IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 25, 2020, accepted June 28, 2020, date of publication July 7, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007745

Rapid Fine-Grained Classification of Butterflies Based on FCM-KM and Mask R-CNN Fusion

AIJIAO TAN, GUOXIONG ZHOU, AND MINGFANG HE
College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410004, China
Corresponding author: Guoxiong Zhou (51840157@qq.com)

This work was supported by the National Natural Science Foundation of China under Grant 61703441.

ABSTRACT Butterfly recognition is a key link in the field of animal and plant observation. In order to realize the location and recognition of butterflies by robot vision system in complex environment, rapid fine-grained classification of butterflies based on the FCM-KM and Mask R-CNN fusion was proposed. First, an adaptive image enhancement algorithm based on fuzzy sets optimized by FOA was used to realize the adaptive fuzzy enhancement of butterfly images in image pre-processing. Then, K-Means clustering algorithm optimized by dynamic population firefly algorithm based on chaos theory and max-min distance algorithm, FCM-KM, was used to determine the optimal clustering number K instead of manual parameter tuning. Finally, while effectively segmenting the butterfly images, the Softmax in Mask R-CNN was used to classify the butterfly images. The recognition accuracy of the trained model in the verification set was 83.62%. To verify the feasibility and effectiveness of the model in complex environment, the rapid fine-grained classification method of butterflies based on the FCM-KM and Mask R-CNN fusion was compared with CNN, Resnet and original Mask R-CNN. The experiment results show that the method proposed in this paper has a good classification effect on butterflies in complex environment.

INDEX TERMS Fine-grained butterfly classification, FCM-KM, FOA, Mask R-CNN.

I. INTRODUCTION

In recent years, with the rapid development of animal and plant observation, the images of animals and plants have proliferated, and the demand for target identification of animals and plants has also increased dramatically. Butterfly recognition is one of the important tasks. It can help natural scientists, biologists and other researchers in related research. Scientists can obtain the distribution of certain butterflies, count the population of butterfly species, monitor and evaluate the butterfly species and ecosystem, analyze the change of ecological environment and assist the butterfly protection by identifying the butterflies automatically. However, the natural environment is complex and harsh, and the image resolution is low. Butterfly subcategories usually have small inter-class differences, so that we often require small local differences to distinguish them. For example, Two-tailed Swallowtail and Three-tailed Swallowtail are very similar, and the differences between them mainly lie in the transverse bands of the forewings and the shapes of the hind wings, but the differences are relatively slight. Compared with inter-class differences, there are usually large intra-class differences in fine-grained image classification, including object attitude, lighting, scale, shielding, angle and background. Especially in the case that the data amount of each category is limited and there is no additional manual labeling information for butterfly parts, it is a very challenging task to realize fine-grained image classification based on weak supervision information.

During the collection and transmission of butterfly images, it is actually easy to be polluted by noise, which reduces the quality of butterfly images and blurs the butterfly images. Therefore, it is necessary to enhance butterfly images. It is conducive to improving the image quality, enhancing some features of the image areas and increasing the contrast of the image areas. At present, image enhancement algorithms include fuzzy processing, frequency domain method and spatial domain method [1]. Fuzzy enhancement processes the original images by fuzzification, then processes them by using various properties in the characteristic plane, and finally realizes the enhancement of the original images by fuzzy inverse transformation. The frequency domain method enhances the images through image transformation.

The associate editor coordinating the review of this manuscript and approving it for publication was Wenming Cao.

The spatial domain method mainly deals with the images directly, and the algorithms mainly include histogram transformation, histogram equalization, local gray level, edge extraction and smooth filtering. These methods all need to determine the crossover point and saturation point manually. Because of these shortcomings, the applications of traditional methods are limited. To deal with the problems that traditional image enhancement algorithms need a large amount of calculation, they lack applicability and their parameters need to be set manually by experience, combined with rapid optimization capability of fruit fly optimization algorithm, an adaptive butterfly image enhancement algorithm based on fuzzy sets optimized by FOA was used, and it is of important theoretical value and practical significance.

Image segmentation is one of key technologies in image processing, and it is also a fundamental and important part of image analysis and computer vision tasks. Since the 1970s, image segmentation has been highly valued by people. Up to now, thousands of segmentation algorithms have been proposed. However, because there is no universal segmentation theory, most of the proposed segmentation algorithms are specific to the certain problems. In addition, there is no standard to choose the suitable segmentation algorithm, which brings a lot of practical problems to the applications of image segmentation. With the rapid development of deep learning, many semantic segmentation methods based on deep learning have been proposed in recent years. Compared with the traditional methods, the image segmentation algorithms based on deep learning have the advantages of precision and efficiency. For example, it was proposed that full convolutional network (FCN) [2] was adapted from VGG16 in 2015. VGG16 [3] is a classification network. Its first half is convolutional layers to extract local features, and its latter half is full connection layers to integrate global information. FCN changes the full connection layers of the latter half into the convolutional layers, which can support the segmentation of input images at any size. Compared with the traditional algorithms in the early stage, its effect is significantly improved. Among the deep learning algorithms, Mask R-CNN, an instance segmentation model, combines target detection with instance segmentation. It can not only determine the position of each target in the pictures, but also complete target segmentation and classification accurately. In recent years, it has been widely concerned and applied. In the past, the methods of instance segmentation were basically to first segment, then classify. At the same time, there were also some multi-stage concatenation methods: the candidate box of the target was predicted, then the target was segmented in the candidate box, and finally target classification was carried out on the basis of the segmentation. These methods were not only slow but also low in accuracy. However, Mask R-CNN completes image segmentation and classification at the same time, which is much simpler and more flexible. Of course, in the same period, there were also other methods which can complete image classification, detection box regression and mask prediction simultaneously, such as FCIS. It is a end-to-end full convolution solution. It can carry out instance mask prediction and classification simultaneously. Although the network has a high degree of integration, its effect is not good. It could produce errors for overlapping instances, and there are some problems in mask boundaries. In addition, some instance segmentation methods are that first semantic segmentation and then instance segmentation, while Mask R-CNN carries out instance segmentation directly.

In recent years, fine-grained image classification which is also called as sub-category recognition is a hot research problem in the field of computer vision, pattern recognition and so on. Its purpose is to carry out more detailed sub-category classification for coarse-grained categories. However, due to the subtle inter-class differences and large intra-class differences between the subcategories, compared with coarse-grained image classification, fine-grained image classification pays more attention to the small but very important local features in the image and is more difficult. Early fine-grained image classification algorithms based on artificial features generally extracted SIFT (scale-invariant feature transform), HOG (histogram of oriented gradient) or other local features from the image and then used VLAD (vector of locally aggregated descriptors), Fisher vector or other coding models for feature coding. Due to the complicated selection process of artificial features and limited expression ability, their classification effects were not good. However, with the rise of deep learning, features obtained from Convolutional Neural Networks (CNN) [4] automatically have more powerful description ability than artificial features. Therefore, a large number of convolutional feature algorithms are proposed, which promotes the rapid development of fine-grained image classification algorithms. In recent years, Convolutional Neural Network (CNN) has made remarkable achievements in general image classification, bringing a new development direction for fine-grained image classification. Researchers begin to select CNN features as image representations for fine-grained image classification. For example, literature [5] used 26 morphological features of the fore and hind wings and color features of the front to identify 43 species of butterfly specimens, which obtained a high recognition rate. Literature [6] established a neural network model based on the radial basis by using the color features of the butterfly fronts and backs, which also obtained a high recognition rate. Literature [7] proposed an image feature extraction method based on Gabor filter and an extreme learning machine (ELM) method to identify 5 butterfly species with a high recognition rate. Soon after, literature [8] used the method which combined the gray-level co-occurrence matrix feature and RGB color features of the butterfly wings' surface with artificial neural network classifier to recognize 14 butterfly species in the Satyridae family in Turkey. In 2014, the above method was improved, and the improved method which combined gray-level co-occurrence matrix (GLCM) with multi-nominal logistic regression realized the automatic recognition of 19 butterfly species. Literature [9] proposed a butterfly recognition method that the training set was expanded by observing butterfly images from multiple perspectives. Literature [10] designed 15 unique features of fishes, butterflies and plants from three aspects including geometric structure, morphology and texture features in their images, and then used neural network for training and species classification. Literature [11] used butterfly recognition model CaffeNet to identify the butterfly pattern photos. Its recognition result was not significantly different from those of the traditional SVM methods, but the recognition rate of butterfly ecological photos was much higher than those of the traditional SVM methods. Literature [12] used the Faster R-CNN algorithm to identify and classify butterfly images captured in the natural environment, which acquired good result. The accuracy rates of the above algorithms were all over 80%. These results prove that convolutional features can play a better role in fine-grained image classification, but there is still room for improvement in classification accuracy. In addition, the existing algorithms are basically based on the images of butterfly specimens, which is inclined to the simple classification task, while the ability to expand in the ecological images is weak, which needs to be further studied [13].

Therefore, a deep learning model suitable for butterfly image classification in natural environment, which is called rapid fine-grained classification of butterflies based on the FCM-KM and Mask R-CNN fusion, was proposed in this paper. The overall flowchart of the method is shown in Fig. 1. First, the collected butterfly images were preprocessed. Image preprocessing included data augmentation and data normalization. Data augmentation operations included vertical flip, horizontal flip, horizontal-vertical flip, contrast enhancement, noise addition and image enhancement based on fuzzy sets optimized by FOA. The data normalization operation was to normalize the RGB color channels of the images to $(-1, 1)$. Then, K-Means clustering algorithm optimized by dynamic population firefly algorithm based on chaos theory and max-min distance algorithm, FCM-KM, was used to determine the optimal clustering number K instead of manual parameter tuning. Finally, while effectively segmenting the butterfly images, the Softmax in Mask R-CNN was used to classify the butterfly images. The experimental results showed that the model achieved the good performance with the recognition accuracy of 83.62%. At the same time, the rapid fine-grained classification method of butterflies based on the FCM-KM and Mask R-CNN fusion was compared with CNN, Resnet and original Mask R-CNN, and the results showed that in the case of complex background, the model proposed in this paper had a better classification effect and more remarkable advantage than other networks.

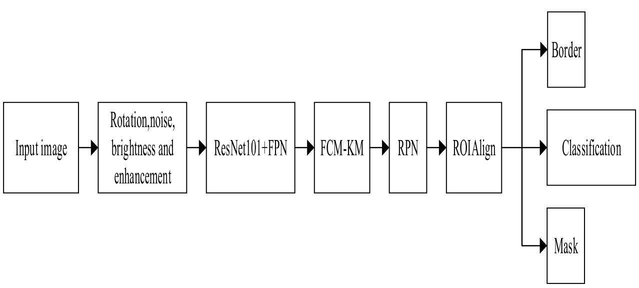


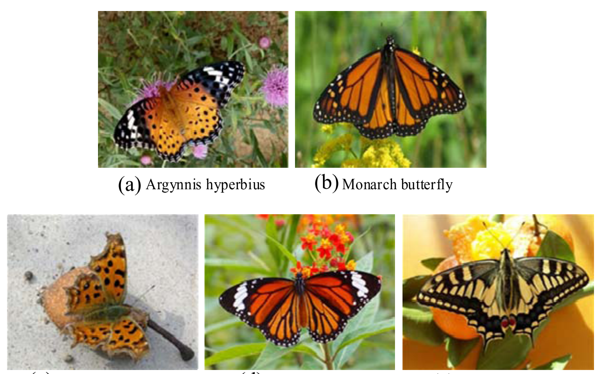
FIGURE 1. Overall flowchart.

The rest of the paper is as follows. The second part introduces the material and method of this paper. The third part analyzes the experimental results. We conclude in part four.

II. MATERIALS AND METHODS

A. DATA COLLECTION

The data set was collected and sorted out on the Internet by the author. The data set is almost a collection of the actual butterfly images captured in the ecological environment, and only a small fraction of the images are the traditional butterfly specimen images. Because of downloading on the Internet, some images were classified incorrectly. The author referred to professional books and reclassified these five kinds of butterfly images, with a total of 1,188. In the data set, the images of Argynnis hyperbius are 226, the images of monarch butterfly are 169, the images of Polygonia c-aureum are 308, the images of Danaus genutia are 235, and the images of Papilio machaon are 250. The background areas of the images are basically complex background, as shown in Fig. 2. Fig. 2 (a) is Argynnis hyperbius, Fig. 2 (b) is monarch butterfly, Fig. 2 (c) is Polygonia c-aureum, Fig. 2 (d) is Danaus genutia, and Fig. 2 (e) is Papilio machaon. In this data set, different species of butterflies have high similarity, and butterflies of the same species have great differences due to environmental influence or abnormal development.



(a) Argynnis hyperbius (b) Monarch butterfly
(c) Polygonia caureum (d) Danaus genutia (e) Papilio machaon

FIGURE 2. Examples of butterfly dataset.

B. BUTTERFLY IMAGE ENHANCEMENT BASED ON FUZZY SETS OPTIMIZED BY FOA

In the process of collection and transmission, butterfly images are susceptible to noise and blurred. There are many disadvantages in traditional fuzzy enhancement algorithms. For example, they need a large amount of calculation, they lack applicability and their parameters need to be set manually by experience. Their results of image enhancement are usually bad and cannot achieve the best effect. Therefore, fruit fly optimization algorithm was introduced into the fuzzy enhancement of butterfly images, and fuzzy entropy was used as an evaluation index for the effect of image enhancement. Use the fruit fly optimization algorithm for automatic optimization to realize the adaptive selection of parameters in butterfly image fuzzy enhancement.

1) BUTTERFLY IMAGE FUZZY PROCESSING

The process of fuzzy enhancement algorithm is as follows:

Step 1: According to the In Equation (2), in view of the different butterfly images and enhancement purpose, the parameters of the membership function (F_e, F_d, g_{max}) are adjusted, all the sets constituted by μ_{mn} are fuzzy feature planes, g_{mn} represents the maximum pixel value, F_e is the exponential fuzzy factor, F_d is the reciprocal fuzzy factor, and the fuzziness can be controlled by adjusting these parameters. Therefore, the selection of fuzzy parameters F_e and F_d can effectively enhance the butterfly images. When $\mu_{mn} = G(g_c) = 0.5$, this point is called the transition point. The selection of transition point g_c and fuzzy parameters satisfies the conditions of Equation (1):

$$G_{mn} = \begin{cases} < 0.5 & g_{mn} < g_c \\ = 0.5 & g_{mn} = g_c \\ > 0.5 & g_{mn} > g_c \end{cases} \tag{1}$$

After determining the transition point g_c, when F_e is known, formula (2) can be used to calculate F_d.

Step 2: The butterfly image is mapped from the spatial domain to the fuzzy domain by G transformation.

$$\mu_{mn} = G(g_{mn}) = [1 + \frac{g_{max} - g_{min}}{F_d}]^{F_e} \tag{2}$$

Step 3: According to the transformation of Equation (3), fuzzy enhancement can be modified. The membership degree of the modified fuzzy enhancement is ($\mu_{mn} \to \mu'_{mn}$):

$$T(\mu_m) = \begin{cases} 2 \cdot [\mu_{mn}]^2, & 0 \le \mu_{mn} \le 0.5 \\ 1 - 2 \cdot [1 - \mu_{mn}]^2, & 0.5 \le \mu_{mn} \le 1 \end{cases} \tag{3}$$

The key point of fuzzy enhancement is to enhance the membership value μ_{mn} that is greater than 0.5, and reduce the membership value that is less than 0.5, so as to reduce the fuzziness of G.

Step 4: The new gray level g'_{mn} is obtained through the G^{-1} inverse transformation, and the butterfly image is mapped from the fuzzy domain to the spatial domain:

$$\begin{aligned} g'_{mn} &= G^{-1}(\mu_{mn}) \\ &= g_{mn} - F_d[(\mu_{mn})^{\frac{-1}{F_e}} - 1] \end{aligned} \tag{4}$$

2) BUTTERFLY IMAGE ENHANCEMENT BASED ON FOA OPTIMIZED FUZZY SET

a: FRUIT FLY OPTIMIZATION ALGORITHM

Fruit Fly Optimization Algorithm (FOA) has the advantages of few control parameters and fast convergence speed. This algorithm is a group intelligence algorithm which simulates the foraging of fruit flies. The algorithm flow is as follows:

Step 1: Initialize the parameters of the algorithm, fruit fly population size and maximum number of iterations. The initial positions of fruit flies are X_begin and Y_begin.

Step 2: According to Equation (5) and (6), calculate the searching direction and distance of fruit fly individuals.

$$x_i = X\_begin + Value \times rand() \tag{5}$$

$$y_i = Y\_begin + Value \times rand() \tag{6}$$

In Equation (5) and (6), x_i and y_i represent the positions of fruit fly individuals, and Value represents the searching distance of fruit fly.

Step 3: Calculate the distance between the fruit fly individual and the origin, and calculate the taste concentration s_i of the fruit fly individual by Equation (7) and (8).

$$d_i = \sqrt{x_i^2 + y_i^2} \tag{7}$$

$$s_i = \frac{1}{d_i} \tag{8}$$

Step 4: Calculate the determination function of taste concentration by Equation (9) to obtain the taste concentration of the fruit fly individual at the current location.

$$Smell_t = Funcyion(s_i) \tag{9}$$

Step 5: Search for the best taste concentration value $Smell_b$ in the fruit fly population and the best location x_b and y_b;

Step 6: Record and reserve the best location and the best taste concentration of the fruit fly population $Snekk_b$, $X_{begin} = x_b$, $Y_{begin} = y_b$, and search in the best location direction of the fruit fly population.

Step 7: Enter the iteration process of optimization and repeat step 2-step 5. If the taste concentration is better than that of the previous iteration, carry out Step 6. Otherwise, return step 2-step 5.

b: IMAGE ENHANCEMENT REGION DEFINITION

During initialization, a certain number of populations are randomly generated, the Fitness(m) of each population is calculated, the maximum value of Fitness(m) in the population is searched, and then the speed and position of particles are updated according to FOA algorithm rules [14]. Until the given algebra is calculated, the fuzzy enhancement parameters F_e, F_d corresponding to the maximum fitness value are used for fuzzy enhancement of butterfly images.

The algorithm steps are as follows:

Step 1: Initialize the position of fruit fly population and the parameters of the algorithm;

Step 2: Calculate the Fitness(m) of each population. If it is better than the optimal value of the individual in history or the optimal value of the population in history, the position of the current value will be retained and the optimal value of the individual in history or the optimal value of the population

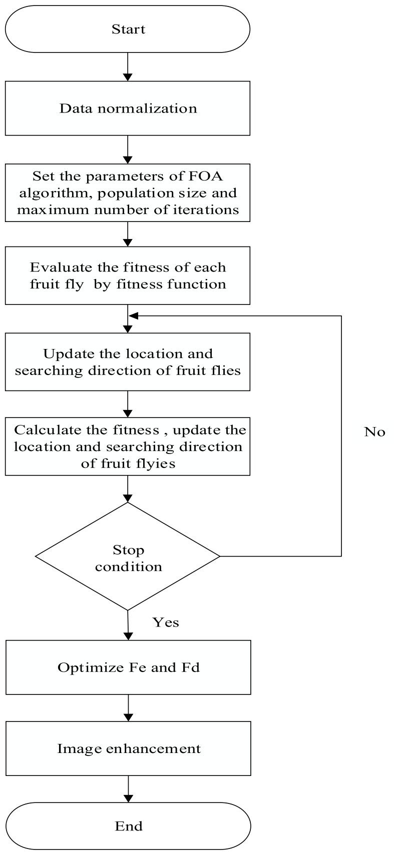


FIGURE 3. Flow chart of image enhancement based on FOA optimization fuzzy set.

in history will be updated at the same time. Otherwise, the previous historical optimal value will be retained.

Step 3: Move the particles to the new location according to the update rules of fruit fly optimized location;

Step 4: Judge whether the maximum algebra is reached. If $Lteration < Maxgen$, the optimization will end. Instead, return to step 2;

Step 5: The fuzzy enhancement parameters F_e, F_d corresponding to the maximum Fitness value which is obtained finally are used for fuzzy enhancement of butterfly images.

C. APPLE DEFECT DETECTION BASED ON MULTIVARIATE IMAGE ANALYSIS

1) MASK R-CNN

Mask R-CNN network, as an extension of Faster R-CNN, extends classification [15]–[18] and regression tasks on the basis of Faster R-CNN, which is the highest level of instance segmentation algorithm [19]–[23]. Mask R-CNN network has two-layer branching structure: the first layer is the structure of original Faster R-CNN, which is used for classification of candidate windows and regression of window coordinates; The second layer uses the full convolutional network structure to predict the binary segmentation mask for each region of interest (ROI) [24]–[26].

The steps of Mask R-CNN algorithm are as follows:

Step 1: Input butterfly images and carry out the corresponding preprocessing operation, or input the preprocessed butterfly images; Then, it is input into a pre-trained neural network to obtain the corresponding feature maps.

Step 2: Set predetermined ROIs for each pixel in the feature map to obtain multiple candidate ROIs.

Step 3: Send these candidate ROIs into the RPN network for binary classification (foreground or background) and BB regression, and filter out some of the candidate ROIs.

Step 4: Carry out ROIAlign operation on the remaining ROIs (first match the pixels of the feature maps with those of the original image, and then match the feature maps with the fixed features).

Step 5: Classify these ROIs into specific butterfly categories (N category classification), obtain the precise position by BB regression and MASK generation (fully convolutional network operation is carried out in each ROI).

For the traditional Mask R-CNN, it can not only correctly find each target in the images, remove part of the complex background and accurately segment these targets, but also effectively complete the target classification [27]–[31]. However, there are still some deficiencies in the traditional Mask R-CNN. For example, its parameters need to be set manually by experience, its efficiency is low and it lacks applicability. To deal with the problems, K-Means clustering algorithm optimized by dynamic population firefly algorithm based on chaos theory and max-min distance algorithm, FCM-KM, was used to determine the optimal cluster number K instead of manual parameter tuning. The specific process is as follows.

2) BUTTERFLY IMAGE NETWORK BASED ON THE FCM-KM AND MASK R-CNN FUSION

(1) FCM-KM

a. K-Means clustering algorithm

K-Means algorithm is a kind of unsupervised learning and a clustering algorithm based on partition, mainly used to classify samples into K categories by the operations. Generally, Euclidean distance is used as an index to measure the similarity between data objects. The similarity is inversely proportional to the distance between data objects. The greater the similarity is, the smaller the distance is. The algorithm need to specify the initial cluster number k and k initial clustering centers in advance, update the locations of the clustering centers constantly according to the similarity between data objects and clustering centers, reduce the Sum of Squared Error(SSE) of class clusters constantly, when the SSE doesn't change anymore or objective function converges, clustering is over and the final result is obtained.

The core idea of K-Means algorithm is as follows: first, randomly select K initial clustering centers $C_i(1 \leq i \leq K)$ from the data set, calculate the Euclidean distance between the remaining data objects and the clustering centers C_i, find the clustering center C_i closest to the target data objects, and assign the data objects to the cluster corresponding to the clustering center C_i. Then calculate the average value of data objects in each cluster as the new clustering centers, and carry out the next iteration until the clustering centers no longer change or the maximum number of iterations is reached.

The calculating formula of Euclidean distance between data objects and clustering center in space is

$$d(x, C_i) = \sqrt{\sum\nolimits_{j-1}^{m} (x_j - C_{ij})^2} \tag{10}$$

In Equation (10), x is the data object; C_i is the i-th clustering center; m is the dimension of the data object; x_j, C_{ij} is the j-th attribute value of x and C_i.

The calculating formula of Sum of Squared Error (SSE) of the whole data set is

$$SSE = \sum\nolimits_{i=1}^{k} \sum\nolimits_{x \in C_i} |d(x, C_i)|^2 \tag{11}$$

In Equation (11), the size of SSE represents the clustering result; K is the number of clusters.

From the above process, we can see that K-Means algorithm is very simple and easy to understand. It has fast convergence speed, low time complexity, and can effectively process large-scale data sets. Also, it is not sensitive to the input order. However, K-Means algorithm has the following defects:

a) No algorithm has been specified for the selection of the clustering center value K. If the selected K value is not reasonable, the clustering precision and computational complexity will be seriously affected.

b) The optimal clustering result corresponds to the extreme point of the objective function, and the clustering center falls near a local minimum point, which easily leads to the algorithm falling into local optimization.

For the traditional K-Means algorithm, given the initial clustering centers, the randomicity and global search ability of the firefly algorithm can be used to accurately divide the data set except the clustering centers, so as to further improve the convergence speed of the algorithm and realize the optimization of K-Means algorithm. At the same time, in view of the defect of the optimized algorithm that it is easy to fall into the local extremum region in the global optimization search, K-Means clustering algorithm optimized by dynamic population firefly algorithm based on chaos theory and max-min distance algorithm, FCM-KM, is used to optimize K-means clustering again according to initial value sensitivity and ergodicity of chaos mapping.

b. Optimization of K-Means clustering algorithm

a) Firefly algorithm

Firefly algorithm is a random search technique based on swarm intelligence. The idea stems from the biological characteristic of fireflies that they move toward brighter fireflies than themselves. In the search space, the positions of the fireflies represent the solutions of the optimization problem, and the brightness corresponds to the adaptive value of the optimization problem. The fireflies keep moving towards the brighter fireflies until the preset termination condition of the algorithm is reached and the optimization task is completed.

Set the number of fireflies to N, and set the dimension to D. The positions of the i-th firefly and the j-th firefly are expressed as $x_i = (x_{i1}, x_{i2,\ldots,} x_{iD})$, $i = 1, 2, \ldots, N$ and $x_j = (x_{j1}, x_{j2,\ldots,} x_{jD})$, $j = 1, 2, \ldots, N$. The calculating formula of distance r_{ij} between the firefly i and the firefly j is as follows:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum\nolimits_{d=1}^{D} (x_{id} - x_{jd})^2} \tag{12}$$

In Equation (12), x_{id} and x_{jd} represent the d-dimensional position of the ith firefly and the j-th firefly respectively.

The calculation formulas for the brightness and attraction of fireflies are as follows:

$$I = I_0 \exp(-\gamma * r_{ij}^2) \tag{13}$$

$$\beta = \beta_0 \exp(-\gamma * r_{ij}^2) \tag{14}$$

In Equation (13) and (14), I_0 and β_0 are the initial brightness and initial attraction of fireflies respectively, and γ is the light absorption coefficient.

The updated formula for the movement of firefly i to firefly j is as follows:

$$x_{id}(t+1) = x_{id}(t) + \beta \left(x_{jd}(t) - x_{id}(t) \right) + \alpha_i(t)\varepsilon \tag{15}$$

In Equation (15), $x_{id}(t)$ and $x_{jd}(t)$ are the d-dimensional positions of t-th generation of the firefly i and the firefly j respectively, and $\alpha_i(t)$ represents step length factor of the t-th generation of the firefly i. ε is uniformly distributed and its value range is$[-0.5, 0.5]$.

b) The Max-Min distance algorithm determines the clustering center value K

The Max-Min distance clustering algorithm is similar to the traditional K-means in that according to the nearest neighbor principle, they both divide the sample points belonging to each clustering center by calculating Euclidean distance. The difference lies in that the former does not directly give the clustering category value K, but selects an object X_i from the sample points as the first clustering center. The Euclidean distances between each point and X_i are calculated, and the point farthest from X_i is taken as the new clustering center. Repeat the above steps until no new clustering centers are generated and the total number of clustering centers K is finally determined. The steps of the algorithm are as follows:

Step 1: θ is given and $0 < \theta < 1$. Select the initial clustering center $Z_1 = x_1$;

Step 2: Generate new clustering centers

Calculate the Euclidean distances between each point and Z_1. $D_{k1} = \max\{D_{i1}\}$. The corresponding x_k is next clustering center Z_2;

Calculate the distances from each point to the clustering center Z_1 and Z_2. If $D_i = \max\{\min(D_{i1}, D_{i2})\}$, $i = 1, 2, L, n$ and $D_i > \theta * D_{12}$, then take x_1 as the third clustering center Z_3.

$$\begin{cases} D_{i1} = \|x_i - Z_1\| = \sqrt{\sum\nolimits_{i=1}^{d} |x_i - Z_1|^2} \\ D_{i2} = \|x_i - Z_2\| \end{cases} \tag{16}$$

If Z_3 exists, determine whether $D_j = max\{\min(D_{i1}, D_{i2}, D_{i3})\}$ exists. If the above conditions are met and $D_i > \theta * D_{12}$, the fourth clustering center is determined. Similarly, if $D_i \leq \theta * D_{12}$, stop searching for the new cluster center.

Step 3: Count the total number of clustering centers K

The clustering result of the algorithm has a lot to do with the selection of parameters and starting point. In order to obtain a good clustering effect, repeated experiments are required without the knowledge of prior sample distribution, so the algorithm is only used to determine the size of the clustering center value K in this paper.

c) Chaos theory optimizes clustering center

Chaotic sequences are characterized by randomness, ergodicity and boundedness, but the ergodic uniformity of chaotic sequences generated by different mappings is different and has different effects on the optimization speed of the algorithm. At present, Logistic mapping is often used to generate chaotic sequences in the literatures, but the uniformity of chaotic sequences generated by Logistic mapping is poor, and their values are usually taken in [0, 0.1] and [0.9, 1]. Through rigorous reasoning, it is verified that the chaotic sequences generated by Tent mapping are more conducive to algorithm optimization, and it is pointed out that the converging rate and ergodic uniformity of Tent mapping are better than logistic mapping.

The mathematical expression of Tent mapping is:

$$x_{t+1} = \begin{cases} 2x_t, 0 \leq \quad x_t \leq \dfrac{1}{2} \\ 2(1 - x_t), \quad \dfrac{1}{2} < x_t \leq 1 \end{cases} \tag{17}$$

The expression after Bernoulli displacement transformation is

$$x_{t+1} = (2x_t)\, mod\, 1 \tag{18}$$

The generation steps of Tent chaotic sequence are as follows:

Step 1: Randomly generate an initial value that is not in (0.2,0.4,0.6,0.8), and write it as $z, z\,(1) = x_0, i = j = 1$.

Step 2: Iterate according to Equation (18) and generate the x sequence.

Step 3: If $x(i) = [0, 0.25, 0.5, 0.75]$ or $x(i) = x(i - k)$, $k = \{0, 1, 2, 3, 4\}$, go to step2.

Step 4: Change the initial value of iteration according to the formula $x\,(i) = z(j+1)$, j=j+1, and go to step 2.

Step 5: If the maximum number of iterations is reached, terminate the operation and save the generated x sequence.

FCM-KM algorithm takes the local optimal solution searched at present as the fiducial point and generates Tent chaotic sequence, and jumps out of the local optimal through Tent search to obtain the global optimal solution.

Suppose the clustering center is $C_x, X_k = \{x_{k1}, x_{k2}, \cdots, x_{kd}\}$

The main steps of Tent chaotic search are as follows:

Step 1: Use the formula $Z_{kj}^0 = \frac{(x_{kj} - X_{min}^j)}{(X_{max}^j - X_{min}^j)}$, and map X_k to (0,1). In the formula, k = 1, 2, . . . , n, d = 1, 2, . . . , D.

Step 2: Substitute the above formula into Equation (18) of Tent mapping and iterate to generate chaotic variable sequence $Z_{kj}^m (m = 1, 2, \ldots, C_{max})$. C_{max} is the maximum number of iterations of chaotic search.

Step 3: Carry Z_{kj}^m within the space neighborhood of the original solution to generate the new solution V_k by the formula $V_{kj} = x_{kj} + \frac{(X_{max}^j - X_{min}^j)}{2} \times (2Z_{kj}^m - 1)$

Step 4: Calculate fluorescence brightness value $F(V_k)$ of V_k, compare it with fluorescence brightness value $F(x_k)$ of local optimal solution, and retain the best solution.

Step 5: If the number of searching C_{max} is reached, stop searching; Otherwise, go to step 2.

d) FCM-KM algorithm

Basic steps of FCM-KM algorithm are as follows:

Step 1: Initialize the parameters: the total number of clustering objects N, absorption coefficient γ, step length factor α, the maximum iteration times of chaotic search C_{max}, the maximum fluorescence brightness I and the maximum attraction β_0

Step 2: Determine the number of clustering centers K through the Max-Min distance algorithm, and record the initial location of clustering centers obtained by the Max-Min distance algorithm.

Step 3: Construct the chaotic search space which takes each clustering center as the fiducial points through the Tent mapping successively.

Step 4: Use Tent chaos to search and update the location of the initial clustering centers until the clustering centers no longer change.

Step 5: Match clustering centers with the target fireflies, and give them the maximum fluorescence brightness. Calculate the Euclidean distance between the remaining sample points and each cluster center, and give them different fluorescence brightness according to formula (13).

Step 6: If $I_i > I_j$, the objective function value of firefly j is less than that of firefly i, and the position of firefly j is better than firefly i. The firefly j will attract firefly i to move towards it. The movement mode is determined by formula (14), and the positions of the fireflies are updated through formula (15).

Step 7: Repeat step 6 until all fireflies are divided into their respective clustering centers.

Step 8: Output the result.

(2) The rapid fine-grained classification method of butterflies based on the FCM-KM and Mask R-CNN fusion

The structure chart of the butterfly classification method based on the FCM-KM and Mask R-CNN fusion is shown in Fig. 4

The steps of this algorithm are as follows:

Step 1: Input butterfly images and carry out the corresponding preprocessing operation, or the preprocessed butterfly images; Then, it is input into the Resnet101 neural network to extract the features and obtain the corresponding feature maps by using the pyramid (FPN) network.

Step 2: Take each pixel in the feature maps as the center, and generate multiple anchors according to the proportion of the generated anchors in the configuration

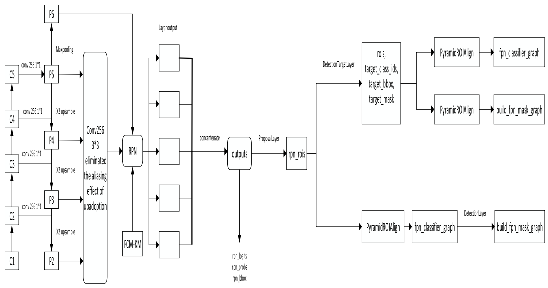


FIGURE 4. Structure chart of the classification method based on the FCM-KM and Mask R-CNN fusion.

file(RPN_ANCHOR_SCALES = (32, 64, 128, 256, 512), RPN_ANCHOR_RATIOS = [0.5,1,2]).

Step 3: Use K-Means clustering algorithm optimized by dynamic population firefly algorithm based on chaos theory and max-min distance algorithm, FCM-KM, to roughly screen the anchors.

Step 4: Send the screened anchors into the RPN network for binary classification (foreground or background) and BB regression.

Step 5: Apply the outputs of RPN network to the anchors. First, sort the probabilities and keep the part with high probability of the foreground. Then, select the corresponding anchors and use the regression value of RPN to correct the anchors for the first time. After fixing these, the anchors are carefully screened using a non-maximal inhibition method.

Step 6: Carry out ROIAlign operation on the remaining ROIs (first match the pixels of the feature map with those of the original image, and then match the feature map with the fixed feature).

Step 7: Classify these ROIs into specific categories of butterflies (Softmax is used for N category classification), obtain the exact location by BB regression and MASK generation (fully convolutional network operation is carried out in each ROI).

III. EXPERIMENTAL RESULTS AND ANALYSIS

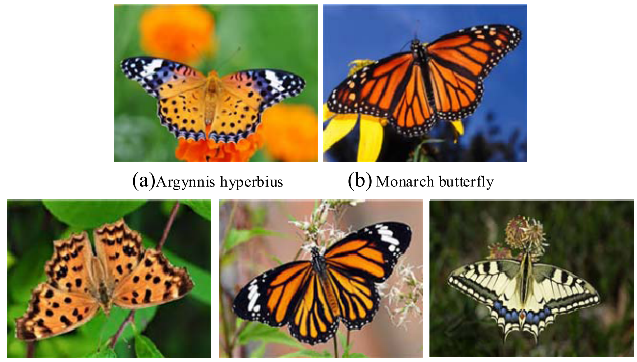
A. EXPERIMENTAL ENVIRONMENT

The hardware information is as follows: Processor is Intel Core i9, 500 GB RAM; Graphics is Nvidia Titan Xp; System Memory is 16 G.

The software information is as follows: LabelMe; PyCharm 2019.1.3; MATLAB 9.0 (R2016a), Keras.

B. DATA PREPROCESSING

The experimental database contains 5 categories of butterflies, and each category contains 169 to 308 images, with a total of 1,188 images. Samples of five butterfly species are shown in Fig. 5. Fig. 5 (a) is Argynnis hyperbius, Fig. 5 (b) is monarch butterfly, Fig. 5 (c) is Polygonia caureum, Fig. 5 (d) is Danaus genutia, and Fig. 5 (e) is Papilio machaon. It can be found that the image backgrounds of the database are complex; The colors and illumination of the targets within



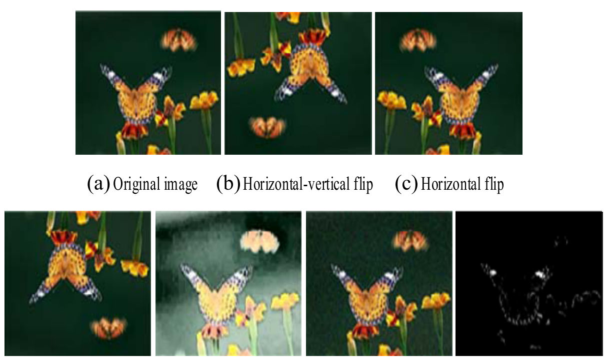
(a)Argynnis hyperbius (b) Monarch butterfly

(c)Polygonia caureum (d)Danaus genutia (e)Papilio machaon

FIGURE 5. Examples of butterfly images in the database.

the same category vary greatly; The shapes and colors of the targets across the categories are very similar.

A total of 913 images were randomly and uniformly selected from these five categories of butterfly images as the training set, and the remaining 275 images were as the validation set. Because the trainings of the deep neural network models require a lot of images to extract the effective features and the models need to alleviate overfitting, data augmentation on the training set was done. The data augmentation operations included vertical flip, horizontal flip, horizontal-vertical flip, contrast enhancement, noise addition and image enhancement based on fuzzy sets optimized by FOA. The resulting image is shown in Fig. 6. Fig. 6 (a) is the original image, Fig.6 (b) is horizontal-vertical flip, Fig. 6 (c) is horizontal flip, Fig. 6 (d) is vertical flip, Fig. 6 (e) is contrast enhancement, Fig. 6 (f) is noise addition, and Fig. 6 (g) is image enhancement based on fuzzy sets optimized by FOA. After data augmentation, the images of Argynnis hyperbius is 277, the images of monarch butterfly is 207, the images of Polygonia c-aureum is 377, the images of Danaus genutia is 288, and the images of Papilio machaon Linnaeus is 313, with a total of 1,462. Because the number of images was fewer, only the training set and verification set were set in the experiment. Table 1 shows the settings of



(a) Original image (b) Horizontal-vertical flip (c) Horizontal flip

(d) Vertical flip (e) Contrast enhancement (f) Noise addition (g) Image enhancement

FIGURE 6. Seven data augmentation methods of butterfly images.

TABLE 1. The settings of training set and verification set in the data set.

settings	Number
training set	913+274(expansion)
validation set	275
total	1462

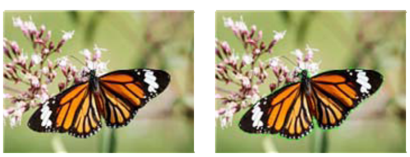
TABLE 2. The number and proportion of five categories of butterfly images after data augmentation.

Butterfly species	Number	Proportion
Argynnis hyperbius	277	18.95%
monarch butterfly	207	14.16%
Polygonia caureum	377	25.79%
Danaus genutia	288	19.70%
Papilio machaon	313	21.41%

the training set and verification set in the data set used in this experiment. Table 2 shows the number and proportion of the five categories of butterfly images after data enhancement.

C. DATA ANNOTATION

Mask R-CNN requires data training before image target detection and instance segmentation. However, data training requires the labeled data set, so it is necessary to annotate the data set. The image annotation tool selected in this paper is LabelMe. Just install LabelMe under Anaconda and stimulate the LabelMe environment in CMD to use. The LabelMe tool saves the annotated annotations in a JSON file, generating a set of polygon points for each object in the image, and storing their coordinate values such as (x, y). When training,the network needs a 'mask image', which will be temporarily generated by loading polygon coordinate points from the JSON file. An image is selected from the Mask R-CNN training set. Fig. 7 (a) is the original image, and fig. 7 (b) is the labeled image.



(a) Original image (b) The labeled image

FIGURE 7. The labeled image.

D. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the effectiveness of the training model for butterfly recognition, the experiment first compared the experimental effect of the method proposed in this paper with that of the original Mask R-CNN. Then, deep learning methods CNN and Resnet which were widely used in butterfly recognition were selected to do the comparative experiments, and finally the experimental conclusion was drawn.

1) EVALUATION INDEXES

The evaluation indexes in this paper are composed of four indexes: accuracy, recall rate, accuracy and loss function. Since the identification results of butterflies need to be evaluated, the evaluation indexes need to consider both accuracy and recall rate. F1 is selected as one of the evaluation indexes of butterfly identification results. F1 is the measurement function of accuracy and recall rate, which is defined as follows, as shown in Equation (19).

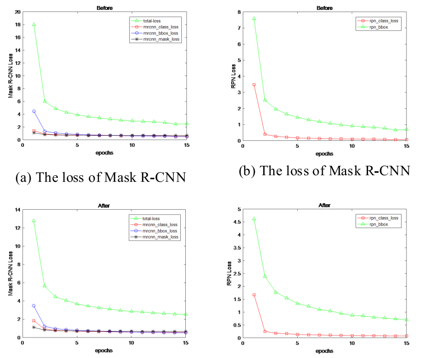
$$\begin{aligned} F1 &= \frac{2PR}{P+R} \\ P &= \frac{TP}{TP+FP} \\ R &= \frac{TP}{TP+FN} \end{aligned} \tag{19}$$

In Equation (19), P represents precision; R represents recall rate.

TP represents the number of the samples which are actually the butterflies and are predicted by the model to be the butterflies (the positive samples are detected as the positive samples). FP represents the number of the samples which aren't actually the butterfly but are predicted by the model to be the butterflies (negative samples are detected as positive samples). FN represents the number of the samples which are actually the butterflies but aren't predicted by the model to be the butterflies (the positive samples are not detected as positive samples).

2) EXPERIMENTAL COMPARISON AND PERFORMANCE EVALUATION

The original Mask R-CNN network and the improved method proposed in this paper, rapid fine-grained classification of butterflies based on the FCM-KM and Mask R-CNN fusion, were used to identify butterflies respectively, and then their effects of butterfly recognition were compared. The loss of Mask R-CNN is composed of the losses of category prediction branch, boundary box prediction branch and segmentation mask branch. The training loss of original Mask R-CNN model is shown in Fig. 8 (a), and the training loss of the improved model based on the FCM-KM and Mask R-CNN fusion is shown in Fig. 8 (c). The loss of the regional suggested network consists of the losses of category prediction and boundary box prediction. The original training loss of the regional suggested network is shown in Fig. 8 (b), and the improved training loss of the regional suggested network is shown in Fig. 8 (d). The horizontal axis represents the number of training batches, and the vertical axis represents the loss in these figures. It can be seen that compared with the original Mask R-CNN model, the loss values of the improved model based on the FCM-KM and Mask R-CNN fusion are lower, and its training effect is better.



(a) The loss of Mask R-CNN (b) The loss of Mask R-CNN

(c) The loss of the improved model (d) The loss of the improved model

FIGURE 8. The loss of Mask R-CNN and the improved model based on the FCM-KM and Mask R-CNN fusion.

TABLE 3. The accuracy of Mask R-CNN and the improved model based on the FCM-KM and Mask R-CNN fusion.

Training network	Without image enhancement	With image enhancement
Mask R-CNN	79.55%	80.43%
Improved model	81.94%	83.62%

The experimental results of the two algorithms are shown in Table 3. After the experiment of the data set without image enhancement, the accuracy of the improved model based on the FCM-KM and Mask R-CNN fusion was 81.94%, which was higher than that of the original Mask R-CNN (+2.39%). After the experiment of the data set with image enhancement, the accuracy of the improved model based on the FCM-KM and Mask R-CNN fusion was 83.62%, which was higher than that of the original Mask R-CNN (+3.19%). Experimental results show that the improved model based on the FCM-KM and Mask R-CNN fusion can effectively optimize the training model.

At the same time, the experimental results of the improved model based on the FCM-KM and Mask R-CNN fusion for each category were also compared with those of the original Mask R-CNN model, as shown in Table 4. It can be seen that the classification accuracy is improved for various categories (+3.78%, +1.74%, +2.11%, 2.94%, +2.92%), and the classification accuracy of the improved model based on the FCM-KM and Mask R-CNN fusion is higher than that of the original Mask R-CNN model. Among these data, the classification accuracies of Argynnis hyperbius, Danaus genutia and Papilio machaon are more than 83%, while the classification accuracies of monarch butterfly and Polygonia caureum are relatively low. This is because, as shown in Table 2, monarch butterfly accounts for less proportion in the data set, its images with complex backgrounds account for a large proportion, and the image backgrounds of the Polygonia caureum are also all complex backgrounds, the training results of these two categories of butterflies are relatively poor. In addition, some of the characteristics of the selected butterflies are very similar, which increases the difficulty of identification. At the same time, different environment background will also affect the classification results.

TABLE 4. Comparison of experimental results of the two models for each category.

Butterfly species	Mask R-CNN	Improved model
Argynnis hyperbius	86.01%	89.79%
Monarch butterfly	62.19%	63.93%
Polygonia caureum	76.34%	78.45%
Danaus genutia	83.59%	86.53%
Papilio machaon	84.29%	87.21%

In this paper, the data set without image enhancement and with image enhancement is used to train different networks. The classification accuracies of different networks are shown in Table 5. The experimental results show that the classification accuracy of the dataset with image enhancement is improved in different networks (+0.15%, 0.42%, 0.88%, 1.68%), and compared with other networks, the classification accuracy of the improved model based on the FCM-KM and Mask R-CNN fusion is significantly improved. The F1 values of different species for different networks are shown in Table 6. The results show that the improved model based on the FCM-KM and Mask R-CNN fusion is better than the other three algorithms in recognizing butterflies, and its F1 value is higher than those of CNN, Resnet and Mask R-CNN. In general, the improved Mask R-CNN has a good recognition effect on butterflies.

TABLE 5. Classification accuracy of different networks.

Training network	Without image enhancement	With image enhancement
CNN	64.01%	64.16%
Resnet	72.86%	73.28%
Mask R-CNN	79.55%	80.43%
Improved model	81.94%	83.62%

TABLE 6. F1 values for each category for different networks.

Species	CNN	Resnet	Mask R-CNN	Improved model
Argynnis hyperbius	83.12%	90.13%	94.26%	96.25%
monarch butterfly	66.45%	72.78%	79.65%	79.65%
Polygonia caureum	76.19%	82.19%	87.80%	88.62%
Danaus genutia	81.48%	88.37%	92.54%	94.50%
Papilio machaon	82.11%	88.61%	92.81%	94.62%

IV. CONCLUSIONS

In this paper, a fine-grained butterfly recognition algorithm based on the improved model based on the FCM-KM and Mask R-CNN fusion was proposed. First, an adaptive image enhancement algorithm based on fuzzy sets optimized by FOA was used to realize the adaptive fuzzy enhancement of butterfly images in image pre-processing. Then, K-Means clustering algorithm optimized by dynamic population firefly algorithm based on chaos theory and max-min distance algorithm, FCM-KM, was used to determine the optimal clustering number K instead of manual parameter tuning. Finally, while effectively segmenting the butterfly images, the Softmax in Mask R-CNN was used to classify the butterfly images. The method proposed in this paper can find each object in the image correctly, remove the complex interference information in the background and segment the objects accurately to simplify the classification work. At the same time, the feature maps of the input images were extracted for classification, and its validity in butterfly classification was verified on the data set. Experimental results showed that compared with the original Mask R-CNN network, the overall classification results improved by about 3%. In order to verify the feasibility and effectiveness of the model in complex environments, the comparative experiments between CNN, Resnet, Mask R-CNN and the improved model based on the FCM-KM and Mask R-CNN fusion were done. The experimental results showed that the improved model had better classification effect and more remarkable advantages in the case of complex background.

The butterfly recognition model proposed in this paper improved the effect of fine-grained butterfly classification in complex backgrounds greatly, but considering that the data set contained fewer butterfly categories, the model will be improved from the perspective of expanding the butterfly categories in the future to enhance the generalization ability of the recognition model and further improve the performance of the algorithm.

REFERENCES

[1] D. L. Peng and T. J. Wu, "A generalized image enhancement algorithm using fuzzy sets and its application," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Nov. 2002, pp. 820–823, doi: 10.1109/ICMLC.2002.1174496.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[3] Z. Song, L. Fu, J. Wu, Z. Liu, R. Li, and Y. Cui, "Kiwifruit detection in field images using faster R-CNN with VGG16," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 76–81, 2019, doi: 10.1016/j.ifacol.2019.12.500.

[4] P. Li and W. Zhao, "Image fire detection algorithms based on convolutional neural networks," *Case Stud. Thermal Eng.*, vol. 19, Jun. 2020, Art. no. 100625, doi: 10.1016/j.csite.2020.100625.

[5] J. W. Zhang, "Automatic identification of butterflies based on computer vision technology," in *College of Agriculture and Biotechnology*. Beijing, China: China Agriculture Univ., 2006, doi: 10.7666/d.y940039.

[6] F. Liu, *The Application of Wings' Color Characters in Butterfly Species Automatic Identification*. Beijing, China: China Agriculture Univ., 2007, doi: 10.7666/d.y1107954.

[7] Y. Kaya, L. Kayci, and R. Tekin, "A computer vision system for the automatic identification of butterfly species via Gabor-filter-based texture features and extreme learning machine: GF+ ELM," *TEM J*, vol. 2, no. 1, pp. 13–20, 2013.

[8] Y. Kaya and L. Kayci, "Application of artificial neural network for automatic detection of butterfly species using color and texture features," *Vis. Comput.*, vol. 30, no. 1, pp. 71–79, Jan. 2014, doi: 10.1007/s00371-013-0782-8.

[9] S.-H. Kang, J.-H. Cho, and S.-H. Lee, "Identification of butterfly based on their shapes when viewed from different angles using an artificial neural network," *J. Asia–Pacific Entomology*, vol. 17, no. 2, pp. 143–149, Jun. 2014, doi: 10.1016/j.aspen.2013.12.004.

[10] A. Hernández-Serna and L. F. Jiménez-Segura, "Automatic identification of species with neural networks," *PeerJ*, vol. 2, p. e563, Nov. 2014, doi: 10.7717/peerj.563.

[11] A. M. Zhou, M. PengPeng, X. TianYu, W. JiangNing, F. Jin, S. ZeZhong, T. YuLei, and Y. Qing, "Automatic identification of butterfly specimen images at the family level based on deep learning method," *Acta Entomologica Sinica*, vol. 60, no. 11, pp. 1339–1348, 2017, doi: 10.16380/j.kcxb.2017.11.012.

[12] X. Juan-Ying, H. Qi, S. Yinghuan, L. Peng, J. Liping, Z. Fuzhen, Z. Junping, T. Xiaoyang, and X. Shengquan, "The automatic identification of butterfly species," *J. Comput. Res. Develop.*, vol. 55, no. 8, pp. 1609–1618, 2018, doi: 10.7544/issn1000-1239.2018.20180181.

[13] D. J. Lee, R. B. Schoenberger, D. Shiozawa, X. Xu, and P. Zhan, "Contour matching for a fish recognition and migration-monitoring system," *Proc. SPIE*, vol. 5606, pp. 37–48, Dec. 2004, doi: 10.1117/12.571789.

[14] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012, doi: 10.1016/j.knosys.2011.07.001.

[15] Y. Chen, F. Feng, and Z. M. Yuan, "Automatic identification of butterfly species with an improved support vector classification," *Acta Entomologica Sinica*, vol. 54, no. 5, pp. 609–614, 2011, doi: 10.1017/S0022112010006427.

[16] P. Pan, Z. R. Shen, L. W. Gao, and H. Z. Yang, "Development of the technology for auto-extracting venation of insects," *Entomotaxonomia*, vol. 30, no. 1, pp. 72–80, 2008, doi: 10.1016/S1005-9040(08)60003-3.

[17] F. Li and Y. Xiong, "Automatic identification of butterfly species based on HoMSC and GLCMoIB," *Vis. Comput.*, vol. 34, no. 11, pp. 1525–1533, Nov. 2018, doi: 10.1007/s00371-017-1426-1.

[18] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2881–2890, doi: 10.1109/CVPR.2017.660.

[19] G. Lin, C. Shen, A. van den Hengel, and I. Reid, "Exploring context with deep structured models for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1352–1366, Jun. 2018, doi: 10.1109/TPAMI.2017.2708714.

[20] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2013, pp. 1139–1147, doi: 10.1007/s00287-015-0911-z.

[21] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1925–1934, doi: 10.1109/cvpr.2017.549.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–6, doi: 10.1109/CVPR.2015.7298594.

[26] S. Xiao, P. Ting, and R. Fu-Ji, "Facial expression recognition using ROI-KNN deep convolutional neural networks," *Acta Automatica Sinica*, vol. 42, no. 6, pp. 883–891, 2016, doi: 10.16383/j.aas.2016.c150638.

[27] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.

[29] P. L. Pan, H. Z. Yang, Z. R. Shen, L. W. Gao, J. W. Zhang, H. Q. Zhao, and X. W. Yu, "Research on applying vein feature for mathematical morphology in classification and identification of butterflies (lepidoptera: Rhopalocera)," *Entomotaxonomia*, vol. 30, no. 2, pp. 151–160, 2008., doi: 10.3969/j.issn.1000-7482.2008.02.013.
[30] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, "Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion," *IEEE Access*, vol. 7, pp. 143190–143206, 2019, doi: 10.1109/access.2019.2943454.
[31] W. Zhang, J. Hu, G. Zhou, and M. He, "Detection of apple defects based on the FCM-NPGA and a multivariate image analysis," *IEEE Access*, vol. 8, pp. 38833–38845, 2020, doi: 10.1109/access.2020.2974262.

AIJIAO TAN was born in 1993. She received the bachelor's degree from the Changzhou Institute of Technology, in 2017. In 2018, she joined the Central South University of Forestry and Technology. Her main research direction of postgraduate study is graphic image processing.

GUOXIONG ZHOU received the B.Sc. degree from Hunan Agricultural University, in 2002, and the M.Sc. and Ph.D. degrees from Central South University, China, in 2006 and 2010, respectively. He is currently an Associate Professor with the Central South University of Forestry and Technology. His main research interests include forest fire prevention and robotics.

MINGFANG HE received the Ph.D. degree in control science and engineering from Central South University, China, in 2015. Her main research interests include science and engineering, specifically artificial intelligence.