number of clusters and randomly select the initial centers. We propose a Stacked-Random Projection dimensionality reduction framework and an enhanced K-means algorithm DPC-K-means based on the improved density peaks algorithm. The improved density peaks algorithm determines the number of clusters and the initial clustering centers of K-means. Our proposed algorithm is validated using seven text datasets. Experimental results show that this algorithm is suitable for clustering of text data by correcting the defects of K-means.

1. Introduction

Clustering is the main technique used for unsupervised information extraction. In clustering, the aim is to divide the unlabelled dataset into multiple nonoverlapping class clusters, making the data points in the cluster as similar as possible, while making the data points between the clusters as different as possible. In text clustering, text vectors are characterized by high dimension, sparsity, and correlation among dimensions, which requires improvements to the clustering algorithm to process high-dimension text [1, 2].

When the K-means method is used to process high-dimensional data, the "Curse of Dimensionality" [3] problem becomes prominent, and the redundancy index also increases. Consequently, the conventional clustering method cannot process the data accurately. Some research [4–9] has proposed improvements on the text clustering algorithm, and some studies [10, 11] have proposed improvements on the K-means algorithm. To apply the K-means, it is necessary to specify the number of clusters in advance and randomly select the initial clustering centers. The clustering result is greatly influenced by the selection of the initial center point. Improper selection of the initial center can easily cause the clustering result trap into the local optimal solution and lead to an inaccurate clustering result.

In recognition of these problems, we propose an enhanced K-means text clustering algorithm based on the clustering by fast search and find of density peaks (DPC) algorithm [12]. Since text-based data is usually high-dimensional and sparse, we propose a deep random projection dimensionality reduction framework, named Stacked-Random Projection (SRP), a greedy layer-wise architecture. We first use the dimensionality reduction method to reduce the dimension of the high-dimensional text feature vectors. Then use the improved density peaks algorithm to determine the number of clusters and the initial clustering centers, after which the K-means algorithm is used for clustering.

The organization of this paper is as follows. The proposed methodology is discussed in Methods. In Experiments and Discussion, experimental results are explained. Finally, Conclusions concludes the paper and highlights future work related to the study.

2. Methods

2.1. Stacked-Random Projection. The basic idea of random projection is to choose a random hyperplane to map original

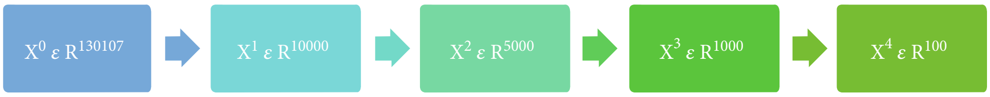

$X^0 \varepsilon R^{130107}$
$X^1 \varepsilon R^{10000}$
$X^2 \varepsilon R^{5000}$
$X^3 \varepsilon R^{1000}$
$X^4 \varepsilon R^{100}$

Figure 1: The SRP dimensionality reduction process for the 20-newsgroups dataset. The 4-layer SRP realizes the dimensionality reduction, using the random projection method from 130,107 down to 10k, down to 5k, down to 1k, and down to 100.

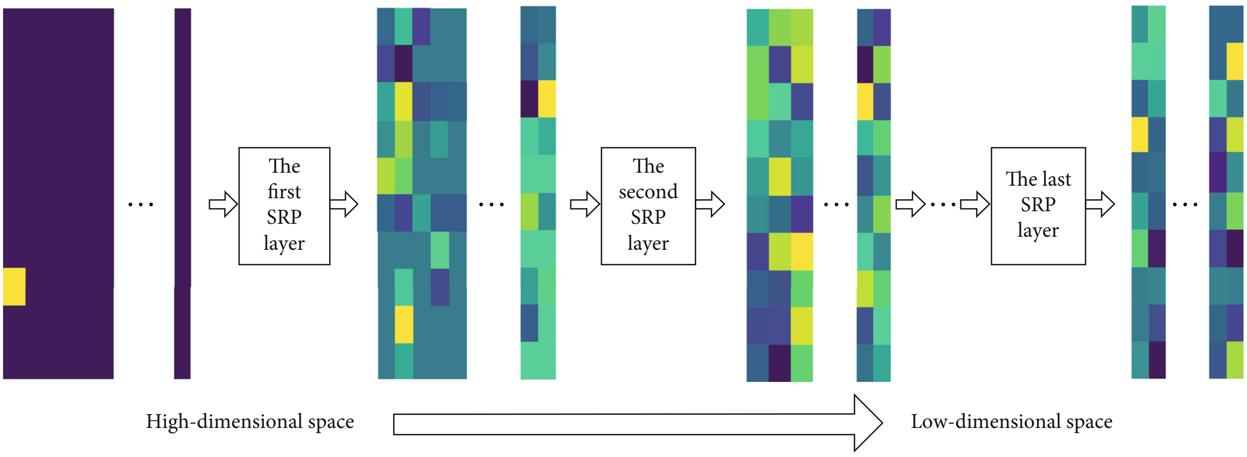


Figure 2: Architecture of Stacked-Random Projection.

variables into a low-dimensional space. In 1948, Johnson and Lindenstrauss proposed a theorem, nowadays termed the Johnson-Lindenstrauss lemma (JL) [13]. JL lemma is the theoretical basis of random projection, which guarantees that the subspace errors generated by random projection are controllable. The JL lemma states that for any $0 < \varepsilon < 1$, and any integer n, let k be a positive integer such that

$$k \geq \left(\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3} \right)^{-1} \ln n. \tag{1}$$

Then, for any n-point set $\mathbf{V}$ in $\mathbf{R}^d$, there is a map f: $\mathbf{R}^d \longrightarrow \mathbf{R}^k$, such that for all u, $v \varepsilon \mathbf{V}$,

$$(1\text{-}\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2. \tag{2}$$

It indicates that by using random projection, the original high-dimensional data is reduced to low-dimensional data, and the distance between the original data is maintained approximately with a high probability. Zhang et al. [14] proposed a random projection ensemble approach and applied it to the prediction of drug-target interaction. Gondara [15] also proposed an ensemble random projection, in which the random projection matrix is applied to different subsets of the original dataset, and which can achieve greater classification accuracy compared with the random forest and AdaBoost methods.

According to the Johnson-Lindenstrauss lemma, the minimum size of the target dimension after dimensionality reduction that guarantee the ε embedding is given by Equation (3):

$$\text{dimension} \geq \frac{4 \log \left(n_{\text{samples}} \right)}{\left(\varepsilon^2/2 - \varepsilon^3/3 \right)}. \tag{3}$$

For example, where n_{samples} is the number of samples, it would require at least 6,515 dimensions to project 2k samples without too much distortion ($\varepsilon = 0.1$). Thousands of dimensions are still high-dimensional data for the following step such as classification or clustering. Inspired by stacked AutoEncoder, we propose a deep random projection framework, named Stacked-Random Projection (SRP), which incorporates random projection as its core stacking element. The SRP framework with k layers uses the input data as the first layer, and the output of the lth $(l < k)$ layer is taken as the $(l+1)$ layer input. In this way, a group of random projections method can be combined layer by layer in a stack.

The main idea of the SRP dimensionality reduction method based on the high-dimensional text feature vector can be illustrated by means of taking the 20-newsgroups dataset as an example (further details are provided in Experiments and Discussion). First, the dataset is subjected to tokenization, stop-words removal, and TF-IDF in order to obtain the high-dimensional sparse text vector space (the feature dimension of the 20-newsgroups dataset was found to be 130,107). Then, a 4-layer SRP is constructed, this process is shown in Figure 1. Thus, the dimensionality reduction process from high dimensionality to low dimensionality is completed. The illustration of our proposed SRP is provided in Figure 2.

2.2. Improved DPC. The DPC algorithm is a granular computing model based on two assumptions: (1) the clustering center is surrounded by neighbour data points with lower local density; (2) the distance between any clustering center and data points with higher density is relatively far. In recent years, DPC has been applied in many fields, particularly natural language processing, due to its process and its effectiveness. The DPC algorithm can cluster data of different dimensions and shapes. At present, many researchers have researched DPC and have also proposed many improved algorithms. The main optimization aspects are speed improvement [16], accuracy improvement [17–19], and other aspects [20, 21]. Heimerl et al. [22] applied the DPC algorithm in the high-dimensional space to estimate the optimal cluster numbers for a given set of documents and assigned stability to one of the peaks based on the density structure of the data; however, the resulting computing speed of the DPC algorithm in the high-dimensional space was slow. Wang et al. [23] used DPC to measure the hierarchical relevance and diversity of sentences and selected highly representative sentences to generate news summaries. However, they reported that if there are multiple peaks in the sentence, then the key sentence will be redundant.

For any point *i*, two properties of the local density and relative distance are required. The calculation of these two attributes depends on the distance between any two points v_i and v_j in the graph. The two attributes are defined as follows:

Definition 1. local density ρ_i (Gaussian kernel):

$$\rho_i = \sum_j e^{-\left(\frac{d_{ij}}{d_c}\right)^2}, \tag{4}$$

where d_{ij} is the Euclidean distance between v_i and v_j, and d_c is the cut-off distance; these are important parameters for calculating ρ_i. One recommended practice is to select d_c so that the average nearest neighbour from each point is 1%~2% of the total dataset size. As can be seen in Equation (4), the more points i contained in d_c, the greater the local density ρ.

Of the text clustering methods, the K-means method based on cosine similarity is still the most widely used text clustering algorithm due to its simplicity and fast convergence [24]. For text vectors, using cosine similarity has a better effect than Euclidean distance. Euclidean distance is a direct measure of the linear interval or length between vectors and is an absolute value of the difference in dimensional values. Cosine similarity describes the similarity between vectors using the cosine value of the angle, that is, the direction, and pays more attention to the difference between the relative levels of the dimensions. In text similarity analysis, one feature of similarity is the occurrence of the same words at the same time, which translates into nonzero values for the same dimension at the same time. We therefore redefine Definition 1 in terms of cosine similarity.

Definition 2. Local density ρ_i based on cosine similarity (Gaussian kernel):

For any two vectors in space $v_i = (x_1, x_2, \cdots, x_n)$ and $v_j = (y_1, y_2, \cdots, y_n)$, the cosine similarity is defined as the cosine of the angle between the two vectors:

$$\cos\,(i, j) = \frac{\sum_{k=1}^{n} x_k y_k}{\sqrt{\sum_{k=1}^{n} x_k^2}\sqrt{\sum_{k=1}^{n} y_k^2}} = \frac{\sum_{k=1}^{n} x_k y_k}{\|x\|\bullet\|y\|}, \tag{5}$$

$$\rho_i = \sum_j e^{-\left(\frac{\cos\,(i,j)}{\cos_c}\right)^2}, \tag{6}$$

where $\cos\,(i, j)$ is the cosine similarity between v_i and v_j, and $\cos_c$ is the cut-off distance which needs to manually set the value to the nearest neighbour number of the sample approximately 1%~2% of the size of the entire dataset. As can be seen in Equation (6), the more points i contained in $\cos_c$, the greater the local density ρ.

Definition 3. Relative distance δ_i:

$$\delta_i = \begin{cases} \max\limits_j \cos\,(i, j) & \rho_{\mathrm{i}} \text{is the maximum} \\ \min\limits_{j:\rho_i<\rho_j} \cos\,(i, j) & \text{otherwise.} \end{cases} \tag{7}$$

Equation (7) indicates that cosine similarity distance δ_i can be obtained by calculating the minimum distance from the data point x_i to any point with a density greater than that. After calculating the two parameters, a decision graph with ρ as the horizontal axis and δ as the vertical axis can be constructed. By observing the decision graph, the decision graph divides the data points into three different types, namely the density peak point, the normal point, and the outlier point. As shown in Figure 3, the data points are arranged in the order of decreasing density. There are five points that stand out, which are spread out towards the upper right corner of the decision graph, with varying high ρ values and higher δ values. These five points indicate that there are no data points with higher density than these five points in a larger area. Therefore, these five points are the so-called peak density points, and so they make a suitable clustering center. In order to better verify the accuracy of the clustering center point in the decision graph, the DPC define another variable $\gamma = \rho * \delta$, where a clustering center point has a large ρ value and δ value, the clustering center has a higher γ value. We conclude from our analysis of the decision graph and that ρ and δ are of two different orders of magnitude. To avoid the influence of different orders of magnitude, it is necessary to normalize them.

$$\rho_i{}' = \frac{\rho_i - \rho_{\min}}{\rho_{\max} - \rho_{\min}}, \tag{8}$$

$$\delta_i{}' = \frac{\delta_i - \delta_{\min}}{\delta_{\max} - \delta_{\min}}, \tag{9}$$

$$\gamma_i = \rho_i{}'\delta_i{}'. \tag{10}$$

The γ values for Equation (10) are plotted in Figures 4 and 5. Figure 4 verifies the correctness of the clustering centers in the decision graph shown in Figure 3. Figure 5 is plotted according to the descending order of γ value, and it can be noted that γ value changes from large to small. The point at the clustering center has an enormous γ value, while the noncenter point has a smaller γ value, and the change tends to be flat. It can be concluded that according to the γ value, there are five clustering centers.

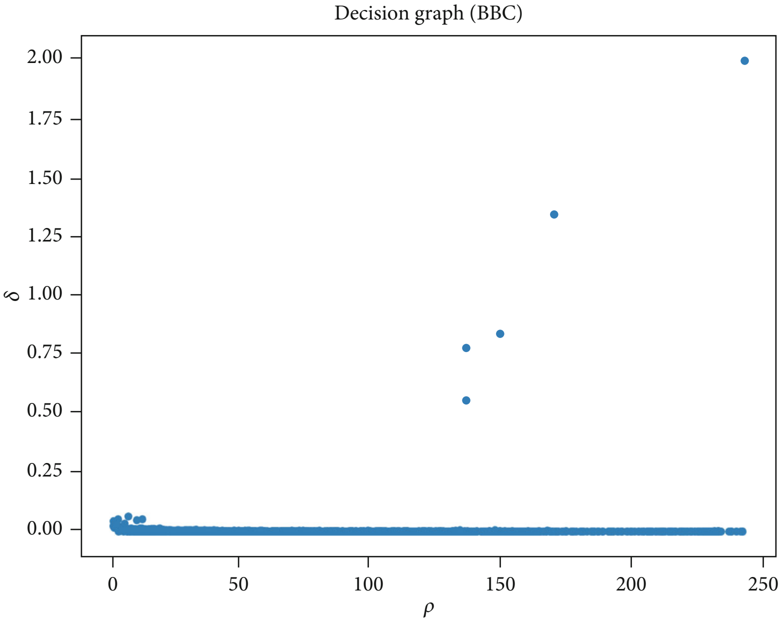


Figure 3: The decision graph of BBC dataset.

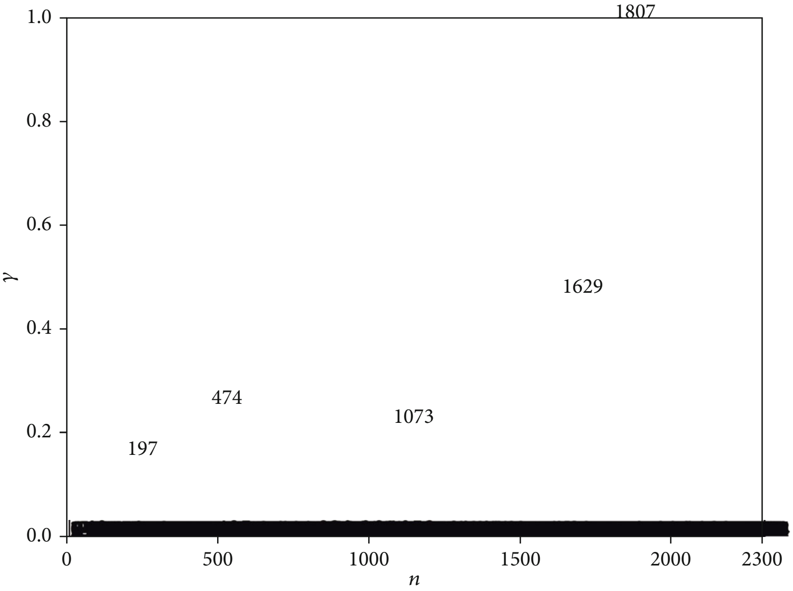


Figure 4: The value of γ according to the $\rho * \delta$ in Figure 3.

2.3. DPC-K-means. The K-means clustering algorithm cannot extract data features effectively when processing high-dimensional data directly, and problems also occur when it randomly selects initial clustering centers and specifies the number of clustering in advance. These problems have been researched in numerous papers over the recent decades, as discussed elsewhere [25–27]. Therefore, we propose an improved method using the DPC algorithm.

We first use SRP or random projection to reduce the dimensionality of high-dimensional text data and then combine it with the improved DPC algorithm. The choice of dimensionality reduction method SRP or random projection depends on whether the feature vector dimension is greater than the target dimension calculated according to Formula (3). If the feature vector dimension is greater than the minimum size of the target dimension, the SRP dimension reduction framework is performed. If the feature vector dimension is less than or equal to the target dimension, random projection is used directly. Using the cosine similarity calculation of ρ and δ, we select some points with high local density, which are far apart from each other as the clustering center; by doing so, the initial clustering center and the number of clusters can be obtained automatically; this makes the clustering algorithm, which we name the DPC-K-means. The improved algorithm is described below:

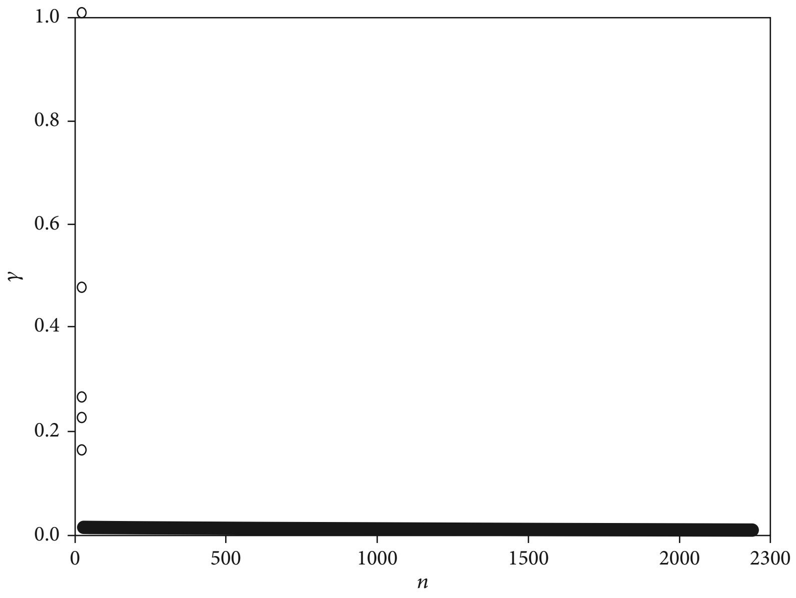


Figure 5: The value of γ of Figure 4 in decreasing order.

```
The DPC-K-means.
Input: text feature vector $\mathbf{A} \in \mathbf{R}^{n\times d}$, $t$ is the minimum size of the target dimension.
Output: the clustering results.
Begin:
Step1: determine whether $d$ is greater than $t$ calculated according to Formula (3). If $d$ is greater than $t$, use the SRP dimension reduction framework in Step2. If $d$ is less than or equal to $t$, random projection is used in Step2.
Step2: the SRP dimension reduction framework is used to reduce the dimensionality of A layer by layer, until matrix $\mathbf{A}^{'}$ after dimension reduction is obtained. Or directly use random projection to reduce the dimension to get the matrix $\mathbf{A}^{'}$.
Step3: Calculate the $\rho$ value and $\delta$ value of $\mathbf{A}^{'}$ according to Equations (6) and (7) and plot the decision graph with $\rho$ and $\delta$ axes.
Step4: calculate the $\gamma$ value according to Equation (10) to verify the clustering centers and the number of clusters.
Step5: perform K-means clustering: the clustering centers obtained in Step4 are used as the initial cluster centers, and the number of clusters is used as the $k$ value for K-means clustering.
```

Algorithm 1:

Suppose n input data, the original dimension $d, t^{'}$ is the dimension of implementing SRP or random projection to reduce dimension to low-dimensional space, and the time complexity analysis of DPC-K-means algorithm is as follows:

(1) The time complexity of a single random projection in Step2 is $O(ndt^{'})$. The time complexity of Stacked-Random Projection is $O(ndh + nhl + \cdots + nrt^{'})$ (l is the target dimension of the second layer, and r is the target dimension for the penultimate layer)

(2) The time complexity of Step3 is to calculate ρ and δ, which is $O(n^2)$

(3) The time complexity of Step4 is to calculate γ and sort γ in descending order, which is $O(n \log_2 n)$

(4) The time complexity of Step5 K-means for specifying the cluster center and the number of clusters is $O(knt^{'})$.

The total time complexity of DPC-K-means algorithm is $O(n^2)$.

Figure 6 shows the overall structure of the proposed method. Table 1 shows the time complexity of several clustering algorithms.

3. Experiments and Discussion

3.1. Summarization Datasets. Experimental work was conducted on seven standard text datasets. The summary of datasets is presented in Table 2. Datasets are described as follows. The features are obtained by tokenization, stop-words removal, and TF-IDF.

The BBC news dataset (http://mlg.ucd.ie/datasets/bbc.html.) has a total of 2,225 text files on five topical areas published on the BBC news website. Text documents were arranged into folders containing five labels: business, entertainment, politics, sports, and technology.

The 20-newsgroups dataset (http://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets) of approximately 20k newsgroup documents was partitioned evenly across the 20 different newsgroups. We selected 1k

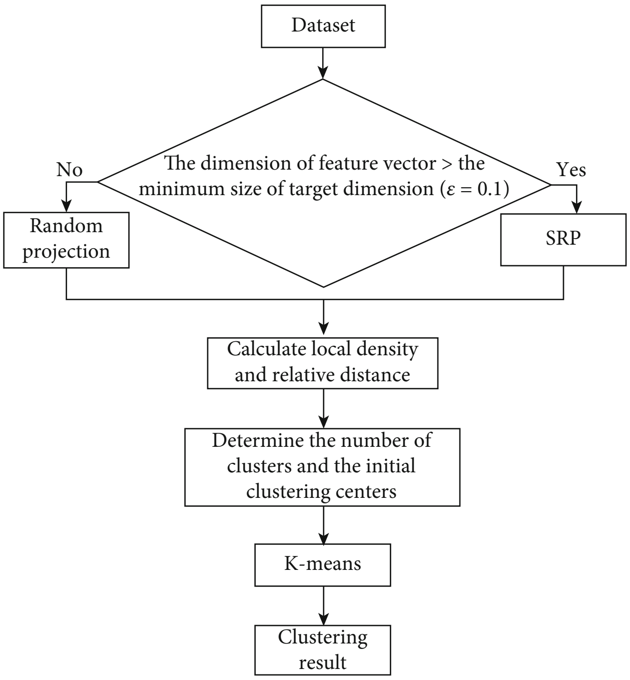

Dataset
The dimension of feature vector > the minimum size of target dimension (ε = 0.1)
No
Yes
Random projection
SRP
Calculate local density and relative distance
Determine the number of clusters and the initial clustering centers
K-means
Clustering result

Figure 6: The detailed process of DPC-K-means.

documents and 4~8 various newsgroups (4 groups~8 groups) for our experimental dataset.

The Sports Article dataset (http://archive.ics.uci.edu/ml/datasets.php) was labelled using Amazon Mechanical Turk as objective or subjective.

The Asian Religious (http://archive.ics.uci.edu/ml/datasets.php) dataset was the words from the bag of words preprocessing of the mini-corpus made up of eight religious books.

The CNAE-9 dataset (http://archive.ics.uci.edu/ml/datasets.php) contains 1,080 documents of free text business descriptions of Brazilian companies which were categorized into a subset of nine categories.

The Stack Overflow dataset (http://www.kaggle.com/c/predict-closed-questions-on-stack-overflow/download/train.zip) is challenge data published on http://Kaggle.com/. The dataset consists of 3,370,538 samples dated from July 31, 2012, to August 14, 2012. In our experiments, we randomly selected 167 question titles from 4 different tags.

The Amazon dataset (http://archive.ics.uci.edu/ml/datasets.php) is the product reviews extracted from websites and marked with positive and negative.

3.2. Simulation Environments. The simulation environments for all algorithms performed in our experiments were as follows: the Python 3.7 software environment running with Intel i7-7500U CPU, 2.70GHz with 8GB RAM.

3.3. Experiment 1. According to Formula (3), the minimum size of the target dimension ($\varepsilon = 0.1$) of the BBC and 20-newsgroups datasets is 6,609 and 5,920. According to the flowchart Figure 6, the feature vector dimensions of the two datasets are larger than the minimum size of the target dimension, so that SRP was used to reduce the dimensionality of these two datasets. We compared the dimensionality reduction performance of Principal Component Analysis (PCA), Multiple Dimensional Scaling (MDS), Random Projection (RP), and Stacked-Random Projection (SRP). To correctly compare the performance of these dimensionality reduction methods, we experimentally reduced the feature vector of the BBC news dataset and 20-newsgroups dataset to 2k, 500, and 100. Table 3 shows the run time (time), mean ratio of distances (projected/original, ratio), and the standard deviation of ratio of distances (projected/original, standard deviation). The mean ratio of distances is the degree to which the distance between the original data is maintained in the low-dimensional space when the original high-dimensional data is reduced to low-dimensional data. The value is approximately close to 1, indicating better preservation. The smaller the standard deviation of the ratio of distances, the closer is it to the mean ratio of distances. As shown in Table 3, RP and SRP considerably shorten the run time of dimension reduction compared with PCA and MDS. We can see that there is little difference in the distribution of the distortion between SRP and RP for high values of the dimension. But for low values of the dimension, the distortion distribution is controlled, and the distances are well preserved by the SRP. Text data is usually high-dimensional and small-sampling data. The characteristic of high-dimensional and small-sampling data is that the number of dimensions is much larger than the number of samples. SRP is suitable for dimensionality reduction of this type of data, which significantly reduces the running time of dimensionality reduction, and the distances are well preserved.

3.4. Experiment 2. Since DPC is a clustering algorithm, we use the Euclidean distance and cosine similarity to calculate DPC local density ρ_i and observe the difference between these two methods by clustering performance metrics. According to Formula (3), the minimum size of the target dimension ($\varepsilon = 0.1$) of the BBC and 20-newsgroups datasets is 6,609 and 5,920. According to the flowchart Figure 6, SRP was used to reduce the dimensionality of these two datasets to 100 dimensions. The minimum size of the target dimension of Sports Article, CNAE-9, and Stack Overflow datasets is larger than the feature vector dimension, so that the dimension can be reduced by random projection to 100 dimensions. The feature dimensions of the Asian Religions and Amazon datasets are ≤100, so there is no need for dimension reduction in this experiment. To correctly compare these two methods' performance, we used the four cluster evaluation metrics—ARI (Adjusted Rand Index), NMI (Normalized Mutual Information), FMI (Fowlkes-Mallows Index), and Clusters (the number of clusters)—to evaluate the performance of the clustering algorithm. ARI, NMI, and FMI are all used to measure the consistency between clustering results and real category data, among which ARI, NMI, and FMI have value ranges of [-1,1], [0,1], and [0,1], respectively. The higher the three evaluation metrics' values, the better the clustering quality, and the more consistent the clustering results are with the real category data. Clusters are the number of clusters after DPC. By comparing with Table 2, we can compare which method of the Euclidean distance and cosine similarity

Table 1: The time complexity of several clustering algorithms.

	DPC-K-means	DPC	K-means	DBSCAN	Spectral Clustering	Affinity Propagation
Time complexity	$O(n^2)$	$O(n^2)$	$O\left(nkt^{'}\right)$	$O(n^2)$	$O(n^3)$	$O(n^2 \log n)$

Table 2: The summary of datasets.

Dataset	Instances	Dimension of features	Clusters	Label
BBC	2,225	11,227	5	Yes
20-newsgroups	1000	13,0107	4~8	Yes
Sports article	1,000	348	2	Yes
Asian Religious	590	39	8	Yes
CNAE-9	1,080	857	9	No
Stack Overflow	167	167	4	Yes
Amazon	100	100	2	Yes

Table 3: The run time, ratio, and standard deviation of each dimension reduction method reduce the dimension to 2,000, 500, and 100.

		Dimension = 2,000			Dimension = 500			Dimension = 100		
		Time (s)	Ratio	Standard deviation	Time (s)	Ratio	Standard deviation	Time (s)	Ratio	Standard deviation
BBC	PCA	30.52	1.00	0.02	15.16	0.57	0.09	4.99	0.23	0.09
	MDS	57.99	1.00	0.02	28.67	1.00	0.07	15.47	1.00	0.07
	RP	1.12	1.00	0.04	0.56	1.00	0.08	0.41	1.01	0.18
	SRP	2.87	1.00	0.05	2.38	1.00	0.07	2.27	1.00	0.12
20-newsgroups	PCA	24.87	1.00	0.00	13.35	1.00	0.08	4.20	1.00	0.1
	MDS	65.27	1.00	0.02	44.74	1.00	0.03	23.13	0.99	0.06
	RP	0.10	0.99	0.06	0.46	1.00	0.12	0.31	0.99	0.28
	SRP	3.33	1.00	0.05	2.85	1.00	0.07	2.72	1.00	0.15

Table 4: The clustering performances of local density calculated by Euclidean distance and cosine similarity.

Dataset	ARI		NMI		FMI		Clusters	
	Euclidean	Cosine	Euclidean	Cosine	Euclidean	Cosine	Euclidean	Cosine
BBC	0.8422	0.9002	0.8223	0.8681	0.8759	0.9204	5	5
4 groups	0.9715	0.9781	0.9523	0.9623	0.9786	0.9836	4	4
5 groups	0.8438	0.8433	0.8411	0.8381	0.8851	0.8846	5	5
6 groups	0.6195	0.6759	0.6874	0.7351	0.7326	0.7700	6	6
7 groups	0.2213	0.5858	0.3039	0.6487	0.4999	0.7031	5	7
8 groups	0.1889	0.4664	0.2672	0.5507	0.4607	0.6138	5	8
Sports Article	0	0	0	0	0.5674	0.7298	1	2
Asian Religious	0.0562	0.0189	0.1288	0.0163	0.3145	0.4665	6	8
Stack Overflow	0	0	0	0	0.3660	0.4399	2	4
Amazon	0	0.0014	0	0.0048	0.5515	0.6696	2	2
CNAE-9	—	—	—	—	—	—	6	9

can cluster accurately. Table 4 shows the clustering performances of local density calculated by the Euclidean distance (Euclidean) and cosine similarity (Cosine).

To further judge the clustering performance proposed in this paper, a paired t-test was used to test the clustering significance. A paired t-test is used to determine whether

Table 5: Table 4's paired t-test results of ARI, NMI, and FMI.

Pairing method	Paired t-test index	ARI	NMI	FMI
Euclidean and Cosine	t	-1.70	-1.40	-4.16
	p	0.1240	0.1958	0.0025

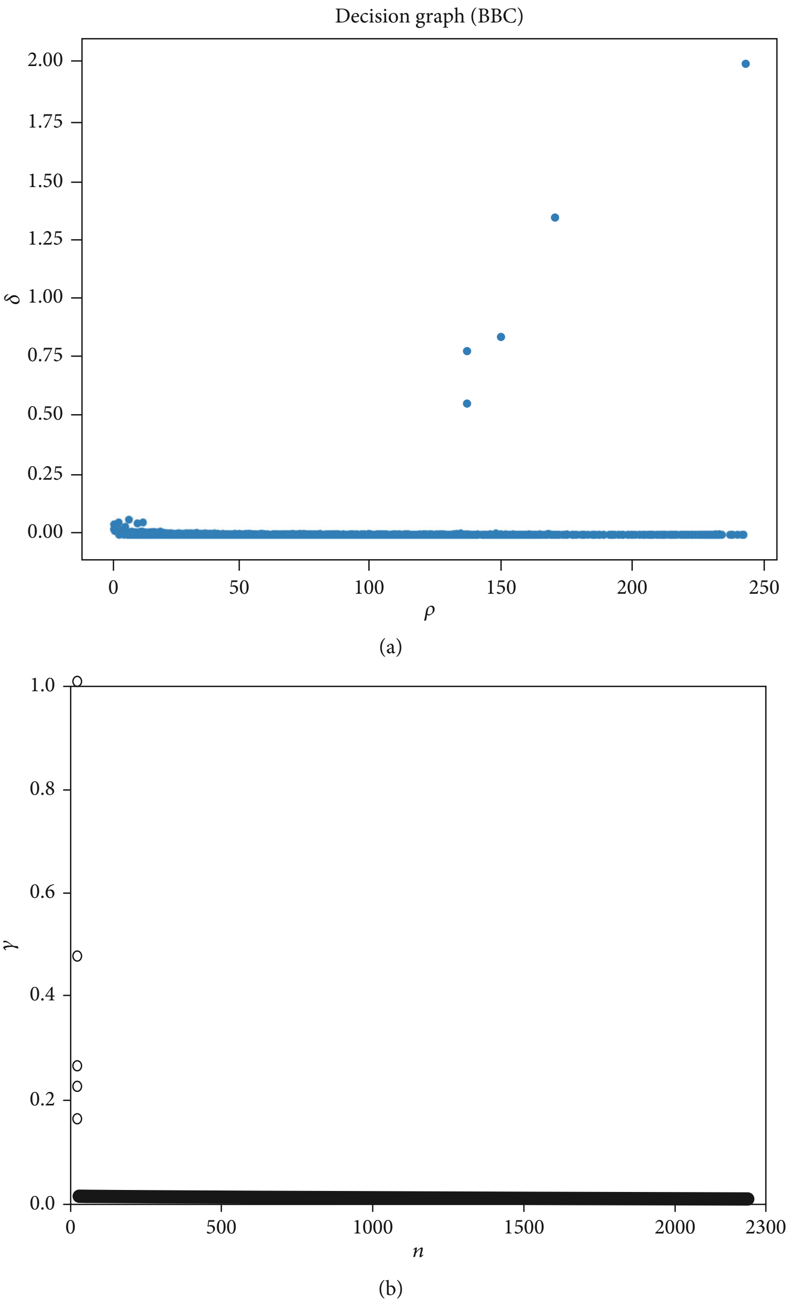
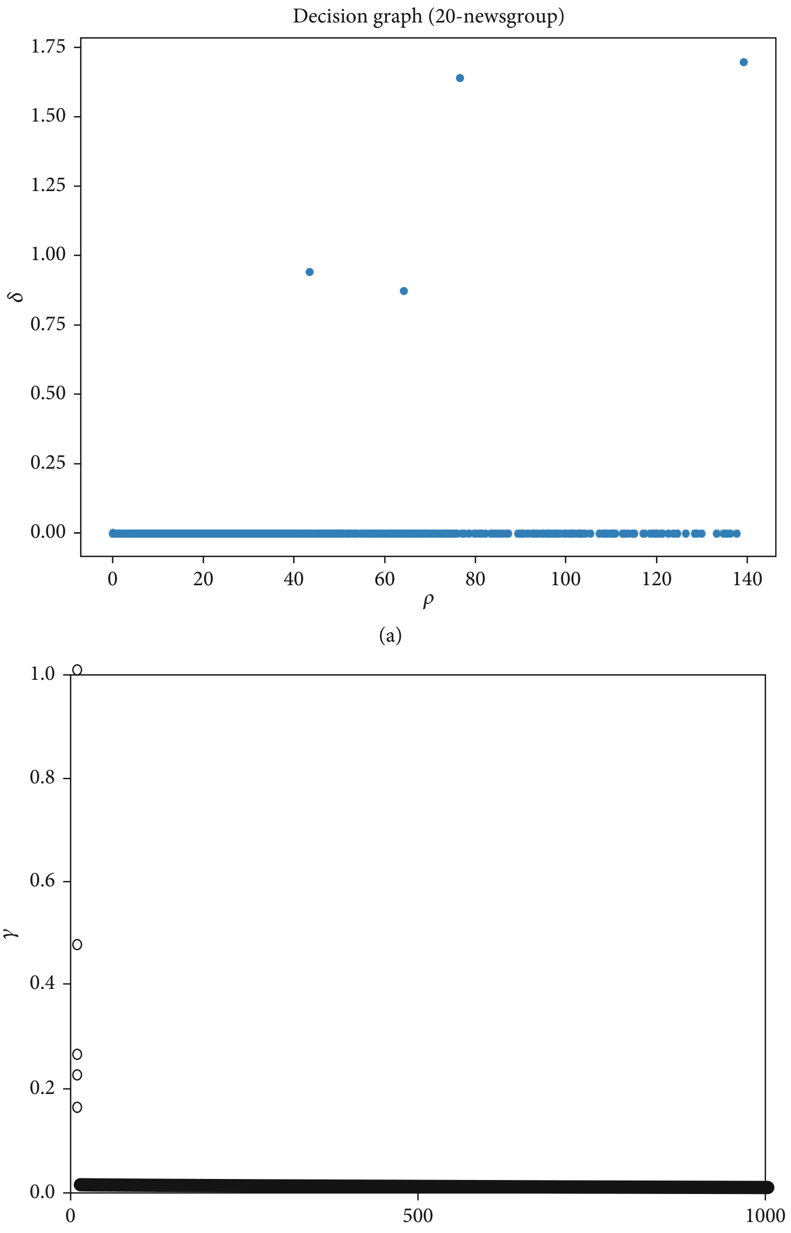


Figure 7: The improved DPC clustering of the BBC dataset. (a) Decision graph. (b) γ value.

there is a significant difference between the two samples. The Euclidean distance and cosine similarity were used to calculate the local density of DPC and test the cluster evaluation metrics. The p value gives the probability of observing the test results under the null hypothesis. The confidence level is at 95%, and the cut-off value of p is 0.05; if $p < 0.05$, the proposed algorithm clustering results and the comparison algorithm are significantly different. If $p \geq 0.05$, there is no significant difference between the proposed algorithm and the comparison algorithm's clustering results. Table 5 shows the paired t-test results of each evaluation metric of Euclidean distance (Euclidean) and cosine similarity (Cosine) in Table 4.

As shown in Table 5, there are substantial differences in FMI between the Euclidean distance and cosine similarity and no significant difference in ARI, NMI. As can be seen from the number of clusters of Tables 2 and 4, the improved DPC of the local density calculated with cosine similarity can accurately determine the number of clusters. Figure 7 shows the decision graph, the γ values of the BBC dataset following dimensionality reduction by SRP. Figure 8 shows the decision graph, the γ values of the four newsgroups in the 20-



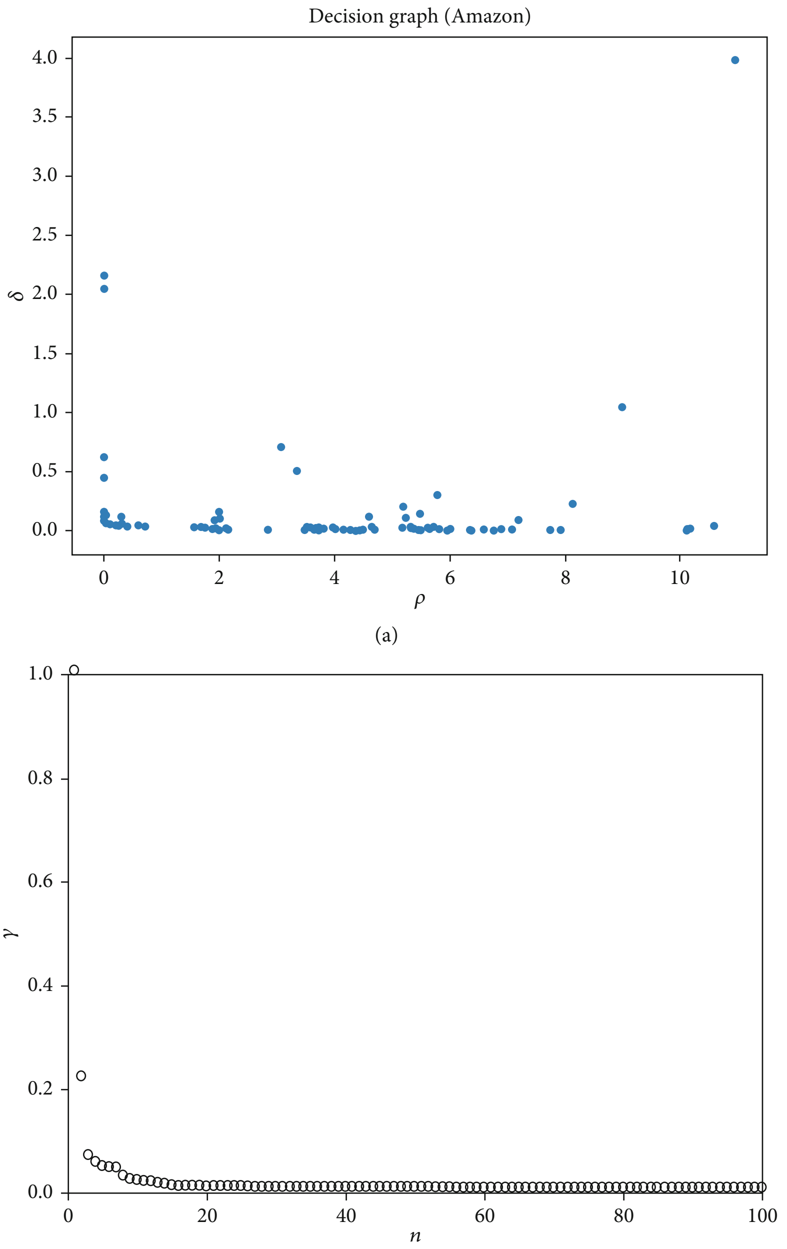
(a)

(b)

Figure 8: The improved DPC clustering of the four newsgroups in the 20-newsgroups dataset. (a) Decision graph. (b) γ value.

newsgroups dataset following dimensionality reduction by SRP. Figure 9 shows the decision graph, the γ values of the Amazon dataset. Figure 10 shows the decision graph, the γ values of the Sports Article dataset. As shown in these figures, improved DPC can accurately determine the dataset of the number of clusters, indicating that using cosine similarity to calculate the local density of DPC is better than using the Euclidean distance. Therefore, cosine similarity is more suitable for text vector calculation.

3.5. Experiment 3. We compared the clustering performance of DPC, DBSCAN, Spectral Clustering, Affinity Propagation, and DPC-K-means. In a comparative study of these clustering algorithms, we used the four evaluation metrics—ARI (Adjusted Rand Index), NMI (Normalized Mutual Information), FMI (Fowlkes-Mallows Index), and MSE (Mean Squared Error)—to evaluate the performance of the clustering algorithm. The mean-square error (MSE) is the average of the sum of squares of the difference between the predicted value and the real value used to measure the expected result. It is nonnegative, and values closer to zero are better. For even comparisons with these methods, we repeated the experiment ten times to obtain the average clustering performance as the final performance of each method. Table 6 shows the ARI of each method. Table 7 shows the NMI of each method. Table 8 shows the FMI of each method. Table 9 shows the MSE of each method.



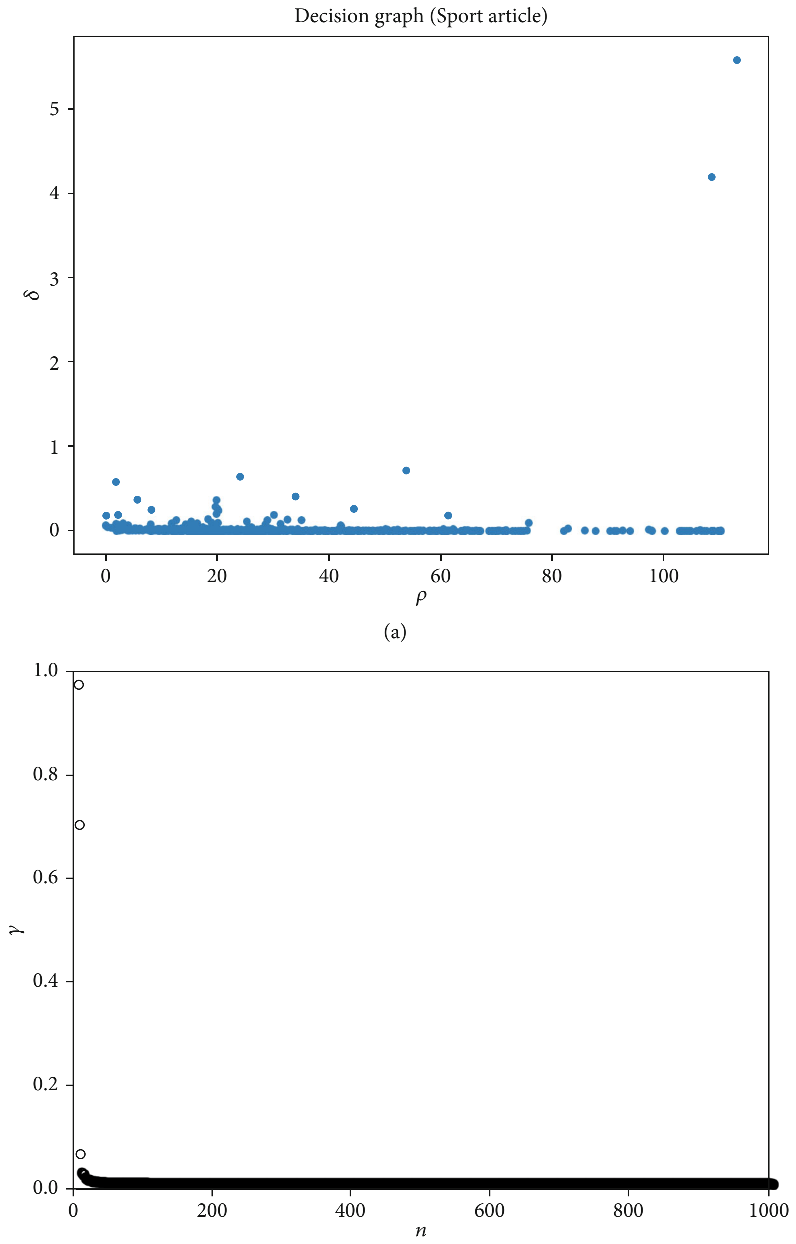
(a)

(b)

Figure 9: The improved DPC clustering of the Amazon dataset. (a) Decision graph. (b) γ value.

To further judge the difference between the clustering results of the algorithm proposed in this paper DPC-K-means and those of other cluster methods, a paired t-test was used to test the clustering results significance. Table 10 shows the paired t-test results of each evaluation metric of these methods in Tables 6–9. The p value gives the probability of observing the test results under the null hypothesis. The confidence level is at 95%, and the cut-off value of p is 0.05; if $p < 0.05$, the proposed algorithm's clustering results and the comparison algorithm are significantly different. If $p \geq 0.05$, there is no significant difference between the proposed algorithm and the comparison algorithm clustering performance.

As shown in Table 10, there are significant changes in NMI, FMI, and MSE metrics between DPC-K-means and comparison methods. DPC-K-means is superior to comparison algorithms in NMI, FMI, and MSE. DPC-K-means compared to DPC and Spectral Clustering has no significant difference in ARI, indicating that DPC and Spectral Clustering are performed as well as DPC-K-means on the ARI metric. A one-sample t-test method is used to evaluate the DPC-K-means algorithm significance on different datasets. Taking the FMI evaluation metric in the BBC dataset as an example of significance testing, the process is as follows: Firstly, a test hypothesis is established and the threshold chosen for statistical significance



(a)

(b)

Figure 10: The improved DPC clustering of the Sports Article dataset. (a) Decision graph. (b) γ value.

was determined ($H_0 : \mu = \mu_0$, $\alpha = 0.1$). Secondly, the t is calculated:

$$\begin{aligned} t &= \frac{\bar{x}-\mu_0}{s}\sqrt{n} = \frac{0.9225-\mu_0}{s}\sqrt{5} \\ &= \frac{0.9225-0.7362}{0.2657}\sqrt{5} = 1.568, \\ v &= 5-1=4, \end{aligned} \tag{11}$$

where $\bar{x}$ represents the FMI value obtained by DPC-K-means on the BBC dataset, μ_0 is the mean FMI of the five comparison algorithms on the BBC dataset, s represents the standard deviation FMI of the five comparison algorithms on the BBC dataset, and n is the sample size. The degree of freedom v used in this test is 4. Finally, the table was queried of t-distribution, the p value was determined, and an inference conclusion was made. According to $\alpha = 0.1$ and $v = 4$, the p value is 1.533, $t = 1.568 > p$, and H_0 is rejected, indicating the difference between the FMI metric of DPC-K-means on the BBC dataset, and the FMI value of other comparison algorithms is statistically significant. According to the above significance test steps, the t-test results of DPC-K-means were calculated on the ARI, NMI, and FMI evaluation metrics. The results are shown in Tables 11–13.

As shown in Table 11, there are significant differences in the ARI metric of DPC-K-means on five datasets, and as

Table 6: The ARI of each clustering algorithm.

Dataset		DPC-K-means	DPC	DBSCAN	Spectral Clustering	Affinity Propagation
BBC		0.9028	0.9002	0.4651	0.8961	0.1477
20-newsgroups	4 groups	0.9783	0.9781	0.7266	0.9756	0.1492
	5 groups	0.8521	0.8433	0.5993	0.8415	0.1508
	6 groups	0.6721	0.6759	0.4166	0.5130	0.1334
	7 groups	0.6078	0.5858	0.4260	0.4914	0.1480
	8 groups	0.4858	0.4664	0.1389	0.4599	0.1589
Sports Article		0.1941	0	0.0354	0.1906	0.0175
Asian Religious		0.1566	0.0189	0	0.1829	0.1064
Stack Overflow		0	0	0.0386	0	0.0349
Amazon		0	0.0014	0	0	0.0114

Table 7: The NMI of each clustering algorithm.

Dataset		DPC-K-means	DPC	DBSCAN	Spectral Clustering	Affinity Propagation
BBC		0.9028	0.8681	0.5652	0.8650	0.3696
20-newsgroups	4 groups	0.9763	0.9623	0.6725	0.9577	0.3628
	5 groups	0.8421	0.8381	0.6709	0.8404	0.3653
	6 groups	0.7721	0.7351	0.5721	0.6987	0.3304
	7 groups	0.7078	0.6487	0.5511	0.6498	0.3602
	8 groups	0.6858	0.5507	0.2846	0.6249	0.3719
Sports Article		0.1870	0	0.1286	0.1849	0.0307
Asian Religious		0.2673	0.0163	0	0.2443	0.2094
Stack Overflow		0	0	0.0518	0.0204	0.0819
Amazon		0	0.0048	0	0	0.0116

Table 8: The FMI of each clustering algorithm.

Dataset		DPC-K-means	DPC	DBSCAN	Spectral Clustering	Affinity Propagation
BBC		0.9225	0.9204	0.5805	0.9172	0.3402
20-newsgroups	4 groups	0.9823	0.9836	0.7930	0.9817	0.3225
	5 groups	0.8864	0.8846	0.6810	0.8831	0.2992
	6 groups	0.8486	0.7700	0.5485	0.6332	0.2593
	7 groups	0.7823	0.7031	0.5415	0.5813	0.2560
	8 groups	0.6535	0.6138	0.3896	0.5468	0.2592
Sports Article		0.7300	0.7298	0.5765	0.6114	0.1653
Asian Religious		0.4802	0.4665	0.4615	0.3833	0.2341
Stack Overflow		0.5512	0.4399	0.4004	0.3816	0.1649
Amazon		0.7192	0.6696	0.7041	0.6892	0.2624

shown in Table 12 there are significant differences in the NMI metric of DPC-K-means on eight datasets. It can be seen from Table 13 that DPC-K-means have significant difference in the FMI metric on the seven datasets. DPC-K-means are statistically significant on most datasets. Combined with Tables 9 and 10, it further shows that DPC-K-means is better than other comparison algorithms.

The clustering performance of DPC-K-means is better than K-means because DPC-K-means can select the number of clusters and obtain the initial clustering center. The

Table 9: The MSE of each clustering algorithm.

Dataset		DPC-K-means	DPC	DBSCAN	Spectral Clustering	Affinity Propagation
BBC		1.0661	3.279	13.2085	6.2378	15.4328
20-newsgroups	4 groups	0.5590	3.2087	6.2040	2.7103	14.6743
	5 groups	1.7203	2.2610	5.8610	3.0047	13.4050
	6 groups	6.6390	8.4530	6.9040	5.0280	13.7290
	7 groups	7.8453	8.0503	9.6103	7.2420	15.9880
	8 groups	5.4723	13.0367	16.6757	7.5143	20.1617
Sports Article		1.3020	2.0950	2.0980	1.8150	8.4410
Asian Religious		5.1898	6.4644	16.4831	11.9678	56.6118
Stack Overflow		3.8084	6.7365	18.0778	11.1916	60.4068
Amazon		0.4700	0.5200	0.5100	0.5300	5.2500

Table 10: Paired t-test results of clustering algorithms.

Pairing method	Paired t-test index	ARI	NMI	FMI	MSE
DPC-K-means and DPC	t	1.73	2.55	2.80	-2.88
	p	0.1171	0.0311	0.0207	0.0181
DPC-K-means and DBSCAN	t	4.39	3.93	5.47	-3.50
	p	0.0017	0.0035	0.0004	0.0067
DPC-K-means and Spectral Clustering	t	1.60	2.60	2.62	-2.35
	p	0.1448	0.0289	0.0056	0.0430
DPC-K-means and Affinity Propagation	t	3.68	3.72	12.52	-3.19
	p	0.0050	0.0047	0.0000	0.0109

Table 11: The results of the t-test of DPC-K-means on ARI.

Dataset		Mean	Standard deviation	t	p	Difference
BBC		0.6624	0.3438	1.398	1.533	No
20-newsgroups	4 groups	0.7617	0.3593	1.348	1.533	No
	5 groups	0.6574	0.3026	1.439	1.533	No
	6 groups	0.4822	0.2293	1.897	1.533	Yes
	7 groups	0.4518	0.1849	1.886	1.533	Yes
	8 groups	0.3420	0.1767	1.820	1.533	Yes
Sports Article		0.0875	0.0965	2.469	1.533	Yes
Asian Religious		0.093	0.0813	1.750	1.533	Yes
Stack Overflow		0.0842	0.1694	1.111	1.533	No
Amazon		0.0026	0.0050	1.150	1.533	No

number of clusters and the initial clustering centers can be used in the K-means algorithm, which achieves better clustering performance than K-means. Figures 11 and 12 illustrate the clustering centers automatically determined by DPC-K-means which are closer to the real class centers.

Tables 6–9 show that the clustering metrics changed significantly from 4 newsgroups to 8 newsgroups; this was caused by the loss of clustering due to irregular data distribution. Due to the inherent nature of the DPC algorithm, it cannot identify the phenomenon of “False peaks,” and its clustering effect on “No density peaks” datasets is low, which are all factors that affect the accuracy of the DPC-K-means algorithm. The algorithm is limited in its processing of more complex datasets.

DPC-K-means has a parameter $\cos_c$, which is the cut-off distance. The value suggested in the literature [12] is set to

Table 12: The results of the t-test of DPC-K-means on NMI.

Dataset		Mean	Standard deviation	t	p	Difference
BBC		0.7141	0.2361	1.787	1.533	Yes
20-newsgroups	4 groups	0.7863	0.2687	1.581	1.533	Yes
	5 groups	0.7114	0.2069	1.413	1.533	No
	6 groups	0.6217	0.1794	1.875	1.533	Yes
	7 groups	0.5835	0.1369	2.029	1.533	Yes
	8 groups	0.5036	0.1699	2.399	1.533	Yes
Sports Article		0.1062	0.0869	2.078	1.533	Yes
Asian Religious		0.1475	0.1290	2.078	1.533	Yes
Stack Overflow		0.0308	0.0356	1.938	1.533	Yes
Amazon		0.0033	0.0051	1.440	1.533	No

Table 13: The results of the t-test of DPC-K-means on FMI.

Dataset		Mean	Standard deviation	t	p	Difference
BBC		0.7362	0.2657	1.568	1.533	Yes
20-newsgroups	4 groups	0.8126	0.2860	1.327	1.533	No
	5 groups	0.7269	0.2548	1.400	1.533	No
	6 groups	0.6119	0.2290	2.311	1.533	Yes
	7 groups	0.5728	0.2014	2.325	1.533	Yes
	8 groups	0.4926	0.1648	2.184	1.533	Yes
Sports Article		0.5626	0.2326	1.609	1.533	Yes
Asian Religious		0.4051	0.1028	1.632	1.533	Yes
Stack Overflow		0.3876	0.1408	2.598	1.533	Yes
Amazon		0.6089	0.1946	1.268	1.533	No

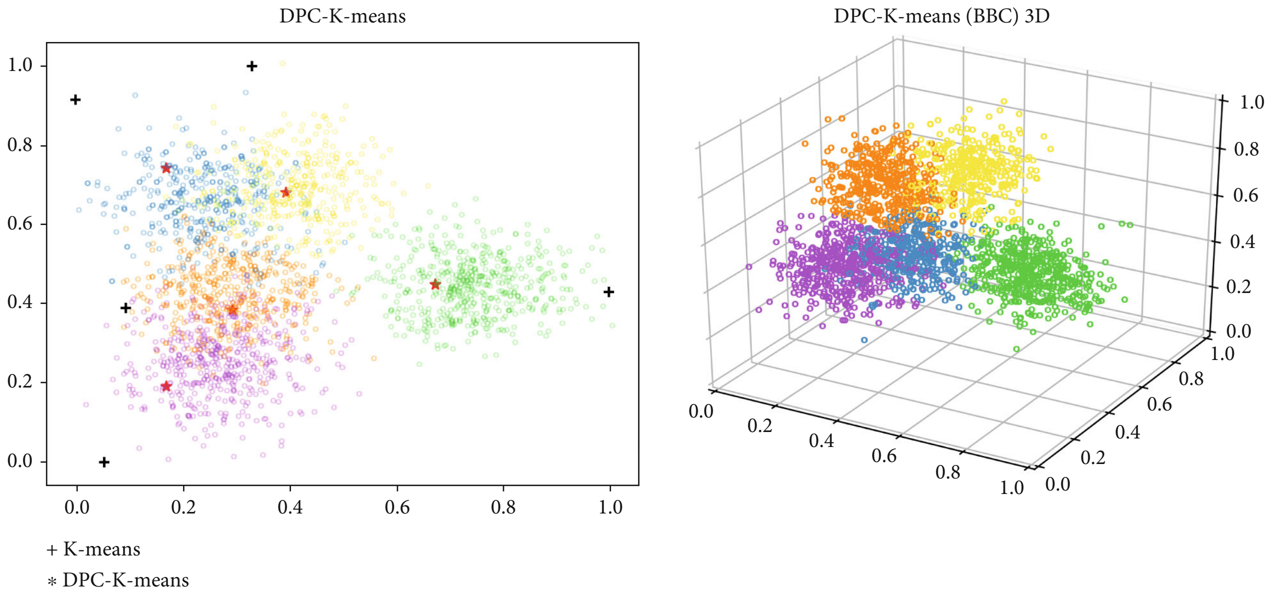


Figure 11: The BBC news dataset clustering center of K-means marked in black, and DPC-K-means clustering center marked in red and the 3D clustering results.

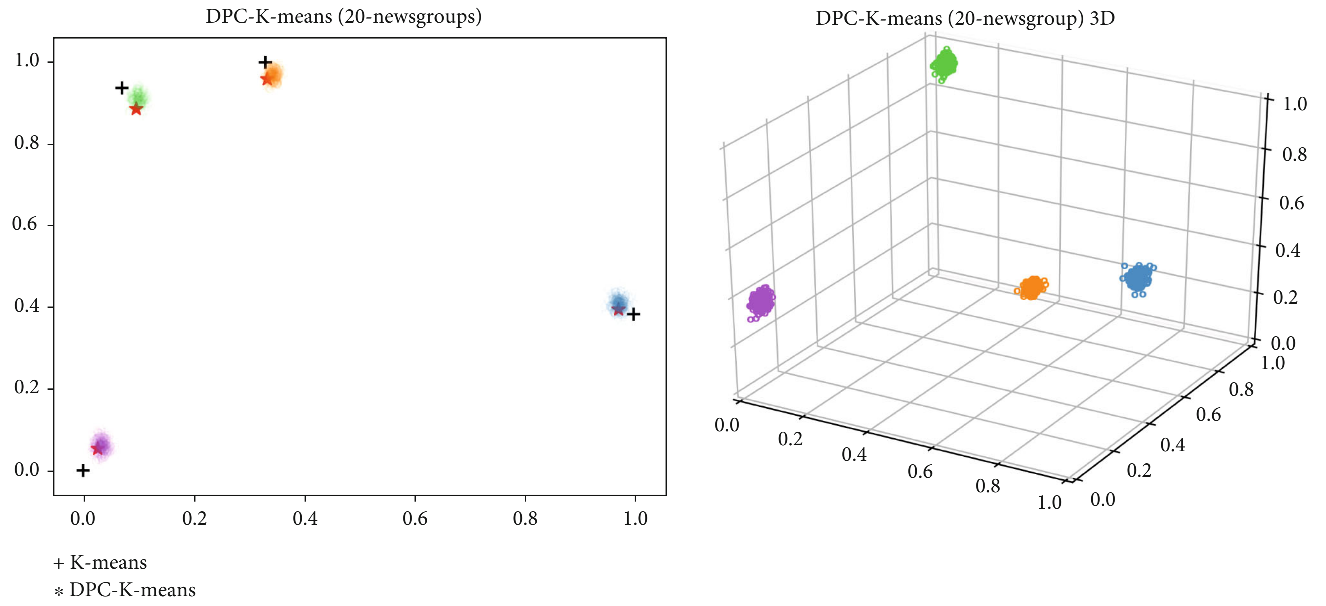


Figure 12: The 4 groups in the 20-newsgroups dataset clustering center of K-means marked in black, and DPC-K-means clustering center marked in red and the 3D clustering results.

the nearest neighbour number of the sample, approximately 1%~2% of the total dataset size. In the experiment, we were able to obtain the correct number of class clusters according to this valued principle. This parameter has no significant influence on the result of the algorithm within the value range of 1%–2% of the entire dataset size. The K-nearest neighbour method was used to establish the similarity matrix in the Spectral Clustering parameters in the experiment. The damping factor in Affinity Propagation was set to 0.9, and the nearest distance measurement of the DBSCAN parameter value was set to "cosine" by cosine similarity.

4. Conclusions

This study proposed a Stacked-Random Projection (SRP) dimension reduction framework based on deep networks and an improved K-means text clustering algorithm based on density peak (DPC-K-means). In the experiment, SRP, the improved DPC, and DPC-K-means were validated by using different datasets. Firstly, we compared SRP with PCA, MDS, and Random Projection. Multiple evaluation metrics demonstrated that SRP maintained a sufficient balance between running time and distance before and after dimension reduction. Secondly, we compared the difference between the Euclidean distance and cosine similarity in calculating DPC local density. Cosine similarity is more suitable for text vector calculation. Finally, DPC-K-means are an improved K-means algorithm that uses a text feature vector's cosine similarity to calculate local density and get the initial clustering center and cluster number. Then, the K-means algorithm is used for clustering. We compared DPC-K-means with DPC, DBSCAN, Spectral Clustering, and Affinity Propagation. We found that DPC-K-means can accurately determine the number of clusters and the initial clustering centers of high-dimensional text data. It is superior to other clustering algorithms in ARI, NMI, FMI, and MSE. Furthermore, we analyzed the influence of parameters on the algorithm and limitations of our proposed methods. We will focus on determining the number of layers and the target dimension of each layer dimensionality reduction for future work and improve the matching degree between DPC-K-means and datasets.

Data Availability

The BBC news data used to support the findings of this study have been deposited in the open-source repository (http://mlg.ucd.ie/datasets/bbc.html). The 20-newsgroups data used to support the findings of this study have been deposited in the open-source repository (http://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets).

Conflicts of Interest

Yujia Sun and Jan Platoš declare that there is no conflict of interest regarding the publication of this paper.

References

[1] C. Aggarwal and C. Zhai, *Mining text data*, Springer, New York, NY, 2012.

[2] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, Norwell, MA, USA, 2002.

[3] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton university press, United States of America, 2015.

[4] X. S. Lu, M. C. Zhou, L. Qi, and H. Liu, "Clustering-algorithm-based rare-event evolution analysis via social media data," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 2, pp. 301–310, 2019.

[5] S. Zhou, X. Xu, Y. Liu, R. Chang, and Y. Xiao, "Text similarity measurement of semantic cognition based on word vector distance decentralization with clustering analysis," *IEEE Access*, vol. 7, pp. 107247–107258, 2019.

[6] A. Onan, "Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering," *IEEE Access*, vol. 7, pp. 145614–145633, 2019.

[7] J. Jokinen, T. Raty, and T. Lintonen, "Clustering structure analysis in time-series data with density-based clusterability measure," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1332–1343, 2019.

[8] X. Xu, J. Li, M. C. Zhou, J. Xu, and J. Cao, "Accelerated two-stage particle swarm optimization for clustering not-well-separated data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4212–4223, 2020.

[9] L. Liu, A. Yang, W. Zhou, X. Zhang, M. Fei, and X. Tu, "Robust dataset classification approach based on neighbor searching and kernel fuzzy c-means," *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 3, pp. 235–247, 2015.

[10] K. Orkphol and W. Yang, "Sentiment analysis on microblogging with K-means clustering and artificial bee colony," *International Journal of Computational Intelligence and Applications*, vol. 18, no. 3, p. 1950017, 2019.

[11] U. H. Atasever, "A novel unsupervised change detection approach based on reconstruction independent component analysis and ABC-Kmeans clustering for environmental monitoring," *Environmental Monitoring and Assessment*, vol. 191, no. 7, 2019.

[12] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[13] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[14] J. Zhang, M. Zhu, P. Chen, and B. Wang, "DrugRPE: random projection ensemble approach to drug-target interaction prediction," *Neurocomputing*, vol. 228, pp. 256–262, 2017.

[15] L. Gondara, "RPC: an efficient classifier ensemble using random projection," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 559–564, Miami, FL, USA, December 2015.

[16] S. Sieranoja and P. Fränti, "Fast and general density peaks clustering," *Pattern Recognition Letters*, vol. 128, pp. 551–558, 2019.

[17] M. Parmar, D. Wang, X. Zhang et al., "REDPC: a residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, 2019.

[18] M. D. Parmar, W. Pang, D. Hao et al., "FREDPC: a feasible residual error-based density peak clustering algorithm with the fragment merging strategy," *IEEE Access*, vol. 7, pp. 89789–89804, 2019.

[19] M. Parmar, D. Wang, A. Tan, C. Miao, J. Jiang, and Y. Zhou, "A novel density peak clustering algorithm based on squared residual error," in *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pp. 43–48, Shenzhen, China, December 2017.

[20] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "A novel cluster validity index based on local cores," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 4, pp. 985–999, 2019.

[21] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and Y. Lijun, "Clustering with local density peaks-based minimum spanning tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, p. 1, 2019.

[22] F. Heimerl, M. John, Q. Han, S. Koch, and T. Ertl, "DocuCompass: effective exploration of document landscapes," in *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 11–20, Baltimore, MD, USA, Oct 2016.

[23] B. Wang, J. Zhang, F. Ding, and Y. Zou, "Multi-document news summarization via paragraph embedding and density peak clustering," in *2017 International Conference on Asian Language Processing (IALP)*, pp. 260–263, Yuexian Zou, Dec 2017.

[24] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[25] P. Krömer and J. Platoš, "Cluster analysis of data with reduced dimensionality: an empirical study," in *Intelligent Systems for Computer Modelling*, pp. 121–132, Springer, Cham, 2016.

[26] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?," *Pattern Recognition*, vol. 93, pp. 95–112, 2019.

[27] T. Sung, L. Kong, P. Tsai, and J. Pan, "A distance coefficient-based algorithm for k-center selection in wireless sensor networks," in *2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 293-294, Taipei, Taiwan, June 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8862353, 9 pages
https://doi.org/10.1155/2020/8862353

Research Article

URDNet: A Unified Regression Network for GGO Detection in Lung CT Images

Weihua Liu, Yuchen Ren, and Huiyu Li

Beijing Lab of Intelligent Information, School of Computer Science, Beijing Institute of Technology, Beijing, China

Correspondence should be addressed to Weihua Liu; liuweihua@bit.edu.cn

Received 30 July 2020; Revised 19 August 2020; Accepted 3 September 2020; Published 17 October 2020

Academic Editor: Chao-Yang Lee

We present a 3D deep neural network known as URDNet for detecting ground-glass opacity (GGO) nodules in 3D CT images. Prior work on GGO detection repurposes classifiers on a large number of windows to perform detection or fine-tuning by box regression based on a previous window classification step. Instead, we consider GGO detection as a multitarget regression problem to focus on the location of GGO. Furthermore, to capture multiscale information, we introduce a backbone network which is a contracting-expanding structure similar to 2D U-net, but we inject the source CT inputs into each layer in the contracting pathway to prevent source information loss at different scales. At last, we propose a two-stage training method for URDNet. In the first stage, the backbone of the network for feature extraction is trained, and in the second, the overall URDNet is fine-tuned based on the previous pretrained weights. By using this training method in conjunction with data augmentation and hard negative mining techniques, our URDNet can be effectively trained even on a small amount of annotated CT images. We evaluate the proposed method on the LIDC-IDRI dataset. It achieves the sensitivity of 90.8% with only 1 false positive per scan. Experimental results show that our detection method achieves the superior detection performance over the state-of-the-art methods. Due to its simplicity and effective, URDNet can be easier to apply to medical IoT systems for improving the efficiency of overall health systems.

1. Introduction

Lung cancer is currently a leading cause of cancer death worldwide and is responsible for more than 1.3 million deaths annually [1]. Detection and treatment of lung cancer at an early stage can improve the survival rate. GGO is a highly important CT imaging sign for detection of lung cancer at an early stage [2], which is defined as increased attenuation of the lung parenchyma without obscuration of the pulmonary vascular markings on the CT images [3]. Recently, the new coronavirus COVID-19 pandemic is prevalent, and its main symptoms are also related to GGO. However, due to their indistinct boundaries and no clear rules for brightness and shape, GGO nodules are easily overlooked, even by experienced radiologists. A promising solution to this problem is the use of computer-aided detection techniques.

The traditional architecture for computer-aided GGO detection typically consists of two stages: GGO candidate detection and false-positive reduction [4]. A small number of papers have been published on this topic. Bastawrous et al. [5] applied a Gabor filter to choose candidates and used an ANN to reduce false positives. Kim et al. [6] extracted tentative regions using binarization and classified the GGO nodules with a linear discriminant function. Jacobs et al. [7] first used intensity, shape, and context features to describe the appearance of candidates and subsequently applied a linear discriminant classifier and a gentle boost classifier to classify candidate regions. Although the conventional methods have yielded promising results, they still suffer from the low sensitivity and poor generalization, especially for notably small GGO nodules.

In recent years, nodule detection based on deep neural networks has achieved state-of-the-art detection performance. For example, Ginneken et al. [8] presented promising results for the extraction of nodule features using an off-the-shelf convolutional neural network (CNN) that was pretrained for a natural image classification task. Setio et al. [9] used multiple CNNs to extract discriminative features from the candidates, and these features were used to classify candidates as nodules or background. Superior performance was achieved in the false-positive reduction track. Roth et al. [10] proposed an effective 2.5D representation for lymph node detection to exploit the 3D information of nodules when training a deep network by taking slices of the CT images from a point of interest in 3 orthogonal views. The slices were subsequently combined into a 3-channel image as the network input. Han et al. [11] proposed hybrid resampling in multi-CNN models for 3D GGO nodules to cover a large range scale, which reduced the risk of missing small or large GGO nodules. In general, these methods rely on classification or a combination of classification and regression for detection. These types of methods usually do not pay enough attention to the location problem of the detection and often produce missed detection and inaccurate locations. In addition, various types of neural works have been applied in various applications, e.g., graph neural network for creative works [12], LVQ neural network for traffic prediction [13], generative adversarial networks (GANs) for style transfer [14], and 3D GANs for the simulation of creative stage scene [15]. The most important is that GGO detection requires a huge amount of computation in 3D CT and generally requires a more efficient detection method to meet actual needs in medical IoT.

In order to overcome the above limitations, we propose a unified regression deep neural network for GGO detection. We consider GGO detection as a multitarget regression problem, straight from 3D CT to bounding box coordinates. To acquire the discrimination information between the object and the background, the same pseudotarget (zero) is set for all the negative samples, so the pseudotarget also is denoted as zero target in our paper. Compared with the classification-based method, the learning goal of our whole detection is just a unified object location regression, which can guide the network to learn better object location information. Therefore, more attention can be paid to the localization problem in the detection by using a unified regression objective function in our approach. And a multi-input and multioutput backbone convolutional network is also applied in our approach to make it more representative. Furthermore, we design a two-stage transfer training method to train our URDNet on small annotated GGO data. To evaluate the effectiveness of our proposed URDNet for GGO detection, we conduct GGO detection experiments on the LIDC-IDRI [16], the currently largest publicly available and mostly often used database of lung nodules. The experiment results show that the network is effective and accurate.

Our main contributions are summarized as follows:

(1) We present an end-to-end deep convolutional neural network which is unified and only regression for GGO detection in 3D CT scans which leads to outstanding performance of GGO detection on LIDC-IDRI. The resultant detection sensitivity is 90.8% at 1 false positive per scan

(2) We introduce a multi-input and multioutput structure for our network's backbone. The backbone not only reserves the subtle locations but also represents the discriminate information of GGO nodules

(3) We propose a unified regression objective function for all samples. For positive samples, the position prediction is regarded as only a conventional regression task. For negative samples, zero target is set. The location of negative samples (boxes) will be regressed to a pseudotarget (zero).

(4) We adopt a two-stage training method to train the complicated 3D detection network given a small amount of annotated samples

The remainder of this paper is organized as follows. Sections 2 and 3 presents our URDNet architecture and its training method, respectively. The implementation details and experimental results are discussed in Section 4. We conclude in Section 5.

2. URDNet Architecture

The network architecture is illustrated in Figure 1 and is composed of a backbone network for feature extraction, which is a multiscale input-output structure, and a detection head which is a single prediction module by location regression, which directly generates a fixed set of 3D bounding boxes. Due to the memory limitation of GPU, the input of the method is only a CT cube with a fixed size ($128 \times 128 \times 128$ in our experiments). The final detection result is the combination of all the detected GGOs in each cube. Figure 1 illustrates the architecture of our network, which takes as input a $128 \times 128 \times 128$ CT cube. The details of CT preprocessing are introduced here. For a 3D CT scan, a voxel size normalization is firstly performed due to various voxel sizes of subjects. The voxel size is set to $1 \times 1 \times 1\,\mathrm{mm}^3$ by using bilinear interpolation (after voxel normalization, even the number of slices in axial dimension is greater than 128 for all the data in our experiments). Then, we divide a 3D CT volume into several $128 \times 128 \times 128$ cubes in a slider-patch way. For each cube, it will be inputted into our URDNet and be processed. More details of URDNet are given in the following subsections.

2.1. Backbone Network. The backbone network includes two main symmetric pathways: a contracting pathway and an expanding pathway. The contracting pathway follows the typical architecture of a convolutional network. To aid the network in capturing spatial information at different scales, we construct a multiscale input structure by downsampling the source input and feeding it into each layer in the contracting pathway, not only the first layer. The feed operation is indicated by the green lines in Figure 1. We referred to these lines connecting the source input CT to the layers in the contracting pathway as source connections. The contracting pathway can be divided into five blocks, in which the output feature maps are downsampled by 2, 4, 8, 16, and 32 w.r.t. the input cube size. This block is a composite function of four consecutive operations: 3D convolution (Conv), batch normalization (BN) [17], rectified linear units (ReLU), and pooling (Avg: average or Max: maximum).

The expanding pathway is an information-expanding process that elucidates higher resolution features via an upsampling strategy. The upsampling procedure is

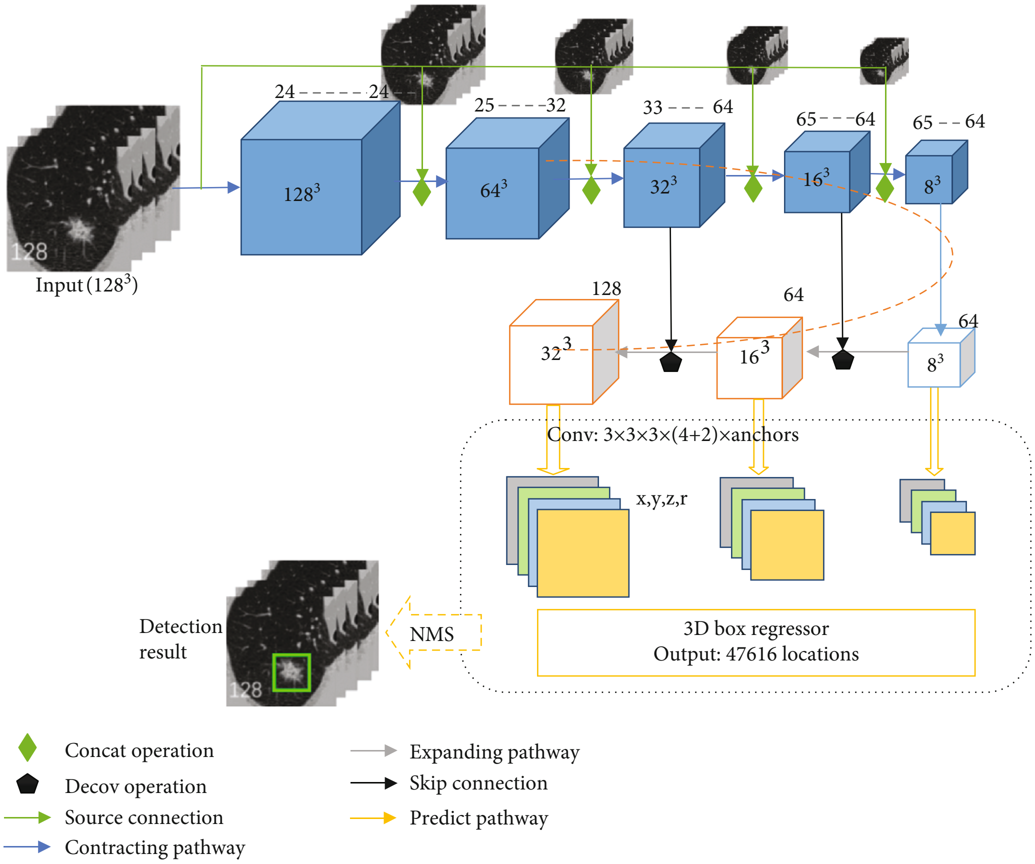


Figure 1: The URDNet architecture, which mainly comprises two components, the backbone network for feature extraction and the detection head. The backbone network is a multi-input and multioutput structure. The detection head merely contains a $3 \times 3 \times 3$ convolutional layer, followed by only one prediction layer, for regression the center location (x, y, z) and the corresponding diameter r of the nodule. (the digits on the cube denote the corresponding numbers of channels.)

implemented by a series of layers including unpooling, deconvolution, BN, and ReLU operations to perform a complicated deconvolution, as described in a previous paper [18]. The expanding pathway is semantically stronger because of the feature map from higher levels (the top of the contracting pathway). In contrast, the contracting feature map consists of lower-level semantics, but its activations are more accurately localized because it was subsampled fewer times. To preserve the localized information, the feature map of the expanding pathway is enhanced with the feature map from the contracting pathway via skip connections. Skip connections associate low-level feature maps across resolutions and semantic levels. Moreover, to create a multiscale feature map that has strong semantics and precise spatial information at all scales, we combine low-resolution and semantically strong features with high-resolution and semantically weak features via a top-down pathway and skip connections. Skip connections [19] are connections that can skip one or more layers. Similar architectures adopting a top-down pathway and skip connections are popular in recent research [20–22]. However, only a single high-level feature map of fine resolution was applied for prediction in previous networks. In contrast, our backbone leverages multiscale feature maps in which predictions are independently generated on each level for GGO detection.

2.2. Detection Head. In the traditional sliding-window detection methods, the entire detection space is eventually discretized into a series of windows. Our network also discretizes the output space of bounding boxes into a set of anchor boxes with different scales over multiple feature maps. Each anchor box is a predefined box centered at a location of the feature map and is associated with a special initial scale, similar to the anchor box used in Faster R-CNN [23].

Our regression prediction head predicts bounding boxes based on a fixed set of anchor boxes and is implemented by regressing 3D box relative offsets from anchor boxes to satisfiable boxes (to better match the GGO shape) using small convolutional filters applied to multiple feature maps. These processes are indicated on the yellow lines and a yellow rectangular box in the bottom area of Figure 1. According to the previous section, the feature map of the expanding pathway progressively increases in size. The multiscale feature maps are composed of multiple feature maps of different resolutions, and each feature map can produce a set of detection predictions. To detect GGO of various sizes, the prediction pathway of our network can naturally combine predictions from the multiscale feature maps. Additionally, at each location of the feature map, we simultaneously predict multiple boxes with different scales but the same center. The multiple boxes are parameterized relative to the corresponding anchor boxes. For example, the left block shown in Figure 2 is used to regress to the resultant boxes. Since we consider 3 sizes of anchors and each resultant result is a 4D vector, the outcome of this block is $12 = 3 \times 4$ vectors for each anchor. The number of anchors for each feature map location must be carefully set to cover a wider and finer range of scale, and more details are provided in Section 4.1.

After the resultant boxes are obtained from the prediction pathway, we perform nonmaximum suppression (NMS) to rule out the overlapping boxes. For each anchor, the prediction finally produces the four-position component map. If the area is background, the corresponding location of the component map is very close to zero. Only the box with non-background will be retained and others will be deleted. Then, the retained boxes will be decided as GGOs if their position is larger than a threshold or otherwise as non-GGO (background). Such threshold will be set up in the applications.

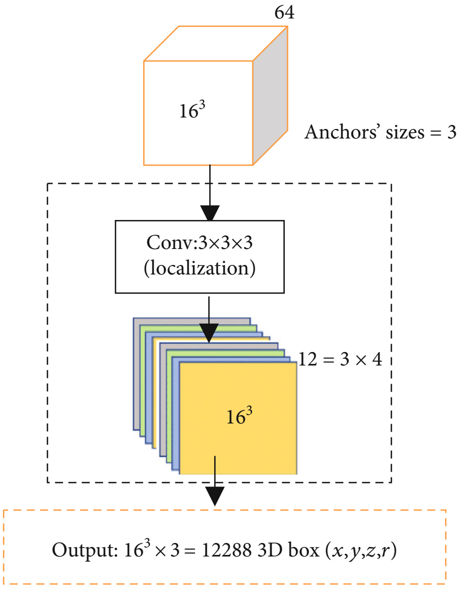


Figure 2: The illustration of the regression in a prediction pathway for one scale of feature map, which is the detail of "3D box regressor" shown in Figure 1.

3. Training Method

Training of URDNet is a multitarget regression procedure because it simultaneously regresses the GGO center and the diameter of GGO. The details of the objective of our network is given in the following subsection. Besides, only a small amount of CT data with annotated GGO nodules is given, and the overall network is difficult to converge. We adopt a two-stage transfer training strategy to solve the problem. More useful strategies also are given in the below subsections.

3.1. Loss Function.

3.1. Loss Function. In our method, a true object can be expressed by a box. Each box corresponds to a 3D square and thus is represented by a 4D vector (x, y, z, r), where (x, y, z) is the 3D center point and r is the side length of the square. We also use a box to express the position of any background, and the components of this box (x, y, z, r) are set to zero. We define the position of this background as zero target (pseudo-target). The target position of true objects is defined as real target. In this way, both positive samples (target window) and negative samples (background window) can be used as position targets. We can express the detection problem as the same learning target and only need position regression.

The conventional regression loss for positive samples and a new design regression loss for negative samples are included in the overall localization loss.

The regression loss for positives is a modified smooth L_1 loss [24] between the predicted 3D bounding box (denoted as l) and the ground-truth 3D bounding box (denoted as g). Similar to Faster R-CNN [12], it can be regressed to the offset terms for the center (xc, yc, and zc) of the anchor 3D bounding box (denoted as d) and its radius (denoted as r).

$$L_{\text{pos}}(l, g) = \sum \left(\beta \sum_{k \in \{xc,yc,zc\}} \text{smooth}_{L_1} \left(l_i^k - \hat{g}_j^k \right) + (1 - \beta) \text{smooth}_{L_1} \left(l_i^r - \hat{g}_j^r \right) \right), \quad (1)$$

where $\widehat{g_j^{xc}} = (g_j^{xc} - d_i^{xc})/d_i^r$, $\widehat{g_j^{yc}} = (g_j^{yc} - d_i^{yc})/d_i^r$, $\widehat{g_j^{zc}} = (g_j^{zc} - d_i^{zc})/d_i^r$, $\widehat{g_j^r} = \log (g_j^r/d_i^r)$, and the weight term β is set to 0.6 through careful experiments in this paper, which means that we focus additional attention on the center point.

At the same time, because of the characteristics of zero target in our negative samples, we design a loss function which regresses to a zero target (denoted as o),

$$L_{\text{neg}}(l, o) = \sum \log \left(\sum_{k \in \{xc,yc,zc,r\}} \| l_k - o \| \right), \quad (2)$$

which means that several position components can be cohered to zero (zero target).

To sum up, the full optimization objective is

$$L_{\text{all}} = L_{\text{pos}} + \lambda_{\text{n}} L_{\text{neg}}, \quad (3)$$

where λ_{n} is the weight for the regression loss for the negative samples and is set to 0.5 in this paper.

3.2. Two-Stage Training. We add a classifier module with a 3D Avg-pooling $(4 \times 4 \times 4)$ layer and a two-class softmax layer behind the backbone of our network to construct a solo GGO classifier. The $64 \times 64 \times 64$ positive cubes are cropped from the lung scans such that they contain only one GGO nodule. More positive cubes are generated by data augmentation. The $64 \times 64 \times 64$ negative cubes without nodules are randomly cropped. It is easier to construct a relatively large-scale dataset for training the nodule classifier than the detector network. Moreover, the solo classifier can be trained much faster than our network because the input size is 1/8 of the URDNet input size, and it is only a binary classification problem. To avoid slow convergence, the weights are initialized with Glorot and Bengio [25] initialization. The classifier network is trained by minimizing the softmax-loss function with the stochastic gradient descent (SGD).

After training the classifier network, we truncate the weights of the classifier network to initialize our backbone and train the URDNet by fine-tuning with a smaller learning rate. We find that the URDNet loss converges quicker than training from scratch. Training our network in this way can effectively alleviate the overfitting problem.

3.3. Data Augmentation. Considering that only a few training samples are available, we tested certain data augmentation operations. We exploit both lossless and lossy augmentation and find the following two simple but useful augmentations for GGO nodule patches:

(1) Flipping. The patches are randomly flipped with respect to coronal, sagittal, and axial dimensions

(2) Resizing. The patches are randomly resized with a ratio between 0.9 and 1.1

We also tested other augmentation methods, such as axis swapping and rotation. However, it has no significant effects. Furthermore, to boost nodule classification performance, we adopt an online hard negative mining technique, similar to the online hard example mining (OHEM) method proposed in a previous paper [26]. We use online hard negative mining in the training procedure. First, we process the input patches using the network to conduct a forward pass and obtain tens of thousands of detections, each of which is associated with a box with a classification confidence score. Second, the acceptable N negative samples are randomly selected from these boxes as a candidate pool. Third, the negative samples in this pool are sorted in a descending order based on their overall loss, and the top K samples are selected as the hard negatives.

4. Experiments

In this section, we first describe the experimental setup. We introduce our dataset's details. The use of anchor boxes improves the speed and efficiency for the detection portion of a deep learning neural network. Due to its importance, we then give more details about anchors. At last, we give the results of our two groups of experiments. One group is used to evaluate our URDNet, and the other group is used to compare the performance with that of other systems.

4.1. Experimental Setup

4.1.1. Dataset. The dataset is collected from the largest publicly available reference database for lung nodules, the LIDC-IDRI [16]. The LIDC-IDRI database contains a total of 1018 CT scans. Each LIDC-IDRI scan was annotated by experienced thoracic radiologists using a two-phase reading process. Four radiologists annotated scans and marked all suspicious lesions as nodule ≥ 3 mm, nodule < 3 mm, or nonnodule. We only considered the GGO nodules. Simultaneously, we only considered the nodules annotated by the majority of the radiologists (at least 3 out of 4 radiologists) in our standard reference. 302 CT scans from 299 patients were collected from LIDC-IDRI. In these CT scans, 635 GGO nodules (271 nodules annotated by 3 radiologists and 364 nodules annotated by 4 radiologists) were found. The diameter of the nodules varied from 3 to 34 mm with a median of 10.3 mm. We use 250 CT scans as a training set and the remaining 52 as a test set. A slice from one patient with a single nodule location is shown in Figure 3.

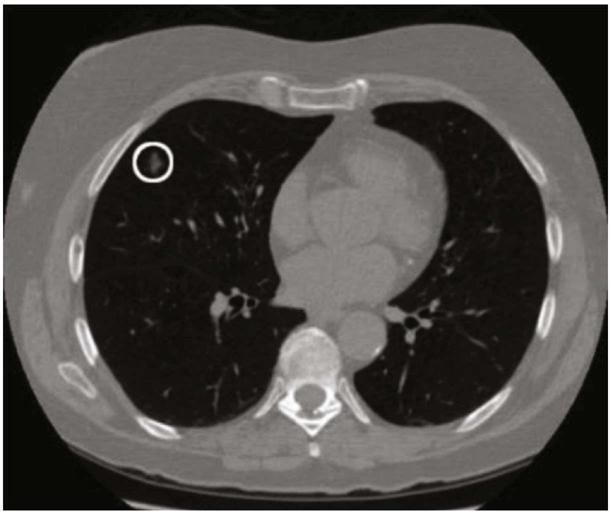


Figure 3: A slice from LIDC-IDRI with a nodule annotated.

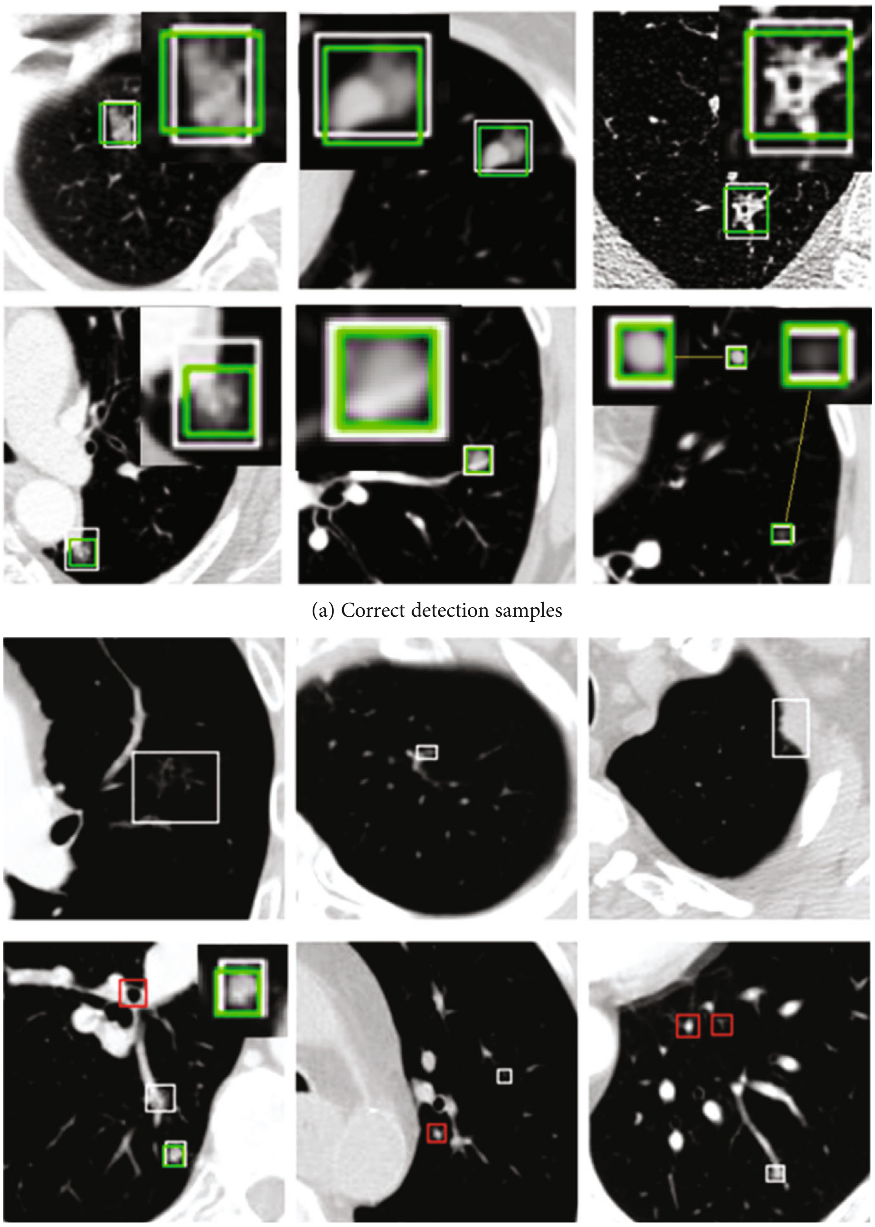
Table 1: Configuration of anchors' sizes for each scale of decoded feature maps.

Feature map	No. of anchor size	No. of anchor box	Size (mm)				
32 × 32 × 32	1	32768	4	\	\	\	\
16 × 16 × 16	3	12288	6	8	10	\	\
8 × 8 × 8	5	2560	12	16	20	26	32
Sum		47616					

Table 2: GGO detection results using SSD, U-net, and our URDNet.

Methods	Sensitivity	FPs/scan
SSD	78.2%	9.1
U-net	83.7%	8.9
Our URDNet	93.5%	6.8

4.1.2. Anchor Configuration. All the scales of information decoded in the expanding pathway of our URDNet, including 32 × 32 × 32, 16 × 16 × 16, and 8 × 8 × 8, are used and inputted into the detection head. The numbers of anchors' sizes are set to be 1, 3, and 5 for these three scales, respectively, which lead to 9 anchors' sizes in total: 4, 6, 8, 10, 12, 16, 20, 26, and 32 mm. The details are listed in Table 1. To sum up, we will generate 47616 $(= 32^3 \times 1 + 16^3 \times 3 + 8^3 \times 5)$ anchors for each CT cube, based on which 47616 resultant boxes will be computed. We find that the detection scale range is between 3 mm and 33 mm, which is larger than the scale range of the anchor box, in our test set. Benefitting from the



(a) Correct detection samples

(b) False detection samples

Figure 4: Examples of the detection by URDNet (the white rectangles denote the ground-truth boxes, the green rectangles denote the detection results, and they are zoomed at the top-right area or the left-bottom area, and the red rectangles denote the wrong results).

multiscale processing, our network can adapt to the changing of nodules' sizes.

4.2. Experimental Results. In our experiments, we use a free-response receiver operating characteristic (FROC) analysis [27] on the filtered GGO dataset from LIDC-IDRI [16] for comparison. In the FROC curve, the sensitivity is plotted as a function of the average number of false positives per scan (FPs/scan). In this work, the sensitivity is defined as the fraction of detected true positives divided by the number of ground-truth GGO nodules. The FROC overall score is defined as the average of the sensitivity at seven predefined false-positive rates: 1/8, 1/4, 1/2, 1, 2, 4, and 8 FPs per scan. This performance metric was introduced into the ANODE09 challenge and referred to as the competition performance metric (CPM) in a previous paper [28].

We first conduct three experiments using a different backbone or head detection to evaluate the effect of our network. We use the same training dataset and data augmentation strategy. Other hyperparameters of the training network are also shared, except for specified changes to components. The SSD method predicts bounding boxes and confidence scores based on a fixed set of anchor boxes, which directly related to our URDNet's head detection. In contrast, U-net has an elegant decoder-encoder structure, but it only uses the last feature map for biomedical applications and ignores the multi-input and multioutput structure. Table 2 lists the results in the GGO detection task.

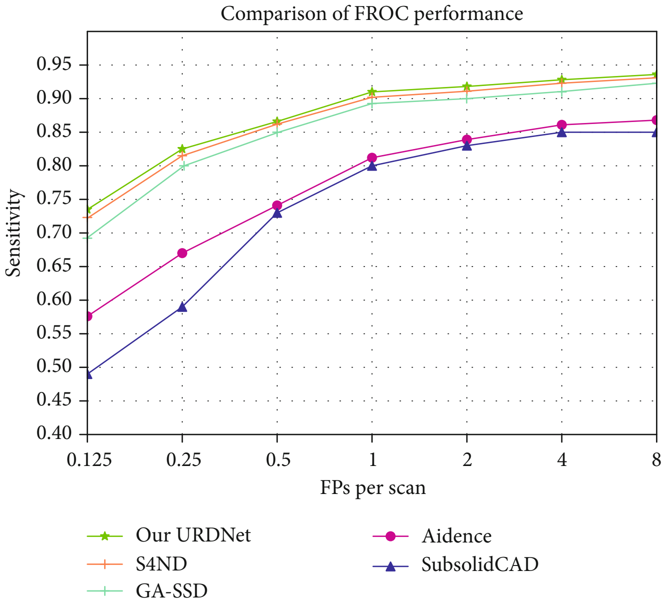


Figure 5: Comparison of performance (our URDNet with four counterparts).

It is obvious that our URDNet achieves the highest sensitivity (93.5%) at the lowest FPs/scan (6.8) among these detection experiments, which demonstrates the superiority of our detection network. Certain examples that are correctly detected are illustrated in Figure 4(a). These results indicate that URDNet can accurately locate the centers of GGO nodules and regress the size of GGO nodules. Examples of GGO nodules false detections are also shown in Figure 4(b). Typically, these nodules are notably low in contrast (pure or diffuse) or located close to the other tissues (blood or chest wall) and can be considered notably low-quality GGO nodules. To further improve the detection performance, additional discriminative features must be learned by a new learning method.

Although a few methods have been developed for GGO nodule detection, it is trivial to compare all other methods. In this paper, we choose three GGO detection systems or methods from different categories for comparison. We first select the SubsolidCAD [29] system, which is a state-of-the-art conventional method that uses 4 categories of handcrafted features to describe the appearance or the internal characteristic of the GGO. The system can reach a sensitivity of 80% at an average of 1.0 false positives per scan with a CPM [28] of 0.734. We also compare our performance with the Aidence [30] system based on convolutional networks, which is the strongest competitor and the top performer in the LUNA16 Challenge. Referring to the report [30], the best score was achieved by Aidence with a CPM of 0.764 in the GGO nodule candidate detection task. The Aidence system uses end-to-end convolutional networks that are trained on a subset of studies from the National Lung Screening Trial [31]. Last, we chose the S4ND [32] method and GA-SSD [33] method for lung nodule detection to compare which are current state-of-the-art methods. The S4ND method is a single-shot and single-scale method, while GA-SSD is an improved method based on SSD by implementing the attention mechanism. Additionally, we list a comparison of performance among our URDNet, SubsolidCAD, Aidence, S4ND, and GA-SSD in Figure 5. We observe that our URDNet attained superior performance. The CPM can reach 0.874, surpassing the SubsolidCAD system (CPM: 0.734) with relative performance gains of 19.07%, the Aidence system (CPM: 0.767) with 13.95%, S4ND (CPM: 0.866) with 0.92%, and GA-SSD (CPM: 0.855) with 2.26%, respectively.

5. Conclusions

In this paper, we present a 3D convolutional detector network known as URDNet that was constructed of a multiscale input-output U-shaped network for GGO detection in CT images. A unified regression objective function is proposed in URDNet in which the location of an object is focused on during learning that can directly regress a fixed set of 3D boxes for all samples. Furthermore, a two-stage training method is designed to help our complicated 3D detector network converge and prevent overfitting, even if given a small amount of annotated GGO nodule CT images. By this training method incorporated with data augmentations and hard negative mining, our network can be efficiently and effectively trained in an end-to-end manner for GGO detection.

We believe that URDNet offers a useful tool for GGO nodule detection in the clinical diagnosis of lung cancer. Moreover, our independent GGO detection algorithm can be easily integrated into the existing lung nodule CAD systems to boost the overall system performance. Immediate future work will extend our network for the detection of other nodules. Because this method does not require large amounts of labeled data and has a simple, unified regression objective, it is easier to be applied to other nodule processing tasks in medical IoT systems.

Data Availability

The data used to support the findings of this study are from previously reported LIDC-IDRI dataset, which is the currently largest publicly available database. The data are available at relevant places with text as reference.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant numbers 60973059, 81171407, and 61901533) and Program for New Century Excellent Talents in University of China (grant number NCET-10-0044).

References

[1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2015," *CA: A Cancer Journal for Clinicians*, vol. 65, no. 1, pp. 5–29, 2015.

[2] C. I. Henschke, D. F. Yankelevitz, R. Mirtcheva, G. McGuinness, D. McCauley, and O. S. Miettinen, "CT screening for lung Cancer," *AJR American Journal of Roentgenology*, vol. 178, no. 5, pp. 1053–1057, 2002.

[3] W. T. Miller Jr. and R. M. Shah, "Isolated diffuse ground-glass opacity in thoracic CT: causes and clinical presentations," *AJR American Journal of Roentgenology*, vol. 184, no. 2, pp. 613–622, 2005.

[4] L. Linying, L. Xiabi, Z. Chunwu, Z. Xinming, and Z. Yanfeng, "A review of ground glass opacity detection methods in lung CT images," *Current Medical Imaging Reviews*, vol. 13, no. 1, pp. 20–31, 2017.

[5] H. A. Bastawrous, T. Fukumoto, M. Tsudagawa, and N. Nitta, "Detection of ground glass opacities in lung CT images using Gabor filters and neural networks," in *2005 IEEE Instrumentationand Measurement Technology Conference Proceedings*, pp. 251–256, Ottawa, Ont., Canada, May 2005.

[6] H. Kim, T. Nakashima, Y. Itai, S. Maeda, J. K. Tan, and S. Ishikawa, "Automatic detection of ground glass opacity from the thoracic MDCT images by using density features," in *2007 International Conference on Control, Automation and Systems*, pp. 1274–1277, Seoul, South Korea, October 2007.

[7] C. Jacobs, C. I. Sánchez, S. C. Saur, T. Twellmann, P. A. de Jong, and B. van Ginneken, "Computer-aided detection of ground glass nodules in thoracic CT images using shape, intensity and context features," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011*, pp. 207–214, Springer, 2011.

[8] B. Van Ginneken, A. A. Setio, C. Jacobs, and F. Ciompi, "Off-the-shelf convolutional neural network features for pulmonary nodule detection in computed tomography scans," in *2015 IEEE 12th International symposium on biomedical imaging (ISBI)*, pp. 286–289, New York, NY, USA, April 2015.

[9] A. A. A. Setio, F. Ciompi, G. Litjens et al., "Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.

[10] H. R. Roth, L. Lu, A. Seff et al., "A New 2.5D Representation for Lymph Node Detection Using Random Sets of Deep Convolutional Neural Network Observations," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pp. 520–527, Springer, 2014.

[11] G. Han, X. Liu, G. Zheng, M. Wang, and S. Huang, "Automatic recognition of 3D GGO CT imaging signs through the fusion of hybrid resampling and layer-wise fine-tuning CNNs," *Medical & Biological Engineering & Computing*, vol. 56, no. 12, pp. 2201–2212, 2018.

[12] F. Zhang, Y. Wang, and C. Wu, "An automatic generation method of cross-modal fuzzy creativity," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 5685–5696, 2020.

[13] F. Zhang, T. Y. Wu, Y. Wang et al., "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555–104564, 2020.

[14] F. Zhang and C. Wang, "MSGAN: generative adversarial networks for image seasonal style transfer," *IEEE Access*, vol. 8, pp. 104830–104840, 2020.

[15] F. Zhang, G. Ding, Q. Lin, L. Xu, Z. Li, and L. Li, "Research of simulation of creative stage scene based on the 3DGans technology," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 6, pp. 1430–1443, 2018.

[16] S. G. Armato III, G. McLennan, L. Bidaut et al., "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): a completed reference database of lung nodules on CT scans," *Medical Physics*, vol. 38, no. 2, pp. 915–931, 2011.

[17] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2015, https://arxiv.org/abs/1511.00561.

[19] C. Bishop and C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Munich, Germany, 2015.

[21] S. Honari, J. Yosinski, P. Vincent, and C. Pal, "Recombinator networks: learning coarse-to-fine feature aggregation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5743–5752, Las Vegas, 2016.

[22] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision – ECCV 2016*, pp. 483–499, 2016.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[24] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, 2015.

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.

[26] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in

Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 761–769, Las Vegas, 2016.

[27] H. L. Kundel, K. S. Berbaum, D. D. Dorfman, D. Gur, C. E. Metz, and R. G. Swensson, "Receiver operating characteristic analysis in medical imaging," *Journal of the Icru*, vol. 79, no. 8, p. 1, 2008.

[28] M. Niemeijer, M. Loog, M. D. Abràmoff, M. A. Viergever, M. Prokop, and B. van Ginneken, "On combining computer-aided detection systems," *IEEE Transactions on Medical Imaging*, vol. 30, no. 2, pp. 215–223, 2011.

[29] C. Jacobs, E. M. van Rikxoort, T. Twellmann et al., "Automatic detection of subsolid pulmonary nodules in thoracic computed tomography images," *Medical Image Analysis*, vol. 18, no. 2, pp. 374–384, 2014.

[30] A. A. A. Setio, A. Traverso, T. de Bel et al., "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge," *Medical Image Analysis*, vol. 42, pp. 1–13, 2017.

[31] C. S. White, "National lung screening trial," *Journal of Thoracic Imaging*, vol. 26, no. 2, pp. 86-87, 2011.

[32] N. Khosravan and U. Bagci, "S4ND: single-shot single-scale lung nodule detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 794–802, Granada, Spain, 2018.

[33] J. Ma, X. Li, H. Li et al., "Group-attention single-shot detector (GA-SSD): finding pulmonary nodules in large-scale CT images," 2018, https://arxiv.org/abs/1812.07166.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8822651, 14 pages
https://doi.org/10.1155/2020/8822651

Research Article

A Node Location Method in Wireless Sensor Networks Based on a Hybrid Optimization Algorithm

Jeng-Shyang Pan,[1,2] Fang Fan,[1,3] Shu-Chuan Chu,[1] Zhigang Du,[1] and Huiqi Zhao[3]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590 Shandong, China*
[2]*Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan*
[3]*College of Intelligent Equipment, Shandong University of Science and Technology, Taian, 271019 Shandong, China*

Correspondence should be addressed to Shu-Chuan Chu; scchu0803@gmail.com

Received 14 July 2020; Revised 20 August 2020; Accepted 6 September 2020; Published 6 October 2020

Academic Editor: Chao-Yang Lee

Wireless sensor networks (WSN) have gradually integrated into the concept of the Internet of Things (IoT) and become one of the key technologies. This paper studies the optimization algorithm in the field of artificial intelligence (AI) and effectively solves the problem of node location in WSN. Specifically, we propose a hybrid algorithm WOA-QT based on the whale optimization (WOA) and the quasi-affine transformation evolutionary (QUATRE) algorithm. It skillfully combines the strengths of the two algorithms, not only retaining the WOA's distinctive framework advantages but also having QUATRE's excellent coevolution ability. In order to further save optimization time, an auxiliary strategy for dynamically shrinking the search space (DSS) is introduced in the algorithm. To ensure the fairness of the evaluation, this paper selects 30 different types of benchmark functions and conducts experiments from multiple angles. The experiment results demonstrate that the optimization quality and efficiency of WOA-QT are very prominent. We use the proposed algorithm to optimize the weighted centroid location (WCL) algorithm based on received signal strength indication (RSSI) and obtain satisfactory positioning accuracy. This reflects the high value of the algorithm in practical applications.

1. Introduction

With the development of the Internet of Things (IoT) to this day, it can be said that the basic task originally conceived has been completed, namely, the interconnection of human, machine, and things [1]. However, people are no longer satisfied with the simple networking and put forward higher requirements. As the rapid expansion of scale, IoT has produced an alarming amount of real-time monitoring data [2]. There is no doubt that scientific management of these data requires a “Super Brain”, and intelligence has become an inevitable choice for the evolution and upgrade. Artificial intelligence (AI) is fully qualified for this role. The history of AI is much longer than that of IoT, but it has been plagued by the difficulty of technical development and the lack of kinetic energy for achievement transformation. In recent years, AI has made breakthrough progress at the technical level, which makes people full of confidence and hope for its future. How to guide AI from the laboratory into various real application scenarios is a challenging task. The booming IoT has provided a broad stage for AI to fully release its potentia, and the naturally existing data sources continuously provide power for it. Relying on the mature platform built by IoT, AI can meet the most urgent demand for technology landing. Although IoT already has a considerable volume in terms of industrial chain and scale, many key technologies still need to be polished and improved. IoT has been thinking for a long time to seek considerable progress and sustainable development, and the empowerment of AI just opens an important window of opportunity for its development. Therefore, the term AIoT (Artificial Intelligence of Things) came into being, which can be understood as the integration of AI and IoT.

AI covers a wide range of technologies, in which the optimization algorithm has received extensive attention from researchers. Among many optimization algorithms, the metaheuristic algorithm is the most active. Some of them are inspired by the evolution phenomenon in nature, represented by genetic algorithm (GA) [3] and differential evolution algorithm (DE) [4]. Some algorithms are inspired by biological collective behavior, including particle swarm optimization (PSO) [5], artificial bee colony algorithm (ABC) [6, 7], Cat swarm optimization (CSO) [8, 9], WOA [10, 11], and symbiotic organism search algorithm [12, 13]. Some are related to physical or mathematical rules, such as black hole (BH) [14, 15] and QUATRE [16, 17]. Unfortunately, there is no one algorithm capable of solving all optimization problems according to the NFL theorem [18]. This has stimulated researchers' endless enthusiasm, either proposing new algorithms and technologies, or improving existing ones [19, 20], or combining different ones [21]. It is found that the hybrid strategy of the algorithm is worth trying in many cases. This article mixes two algorithms, WOA and QUATRE, and introduces the technology of dynamically shrinking the search space. In the test of the benchmark function, it is found that the hybrid algorithm has made great progress.

As the key core technology of IoT, WSN has been extended from military applications originally to a large number of civil fields, such as environmental and ecological protection, medical and health monitoring, agricultural planting, traffic management, and logistics management [22]. The main research contents of WSN include network topology control and network protocol, node deployment [23, 24], network security [25, 26], location technology, and data fusion and data management. The development of WSN also faces many challenges. For example, how to use the limited communication ability to complete the transmission of sensing data needs further research. And how to maximize the service life of the network through low-power design is also an urgent problem to be solved. This paper introduces the optimization algorithm into the node location problem of WSN and achieves satisfactory location accuracy.

The rest of this article is as follows: the second part is the related research work, mainly for the brief introduction of WOA and QUATRE; the third part discusses the combination and improvement strategy of the two algorithms in detail; the fourth part is the statistics and analysis of experimental data based on benchmark functions; the fifth part is to use the algorithm to realize the location in WSN; and the last part is the summary and outlook.

2. Related Works

2.1. Whale Optimization Algorithm (WOA). As a metaheuristic optimization algorithm based on swarm intelligence, WOA is exquisitely designed and full of characteristics. It simulates the hunting behavior of humpback whale groups. Inspired by this hunting method, WOA designed three models: searching for prey, encircling prey, and bubble-net attacking. First, initialize the position of each whale: $\vec{X} = (x_1, x_2, \cdots, x_n)$, which represents an n-dimensional candidate solution. Through the cooperation of two stages of exploration and exploitation, the best position is gradually found (i.e., the optimal solution).

Three vectors appear in the algorithm: $\vec{A}$, $\vec{C}$ and $\vec{D}$. $\vec{A}$ and $\vec{C}$ are defined as follows:

$$\vec{A} = \vec{a}\bullet\left(2\vec{r} - 1\right), \quad \vec{C} = 2\vec{r} \tag{1}$$

where $\vec{r}$ is a random vector between $[0, 1]$. As the iteration progresses, $\vec{a}$ decreases linearly from 2 to 0. Therefore, $\vec{A} \in [-2, 2]$, $\vec{C} \in [0, 2]$. There is a random number prob $\in [0, 1]$, which is used to indicate the selection probability of the updated mode. The random vector $\vec{A}$ and the random number prob jointly control the conversion between exploration and exploitation. Vector $\vec{D}$ has different representations at different stages. The operation symbol $(\bullet)$ means multiplying item by item, and $|\bullet|$ means taking the absolute value.

(1) Exploration: searching mode

WOA's exploration phase mimics the process of whales searching for prey in the ocean. Use a randomly selected search agent to update the solution. To achieve the goal of global search, the search agent is forced to leave the reference whale. The position update formula for exploration is as follows:

$$\begin{cases} \vec{D} = \left|\vec{C}\bullet\vec{X}_{\mathrm{rand}} - \vec{X}(t)\right|, \\ \vec{X}(t+1) = \left|\vec{X}_{\mathrm{rand}} - \vec{A}\bullet\vec{D}\right|. \end{cases} \tag{2}$$

$\vec{X}(t)$ is the solution vector at the $t-$th iteration. $\vec{X}_{\mathrm{rand}}$ represents a random solution in the current population. When prob < 0.5 and $|\vec{A}| \geq 1$, Equation (2) will be used to update the solution vector.

(2) Exploitation

The exploitation phase of WOA includes two modes: encircling mode and bubble-net attacking mode.

2.1.1. Encircling Mode. Once humpback whales find their prey, they surround them. Since the best position is unknown, it is assumed that the current best solution is the destination or the approximate best position. Other search agents will update their position with reference to the current optimal, as shown in Eq.(3):

$$\begin{cases} \vec{D} = \left|\vec{C}\bullet\overrightarrow{X^*}(t) - \vec{X}(t)\right| \\ \vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}\bullet\vec{D} \end{cases} \tag{3}$$

$\overrightarrow{X^*}(t)$ denotes the current optimal. When $prob < 0.5$ and $|\vec{A}| < 1$, the solution will be updated by the above formula.

2.1.2. Bubble-Net Attacking Mode. Bubble-net attacking is the most distinctive hunting method of humpback whales. In addition to swimming around prey, whales also move along a spiral path. WOA creates a spiral equation to simulate through the distance, as shown in the following formula:

$$\begin{cases} \overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right|, \\ \vec{X}(t+1) = \overrightarrow{D'} \bullet e^{bl} \bullet \cos\left(2\pi l\right) + \overrightarrow{X^*}(t). \end{cases} \tag{4}$$

Here, the distance between $\vec{X}(t)$ and $\overrightarrow{X^*}(t)$ is recorded as $\overrightarrow{D'}$, which can be understood as the distance between the whale and its prey. The constant b is used to define the shape of the logarithmic spiral. l is a random number between [0, 1]. When prob ≥ 0.5, Equation (4) will be used to generate a new solution.

2.2. QUasi-Affine Transformation Evolutionary (QUATRE) Algorithm. QUATRE is a coevolutionary algorithm based on quasi-affine transformation. As a new structure of evolutionary calculation, it can be used to solve the problem of distributed bias in the DE algorithm. The evolution formula is similar to the affine transformation, as shown below:

$$X \leftarrow M \bigotimes X + \bar{M} \otimes B. \tag{5}$$

Suppose there are ND-dimensional solution vectors X_i, which constitute a population matrix: $X = (X_1, X_2, \cdots, X_i, \cdots, X_N)^T$. There are two matrices: the donor matrix B and the coevolutionary matrix M. The operator $\bigotimes$ represents the bitwise multiplication of matrix elements. M and $\bar{M}$ are mutually inverse matrices, and and they are both binary matrices (matrix elements only take 0 or 1).

2.2.1. The Coevolutionary Matrix M. The QUTARE algorithm considers that all search agents in the population have the same status. By the cooperative structure, the search agents cooperate with each other and share information. The initialized matrix M_{initial} is constructed, and then, through a series of transformations, the coevolutionary matrix M is obtained.

There are three methods for constructing M_{initial}. The specific choice depends on the size relationship between the population size ps and the search space dimension D. If $\text{ps} = D$, M_{initial} is a D-dimensional binary lower triangular matrix, as shown in Equation (6). When $\text{ps} > D$, M_{initial} needs to be expanded. If $\text{ps} = p \bullet D$, M_{initial} is a matrix of pD-dimensional binary lower triangular matrices stacked vertically. If $\text{ps} = p \bullet D + q$, the first $p \bullet D$ rows are handled according to the previous situation, and the rest take the first q lines in Equation (6), as shown in Equation (7). Next, M can be obtained after two steps of transformation of M_{initial}. First, the elements of each row in M_{initial} are randomly arranged. Then, the elements of each row vector remain unchanged while their positions in the matrix are randomly arranged. So we can get M, see

$$M_{\text{initial}} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & & \cdots & \\ 1 & 1 & \cdots & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & & 1 & \\ 1 & 1 & \cdots & \\ & & \cdots & \\ & 1 & \cdots & 1 \end{bmatrix} = M. \tag{6}$$

$$M_{\text{initial}} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & & \cdots & \\ 1 & 1 & \cdots & 1 \\ 1 & & & \\ 1 & 1 & & \\ & & \cdots & \\ 1 & 1 & \cdots & 1 \\ 1 & & & \\ 1 & 1 & & \\ \cdots & & & \end{bmatrix} \sim \begin{bmatrix} 1 & & & \\ 1 & \cdots & & 1 \\ & 1 & \cdots & \\ 1 & & \cdots & 1 \\ 1 & & & \\ 1 & \cdots & 1 & \\ & & \cdots & \\ 1 & 1 & \cdots & 1 \\ 1 & 1 & & \\ 1 & & & 1 \\ & & 1 & \end{bmatrix} = M. \tag{7}$$

2.2.2. The Donor Matrix B. The donor matrix B dominates the way of evolution. Similar to the different mutation strategies in DE, there are many ways to generate B. The mode used here is called QUATRE/best/1, as shown in

$$B_{i,G} = X_{g\text{best},G} + F \bullet \left(X_{r1,G} - X_{r2,G}\right). \tag{8}$$

Assume that the population matrix of the Gth generation is $X_G = (X_{1,G}, X_{2,G}, \cdots, X_{i,G}, \cdots, X_{N,G})^T$, which consists of ND-dimensional candidate solutions $X_{i,G}$. Each $X_{i,G}$ has an evolution guide vector $B_{i,G}$ corresponding to it. $X_{g\text{best},G}$ represents the best individual in the population. X_{r1} and X_{r2} are generated by randomly arranging the row vectors of X_G. $F \in (0, 1]$ is the scaling factor.

3. The Hybrid Algorithms Based on WOA and QUATRE (WOA-QT)

The metaheuristic algorithm has two basic elements: exploration and exploitation. The task of exploration is to trial different solutions in the search space and try to find the global optimal. The task of the exploitation is to search in the local scope and fully tap the optimization opportunities brought by the current optimal. Ideally, if the two stages can cooperate with each other and realize smooth transition, the local optimum can be avoided and the optimal solution can be found finally [27].

WOA has proved to be superior to some advanced optimization algorithms, for example, PSO, DE, GSA, and fast evolutionary programing (FEP) [28]. It provides multiple update modes and strives to balance the relationship between exploration and exploitation. But inevitably, it still faces many problems, such as how to further improve the exploration ability, release the exploitation potential, and avoid local optimization. Local optimal stagnation and slow convergence are its two main disadvantages. In order to improve the ability to escape from the local optimum, this paper introduces the QUATRE algorithm into WOA and use the strategy of dynamically shrinking the search space to improve time efficiency. In this part, we will analyze the causes of the problems in detail and explain the relevant countermeasures.

3.1. The Combination of WOA and QUATRE. Although WOA has two different exploitation models, both of them are highly dependent on the current optimum. When the problem is a multimodal function or a complex function, there is a risk of falling into a local optimum. How to deal with this situation requires careful consideration.

QUATRE is a coevolution algorithm, which uses matrix mode calculation and concise parameters. From the point of view of statistics and probability theory, the moving way of individuals is more reasonable in the process of evolution. In addition, the coevolutionary matrix in the algorithm can perceive the structure of objective function more comprehensively, which makes the search of solution space more effective.

Based on the above considerations, this paper proposes a hybrid algorithm based on WOA and QUATRE. The specific idea is to embed QUATRE in the exploitation stage of WOA. The current optimal solution $\overrightarrow{X^*}(t)$ plays an absolute dominant role in both the encircling mode and the bubble-net attacking mode. The QUATRE algorithm can be used to preprocess the current optimal, and the effect is equivalent to perturb the current optimal solution in accordance with statistics and probability theory to make up for the lack of randomness and diversity. The fusion method of the two algorithms in the encircling mode is as follows:

$$\begin{cases} \overrightarrow{X_Q}(t) = M \bigotimes \overrightarrow{X^*}(t) + \bar{M} \otimes B. \\ \vec{D} = \left| \vec{C} \bullet \overrightarrow{X_Q}(t) - \vec{X}(t) \right|. \\ \vec{X}(t+1) = \overrightarrow{X_Q} - \vec{A} \bullet \vec{D}. \end{cases} \tag{9}$$

Here, the current optimal solution $\overrightarrow{X^*}(t)$ is processed by the QUATRE to get $\overrightarrow{X_Q}(t)$, which will replace $\overrightarrow{X^*}(t)$ to guide the position update of the encircling mode in the exploitation stage.

Similarly, in the bubble-net attacking mode, the fusion of the two algorithms is also performed, as shown in the following formula:

$$\begin{cases} \overrightarrow{X_Q}(t) = M \bigotimes \overrightarrow{X^*}(t) + \bar{M} \otimes B, \\ \overrightarrow{D'} = \left| \overrightarrow{X_Q}(t) - \vec{X}(t) \right|, \\ \vec{X}(t+1) = \overrightarrow{D'} \bullet e^{bl} \bullet \cos(2\pi l) + \overrightarrow{X_Q}(t). \end{cases} \tag{10}$$

$\overrightarrow{X_Q}(t)$ replaces $\overrightarrow{X^*}(t)$ to guide the position update of the bubble-net attacking mode during the exploitation phase. In the proposed WOA-QT, Equations (9) and (10) will still have a 50% probability of being selected.

QUATRE has different mutation strategies, which emphasize different focuses of search capabilities [29, 30]. The mode QUATRE/best/1 is concise in form which only uses the current optimal solution $X_{g\text{best},G}$ and two randomly selected solutions X_{r1} and X_{r2}. $X_{g\text{best},G}$ dominates the evolution of the population, so that individuals with higher fitness values can be fully utilized to perform good local search, and the convergence speed is faster. Although the combination of the two algorithms has increased the randomness of the population, we choose QUATRE/best/1 as the mutation strategy to avoid overcorrection. Therefore, both the population diversity is ensured and the convergence speed is also taken into account. Therefore, both the population diversity is ensured and the convergence speed is also taken into account.

3.2. Strategy of Dynamically Shrinking Search Space (DSS). The candidate solution should gradually approach the global optimal solution. The general trend is that the size of search space will decrease as the iteration progresses. However, the search space of WOA is always the size at the time of initialization. It is obviously unreasonable to roam randomly in such a fixed-size space, which may cause delay in the optimization process. In this paper, we propose a mechanism to dynamically shrink the search space, which is used in the exploration phase of WOA. As the optimization proceeds, the upper and lower limits of the selection range of the random solution will change accordingly, so that the size of the search space is gradually reduced, thereby improving search efficiency. The specific method is as follows:

First define four variables: the upper limit of the newly generated search space ub_new, the lower limit of the newly generated search space lb_new, the largest solution component of the current iteration ub_ max, and the smallest solution component of the current iteration lb_ min. The initial values of the above variables are set according to the initial size of the search space.

$$\text{ub}_{\text{new}} = \text{ub}_{\text{new}} + (\text{ub}_{\max} - \text{ub}_{\text{new}}) * \left(\frac{t}{\max_{\text{iter}}} \right), \tag{11}$$

$$\text{lb}_{\text{new}} = \text{lb}_{\text{new}} + (\text{lb}_{\min} - \text{lb}_{\text{new}}) * \left(\frac{t}{\max_{\text{iter}}} \right). \tag{12}$$

where t represents the current number of iterations and max _iter is the maximum number of iterations. As the iteration progresses, ub_new will decrease approximately linearly according to Equation (11), and lb_new will increase approximately linearly according to Equation (12). As a result, the exploration scope is gradually reduced, and the global optimum is constantly approached. The random solution in the exploration process is now generated from a dynamically shrinking search space defined by ub_new and

```
Generate initial population $X$ containing $N$ individuals $X_i (i = 0, 1, 2, 3, \cdots, N)$.
Initialize the parameters related to the algorithm: $a$, $r$, $b$, $prob$, $t$, $max\_iter$.
Initialize two matrices in QUATRE:
    the coevolutionary matrix $M$ (Eq. (6) or Eq. (7)).
    the donor matrix $B$ (Eq. (8)).
Initialize the parameters related to the search space size:
    $ub = ub\_new = ub\_max$.
    $lb = lb\_new = lb\_min$.
Calculate the fitness value of each solution.
$X^* =$ the best individual.
    While ($t <$ max_iter) and (not obtain the predefined function value).
    For each solution.
    Update $a$, $r$, $b$, $l$, $prob$, $M$, $B$.
        If1 (prob <0.5).
        If2 ($|A| < 1$).
            Update the position of the current solution using Eq. (9).
        Else if2 ($|A| \geq 1$).
            Select a random solution $X_{\mathrm{rand}}$.
            Update the position of the current solution using Eq. (2).
        End if2.
Else if1 ($prob \geq 0.5$).
            Update the position of the current solution using Eq. (10).
        End if1.
    End for.
    Check if any solution goes beyond the search space and amend it.
    Calculate the fitness value of each solution.
    Update $X^*$ if there is a better solution.
    $t = t + 1$.
    Shrink the search space size using Eq. (11) and (12).
End while.
Return $X*$
```

Pseudocode 1: The pseudocode of the proposed WOA-QT.

lb_new. And the contraction speed is not too fast, which can effectively avoid local optimal stagnation.

3.3. Hybrid Algorithm WOA-QT. This section describes the optimization steps of WOA-QT. The pseudo code is shown in Pseudocode 1.

(1) Initialization: generate N individuals $X_i (i = 0, 1, 2, 3, \cdots, \mathrm{N})$ randomly to form population X. Set initial values to parameters in WOA and QUATRE. Specify the maximum number of iterations max _iter and parameters related to the size of the search space (including initial upper limit ub and lower limit lb, the upper limit ub_new and lower limit lb_ new of search space after each iteration, the maximum solution component ub_ max and minimum solution component lb_ max of the current iteration).

(2) Evaluation: calculate each individual's fitness value $f(X_i)$. By comparing with each other, determine the individual X^* who currently has the best fitness value.

(3) Update: choose one of the modes to update the individual's position.

(i) Exploration: when $prob < 0.5$ and $|A| \geq 1$, use Equation ((2)) to update the solution. That is to select a random solution X_{rand} in the current search space (between the upper limit ub_new and lower limit lb_new).

(ii) Exploitation

(A) When $\mathrm{prob} < 0.5$ and $|A| < 1$, use Equation (9) to update the solution. This is the WOA encircling mode embedded with QUATRE

(B) B. When $\mathrm{prob} \geq 0.5$, use Equation (10) to update the solution. This is the WOA bubble-net attacking mode embedded with QUATRE

(4) Reevaluation: re-evaluate the population to find the best individual X^*. And shrink the size of search space according to the Equations (11) and (12).

(5) Termination: repeat steps (3) to (4). If a predefined function value has been obtained or all iterations have been completed, record the global optimal solution X^* and its best fitness value $f(X^*)$. The optimization process ends here.

4. Experiment Results and Analysis

This part selects 30 benchmark functions for performance evaluation, as shown in Tables 1, 2, and 3. Most of the benchmark functions come from literature [31], including three categories: unimodal, multimodal, and fixed-dimension complex functions. We also separately evaluate the performance in solving low-dimensional and high-dimensional problems. The selected unimodal functions and multimodal functions are scalable functions covering low and high

Table 1: Details of 10 unimodal functions.

Functions	DIMENSION	RANGE	$f\_\min$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30,500	[100, 100]	0
$F_2(x) = \sum_{i=1}^{n} ix_i^2$	30,500	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{n} \lvert x \rvert + \Pi_{i+1}^{n}$	30,500	[−10, 10]	0
$F_4(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j^2 \right)$	30,500	[100,100]	0
$F_5(x) = \max_i \{\lvert x_i \rvert, i\epsilon[1, m]\}$	30,500	[100,100]	0
$F_6(x) = \sum_{i=1}^{n-1} \left[100\left(x_{x+1} - x_i^2\right) + (x_i - 1)^2\right]$	30,500	[−30, 30]	0
$F_7(x) = \sum_{i=1}^{n-1} (x_i + 0.5)^2$	30,500	[100,100]	0
$F_8(x) = \sum_{i=1}^{n} ix^4$	30,500	[−1.28,1.28]	0
$F_9(x) = \sum_{i=1}^{n} ix^4 + \text{rand } [0, 1)$	30,500	[−1.28,1.28]	0
$F_{10}(x) = \sum_{i=1}^{n} \lvert x_i^2 \rvert$	30,500	[−1, 1]	0

dimensions. All the experiments were compared among WOA-QT, WOA, and QUATRE. Each algorithm runs independently 40 times. The specific experimental scheme is as follows.

4.1. Experiments for Low-Dimensional Unimodal and Multimodal Functions. First, the low-dimensional unimodal functions and multimodal functions are tested, and the dimension is set to 30. The comparative indicators we selected include the minimum (i.e., optimal value), medium, maximum, mean, and standard deviation values of each experiment. Figure 1 shows the convergence curve of each function.

In the test of 10 unimodal functions, we can see that WOA-QT has achieved the first place in the comparison of the optimal values for 9 times (F1-F6, F8-F10). Compared with the other two algorithms, WOA-Q's optimization accuracy and convergence speed have been significantly improved, showing excellent local search capabilities. In addition, WOA-QT also has an absolute advantage in the comparison of other statistical data, which shows that its performance stability is far better than WOA and QUATRE.

From the experimental results of 10 multimodal functions, we can see that WOA-QT has an absolute advantage in the ranking of all indicators. This shows that it can effectively avoid falling into the local optimal.

4.2. Experiments for High-Dimensional Unimodal and Multimodal Functions. This experiment continues to use the test functions from the previous section but expand the dimension to 500. The conclusion of this experiment is basically consistent with that of the last one. The WOA-QT still maintains good performance in high-dimensional situations. Please see Figure 2 for the function convergence curves.

4.3. Experimental Results of Fixed-Dimension Complex Function. We also tested 10 fixed-dimension complex functions. It is found that the comprehensive performance of WOA-QT is still the best of the three. Figure 3 shows the function convergence curves obtained in this experiment.

4.4. Analysis of Experiment Results. In this paper, different types of benchmark functions are selected for testing for the sake of the scientificity and fairness of evaluation. It also examines the algorithm's ability to solve scalable dimensional problems.

When solving the unimodal problem, WOA-QT embodies the outstanding local exploitation ability. In the face of multimodal problems, it can well avoid falling into local optimum. Both optimization accuracy and speed benefit greatly from the effective hybrid mechanism of the two algorithms. In addition, when the dimension of the problem space expands or shrinks, or when the problem is gradually complicated and the amount of calculation increases suddenly, WOA-QT still maintains reliable optimization performance, especially with a satisfactory convergence speed. The high time efficiency proves the effectiveness of the DSS strategy.

Table 2: Details of 10 multimodal functions.

Functions	DIMENSION	RANGE	$f\_$ min
$F_{11}(x) = 1 - \cos\left(2\Pi\sqrt{\left(\sum_{i=1}^{n} x_i^2\right)}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$	30,500	[−100,100]	0
$F_{12}(x) = \sum_{i=1}^{n}\left[x_i^2 - 10 * \cos(2\pi x_i) + 10\right]$	30,500	[−5.12,5.12]	0
$F_{13}(x) = -208\exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i}^{n} x_i^2}\right) - \exp\left(\frac{1}{m}\sum_{i=1}^{n}\cos(2\pi x_i) + 20 + 2.718\right)$	30,500	[−32, 32]	0
$F_{14}(x) = \frac{1}{4000} + \sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,500	[−600,600]	0
$F_{15}(x) = \frac{\pi}{n} * \left\{10 * \sin(\pi y_1) + \sum_{i-1}^{n-1}(y_i - 1)^2\left[1 + 10 * \sin^2(\pi y_{i+1})\right]\right\}$ $y_i = 1 + \frac{x_i + 1}{4} * u(z_i, a, k, m) = \begin{cases} k(x_i - a), x > a \\ 0, -a < x_i < a \\ k(-x_i - a), x > a \end{cases}$	30,500	[−50, 50]	0
$F_{16}(x) = \sum_{i=1}^{n} \mid x_i \sin(x_i) + 0.1x_i \mid$	30,500	[−100,100]	0
$F_{17}(x) = 0.59 + \frac{\sin^2\left(\sqrt{\left(\sum_{i=1}^{n} x_i^2\right)}\right)^{-0.5}}{1 + 0.0001\left(\sum_{i=1}^{n} x_4^2\right)}$	30,500	[−100,100]	0
$F_{18}(x) = \sum_{i=1}^{n} 0.1n - \left(0.1\sum_{i=1}^{n}\cos(5\pi x_i) - \sum_{i=1}^{n}(x_i - \pi)^2\right)$	30,500	[−10, 10]	0
$F_{19}(x) = \sum_{i=1}^{n-1} 0.1n\left(x_i^2 + x_{x+1}^2\right)^{0.25} \times \left[1 + \sin^2\left(50\left(x_i^2 + x_{x+1}^2\right)\wedge(0.1)\right)\right]$	30,500	[−1, 1]	0
$F_{20}(x) = \sum_{i=1}^{n}\left(10^6\right)^{\frac{i-1}{n-1}} x_i^2$	30,500	[−10, 10]	0

5. Node Location in WSN

WSN, RFID, cloud computing, and wireless data communication, etc., together constitute the technical foundation of IoT. Among them, WSN is not only an important information collection subject of the perceptual layer of IoT but also a basic transmission channel of the network layer. WSN plays a vital role in how to realize the intelligent perception, transmission, processing, and application of data in IoT. There are many research topics in WSN, among which node location has always been a research hotspot. Because the monitoring data with unknown location information is usually meaningless, and the location problem is the basis and premise of most applications, the performance of the sensor node location algorithm directly affects its usability. The commonly used evaluation indicators include location accuracy, scale, node density, and power consumption. This paper uses the proposed WOA-QT algorithm to solve the node location problem in WSN. Specifically, it is to intelligentize the node location algorithm to further improve the positioning accuracy.

Although there are many node location algorithms, they can be classified into range-based and range-free. Range-based location algorithm calculates the position of unknown nodes by measuring distance or angle information. Commonly used ranging technologies include Time of Arrival (TOA), Time Difference of Arrival (TDOA), Received Signal Strength of Indicator (RSSI), and Angle of Arrival (AOA). Range-free localization algorithm only determines the node position according to the connectivity of the network and other information without knowing the distance or angle. The commonly used methods are convex programming, DV-hop, centroid location algorithm, and so on.

Different location algorithms have their own advantages and disadvantages. The location algorithm used in this paper integrates RSSI ranging and weighted centroid positioning algorithms (WCL_RSSI), which can improve positioning accuracy to a certain extent. To meet the requirements of further reducing the positioning error, this paper uses the proposed WOA-QT algorithm to optimize WCL_RSSI.

5.1. RSSI Ranging Technology. The principle of the RSSI ranging method is to convert the strength of the received wireless signal energy into the distance between the nodes, and then use it to estimate the location of the unknown node. Because no additional hardware is required, and the cost and power

Table 3: Details of 10 fixed-dimension complex functions.

Functions	DIMENSION	RANGE	$f\_\min$
$F_{21}(x) = \left(\sum_{i=1}^{5} i * \cos\left((i+1)x_1 + i \right) \right) * \left(\sum_{i=1}^{5} i * \cos\left((i+1)x_2 \right) + i \right)$	2	[−5.12,5.12]	−186.7309
$F_{22}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 \left(b_i^2 + b_i x_2 \right)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[−5, 5]	0.0003
$F_{23}(x) = -\frac{1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{0.5\left(x_1^2 + x_2^2\right) + 2}$	2	[−5.12,5.12]	−1
$F_{24}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	[−5, 5]	0.398
$F_{25}(x) = \left[1 + (x_1 + x_2 + x_3)^2 * \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$ $\times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)$	2	[−2, 2]	3
$F_{26}(x) = -0.0001 \left(\left\lvert \sin(x_1) \sin(x_2) \exp\left(\left\lvert 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right\rvert \right) \right\rvert + 1 \right)^{0.1}$	2	[−10, 10]	−2.0626
$F_{27}(x) = -0.0001 \left(\left\lvert \cos(x_1) \cos(x_2) \exp\left(\left\lvert 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right\rvert \right) \right\rvert + 1 \right)^{0.1}$	2	[−10, 10]	−2.0626
$F_{28}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.1532
$F_{29}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.4028
$F_{30}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.5363

consumption are relatively low, the RSSI method is widely used, especially for large-scale WSN node location problems. However, environmental factors such as distance and obstacles will have a great impact on the transmission of wireless signals, resulting in severe attenuation of the signal's power strength. This paper selects the log-distance distribution model as the distance measurement model of wireless signal propagation path loss, as shown below:

$$\mathrm{RSSI}(d) = \mathrm{RSSI}(d_0) - 10\gamma lg\left(\frac{d}{d_0}\right) + X_\sigma. \tag{13}$$

Assuming that the distance between the node to be located and the signal source is d, $\mathrm{RSSI}(d)$ indicates the signal strength received by the node; d_0 is the reference distance, generally $1m$; γ is the path loss factor that reflects the influence of the environment on signal transmission. The more obstacles in the transmission path, the greater the value of γ; X_σ is a random variable, obeying normal distribution: $X_\sigma \sim N(\mu, \sigma^2)$. By Equation (13), the distance d can be obtained: $d = 10^{(RSSI(d0) - RSSI(d) + X_\sigma / 10\gamma)}$.

There are K beacon nodes in the target area: $S_i(x_i, y_i)$, $i = 1, 2, \cdots, K$. $N(x, y)$ is the unknown node. If the distance between N and S_i is too far, the signal power will be greatly attenuated, which will seriously affect the location accuracy. Generally speaking, the unknown node and the beacon node are not directly connected, and there may be several intermediate nodes between the two. Therefore, it can be considered that RSSI starts from the beacon node S_i and is finally transmitted to the unknown node N in a relay manner. The distance of the connection path between the two is recorded as D_i. If M intermediate nodes are passed, then $D_i = \sum_{l=1}^{M+1} d_r$, where d_r can be obtained by Equation (13).

In order to convert D_i to the Euclidean distance ρ_i between N and S_i, this requires knowing the approximate proportional relationship between the two distances. Assume that the connection distance between two beacon nodes S_i and S_j is $D_{i,j}$ (can be obtained by RSSI ranging), and the Euclidean distance is

$$d_{i,j} = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2}. \tag{14}$$

We define α_i as the path scale factor of the beacon node S_i, as shown below:

$$\alpha_i = \sum_{j=1, j \neq i}^{K} \left(D_{i,j} / d_{i,j}\right). \tag{15}$$

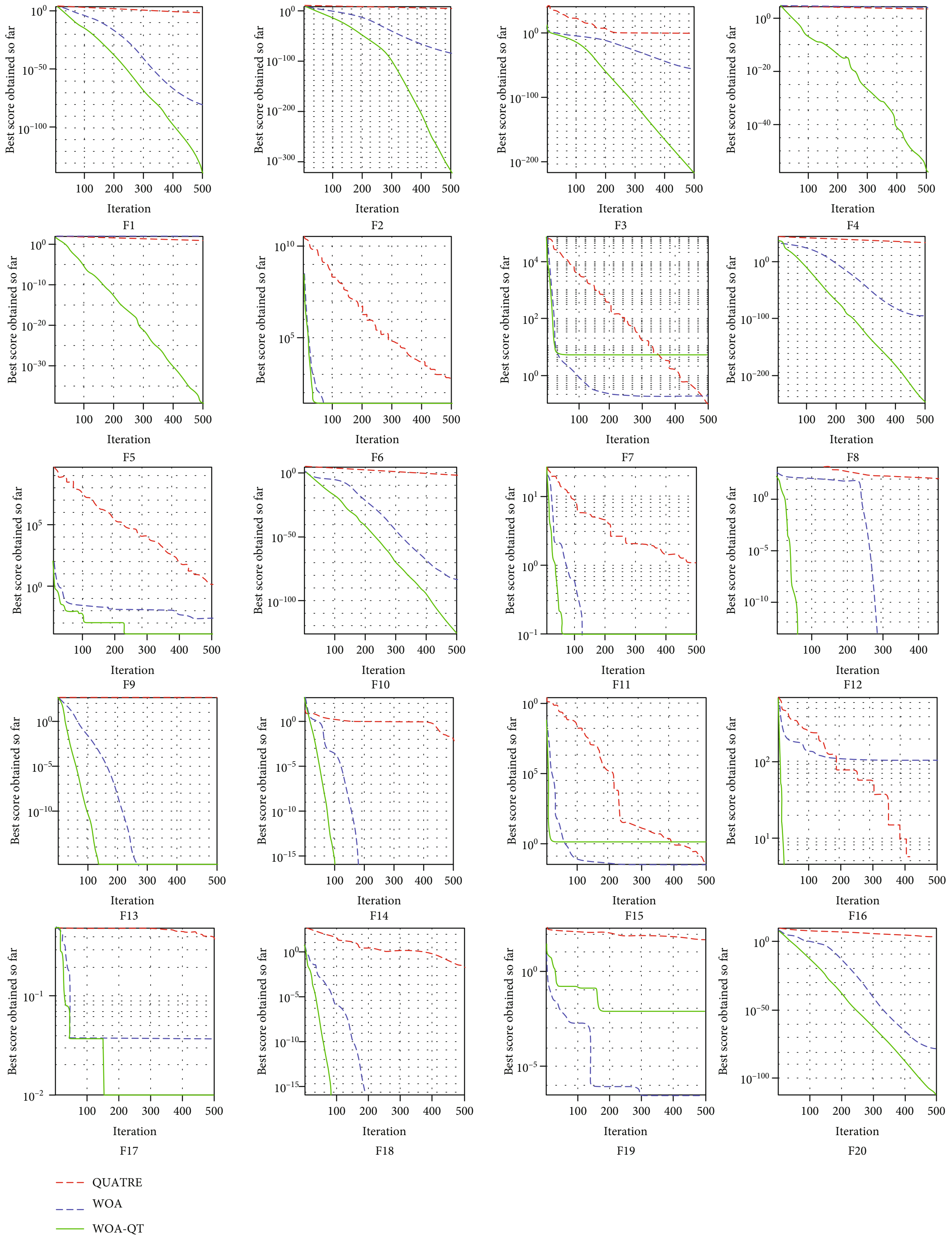


Figure 1: The convergence curves of 30-dimensional functions (F1-F20).

Obviously, the distance ρ_i can be obtained with the following formula:

$$\rho_i = D_i/\alpha_i. \tag{16}$$

5.2. Weighted Centroid Location Algorithm Based on RSSI (WCL_RSSI). It is not hard to see, if only relying on RSSI ranging to locate, the accuracy is poor. Therefore, a weighted centroid location algorithm (WCL) based on

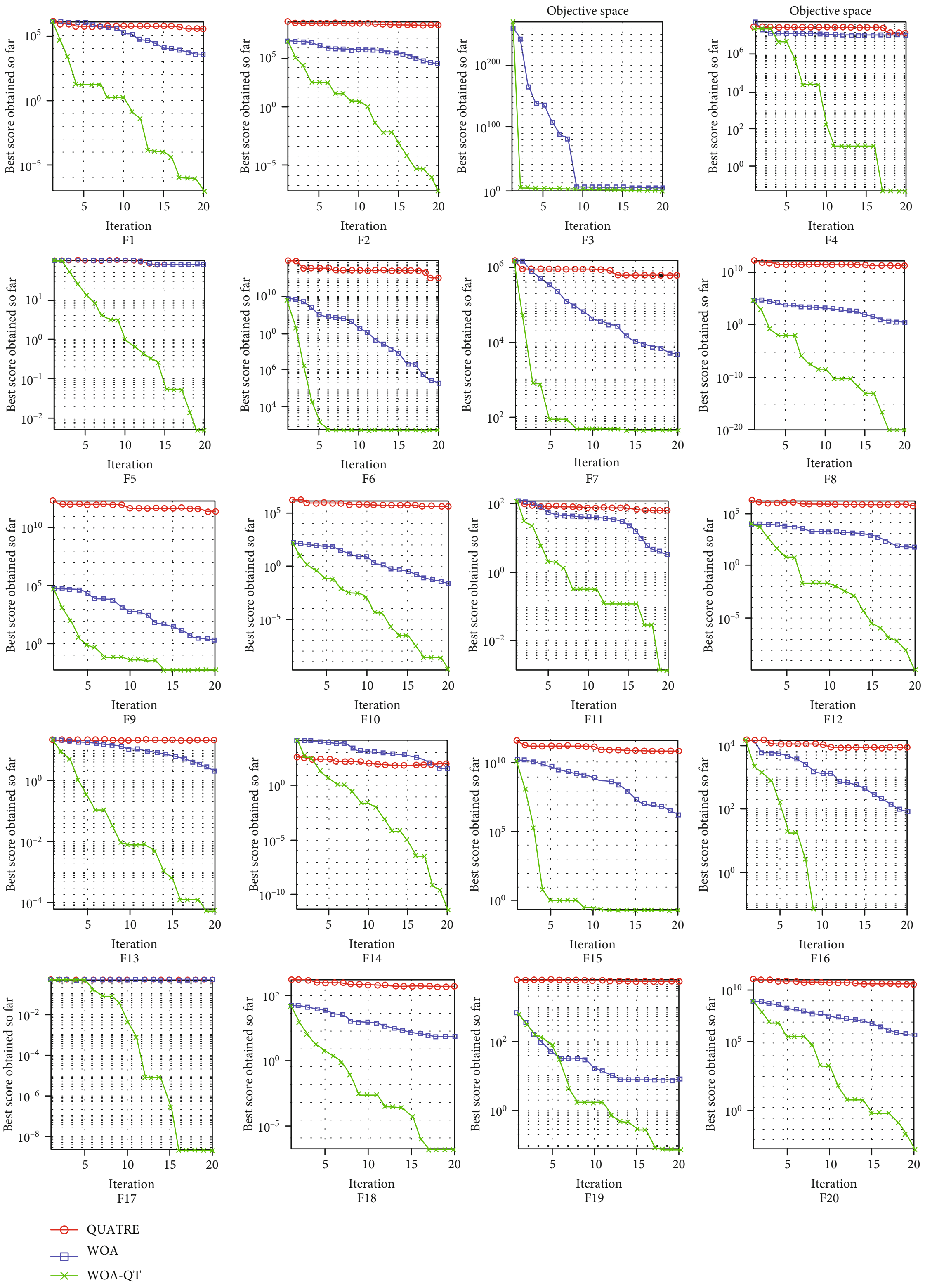


Figure 2: The convergence curves of 500-dimensional functions (F1-F20).

RSSI has emerged. The ordinary centroid location algorithm considers that all beacon nodes have equal status. In fact, different beacon nodes have different influences on the location of unknown nodes. The WCL algorithm is to reflect the degree of influence of each beacon node on the position of the centroid (i.e., the unknown node) through the weight factor. The triangle centroid positioning algorithm is used here. The pairwise distance between

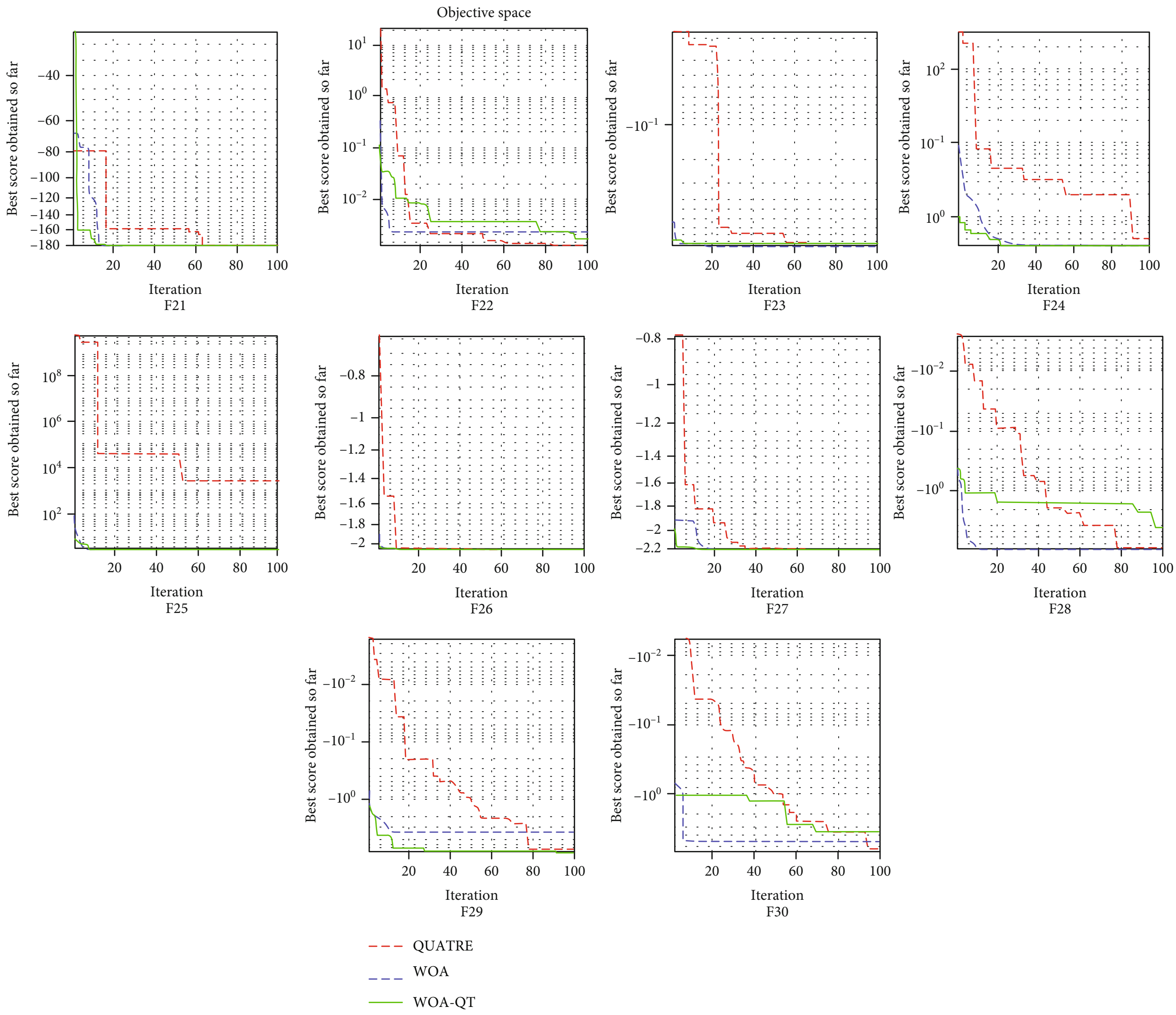


Figure 3: The convergence curves of fixed-dimension functions (F21-F30).

the unknown node and the three beacon nodes is recorded as (ρ_1, ρ_2, and ρ_3), which are used to construct the weight coefficient. The relevant formulas are as follows:

$$\mathrm{x}_l = \frac{x_1/(\rho_1+\rho_2)+x_2/(\rho_2+\rho_3)+x_3/(\rho_1+\rho_3)}{1/(\rho_1+\rho_2)+1/(\rho_2+\rho_3)+1/(\rho_1+\rho_3)},$$
$$y_l = \frac{y_1/(\rho_1+\rho_2)+y_2/(\rho_2+\rho_3)+y_3/(\rho_1+\rho_3)}{1/(\rho_1+\rho_2)+1/(\rho_2+\rho_3)+1/(\rho_1+\rho_3)}. \tag{17}$$

$S_1(x_1, y_1)$, $S_2(x_2, y_2)$, and $S_3(x_3, y_3)$ represent three beacon nodes, and the coordinate of the centroid of the triangle enclosed by them is the position of unknown nodes $N_l(x_l, y_l)$.

WCL_RSSI has the advantages of low computational complexity, simple implementation and high location accuracy. Moreover, the location process is less affected by the change of the transmission environment and does not require interaction between nodes [32].

5.3. Apply the Proposed WOA-QT Algorithm to WCL_RSSI. This article uses WOA-QT to improve WCL_RSSI in order to obtain the node position with higher accuracy. In the two-dimensional plane, the solution of the WOA-QT corresponds to the candidate position coordinates of the unknown node, which is denoted as $N_l(x, \mathrm{y})$. The Euclidean distance between $N_l(x, \mathrm{y})$ and the beacon node $S_i(x_i, y_i)$ is $d_o = \sqrt{(x-x_i)^2+(y-y_i)^2}$. $N_l(x_w, y_w)$ represents the location of unknown node estimated by the WCL_RSSI. Similarly, the distance between $N_l(x_w, y_w)$ and $S_i(x_i, y_i)$ is $d_w = \sqrt{(x_w-x_i)^2+(y_w-y_i)^2}$. Obviously, there is a deviation between d_o and d_w. It should be noted that the connection path between the unknown node and the beacon node has a great influence on the location accuracy. The greater the number of hops, the smaller the positioning error tends to be. The weight of hops needs to be considered when

Table 4: The average distance error of each algorism.

WOA-QT(WCL_RSSI)	PSO(WCL_RSSI)	WOA(WCL_RSSI)	QUATRE(WCL_RSSI)	WCL_RSSI
20.7632	23.315	22.7465	22.6332	24.9505

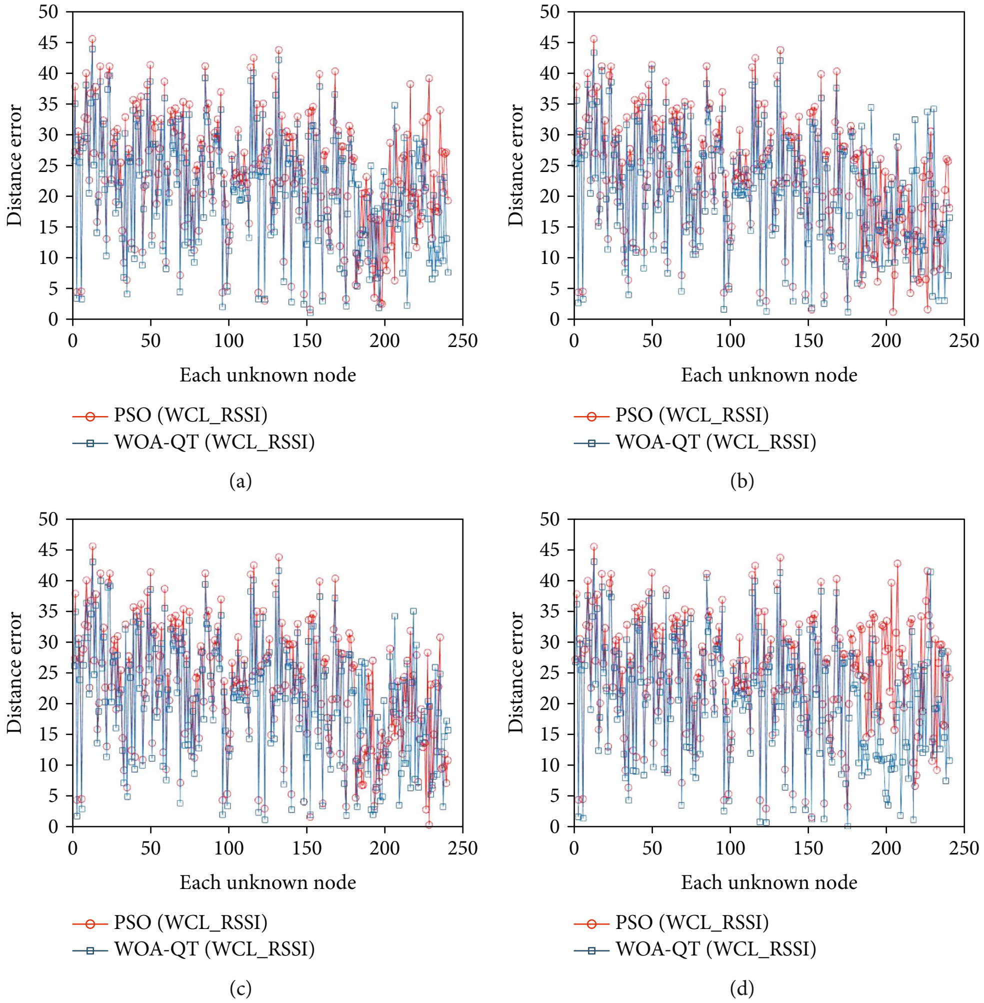


Figure 4: The comparison of location errors. (a) PSO and WOA-QT. (b) QUATRE and WOA-QT. (c) WOA and WOA-QT. (d) WCL based on RSSI and WOA-QT.

measuring the distance error. Therefore, the evaluation function $f(x, y)$ is as follows:

$$f(x, y) = \min \left(\sqrt{\frac{1}{K} \sum_{i=1}^{K} \left(\frac{d_o - d_w}{\text{hop}_i} \right)^2} \right). \quad (18)$$

There are K beacon nodes. The hop number between the beacon node S_i and the unknown node is recorded as hop_i. Equation (18) expresses the optimization goal, that is, to minimize the root mean square error of the difference between the two distances.

5.4. Simulation Results. Assume that in a two-dimensional area of 1000 m × 1000 m, 60 beacon nodes are deployed, and the positions of 240 unknown nodes are to be determined. The communication radius of sensor node is 15 meters, and the maximum number of iterations of the algorithm is 400. Table 4 lists the average distance error between the unknown node position estimated by each algorithm and the true position. The simulation experiment involves five algorithms. Simulation results show that WCL_RSSI based on WOA-QT has the smallest error, and the positioning accuracy is improved obviously. In order to visualize the positioning error, Figure 4 marks the distance deviation between the estimated position of each unknown node and its true position.

6. Conclusion

AI and IoT are compatible and symbiotic, who are born to be the best partner. They have broken through the bottlenecks of their respective parallel developments, enabling high-frequency resonance and coordinated development.

In this paper, the optimization algorithm in AI is used to solve the node location problem in WSN which is the key technology of IoT. There are two optimization algorithms involved: WOA and QUATRE. WOA is derived from natural wisdom, and the QUARTRE has the beauty of mathematics. The two algorithms utilize each other and work closely together to form an organic whole. The overall optimization performance is much better than that when the two are separated. The potential of local exploitation is further tapped through the mutual fusion, so that the local stagnation is effectively avoided in the process of searching for the global optimal. Although algorithm hybridization will inevitably bring more calculations, but with the cooperation of the two algorithms and the DSS strategy, the time efficiency of the algorithm has not been affected, and even better in many cases. So the increase in complexity of this algorithm is worthwhile. With the help of optimization technology, the node location accuracy of WSN has been significantly improved. It can be said that a successful AIoT case with application value is realized.

The QUATRE algorithm has many variants. When combined with other algorithms, we can consider how to choose different variants adaptively [33]. In addition, the complexity caused by algorithm hybridization can be considered to use surrogate-assisted technology to ease [34, 35]. In the field of WSN, there are many challenging problems that need to be overcome [36, 37]. It can be combined with other AI technologies such as machine learning [38–40] to promote the birth of more application scenarios of AIoT.

Data Availability

1. 30 well-known benchmark functions. 2. Random position location of sensor nodes by computers.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] J. Gubbi, B. Rajkumar, M. Slaven, and P. Marimuthu, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] A. Triantafyllou, P. Sarigiann, and T. D. Lagkas, “Network Protocols, Schemes, and Mechanisms for Internet of Things (IoT): Features, open Challenges, and Trends,” *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5349894, 24 pages, 2018.

[3] K. Deb, A. Samir, P. Amrit, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] K. V. Price, *Differential Evolution*, Springer, Berlin, Heidelberg, 2013.

[5] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, Australia, November 1995.

[6] D. Karaboga and C. Ozturk, “A novel clustering approach: artificial bee colony (ABC) algorithm,” *Applied Soft Computing*, vol. 11, no. 1, pp. 652–657, 2011.

[7] P. W. Tsai, M. K. Khan, J.-S. Pan, and B. Y. Liao, “Interactive artificial bee colony supported passive continuous authentication system,” *IEEE Systems Journal*, vol. 8, no. 2, pp. 395–405, 2014.

[8] S.-C. Chu, P. W. Tsai, and J.-S. Tsai, “Cat swarm optimization,” in *Pacific Rim international conference on artificial intelligence*, Springer, Berlin, Heidelberg, August, 2006.

[9] P. W. Tsai, J.-S. Pan, S. M. Chen, and B. Y. Liao, “Enhanced parallel cat swarm optimization based on the Taguchi method,” *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309–6319, 2012.

[10] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, no. 5, pp. 51–67, 2016.

[11] J.-S. Pan, J.-L. Liu, and E.-J. Liu, “Rank-based whale optimization algorithm for solving parameter optimization of solar cells,” *International Journal of Modeling and Optimization*, vol. 9, no. 4, pp. 209–215, 2019.

[12] A. E. Ezugwu and D. Prayogo, “Symbiotic organisms search algorithm: theory, recent advances and applications,” *Expert Systems with Applications*, vol. 119, pp. 184–209, 2019.

[13] S.-C. Chu, Z. G. Du, and J.-S. Pan, “Symbiotic Organism Search Algorithm with Multi-Group Quantum-Behavior Communication Scheme Applied in Wireless Sensor Networks,” *Applied Sciences*, vol. 10, no. 3, p. 930, 2020.

[14] H. Abdolreza, “Black hole: a new heuristic optimization approach for data clustering,” *Information Sciences*, vol. 222, pp. 175–184, 2013.

[15] J.-S. Pan, Q. W. Chai, S.-C. Chu, and N. X. Wu, “3-D Terrain Node Coverage of Wireless Sensor Network Using Enhanced Black Hole Algorithm,” *Sensors*, vol. 20, no. 8, p. 2411, 2020.

[16] Z. Y. Meng, J.-S. Pan, and H. Xu, “QUasi-Affine TRansformation evolutionary (QUATRE) algorithm: a cooperative swarm based algorithm for global optimization,” *Knowledge-Based Systems*, vol. 109, pp. 104–121, 2016.

[17] Z. Y. Meng and J.-S. Pan, “QUasi-Affine TRansformation evolution with external ARchive (QUATRE-EAR): an enhanced structure for differential evolution,” *Knowledge-Based Systems*, vol. 155, pp. 35–53, 2018.

[18] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[19] T. T. Nguyen, J.-S. Pan, and T. K. Dao, “An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network,” *IEEE Access*, vol. 7, pp. 75985–75998, 2019.

[20] J.-S. Pan, P. Hu, and S.-C. Chu, “Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power,” *PRO*, vol. 7, no. 11, pp. 845–868, 2019.

[21] P. Hu, J.-S. Pan, S.-C. Chu, Q. W. Chai, T. Liu, and Z. C. Li, “New Hybrid Algorithms for Prediction of Daily Load of Power Network,” *Applied Sciences*, vol. 9, no. 21, p. 4514, 2019.

[22] J.-S. Pan, P. C. Song, S.-C. Chu, and Y. J. Peng, “Improved Compact Cuckoo Search Algorithm Applied to Location of Drone Logistics Hub,” *Mathematics*, vol. 8, no. 3, p. 333, 2020.

[23] M. Rout and R. Roy, "Optimal wireless sensor network information coverage using particle swarm optimisation method," *International Journal of Electronics Letters*, vol. 5, no. 4, pp. 491–499, 2016.

[24] T.-W. Sung and C.-S. Yang, "Distributed Voronoi-based self-redeployment for coverage enhancement in a mobile directional sensor network," *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, Article ID 165498, 2016.

[25] S.-C. Chu, T. K. Dao, and J.-S. Pan, "Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–15, 2020.

[26] T. K. Dao, T. T. Nguyen, J.-S. Pan, Y. Qiao, and Q.-A. Lai, "Identification failure data for cluster heads aggregation in WSN based on improving classification of SVM," *IEEE Access*, vol. 8, pp. 61070–61084, 2020.

[27] P. W. Tsai and C.-W. Chen, "Review on swarm intelligence for optimization," *Computing Science and Technology International Journal*, vol. 2, no. 1, pp. 13–17, 2014.

[28] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[29] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[30] Q. Fan, X. F. Yan, and Y. Xue, "Prior knowledge guided differential evolution," *Soft Computing*, vol. 21, no. 22, pp. 6841–6858, 2016.

[31] S. Gupta, K. Deep, S. Mirjalili, and J. H. Kim, "A modified sine cosine algorithm with novel transition parameter and mutation operator for global optimization," *Expert Systems with Applications*, vol. 154, article 113395, 2020.

[32] J. Wang, P. Urriza, Y. X. Han, and D. Cabric, "Weighted centroid localization algorithm: theoretical analysis and distributed implementation," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3403–3413, 2011.

[33] S. L. Wang, T. F. Ng, and F. Morsidi, "Self-adaptive ensemble based differential evolution," *International Journal of Machine Learning and Computing*, vol. 8, no. 3, pp. 286–293, 2018.

[34] C. L. Sun, Y. C. Jin, R. Cheng, J. L. Ding, and J. C. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644–660, 2017.

[35] S. F. Qin, C. L. Sun, Y. C. Jin, and G. C. Zhang, "Bayesian approaches to surrogate-assisted evolutionary multi-objective optimization: a comparative study," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Xiamen, China, December 2019.

[36] V. Sadhu, X. Y. Zhao, and D. Pompili, "Energy-efficient analog sensing for large-scale and high-density persistent wireless monitoring," *IEEE Internet of Things Journal*, vol. 7, p. 1, 2020.

[37] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.

[38] M. A. Alsheikh, S. W. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2019.

[39] K. Thangaramya, K. Kulothungan, R. Logambigai, M. Selvi, S. Ganapathy, and A. Kannan, "Energy aware cluster and Neuro-fuzzy based routing algorithm for wireless sensor networks in IoT," *Computer Networks*, vol. 151, pp. 211–223, 2019.

[40] L. F. Gou, H. H. Li, H. Zheng, H. C. Li, and X. X. Pei, "Aeroengine control system sensor fault diagnosis based on CWT and CNN," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5357146, 12 pages, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8870649, 9 pages
https://doi.org/10.1155/2020/8870649

Research Article

Flower End-to-End Detection Based on YOLOv4 Using a Mobile Device

Zhibin Cheng[1] **and Fuquan Zhang**[2]

[1]*Educational Administration and Scientific Research, Fujian Polytechnic of Information Technology, Fuzhou, Fujian 350003, China*
[2]*College of Computer and Control Engineering, Minjiang University, Fuzhou 350108, China*

Correspondence should be addressed to Fuquan Zhang; 8528750@qq.com

Received 30 July 2020; Revised 27 August 2020; Accepted 4 September 2020; Published 17 September 2020

Academic Editor: Chao-Yang Lee

In this paper, a novel flower detection application anchor-based method is proposed, which is combined with an attention mechanism to detect the flowers in a smart garden in AIoT more accurately and fast. While many researchers have paid much attention to the flower classification in existing studies, the issue of flower detection has been largely overlooked. The problem we have outlined deals largely with the study of a new design and application of flower detection. Firstly, a new end-to-end flower detection anchor-based method is inserted into the architecture of the network to make it more precious and fast and the loss function and attention mechanism are introduced into our model to suppress unimportant features. Secondly, our flower detection algorithms can be integrated into the mobile device. It is revealed that our flower detection method is very considerable through a series of investigations carried out. The detection accuracy of our method is similar to that of the state-of-the-art, and the detection speed is faster at the same time. It makes a major contribution to flower detection in computer vision.

1. Introduction

In recent years, flower classification and detection has been of considerable interest to the computer vision community which can be applied in AIoT for the smart garden. The flowers in the smart garden can be automatically designed and recommended to make it more beautiful. The previous work mostly focused on flower classification [1–5] using a traditional detector and method [6, 7]. While it has become a tendency in flower classification and detection based on deep learning anchor-based approaches, flower detection was paid little attention. In most studies of deep learning anchor-based flower detection, the approaches can be divided into two general classes, two-stage object detection [8–10] and one-stage object detection [11–14].

1.1. Flower Classification and Detection Based on Deep Learning Two-Stage Approaches. As we all know, flower detection has become a hot topic in object detection since the convolutional neural network reborns worldwide in 2012. And mainstream flower detection was divided into two categories, one-stage detector approaches and two-stage detector approaches, which were based on anchor approaches. The essence of an anchor is the candidate boxes, which are designed with different scales and proportions and are classified by DNN. The positive anchor can learn how to return it to the right place, and it plays a role which is similar to the sliding window mechanism in traditional detection algorithms. In recent years, many researchers perform flower and fruit classification and detection based on convolutional neural network (CNN) approaches [15–18], which is a kind of feedforward neural network convolution computation contained and a deep structure. And it is one of the representative algorithms commonly used in deep learning. In 2014, Girshick et al. proposed a region with CNN characteristics (RCNN) of the first two-stage object detection (RCNN). And in 2015, they presented Fast RCNN, which enables us to train both the detector and the bounding box regressor in the same network configuration. Based on these models, in 2015, Ren et al. claimed a Faster RCNN detector, which is the first end-to-end and almost in a real-time deep learning detector (Faster RCNN) which proposal detection, feature extraction, bounding box regression, and so on have been gradually integrated into a unified end-to-end learning framework in Faster RCNN. So many researchers have applied them in flower and all kinds of fruit detection including all kinds of fruit like mangoes, almonds, and apples [19–22].

Great progress has been made in flower detection based on two-stage approaches in high accuracy. However, it is concluded that the speed of it needs to be increased.

1.2. Flower Classification and Detection Based on Deep Learning One-Stage Approaches. To overcome the above limitations, considering the speed of the running time, the other one-stage flower detection method was demonstrated. Redmon et al. in 2015 demonstrated the YOLO one-stage detector, which divides the image into several regions and predicts the boundary box and probability of each region. It has fast detection speed which can contribute to processing streaming media video in real time. Compared with other algorithms such as the two-stage detection method, it has half as many false backgrounds as them on account of capturing contextual information effectively and a strong versatility and generalization ability. Then, Redmon et al. made a series of improvements based on YOLO [12], including YOLOv2 [13] and YOLOv3 [11], which has the same accuracy as the two-stage detector at a higher speed. It is faster in YOLOv2 than other detection systems in a variety of monitoring data sets, besides the tradeoff between speed and accuracy can be made. As we all know, improving recall and positioning accuracy was focused on in YOLOv2, while maintaining classification accuracy. However, in YOLOv3, on the premise of maintaining the speed advantage, the prediction accuracy is improved; particularly, the recognition ability of small objects is strengthened. It adjusted the network structure and multiscale feature object detection method used and softmax utilized for object classification that counts. Therefore, many efforts were made to perform flower and fruit detection based on these algorithms of the YOLO detection method [23–25]. While its detection speed has been greatly improved compared with the two-stage detector, the positioning accuracy of it has been reduced, especially for some small objects. To overcome these limitations, Liu et al. in 2015 presented a Single Shot MultiBox Detector (SSD) to raise accuracy, especially small objects. The main contribution of SSD is the introduction of multireference and multiresolution detection technology to perform the detection only on its top layer. So many fruit and flower detection methods were based on SSD [26, 27], which achieved great success.

While great progress has been made in one-stage flower detection and has become a tendency in recent years, however, to obtain abundant image information and enhanced accuracy is still a challenge. Most of the studies were focused on accuracy or speed; however, few studies have considered the accuracy coming up with the speed of the object detection. So it is emphasized that the accuracy should be matched with the speed of the detector in our method. In conclusion, flower detection is based on one-stage approaches working well; the accuracy is still to be enhanced a little.

Although current anchor-based deep learning flower detection methods work well, they still suffer from the following six problems: (1) Due to the irregular shape of the flowers, the bounding box covers a great deal of nonflower regions, which caused a lot of interference. (2) The setting of anchor needs to be designed manually, and different designs are required for different flower datasets, which are quite troublesome and does not conform to the design idea of DNN. (3) The matching mechanism of an anchor makes the frequency of extreme scale (very large and very small object) to match lower than that of moderate size object. It is not easy for DNN to learn these extreme flower samples well when learning. (4) The large number of anchors causes serious imbalance. (5) On account of the unlabelled flower dataset, which only can be trained in the flower classification model, it cannot perform flower detection in a mobile device and AIoT. (6) Labelling the flower dataset consumes a lot of time and power, therefore most of the researchers were not able to pay attention to the flower detection which can be applied in a mobile device and AIoT.

In order to overcome the above limitations, a new object detection method based on new anchor-based approaches and a new-labelled flower dataset is adopted in this paper. To acquire more useful information and more precious object position in the image, attention mechanism SAM is utilized, which the output feature map of the channel attention module is taken as the input feature map of this module. In our paper, we can regard SAM as an attention mechanism to be applied to both channel and spatial dimensions and it can be embedded in most of the current mainstream networks and can improve the feature extraction capability of network models without significantly increasing the amount of computation and parameters. The baseline of our dataset is Oxford 102 Flower dataset, and we have labelled the dataset with all categories and all label and geometry positions to perform flower detection and other plant detection in a smart garden. Our backbone network is the CSPDarnet53 network, and it was designed to solve the previous work in the reasoning process which requires a lot of computation from the point of view of network structure. Also, SSP block is added in CSPDarknet53 and it significantly increases the acceptance field, extracts the most important contextual characteristics, and does little to slow down network operations. Our contributions are summarized as follows:

(i) We present a flower detection method, an end-to-end deep convolutional neural network for flower detection applied in a smart garden in AIoT. The backbone network we applied is the CSPDarnet53 network, which can reduce computation when ensuring accuracy by integrating gradient changes into feature maps from end to end. And the running time of the flower detection in the architecture we used is the state-of-the-art fastest model compared with other models, especially when integrated into the mobile device. In order to extract the most important contextual characteristics and also increase the acceptance field, SSP block is added in the backbone network when keeping the network operations. Besides, attention mechanism SAM is utilized to select the information that is more critical to the current task target from the numerous information

(ii) We labelled Oxford 102 Flower dataset with annotation containing 102 categories of flowers common in the UK, with each category containing 40 to 258 images, for a total of 8,189 images to perform flower detection. As we all know, the previous work attached much importance to the flower classification with an unannotated dataset. In our work, flower detection was paid much attention to with the annotation flower dataset which can be trained in our architecture. Also, the flower detection we proposed can be integrated into the mobile device to make it convenient to operate in a smart garden which can be applied in flower arrangement and flower horticulture. What is more, it can not only enhance the user's human-computer interaction experience of flower arrangement but save time and cost of horticulturists. It has preliminarily realized the concept of the smart garden in AIoT

The paper is organized as follows. Section 2 describes the materials and methods in our paper including the network architecture and the dataset we labelled. Section 3 describes our results and discussion. We conclude in Section 4.

2. Materials and Methods

The architecture of the flower detection system is shown as in Figure 1, when the flower picture is acquired by the mobile device which is based on 5G networks meaning that Internet of Everything; it is passed through our backbone network, the neck module with SPP and PAN and the module of YOLOv3 head. At last, the result of the flower picture can be obtained which can be applied in the smart garden in AIoT.

2.1. Network Architecture. The architecture of our network is shown in Table 1, and the backbone network we utilized is the CSPdarknet53 network, combining CSPNet [28] and darknet53. The CSPNet was designed to solve the previous work in the reasoning process which requires a lot of computation from the point of view of network structure. It is considered that the problem of high inference computation is caused by the gradient information repetition in network optimization while CSPNet can reduce computation when ensuring accuracy by integrating gradient changes into feature maps from end to end. It not only can enhance the learning ability of the CNN but also can reduce computing bottlenecks and memory costs while keeping accuracy in the lightweight. It is a utilized idea ResNET for reference and adding a residue module to the Darknet53 network which can help to solve the deep network gradient problem. Two convolutional layers and one shortcut connection are contained in each residue module, and there are several duplicate residual modules in the layers. The pool layer and full connection layer were not involved in the architecture; the network undersampling is achieved by setting the stride of convolution as 2. After passing through this convolutional layer, the size of the image will be reduced to half. Convolution, BN, and Leaky Relu are contained in each convolutional layer, and a zero padding is added after each residual module.

It is studied that CSPDarkNet53 has advantages in object detection, which is better as the backbone of the test model. In CSPDarknet53, the parameters of CSPDarknet53 for image classification are 27.6 M which is bigger than other neural networks, and the receptive field size is also much bigger than neural networks. So it is a very suitable backbone for our proposed flower detection method.

In addition, the SSP block is attached to the backbone network CSPDarknet53, which can produce an output of fixed size regardless of input size and can use different dimensions of the same image as input to get pooling features of the same length. When SPP is placed behind the last convolutional layer, it has no influence on the structure of the network and just replaces the original pooling layer. It can be used not only for image classification but also for object detection. When the SSP block is made use of in CSPDarknet53, it significantly increases the acceptance field, extracts the most important contextual characteristics, and does little to slow down network operations.

Also, PANET [29] was used instead of FPN in YOLOv3 as a parametric polymerization method for different bone levels for different detector levels. It is a bottom-up path enhancement that is aimed at facilitating the flow of information which can contribute to shortening the information path, enhancing the feature pyramid, and accurately locating the signal present at low levels to enhance the whole feature level. PANET developed adaptive feature pools to connect the feature grid to all the feature layers to enable useful information from each feature layer to propagate directly to the proposed subnetwork below.

YOLOv3 [11] is utilized in the head of the architecture, which is an anchor-based detection model. The residual network structure is used for reference to form a deeper network level and multiscale detection in the YOLOv3 model, which can improve mAP and small object detection effect.

2.2. Flower Dataset. As shown in Figure 2, Oxford 102 Flower dataset is a common flower dataset used in research and experiment when it comes to plant field. It contains 102 categories of flowers common in the UK, with each category containing 40 to 258 images, for a total of 8,189 images. Large proportions, gestures, and light changes are involved in the images which can be used for image classification studies. However, it cannot be used for flower detection when it comes to a smart garden in AIoT. Besides, the unlabelled flower dataset cannot be applied in a mobile device and other smart devices.

In our work, the Oxford 102 Flower dataset is labelled not only containing flower classification to perform flower detection to be utilized in a smart garden in AIoT. As shown in Figure 3, the labelled flower dataset can be contributed to all flower, plant, and fruit detection studies, especially in a completed natural environment. The dataset was labelled by a professional data annotator to classify the flower dataset which can contribute to the accuracy of the flower detection model and make it easier to perform real-time flower detection.

More training data can lead to a more sound model. If there are limited data volumes, it can make use of data augmentation to increase the diversity of the training sample to enhance model robustness, avoid overfitting, and improve

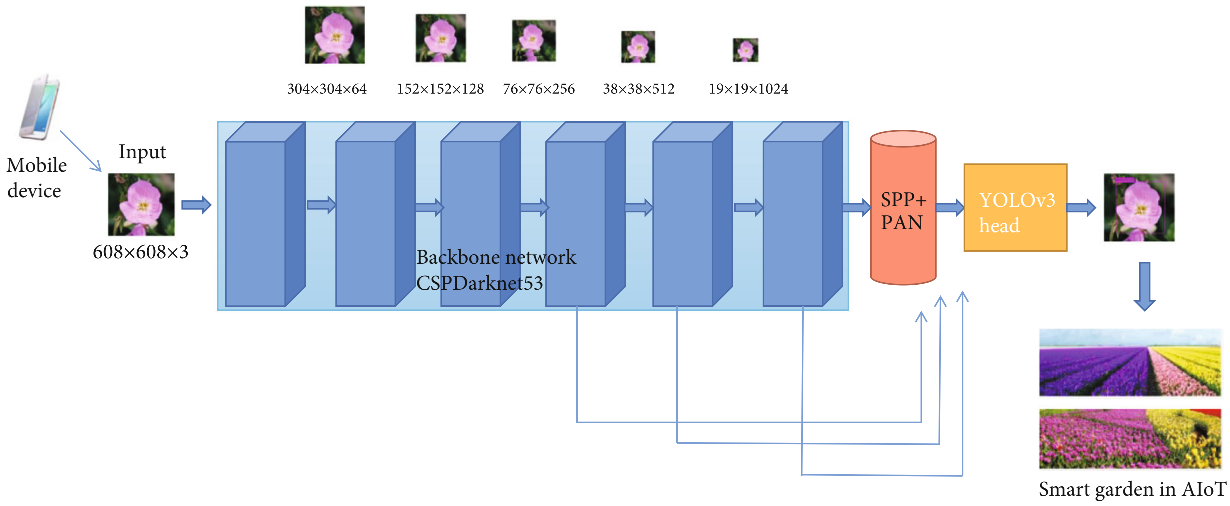


Figure 1: The architecture of our flower detection system.

Table 1: The architecture of the backbone network CSPDarknet53.

Type	Filters	Output
DarknetConv2D BN Mish	32	608 × 608
ResBlock	64	304 × 304
2 × ResBlock	128	152 × 152
8 × ResBlock	256	76 × 76
8 × ResBlock	512	38 × 38
4 × ResBlock	1024	19 × 19

the generalization ability of the model. The common methods include flip, rotation, shift, resize, random crop or pad, color jittering, and noise. In our dataset, data augmentation is also used to enhance our model.

2.3. Loss Function. In our work, the loss function DIoU loss is the loss function we used, which is feasible to directly minimize the normalized distance between the anchor frame and the target frame to achieve faster convergence speed and is more accurate and faster when overlapped or even included with the target box when making regression.

DIoU loss is on the basis of IoU (Intersection over Union), considering the information of the center distance of the bounding box. The definition of it can be defined as follows, formula (1), where B^{gt} is the target box and B is the prediction box. And the loss function of the IoU is defined as formula (2); it is shown that it works if the bounding boxes overlap; there is no overlap if the gradient does not change.

$$\mathrm{IoU} = \frac{B \cap B^{gt}}{B \cup B^{gt}}, \tag{1}$$

$$L_{\mathrm{IoU}} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}. \tag{2}$$

Therefore, it is GIoU that can improve the loss of IoU in the case that the gradient does not change without overlapping boundary boxes, which adds a penalty term on the basis of the loss function of IoU. It is defined as the following formula (3). The additional parameter C in the formula represents the minimum boundary box that can cover both B and B^{gt}. However, if one of B or B^{gt} overrides the other box in the case, the penalty term cannot work, which can be regarded as an IoU loss.

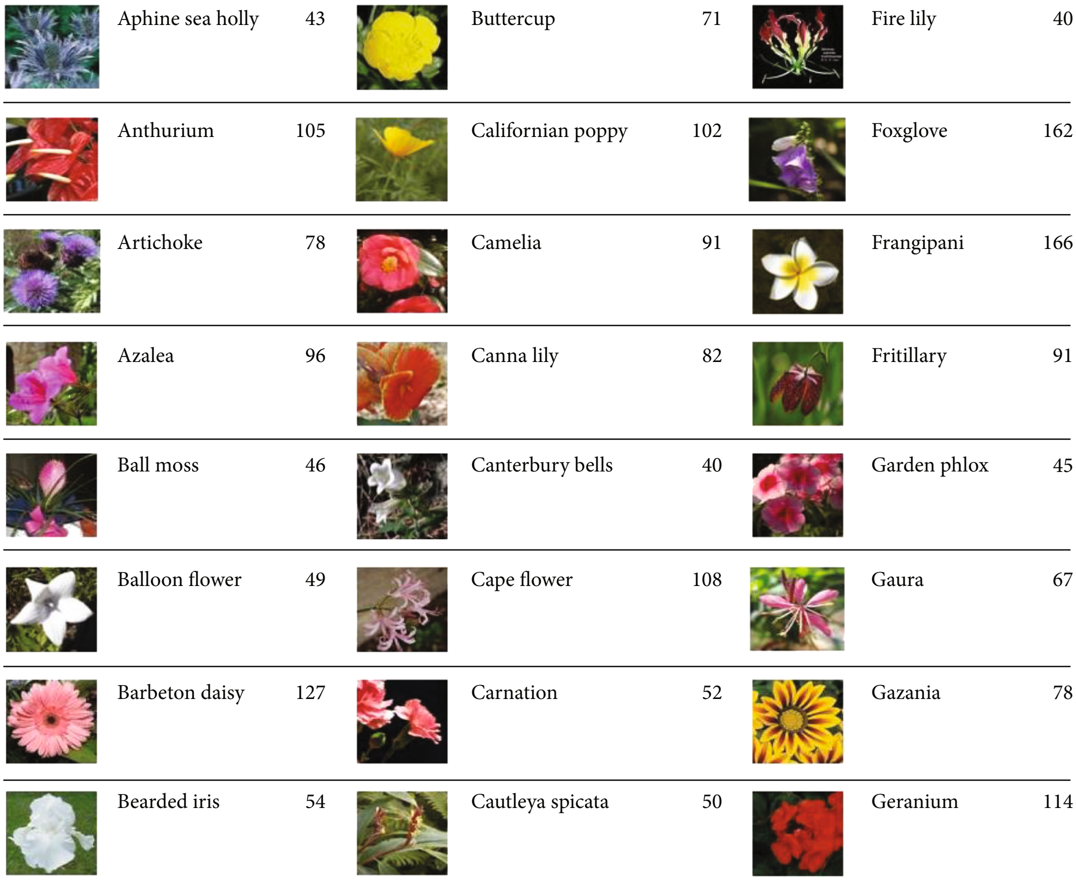
$$L_{\mathrm{GIoU}} = 1 - \mathrm{IoU} + \frac{\left|C - B \cap B^{gt}\right|}{|C|}. \tag{3}$$

To solve the limitations, the DIoU is introduced which can be defined as formula (4), where b, b^{gt} represents the center point of the anchor frame and target frame, respectively, and p is the Euclidean distance between two centers. Also, c represents the diagonal distance of the minimum rectangle that can cover the anchor and target box simultaneously. What is more, the normalized distance between the anchor box and the target box is modeled in DIoU. The loss function of DIoU can be defined as formula (5).

$$R_{\mathrm{DIoU}} = \frac{p^2\left(b, b^{gt}\right)}{c^2}, \tag{4}$$

$$L_{\mathrm{DIoU}} = 1 - \mathrm{IoU} + \frac{p^2\left(b, b^{gt}\right)}{c^2}. \tag{5}$$

Besides, CIoU is also proposed to our loss function. There are several advantages in the CIoU loss function: (1) It can increase the overlap area between the ground truth box and the prediction box. (2) It can minimize the distance between the center points. (3) It can keep the frame height ration consistent.



Aphine sea holly	43	Buttercup	71	Fire lily	40
Anthurium	105	Californian poppy	102	Foxglove	162
Artichoke	78	Camelia	91	Frangipani	166
Azalea	96	Canna lily	82	Fritillary	91
Ball moss	46	Canterbury bells	40	Garden phlox	45
Balloon flower	49	Cape flower	108	Gaura	67
Barbeton daisy	127	Carnation	52	Gazania	78
Bearded iris	54	Cautleya spicata	50	Geranium	114

Figure 2: Oxford 102 Flower dataset, which contains 102 categories of flowers, can be used for image classification studies.

Figure 3: Oxford 102 Flower dataset with annotation, which contains 102 categories of flowers, can be used for flower detection studies.

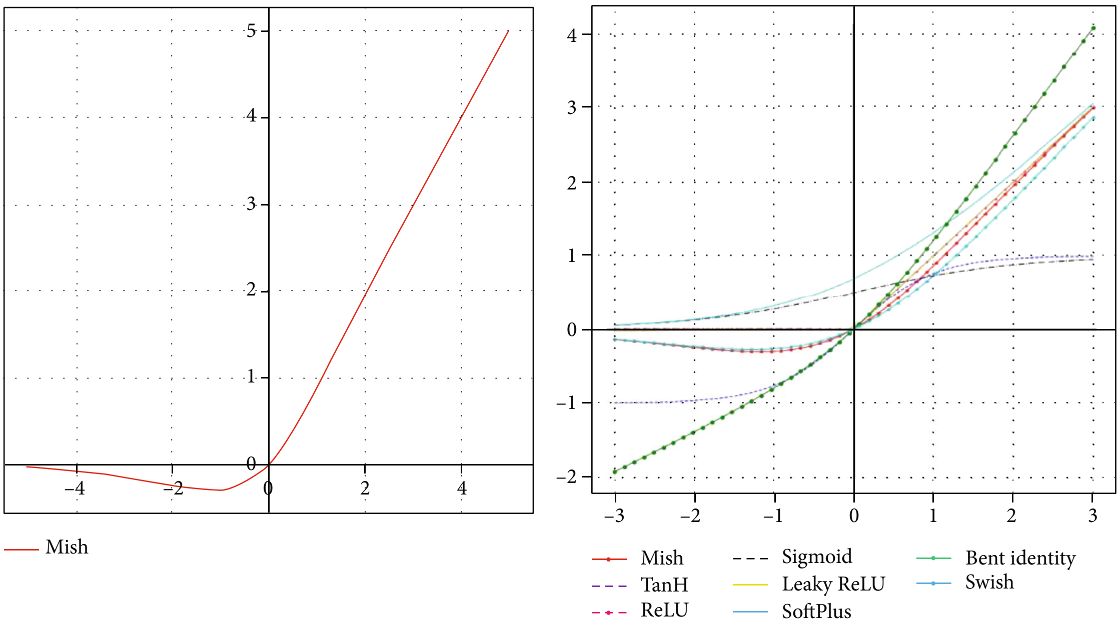


Figure 4: Representation of Mish activation function and its performance: (a) Mish activation function; (b) comparison of the different commonly used activation.

The CIoU can be defined as formula (6) based on formula (4), where α represents a positive trade-off and v is defined in formula (7) to measure the consistency of the aspect ratio added.

$$R_{\mathrm{CIoU}} = \frac{p^2\left(b, b^{gt}\right)}{c^2} + \alpha v, \tag{6}$$

$$v = \frac{4}{\pi^2}\left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h}\right)^2. \tag{7}$$

The loss function of CIoU can be defined as formula (8), and the trade-off parameter α can be defined in formula (9).

$$L_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{p^2\left(b, b^{gt}\right)}{c^2} + \alpha v, \tag{8}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v}. \tag{9}$$

In our work, the DIoU loss function can be applied in NMS (Nonmaximum Suppression) to delete the redundant detection box. Not only the overlapping area is considered but also the distance between the detection box, and the center point of the target box is taken into account, which can effectively avoid the above two loss function mistakes.

2.4. Activation Function. Activation function is a function running on the neuron of the neural network, which is responsible for mapping the input of the neuron to the output. Its function is to increase the nonlinear change of the neural network model. In Figure 4, the Mish activation function and the comparison of different commonly used activation function are shown, in which the original figure can be found in Mish [30].

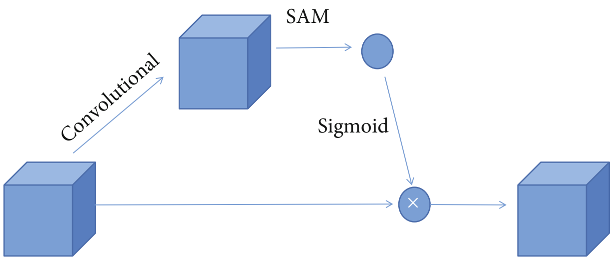


Figure 5: The modified SAM architecture.

In our architecture, the Mish [30] activation function is the activation function we utilized, which replaces Leaky Relu which is a very small constant leak that contained the improved function of Relu with Mish in YOLOv3 [11]. Leaky Relu is a self-regular nonmonotone neural activation function and a smooth activation function allowing better information into the neural network to obtain better accuracy and generalization. It can be defined as formula (10), where in the formula (10), it is shown that $\varsigma(x) = \ln(1 + e^x)$.

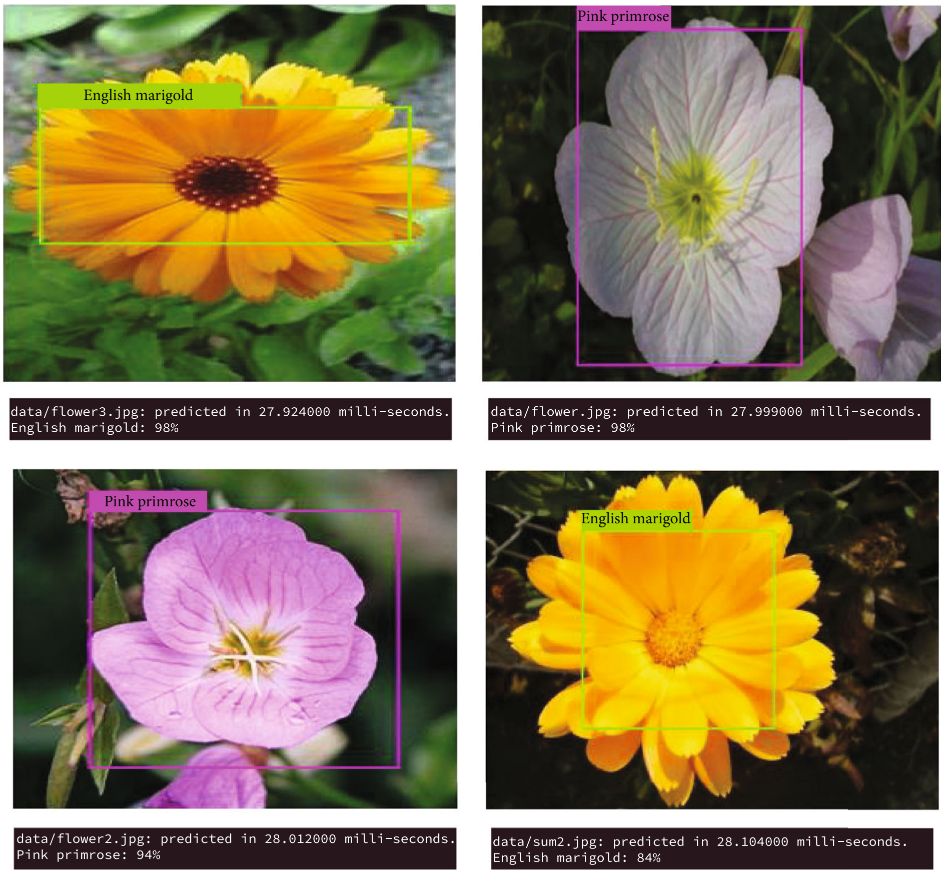
$$f(x) = x \cdot \tanh(\varsigma(x)). \tag{10}$$

It is more accurate and precious than Swish [31] defined as formula (12) and Relu [32] defined as formula (11) when performing on the experiments.

$$f(x) = \max(0, x), \tag{11}$$

$$f(x) = x \cdot \mathrm{sigmoid}(x). \tag{12}$$

2.5. Attention Mechanism. As we all know, attention mechanism has become an important part of the structure of neural



```
data/flower3.jpg: predicted in 27.924000 milli-seconds.
English marigold: 98%
```

```
data/flower.jpg: predicted in 27.999000 milli-seconds.
Pink primrose: 98%
```

```
data/flower2.jpg: predicted in 28.012000 milli-seconds.
Pink primrose: 94%
```

```
data/sum2.jpg: predicted in 28.104000 milli-seconds.
English marigold: 84%
```

Figure 6: The results of flower detection.

networks and has a large number of applications in fields such as natural language processing, statistical learning, and computer vision in the field of artificial intelligence. The attention mechanism in deep learning is similar to the selective visual attention mechanism of human beings in essence. The core goal is to select the information that is more critical to the current task target from the numerous information.

The SE [33] and SAM [34] model is the common attention mechanism used in deep learning networks. The purpose of the SE model is to reweight to the feature channel and only pay attention to which layer on the channel level will have stronger feedback ability, but it cannot reflect the meaning of "attention" on the spatial dimension, while SAM which is the output of the feature map of the channel attention module that is taken as the input feature map of this module is more adaptive to our architecture since it saves a lot of computing resources. In our work, the modified SAM has been utilized, which is shown in Figure 5. It is seen that spacewise attention is used instead of pointwise attention in it. Besides, PAN is replaced with concatenation connections.

3. Results and Discussion

We describe the results of our flower detection in Figure 6, which shows the result of flower detection containing category and degree of confidence added in each category. The confidence levels are 98%, 98%, 94%, and 84%, respectively, in the test results, which achieved the desired effect. It is demonstrated that the flower detection can be achieved at a high standard which the accuracy has matched the speed. The GPU we used are 2 Titan Xp. The basic requirements for a camera on a mobile device are based on HUAWEI P20 Pro, and the processor is HiSilicon Kirin 970. When applied in a smart garden in AIoT, the basic mobile network relied on a 5G network, which is a high-speed, low-delay, low-power consumption and ubiquitous network.

4. Conclusions

In conclusion, it seems that flower detection based on the state-of-the-art method is very considerable. Although widely accepted, it suffers from some limitations due to the flower detection algorithms that have not been integrated into the mobile device completely to perform the final application results of our flower detection to enhance user's human-computer interaction experience in a smart garden combining with virtual reality. It is a tendency for the flower detection proposed in our paper combined with virtual reality. In a smart garden, it can be realized that flowers can be arranged intelligently not depending on people just on the smart phone or other mobile devices. Furthermore, intelligent gardening can come true which can reduce the cost of human and finance and it just relies on the human-computer interaction. In the future work, we will pay much importance to it to accomplish the goal in our future work.

Data Availability

The data we used is available, which is named Oxford 102 Flowers that part of them can be accessed and downloaded to perform the flower classification research and other experiments. And in our work, our dataset was labelled based on this dataset with annotation. So part of them with annotation is available to you from the corresponding author upon request (8528750@qq.com).

Conflicts of Interest

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

Acknowledgments

Our work is supported by the National Natural Science Foundation of China, Project No.: 61503082, Project name: Research on Large-Scale Ontology Matching Based on Evolutionary Algorithm.

References

[1] A. D. Aggelopoulou, D. Bochtis, S. Fountas, K. C. Swain, T. A. Gemtos, and G. D. Nanos, "Yield prediction in apple orchards based on image processing," *Precision Agriculture*, vol. 12, no. 3, pp. 448–456, 2011.

[2] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.

[3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyound bags of features: spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, June 2006.

[4] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pp. 1447–1454, New York, NY, USA, June 2006.

[6] J. Lu, W. S. Lee, H. Gan, and X. Hu, "Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis," *Biosystems Engineering*, vol. 171, pp. 78–90, 2018.

[7] L. Haozhou, C. Lipeng, M. Longtao, G. Zongbin, and C. Yongjei, "A recognition method of kiwifruit flowers based on K-means clustering," *Journal of Agricultural Mechanization Research*, vol. 42, no. 2, pp. 22–26, 2020.

[8] 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, 2017.

[9] R. Girshick, "Fast r-cnn," *2015 IEEE International Conference on Computer Vision (ICCV) IEEE*, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2014.

[11] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, http://arxiv.org/abs/1804.02767.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[14] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Computer Vision – ECCV 2016*, 2016.

[15] P. A. Dias, A. Tabb, and H. Medeiros, "Multispecies fruit flower detection using a refined semantic segmentation network," *IEEE Robotics & Automation Letters*, vol. 3, no. 4, pp. 3003–3010, 2018.

[16] B. Shree, R. B. Divya, and N. S. Rani, "Fruit detection from images and displaying its nutrition value using deep Alex network," in *Soft Computing and Signal Processing. Advances in Intelligent Systems and Computing, vol 898*, J. Wang, G. Reddy, V. Prasad, and V. Reddy, Eds., Springer, Singapore, 2019.

[17] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. C. Grappadelli, and L. Manfrini, "Comparing deep-learning networks for apple fruit detection to classical hard-coded algorithms," *Acta Horticulturae*, vol. 1279, no. 1279, pp. 209–216, 2020.

[18] S. Bargoti and J. Underwood, "Image segmentation for fruit detection and yield estimation in apple orchards," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1039–1060, 2017.

[19] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Computer Networks*, vol. 168, article 107036, 2020.

[20] J. Lim, H. S. Ahn, M. Nejati, J. Bell, H. Williams, and M. D. BA, *Deep neural network based real-time kiwi fruit flower detection in an orchard environment*, 2020.

[21] P. Lin, W. S. Lee, Y. M. Chen, N. Peres, and C. Fraisse, "A deep-level region-based visual representation architecture for detecting strawberry flowers in an outdoor field," *Precision Agriculture*, vol. 21, no. 2, pp. 387–402, 2020.

[22] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, Singapore, Singapore, June 2017.

[23] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417–426, 2019.

[24] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of ' MangoYOLO '," *Precision Agriculture*, vol. 20, no. 6, pp. 1107–1135, 2019.

[25] J. Zhao and J. Qu, "A detection method for tomato fruit common physiological diseases based on YOLOv2," in *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, Qingdao, China, China, August 2019.

[26] H. X. Peng, B. Huang, Y. Y. Shao et al., "General improved SSD model for picking object recognition of multiple fruits in natural environment," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 34, no. 16, pp. 155–162, 2018.

[27] Q. Liang, W. Zhu, J. Long, Y. Wang, W. Sun, and W. Wu, "A real-time detection framework for on-tree mango based on SSD network," in *International Conference on Intelligent Robotics & Applications*, Springer, Cham, 2018.

[28] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop)*, Seattle, WA, USA, USA, June 2020.

[29] S. Liu, Q. Lu, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8759–8768, Salt Lake City, UT, USA, June 2018.

[30] D. Misra, "Mish: a self regularized non-monotonic neural activation function," 2019, http://arxiv.org/abs/1908.08681v2.

[31] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, http://arxiv.org/abs/1710.05941.

[32] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," *Proceedings of International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

[33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8873366, 14 pages
https://doi.org/10.1155/2020/8873366

Research Article

Quasiconformal Mapping Kernel Machine Learning-Based Intelligent Hyperspectral Data Classification for Internet Information Retrieval

Jing Liu and Yulong Qiao

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Yulong Qiao; qiaoyulonghrbeu@126.com

Received 2 May 2020; Revised 13 July 2020; Accepted 20 August 2020; Published 1 September 2020

Academic Editor: Pei-Wei Tsai

Intelligent internet data mining is an important application of AIoT (Artificial Intelligence of Things), and it is necessary to construct large training samples with the data from the internet, including images, videos, and other information. Among them, a hyperspectral database is also necessary for image processing and machine learning. The internet environment provides abundant hyperspectral data resources, but the hyperspectral data have no class labels and no so high value for applications. So, it is important to label the class information for these hyperspectral data through machine learning-based classification. In this paper, we present a quasiconformal mapping kernel machine learning-based intelligent hyperspectral data classification algorithm for internet-based hyperspectral data retrieval. The contributions include three points: the quasiconformal mapping-based multiple kernel learning network framework is proposed for hyperspectral data classification, the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning, and the objective function of measuring the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. Experiments show that the proposed scheme is effective for hyperspectral image classification and retrieval.

1. Introduction

Intelligent data mining is an important issue of AIoT (Artificial Intelligence of Things), and with the development of machine learning, a large training dataset is necessary for the learning tasks, including images and videos. Among these applications, hyperspectral databases are also very necessary for hyperspectral image processing and machine learning. So, internet environment-based hyperspectral data retrieval is an important issue of AIoT, and it is also an effective way to create a large-scale hyperspectral training database for some applications. The internet environment provides abundant hyperspectral data resources but is included in other complex data. Moreover, the hyperspectral data have no class labels and no detail class knowledge information for these data. So, the hyperspectral data have no high value without class information. So, further intelligent hyperspectral data-based machine learning is necessary through internet-based data retrieval. Intelligent hyperspectral data retrieval under the internet environment combination is the application of AI (Artificial Intelligence) and IoT (Internet of Things).

Hyperspectral data-based machine learning is a feasible and effective method to extract the features for image retrieval. The machine learning methods are divided into unsupervised and supervised learning. The unsupervised learning includes multidimensional scaling, NMF, ICA, neighborhood preserving embedding, Locality Preserving Projection (LPP) [1], and other computing methods [2], and for the supervised learning, generalized discriminant analysis [3], uncorrelated discriminant vector analysis [4], and some acceleration algorithm [5, 6]. In recent years, the kernel-based machine learning algorithms were presented for the feature extraction; this paper proposes an improved kernel function supervised kernel-based LPP, local structure supervised feature extraction [7], kernel subspace LDA [8], kernel MSE [9], and quasiconformal mapping-based kernel machine [10]. Many kernel learning methods are proposed to improve the accuracy performance for the practical kernel learning system, for example, sparse multiple kernel learning [11], large scale multiple kernel learning [12], and Lp-norm multiple kernel learning [13]. With the development of the deep learning theory, the framework of the deep learning-based multikernel machine is an effective framework, and the learning method has been widely used in image analysis [14, 15], image annotation [16], image classification [17], image segmentation [18], and anomaly detection [19]. Researchers proposed deep kernel learning, namely, LMKL [20], and in the other work, the estimated value of missing an error is adjusted instead of the double objective function [21]. And recent research includes machine learning-based image processing [22, 23] under the internet environment for the application of AI and IoT. Mathematically, it has been proved that multilayer can improve the richness of representation, and the researchers combine the support vector machine and multiple classifiers and use an adaptive back propagation algorithm to update coefficients and weights [20, 26, 27].

The kernel-based machine learning on the hyperspectral data is proposed to retrieve the spectral data in the internet environment. In the algorithm, we proposed the quasiconformal mapping-based kernel learning for hyperspectral data classification for data retrieval in the internet environment. And the Mahalanobis distance kernel function is applied to extract the nonlinear feature, with higher discriminative ability than Euclidean distance-based kernel function learning. The objective function of quasiconformal kernel learning created with the Fisher criterion is proposed to seek the optimal parameters of the quasiconformal mapping projection. The proposed scheme is effective to hyperspectral image retrieval under the internet environment.

2. Proposed Algorithm

2.1. Motivation and Framework. Intelligent hyperspectral retrieval extracts the features of the spectrum through sensing data processing and analysis. Motivated by the fact that kernel machine-based spectrum learning is effective to the nonlinear classification, we present a framework of quasiconformal mapping-based multiple kernel learning with Mahalanobis distance kernel functions. The contributions include three points: the quasiconformal mapping-based multiple kernel learning network framework is proposed for hyperspectral data classification; the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning; and the objective function of measuring the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. The performance is improved with two facts: one is to optimize the data structure in the kernel empirical space with quasiconformal mapping and second is to improve the discriminant ability with Mahalanobis distance-based kernel. The proposed algorithm has the highly effective performance on characterizing the data through solving complex visual learning tasks. The learning framework of hyperspectral image classification is presented in Figure 1.

2.2. Quasiconformal Kernel Mapping Learning. Kernel-based learning is used in data classification, with "empirical kernel map." Suppose that matrix $K = [k_{ij}]_{m\times m}$, K is decomposed as

$$K_{m\times m} = P_{m\times r}\Lambda_{r\times r}P^T_{r\times m}, \tag{1}$$

where Λ is a diagonal matrix with r positive eigenvalues of K, P. The mapping is the empirical kernel map $\Phi^e_r \in R^r$ as

$$\begin{gathered}\Phi^e_r : \chi \longrightarrow R^r,\\ x \longrightarrow \Lambda^{-1/2}P^T(k(x,x_1), k(x,x_2), \cdots, k(x,x_m)).\end{gathered} \tag{2}$$

Different kernels have different abilities on classification, and the kernel function based on feature similarity refers that features exist in the form of distance, that is, similarity in a function expression. For sample features $x, y \in R^N$, if the Euclidean distance between the two is $d_U(x,y) = \|x-y\|$, the general form of such a kernel function is

$$k^U(x,y) = f(d_U(x,y)). \tag{3}$$

The RBF kernel is the most typical representative of this kind of kernel function, in addition to the negative distance kernel, the logarithmic function kernel, and the Bn spline kernel. For this type of kernel function, the Euclidean distance $d_U(x,y)$ can be simply replaced with the Mahalanobis distance $d_M(x,y) = \sqrt{(x-y)^T\mathbf{M}(x-y)}$, that is,

$$k^U_M(x,y) = f(d_M(x,y)). \tag{4}$$

In particular, the expression for a typical Mahalanobis distance RBF kernel function is

$$k^{\mathrm{RBF}}_M(x,y) = \exp\left(-\frac{1}{2\sigma^2}(x-y)^T\mathbf{M}(x-y)\right). \tag{5}$$

Similar to the above expression, given the inner product $I(x,y) = x\cdot y$ of two features, the general form of the kernel function based on the feature inner product is

$$k^I(x,y) = f(I(x,y)). \tag{6}$$

The polynomial kernel is the most typical representative of this kind of kernel function, in addition to the Sigmoid kernel. For this type of kernel function, a transformation is required when expanding to the Mahalanobis distance kernel function. Considering that in the case of Euclidean distance, the inner product of the two features satisfies

$$I(x,y) = x\cdot y = \frac{1}{2}\cdot\left(\|x-0\|^2 + \|y-0\|^2 - \|x-y\|^2\right). \tag{7}$$

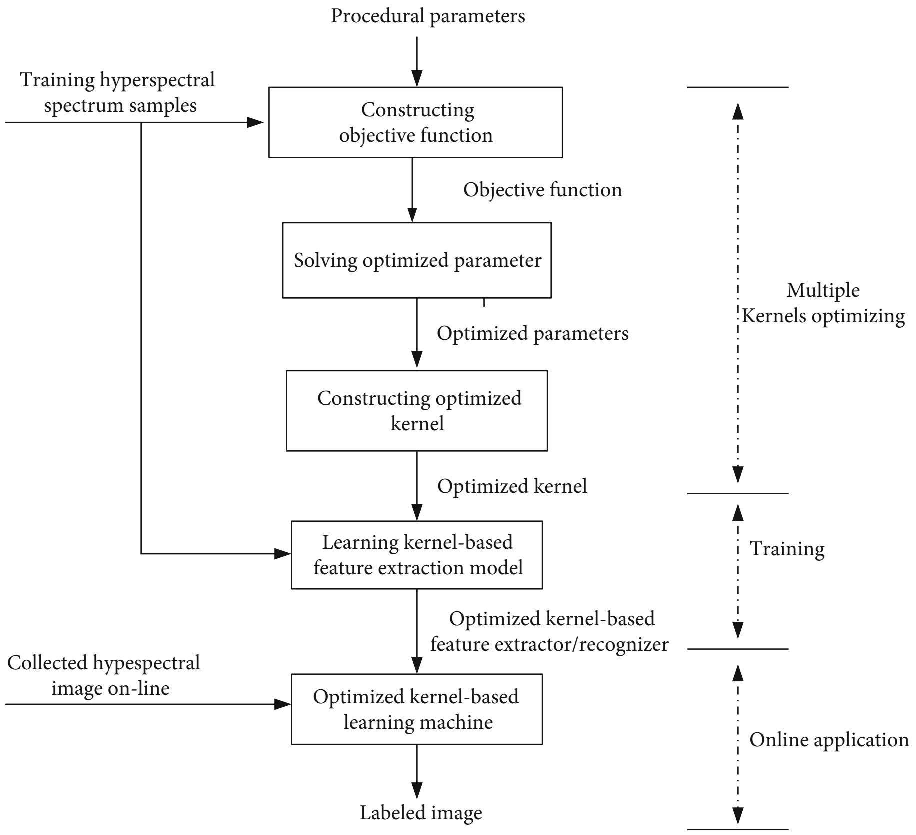

Procedural parameters
Training hyperspectral spectrum samples
Constructing objective function
Objective function
Solving optimized parameter
Optimized parameters
Constructing optimized kernel
Optimized kernel
Learning kernel-based feature extraction model
Optimized kernel-based feature extractor/recognizer
Collected hypespectral image on-line
Optimized kernel-based learning machine
Labeled image
Multiple Kernels optimizing
Training
Online application

Figure 1: Algorithm framework.

Therefore, the Mahalanobis distance of the inner product can be written as

$$d_M^I(x, y) = \frac{1}{2}\left(d_M^2(x, 0) + d_M^2(y, 0) - d_M^2(x, y)\right). \quad (8)$$

In this way, the Mahalanobis distance of the inner product can be obtained by calculating the Mahalanobis distance between the feature and the origin 0 and the feature. Similarly, the Mahalanobis distance of the inner product can be obtained:

$$k_M^I(x, y) = f\left(d_M^I(x, y)\right). \quad (9)$$

In particular, the expression for a typical Mahalanobis distance polynomial kernel function is

$$k_M^{\mathrm{POL}}(x, y) = \left(\frac{1}{2}\left(x^T\mathbf{M}x + y^T\mathbf{M}y - (x - y)^T\mathbf{M}(x - y)\right) + c\right)^d. \quad (10)$$

In this paper, we introduce the quasiconformal kernel, $k_q(x, y)$, as follows:

$$k_q(\mathbf{x}, \mathbf{y}) = f(\mathbf{x})f(\mathbf{y})k(\mathbf{x}, \mathbf{y}), \quad (11)$$

where $\mathbf{x}$, $\mathbf{y}$ are the sample vectors and $f(\mathbf{x})$ is

$$f(\mathbf{x}) = b_0 + \sum_{n=1}^{N_{\mathrm{XV}}} b_n e(\mathbf{x}, \tilde{\mathbf{x}}_{\mathbf{n}}), \quad (12)$$

where $e(\mathbf{x}, \tilde{\mathbf{x}}_{\mathbf{n}}) = e^{-\delta\|\mathbf{x}-\mathbf{x}_{\mathbf{n}}\|^2}$, δ is a free parameter, $\tilde{x}_n$ is called the “expansion vectors (XVs)”, N_{XV} is the number of XVs, and b_n $(n = 0, 1, 2, \cdots, N_{\mathrm{XV}_s})$ is the “expansion coefficients” associated with $\tilde{\mathbf{x}}_{\mathbf{n}}$ $(n = 0, 1, 2, \cdots, N_{\mathrm{XV}_s})$.

On the multiple kernels, the quasiconformal mapping kernel is described as

$$k\left(\mathbf{x}, \mathbf{x}'\right) = f(\mathbf{x}) \sum_{i=1}^{m} a_i k_{0,i}\left(\mathbf{x}, \mathbf{x}'\right) f\left(\mathbf{x}'\right), \quad (13)$$

where $\mathbf{x}, \mathbf{x}' \in^p$, $k_{0,i}(\mathbf{x}, \mathbf{x}')$ is the ith basic kernel, m is the number of basic kernels for combination, $a_i \geq 0$ is the weight for the ith basic kernel function, and $q(\cdot)$ is the factor function defined by

$$f(\mathbf{x}) = b_0 + \sum_{i=1}^{n} b_i k_0(\mathbf{x}, \mathbf{a}_i), \quad (14)$$

where $k_0(\mathbf{x}, \mathbf{a}_i) = e^{-\gamma\|\mathbf{x}-\mathbf{a}_i\|^2}$, $\mathbf{a}_i \in \mathbb{R}^d\{\mathbf{a}_i, ⊠ i = 1, 2, \cdots, n\}$ are selected by the training samples, and b_i is the coefficient for the combination. $k(\mathbf{x}, \mathbf{x}')$ satisfies the Mercer condition, $k(\mathbf{x}$

$, \mathbf{x}')$ is rewritten as $k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{m} d_i [f(\mathbf{x}) k_{0,i}(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')]$ of optimized transformation $k_{0,i}(\mathbf{x}, \mathbf{x}')$, and $q(\mathbf{x}) k_{0,i}(\mathbf{x}, \mathbf{x}') q(\mathbf{x}'$ $)$ is the linear combination of kernels.

Given $\mathbf{d} = [d_1, d_2, \cdots, d_m]$, $\alpha = [\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_n]$, and the optimized multiple kernels $k(x, x') = K_f(k_{0,i}(x, x'), \mathbf{d}, \boldsymbol{\alpha})$, $\mathbf{d}$, $\boldsymbol{\alpha}$ are to be optimized for the classification task. Finally, the joint convex formulation can be formed as

$$\begin{gathered} \max_{\mathbf{d},\boldsymbol{\alpha}} F_c(K_0, \mathbf{d}, \boldsymbol{\alpha}) \\ \text{Subject to } \|\mathbf{d}\| = 1, \|\boldsymbol{\alpha}\| = 1. \end{gathered} \tag{15}$$

$F_c(\cdot)$ measures the class discriminative ability. $\mathbf{d}$, $\boldsymbol{\alpha}$ can be solved in two stages, respectively. In the first stage, the centered kernel alignment [28] is applied to define the objective optimization function; in the second stage, Fisher-based and Margin-based optimization function is used to solve $\boldsymbol{\alpha}$.

Step 1. Optimize the weight vector of kernels.

In multikernel learning, the crucial step is to select the adaptive weights of multiple kernels. The weight $\mathbf{d} = [d_1, d_2$ $, \cdots, d_m]$ is to solve with centered kernel alignment [22]:

$$\begin{gathered} \max \; O_c\left(K_0^{(C)}, K^*\right) \\ \text{Subject to } K_0^{(C)} = \sum_{i=1}^{m} d_i K_{0,i}^{(C)}, \operatorname{tr}(K_0) = 1, d_i \geq 0, \forall i, \end{gathered} \tag{16}$$

where $O_c(K_0^{(C)}, K^*)$ is the objective function,

$$K^*\left(x, x'\right) = \begin{cases} 1, & \text{if } y = y', \\ \dfrac{-1}{c-1}, & \text{if } y \neq y', \end{cases} \tag{17}$$

is the optimal kernel, and tr is the matrix trace. $K_0^{(C)} = [I - 11^T/m] K_0 [I - 11^T/m]$ is the centered kernel matrix of K_0, I is the identity matrix, and $\mathbf{1}$ is a vector with all entries equal to 1. Accordingly, $K_{0,i}^{(C)} = [I - 11^T/m] K_{0,i} [I - 11^T/m]$ is the center of $K_{0,i}$, $i = 1, 2, \cdots, m$. The objective function $O_c(K_0^{(C)}$ $, K^*) = \langle K_0^{(C)}, K^* \rangle_F / \|K_C^*\|_F \|K_0^{(C)}\|_F$, K_C^* is the centered kernel matrix of K^*, where $\langle \cdot, \cdot \rangle_F$ is the Frobenius norm between two matrices, i.e., $\langle D, E \rangle_F = \sum_{i=1}^{m} \sum_{j=1}^{m} d_{ij} e_{ij} = \operatorname{tr}(DE^T)$. Suppose $\mathbf{d} = [d_1, d_2, \cdots, d_m]$, $\mathbf{d}^*$ is solved with the quadratic programming (QP) problem:

$$\begin{gathered} \min \; v^T \mathbf{T} v - 2 v^T \boldsymbol{\eta} \\ \text{Subject to } vi \geq 0, \forall i, \end{gathered} \tag{18}$$

where $\mathbf{d}^* = v^* / \|\mathbf{d}^*\|_2$, $\boldsymbol{\eta} = [\langle K_{0,1}^{(C)}, K^* \rangle_F \langle K_{0,2}^{(C)}, K^* \rangle_F, \cdots, \langle K_{0,m}^{(C)}, K^* \rangle_{\mathbf{F}}]^T$, and $\mathbf{T}$ is defined by $\mathbf{T}_{ij} = \langle K_{0,1}^{(C)}, K_{0,j}^{(C)} \rangle_F$, $i, j = 1, 2, \cdots, m$. Based on this centered kernel alignment, the optimization problem can be transformed into a QP problem, which can be effectively solved with OPTI toolbox [28].

Step 2. Optimize the coefficients of kernels.

The coefficients $\boldsymbol{\alpha} = [b_1, b_2, \cdots, b_n]$ is solved based on the Fisher criterion and $J_{\text{Fisher}}(\boldsymbol{\alpha})$ is defined as

$$J_{\text{Fisher}}(\boldsymbol{\alpha}) = . \tag{19}$$

Then, $J_{\text{Fisher}}(\boldsymbol{\alpha}) / \partial \boldsymbol{\alpha} = \mathbf{2} / \mathbf{J}_2^2 (\mathbf{J}_2 \mathbf{E}^\mathbf{T} \mathbf{B}_0 \mathbf{E} - \mathbf{J}_1 \mathbf{E}^\mathbf{T} \mathbf{W}_0 \mathbf{E}) \boldsymbol{\alpha}$, where J_{Fisher} is solved with the eigenvalue problem of $(\mathbf{E}^\mathbf{T} \mathbf{W}_0 \mathbf{E})^{-1}($ $\mathbf{E}^\mathbf{T} \mathbf{B}_0 \mathbf{E})$ and $\boldsymbol{\alpha}$. The matrix $(\mathbf{E}^\mathbf{T} \mathbf{W}_0 \mathbf{E})^{-1} (\mathbf{E}^\mathbf{T} \mathbf{B}_0 \mathbf{E})$ maybe is not symmetrical or the matrix E^TWE is singular; $\boldsymbol{\alpha}$ is solved as

$$\boldsymbol{\alpha}^{(n+1)} = \boldsymbol{\alpha}^{(n)} + \varepsilon \left(\frac{1}{J_2} \mathbf{E}^\mathbf{T} \mathbf{B}_0 \mathbf{E} - \frac{J_{\text{Fisher}}}{J_2} \mathbf{E}^\mathbf{T} \mathbf{W}_0 \mathbf{E} \right) \boldsymbol{\alpha}^{(n)}, \tag{20}$$

where ε is the learning rate, $\varepsilon(n) = \varepsilon_0 (1 - n/N)$ is the learning rate of the nth iteration, ε_0 is the initial learning rate, and N is the number of the total iterations.

2.3. Metric Similarity-Based Learning

2.3.1. Similar/Dissimilar Function. Suppose the initial distance metric function $d(x, y)$ (generally using Euclidean distance), the purpose of metric learning is to construct a new distance $\tilde{d}(x, y)$ based on some prior information, which is more consistent with the description of the sample features than the initial distance metric. In order to achieve this, the new metric function $\tilde{d}(x, y)$ can be converted into $d(f(x), f$ $(y))$, that is, by defining a mapping f and using the original distance metric function to calculate a new metric function, thereby converting the metric learning problem into a learning mapping function f problem.

Given a sample set $X \in R^N$, x, y, z is a sample in the sample set; if a function $d: X \times X \longrightarrow R^+$defined in the vector space satisfies the following properties, then d is called a distance measure function: symmetry: $d(x, y) = d(y, x)$, nonnegative: $d(x, y) \geq 0$, distinguishability: $d(x, y) = 0 \Leftrightarrow x = y$, and triangle inequality: $d(x, y) + d(y, z) \geq d(x, z)$. In the metric function that satisfies the above properties, the Euclidean distance is the most common distance metric function, which measures the absolute distance between spatial sample points, which is defined as

$$d(x, y) = \sqrt{(x - y)^T (x - y)}. \tag{21}$$

The Euclidean distance is characterized by simple calculation, but since the absolute distance measured is directly related to the coordinates of the position of each point, the adaptability to the data is poor in terms of feature scale and coupling degree between features. Cosine similarity mainly measures the consistency of direction between two vectors, which is defined as

$$d(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|}. \tag{22}$$

Compared with the Euclidean distance, the cosine similarity measures the angle between the space vectors, which reflects the difference in the direction of the vector and is insensitive to the absolute value.

The Minkowski distance is a general expression of a class of distance functions. Given two N-dimensional vectors: $x = \{x_1, x_2, \cdots, x_N\}$ and $y = \{y_1, y_2, \cdots, y_N\}$, the Minkowski distance is defined as

$$d(x, y) = \left(\sum_{s=1}^{N} (x_s - y_s)^P \right)^{1/P}, \tag{23}$$

where P takes a different value; the distance is derived as a different type of distance. When $P = 1$, the distance is called Manhattan distance, that is, $d(x, y) = (\sum_{s=1}^{N} |x_s - y_s|)$. This distance is used to indicate the absolute wheelbase sum of the two points on the standard coordinate system. When $P = 2$, the distance is called the Euclidean distance. When $P \longrightarrow \infty$, the Minkowski distance is called the Chebyshev distance, that is, $d(x, y) = \max_s (|x_s - y_s|)$, which represents the maximum value of the numerical difference between the coordinates. Similar to the Euclidean distance, the Minkowski distance is still related to the dimension of the feature, and the correlation between the features is not considered.

The Mahalanobis distance was proposed by P.C. Mahalanobis. It is defined as

$$d(x, y) = \sqrt{(x - y)^T \mathbf{S}^{-1} (x - y)} = \sqrt{(x - y)^T \mathbf{M} (x - y)}, \tag{24}$$

where S is the data correlation matrix and $\mathbf{M} = \mathbf{S}^{-1}$ is the Mahalanobis matrix. When M is a unit array, the Mahalanobis distance degenerates into a Euclidean distance, indicating that the Euclidean distance is a special case of the Mahalanobis distance. The greatest advantage of the Mahalanobis distance is the ability to remove the coupling between various features and is scale-invariant.

As can be seen from the above definition, the traditional distance measurement function is fixed in form and can be calculated directly according to the formula without a process of "learning" to the sample. Obviously, this approach does not meet the diverse task requirements nor does it make full use of the information contained in the sample features. Therefore, it is necessary to construct a suitable distance metric function by learning the multidimensional information provided by the sample features for specific problems, so as to provide the best expression of feature similarity.

2.3.2. Distance Measure Function. Linear metric learning refers to obtaining a new metric function by linear transformation; that is, the form of the mapping function f is $f(x) = \mathbf{A}^T x$, where A is a projection matrix. The purpose of learning is to be able to find a suitable matrix A. In the case where the initial distance measure is Euclidean distance, the new measure function is

$$\tilde{d}(x, y) = \sqrt{\left(\mathbf{A}^T x - \mathbf{A}^T y\right)^T \left(\mathbf{A}^T x - \mathbf{A}^T y\right)}. \tag{25}$$

At this time, for the real matrix A, if $\mathbf{M} = \mathbf{A}\mathbf{A}^T$, M is a semipositive symmetric matrix. The formula can be written as follows:

$$\begin{aligned} \tilde{d}(x, y) &= \sqrt{\left(\mathbf{A}^T x - \mathbf{A}^T y\right)^T \left(\mathbf{A}^T x - \mathbf{A}^T y\right)} \\ &= \sqrt{(x - y)^T \mathbf{A}\mathbf{A}^T (x - y)} \\ &= \sqrt{(x - y)^T \mathbf{M} (x - y)}. \end{aligned} \tag{26}$$

It can be seen from the above equation that under the effect of matrix A, the sample is linearly mapped to a new feature space. When A is a unit matrix, the distance is the Euclidean distance. When A is a diagonal matrix, it means that the original space is scaled, which is equivalent to weighting the samples. When A is an orthogonal matrix, it means that the original space is rotated and transformed. When A is a normal square matrix, it means that the original space is simultaneously scaled and rotated. When A is not a square matrix, it is equivalent to a dimensionality reduction operation in addition to rotation and scale transformation.

The above formula is similar to the Mahalanobis distance metric in form, but the traditional Mahalanobis distance metric uses the inverse of the covariance matrix as the Mahalanobis matrix. M in the above equation expands to the semipositive array, representing a more general Mahalanobis matrix, so linear metric learning is also known as Mahalanobis metric learning. From the perspective of using sample information, the traditional Mahalanobis matrix only uses the internal structure of the data, focusing on describing the distribution properties of the data, while the Mahalanobis matrix obtained by the metric learning makes full use of the relationship between the feature and the category label and focuses on features that adequately reflect sample class differences to achieve a better metric function. In general, a metric learning problem can be transformed into a constrained optimization problem:

$$\begin{gathered} \min_{\mathbf{M}} \mathscr{L}_X(\mathbf{M}) + \lambda r(\mathbf{M}) \\ \text{s.t.} c_X(\mathbf{M}), \end{gathered} \tag{27}$$

where $\mathscr{L}_X(\mathbf{M})$ represents the loss function on the training set X; $r(\mathbf{M})$ is a regularization term, which is used to correct the overfitting; and λ is the preset regularization factor, which is used to adjust the influence degree of the regularization term in the training process. $c_X(\mathbf{M})$ is the constraint on the training set. Different learning algorithms can be derived depending on the difference of loss function, the regularization term, and the constraints.

2.4. Similar/Dissimilar Learning Criterions. Two popular criterions, Fisher criterion and large margin nearest neighbor criterion, are used in similar-/dissimilar-based learning.

2.4.1. Fisher Criterion. The method is based on the pairwise constraint information provided by the sample as a priori information to minimize the similar sample pairs and control the distance between the nonsimilar sample pairs as the basic idea to construct a convex optimization problem and achieve the purpose of learning the Mahalanobis distance matrix.

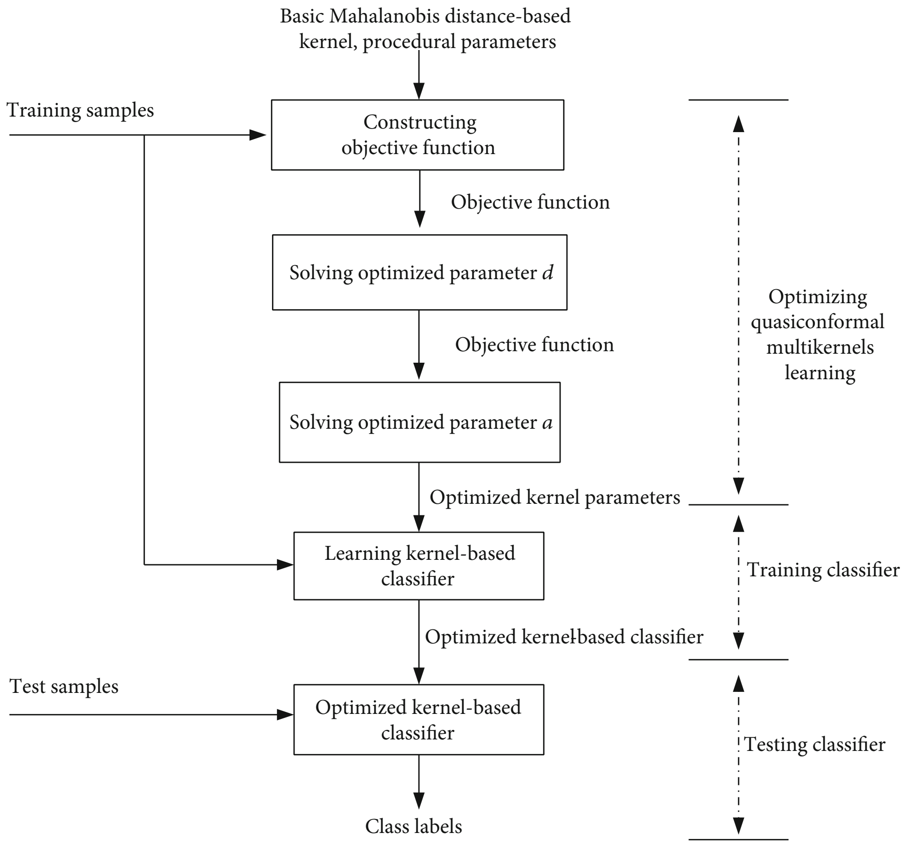

Basic Mahalanobis distance-based kernel, procedural parameters
Training samples
Constructing objective function
Objective function
Solving optimized parameter d
Objective function
Solving optimized parameter a
Optimized kernel parameters
Learning kernel-based classifier
Optimized kernel-based classifier
Test samples
Optimized kernel-based classifier
Class labels
Optimizing quasiconformal multikernels learning
Training classifier
Testing classifier

Figure 2: Data preprocessing procedure.

First, given a sample set $X : \{x_i\}_{i=1}^{M} \subset \mathbb{R}^N$, depending on whether the sample pairs belong to the same category, two constraint sets can be obtained: a homogeneous constraint set W and a nonlike constraint set B. If the categories of the sample pairs are the same (similar), then the sample pair belongs to the set W, and if the categories of the sample pairs are not the same (similar), the sample pair belongs to the set B. If the Euclidean distance is used as the initial distance metric, considering that the postlearning metric can make the distance between the pairs of similar samples as small as possible, a convex optimization problem can be constructed:

$$\begin{aligned}&\min_{\mathbf{M}\geq 0} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in W} d_M^2(x_i, x_j)\\ &\text{s.t.} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in B} d_M(x_i, x_j) \geq 1,\end{aligned} \tag{28}$$

where $d_M^2(x_i, x_j) = (x_i - x_j)^T \mathbf{M}(x_i - x_j)$ and M are semipositive definite matrices. The constraint is added mainly to remove the trivial solution of $M = 0$. In the specific solution, an iterative update method can be used to solve. In each iteration, the mature Newton downhill method is used to perform the gradient descent process to obtain an updated Mahalanobis matrix. The matrix is then iteratively mapped onto the constraint set. Although the algorithm is relatively simple in implementation, the corresponding calculation amount is large in the case of large data size because the algorithm needs to construct all pairs of similar samples and nonsimilar samples in the whole dataset. And the convergence speed of the algorithm is also slow.

First, by introducing the projection matrix A, the distance between the pair of points becomes

$$d_M(x_i, x_j) = \sqrt{\left(\mathbf{A}^T x_i - \mathbf{A}^T x_j\right)^T \left(\mathbf{A}^T x_i - \mathbf{A}^T x_j\right)}. \tag{29}$$

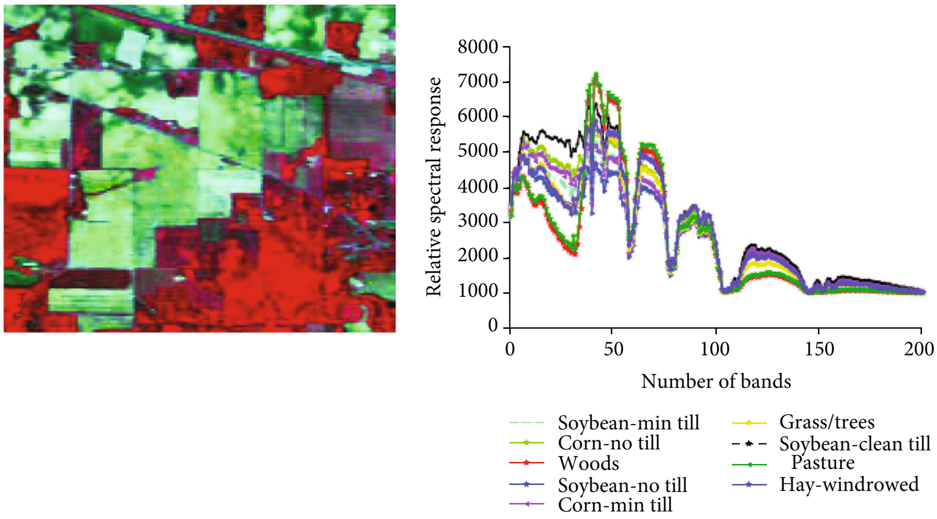
Considering the constraint set W, after the action of the projection matrix A, the sum of the squares of the distances between all pairs of points is

$$d_W = \sum_{(x_i,x_j)\in W} \left(\mathbf{A}^T x_i - \mathbf{A}^T x_j\right)^T \left(\mathbf{A}^T x_i - \mathbf{A}^T x_j\right). \tag{30}$$

The sum of the squares of the distances between all pairs of points in constraint set B can be calculated as

$$d_B = \sum_{(x_i,x_j)\in B} \left(\mathbf{A}^T x_i - \mathbf{A}^T x_j\right)^T \left(\mathbf{A}^T x_i - \mathbf{A}^T x_j\right) = \operatorname{tr}\left(\mathbf{A}^T \mathbf{S}_B \mathbf{A}\right). \tag{31}$$

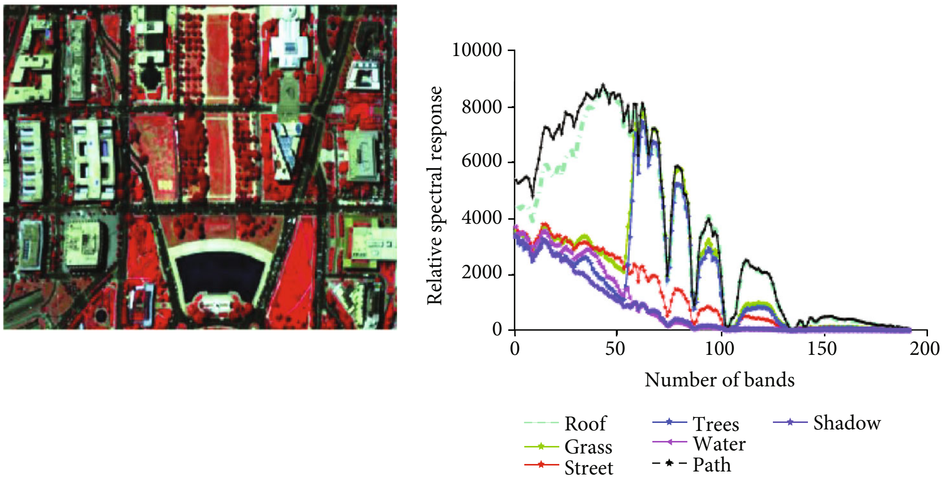
Considering an excellent projection matrix A, it should maximize the distance between the samples in the constraint set B and reduce the distance between the samples in the



(a) Three-band false color composite

(b) Spectral signatures

Figure 3: One example of Indian Pines data.



(a) Three-band false color composite

(b) Spectral signatures

Figure 4: One example from D.C. Mall data.

Table 1: Performance on the Indian Pines data (%).

Class	1	2	3	4	5	6	7	8	9	10	11	12
SVC (polynomial)	49.32	58.73	96.45	39.26	65.82	93.65	62.92	85.33	99.01	65.83	72.33	58.41
SVC (Gaussian)	78.02	73.65	99.16	76.92	80.52	97.12	79.78	89.80	99.79	83.64	86.04	80.74
KSRC (polynomial)	51.83	59.68	96.13	49.12	78.56	93.87	62.83	84.72	98.23	67.57	75.27	60.77
KSRC (Gaussian)	77.84	76.47	99.12	75.56	79.06	97.42	82.71	88.73	98.69	83.93	86.38	81.12
SVC (Q-kernel)	78.32	80.49	99.97	82.55	90.25	99.29	82.70	98.57	99.89	86.85	90.26	84.46
QMK (Q-kernel)	79.45	83.56	99.82	83.45	92.63	99.41	82.87	98.30	99.24	87.86	91.02	85.66

constraint set W. Therefore, an objective function can be constructed by using the ratio of d_W and d_B to get an optimization problem; the optimal $A*$ for the solution is

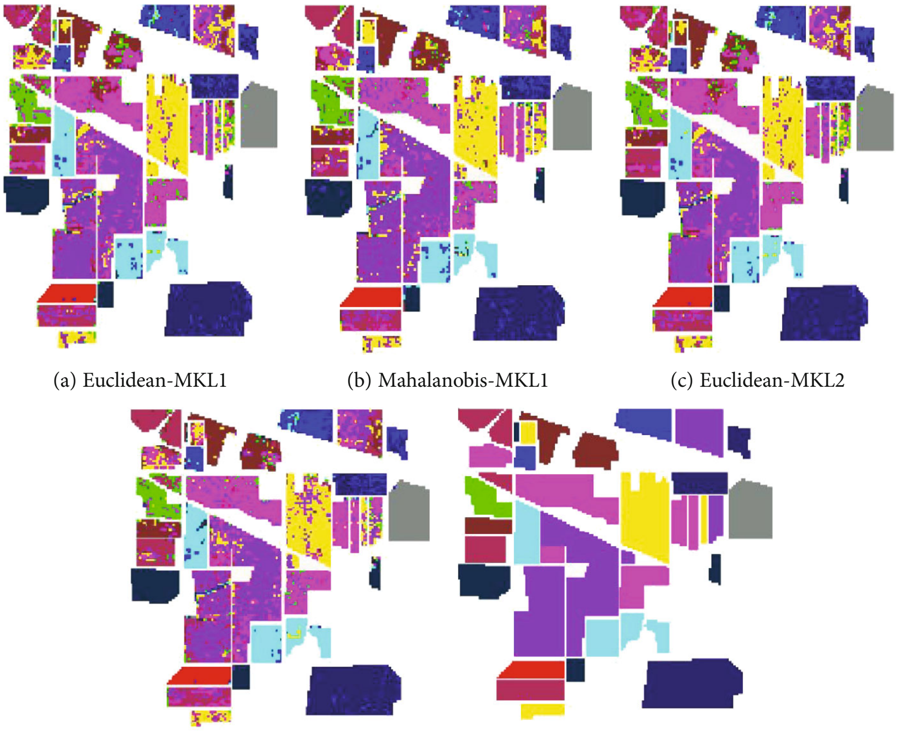
$$\mathbf{A}^{*} = \underset{\mathbf{A}^{T}\mathbf{A}=\mathbf{I}}{\arg\max} \frac{\operatorname{tr}\left(\mathbf{A}^{T}\mathbf{S}_{B}\mathbf{A}\right)}{\operatorname{tr}\left(\mathbf{A}^{T}\mathbf{S}_{W}\mathbf{A}\right)}. \qquad (32)$$

Thus, calculating the Mahalanobis distance matrix is $\mathbf{M}^{*} = \mathbf{A}^{*}(\mathbf{A}^{*})^{T}$ after learning.

2.5. Procedural Steps and Discussion. The procedure is shown in Figure 2. The procedure includes three stages: the multikernel optimization, training, and testing. The first step is optimizing the weight vector of multiple

Table 2: Performance on the Pavia University data (%).

Class	SVM (polynomial)	SVM (Gaussian)	MKL (polynomial)	MKL (Gaussian)	KSRC (polynomial)	KSRC (Gaussian)	Proposed (Q-kernel)
1	83.93	84.45	84.24	88.22	84.93	85.42	92.24
2	85.23	91.45	90.29	93.15	86.26	92.45	94.36
3	70.24	74.14	74.61	78.23	71.27	75.13	83.27
4	87.27	90.47	90.26	89.22	88.28	91.45	91.93
5	96.28	97.62	97.23	97.24	97.23	98.65	98.73
6	70.41	78.77	77.47	84.26	71.47	79.72	84.15
7	69.33	71.13	70.16	76.27	69.92	72.17	82.62
8	76.24	81.53	80.22	82.93	77.26	82.56	86.27
9	98.56	99.42	99.24	99.26	98.85	99.93	99.93



(a) Euclidean-MKL1

(b) Mahalanobis-MKL1

(c) Euclidean-MKL2

(d) Mahalanobis-MKL2

(e) True ground

Figure 5: Classification results of different methods on the Indian Pines dataset.

quasiconformal kernels with Fisher and maximum margin criterions; the second step is optimizing multiple kernels for the classification. And the final step is classifier design with the optimal kernels.

On learning bounds of the proposed algorithm, the learning criterion is to maximize the accuracy of the test data. Given the function $f : x \longrightarrow R$, the threshold version of f is defined as

$$er(f) = \frac{1}{n}\left|\left\{n+1 \le i \le 2n : y_i f(x_i) \le 0\right\}\right|, \tag{33}$$

where the kernel-based classifiers are the threshold of kernel expansions of the form, and the bounded norm is

$$\|w\|^2 = \sum_{i,j=1}^{2n} \alpha_i \alpha_j k\left(x_i, x_j\right) = \alpha^T K \alpha \le \frac{1}{\gamma^2}. \tag{34}$$

For any $\gamma > 0$ with a probability at least $1-\delta$ over the data x_i, y_i, each function $f \in F_k$ has $er(f)$ no more than $1/n$ $\sum_{i=1}^{n} \max\left(1 - y_i f(x_i)\right) + 1/\sqrt{n}(4 + \sqrt{2 \log(1/\delta)} +$

Table 3: OA (%) of different methods on the Indian Pines dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	57.23 ± 2.13	65.42 ± 1.95	67.98 ± 1.90	72.05 ± 1.24	74.23 ± 1.16
Mahalanobis-MKL1	62.25 ± 2.45	67.84 ± 2.33	71.97 ± 1.51	74.69 ± 1.23	75.10 ± 0.98
Euclidean-MKL2	59.10 ± 2.24	66.75 ± 1.82	71.08 ± 1.92	73.90 ± 1.24	76.07 ± 1.21
Mahalanobis-MKL2	62.04 ± 2.34	68.63 ± 2.43	73.34 ± 1.59	74.97 ± 1.29	77.12 ± 1.10

Table 4: Kappa coefficient of different methods on the Indian Pines dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	0.532	0.598	0.595	0.678	0.696
Euclidean-MKL2	0.544	0.612	0.663	0.698	0.728
Mahalanobis-MKL1	0.565	0.641	0.682	0.712	0.732
Mahalanobis-MKL2	0.570	0.648	0.689	0.729	0.742

$\sqrt{\varsigma(k)/n\gamma^2})$, where $\varsigma(K) = E \max_{K\in K} \sigma^T K\sigma$ is the expectation over σ. So, $\varsigma(K) = cE \max_{K\in K} \sigma^T K/(\mathrm{trace}(K))\sigma$ and

$$\varsigma(K) \leq c \min\left(m, n \max_{j} \frac{\lambda_j}{\mathrm{trace}\left(K_j\right)}\right), \qquad (35)$$

where λ_j is the largest eigenvalue of K_j.

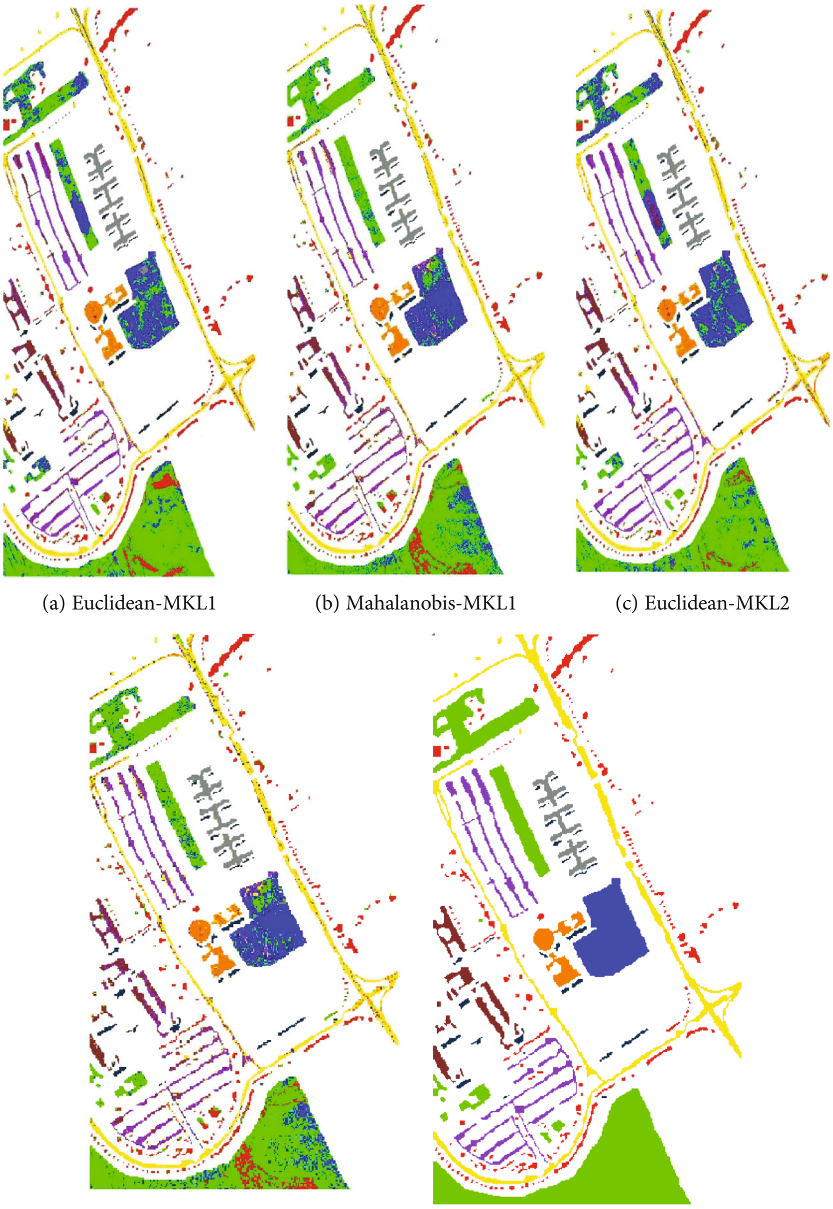
3. Experiments and Analysis

3.1. Experiment Setting. The performance of the proposed intelligent hyperspectral instrument is evaluated. The accuracy of spectrum classification is an important index to evaluate the performance of spectrum classification. The experiment was carried out on two sensing datasets of a hyperspectral imager, i.e., Indian Pines dataset and Pavia University dataset. The dataset of Indian Pines is based on an airborne platform, under the various spectral and the spatial resolutions. The data includes 224 0.4-2.5 μm bands. Nine kinds of 145 × 145-pixel images are realized in the experiment. The data collection at the University of Pavia is based on a reflective optics system imaging spectrometer (ROSIS). The data includes 115 bands. In the experiment, the performance of 9 kinds of 610 × 340 images is verified. Except for the feature dimensions of the participating categories, the rest of the two experiments were identical. In the use of classification features, considering the computational efficiency and stability of the Mahalanobis matrix, the dimension of original spectral features is reduced by PCA. After the dimension reduction, the features are normalized to eliminate the deviation caused by the sampling method. For the first experiment, the top 30 principal components are selected to participate in the classification; that is, the feature dimension was 30. For experiment 2, the first 40 principal components are selected to participate in the classification; that is, the feature dimension is 40. On the classifier settings, the preset parameter values are selected by cross-validation by a standard multiclass SVM. In the kernel function setting, the Gaussian kernel function and the Mahalanobis Gaussian kernel function are used as the basis kernel functions, respectively. The scale parameter σ is set between [0.01, 2], and the number of basis kernels is 10. In terms of the evaluation index, the overall classification accuracy (OA) and Kappa coefficient (KC) are used as performance evaluation indicators, and information such as classifier training time, test time, and support vector number is collected. In the comparison method, the average multikernel and different multiple kernel learning methods are used as the multikernel combination coefficient algorithm, and the Euclidean distance Gaussian kernel and the Mahalanobis Gaussian kernel are, respectively, used for comparison. In these experiments, we implement four algorithms as follows.

The Indian Pines dataset is collected under various spectral and spatial resolutions. The spectral curves denote the different remote sensing environments with an airborne platform. The data cube has 224 bands of spectral resolution through 0.4-2.5 μm range, and it has the spatial resolution of 20 m per pixel. We removed the noisy and water vapor absorption bands, and 200 bands of images are used in the experiments. The whole scene consists of 145 × 145 pixels and 16 classes of interested objects with the size ranging from 20 to 2468 pixels; 9 classes are used in the experiments. One example is shown in Figure 3.

Pavia University data was acquired by the reflective optics system imaging spectrometer (ROSIS) over the urban area of the University of Pavia, Northern Italy. The dataset consists of 115 spectral bands and 610 × 610 pixels with the spatial resolution of 1.3 m by pixel. Several undesirable bands influenced by the atmospheric absorption are discarded, leaving 103 bands in the 0.43–0.86 lm region. We cut a patch sized 610 × 340, consisting of 9 classes of land covers from the set. The example is shown in Figure 4.

3.2. Experiments on the Performance on Quasiconformal Kernel Mapping. In the experiments, we evaluate the performance on quasiconformal kernel mapping on the two databases. The performance of quasiconformal kernel mapping is testified and evaluated with the polynomial kernel and Gaussian kernel. We have the Kernel Sparse Representation Classifier (KSRC) and Support Vector Classifier (SVC) for



(a) Euclidean-MKL1

(b) Mahalanobis-MKL1

(c) Euclidean-MKL2

(d) Mahalanobis-MKL2

(e) True map

Figure 6: Classification results of different methods on the Pavia University dataset.

Table 5: OA (%) of different methods on the Pavia University dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	61.98 ± 3.16	67.78 ± 3.24	71.35 ± 2.39	75.35 ± 1.62	78.69 ± 1.12
Mahalanobis-MKL1	62.53 ± 3.23	69.34 ± 2.78	72.80 ± 2.10	76.24 ± 1.19	79.78 ± 1.05
Euclidean-MKL2	61.92 ± 3.21	66.97 ± 3.28	71.76 ± 2.39	75.56 ± 1.67	79.24 ± 1.33
Mahalanobis-MKL2	62.89 ± 3.22	69.42 ± 2.81	73.10 ± 2.23	76.34 ± 1.23	79.86 ± 1.16

classification. For comparisons, we also implement other algorithms, including SVM [29], RMKL-SVM [30], and POL-KSRC [31]. The experimental results on two datasets are shown in Tables 1 and 2.

3.3. Experiments on the Performance on Mahalanobis Distance Kernel. In the comparison method, the average multikernel and different multiple kernel learning methods are used as the multikernel combination coefficient algorithm,

Table 6: Kappa coefficient of different methods on the Pavia University dataset.

Method	Feature dimension				
	10	20	30	40	50
Euclidean-MKL1	0.548	0.608	0.658	0.704	0.721
Euclidean-MKL2	0.549	0.611	0.662	0.710	0.729
Mahalanobis-MKL1	0.554	0.619	0.672	0.716	0.734
Mahalanobis-MKL2	0.555	0.622	0.678	0.721	0.741

Table 7: Performance comparisons on two databases.

Datasets	Indian Pines dataset		Pavia University dataset	
Methods	OA (%)	KC (%)	OA (%)	KC (%)
RBF	73.23	63.73	75.43	68.76
Poly	75.77	66.28	78.08	72.07
Mahal-RBF	76.92	67.79	76.26	69.87
Mahal-Poly	77.64	68.62	79.07	73.24
SK-CV-RBF	67.03	64.23	75.71	69.14
SK-Poly	69.37	66.96	77.62	71.37
NMF-MKL	67.48	64.81	71.57	64.42
KNMF-MKL	68.22	65.63	72.80	65.81
Euclidean-MKL1	74.23	69.63	78.69	72.11
Euclidean-MKL2	76.07	72.84	79.24	72.92
Mahalanobis-MKL1	75.90	73.25	79.78	73.43
Mahalanobis-MKL2	78.22	74.25	79.86	74.16
Proposed Mahalanobis-QMKL1	76.96	74.35	80.27	74.52
Proposed Mahalanobis-QMKL2	79.28	75.85	80.96	75.34

and the Euclidean distance Gaussian kernel and the Mahalanobis Gaussian kernel are, respectively, used for comparison. In these experiments, we implement four algorithms as follows: Euclidean-MKL1 [32]: the Euclidean distance kernel function. Each kernel function is combined according to the same weight; that is, the combination coefficient of each kernel function is the reciprocal of the number of kernel functions (see the description of literature [32] for details). Mahalanobis-MKL1: the Mahalanobis distance kernel function is used for kernel learning. Euclidean-MKL2 [33]: the Euclidean distance kernel function is used, which describes the combination coefficient as described in [33]. Mahalanobis-MKL2: the Mahalanobis distance kernel function is used, and the kernel learning is the same as Euclidean-MKL2.

The experimental results of different methods on the Indian Pines dataset are shown in Figure 5 and Tables 3 and 4. The experimental results of different methods on the Pavia University dataset are shown in Figure 6 and Tables 5 and 6. As these results, the proposed Mahalanobis distance kernel has the highest performance on the accuracy.

3.4. Experiment Comparisons. For the comparisons, we have some experiments to compare the performance of the proposed algorithm, and the following 14 methods are implemented as the comparison: (1) RBF: RBF Euclidean kernel as the kernel function in the SVM [31]; (2) Poly: polynomial Euclidean kernel as the kernel function in the SVM [31]; (3) Mahal-RBF: Mahalanobis distance-based RBF kernel as the kernel function in the SVM [34]; (4) Mahal-Poly: Mahalanobis distance-based polynomial kernel as the kernel function in the SVM [34]; (5) SK-CV (RBF): a SVM with a single kernel and adopting the RBF kernel as the kernel function in the SVM [35]; (6) SK-Poly: standard SVM with a single kernel and adopting a polynomial kernel as the kernel function in the SVM [35]; (7) NMF-MKL: the nonnegative matrix factorization (NMF) MKL proposed by Gu et al. [28], which combines multiple kernels with NMF; (8) KNMF-MKL: the kernel-based nonnegative matrix factorization (KNMF) MKL method, also proposed by Gu et al., which combines multiple kernels with the KNMF method; (9) Euclidean-MKL1 [32]: the Euclidean distance kernel function. Each kernel function is combined according to the same weight; that is, the combination coefficient of each kernel function is the reciprocal of the number of kernel functions (see the description of literature for details). (10) Euclidean-MKL2 [33]: the Euclidean distance kernel function is used, which describes the combination coefficient as described in [33]; (11) Mahalanobis-MKL1: proposed the Mahalanobis distance-based multiple kernel function, and learning criterions are same as Euclidean-MKL1; (12) Mahalanobis-MKL2: the Mahalanobis distance kernel function is used, and learning criterions are the same as Euclidean-MKL2; (13) Mahalanobis-QMKL1: proposed the Mahalanobis distance-based multiple quasiconformal kernel function, and learning criterions are the same as Euclidean-MKL1; (14) Mahalanobis-QMKL2: the Mahalanobis distance quasiconformal kernel function is used, and learning criterions are the same as Euclidean-MKL2.

As shown in Table 7, the proposed scheme is effective to the hyperspectral image classification. The quasiconformal mapping-based multiple kernel learning network framework is effective and feasible for hyperspectral data classification, and the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning, and the objective function of measuring the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. Compared with other kernel-based learning methods, the proposed algorithm is effective and performs best.

3.5. Computation Efficiency and Practical Applications. In the experiments, the computational cost was recorded with a PC with a 2.6 GHz i5-3320 processor and 4 GB RAM. The different computational costs are achieved under the different features. The proposed algorithms omit the parameter optimization process under the same dimension of feature vector, so the high computation efficiency is achieved. Both Euclidean-based method and Mahalanobis-based method adopt nonnegative matrix factorization to optimize the kernel weights, and a higher dimension of features requires more time because it needs more memory to save the kernel matrix and has more dimensions to compute. So, Mahalanobis-based kernel learning has the higher

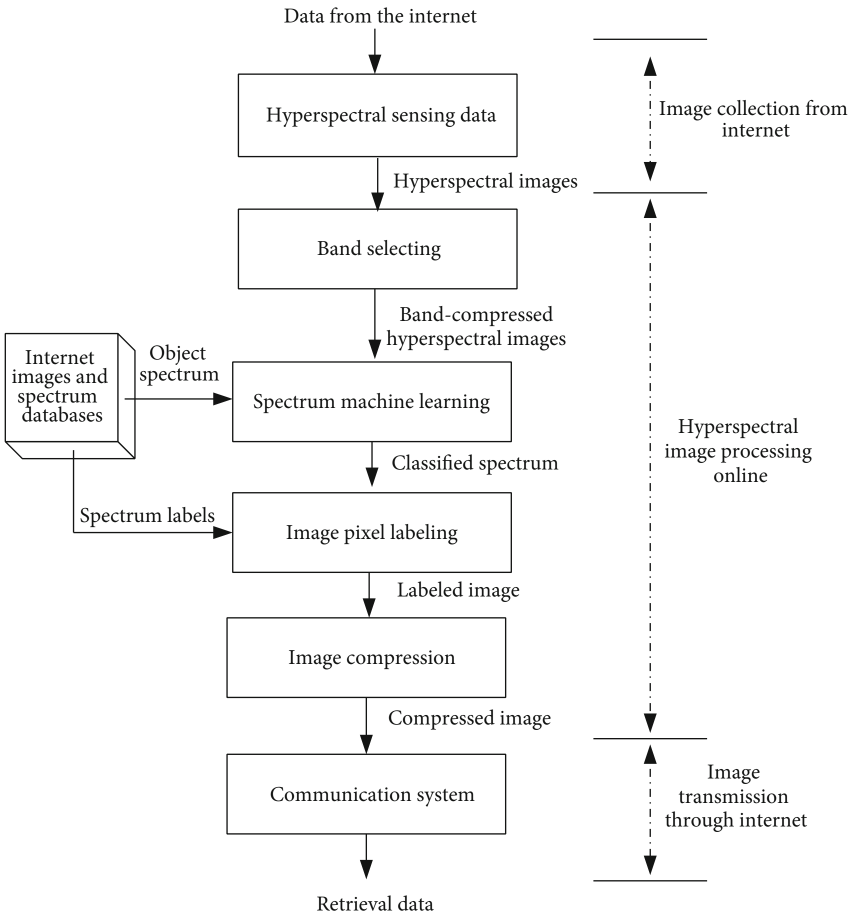

Data from the internet
Hyperspectral sensing data
Hyperspectral images
Band selecting
Band-compressed hyperspectral images
Internet images and spectrum databases
Object spectrum
Spectrum machine learning
Classified spectrum
Spectrum labels
Image pixel labeling
Labeled image
Image compression
Compressed image
Communication system
Retrieval data
Image collection from internet
Hyperspectral image processing online
Image transmission through internet

Figure 7: The practical application framework of hyperspectral data retrieval under internet environment.

computation efficiency than Euclidean-based method under the same dimension of the features.

In the practical application system, the system framework is shown in Figure 7. For the quasiconformal mapping kernel machine learning-based intelligent hyperspectral data retrieval under internet environment, the framework includes three stages: image collection, image processing online, and image transmission. In the framework, the second stage is important. Different from the experiment setting, the practical application system includes the image collection and image transmission.

4. Conclusion

In this paper, we present the quasiconformal mapping kernel machine learning-based intelligent hyperspectral data classification algorithm for the internet-based hyperspectral data retrieval and with the wide application in intelligent internet data mining as the application of AIoT (Artificial Intelligence of Things). The contributions of the algorithm lie in the following points: the quasiconformal mapping-based multiple kernel learning network framework is proposed for hyperspectral data classification, and the Mahalanobis distance kernel function is as the network nodes with the higher discriminative ability than Euclidean distance-based kernel function learning, and the objective function of measuring the class discriminative ability is proposed to seek the optimal parameters of the quasiconformal mapping projection. Experiments show that the proposed scheme is effective to the hyperspectral image classification. The proposed algorithm has advantages on the large training sample construction with the data from the internet, including images, videos, and other information.

Data Availability

We have not used specific data from other sources for the simulations of the results. The two popular hyperspectral datasets in this paper, Indian Pines dataset and Pavia University data, are downloaded free from the website: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes. The proposed algorithm is implemented in Python.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We would like to thank Dr. Li Li, Prof. Junbao Li, and Prof. Yanfeng Gu for providing the kernel-based learning programs of their papers for comparison in the experiments. This work is supported by the National Science Foundation of China under Grant No. 61871142.

References

[1] A. L. Afzal and S. Asharaf, "Deep kernel learning in core vector machines," *Pattern Analysis & Applications*, vol. 21, no. 3, pp. 721–729, 2018.

[2] J.-B. Li, J.-S. Pan, and S.-C. Chu, "Kernel class-wise locality preserving projection," *Information Sciences*, vol. 178, no. 7, pp. 1825–1835, 2008.

[3] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

[4] L. Wang, L. Chan, and P. Xue, "A Criterion for Optimizing Kernel Parameters in KBDA for Image Retrieval," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 3, pp. 556–562, 2005.

[5] J. S. Pan, P. Hu, and S. C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *PRO*, vol. 7, no. 11, pp. 845–856, 2019.

[6] J.-S. Pan, C.-Y. Lee, A. Sghaier, M. Zeghid, and J. Xie, "Novel systolization of subquadratic space complexity multipliers based on Toeplitz matrix-vector product approach," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 27, no. 7, pp. 1614–1622, 2019.

[7] H. Zhao, S. Sun, Z. Jing, and J. Yang, "Local structure based supervised feature extraction," *Pattern Recognition*, vol. 39, no. 8, pp. 1546–1550, 2006.

[8] J. Huang, P. Yuen, W. Chen, and J. Lai, "Kernel subspace LDA with optimized kernel parameters on face recognition," *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, 2004, pp. 187–192, Seoul, South Korea, South, May 2004.

[9] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.

[10] X. Xie, B. Li, and X. Chai, "A framework of quasiconformal mapping-based kernel machine with its application to hyperspectral remote sensing," *Measurement*, vol. 80, pp. 270–280, 2016.

[11] N. Subrahmanya and Y. Shin, "Sparse multiple kernel learning for signal processing applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 788–798, 2010.

[12] S. Sören, G. Rätsch, and C. Schäfer, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[13] K. Marius, U. Brefeld, and S. Sonnenburg, "lp-norm multiple kernel learning," *The Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.

[14] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2017.

[15] G. Wu, M. Kim, Q. Wang, B. C. Munsell, and D. Shen, "Scalable high-performance image registration framework by unsupervised deep feature representations learning," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1505–1516, 2016.

[16] M. Jiu and H. Sahbi, "Nonlinear deep kernel learning for image annotation," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1820–1832, 2017.

[17] C. Wang, J. Shi, Q. Zhang, and S. Ying, "Histopathological image classification with bilinear convolutional neural networks," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4050–4053, Seogwipo, South Korea, 2017.

[18] W. Li, X. Qian, and J. Ji, "Noise-tolerant deep learning for histopathological image segmentation," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3075–3079, Beijing, China, September 2017.

[19] N. Kumar, A. V. Rajwade, S. Chandran, and S. P. Awate, "Kernel generalized Gaussian and robust statistical learning for abnormality detection in medical images," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 4157–4161, Beijing, China, September 2017.

[20] I. Rebai, Y. BenAyed, and W. Mahdi, "Deep multilayer multiple kernel learning," *Neural Computing and Applications*, vol. 27, no. 8, pp. 2305–2314, 2016.

[21] I. Rebai, Y. BenAyed, and W. Mahdi, "Deep kernel-SVM network," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1955–1960, Vancouver, BC, Canada, July 2016.

[22] E. K. Wang, C.-M. Chen, M. M. Hassan, and A. Almogren, "A deep learning based medical image segmentation technique in Internet-of-Medical-Things domain," *Future Generation Computer Systems*, vol. 108, pp. 135–144, 2020.

[23] E. K. Wang, C. M. Chen, F. Wang, M. K. Khan, and S. Kumari, "Joint-learning segmentation in Internet of drones (IoD)-based monitor systems," *Computer Communications*, vol. 152, pp. 54–62, 2020.

[24] J. Zhuang, I. W. Tsang, and S. C. H. Hoi, "Two-layer multiple kernel learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 909–917, Ft. Lauderdale, 2011.

[25] E. V. Strobl and S. Visweswaran, "Deep multiple kernel learning," in *2013 12th International Conference on Machine Learning and Applications*no. 1, pp. 414–417, Miami, FL, USA, December 2013.

[26] I. Rebai, Y. BenAyed, and W. Mahdi, "Deep architecture using multi-kernel learning and multi-classifier methods," in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–6, Marrakech, Morocco, November 2015.

[27] L. Le, J. Hao, Y. Xie, and J. Priestley, "Deep kernel: learning kernel function from data using deep neural network," in *2016 IEEE/ACM 3rd International Conference on Big Data Computing Applications and Technologies (BDCAT)*, pp. 1–7, Shanghai, China, December 2016.

[28] Y. Gu, Q. Wang, H. Wang, D. You, and Y. Zhang, "Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2739–2751, 2015.

[29] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "Simple MKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[30] S. Zhuo, W. Cheng, and H. Ying, "Multiple kernel learning for classification of hyperspectral imagery with neighborhood preserving embedding," *The Journal of Computer Information Systems*, vol. 9, no. 9, pp. 3563–3570, 2013.

[31] D. Wang, D. S. Yeung, and E. C. C. Tsang, “Weighted Mahalanobis distance kernels for support vector machines,” *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1453–1462, 2007.

[32] M. Gonen and E. Alpaydin, “Multiple kernel learning algorithms,” *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[33] Y. Gu, C. Wang, Y. Z. Di You, Y. Zhang, S. Wang, and Y. Zhang, “Representative multiple kernel learning for classification in hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 7, pp. 2852–2865, 2012.

[34] L. Li, C. Sun, L. Lin, J. Li, and S. Jiang, “A Mahalanobis metric learning-based polynomial kernel for classification of hyperspectral images,” *Neural Computing and Applications*, vol. 29, no. 4, pp. 1103–1113.

[35] L. Li, C. Sun, L. Lin, J. Li, and S. Jiang, “A dual-layer supervised Mahalanobis kernel for the classification of hyperspectral images,” *Neurocomputing*, vol. 214, pp. 430–444, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8815001, 8 pages
https://doi.org/10.1155/2020/8815001

Research Article

Semantic Integration of Sensor Knowledge on Artificial Internet of Things

Yikun Huang,[1] Xingsi Xue ,[2,3] and Chao Jiang [4]

[1] *Department of Information Technology, Concord University College Fujian Normal University, Fuzhou/350118, China*
[2] *Fujian Key Lab for Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou/350118, China*
[3] *Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, Guilin/541004, China*
[4] *Intelligent Information Processing Research Center, Fujian University of Technology, Fuzhou/350118, China*

Correspondence should be addressed to Xingsi Xue; jack8375@gmail.com

Received 25 May 2020; Revised 27 June 2020; Accepted 2 July 2020; Published 25 July 2020

Academic Editor: Pei-Wei Tsai

Artificial Internet of Things (AIoT) integrates Artificial Intelligence (AI) with the Internet of Things (IoT) to create the sensor network that can communicate and process data. To implement the communications and co-operations among intelligent systems on AIoT, it is necessary to annotate sensor data with the semantic meanings to overcome heterogeneity problem among different sensors, which requires the utilization of sensor ontology. Sensor ontology formally models the knowledge on AIoT by defining the concepts, the properties describing a concept, and the relationships between two concepts. Due to human's subjectivity, a concept in different sensor ontologies could be defined with different terminologies and contexts, yielding the ontology heterogeneity problem. Thus, before using these ontologies, it is necessary to integrate their knowledge by finding the correspondences between their concepts, i.e., the so-called ontology matching. In this work, a novel sensor ontology matching framework is proposed, which aggregates three kinds of Concept Similarity Measures (CSMs) and an alignment extraction approach to determine the sensor ontology alignment. To ensure the quality of the alignments, we further propose a compact Particle Swarm Optimization algorithm (cPSO) to optimize the aggregating weights for the CSMs and a threshold for filtering the alignment. The experiment utilizes the Ontology Alignment Evaluation Initiative (OAEI)'s conference track and two pairs of real sensor ontologies to test cPSO's performance. The experimental results show that the quality of the alignments obtained by cPSO statistically outperforms other state-of-the-art sensor ontology matching techniques.

1. Introduction

Internet of Things (IoT) [1] consists of interconnected things with built-in sensors, and Artificial IoT (AIoT) [2] further integrates Artificial Intelligence (AI) with IoT to create the sensor network that can communicate and process data. To implement the communications and co-operations among intelligent systems on AIoT, it is necessary to annotate sensor data with the semantic meanings to overcome heterogeneity problem among different sensors, which requires the utilization of sensor ontology [3]. Sensor ontology formally models the knowledge on AIoT by defining the concepts, the properties describing a concept, and the relationships between two concepts. Since sensor ontologies are regarded as the solution to data heterogeneity on AIoT, and in recent years, many sensor ontologies [4] have been developed. However, due to human's subjectivity, the overlapping information in these ontologies could be defined with different terminologies and contexts, yielding the ontology heterogeneity problem. Therefore, before using them, it is necessary to integrate their knowledge by finding the correspondences between their concepts. Ontology matching can bring sensor ontologies into mutual agreement by automatic determining identical concept correspondences (i.e., ontology alignment), which is regarded as an effective technique to address the ontology heterogeneity problem.

Due to high computational complexity in the matching process, the Swarm Intelligence algorithm (SI) has become a popular methodology for integrating heterogeneous ontologies [5–9]. Martinez-Gil and Montes [10] propose the Genetics for Ontology Alignments (GOAL), which first generates a similarity matrix for each similarity measure, and then uses the Genetic Algorithm (GA) to optimize the weights for aggregating these matrices. Aggregating weights determined by GOAL can be reused to match the ontologies with similar heterogeneous features. Ginsca and Iftene [11] not only optimize the parameters in the matching process but also the threshold in the alignment filtering process. Acampora et al. [12] try to improve GA's converging speed as well as the solution's quality by introducing a local search strategy. Xue and Wang [13] propose a new metric to approximately measure the alignment's f-measure [14], and on this basis, utilize the hybrid GA to execute the instance-level matching in the Linked Open Data cloud (LOD). More recently, He et al. [15] propose an Artificial Bee Colony algorithm (ABC) based matching technique to aggregate different similarity measures, which can improve the alignment's quality. These SI-based matching techniques need to first store the similarity matrices determined by the similarity measures, which sharply increase the computational complexity. To this end, Genetic Algorithm based Ontology Matching (GAOM) [16] models the ontology matching as a bipartite graph matching process and tries to use GA to directly determine the alignment with high quality. Since the instance information can effectively improve the alignment's precision value, Alves et al. [17] first propose an instance-based similarity measure and then utilize a hybrid GA to determine the optimal mappings. MapPSO [18] models the ontology matching as a bipartite graph matching problem, and it proposes to use the Particle Swarm Optimization algorithm (PSO) [19] to address it. MapPSO utilizes the statistical information of the alignment to approximately evaluate its quality and guide the algorithm's search direction, which can automatically determine high-quality alignments. For dynamic applications on SSW, it is necessary to integrate the sensor ontologies online, and thus, besides the quality of the alignments, the matching efficiency is also of prime importance. Being inspired by the success of compact SI in various applications [20–23], this work proposes a compact PSO (cPSO) to integrate the sensor knowledge in AIoT. Our proposal uses a probabilistic representation of the population to execute the optimizing process, which simulates the population behaviour as it extensively explores the decision space at the beginning of the optimization process and progressively focuses the search on the most promising genotypes and narrows the search radius. Thus, a run of cPSO requires much more limited memory consumption comparing to the standard PSO. In particular, we formally define the sensor ontology matching problem and propose a problem-specific cPSO to effectively address the problem and integrate the sensor knowledge inside.

The rest of the paper is organized as follows: Section 2 presents the concept similarity measures and the mathematical model of sensor ontology matching problem; Section 3 gives the details of cPSO; Section 4 shows the experimental results; and finally; Section 5 draws the conclusions.

2. Preliminaries

2.1. Concept Similarity Measure. Concept Similarity Measure (CMS) is a function that takes as input two concepts' information, and output a real number in [0, 1] which represents their similarity value. In general, there are three kinds of CMS, i.e., string-based CMS, linguistic-based CMS, and structure-based CMS. In particular, string-based CMS takes as input two concepts' labels and compares their syntax information, linguistic-based CMS also compares two concepts' labels but it uses the external digital dictionary such as Wordnet [24] to calculate their similarity value, and structure-based CMS calculate the similarity value of two concepts based on their direct super-concepts and subconcepts.

Given two concepts c_1 and c_2, we first remove the meaningless words (such as the stop word) from their labels and convert them into two token sets T_1 and T_2, then the string-based similarity value is calculated as follows:

$$\mathrm{sim}_s(c_1, c_2) = \min\left\{\frac{|T_1 \cap T_2|}{|T_1|}, \frac{|T_1 \cap T_2|}{|T_2|}\right\}, \quad (1)$$

where $|T_1|$ and $|T_2|$ are, respectively, the cardinalities of $|T_1|$ and $|T_2|$. The first ratio indicates the overlap fraction of $|T_1|$ with respect to $|T_2|$, the second one indicates $|T_2|$ with respect to $|T_1|$, and the minimum value is selected as their string-based similarity value. The linguistic-based similarity value is defined as follows:

$$\mathrm{sim}_l(c_1, c_2) = \frac{\sum_{i=1}^{|T_1|} \max\left\{\mathrm{sim}'\left(T_{1,i}, T_{2,j}\right)\right\}_{j=1}^{|T_2|}}{|T_1|}, \quad (2)$$

where $T_{1,i}$ denotes the ith token in $|T_1|$ and $T_{2,j}$ is the jth token in $|T_2|$, $\mathrm{sim}'(T_{1,i}, T_{2,j}) = 1$ if they are synonymous in Wordnet, otherwise 0. Finally, supposing superC1 and superC2 are, respectively, the super-concept set of c_1 and c_2, subC_1 and subC_2 are, respectively, the direct subconcept set of c_1 and c_2, the structure similarity value is defined as follows:

$$\mathrm{sim}_u(c_1, c_2) = \frac{\sum_{i=1}^{|\mathrm{super}C_1|} \max\left\{\mathrm{sim}_l\left(\mathrm{super}C_{1,i}, \mathrm{super}C_{2,j}\right)\right\}_{j=1}^{|\mathrm{super}C_2|} + \sum_{i=1}^{|\mathrm{sub}C_1|} \max\left\{\mathrm{sim}_l\left(\mathrm{sub}C_{1,i}, \mathrm{sub}C_{2,j}\right)\right\}_{j=1}^{|\mathrm{sub}C_2|}}{|\mathrm{super}C_1| + |\mathrm{sub}C_1|}, \quad (3)$$

where super$C_{1,i}$ denotes the ith super-class of c_1 and super $C_{2,j}$ the jth super-class of c_2, sub$C_{1,i}$ is the ith direct subclass of c_1 and sub$C_{2,j}$ the jth subclass of c_2.

Since none of the CSMs can ensure the effectiveness in all context, i.e., distinguishing all the heterogeneous concepts, usually they are combined together to enhance the result's confidence. Due to its flexibility, the weighted average strategy becomes a popular way of aggregating CSMs, which is defined as follows:

$$\mathrm{sim}(c_1, c_2) = w_1 \times \mathrm{sim}_s(c_1, c_2) + w_2 \times \mathrm{sim}_l(c_1, c_2) + w_3 \times \mathrm{sim}_u(c_1, c_2), \tag{4}$$

where $w_i \in [0, 1], i = 1, 2, 3, \sum w_i = 1$.

2.2. Alignment Extraction. Each aggregating weight set corresponds to a unique aggregated CSM, which can be further used to construct a similarity matrix M whose element m_{ij} is the similarity value between the ith concept of one ontology and the jth concept of the other. On this basis, we can extract an alignment with the cardinality 1 : 1 (one concept from the source ontology is only mapped with one concept from the target ontology and vice versa) from M according to the following steps: (1) sort all the similarity values in M in descending order; (2) output the element with the largest value, say m_{ij}, as a concept correspondence (c_i, c_j, m_{ij}) in the extracted alignment; (3) replace the elements that are in the same row or column with m_{ij} as 0; (4) repeat the steps (1) to (3) until all the elements in M are 0. Figure 1 shows an example of the extracting process, where O_1 and O_2 are two ontologies, $c_{i,j}$ denotes the ith ontology's jth concept. Finally, six correspondences are extracted: $(c_{1,1}, c_{2,1}, 0.88)$, $(c_{1,2}, c_{2,2}, 0.88)$, $(c_{1,3}, c_{2,3}, 0.6)$, $(c_{1,5}, c_{2,5}, 0.6)$, $(c_{1,1}, c_{2,1}, 0.88)$, $(c_{1,4}, c_{2,4}, 0.08)$. With respect to the last correspondence, since its similarity value is low, which is regarded as unauthentic, the final alignment consists of top five correspondences.

2.3. Sensor Ontology Matching Problem. Since the quality of an alignment is directly proportional to the mean similarity value of all the correspondences found and the cardinality of the alignment, we utilize the following equation to calculate an alignment A's quality:

$$f(A) = \sqrt{\frac{|A|}{\max\{|O_1|, |O_2|\}} \times \frac{\mathrm{sim}_i}{|A|}}, \tag{5}$$

where $|O_1|$, $|O_2|$, and $|A|$ are, respectively, the cardinalities of two ontologies O_1, O_2, and A, sim_i is the ith correspondence's similarity value.

On this basis, the mathematical model of sensor ontology matching problem can be defined as follows:

$$\begin{cases} \min F(W) \\ \mathrm{s.t.}\, W = (w_1, w_2, \cdots)^T \\ w_i \in [0, 1] \\ \sum w_i = 1 \end{cases} \tag{6}$$

where $w_1, i = 1, 2, \cdots$ represents the ith similarity measure's aggregating weight, and $F(W)$ calculates the aggregating weight set W's corresponding alignment's quality.

3. Compact Particle Swarm Optimization Algorithm

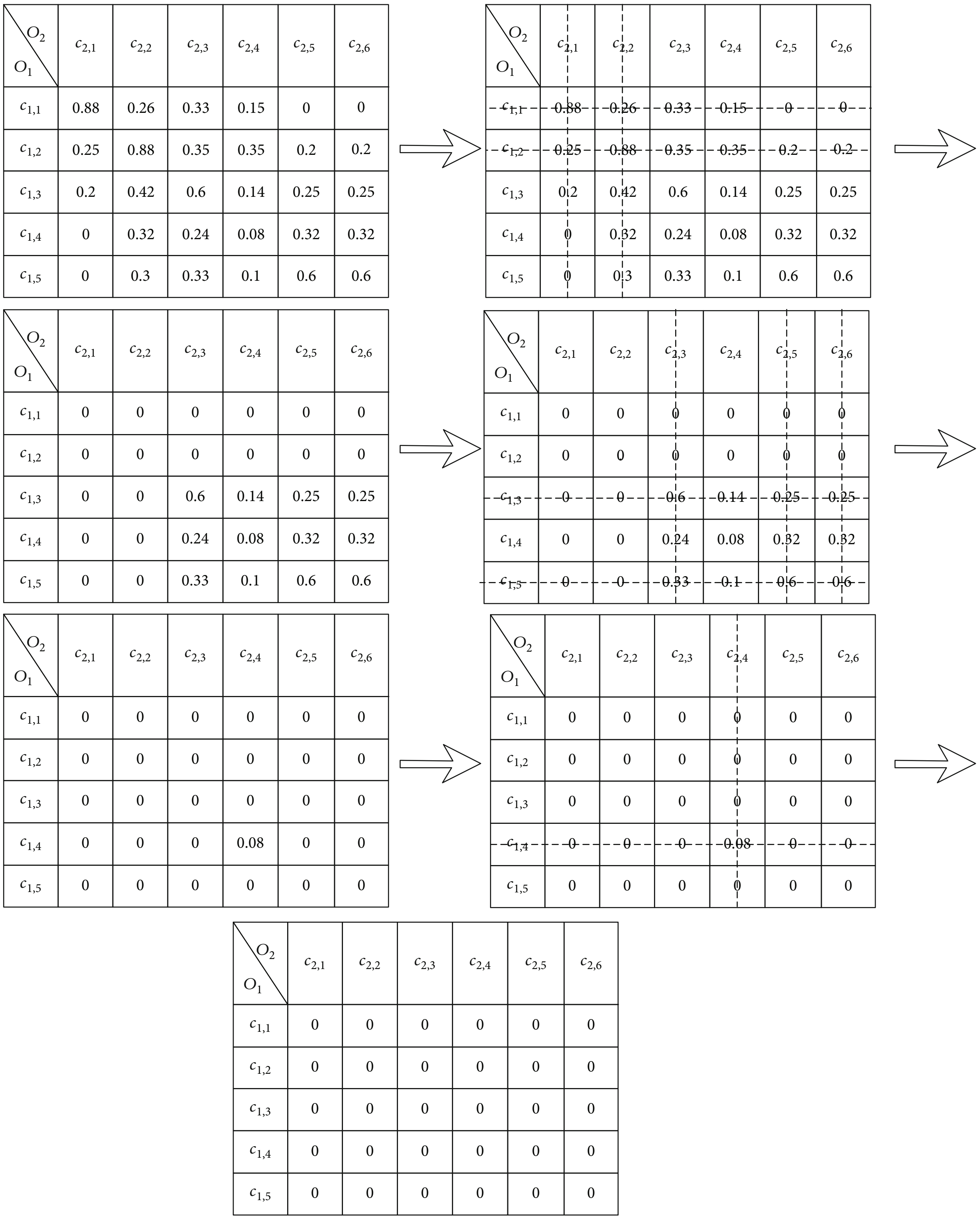
PSO is inspired by the behaviour of birds, where each bird (particle) has the memory of the best-visited position and moves to a leading bird (elite particle) with some degree of randomization. This procedure can be described with the following update sequence, for the ith particle in generation t:

$$v_i^{t+1} = \Phi_1 v_i^t + \Phi_2 \left(\mathrm{particle}_{i-lb}^t - \mathrm{particle}_i^t\right) + \Phi_3 \left(\mathrm{particle}_{i-gb}^t - \mathrm{particle}_i^t\right), \tag{7}$$

$$\mathrm{particle}_i^{t+1} = \mathrm{particle}_i^t + v_i^{t+1}, \tag{8}$$

where v_i^t is the velocity, i.e., a perturbation vector, particle$_i^t$ is the ith particle's position in current generation and particle$_{i-lb}^t$ is its best position visited in the history, particle$_{i-gb}^t$ is the best position found by all the particles, and Φ_i, i = 1, 2, 3, is the weight vector. Eq. (7) and Eq. (8) indicate that PSO update each particle by exchanging its gene values with both local best particle and global best particle to find a better position. Clearly, the original PSO is a population-based SI, and in this work, we further propose a compact version of PSO to improve the algorithm's performance.

3.1. Encoding and Decoding Mechanism. This work uses a binary encoding mechanism, i.e., Gray code, and each particle's gene values can be divided into two parts, one stands for the weight set for aggregating the similarity measures and the other for the similarity threshold for filtering the correspondences with low similarity values. Concerning the characteristics of the weights in Section 2.1, we normalize them when decoding. We utilize one Probability Vectors (PVs) to represent a population, whose element number is equal to the length of a particle. Each PV's element represents the probability of being 1 corresponding to each gene bit of a particle. We can use PV to generate various binary particles through its probability in each dimension, and when each dimension value is closed to 1 or 0, the algorithm is about to converge. In addition, PV should be updated in each generation to move toward the elite, which can make the new particles generated in the next are more closed to the elite.



O_2 O_1	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$
$c_{1,1}$	0.88	0.26	0.33	0.15	0	0
$c_{1,2}$	0.25	0.88	0.35	0.35	0.2	0.2
$c_{1,3}$	0.2	0.42	0.6	0.14	0.25	0.25
$c_{1,4}$	0	0.32	0.24	0.08	0.32	0.32
$c_{1,5}$	0	0.3	0.33	0.1	0.6	0.6

O_2 O_1	~~$c_{2,1}$~~	~~$c_{2,2}$~~	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$
~~$c_{1,1}$~~	~~0.88~~	~~0.26~~	~~0.33~~	~~0.15~~	~~0~~	~~0~~
~~$c_{1,2}$~~	~~0.25~~	~~0.88~~	~~0.35~~	~~0.35~~	~~0.2~~	~~0.2~~
$c_{1,3}$	~~0.2~~	~~0.42~~	0.6	0.14	0.25	0.25
$c_{1,4}$	~~0~~	~~0.32~~	0.24	0.08	0.32	0.32
$c_{1,5}$	~~0~~	~~0.3~~	0.33	0.1	0.6	0.6

O_2 O_1	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$
$c_{1,1}$	0	0	0	0	0	0
$c_{1,2}$	0	0	0	0	0	0
$c_{1,3}$	0	0	0.6	0.14	0.25	0.25
$c_{1,4}$	0	0	0.24	0.08	0.32	0.32
$c_{1,5}$	0	0	0.33	0.1	0.6	0.6

O_2 O_1	$c_{2,1}$	$c_{2,2}$	~~$c_{2,3}$~~	$c_{2,4}$	~~$c_{2,5}$~~	~~$c_{2,6}$~~
$c_{1,1}$	0	0	~~0~~	0	~~0~~	~~0~~
$c_{1,2}$	0	0	~~0~~	0	~~0~~	~~0~~
~~$c_{1,3}$~~	~~0~~	~~0~~	~~0.6~~	~~0.14~~	~~0.25~~	~~0.25~~
$c_{1,4}$	0	0	~~0.24~~	0.08	~~0.32~~	~~0.32~~
~~$c_{1,5}$~~	~~0~~	~~0~~	~~0.33~~	~~0.1~~	~~0.6~~	~~0.6~~

O_2 O_1	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$
$c_{1,1}$	0	0	0	0	0	0
$c_{1,2}$	0	0	0	0	0	0
$c_{1,3}$	0	0	0	0	0	0
$c_{1,4}$	0	0	0	0.08	0	0
$c_{1,5}$	0	0	0	0	0	0

O_2 O_1	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	~~$c_{2,4}$~~	$c_{2,5}$	$c_{2,6}$
$c_{1,1}$	0	0	0	~~0~~	0	0
$c_{1,2}$	0	0	0	~~0~~	0	0
$c_{1,3}$	0	0	0	~~0~~	0	0
~~$c_{1,4}$~~	~~0~~	~~0~~	~~0~~	~~0.08~~	~~0~~	~~0~~
$c_{1,5}$	0	0	0	~~0~~	0	0

O_2 O_1	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$
$c_{1,1}$	0	0	0	0	0	0
$c_{1,2}$	0	0	0	0	0	0
$c_{1,3}$	0	0	0	0	0	0
$c_{1,4}$	0	0	0	0	0	0
$c_{1,5}$	0	0	0	0	0	0

Figure 1: An example of extracting an alignment through similarity matrix.

Figure 2 shows an example of generating a particle through PV. Given a PV $(0.1, 0.3, 0.5, 0.9)^T$, generate four random numbers in [0, 1], e.g., 0.2, 0.4, 0.6, and 0.1, and we can determine a new particle by comparing them with PV's elements accordingly. To be specific, since $0.2 > 0.1$, $0.4 > 0.3$, $0.6 > 0.5$, and $0.1 < 0.9$, the newly generated particle is 0001. In each generation, PV's elements are updated according to the best particle found. If the bit value of the elite particle is

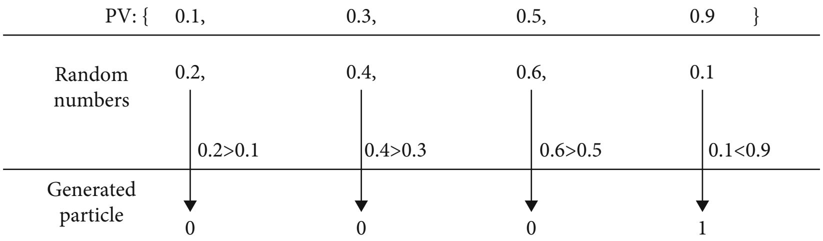


Figure 2: An example of generating a particle through PV.

1 (or 0), the corresponding PV's element will be increased (decreased) by step, which can make the newly generated particle more closer to the elite particle. For example, given a PV $(0.1, 0.3, 0.5, 0.9)^T$, an elite particle 1110 and the step length $\text{step} = 0.1$, since the first-bit value of the elite particle is 1, accordingly, we update the first element of PV by step, which makes the first-bit value of newly generated particle is more likely to be 1 (the same with elite particle's first-bit value). Therefore, after updating all elements of PV, the newly generated particles would be closer to the elite particle in terms of each bit value. When all elements of PV are 1 or 0, the newly generated particles will be the same and the algorithm converges.

3.2. Crossover Operator. Given two particles particle_1 and particle_2, the crossover operator generates one offspring by exchanging their gene values. In this work, we generate the offspring particle $\text{particle}_{\text{off}}$ by copying a sequential gene fragment of from particle_1 into the corresponding gene bits of particle_2, so that $\text{particle}_{\text{off}}$ inherits a sequential gene fragment from both particle_1 and particle_2. For the sake of clarity, the pseudo-code of the crossover operator is shown in Algorithm 1.

```
$particle_{off}$=$particle_2$
position=round(rand(0, $particle_{off}$.length)
while rand $(0, 1) < P_{cr}$ do
        $particle_{off,i} = particle_{1,i}$;
    $num = num + 1$;
    $position = (position + 1) \bmod particle_{off}.length$;
end while
return $particle_{off}$
```

Algorithm 1. Crossover operator.

4. The Pseudo-Code of Compact Particle Swarm Optimization Algorithm

The pseudo-code of cPSO is presented in Algorithm 2. cPSO first initializes the probability vector PV by setting all the elements as 0.5, which is then used to initialize the local best particle particle_{lb} and global best particle particle_{gb}. In each generation, cPSO first tries to update the particle_{lb}through the crossover between it and a new particle, and then the particle_{gb} by exchanging its gene values with particle_{lb}. Finally, cPSO updates PV according to particle_{gb} to move the new generated particle towards it.

5. Experimental Results and Analysis

5.1. Experimental Setup. In the experiment, we use the Ontology Alignment Evaluation Initiative (OAEI)'s Conference track "http://http://oaei.ontologymatching.org/2019/conference/index.html" and two pairs of real sensor ontologies to test cPSO's performance. The experiment compares cPSO with four state-of-the-art sensor ontology matching techniques, i.e., ASMOV [25], CODI [26], SOBOM [27], and FuzzyAlign [28], on all testing cases in terms of f-measure. We empirically set cPSO's crossover probability as 0.8 and maximum generation as 3000, and cPSO's results in the tables are the average of thirty independent runs. In Table 1, we briefly describe the ontologies in these testing cases.

5.2. Statistical Comparison. We utilize two popular statistical testing methods, i.e., Friedman's Test (FT) [29] and Holm's Test (HT) [30], to compare different competitors' performance. In particular, FT aims at checking whether there are differences among the competitors, and HT is further used to find whether one competitor statistically outperforms others. First, we need to reject HT's null-hypothesis that all competitors' performances are the same. To this end, the computed value X_r^2 must be equal to or greater than the tabled critical chi-square value at the specified level of significance $\alpha = 0.05$. In this work, since we are comparing 5 matchers, the critical value for 4 degrees of freedom $X_{0.05}^2$ is 9.488.

In Table 2, since $X_r^2 = 85.09$, which is greater than 9.488, and therefore, the null hypothesis is rejected. Then, HT is further carried out. Since cPSO ranks with the lowest value, it is set as a control matcher that will be compared with others. HT's z value is the testing statistic for comparing the ith and jth competitors, which is used for finding the p value that is the corresponding probability from the table of the normal distribution. The p value is then compared with $\alpha/k - i$, and according to Table 3, we can state that cPSO statistically outperforms other competitors on f-measure at 5% significance level.

```
**Initialization **
set generation $t = 0$;
set all elements in $PV$ as 0.5;
generate one particle through $PV$ to initialize the local best particle $\text{particle}_{lb}$ and global best particle $\text{particle}_{gb}$;
**Evolution **
while $t <$ maxGen do
  generate a new particle $\text{particle}_{\text{new}}$ through $PV$;
*** Update Particle ***
  $\text{particle}_{\text{new}} = \text{crossover}(\text{particle}_{\text{new}}, \text{particle}_{lb})$;
   $[\text{winner}, \text{loser}] = \text{compete}(\text{particle}_{\text{new}}, \text{particle}_{lb})$;
   if winner == $\text{particle}_{\text{new}}$then
      $\text{particle}_{lb} = \text{particle}_{\text{new}}$;
   end if
   $\text{particle}_{\text{new}} = \text{crossover}(\text{particle}_{lb}, \text{particle}_{gb})$;
   $[\text{winner}, \text{loser}] = \text{compete}(\text{particle}_{\text{new}}, \text{particle}_{gb})$;
   if winner == $\text{particle}_{\text{new}}$then
      $\text{particle}_{gb} = \text{particle}_{\text{new}}$;
   end if
   **** Update PV ****
   for$i = 0$; $i < PV$.length; $i + +$do
      if$\text{particle}_{gb}, i == 1$ then
         $PV_i = PV_i + (1/\text{particle}_{gb}.\text{length})$;
      else
         $PV_i = PV_i - 1/\text{particle}_{gb}.\text{length}$;
      end if
   end for
   $t = t + 1$;
end while
return;
```

Algorithm 2. Compact particle Swarm Optimization algorithm.

Table 1: Descriptions on testing cases.

Testing case	Ontology	Scale
Conference track	Cmt	36 classes
	Conference	60 classes
	Confof	38 classes
	Edas	104 classes
	Ekaw	77 classes
	Iasted	140 classes
	Sigkdd	49 classes
Real sensor ontology	CSIRO sensor ontology^2	33,205 classes
	Semantic sensor network ontology (SSN)^3	32,298 classes
	MMI device ontology^4	24,034 classes

6. Conclusion

AIoT aims at creating a sensor network that can communicate and process data, which can be technically implemented by using the sensor ontologies to annotate sensor data with the semantic meanings. To support the co-operations among AIoT applications based on ontologies, it is necessary to integrate these sensor ontologies by finding the alignment between them. In this work, a novel matching framework is proposed, which aggregates three kinds of CSMs and an alignment extraction approach to determine the ontology alignment. We propose a compact PSO and use it to optimize the aggregating weights for the CSMs and a threshold for filtering the alignment, which ensures the quality of the results. The experimental results show that our proposal can effectively match different sensor ontologies, and the quality of

Table 2: Friedman's test on f-measure of the alignment, and the number in round parentheses is the corresponding computed rank.

Testing case	SOBOM	CODI	ASMOV	FuzzyAlign	cPSO
Cmt-conference	0.50 (5)	0.75 (3)	0.59 (4)	0.87 (2)	0.92 (1)
Cmt-Confof	0.21 (5)	0.38 (3)	0.28 (4)	0.45 (2)	0.57 (1)
Cmt-Edas	0.48 (4)	0.75 (3)	0.42 (5)	0.86 (2)	0.88 (1)
Cmt-Ekaw	0.52 (5)	0.75 (3)	0.59 (4)	0.88 (2)	0.93 (1)
Cmt-Iasted	0.54 (4)	0.78 (3)	0.50 (5)	0.87 (2)	0.90 (1)
Cmt-Sigkdd	0.14 (5)	0.68 (2)	0.34 (4)	0.61 (3)	0.70 (1)
Conference-Confof	0.40 (5)	0.75 (3)	0.51 (4)	0.86 (1.5)	0.86 (1.5)
Conference-Edas	0.44 (5)	0.75 (3)	0.50 (4)	0.88 (2)	0.92 (1)
Conference-Ekaw	0.38 (4)	0.70 (3)	0.32 (5)	0.87 (1.5)	0.87 (1.5)
Conference-Iasted	0.54 (4)	0.73 (3)	0.50 (5)	0.89 (2)	0.94 (1)
Conference-Sigkdd	0.40 (5)	0.70 (3)	0.59 (4)	0.80 (2)	0.86 (1)
Confof-Edas	0.14 (5)	0.36 (3)	0.27 (4)	0.59 (2)	0.73 (1)
Confof-Ekaw	0.25 (5)	0.41 (3)	0.28 (4)	0.52 (2)	0.64 (1)
Confof-Iasted	0.20 (5)	0.70 (3)	0.31 (4)	0.71 (2)	0.72 (1)
Confof-Sigkdd	0.35 (5)	0.61 (3)	0.48 (4)	0.82 (2)	0.89 (1)
Edas-Ekaw	0.10 (5)	0.25 (4)	0.39 (3)	0.44 (2)	0.57 (1)
Edas-Iasted	0.32 (4)	0.72 (3)	0.20 (5)	0.84 (2)	0.88 (1)
Edas-Sigkdd	0.25 (5)	0.58 (3)	0.33 (4)	0.66 (2)	0.75 (1)
Ekaw-Iasted	0.30 (5)	0.64 (3)	0.37 (4)	0.87 (2)	0.94 (1)
Ekaw-Sigkdd	0.22 (5)	0.78 (1)	0.25 (4)	0.74 (3)	0.76 (2)
Iasted-Sigkdd	0.17 (5)	0.72 (3)	0.20 (4)	0.77 (2)	0.85 (1)
MMI device-SSN	0.77 (4)	0.80 (3)	0.73 (5)	0.88 (2)	0.95 (1)
CSIRO-SSN	0.78 (4)	0.79 (3)	0.75 (5)	0.88 (2)	0.94 (1)
MMI device-CSIRO	0.72 (5)	0.78 (3)	0.75 (4)	0.87 (2)	0.92 (1)
Average	0.38 (4.70)	0.66 (2.91)	0.43 (4.24)	0.76 (2.04)	0.82 (1.08)

Table 3: Holm's test on the alignment's quality.

i	Approach	z value	Unadjusted p value	$\frac{\alpha}{k-i}$, $\alpha = 0.05$
4	FuzzyAlign	2.04	0.035	0.050
3	CODI	3.91	5.64×10^{-10}	0.025
2	ASMOV	4.24	4.51×10^{-12}	0.012
1	SOBOM	4.70	2.22×10^{-15}	0.008

the alignments obtained by cPSO statistically outperforms other state-of-the-art sensor ontology matching techniques.

In the future, we will further improve cPSO to match the large-scale sensor ontologies, and address the problem of Instance Coreference Resolution (ICR) in the sensor network domain, which requires matching large-scale sensor instances in the Linked Open Data cloud (LOD). We also want to extend cPSO to match the ontologies in the specific domains such as the biomedical domain and geographical domain. The particular strategies and techniques need to proposed and used to improve the alignment's precision and recall because these matching tasks require specific background knowledge base and complex forms of alignment.

Data Availability

The data used to support this study can be found in http://oaei.ontologymatching.org.

Conflicts of Interest

The authors declare that they have no conflicts of interest in the work.

Acknowledgments

This work is supported by the Fujian province undergraduate universities teaching reform research project (No. FBJG20190156), the 2018 Program for Outstanding Young Scientific Researcher in Fujian Province University, the Program for New Century Excellent Talents in Fujian Province University (No. GY-Z18155), the Scientific Research Foundation of Fujian University of Technology (No. GY-Z17162), the Science and Technology Planning Project in Fuzhou City (No. 2019-G-40), the Foreign Cooperation Project in Fujian Province (No. 2019I0019), and the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (No. YQ20206).

References

[1] P. Pande and A. R. Padwalkar, "Internet of things–a future of internet: a survey," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 2, pp. 354–361, 2014.

[2] G. Katare, G. Padihar, and Z. Qureshi, "Challenges in the integration of artificial intelligence and internet of things," *International Journal of System and Software Engineering*, vol. 6, pp. 10–15, 2018.

[3] A. Sheth, C. Henson, and S. S. Sahoo, "Semantic sensor web," *IEEE Internet Computing*, vol. 12, no. 4, pp. 78–83, 2008.

[4] X. Wang, X. Zhang, and M. Li, "A survey on semantic sensor web: sensor ontology mapping and query," *International Journal of u-and e-Service, Science and Technology*, vol. 8, pp. 325–342, 2015.

[5] X. Xue and J. S. Pan, "An overview on evolutionary algorithm based ontology matching," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, pp. 75–88, 2018.

[6] X. Xue and Y. Wang, "Optimizing ontology alignments through a Memetic algorithm using both MatchFmeasure and unanimous improvement ratio," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[7] X. Xue, J. Chen, and X. Yao, "Efficient user involvement in semi-automatic ontology matching," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2018, pp. 1–11, 2018.

[8] X. Xue and J. Chen, "Using compact evolutionary Tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.

[9] S. Chu, X. Xue, J.-S. Pan, and X. Wu, "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–23, 2020.

[10] J. Martinez-Gil and J. F. A. Montes, "Evaluation of two heuristic approaches to solve the ontology meta-matching problem," *Knowledge and Information Systems*, vol. 26, no. 2, pp. 225–247, 2011.

[11] A. L. Ginsca and A. Iftene, "Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment," in *9th Roedunet International Conference*, pp. 118–122, Sibiu, Romania, June 2010.

[12] G. Acampora, V. Loia, and A. Vitiello, "Enhancing ontology alignment through a memetic aggregation of similarity measures," *Information Sciences*, vol. 250, pp. 1–20, 2013.

[13] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 580–591, 2015.

[14] C. J. V. Rijsberge, *Information Retrieval*, University of Glasgow, Butterworth, London, 1975.

[15] Y. He, X. Xue, and S. Zhang, "Using artificial bee colony algorithm for optimizing ontology alignment," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, pp. 766–773, 2017.

[16] J. Wang, Z. Ding, and C. Jiang, "GAOM: genetic algorithm based ontology matching," *Proceedings of IEEE Asia-Pacific Conference on Services Computing*, pp. 617–620, 2006.

[17] A. Alves, K. Revoredo, and F. Baião, "Ontology alignment based on instances using hybrid genetic algorithm," *Proceedings of the 7th International Conference on Ontology Matching-Volume 946. CEUR-WS. org*, pp. 242-243, 2012.

[18] J. Bock and J. Hettenhausen, "Discrete particle swarm optimisation for ontology alignment," *Information Sciences*, vol. 192, pp. 152–173, 2012.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, pp. 1942–1948, Perth, WA, Australia, Australia, December 1995.

[20] S. Chu, "An optimal deployment wireless sensor network based on compact differential evolution," *Journal of Network Intelligence*, vol. 2, no. 3, pp. 263–274, 2017.

[21] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.

[22] T. T. Nguyen, J.-S. Pan, and T. K. Dao, "A compact bat algorithm for unequal clustering in wireless sensor networks," *Applied Sciences*, vol. 9, no. 10, pp. 1–18, 2019.

[23] J.-S. Pan, P.-C. Song, S.-C. Chu, and Y.-J. Peng, "Improved compact cuckoo search algorithm applied to location of drone logistics hub," *Mathematics*, vol. 8, no. 3, p. 333, 2020.

[24] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using WordNet and lexical chains," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2264–2275, 2015.

[25] J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt, "Leveraging terminological structure for object reconciliation," in *The Semantic Web: Research and Applications. ESWC 2010. Lecture Notes in Computer Science, vol 6089*, pp. 334–348, Springer, Berlin, Heidelberg, 2010.

[26] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," *Journal of Web Semantics*, vol. 7, no. 3, pp. 235–251, 2009.

[27] P. Xu, Y. Wang, L. Cheng, and T. Zang, "Alignment results of SOBOM for OAEI 2010," *Ontology Matching*, vol. 203, pp. 7–11, 2010.

[28] S. Fernandez, I. Marsa-Maestre, J. Velasco, and B. Alarcos, "Ontology alignment architecture for semantic sensor web integration," *Sensors*, vol. 13, no. 9, pp. 12581–12604, 2013.

[29] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[30] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, pp. 65–70, 1979.