

Research Article

Study on Waste Type Identification Method Based on Bird Flock Neural Network

Shifeng Li ¹ and Liyu Chen ²

¹College of Physics Science and Technology, Tangshan Normal University, Tangshan 063000, China

²College of Mathematics and Computer Science, Tangshan Normal University, Tangshan 063000, China

Correspondence should be addressed to Shifeng Li; lsf4128@163.com

Received 22 April 2020; Revised 17 July 2020; Accepted 11 August 2020; Published 25 August 2020

Academic Editor: Nazrul Islam

Copyright © 2020 Shifeng Li and Liyu Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

According to the waste type identification requirement in waste classification, a waste type identification method based on a bird flock neural network (BFNN) was proposed. The problem of obtaining the feature dataset of waste images was considered, and color histogram and texture feature extraction techniques were used. The local optimum problem of a typical backpropagation neural network (BPNN) was considered, and a bird flock optimization (BFO) algorithm was proposed. The accuracy problem of the typical BPNN was considered, and a new online weight adjustment method of neurons was proposed. The number of hidden layer neurons (nodes) of the typical BPNN was considered, and an online adjustment method was proposed. The experimental results show that the recyclables (paper, plastic, glass, and cloth) and nonrecyclables can effectively be identified by the waste type identification method based on the BFNN, and the recognition accuracy is 81% which meets actual needs.

1. Introduction

“Shanghai Household Waste Management Regulation” had been implemented since July 2019. The domestic waste was divided into recyclables (including paper, plastic, glass, metal, and cloth) and nonrecyclables (including hazardous, wet, and dry waste) based on the regulation. Since waste classification is a heavy work, the intellectualization of waste classification has become an important topic.

In recent years, the study on neural network (NN) and image processing has been a research hotspot, but there are limited studies on waste classification using image processing and NN. Laser-induced breakdown spectrometry together with discriminant function analysis was used for the identification and classification of five groups of polymers in the electronics industry [1]. A thermal imaging-based system for classifying useful recyclables from simulated municipal solid waste (MSW) sample was reported, and the system could be used in recycling plants for processing MSW in developing countries [2]. An intelligent waste material classification system was proposed, which

was developed by using the 50-layer residual net pretrained convolutional neural network (CNN) model [3]. Two reinforcement learning-based metalearning approaches, MetaQNN and efficient neural architecture search, were investigated and compared to find suitable CNN architectures for multitarget concrete defect classification with the CONcrete DEfect BRidge IMage dataset [4].

In recent years, the study on NN and related optimization algorithm has been a research hotspot. The differential evolution- (DE-) based metaheuristic optimization algorithm was used for optimizing the search space of artificial neural network (ANN), and the effect of ANN-DE-based model predictions was better than the traditional quadratic model predictions [5]. Jicama peroxidase was immobilized on the surface modified buckypaper/polyvinyl alcohol membrane and employed for the application of methylene blue dye removal from aqueous solution, and the parameters of the isotherm and kinetic models were estimated using ant colony optimization (ACO) [6]. The GA-ANN method was used to select the most important features affecting soil erodibility to wind, and minimization of the

cost function was better than comparison algorithm [7]. A new hybrid approach which combined the recurrent fuzzy neural network (RFNN) with particle swarm optimization (PSO) algorithm to simulate the flyrock distance induced by mine blasting [8]. ACO algorithm was employed with a modified heuristic value to obtain sparsely placed nodes in the solution, and the improved ACO was used for reducing the cost of wireless sensor networks and sensor nodes [9]. A new adaptive version of particle swarm optimizer was used for improving the robustness and the stability of the controller according to the climatic changes and overcoming the problem of local optimum [10]. A NN was used for designing sensors, the parameters were modeled by ANN and optimized by genetic algorithm (GA) within the practical levels, and the optimization process led to find the most accurate optimal conditions possible [11].

It is well known that CNN can directly extract image features, which is an important algorithm especially suitable for image classification [12–14]. However, CNN requires a long training time. Even if a GPU is used for calculation, some training processes of CNNs will take several hours or even days (it takes only a few seconds for the proposed algorithm in this paper). In this paper, an algorithm was proposed for solving the problem of waste identification and making theoretical preparation for manufacturing trash can which can automatically identify and classify waste. Because the computing power of hardware of trash can is usually limited, the CNN is not used.

NN is an effective method for classification. Trained NN can automatically identify the type of waste and sort the waste to a suitable location by hardware so as to save labor cost. Backpropagation neural network (BPNN) is a classic neural network; at present, BPNN has been applied in many fields [15–17]. There are problems of local optimum [18], accuracy [19], and hidden layer neuron (nodes) adjustment [20] in a typical BPNN and other improved algorithms. According to the above problems, a bird flock neural network (BFNN) was proposed, and then a waste type identification method based on the BFNN was proposed.

2. Dataset Acquisition

The image features of waste were extracted to generate dataset, and the dataset was used as the input of NN. The type of waste was manually determined and coded, and the encoded data were used as the output of NN. The obtained dataset was used for training a NN. After training, the NN could identify the type of waste.

2.1. Obtaining Input Data. The images of some waste are shown in Figure 1.

The image features in Figure 1 were extracted to generate dataset, and the dataset was used as the input data of NN.

2.1.1. Extracting Feature Information of Color Histogram of Image. The color histogram reflects the statistical characteristics of the color distribution of images, and it is suitable for describing images which are difficult to segment and do

not need to consider the position of the object. A camera was used to capture the image within a fixed square, and the background was white. The captured result was an image with fixed size (200×200 pixel). Because the number of pixels in the image was fixed, the number of the pixels of certain color could be used as characteristic vector.

The tricolor histogram of Figure 1(d) is shown in Figure 2.

In Figure 2, the abscissa is the pixel level, and the ordinate is the number of pixels.

The number of 0–50 colored pixels of the red component histogram was captured as feature vector 1. The number of 51–100 colored pixels of the red component histogram was captured as feature vector 2. The number of 101–150 colored pixels of the red component histogram was captured as feature vector 3. The number of 151–200 colored pixels of the red component histogram was captured as feature vector 4. The number of 201–255 colored pixels of the red component histogram was captured as feature vector 5.

The extraction method of the feature vectors of green component and blue component was the same as the red component. A total of 15 feature vectors were generated.

2.1.2. Extracting Feature Information of Texture of Image.

The surface properties of images and corresponding objects are described by texture features. Three texture feature vectors including mean, contrast, and entropy were used in this paper.

Let (x, y) be a point in an image, and the calculation formula of gray difference between the point and another point $(x + \Delta x, y + \Delta y)$ is as follows:

$$\Delta g(x, y) = g(x, y) - g(x + \Delta x, y + \Delta y). \quad (1)$$

The calculation formulas of mean, contrast, and entropy are as follows [21]:

$$\text{mean} = \frac{1}{m} \sum_i i p(i),$$

$$\text{con} = \sum_i i^2 p(i), \quad (2)$$

$$\text{entropy} = - \sum_i p(i) \log_2 [p(i)],$$

where mean is the average value, m is the number of gray level, i is the gray level number, $p(i)$ is the probability of $\Delta g(x, y)$, con is the contrast value, and entropy is the entropy value.

The input of NN includes 15 color histogram feature vectors and 3 texture feature vectors and a total of 18 feature vectors.

2.2. Determining Output Data. Recyclables were divided into 5 kinds: paper, plastic, glass, metal, and cloth. Metals could be directly identified by sensors, so there was no need to be identified by image features. Therefore, the waste type

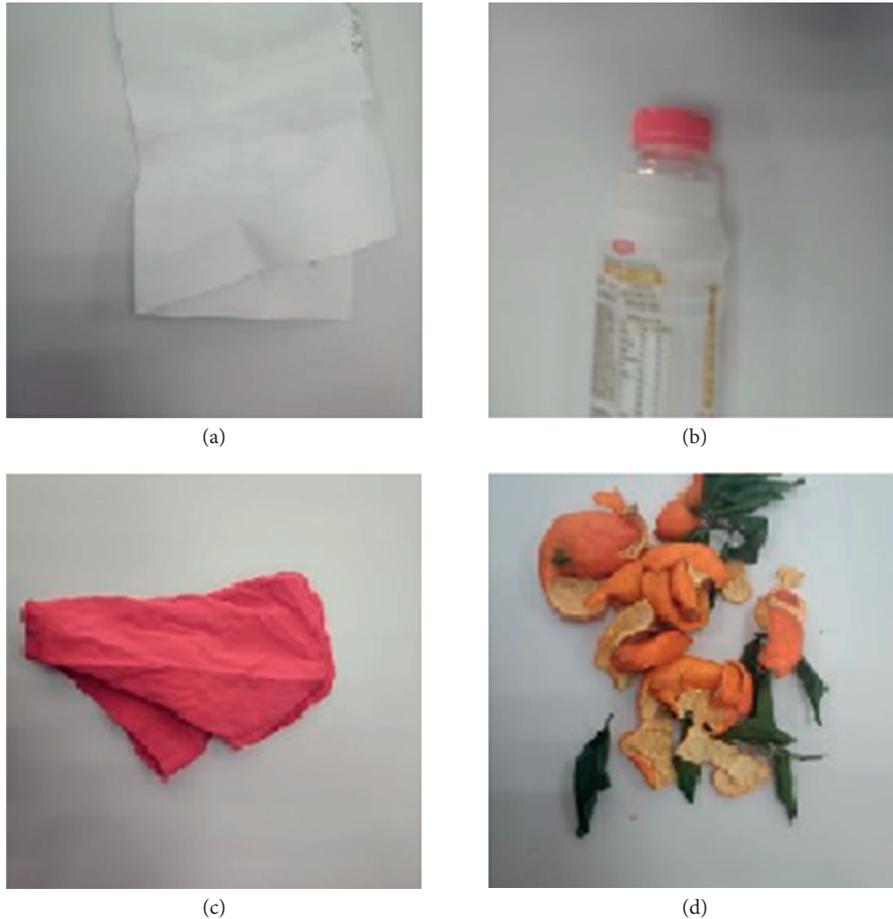


FIGURE 1: Images of waste: (a) paper; (b) plastic; (c) cloth; (d) nonrecyclables.

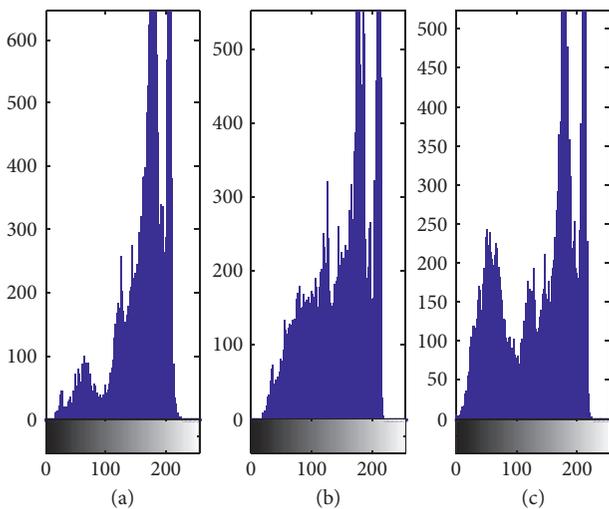


FIGURE 2: Tricolor histogram: (a) red component; (b) green component; (c) blue component.

identification method based on the BFNN was used to identify some recyclables (paper, plastic, glass, and cloth) and nonrecyclables by image features. The corresponding output of NN was in the order of 00001, 00010, 00100, 01000, and 10000.

3. Differences between BFNN and Typical BPNN

The structure of a BFNN is shown in Figure 3.

There are some characteristics of the BFNN in Figure 3. The network consists of at least three layers. The neurons in each layer are fully connected to the neurons in the next layer, and there is no connection between the neurons in the same layer. Because the BFNN is trained by offline algorithm and online algorithm, the accuracy of the algorithm can be guaranteed and the data can be added online. Because the BFNN uses online adjustment method, the number of the hidden layer neurons can be increased or decreased online.

The main differences between BFNN and typical BPNN are as follows.

3.1. A Bird Flock Algorithm. The bird flock optimization (BFO) algorithm is an intelligent optimization algorithm proposed in this paper, which will be compared with PSO [22, 23].

The differences between BFO and PSO are as follows:

- (i) PSO. Each individual emulates its most successful neighbor by updating its velocity and position to

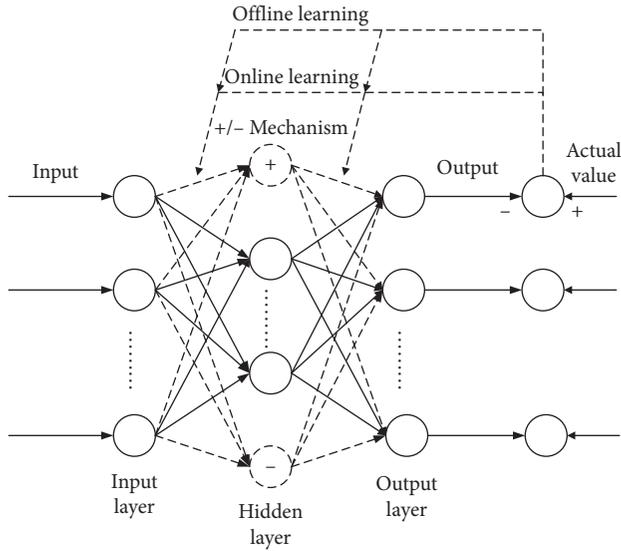


FIGURE 3: Structure of BFNN.

follow the most successful position of neighbors for global search [24].

- (ii) *BFO*. The BFO is proposed by imitating the process of bird flock seeking food.

Biological foundation: the process of bird flock seeking food.

The purpose of this operation: the BFO has the function of global optimum (preventing local optimum), and it is used to optimize the weights of the NN.

3.2. *A New Weight Adjustment Method of Neurons*. The differences between the weight adjustment method of BFNN and typical BPNN are as follows:

- (i) *Typical BPNN*. The weight adjustment formulas of the typical BPNN can be found in literature [25, 26].
- (ii) *BFNN*. A new adjustment method is used to update the weights of neurons.

Biological foundation: the medium between two neurons changes when the neurons are continuously activated.

The purpose of this operation: the new weight adjustment method of the BFNN can add data one by one for online learning, and it has better capabilities of local optimum.

3.3. *An Online Adjustment Method of the Number of Hidden Layer Neurons*. The differences between the adjustment method of the number of the hidden layer of BFNN and typical BPNN are as follows:

- (i) *Typical BPNN*. The typical BPNN obtains the number of the hidden layer neurons by empirical formula and repeated tests. Once the number of the hidden layer neurons is found, it cannot be increased or decreased.

- (ii) *BFNN*. The BFNN obtains the number of the hidden layer neurons by empirical formula, and then the number of neurons is adjusted by the online adjustment method.

Biological foundation: the number of biological neurons is not fixed and can be increased or decreased.

The purpose of this operation: the BFNN can increase or decrease the number of the hidden layer neurons online after the adjustment method is used.

4. Principle of BFNN

4.1. *Feedforward Calculation*. The process of feedforward calculation includes three parts: determining structure, preprocessing data, and calculating output values.

4.1.1. *Determining Structure*. The number of input and output layer neurons was manually determined on the basis of dataset information.

The calculation formulas of the number of initial hidden layer neurons are as follows:

$$\begin{aligned} l_1 &= m - 1, \\ l_2 &= \sqrt{m + n} + 5, \\ l_3 &= \log_2 m, \end{aligned} \quad (3)$$

$$l = \frac{l_1 + l_2 + l_3}{3},$$

where l is the number of hidden layer neurons, m is the number of input layer neurons, and n is the number of output layer neurons.

4.1.2. *Preprocessing Data*. The dataset is normalized, and the calculation formula is as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (4)$$

where x is the original data, x' is the normalized data, x_{\max} is the maximum value of x , and x_{\min} is the minimum value of x .

The normalized dataset was divided into four parts: training input, training output, test input, and test output.

4.1.3. *Calculating Output Values*. Let $O_1(k), O_2(k), \dots, O_m(k)$ be the NN input, namely, $O_i(k)$, where i is the input layer neuron number. For the convenience of writing, the sample number is omitted. The input data pass through the hidden layer to the output layer. The input of hidden layer is $\text{net}_j(k) = \sum_{i=1}^m w_{ij}(k)O_i(k)$, where w_{ij} is the weight between input layer and hidden layer. The output of hidden layer is $O_j(k) = f(\text{net}_j(k))$, where f is excitation function and j is the hidden layer neuron number. The excitation function is a hyperbolic tangent function, and the calculation formula is $y = (2/1 + e^{-2x}) - 1$. The input of output layer is $\text{net}_k(k) = \sum_{j=1}^q w_{jk}(k)O_j(k)$, where q is the hidden

layer neuron number and k is the output layer neuron number. The output layer excitation function is a linear excitation function, and the output of output layer neurons is $O_k = \text{net}_k$.

4.2. BFO. BFO is used to obtain a set of optimal weights of NN.

After the feedforward calculation, we need to determine if there is a known dataset. If yes, the BFO is used to train the NN, and the known data are mapped to the NN; if no, the algorithm flow jumps to Section 4.3.

The algorithm flow of the BFO is shown in Figure 4.

In BFO, a bird flock is a group of individuals. A bird is an individual, and it is a set of weights of NN in this paper. The predation effect is fitness. Flight times are the iteration times of BFO. Surrounding birds are the individuals within a certain distance. The overall predation effect of bird flock is average fitness.

The principle of BFO is as follows.

4.2.1. *Generating the Number of Initial Birds.* The accuracy and running speed of BFO will be considered by setting the appropriate number of birds in an initial flock. Increasing the number of birds will improve accuracy and increase time consumption of BFO and vice versa. BFO can evaluate dataset and automatically generate the appropriate number of birds based on dataset information. The algorithm for generating the number of the birds of initial flock is as follows.

Sort the value of each input vector from small to large, normalize them, and keep two decimal places. Add each different number to the abscissa and count the number of abscissa value of dataset into the ordinate. Remove ordinate points which are 0, connect adjacent points, and detect the peaks which are greater than the threshold b_1 in the graph, and the number of the peaks is denoted by peaknum_i . Set the threshold b_2 ; if $\text{peaknum}_i > b_2$, then $\text{classnum}_i = \text{peaknum}_i$; if $\text{peaknum}_i < b_2$, then $\text{classnum}_i = 1$. Calculate t classnum based on (10).

$$t \text{ classnum} = \prod_{i=1}^M \text{classnum}_i, \quad (5)$$

where t classnum is the number of classes of dataset and classnum_i is the number of classes on the i -th input vector of dataset.

The calculation formula of the initial bird flock is as follows:

$$\text{ininum} = \left(\prod_{i=1}^M \text{classnum}_i \right)^{2/M}, \quad (6)$$

where ininum is an integer between 10 and 200, and it is an integer multiple of 10.

4.2.2. *Bird Flock Search Strategy.* The bird flock search strategy is as follows:

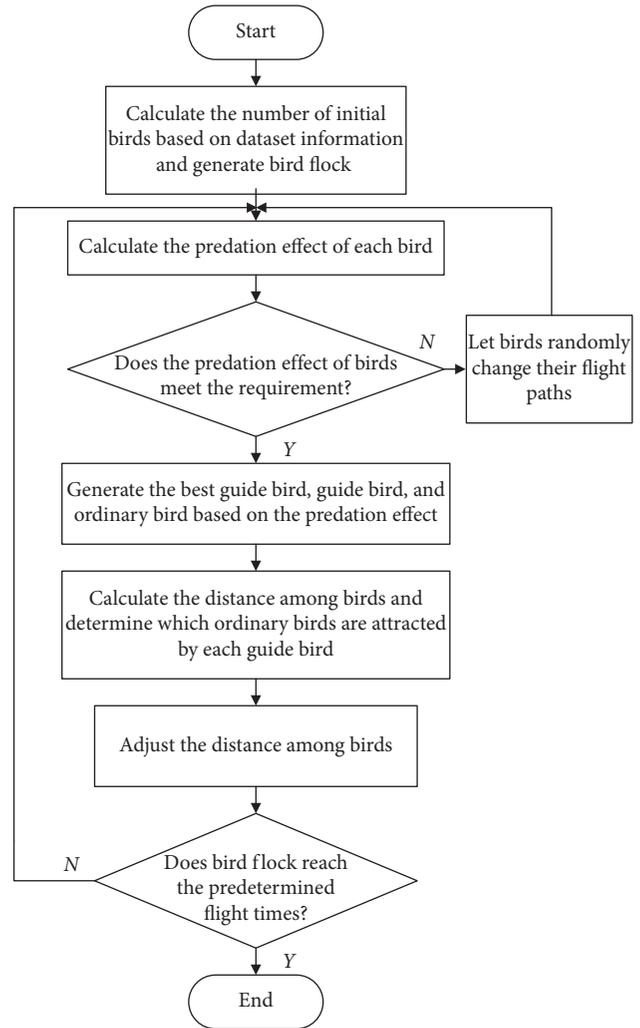


FIGURE 4: Flowchart of BFO.

- (i) The birds which have better predation effect stay, and the birds which have poorer predation effect fly away. The birds which fly away do not prey near the staying birds, so as to ensure the predation effect of the whole flock.

Set appropriate fitness function according to project. If it is a classification project, the number of correct classification can be used as fitness. If it is a prediction project, the following formula can be used to calculate fitness:

$$\text{fit} = \sum \frac{1}{|a \text{ value} - p \text{ value}|} \quad (7)$$

where fit is the fitness of individual, a value is the actual value of output, and p value is the predicted value of output.

The average fitness, meanfit , is calculated after running the algorithm. The individuals which are below meanfit will be reset randomly, and the new individuals are not in a range of the generated individuals. After the NN is fully trained, it can be known that most of the weights of the NN are distributed between 0.1–1. The value of weights are rounded, and they are kept one decimal place. After the above

operation, the new individual and the old individual cannot be the same. If the two are the same, the new individual will be reset randomly. The above operation ensures that the new individual does not appear within a certain range of the old individual. This operation needs to set the upper limit of reset times (2 in this paper) to prevent an endless loop.

The significance of strategy:

- (i) Ensure the individuals have the uniform distribution positions approximately in the dimensional space, and ensure each individual has a good initial fitness. This strategy can improve the global search performance and prevent the local optimum of NN.
- (ii) In a small area, the bird which has the best predation effect (called guide bird) will attract surrounding birds. The guide birds will attract nearby birds to search a smaller area together.

The inimum individuals are sorted from high to low based on the fitness. The first 20% individuals are taken as the guide individuals to attract their surrounding individuals.

- (iii) The best guide bird (namely, the global best individual) will attract the surrounding ordinary birds and other guide birds. A guide bird will only attract ordinary birds within a certain range. The guide birds do not attract each other (except the best guide bird).

The distance between two individuals is calculated as follows:

$$d = \sqrt{(x_1 - x'_1)^2 + \dots + (x_k - x'_k)^2}, \quad (8)$$

where d is the distance between two individuals, x is an element (weight) of the individual, x' is an element (weight) of another individual, and k is an element (weight) number.

- (iv) An ordinary bird will only be attracted by one guide bird. The distance between the attracted individual and the guide individual is calculated according to (8). The attracted individual can only be attracted by the closest guide individual.

The significance of strategies (ii), (iii), and (iv): the guide bird strategy can improve the local search accuracy of the algorithm. The conflict avoidance strategy is used to ensure the stable operation of the algorithm.

- (v) Flight and predation of birds: the flight and predation of birds correspond to the running and optimization of the algorithm.

All the guide individuals are a group, and the optimization formulas are as follows. Each guide individual and the guided individuals are a group, and the optimization formulas are the same as the former.

$$\begin{aligned} x1 &= C1 \cdot \text{chrom}, \\ x2 &= C2 \cdot \text{rand} \cdot (\text{guichrom} - \text{chrom}), \\ \Delta\text{chrom} &= x1 + x2, \\ \text{chrom}' &= \text{chrom} + \Delta\text{chrom}, \end{aligned} \quad (9)$$

where $C1$ and $C2$ are weighting coefficients, chrom is the original individual, rand is a random number between 0-1, guichrom is the guide individual, Δchrom is the adjustment amount of the individual, and chrom' is the new generated individual.

- (vi) Obtaining the best bird: set the upper limit of iteration times and run the algorithm. When the iteration times reach the predetermined value, the algorithm stops.

The individual which has the highest fitness will be recorded in the entire computation process, and the individual is a group of the best weights of BFNN.

4.3. New Weight Adjustment Method. After the algorithm in the above sections has been completed, a new neuron weight adjustment method is used to continue training the NN.

The adjustment formulas of weights between hidden layer and output layer are as follows:

$$\begin{aligned} \Delta w_{jk}(k) &= \eta \cdot \text{sqrt}(e(k) \cdot \Delta O_j(k)), \\ e(k) &= d(k) - y(k), \end{aligned} \quad (10)$$

where Δw is the adjustment amount of weight, j is the hidden layer number, k is the output layer number, η is the learning rate (it is an empirical value between 0.01–0.5), d is the expected output matrix, y is the actual output matrix, e is the difference matrix between the expected output and the actual output (output layer information), and ΔO_j is the difference matrix between the current and previous hidden layer output (hidden layer information).

The adjustment formulas of weights between input layer and hidden layer are as follows:

$$\begin{aligned} \Delta w_{ij}(k) &= \eta \cdot \text{sqrt}(\delta_j(k) \cdot \Delta O_i(k)), \\ \delta_j(k) &= \left(\sum_{k=1}^L e(k) w_{jk}(k) \right) f'(\text{net}_j(k)), \end{aligned} \quad (11)$$

where Δw is the adjustment amount of weight, i is the input layer number, j is the hidden layer number, η is the learning rate (it is an empirical value between 0.01–0.5), δ is the local gradient (hidden layer information), ΔO_i is the difference matrix between the current and previous input layer output (input layer information), f' is used to obtain the derivation of the excitation function, net_j is the input of hidden layer, and sqrt is used to obtain the square root of the nonnegative part of a number.

The weight calculation formula is as follows:

$$w(k) = w(k-1) + \Delta w(k). \quad (12)$$

4.4. Online Adjustment of the Number of Hidden Layer Neurons. After the algorithm of Section 4.3 starts to run, the following method can be used to increase or decrease neurons online.

4.4.1. Method of Increasing Hidden Layer Neuron.

Principle: the initial value of the weight of this NN is a random number which is uniformly distributed between -1 and 1 , namely, its absolute value is a random number between 0 and 1 . All weights are grouped into four intervals: $0-0.25$, $0.25-0.5$, $0.5-0.75$, and $0.75-1$. The number of weights in each interval accounts for about 25% . After many datasets were tested, the following rules were found: when the number of hidden layer neurons is the best, the training will cause most initial weights to be adjusted smaller. When the hidden layer neurons are lacking, the training will cause some weights to be adjusted larger.

The above rules can be used to determine if the hidden layer neurons are insufficient.

Concrete operation: after the algorithm of Section 4.3 starts to run, determine if the number of weights (greater than 0.75) is greater than 25% (initial ratio). If yes, it means that some neurons are too active, namely, the hidden layer neurons are insufficient, and a hidden layer neuron will be added; if no, the number of neurons is unchanged. The new added weights are set to some small values, and they are uniformly distributed random numbers between -0.1 and 0.1 . Continue to train k (100 in this paper) times and repeat the above judgment operation.

4.4.2. Method of Decreasing Hidden Layer Neuron.

Principle: when the number of the hidden layer neurons is the best, after the NN is fully trained, each hidden layer neuron is fully activated, namely, the weight value of each neuron will not be too low. If it is too low (less than 0.001), it means that some neurons are not fully activated and they belong to redundant neurons.

The above rules can be used to determine if the hidden layer neurons are excessive.

Concrete operation: after the algorithm of Section 4.3 starts to run, determine if there are neurons whose weight mean is less than 0.001 . If yes, these neurons are redundant neurons, and delete a neuron whose weight mean is the lowest; if no, the number of neurons is unchanged. Continue to train k times and repeat the above judgment operation.

5. Waste Type Identification Method Based on BFNN

The flowchart of waste type recognition algorithm based on BFNN is shown in Figure 5.

The general operation process of the algorithm is as follows:

- (i) Dataset acquisition: the waste images were analyzed and the feature vectors were extracted (see Section 2).
- (ii) Feedforward calculation: it includes three parts: determining structure, preprocessing data, and calculating output values (see Section 4.1).
- (iii) BFO: the BFO was used to learn a known dataset, and the main purpose of the BFO was to find the

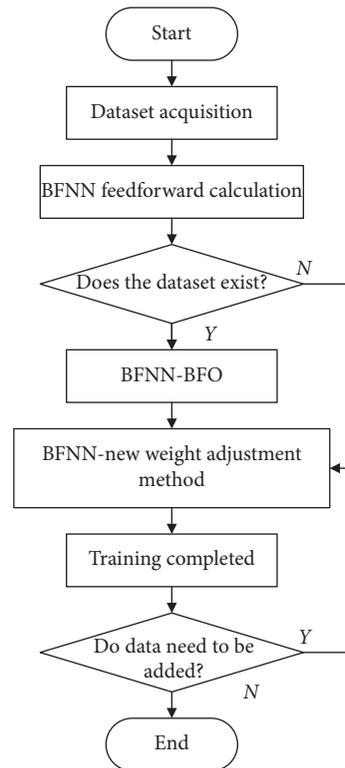


FIGURE 5: Flowchart of waste type recognition algorithm based on BFNN.

global optimum (preventing local convergence) (see Section 4.2).

- (iv) New weight adjustment method: a new weight adjustment method was used to learn data, and the main purpose of this method was to find the local optimum (improving accuracy) (see Section 4.3).
- (v) Online adjustment method of the number of hidden layer neurons: the hidden layer neurons could be increased and decreased online by this method (see Section 4.4).
- (vi) Training completed: after the training, the BFNN could identify the types of waste or continue to add data online.

6. Experimental Verification and Result Analysis

6.1. Iteration Times. The number of training samples of dataset was set to 400 . According to the calculation of Section 4.2.1, the initial number of the individuals of BFO was 100 .

The same number of initial individuals was set to 100 , and the same iteration times was set to 100 . Run typical GA, typical PSO, and BFO 10 times and obtain the mean, and the experimental results are shown in Figure 6.

The data in Figure 6 are the best fitness of each generation. In order to observe easily that the data in Figure 6 are enlarged ($\times 10000$), the calculation formula of the fitness of the GA, the PSO, and the BFO is (12).

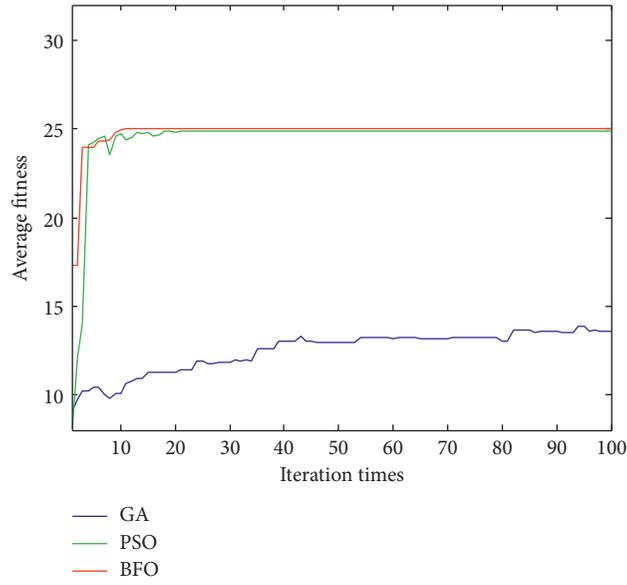


FIGURE 6: Comparison of iterative effects of different algorithms.



FIGURE 7: Some successful recognition result images of cloth.

TABLE 1: Comparison of recognition results after 400 samples are used.

	Hidden layer neurons	Recognition accuracy						Time consumption
		Paper (%)	Plastic (%)	Glass (%)	Cloth (%)	Nonrecyclables	Total	
BPNN	10	80.9	44.4	47.9	52.3	79.4	60.8	1.25
GA-BPNN	10	72.5	32.8	54.0	61.9	66.9	61.4	2.70
PSO-BPNN	10	81.9	58.5	43.0	61.4	77.0	64.2	2.64
BFNN	11	87.6	77.2	72.9	84.4	85.3	81.4	3.41

TABLE 2: Comparison of recognition results after 200 samples are supplemented online.

	Hidden layer neurons	Recognition accuracy						Time consumption (s)
		Paper (%)	Plastic (%)	Glass (%)	Cloth (%)	Nonrecyclables (%)	Total (%)	
BPNN	10	74.8	47.3	44.9	56.9	72.0	60.1	1.12
GA-BPNN	10	73.0	27.8	60.5	62.1	69.4	58.8	2.57
PSO-BPNN	10	81.4	41.6	55.4	58.2	67.4	60.9	2.50
BFNN	10	87.6	76.9	72.0	84.5	85.3	81.0	2.83

TABLE 3: Confusion matrix for waste identification.

		Predicted class				
		Cloth	Paper	Plastic	Glass	Nonrecyclables
Actual class	Cloth	30	0	1	0	2
	Paper	0	27	11	1	0
	Plastic	0	2	40	4	2
	Glass	0	1	0	32	5
	Nonrecyclables	0	2	4	0	36

As shown in Figure 7, in terms of iteration times, the BFO has the best fitness after 10 iterations, and the iteration time is less than that of the GA and the PSO. In terms of accuracy, the BFO is slightly better than the PSO and obviously better than the GA.

6.2. Recognition Accuracy and Time Consumption. 400 samples were used to train the BFNN and comparison algorithms. After the training, 200 unknown samples were identified, and each algorithm ran 10 times for obtaining mean value. The experimental results are shown in Table 1.

Here, hidden layer neurons are short for the number of hidden layer neurons.

As shown in Table 1, in terms of recognition accuracy, the recognition accuracy of the BFNN is 81.0%, which is higher than the comparison algorithms and meets the actual needs. In terms of running time, the BFNN is close to the typical GA-BPNN and the typical PSO-BPNN and higher than the BPNN.

On the basis of the above experiments, 200 samples were supplemented online. After training, the NNs would continue identify the types of waste, and each algorithm ran 10 times for obtaining mean value. The experimental results are shown in Table 2.

As shown in Table 2, after the 200 samples have been supplemented online, in terms of recognition accuracy, the recognition accuracy of the BFNN is slightly higher than the

former in Table 1, and the BFNN can effectively supply data. In terms of running time, the time consumption of the BFNN is slightly higher than the former in Table 1. In terms of the number of hidden layer neurons, a hidden layer neuron is added online.

6.3. Confusion Matrix. A confusion matrix for the best case of BFNN is shown in Table 3, where the sum of each row represents the number of samples of an actual class, and the sum of each column represents the number of samples of a predicted class.

Table 3 shows that the recognition effect of cloth is the best and the recognition effect of paper is poor, where some papers were misjudged as plastic. The analysis of the above recognition results is as follows.

6.3.1. Some Successful Recognition Result Images and Analysis. Some successful recognition result images of cloth are shown in Figure 7.

The feature information of Figure 7 is shown in Tables 4 and 5, where these numbers (1–15) are characteristic information number (see Section 2.1 for details).

As shown in Tables 4 and 5, for each individual image of cloth image (successful), the number of pixels concentrated in a certain color level range is significantly higher than others, namely, the color is relatively monotonous, which is consistent with the intuitive experience of observing

TABLE 4: The feature information (1–8) of Figure 7.

	Input feature vector number of NN							
	1	2	3	4	5	6	7	8
Figure 7(a)	0	1	6072	25503	8424	1896	4266	5573
Figure 7(b)	0	158	9659	19705	10478	0	105	9443
Figure 7(c)	8	525	10411	20218	8838	2	338	10133
Figure 7(d)	1711	1106	4482	21740	10961	1114	1713	5011

TABLE 5: The feature information (9–15) of Figure 7.

	Input feature vector number of NN						
	9	10	11	12	13	14	15
Figure 7(a)	20419	7846	155	5787	6376	25812	1870
Figure 7(b)	20295	10157	3031	4377	6571	23346	2675
Figure 7(c)	21007	8520	1460	4736	7492	20893	5419
Figure 7(d)	21060	11102	33	2760	5554	28140	3513

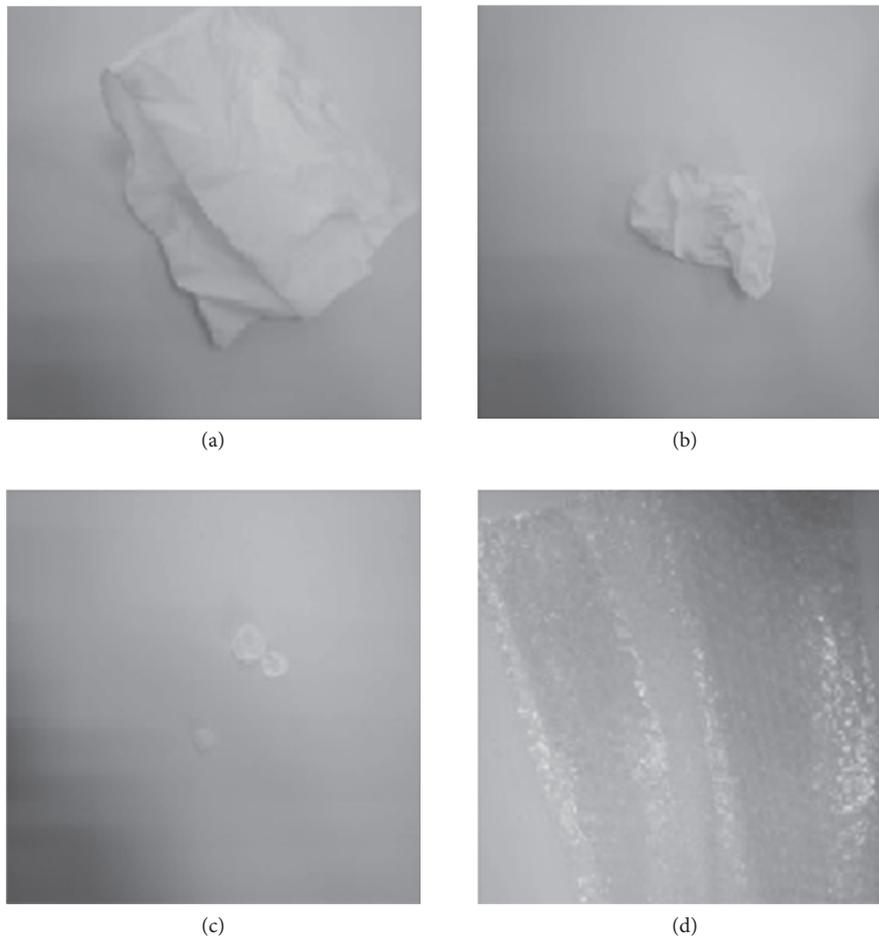


FIGURE 8: Some unsuccessful recognition result images of paper and comparison images: (a) toilet paper; (b) toilet paper; (c) white plastic gears; (d) packaging plastic.

Figure 7. Because the characteristics of cloth images are obvious, the NN has the highest recognition accuracy for cloth.

6.3.2. *Some Unsuccessful Recognition Result Images and Analysis.* Some unsuccessful recognition result images of paper and comparison images are shown in Figure 8.

TABLE 6: The feature information (1–8) of Figure 8.

	Input feature vector number of NN							
	1	2	3	4	5	6	7	8
Figure 8(a)	0	1151	14478	24771	0	0	1494	14415
Figure 8(b)	0	1467	16013	22520	0	0	1691	16634
Figure 8(c)	0	2887	9438	27668	7	0	3191	10972
Figure 8(d)	0	865	12251	26619	265	0	1015	12213

TABLE 7: The feature information (9–15) of Figure 8.

	Input feature vector number of NN						
	9	10	11	12	13	14	15
Figure 8(a)	24088	3	0	1634	16705	21661	0
Figure 8(b)	21675	0	0	1704	18901	19395	0
Figure 8(c)	25832	5	0	3166	15647	21187	0
Figure 8(d)	26492	280	0	742	14416	24671	171

The feature information of Figure 8 is shown in Tables 6 and 7, where these numbers (1–15) are characteristic information number (see Section 2.1 for details).

As shown in Tables 6 and 7, for each individual image of paper image (unsuccessful), the number of pixels concentrated in a certain color level range is significantly higher than others, namely, the color is relatively monotonous. However, the color range of aggregated pixels of the paper images (unsuccessful) is similar to the plastic images, namely, the color of the paper images (unsuccessful) and the plastic images is similar, which is consistent with the intuitive experience of observing Figure 8. Because of the above reasons, the recognition effect of paper by NN is relatively poor, and some papers are misjudged as plastic.

7. Conclusion

There are some problems of NN for waste type identification, such as image feature extraction, local optimum, accuracy, and obtaining the number of hidden layer neurons. In this paper, the features of waste images were extracted; a BFO, a new neuron weight adjustment method, and a hidden layer neuron online adjustment method were proposed. The experimental results show that the recyclables (paper, plastic, glass, and cloth) and nonrecyclables can effectively be identified by the waste type identification method based on the BFNN, and the recognition accuracy is 81% which meets actual needs.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Scientific Research Foundation of Tangshan Normal University of China (grant no. 2019A03).

References

- [1] V. C. Costa, F. W. B. Aquino, C. M. Paranhos, and E. R. Pereira-Filho, "Identification and classification of polymer e-waste using laser-induced breakdown spectroscopy (LIBS) and chemometric tools," *Polymer Testing*, vol. 59, pp. 390–395, 2017.
- [2] S. P. Gundupalli, S. Hait, and A. Thakur, "Multi-material classification of dry recyclables from municipal solid waste based on thermal imaging," *Waste Management*, vol. 70, pp. 13–21, 2017.
- [3] O. Adedeji and Z. Wang, "Intelligent waste classification system using deep learning convolutional neural network," *Procedia Manufacturing*, vol. 35, pp. 607–612, 2019.
- [4] M. Mundt, S. Majumder, S. Murali, P. Panetsos, and V. Ramesh, "Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11196–11205, Long Beach, CA, USA, June 2019.
- [5] R. R. Karri and J. N. Sahu, "Process optimization and adsorption modeling using activated carbon derived from palm oil kernel shell for Zn (II) disposal from the aqueous environment using differential evolution embedded neural network," *Journal of Molecular Liquids*, vol. 265, pp. 592–602, 2018.
- [6] L. Y. Jun, R. R. Karri, N. M. Mubarak et al., "Modelling of methylene blue adsorption using peroxidase immobilized functionalized Buckypaper/polyvinyl alcohol membrane via ant colony optimization," *Environmental Pollution*, vol. 259, Article ID 113940, 2020.
- [7] I. Kouchami-Sardoo, H. Shirani, I. Esfandiarpour-Boroujeni et al., "Prediction of soil wind erodibility using a hybrid Genetic algorithm– artificial neural network method," *Catena*, vol. 187, p. 10, Article ID 104315, 2019.
- [8] P. T. Kalaivaani, T. Akila, M. M. Tahir, M. Ahmed, and A. Surendar, "A novel intelligent approach to simulate the blast-induced flyrock based on RFNN combined with PSO-flyrock based on RFNN combined with PSO," *Engineering with Computers*, vol. 36, no. 2, pp. 435–442, 2019.
- [9] T. Qasim, M. Zia, Q.-A. Minhas et al., "An ant colony optimization based approach for minimum cost coverage on 3-D grid in wireless sensor networks," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1140–1143, 2018.
- [10] H. Hamdi, C. Ben Regaya, and A. Zaafouri, "Real-time study of a photovoltaic system with boost converter using the PSO-RBF neural network algorithms in a MyRio controller," *Solar Energy*, vol. 183, pp. 1–16, 2019.
- [11] A. Nezhadali and S. Sadeghzadeh, "Experimental design-artificial neural network-genetic algorithm optimization and computer-assisted design of celecoxib molecularly imprinted polymer/carbon nanotube sensor," *Journal of Electroanalytical Chemistry*, vol. 795, pp. 32–40, 2017.
- [12] S. H. Lee, C. S. Chan, and P. Remagnino, "Multi-organ plant classification based on convolutional and recurrent neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4287–4301, 2018.
- [13] Z. Zhou, G. Zhao, R. Kijowski, and F. Liu, "Deep convolutional neural network for segmentation of knee joint

- anatomy," *Magnetic Resonance in Medicine*, vol. 80, no. 6, pp. 2759–2770, 2018.
- [14] Y. Sun, Y. Yu, and W. Wang, "Moiré photo restoration using multiresolution convolutional neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4160–4172, 2018.
- [15] H. Hao, J. Zhang, Q. Zhang, L. Yao, and Y. Sun, "Improved gray neural network model for healthcare waste recycling forecasting," *Journal of Combinatorial Optimization*, vol. 38, pp. 1–18, 2019.
- [16] M. Peng, D. Lv, D. Xiong, W. Shen, W. Song, and R. Tan, "Facile preparation of a ZnO/SnO₂-based gas sensor array by inkjet printing for gas analysis with BPNN," *Journal of Electronic Materials*, vol. 48, no. 4, pp. 2373–2381, 2019.
- [17] W. Wang, S. Zhao, L. Jiao et al., "Estimation of PM_{2.5} concentrations in China using a spatial back propagation neural network," *Scientific Reports*, vol. 9, no. 1, p. 10, Article ID 13788, 2019.
- [18] W. Zhang, S. Zhang, S. Zhang, D. Yu, and N. Huang, "A novel method based on FTS with both GA-FCM and multifactor BPNN for stock forecasting," *Soft Computing*, vol. 23, no. 16, pp. 6979–6994, 2019.
- [19] X.-y. Chen, Z. Chen, and Y. Zhao, "Numerical research on virtual reality of vibration characteristics of the motor based on GA-BPNN model," *Neural Computing and Applications*, vol. 29, no. 5, pp. 1343–1355, 2018.
- [20] D. Fichou and G. E. Morlock, "Powerful artificial neural network for planar Chromatographic image evaluation, shown for denoising and feature extraction," *Analytical Chemistry*, vol. 90, no. 11, pp. 6984–6991, 2018.
- [21] D. Yang, H. B. Zhao, Z. Long et al., *MATLAB Image Processing Example and Detailed Explanation*, Tsinghua University Press, Beijing, China, 2013.
- [22] G. G. Xu and Y. Jia, *Matlab Realization of Pattern Recognition and Intelligent Calculation*, Beihang University Press, Beijing, China, 2012.
- [23] Y. X. Zhang, A. J. Liu, X. F. Pan, S. He, C. Gong et al., "A generalization belief propagation decoding algorithm for polar codes based on particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–10, Article ID 606913, 2014.
- [24] R. R. Karri and J. N. Sahu, "Modeling and optimization by particle swarm embedded neural network for adsorption of zinc (II) by palm kernel shell based activated carbon from aqueous environment," *Journal of Environmental Management*, vol. 206, pp. 178–191, 2018.
- [25] M. Chen, *MATLAB Neural Network Principles and Examples*, Tsinghua University Press, Beijing, China, 2013.
- [26] B. Yan, Y. Cui, L. Zhang et al., "Beam structure damage identification based on BP neural network and support vector machine," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–8, Article ID 850141, 2014.