KRAWTCHOUK MOMENT AND PARTICLE SWARM OPTIMIZED BP NEURAL NETWORK TO RECOGNIZE RICE PLANTHOPPER

/

基于 Krawtchouk 矩和 PSO 神经网络的稻飞虱识别研究

Assoc. Prof. Ph.D. Xiuguo Zou *[1], MEE. Stud. Siyu Wang [2], Assoc. Prof. Ph.D. Yan Qian [1], MAE. Stud. Shuaitang Zhang [1]

[1] College of Engineering, Nanjing Agricultural University / China;

[2] School of environmental science and Engineering, Nanjing University of Information Science and Technology / China

Tel: +862558606585; E-mail: xiuguozou@gmail.com

Keywords: *image processing, Krawtchouk moment, particle swarm optimized BP neural network, rice planthopper recognition*

ABSTRACT

Aimed at the problem of unreasonable spraying pesticide in paddy, machine vision can be used in the field identification technology of rice planthoppers. At first, gray processing of the planthopper images, the Gaussian algorithm, OTSU and mathematical morphology were applied for pre-processed images. Then Krawtchouk moment was used for features extraction, and the parameters were selected to optimize the particle swarm optimized neural network for training and testing. Experiments were implemented on MATLAB to train and test 100 Sogatella samples, 100 striatellus samples, and 100 brown planthopper samples. The results show that by using Krawtchouk moment to extract the shape features of the planthopper images and selecting parameters to improve particle swarm optimized BP neural network algorithm, the recognition rate reached 95%. Therefore, it can be used as the planthopper recognition algorithm.

摘要

针对大田中稻飞虱图像纹理和颜色效果不好的问题，研究了基于 Krawtchouk 矩提取形状特征值对稻飞虱进行分类识别。先对稻飞虱图像灰度化后用高斯算法滤波，再用大津法二值化，最后用数学形态学去噪；对预处理后的二值图像先采用 Krawtchouk 矩提取特征值，再用参数选择改进 PSO 优化神经网络的算法进行训练和测试。实验对白背飞虱、灰飞虱和褐飞虱各 100 个样本进行了训练和测试，结果表明 Krawtchouk 矩提取稻飞虱图像形状特征值后，用参数选择改进粒子群优化 BP 神经网络算法的整体识别率达到了 95%，而且收敛性很好，可以作为稻飞虱的识别算法。

INTRODUCTION

Aimed at the problem of unreasonable spraying pesticide in paddy, machine vision was used in the field recognition technology of rice planthoppers (*Finbarr G. H., 2015; Dale G. B., 2012)*. Some researchers of Nanjing Agricultural University have already used machine vision method to try to identify rice planthopper (*Zhao S., 2009; Liu D., 2012; Zou X., 2013*). Compared to the extraction of the texture features or color features, the planthopper shape feature extraction is easy and efficient, thus it plays an important role in real-time recognition. In this paper, the Krawtchouk invariant moment is adopted to extract the shape features of the planthopper. Krawtchouk moment is a discrete orthogonal moment proposed by (*Yap P. T., 2003*). Due to rotation and scale invariance of the moment variables, it is highly helpful for shape extraction of images. To the best of our knowledge, Krawtchouk moment has not been introduced to the recognition of insects such as planthopper before. When it is used for the description of the target images, the moment variable is the ideal feature of the planthopper images, because it provides an approach to image changes caused by the target's scale and position changes (*Yang L., 2008; Wang C., 2016*).

The PSO (Particle Swarm Optimization) algorithm is a fast search global optimization algorithm, which can return good optimization results. Like most evolutionary computation methods (e.g. genetic algorithm and ant colony optimization algorithm), PSO is also based on the swarm. But when the particles converge to the optimal positions, the fast convergence effect will occur among the particle swarm. As a result, it may be stuck in local extremum or cause premature convergence and low accuracy of results. Therefore, various countermeasures have been proposed (*Fang F., 2013*). In this paper, by selecting parameters, the particle swarm optimized BP neural network algorithm is improved for planthopper recognition.

MATERIALS AND METHODS

Image collection

The planthopper images were captured by the SunTime200A USB industrial camera equipped with the SunTimeT100 industrial lens, providing a focal length of 15:1. Three planthopper images are shown in Figure 1. In the next step, the images were sent to the computer for storage.

(a)Sogatella (b) Striatellus (c) Brown planthopper

Fig 1 - Sample images of rice planthoppers

Gray processing and Binarization

Let *R, G, B* denote the gray levels of the red, green and blue color components, and Y denotes the image brightness. Defining Y component as the gray level of the image pixels is suited for the human eye's perception of colors. The equation for converting color pixels to gray pixels is given in Equation 1.

$$Y = 0.299R + 0.587G + 0.114B \tag{1}$$

Binarization was done via OTSU, which is also known as the maximum between-cluster variance method commonly used to efficiently and easily convert gray images to binary images. By setting up a threshold t and according to the gray levels, OTSU classifies image pixels into two types: C_0 for 0 to t, and C_1 for t+1 to $l-1$, where l denotes the gray level of the image. The optimal threshold, t_0, can be determined by maximizing the equivalent decision criterion function of t in Equation 2 (*Wang H., 2012*).

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2} \quad \eta = \frac{\sigma_B^2}{\sigma_T^2} \quad \kappa = \frac{\sigma_T^2}{\sigma_W^2} \tag{2}$$

where: λ, η, κ - the equivalent decision criterion function of t; t-OTSU threshold;

$\sigma_W^2, \sigma_B^2, \sigma_T^2$ denote the between-cluster, inter-cluster and total variance, respectively.

Gaussian filtering

The Gaussian filter is a linear smooth filter commonly used in image denoising. In this filter, a template is used to scan each pixel of the image, and the pixel at the center of the template is replaced with the weighted average of the gray levels of neighboring pixels. The form is given in Equation 3 (*Wan C., 2017*).

$$E(x, y) = e^{-D^2(x,y)/2\sigma^2} \tag{3}$$

where: E- the weighted average of the gray levels of neighboring pixels; D- the radius function of Gaussian filter; x - distance from the origin in the horizontal axis; y - distance from the origin in the vertical axis; σ- standard deviation of the Gaussian distribution;

$D(x,y)$ denotes the fuzzy radius $r^2 = D^2(x,y) = x^2 + y^2$. The filter that takes the Gaussian function as the weight function (W) is known as the Gaussian filter and its weight function is given in Equation 4.

$$W(r) = \frac{1}{\sqrt{2\pi\sigma}} e^{-r^2/2\sigma^2} \tag{4}$$

Where: σ denotes the distribution parameter. r - the radius of Gaussian filter; The 6σ-width area at the center of the Gaussian curve occupies 99.73% of the entire area under the curve. Thus, after stipulating that the weight function in this interval is dominant for filtering, the signals can be filtered by changing the value of σ.

Morphology-based denoising

After binarization, there may be noises around the truncus of the planthoppers, such as feet, antennae, and bristles. Hence, the morphology-based operations should be performed, which mainly refer to the erosion and dilation operations.

Erosion operations of binary images

Erosion operations can eliminate boundary points of objects as well as small and meaningless objects. It is based on the concepts of structural element, which is template for shrinking the object boundary to improve the recognition accuracy of planthoppers. If tiny connections exist between two objects, a sufficiently

large structural element will separate the two objects via erosion operations. Let $X\Theta S$ express erosion, so the equation is given as follows (*Mao W., 2008*).

$$X\Theta S = \{\chi \mid S[\chi] \subseteq X\} \tag{5}$$

where:

X denotes the set of pixels of target image, and x is each pixel point in X. S is the structural element, and $S[\chi]$ is the set that each pixel point of S moves the distance from the pixel point to χ.

Dilation operations of binary images

In dilation operations, each point χ in the image X is expanded to $S[\chi]$. This operation is designed to merge the background points around the image into the object. If two objects are close to each other, then they can be connected via dilation operations, as it is very helpful in filling in the holes in the objects after image segmentation. Let $X \oplus S$ express the dilation operation, so the equation is given as follows (*Jufriadif N., 2017*).

$$X \oplus S = \{\chi \mid S[\chi] \cap \chi \neq \Phi\} \tag{6}$$

where: Φ is the empty set.

Features extraction method

The Krawtchouk moment invariant is derived from the Krawtchouk polynomial. The n^{th} weighted Krawtchouk polynomial at the discrete point x is defined as follows (*Yap P. T., 2003*).

$$K_n(x;p,N-1) = \sqrt{\frac{\omega(x;p)}{\rho(n;p)}} F_1\left(-n,-x;-(N-1);\frac{1}{p}\right) \tag{7}$$

where:

$x, n = 0,1,2,\ldots,N, \quad N>0, \quad p\in(0,1).$

$F_1(a,b;c;z) = \sum_{v=0}^{n} \frac{(a)_v (b)_v}{(c)_v} \frac{z^v}{v!}$ is the hypergeometric function, a, b, c, z are the formal parameters of the function, v is the argument in the range $[0,n]$, $(a)_v$ denotes the reduced order power from $a+v+1$ to a. Definitions of $(b)_v$ and $(c)_v$ are similar to that of $(a)_v$.

$\omega(x;p) = \binom{N-1}{x} p^x (1-p)^{N-1-x}$ is the Binomial-distribution weight computation function.

$\rho(n;p)$ is the normalized coefficient and the equation is shown as follows:

$$\rho(n;p) = (-1)^n \left(\frac{1-p}{p}\right)^n \frac{n!}{-(N-1)_n} \tag{8}$$

With the above polynomial definition, the Krawtchouk moment can be defined according to the separability. An approximation of $f(x,y)$ is:

$$f(x,y) \cong \sum_{n=0}^{n_{max}} \sum_{n=0}^{m_{max}} Q_{nm} K_n(x;p_1,N-1) K_m(y;p_2,N-1) \tag{9}$$

According to Equation (9), the $(n+m)$th term of the Krawtchouk moment, Q_{nm}, is given as follows.

$$Q_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} K_n(x;p_1,N-1) K_m(y;p_2,N-1) f(x,y) \tag{10}$$

where Q_{nm} is the Krawtchouk moment invariant.

PSO algorithm

The PSO algorithm originates from modelling the preying of birds and is based on the concepts of swarm and fitness. Each particle in the swarm can be characterized by position and speed and represents a possible solution by position. The quality of the position is measured by fitness in the algorithm. A group of particles is randomly initialized and the optimal solution is determined via iterations (*Yap P. T., 2003*).

In H-dimensional space, there are M particles, and i= 0,1,2,…,M, M>0, then the position and speed of the i^{th} particle are shown as Equation 11 and Equation 12.

$$u_i = (u_{i1}, u_{i2}, ...u_{iH}) \tag{11}$$

$$s_i = (s_{i1}, s_{i2}, ...s_{iH}) \tag{12}$$

During each iteration, the particle updates the following two solutions by tracking two extremums.

The first is the optimal solution of the particle itself, which is called the individual extremum, and the equation is shown as follows:

$$pb_i = (pb_{i1}, pb_{i2}, ..., pb_{iH}) \tag{13}$$

The second is the optimal solution that the entire population has found the optimal solution, which is called the global extremum, and the equation is shown as follows:

$$gb = (gb_1, gb_2, ..., gb_H) \tag{14}$$

While searching these two optimal values, the particle updates its position and speed in H-dimensional space using the following equations:

$$u_{ih}^{\ k} = u_{ih}^{\ k-1} + s_{ih}^{\ k-1} \tag{15}$$

$$s_{ih}^{\ k} = Ts_{ih}^{\ k-1} + c_1 \cdot rand() \cdot (pb_{ih}^{\ k-1} - u_{ih}^{\ k-1}) + c_2 \cdot rand() \cdot (gb_h^{\ k-1} - u_{ih}^{\ k-1}) \tag{16}$$

where:

h = 0, 1, 2 ,…, H, $u_{ih}^{\ k}$ and $s_{ih}^{\ k}$ denote the h^{th} dimensional components of the position and speed of the particle i at each iteration time k, pb_{id}^{k} is the best individual position of the particle i and gb_{id}^{k} are the best position of global particle position i at each iteration time k, $rand()$ denotes the random number that follows (0,1) uniform distribution, T is the inertia weight, and c_1, c_2 are the acceleration coefficients.

Improved PSO neural network based on parameter selection

In PSO algorithm, when the particles converge to the optimal positions, the fast convergence effect will occur among the particle swarm, inclining it to be stuck in local extremum or causing premature convergence and low accuracy of results (*Zhang J., 2016*). So, the algorithms should be improved. In this paper, the PSO algorithm is improved via parameter selection. The improved algorithm is called Improved PSO. The parameters that need to be optimized include the maximum speed, two acceleration coefficients, and the constriction factor.

Step 1: Selection of the maximum speed.

The particle speed in Equation 16 is a random variable. The motion tracks generated by the particle position updating Equation 15 are uncontrollable, so the particles jump repeatedly in the problem space. To alleviate irregular jumping, the speed is usually bound by [-s_{max}, s_{max}]. A large s_{max} is helpful for global search, whereas a small s_{max} is helpful for local exploitation. But if s_{max} is too high, the motion tracks of particles may be irregular or even beyond the area of the optimal solutions. In this case, the algorithm can hardly converge and thus comes to a standstill. If s_{max} is too small, then the step of the particles may be also too small and the algorithm may be stuck in local extreme values. In this paper, s_{max} is set to 2.0 through the experiments. (*Hotaka Y., 2017*).

Step 2: Selection of acceleration coefficients.

The acceleration coefficients in Equation 16, c_1 and c_2, are used to make the particle move towards itself or the optimal position in the neighborhood, respectively. A time variable adaptive strategy is adopted: c_1 is linearly reduced from 2.5 to 0.5, and c_2 is linearly increased from 0.5 to 2.5 among evolutional generations.

Step 3: Selection of constriction factor.

If the PSO speed is updated according to Equation 17, then even though s_{max} and the two acceleration coefficients are selected properly, the particles are still likely to go beyond the problem space or even converge to the infinity, causing swarm explosion. The constriction factor A is properly selected to alleviate this problem.

The PSO algorithm with the constriction factor was proposed by (Kennedy, 1995), and the simplest version of the speed updating equation is given as follows.

$$s_{ih}^{\ k} = A\left[s_{ih}^{\ k-1} + c_1 \cdot rand() \cdot (pb_{ih}^{\ k-1} - u_{ih}^{\ k-1}) + c_2 \cdot rand() \cdot (gb_h^{\ k-1} - u_{ih}^{\ k-1}) \right] \quad (17)$$

where: A is the constriction factor, and the equation is shown as Equation (18).

$$A = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} \quad , \varphi = c_1 + c_2 , \varphi > 4 \quad (18)$$

The improved PSO optimized BP neural network algorithm is shown in Figure 2 (*Shi F., 2011*).

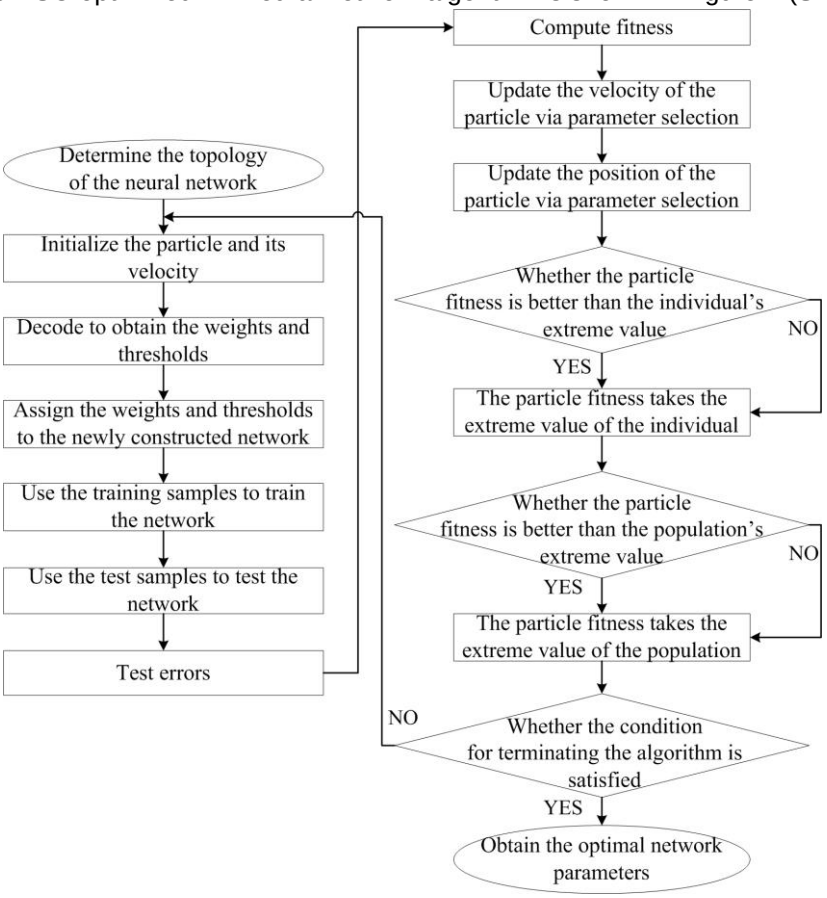


Fig. 2 - Flow chart of improved PSO optimized BP neural network

Planthoppers recognition using improved PSO optimized BP neural network algorithm

The steps for recognizing the planthoppers via the PSO optimized BP neural network algorithm are given as follows.

Step 1: Initialize a group of particles.

This step involves the size of the population, the neural network topology, random positions and speeds, acceleration coefficient, number of iterations, upper limit of errors, etc. For the BP neural network trained by PSO, the size of the population is set to 50. All the weights of the BP network and the thresholds of the hidden-layer nodes are defined as the initial positions of particles. Real numbers are used for particle coding, the initial number of nodes at the hidden layer is set to 5, and the largest number of nodes at the hidden layer is set to 20. The stepwise increase method is adopted for network training to increase the number of hidden-layer nodes until the requirements are satisfied. The maximum speed of particles is set to 2.0, and the number of iterations k is set to 2,000 (*Kim T. H., 2008*).

Step 2: Evaluate the fitness of each particle.

The fitness function is a performance measure of BP neural network. MSE of the neural network outputs is defined as the objective function, and its reciprocal is defined as the fitness function. (*Sun W., 2009*). The equation for computing function is shown as follows.

$$MSE = \frac{2}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} \left(d_{ij} - w_{ij} \right) \quad (19)$$

where: I is the number of output nodes, J is the number of training samples, d_{ij} is the expected output of the network, and w_{ij} is the actual output of the network.

Step 3: Update each particle's speed and position via parameter selection.

By using the optimization parameters (i.e. the maximum speed, two acceleration coefficients, and the constriction factor), each particle's speed and position can be computed and updated according to Equation 15, Equation 16 and Equation 17.

The speed of each particle should be checked to see if it is in the range [$-s_{max}$, s_{max}], whereas the position of each particle should be checked to see if it is in the range [$-u_{max}$, u_{max}] (*Lei, Y., 2013*).

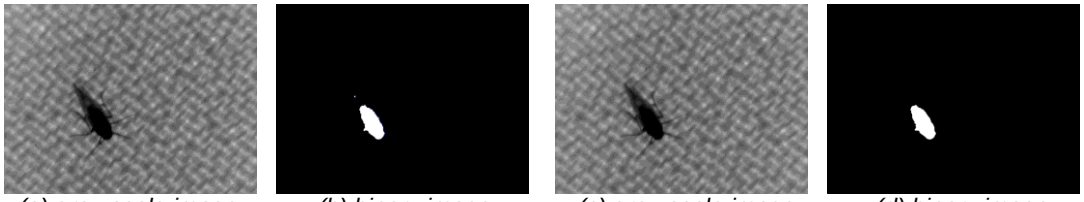
Step 4: Compare the fitness of each particle i with its previous extreme value pb_i. If it is better than pb_i, then it is defined as the current pb_i. Continue to compare it with the global extreme value gb. If it is better than gb, then i is defined as the current gb (*Shi B., 2010*).

Step 5: The number of evolution times is set to 100. If the fitness is sufficiently good or the preset maximum number of iteration is reached, then improved PSO neural network is constructed. Next, follow step 6, otherwise, increase the number of iteration times by 1 and jump to step 2.

Step 6: Use the improved PSO neural network to train and recognize the planthoppers.

RESULTS

After collecting planthopper images, gray processing, the Gaussian algorithm, OTSU and mathematical morphology were applied for pre-processed images. Figure 3 shows the gray-scale image and the binary image before and after Gaussian filtering.



(a) gray-scale image before Gauss filtering *(b) binary image before Gauss filtering* *(c) gray-scale image after Gauss filtering* *(d) binary image after Gauss filtering*

Fig. 3 - Gray-scale images and binary images before and after Gauss filtering

Figure 4 shows the binary images that were processed by Gaussian filtered algorithm, OTSU binarized algorithm and denoised algorithm using the morphology-based method in sequence.

Fig. 4 - Binary images after denoising

After binarization, 6 Krawtchouk moment features can be extracted from each of the three types of planthopper images in Figure 1. The extracted feature values are listed in Table 1.

Table 1

Extracted features of rice planthoppers by Krawtchouk moment

Features of Krawtchouk moment	Sogatella	Striatellus	Brown planthopper
Krawtchouk20	4.054529	2.849234	3.498444
Krawtchouk02	2.597408	1.400971	2.323513
Krawtchouk30	2.360537	0.938745	1.899877
Krawtchouk03	2.659690	1.555703	2.514412
Krawtchouk40	5.547901	4.558641	5.240641
Krawtchouk04	1.405166	1.158586	1.099641

The 300 collected image samples were used in the experiment, consisting of 100 Sogatella samples, 100 striatellus samples, and 100 brown planthopper samples. The first 240 images were chosen as the training samples and the rest 60 images that consist of 20 Sogatella images, 20 striatellus images, and 20 brown planthopper images were chosen as the test samples. The training and test outputs are classified into

three types, and the values for the Sogatella, striatellus and brown planthopper are set to 100, 010 and 001, respectively. The algorithm was implemented on MATLAB.

When the BP neural network is trained by PSO, the size of the population is set to 50, the maximum speed of the particle: s_{max}=2.0, and the number of iterations k is set to 2,000. The trained network obtains the weights and thresholds that satisfy the requirements, whereas the number of evolution times is set to 100.

The test and actual results are compared in Figure 5.

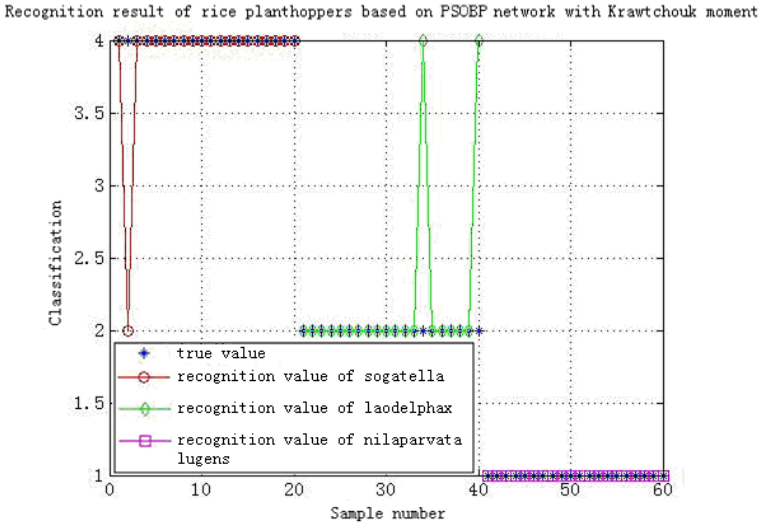


Fig. 5 - The actual value compared with test value by PSO optimized BP neural network

Figure 5 shows that the recognition rates for Sogatella, striatellus and brown planthopper are 95%, 90% and 100%, respectively. The overall recognition rate is up to 95%.

CONCLUSIONS

The collected planthoppers images can be used to yield the high-quality binary planthoppers images by gray processing, Gaussian filtering, OTSU binarization and morphology-based denoising. The images processed in this step facilitate features extraction and object recognition. The Krawtchouk moment-based extraction of shape features of planthopper images can represent global features and show good locality. Experiments prove that the recognition rate can be guaranteed if it is used for planthopper features extraction. In addition, if the Krawtchouk moment is used, only a small number of features will be extracted. So the computational complexity is low and helpful for real-time recognition. In PSO, particles may be stuck in local extremum or cause premature convergence and low accuracy of results. The parameter selection strategy is used to address this problem. Parameters are properly chosen to optimize the PSO algorithm and recognize planthoppers. The overall recognition rate is 95% while ensuring fast evolutionary rate and good convergence behavior. The proposed method is a good choice for real-time recognition of planthoppers.

ACKNOWLEDGEMENTS

This work was supported by China Postdoctoral Science Foundation (2015M571782), the Fundamental Research Funds for the Central Universities of China (KYTZ201661), and Jiangsu Agricultural Machinery Foundation (GXZ14002).

REFERENCES

[1] Carmine C., Luca P., Domenico G., (2017), Automatic target recognition of military vehicles with Krawtchouk moments, *IEEE Transactions on Aerospace and Electronic Systems,* Vol. 53, Issue 1, pp. 493-500, Piscataway, New Jersey/USA;

[2] Dale G. B., Kenneth G. S., (2012), Resurrecting the ghost of green revolutions past: The brown planthopper as a recurring threat to high-yielding rice production in tropical Asia, *Journal of Asia-Pacific Entomology,* Vol. 15, Issue 1, pp. 122-140, Suwon/South Korea;

[3] Fang F., Zhang Z., Zhang X., et al., (2013), Reduction in Activity/Gene Expression of Anthocyanin Degradation Enzymes in Lychee Pericarp is Responsible for the Color Protection of the Fruit by Heat and Acid Treatment, *Journal of Integrative Agriculture*, Vol. 12, Issue 9, pp. 1694-1702, Beijing/China;

[4] Finbarr G. H., Angelee F. R., Jagadish S. B., et al., (2015), Virulence of brown planthopper (Nilaparvata lugens) populations from South and South East Asia against resistant rice varieties, *Crop Protection*, Vol.78, Issue 11, pp. 222-231, Oxford/U.K.;
[5] Hotaka Y., Yoshikazu F., (2017), Parallel multi-population differential evolutionary particle swarm optimization for voltage and reactive power control in electric power systems, 56th Annual Conference of the Society of Instrument and Control Engineers of Japan, *IEEE*, Kanazawa/Japan;
[6] Jufriadif N., (2017), Edge detection on objects of medical image with enhancement multiple morphological gradient method, *4th International Conference on Electrical Engineering, Computer Science and Informatics*, *IEEE*, Yogyakarta/Indonesia;
[7] Kennedy, J., Russell C., (1995), Particle swarm optimization. *IEEE international conference on neural networks*, *Perth/ Australia;*
[8] Kim T.H., Maruta I. and Sugie T., (2008), Robust PID controller tuning based on the constrained particle swarm optimization, *Automatica*, Vol. 44, Issue 4, pp. 1104-1110,
[9] Lei Y., Xiaokang, D., Jianlei, K., (2013), Parameters optimization algorithms for improving the performance of obstacles identification in forest area, *INMATEH - Agricultural Engineering*, Vol. 40, Issue 2, pp. 43-52, Bucharest/Romania;
[10] Liu D., Zhao S., Ding W., et al., (2012), Identification method for rice planthoppers based on image spectral characteristics, *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 28, Issue 7, pp. 184-188, Beijing/China;
[11] Mao W., Zheng Y., Yuan Y., (2008), Locust information extraction using Hue and shape feature, *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 39, Issue 9, pp. 104-107, Beijing/China;
[12] Shi B., Li Y., Yu X., (2010), Long-term runoff forecast method based on dynamic adjustment particle swarm optimizer algorithm and Holt-Winters linear seasonal model, Transactions of the Chinese Society of Agricultural Engineering, Vol. 26, Issue 7, pp. 8-13, Beijing/China;
[13] Shi F., Wang H., Yu L., (2011), Matlab intelligent algorithm analysis of 30 cases, *Beijing University of Aeronautics and Astronautics Press*, Beijing/China;
[14] Sun W., Zhang W., Li X., (2009), Driving Fatigue Fusion Detection Based on T-S Fuzzy Neural Network Evolved by Subtractive Clustering and Particle Swarm Optimization, *Journal of Southeast University (English Edition),* Vol. 25, Issue 4, pp. 356-361, Nanjing/China;
[15] Wan C., Ye M., Yao C. et al., (2017), Brain MR image segmentation based on Gaussian filtering and improved FCM clustering algorithm, *10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, IEEE* ,Shanghai/China;
[16] Wang C., Li Z., (2016), Weed recognition using SVM model with fusion height and monocular image features, *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 32, Issue 15, pp. 165-174, Beijing/China;
[17] Wang H., Feng X., Li L., (2012), Detection algorithm of white foreign fibers based on improved two-dimensional maximum between-class variance method, *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 28, Issue 8, pp. 214-219, Beijing/China;
[18] Yang L., Ding W., Liu D., et al., (2008), Study of feature measuring and extraction for delphacidae images, *Journal of Agricultural Mechanization Research,* Vol. 30, Issue 6, pp. 180-183, Haerbin/China;
[19] Yap P. T., Paramesran R., Omg S. H., (2003), Image analysis by Krawtchouk moments, *IEEE Transaction on Image Processing*, Vol. 12, Issue 11, pp. 1367-1377, Piscataway, New Jersey /USA;
[20] Zhang J., Qi L. and Ji R., (2016), Cotton diseases identification based on rough sets and BP neural network, *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 47, Issue 7, pp. 35-41, Beijing/China;
[21] Zhao S., Ding W., Liu D., (2009), Rice hopper shape recognition based on Fourier descriptors, *Translations of the Chinese Society for Agricultural Machinery,* Vol. 40, Issue 8, pp. 181-184, Beijing/China;
[22] Zou X., Ding W., Liu D. et al., (2013), Recognition System of Rice Planthopper Based on Improved Hu Moment and Genetic Algorithm Optimized BP Neural Network, *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 44, Issue 6, pp. 222-226, Beijing/China.

The Information of the variables or parameters in the equations are listed in Table 2.

Table 2

Information of the variables or parameters in the equations

Name	Notation	Name	Notation
Constriction factor	A	Global extremum	gb
Blue color components	B	The best position of global particle position i	gb_{id}
The types of image pixels	C_0 C_1	The gray level of the image	l
The radius function of Gaussian filter	D	Individual extremum	pb_i
The weighted average of the gray levels of neighboring pixels	E	The best individual position of the particle i	pb_{id}
Hypergeometric function	F_1	The radius of Gaussian filter	r
Green color components	G	The speed of the ith particle	s_i
Dimensional space	H	The hth dimensional components of the speed of the particle i	s_{ih}
The number of output nodes in BP neural network	I	Maximum speed	s_{max}
The number of training samples in BP neural network	J	OTSU threshold	t
Krawtchouk polynomial	K	Optimal threshold	t_0
The number of the particles in PSO algorithm space	M	The position of the ith particle	u_i
Red color components	R	The hth dimensional components of the position of the particle i	u_{ih}
Structural element	S	The actual output of the neural network	w_{ij}
Inertia weight	T	The distance from the origin in the horizontal axis	x
Krawtchouk moment invariant	Q_{nm}	The distance from the origin in the vertical axis	y
The weight function of Gaussian filter	W	The equivalent decision criterion function of t	λ η κ
The pixels set of target image	X	The standard deviation of the Gaussian distribution	σ
Image brightness	Y	Between-cluster, inter-cluster and total variance	$\sigma_W^2, \sigma_B^2, \sigma_T^2$
Acceleration coefficients	c_1, c_2	Each pixel point in X	χ
The expected output of the neural network	d_{ij}	The Binomial-distribution weight computation function	ω
The digital image function	f		