Received June 14, 2020, accepted July 15, 2020, date of publication August 18, 2020, date of current version August 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017560

Instance Segmentation and Classification Method for Plant Leaf Images Based on ISC-MRCNN and APS-DCCNN

XIAOBO YANG[1,2], AIBIN CHEN[1,3], GUOXIONG ZHOU[1], JIANWU WANG[4], WENJIE CHEN[1], YUAN GAO[1], AND RUNDONG JIANG[1]

[1]Institute of Artificial Intelligence Application, College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410004, China
[2]College of Information Engineering, Tongren Polytechnic College, Tongren 554309, China
[3]Hunan Provincial Key Laboratory of Urban Forest Ecology, Central South University of Forestry and Technology, Changsha 410000, China
[4]Huangfengqiao State-Owned Forest Farm, Zhuzhou 412313, China

Corresponding author: Aibin Chen (5708111@qq.com)

This work was supported by the National 948 Project of China "Forest Fire Prediction and Fire Fighting Resource Dispatching Technology" under Grant 2014-4-09.

ABSTRACT To solve the complex background problems (e.g. Noise interference, object overlap, and different illumination) that affect the classification performance on plant leaf images, this paper proposes an instance segmentation and classification method for plant leaf images based on IFPN SNMS CFFI-Mask R-CNN (ISC-MRCNN) and ACPSOSVM-Dual Channels Convolutional Neural Network (APS-DCCNN). To obtain the foreground of plant leaf images, the lateral connection structure of the feature map pyramid in ISC-MRCNN fuses the feature maps of different depths, so that the network learns more detailed features. Then, the Soft Non-Maximum Suppression Algorithm is employed to improve the detection performance of overlapping objects. Next, the pooling method of integrating the continuous function can reduce the precision loss during the alignment of the mapping between the feature map and the original image. Finally, by constructing a mask filter layer, complex backgrounds are masked. To distinguish the similarity between plant leaf images, APS-DCCNN is used to classify the foreground images. In this process, the Support Vector Machine is used to replace softmax and then an Adaptive Chaotic Particle Swarm Algorithm is employed to optimize it. The experimental results show that compared with Mask R-CNN, the average precision of ISC-MRCNN has increased by 1.89% under different thresholds. The proposed method is suitable for the object detection and instance segmentation problems with complex background. Besides, compared with traditional CNN, the average precision of the classification results obtained by APS-DCCNN has improved by 1.59%. This has shown that the proposed method is suitable for the classification of plant leaves.

INDEX TERMS Object detection, instance segmentation, plant leaf, adaptive chaotic particle swarm algorithm, Mask R-CNN, dual channels convolutional neural network, support vector machine.

I. INTRODUCTION

A. RESEARCH BACKGROUND AND SIGNIFICANCE

The recognition and classification of plant species aims to distinguish the species of plants and explore the origin of plant species. Plant species recognition is not only important for botany research [1], but also greatly significant for human development and science popularization. Currently, the recognition of plant species is mainly based on plant morphology and achieved through manual recognition. However, there are thousands of plant species in nature. In this case, manual recognition is time-consuming, labor-intensive, and inefficient, and the recognition results depend largely on the human experience. This brings difficulties to the in-depth study of plant species. With the increase of plant images, limited expert knowledge can no longer meet the needs of large-scale image processing. As a result, the automatic recognition of plant species has received increasingly more attention. With the development of imaging technology, people can easily acquire clear plant images, and the

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

computer-based recognition of plant images is currently a research hotspot [2]. As the leaves of different species have different characteristics, and the leaves are not difficult to collect, it is the most direct and effective method to recognize plant species through leaves [3].

B. RELATED WORK

1) LEAF IMAGE RECOGNITION BASED ON MACHINE LEARNING

At the meanwhile, plant leaf recognition methods especially machine learning methods had gained much attention. Moreover, the combination of extracted features with machine learning methods has shown better performance in recognition rate compared with traditional leaf recognition methods. For example, Ingrouille and Laird [4] used 27 leaf features to classify oak trees based on the Principal Components Analysis. Zheng *et al.* [5] combined the leaf geometry, Hu invariant moment feature, grayscale co-occurrence matrix feature, and fractal dimension to recognize plant leaves based on Support Vector Machine (SVM) classifiers and obtained a higher recognition rate. Fu *et al.* [6] combined Local Binary Pattern features with leaf shape features and used the nearest neighbor method to classify and recognize leaves. The performance of the above methods depends to a large extent on whether the characteristics of artificial selection are reasonable, while the feature selection often depends on experience. Besides, the extracted features are relatively single or the classifier structure is relatively simple, so that the recognition rate is low. Currently, most of the plant leaf images are collected in a laboratory environment using a high-resolution imaging device, and the collected images have a single background, easily leading to good recognition performance. By contrast, the images taken in the natural environment usually have strong interference, such as complex background, which poses a great challenge to leaf recognition. Therefore, it is necessary to perform image preprocessing operations on plant leaf images before recognition. Such preprocessing operations are to achieve the object detection and instance segmentation of plant leaves, thereby removing strong interference in images.

2) OBJECT DETECTION AND INSTANCE SEGMENTATION

Instance segmentation is a task that finds the object contours at the pixel level and is one of the most difficult visual tasks in computer vision. Instance segmentation differs from object detection in that object detection only outputs the number of objects in an image, while instance segmentation outputs not only the number of objects in the image, but also the position and edge of objects as well as classify the pixels. Instance segmentation is usually implemented based on object detection. Traditional object detection methods, such as scale-invariant feature transform [7], gradient direction histogram [8], and variable component model [9], design features based on prior knowledge. Although these traditional methods have achieved high detection speeds and precisions in certain scenarios, they greatly rely on prior knowledge, leading to poor adaptability to other scenarios. In recent years, the object detection methods based on deep learning can adaptively extract the features of different layers of an object, and apply the trained model to different scenarios, which effectively improves the detection precision and adaptability. The object detection models based on deep learning can be divided into single-stage object detection and two-stage object detection according to whether classification regression and region extraction are separated. 1) Single-stage object detection, that is, regression-based object detection models, where default boxes are pre-defined according to the feature map, and then objects can be classified. Typical regression-based methods include YOLO [10], SSD [11], and YOLOv3 [12] which can successfully extract the bounding boxes and greatly improve the detection speed, but the detection precision is poor [13]. 2) Two-stage object detection, that is, Candidate region-based object detection models, where the boxes of objects are extracted firstly, and then the output is input into the pooling layer of Region of Interests (RoI) together with the feature map to achieve classification and localization of objects. Girshick *et al.* [14] proposed the algorithm Regions with Convolution Neural Network (R-CNN), and introduced the deep learning into the object detection field for the first time, where the image features were extracted using R-CNN, to achieve adaptive object detection and locate the segmented objects. Then, Fully Convolutional Networks [15] (FCN) was proposed, where the fully connected layer was replaced with a convolutional layer. FCN fuses an image's local and global information, and trains end-to-end full convolutional networks for arbitrarily-sized input images, to achieve pixel-by-pixel classification and semantic segmentation. On this basis, many researchers have further improved the FCN method. For example, SPP-Net [16] introduced the spatial pyramid pooling layer in R-CNN, which improved the detection precision and reduced the influence of input image size on the network. Fast R-CNN [17] further adopted single-scale pooling based on the spatial pyramid pooling layer of SPP-Net, which greatly improved the detection speed. Faster R-CNN [18] introduced Region Proposal Network (RPN) in the process of extracting candidate regions by Fast R-CNN, which enabled end-to-end training and improved the precision of the region extraction.

The instance segmentation methods are divided into single-stage and two-stage instance segmentation methods based on object detection methods. In the two-stage methods, there are two lines in the research of instance segmentation for a long time, which are the bottom-up method based on semantic segmentation and the top-down method based on object detection.

The idea of the top-down instance segmentation method is to first find out the area of the instance (bounding box) through the method of object detection, and then perform semantic segmentation in the detection box. Finally, each segmentation result is output as a different instance. The typical representative is Mask R-CNN [19] and Mask Scoring R-CNN [20]. Based on Faster R-CNN, Mask R-CNN improved the RoIPooling layer to the RoIAlign pooling layer, and reduced position errors of bounding box regression by using a bilinear interpolation method. Besides, a mask branch was added to achieve high-precision segmentation at the pixel level, so that the overall average precision was improved. Mask Scoring R-CNN adds a branch based on Mask R-CNN, which is used to score the mask to predict a more accurate score, and its performance exceeds Mask R-CNN. The idea of the bottom-up instance segmentation methods is to first perform pixel-level semantic segmentation, and then distinguish different instances through clustering, Metric Learning [21], and other means. For example, the Instance Embedding method proposed by De Brabandere *et al.* [22] first obtains all object masks through semantic segmentation, then uses a discriminant loss function for pixel embedding, and finally uses a clustering method to output different instances.

Similarly, the single-stage instance segmentation methods are affected by the single-stage object detection research, and there are also two lines. One is inspired by anchor-based detection models, such as YOLO, RetinaNet [13], the representative models are YOLACT [23], and SOLO [24]. The other is inspired by the anchor-free detection model such as FCOS [25], representative models are PolarMask [26]. For now, the model with the highest accuracy for stage instance segmentation should be the BlendMask [27], which surpasses Mask R-CNN in both accuracy and speed. BlendMask is a single-stage dense instance segmentation method that combines the ideas of top-down and Bottom-up methods. It adds a bottom module to extract low-level detailed features based on the detection model FCOS, and predicts an attention on instance-level; and proposes a Blender module to better integrate these two specialties. BlenderMask's accuracy and speed on COCO surpass Mask R-CNN. Although this method has high accuracy and fast speed, the innovation is not outstanding, and the ideas of YOLACT and Mask R-CNN are still used.

YOLACT splits the instance segmentation task into two parallel subtasks: (1) Using protonet network to generate k prototype masks for each image. (2) Predicting k linear combination coefficients for each instance. Finally, the instance mask is generated through linear combination. Although the speed is improved, but the accuracy is not greatly improved. SOLO transforms the problem of instance segmentation into the classification problem of pixels' "instance labels". It surpasses Mask R-CNN in the accuracy. Compared with PolarMask with similar ideas, it also has a greater advantage. PolarMask proposes a new instance segmentation modeling method, which models the contour of an object based on polar coordinates and converts instance segmentation into instance center classification and dense distance regression. PolarMask achieved 32.9 mAP on coco test-dev under the configuration of ResNet101. From the experimental results, PolarMask's accuracy is not very high, and there is no advantage in speed, but it brings us a new idea, which has great enlightening significance for the subsequent research. CondInst [28] proposed a method of dynamic convolution, and made it get rid of the shackles of the box, which is about 0.5∼1.5 percentage points higher in accuracy than MaskR-CNN, and slightly better in speed than Mask R-CNN. SOLOv2 [29] proposed the suppression algorithm of Matrix NMS. In the case of the same backbone, the accuracy is one percentage point higher than Mask R-CNN, but from the perspective of speed and accuracy, it is similar to Mask R-CNN.

It's undeniable that Mask R-CNN has become one of the best methods in all current instance segmentation method models. The Mask R-CNN can effectively solve the challenge in the recognition and classification of plant leaf images with noise interference, object overlap, and bad illumination. After solving the problem of complex background in plant leaf images, the recognition and classification methods based on Convolution Neural Network (CNN) for plant leaf images can achieve excellent recognition results.

3) IMAGE CLASSIFICATION BASED ON CONVOLUTIONAL NEURAL NETWORK

When machine learning methods are employed to recognize plant leaves, CNN-based plant leaf recognition methods have been widely used [30], where the recognition rate, precision, and speed have been greatly improved compared with traditional machine learning methods such as SVM [31]. CNN has shown great achievements in the field of image classification, and its powerful feature learning and classification ability makes this method widely used. CNN provides an end-to-end learning model, where the parameters in the model can be trained by gradient descent algorithm and backpropagation algorithm. The trained CNN can learn the features of images autonomously and achieve the extraction and classification of objects in images. In 1998, Lecun *et al.* [32] proposed LeNet-5 where gradient-based backpropagation algorithm was employed and feature image and weight sharing were added in the network structure to supervise the network. LeNet-5 achieved great success in the field of handwritten character recognition, which made CNN gain widespread attention. However, the CNN methods are only suitable for small-scale data and simple scenarios due to the small number of layers. In 2012, Krizhevsky *et al.* [33] proposed AlexNet to win the championship in the image classification competition of the large image database named ImageNet with a precision exceeding the 11% of the second place, which made the neural network gain further attention. After that, the researchers further improved the network performance, such as VGG [34] of Oxford University, GoogLeNet [35] of Google, ResNet [36] of Microsoft, etc. They show the trend that the number of network layers is gradually increasing and more parameters are used. Therefore, the precision of image recognition has increased dramatically, surpassing the record created by AlexNet on ImageNet. Owing to the huge advantages of CNN, this paper uses CNN to recognize plant leaf images. In general, CNN uses the softmax activation function to predict and minimize cross-entropy loss.

However, SVM is a widely-used alternative [37]. Tang *et al.* [38] used CNN to extract features, used SVM classification, and achieved good performance in the popular deep learning datasets like MNIST, CIFAR-10. Chen *et al.* [39] used CNN to extract the multi-scale features of car images with complex road background, and then combined SVM to classify multi-scale features to achieve seat belt detection in road surveillance images. For SVM, the choice of kernel parameters, the setting of parameters, and the selection of features are all critical. Unsuitable parameters and features may lead to bad results. Some scholars have researched parameter estimation and feature selection of SVM. The methods widely used in SVM parameter optimization include genetic algorithm, and Particle Swarm Optimization (PSO), etc. Zhang and Mao [40] used the PSO's optimization capabilities to select the original features for training the SVM classifier. Gao *et al.* [41] proposed to employ both genetic algorithm and PSO for feature selection and parameter optimization, and the model is applied to high-resolution image classification. The above-mentioned methods have achieved good results and improved the performance of the SVM algorithm to some extent. However, there are also some problems. For genetic algorithms, a large number of swarms and iterations are required to ensure stability. For PSO, there are problems such as premature convergence and easily falling into local extremum. Based on the PSO algorithm, the Chaotic Particle Swarm Optimization (CPSO) algorithm is combined with SVM to solve the problem of SVM parameter setting and feature selection. Therefore, after image preprocessing operation, CNN is used for feature extraction, SVM is used for recognition and classification, and Adaptive Chaotic Particle Swarm Optimization (ACPSO) algorithm is used to optimize it to achieve better recognition results.

To obtain a better recognition performance on plant leaf images with complex background, our contributions are as follows: (1) a plant leaf image preprocessing method based on ISC-MRCNN is proposed to achieve object detection, instance segmentation, feature extraction, and removing interference information. (2) a plant leaf image classification method based on APS-DCCNN is proposed to solve the inter-class similarity between plant leaves. The method first constructs a structure with dual channels CNN and replaces the last layer of the fully-connected layer with SVM and the softmax classifier. Then the ACPSO algorithm is employed to optimize the SVM, to solve the problems of feature selection and parameter setting. Finally, the overall efficiency of recognizing plant leaf images is improved.

II. OBJECTIVES

In the natural environment, plant leaf images taken with high-definition equipment have background information interference, object overlap, different light intensity and illumination direction, and different shooting angles. Some of the plant leaf images are shown in Figure 1.

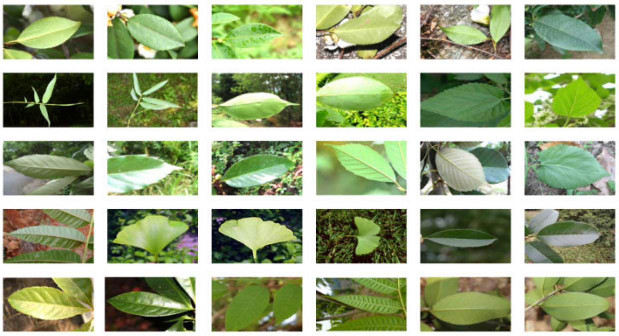


FIGURE 1. Some of the plant leaf images.

According to the plant leaf images shown in Figure 1, it is important to extract the objects from the complex background to separate the background and foreground. To solve this problem, the whole recognition process is divided into three modules, namely, data collection and sorting, image preprocessing, and image classification.

In the data collection and sorting module, the first step is to complete the shooting and downloading of the images, classifying and sorting, as well as a series of image processing operations, including cropping, noise addition, translation, rotation, and brightness enhancement. Secondly, labels are made to the sorted images, and the images are divided into a training set, a verification set, and a testing set.

In the image preprocessing module, the Mask R-CNN network model is used to segment the background of plant leaf images. The Mask R-CNN network model is roughly divided into four parts: feature extraction, region suggestion network, RoIPooling layer, and classification and regression. This article will focus on improving the first three parts of the Mask R-CNN network model. In this module, the following three works were mainly done:

(1) CNN is used to extract image features. In the Feature Pyramid Network (FPN) structure, a layer is added to the detection process of the lowest layer network, so that the feature maps of different depths can learn more detailed information. At the same time, the reverse side connection is constructed to fuse the feature maps of different depths based on the improved feature pyramid and generate multi-scale feature maps. Therefore, the feature maps after fusion can learn more features.

(2) The feature maps of different depths are input into the RPN, and a plurality of detection boxes of different sizes are generated according to the size of the bounding box set by the model itself. The detection boxes probably contain the target objects. And when there are many overlapping objects in the image to be detected, there may be a case where the Intersection over Union (IoU) ratio of the box to be detected and the detected box is greater than the threshold confidence. In this case, the miss detection will occur, resulting in a low detection rate for overlapping objects, which further reduces the average precision. Moreover, the retention and suppression measures are adopted when the IoU ratio is greater than the threshold confidence. This is not to delete the detection box, but to lower the score of the box to be detected. The most

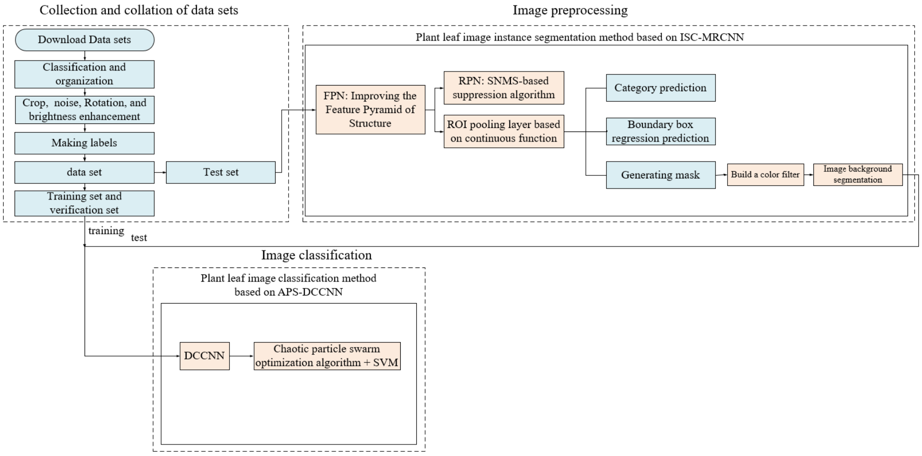

Collection and collation of data sets
Download Data sets
Classification and organization
Crop, noise, Rotation, and brightness enhancement
Making labels
data set
Test set
Training set and verification set
training
test
Image preprocessing
Plant leaf image instance segmentation method based on ISC-MRCNN
FPN: Improving the Feature Pyramid of Structure
RPN: SNMS-based suppression algorithm
ROI pooling layer based on continuous function
Category prediction
Boundary box regression prediction
Generating mask
Build a color filter
Image background segmentation
Image classification
Plant leaf image classification method based on APS-DCCNN
DCCNN
Chaotic particle swarm optimization algorithm + SVM

FIGURE 2. **Overall structure of this paper.**

suitable detection appropriate threshold confidence, which not only improves the detection effect on overlapping objects, but also increases the average precision. To select the most suitable detection box for each object, the suppression algorithm based on Soft Non-Maximum Suppression (SNMS) is used to screen out the excess detection boxes and the traditional Non-Maximum Suppression (NMS) is not used here.

(3) The RPN will generate many RoIs that may contain the target objects, and then RoI pooling the obtained RoIs. Then, the RoIs is divided into grid regions of the same size, and the pixel values in each grid region are calculated by integrating a continuous function. It means that We can guarantee that there are gradient updates during calculating the pixels and do not care about the fine precision loss caused by the bilinear interpolation method. In this way, the precision loss between the feature map and the original image is reduced to a certain extent, and more detailed information can be retained.

At last, the RoIPooled images are fully connected and fully convoluted. The full connection operation is used in the category prediction of objects and the regression prediction of the boundaries of detected objects. The full convolution operation is used to generate masks for each object. The mask is used to construct a color filter to achieve background segmentation.

In the image classification module, due to the great similarity of plant leaf images, this paper constructs a two-channel CNN model. The training set and the verification set obtained in the first module are used for the training of the above CNN model, and the weights of the convolutional layer are obtained. At this time, the last fully-connected layer and softmax classifier of the original CNN is replaced by SVM. The images with background separated from the foreground are used to construct a new testing set. Then the trained CNN is combined with the SVM model to test the new testing set, and the ACPSO algorithm is used to optimize the SVM and finally, image classification is performed.

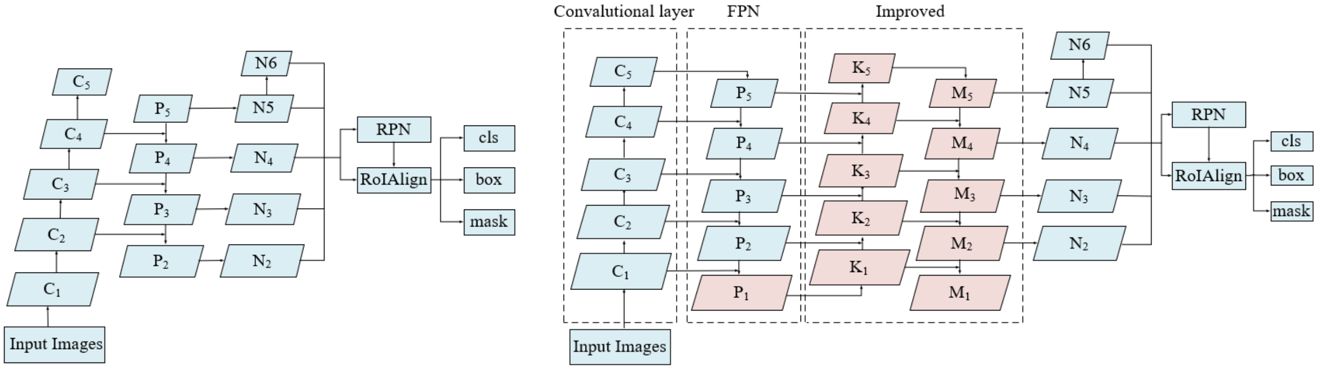
The overall structure of this paper is shown in Figure 2.

III. PLANT LEAF IMAGE PREPROCESSING BASED ON ISC-MRCNN

A. INSTANCE SEGMENTATION METHOD FOR PLANT LEAF IMAGES BASED ON IMPROVED FEATURE PYRAMID NETWORK

In a plant leaf image with a complex background, there may be several objects to be detected. These objects to be detected are located at different locations of the image, and their sizes are different. This will cause the miss detection of small objects in the detection process. It should be noted that, when using CNN to extract features of images, the low-layer network has high resolution and learns the detailed features of the images, while the high-layer network has low resolution and learns semantic features. To enable the feature maps of different depths to learn more semantic features, and learn the detailed features as much as possible, the structure in Figure 3(a) is further improved to formulate a new network structure, as shown in Figure 3(b). Thus, this paper proposes an instance segmentation method for plant leaf images based on the Improved Feature Pyramid Network (IFPN).

In literature [19], CNN was used to extract features, form side connections, and construct a network structure of feature pyramids. The network structure is shown in Figure 3(a). The network structure divides the process of extracting features by CNN into five stages, i.e. $C_1 \sim C_5$. The process of generating the side connections is divided into $P_2 \sim P_4$, and the process of generating the final feature map is divided into $N_1 \sim N_6$. At the same time, the network structure directly uses the various feature maps generated in the CNN to make predictions, so that the feature maps of different depths can learn the same semantic features and abandon the traditional FPN structure. For example, 1) only the last layer of feature map is used for prediction, but the mapping of the feature map from large size to small size causes the loss of details. As a consequence, the small objects that are not obvious will be ignored, so that the performance of detecting small objects is drastically reduced. 2) Prediction is performed using only a single-layer feature map, but the detailed features are lost. 3) Various scales of an



(a) The network structure

(b) The improved network structure

FIGURE 3. Flow chart of the detection framework before and after mask R-CNN improvement.

image are used to train and test the images, where images are scaled into multiple scales, and the feature map of each-scale image is extracted separately for prediction, which can obtain better results, but it is time-consuming and is not suitable for practical applications.

In the network structure, the pyramid on the left is a bottom-up process, which is a common forward propagation process of a neural network. Through convolution kernel calculation, the feature map is gradually becoming smaller, and the semantic features become richer. The pyramid on the right is a top-down process that upsamples the more abstract, more semantic high-layer feature map and laterally connects the feature to the previous layer and maintains that the horizontally-connected two-layer features are consistent in spatial size. This has enhanced the advanced features. The feature maps used in the prediction of each layer combine the features of different resolutions and different semantic strengths, which helps detect the objects with different resolutions and sizes. Thus, each layer has the appropriate resolution and advanced semantic features. Then the feature map of the bottom layer is downsampled and its size is kept consistent with that of the upper layer. The downsampled feature map is horizontally connected with the side edge of the feature map of the upper layer to construct a bottom-up process. Besides, a smaller size downsampling is added at the top, the feature map of each layer obtained in the process is input to the region proposal network to generate region proposals. In this way, FPN solves the multi-scale problem in object detection, and through simple network connection changes, the performance of small object detection is greatly improved without substantially increasing the calculation amount.

Although the above network structure predicts the feature maps of different sizes extracted by CNN, and the feature maps of different depths can learn the same semantic features, it is found that the feature map of the lowest layer extracted by CNN is used only once. Although the above method retains the semantic features of the feature map as much as possible, the details of the bottom- layer networks are also lost. Many researchers have investigated this problem and tried to solve it, but most of them have failed to retain more details [42].

The improved network structure is shown in Figure 3(b), where C_i is the shared convolutional layer, $P_i(1 \leq i \leq 5)$ is FPN, and $K_i(1 \leq i \leq 5)$, $M_i(1 \leq i \leq 5)$ and $N_i(2 \leq i \leq 6)$ are the feature maps generated by the proposed method.

1)Add a layer of detection step to the FPN, and the original $P_2 \sim P_5$ is changed to $P_1 \sim P_5$, so that the number of detected details are increased.

2) Generate feature maps of different sizes for the FPN, construct a bottom-up reverse side connection, and add a bottom-up path, as shown by $K_1 \sim K_5$ in Figure 4(b). K_1 has the same size as P_1, a 3 × 3 convolution with a step size of 2 is performed on $K_j(1 \leq j \leq 4)$ to obtained a feature map of the same size as $P_{i+1}(1 \leq i \leq 4)$ and is then added with $P_{i+1}(1 \leq i \leq 4)$, followed by a convolution operation to obtain K_{j+1} $(1 \leq j \leq 4)$.

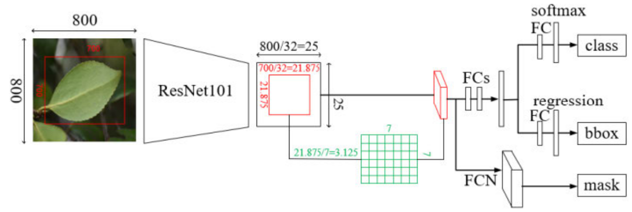


FIGURE 4. Principle of RoIPooling layer.

3) Fuse multi-scale feature maps. According to Figure 3(b), the specific steps are as follows. N_5 is processed by 1 × 1 convolution to get M_5. $M_s(2 \leq S \leq 5)$ receives upsampling twice to obtain a feature map of the same size as $K_{r-1}(r = s)$ and is added with K_{r-1} to obtain $(2 \leq S \leq 5)$. Finally, $M_2 \sim M_5$ receives the convolution of 3 × 3 to obtain $N_2 \sim N_5$. 0.5-times downsampling is performed on N_5 to obtain N_6. To reduce the calculation, M_1 is discarded, N_2-N_6 is selected, and then the sigmoid activation function is used to get the input of RPN.

After the original FPN structure is improved, the details of the bottom layer of the plant leaf images are fully utilized, and the feature maps of the adjacent layers are fused to reduce the missing detection for small object objects by the original structure. This has further improved the overall detection precision. After passing through the output of the feature pyramid network, the feature map is input into the region proposal network, and the size of the box is used to generate a plurality of to-be-detected boxes of different sizes. Therefore, the suppression algorithm is required to filter the to-be-detected boxes.

B. INSTANCE SEGMENTATION METHOD FOR PLANT LEAF IMAGES BASED ON SOFT-NON MAXIMUM SUPPRESSION

After the features of the plant leaf images are extracted through CNN, the obtained feature map is input RPN to generate the to-be-detected boxes. The best testing result is to reserve only one optimal detection box for each object in the plant leaf image. In this process, an instance segmentation method based on Soft-Non Maximum Suppression is proposed.

The original framework adopts the NMS algorithm, which is to suppress the non-maximum value and perform the filtering of to-be-detected boxes. This is an algorithm for searching the local maximum value. NMS is used to process the numerous generated candidate boxes to remove redundant candidate boxes and obtain an optimal detection box, which speeds up the detection efficiency. The main idea of NMS is that it is probably to obtain multiple detection boxes by classifying candidate regions, and then filter the series of detection results, to get an optimal detection box. The NMS algorithm is indispensable in the current object detection algorithm. Whether for two-stage models such as Faster R-CNN or Mask R-CNN, or one-stage models like SSD and YOLO, there are many candidate boxes. However, the number of actual objects in the image is not large, far less than the number of candidate boxes. However, for the same object, multiple candidate boxes may be generated, and the degree of overlap between them is high. Therefore, to eliminate redundant detection boxes and obtain the final candidate boxes, NMS is widely used.

For traditional NMS algorithms, they are executed mainly according to the following four steps: 1) set the threshold of the coincidence degree IoU; 2) sort the boxes after candidate region classification according to their score from high to low, and select the box with the highest score; 3) traverse calculation of the ratio of the remaining detection box to the overlapping area of the box, that is, the size of IoU threshold. If the IoU value is greater than the threshold, the detection box is deleted, and the retained detection boxes are resorted according to the size of the score; 4) select the box with the highest score from the remaining boxes and repeat the above process until all boxes are processed.

The NMS algorithm is dependent on the preset threshold. If the threshold is too large, the suppression is too strong, and missing detection will occur. If the threshold is too small, the suppression will be too weak, and mistakenly predict the incorrect detection box as the correct one. Secondly, when two similar objects are approaching, the biggest problem with NMS is that it deletes the detection box whose overlap is larger than the threshold. When the real object appears in the overlapping area, the detection of the object will fail and the average precision of the algorithm will be reduced. To solve this problem, the NMS algorithm is optimized. The main idea of the SNMS algorithm is not to directly delete the boxes whose IoU is larger than the threshold, but to reduce its confidence. According to the principle of the algorithm, a confidence threshold is specified after processing, and the boxes with a score larger than the threshold are retained.

Currently, detection evaluation criteria emphasize the average precision of a detection box accurately locating and measuring multiple overlapping thresholds ranging from 0.5 to 0.95. Therefore, the application of an NMS with a low threshold of, for example, 0.3 may result in a decrease in the average precision when the overlap criterion during the evaluation of true positive is 0.7, and the detection evaluation threshold is referred to as O_t. This is because there may be a detection box b_i that is very close to the object (within 0.7 overlap), but its score is slightly lower than M, that is, M does not cover the object. Therefore, b_i is suppressed by low N_t. As the overlap threshold criteria increase, the likelihood of this situation increases. Therefore, using low N_t to suppress all nearby detection boxes increases the miss rate. Besides, when O_t is lower, using a high N_t like 0.7 increases false alarm, which reduces the average precision of multiple thresholds.

The principle of NMS algorithm is given by:

$$s_i = \begin{cases} s_i, & iou(M, b_i) < N_t \\ 0, & iou(M, b_i) \geq N_t \end{cases} \tag{1}$$

where M is the current box with the highest score, b_i is the to-be-detected box, N_t is the threshold confidence, which is set to 0.5 by literature [19]. Eq (1) is optimized based on the idea that the larger the IoU of b_i and M is, the stronger the score S_i of b_i will fall, instead of being directly set to zero as shown in Eq (1).

The proposed improvement method is an SNMS-based suppression method, as shown in Eq (2).

$$s_i = \alpha s_i e^{-\frac{iou(M,b_i)^2}{\sigma}}, \quad \forall b_i \notin D \tag{2}$$

where σ is a preset standard deviation and is set to 0.5 in this paper [43], D is the set of reserved detection boxes, and α is the suppression coefficient and set to 1. For the exponential function, the larger the IoU value, the smaller the result of the exponential function. This means that when M is the current box with the highest score, the larger the IoU of the to-be-processed box and M, the stronger the score of the to-be-detected box drops. According to the principle of the algorithm, a confidence threshold is specified after processing, and the detection box with a score greater than the threshold is retained.

Unlike in the NMS where a single threshold is used to suppress the result M that exceeds the threshold with the maximum score, the proposed method attenuates the scores of all the objects by the corresponding continuous function, and then the threshold confidence is set to retain the detection boxes with a score larger than the threshold. The proposed SNMS algorithm has improved the performance in detecting overlapping objects, and the feature map containing RoIis input into the pool. These RoIs may contain plant leaf objects. The features of the plant leaf image are further extracted for subsequent classification, regression, and generation of masks.

C. INSTANCE SEGMENTATION METHOD FOR PLANT LEAF IMAGES BASED ON CONTINUOUS FUNCTION TO FIND INTEGRAL

To ensure the alignment of the mapping between the feature map and the plant leaf image, it is important to choose the most appropriate pooling method to reduce the loss of precision to a certain extent. This paper proposes a RoIPooling method based on Continuous Function to Find Integral (CFFI) [44].

After the feature map is input into the RPN network, the output generated feature map includes an RoI, and then each RoI is subjected to a RoIAlign layer processing. The task of this stage is to make the region proposal map generated by the RPN network generate a fixed-size feature map. The feature maps generated through the RPN network have different sizes, and each RoI in the feature map also has a different size. Therefore, the feature map input to the layer has the multi-scale characteristics, and the size needs to be unified, followed by inputting the feature map of the unified size to the subsequent network for classification and regression.

In Faster R-CNN, RoIPooling performs two quantization operations, and there is always a problem of loss of precision for each quantization operation. The first is that the box predicted before RoIPooling is the float value and it is quantized here. The second is to divide the RoI on the feature map into a fixed number of grid regions (such as 7 × 7). The coordinates of the boundary points in each grid region are usually not integers, and they are rounded and quantized before pooling operation. The pixels in the RoI are discrete, with no gradient updates, so no training adjustments can be performed.

In Mask R-CNN, the usual practice of the RoIAlign layer is to select a fixed number of equally-spaced sampling points for each grid region after the RoI region is divided on the feature map and calculate the eigenvalues of the sampling points by bilinear interpolation according to the neighboring eigenvalues of the sampling points, and then averaging pooling is performed on the eigenvalues of the sampling points. RoIAlign improves the two quantization operations in ROIPooling to ensure the precision. Moreover, bilinear interpolation is performed on the N interpolated pixels in each grid region to update the gradient. However, new parameters are introduced in each grid region in the RoI, that is, the number N of points that need to be interpolated. The value of N is preset and cannot be adaptively adjusted according to the feature map. The gradient of these N points is only related to the pixel points of the N integer positions above and below, but not the pixels of the entire area.

In summary, RoIPooling is to pool the divided grid regions, and there exists quantization error in calculating the coordinates of the boundary points of the grid. By contrast, RoIAlign performs the polling by sampling the grid region, which is more accurate due to the use of interpolation to calculate the eigenvalues of the sampling points. However, in this process, the pixel-to-pixel operation needs to ensure that the RoI feature maps to the alignment of the original image, and RoIAlign is used to solve the alignment problem, which reduces the pixel-level alignment error to some extent. The method of RoIAlign only reduces the error to a relatively small extent, but there is still a large error.

To further reduce the information lost during the RoI feature mapping process, we set the network structure of the RoIAlign layer as shown in Figure 4.

RoIPooling, RoIAlign, and Proposed Polling are all extracting the features based on RoI coordinates and feature maps, but they differ in calculation methods, as shown in Figure 5.

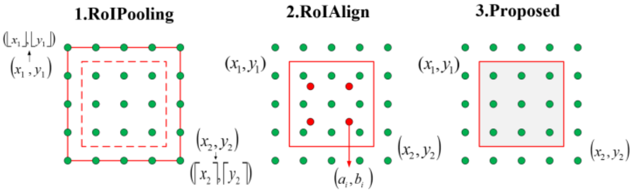


FIGURE 5. Comparison of three pooling methods of RoIPooling, RoIAlign and proposed pooling.

RoIPooling pooling is expressed in Eq (3). After the second (quantitative) rounding of (x_1, y_2) and (x_2, y_2), the upper and lower boundaries of each grid region are obtained, where $w_{i,j}$ is the weights corresponding to the pixel points.

$$\frac{\sum_{i=\lfloor x_1 \rfloor}^{\lceil x_2 \rceil} \sum_{j=\lfloor y_1 \rfloor}^{\lceil y_2 \rceil} w_{i,j}}{(\lceil x_2 \rceil - \lfloor x_1 \rfloor + 1) \times (\lceil y_2 \rceil - \lfloor y_1 \rfloor + 1)} \tag{3}$$

RoIAlign pooling is expressed in Eq (4), where N is the number of points that need to be interpolated in the grid region and f is a bilinear interpolation function.

$$\sum_{i=1}^{N} f(a_i, b_i)/N \tag{4}$$

Proposed Pooling is expressed in Eq (5) ∼ Eq (7), where (x,y) is the pixel value in the grid region, i, j is the order of the pixel points, $w_{i,j}$ is the weight, (x_1,y_2) and (x_2, y_2) are the upper and lower boundaries of each grid region.

$$\begin{aligned} IC(x, y, i, j) &= \max(0, 1 - |x - j|) \\ &\quad \times \max(0, 1 - |y - i|) \end{aligned} \tag{5}$$

$$f(x, y) = \sum_{i,j} IC(x, y, i, j) \times w_{i,j} \tag{6}$$

$$\frac{\int_{y_1}^{y_2} \int_{x_1}^{x_2} f(x, y) dxdy}{(x_2 - x_1) \times (y_2 - y_1)} \tag{7}$$

Proposed Pooling uses the integral method to calculate the value of each grid region. The biggest difference between Proposed Pooling and RoIAlign is that when calculating the value of a grid region, Proposed Pooling not only considers the mean value of N interpolation points in the grid region, but also regards the interpolation in the grid region as continuous. Therefore, the sum of the points surrounded by the grid region can be obtained by integrating all the interpolation points in the grid, and finally obtain the value of the grid region through dividing it by the area. That is to say, the results are more accurate. The RoI features are calculated by the integral method, so that the error of the forward calculation is further reduced. Besides, the back-propagation is based on the continuous input value to calculate the gradient so that the back-propagation can be continuously guided. As a comparison, RoIPooling and RoIAlign inevitably bring some noise due to the quantization or interpolation of several points when obtaining the RoI features, and only return the gradient to the specific input in the reverse derivation.

Therefore, instead of using bilinear interpolation, this paper adopts the method of obtaining integrals using a continuous function in the grid region. The Proposed Pooling method can ensure the perfect alignment of the pixels between the pixels of the feature map and the plant leaf image in the mapping process, which can reduce the loss of precision. The feature map output at this stage is going to take full connection, conduct object classification, and boundary regression prediction, followed by full convolution to generate masks.

IV. CLASSIFICATION OF PLANT LEAF IMAGES BASED ON APS-DCCNN

A. CLASSIFICATION OF PLANT LEAF IMAGES BASED ON APS-DCCNN

After tree leaf images have been pre-processed, a single background image is obtained. At this time, it is the most effective method to classify the images by CNN. However, due to the great similarity between tree species, a two-channel CNN network model is constructed in this research to distinguish the types of tree leaves. At the same time, SVM also shows good performance in multi-classification problems, and has the best reliability of classification among machine learning tools. Therefore, the feature vectors extracted by the CNN convolutional layer are sent to the SVM for classification. The generalization ability of SVM is better than that of the softmax, and then the ACPSO algorithm is used to optimize the SVM. On this basis, this paper proposes a tree leaf image classification method based on APS-DCCNN.

The classification model of SVM is constructed by training data. The selection of kernel function and the setting of parameters can directly affect the performance of the classifiers. The radial kernel function has been widely used due to its strong learning skills for the classification problems with complex multi-factors. In the SVM of the radial kernel function, the penalty factor C and the kernel function parameter σ are the main parameters affecting the precision rate of SVM-based classification. To get a good classification effect, these two parameters must be reasonable. The penalty factor C adjusts the confidence range and empirical risk ratio of the learning machine in the feature space; too large C will reduce the generalization ability, and too small C leads to the failure of the classification model due to reduced classification precision. The value of the kernel function parameter σ affects the separation result of the feature space, and also affects the generalization ability of the system to some extent. Selecting the most appropriate values of C and σ can effectively improve the classification precision of the SVM.

Unsuitable parameters and features can both lead to poor classification results. For a specific classification, all features are not equally important, and some redundant or even irrelevant feature information will inevitably be included. Thus, discarding these unimportant features will improve the classification effect. Besides, reducing the spatial dimension of the input features may also reduce the computational time. In summary, it can be seen that to obtain a fast and accurate SVM classifier, it is required to solve the problem of selecting the optimal feature subset and the optimal parameters, and the selection of parameters will affect the selection of the feature subset. Therefore, this paper adopts an ACPSO algorithm to optimize SVM, and to conduct the feature and parameter selection for SVM.

On this basis, this paper combines a variety of mainstream deep CNN models to construct a two-channel CNN model and use SVM to replace the last layer of the DCCNN network model and the softmax classifier. That is, the output of the second-to-last layer of the fully connected layer is transformed into a feature vector and input into SVM for classification. At the same time, to solve the problem of parameter and feature selection, the ACPSO algorithm is used to optimize the SVM to further improve the running speed and recognition effect. The constructed network model and parameter setting of each layer are shown in Figure 6 and Table 1.

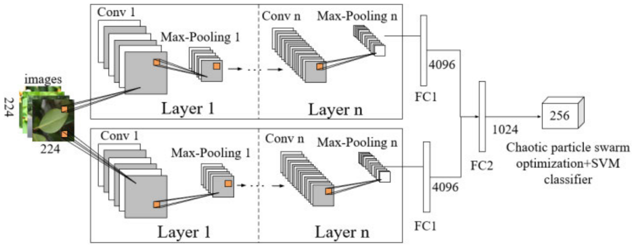


FIGURE 6. Comparison of three pooling methods of RoIPooling, RoIAlign and proposed pooling.

In Figure 6, Flow A and Flow B are used in the model to represent the two data streams of the model. The two data streams use the same data input and are fused at the top of the exchange stream to extract more abundant image features. The same convolution kernel is used in all convolution

TABLE 1. Layer settings of DCCNN model.

Operation type	Flow A		Flow B	
	K/S/N	F/M_{out}	*K/S/N*	F/M_{out}
Conv1	3×3/1/2	224×224/64	7×7/2/2	112×112/64
Max pooling	2×2/2	112×112/128	2×2/2	56×56/128
Conv2	3×3/1/2	112×112/128	7×7/1/2	56×56/128
Max pooling	2×2/2	56×56/256	2×2/2	28×28/256
Conv3	3×3/1/3	56×56/256	5×5/1/3	28×28/256
Max pooling	2×2/2	28×28/512	2×2/2	14×14/512
Conv4	3×3/1/3	28×28/512	5×5/1/3	14×14/512
Max pooling	2×2/2	14×14/512	2×2/2	7×7/512
Conv5	3×3/1/3	14×14/512	3×3/1/3	7×7/512
Max pooling	2×2/2	7×7/512	2×2/2	7×7/512
FC1	—	1×1/4096	—	1×1/4096
FC2	1×1/1024			
SVM	1×1/256			

operations in Flow A, and convolution kernels of different sizes are used in Flow B to extract the features different from those in Flow A, thereby increasing the robustness of the features.

In Table 1, is the size of the convolution kernel or pooled core of the layer, is the convolution or pooling step size, is the number of consecutive convolution or pooling in the layer, F is the output characteristic map of the layer, is the number of output feature maps, Type represents the parameters of each layer in the model, where Conv is the convolution operation and FC is the full convolution operation.

The specific steps for applying this model are as follows:

Step1: Process the sample data.

Step2: Build the architecture of the DCCNN model.

Step3: Train with training data to obtain the weight parameters of the convolution layer.

Step4: Convert the convolutional layer output to an SVM output vector.

Step5: Train the SVM with the feature vector obtained in the previous step.

Step6: Use the test dataset for precision testing.

Therefore, the advantage of using the DCCNN network over the traditional single-channel models is that it can extract more feature information and fuse these features. The method is used as the classification and recognition of tree leaf images.

B. THE METHOD FOR OPTIMIZING SVM BASED ON ACPSO ALGORITHM

1) ADAPTIVE CHAOTIC PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) [45] is an efficient parallel search evolution algorithm based on swarm intelligence, but it has the problems of premature convergence, easy to fall into local extremum and low precision. To solve these problems, many efforts have been made and effective improvements have been seen. The Chaotic Particle Swarm Optimization (CPSO) algorithm is one of the PSO algorithms, and it has been greatly improved in terms of convergence speed and classification precision.

The basic idea of the ACPSO algorithm is as follows: using chaotic sequence to initialize the speed and position of particles, thereby improving the ergodicity of particle search and swarm diversity, without generating randomness, and then generating a large number of initial swarms and finally finding the excellent initial swarms. Based on the optimal position searched by the current particle swarm, the chaotic sequence is generated, and the particle with the optimal position in the generated chaotic sequence is replaced as one particle in the current particle swarm. The search algorithm that introduces chaotic sequences generates many neighborhood points of the local optimal solution in the iteration, to quickly search for the optimal solution.

This paper uses a full mapping in the chaotic system, i.e. Logistic mapping, which is expressed as:

$$\beta_{i(k+1)} = \mu\beta_{ik}(1-\beta_{ik}) \tag{8}$$

where μ represents the control parameter is represented, and $\beta_{ik}(i = 1, 2, 3, 4, \ldots\ldots, N)$ is the k-iterated N-dimensional chaotic vector.

Let $x_k = (x_{1k}, x_{2k}, \cdots x_{Nk})$ be the vector mapped in [0,1] of the current chaos operation, $\beta_k = (x_{1k}, x_{2k}, \cdots x_{Nk})$ be the chaotic vector after the k iterations, then the chaotic vector x_{k+i} after the random perturbation can be expressed as:

$$x_{k+i} = (1-a)x_k + a\beta_k, \quad (0 < a < 1) \tag{9}$$

To ensure that the chaotic variables can be fully traversed, the number of iterations k of the chaotic sequence is generally set to about 500. Finally, the N particle swarms with the highest fitness are selected to form the initial swarm. Let f_i represent the fitness value of the particle at current iteration time, f_g be the fitness value of the optimal particle, and f_{avg} be the average value of the current fitness value of all the particles, f'_{avg} is the average value of the fitness value better than f_{avg}.

The adaptively-adjusting inertia weight ω of the particle with fitness value f_i is: (1) When the particle is the optimal particle in the swarm, f_i is better than f'_{avg} and approaches the global optimal value, and the inertia weight should be small to accelerate the global optimal convergence. The formula for adjusting the inertia weight according to the particle fitness value is given by:

$$\omega = \omega - (\omega - \omega_{\min}) \times \left| \frac{f_i - f'_{avg}}{f_g - f'_{avg}} \right| \tag{10}$$

where $\omega_{min} = 0.5$ and is the minimum value of ω. The better the fitness value is, the smaller the inertia weight is, and the local optimization can be enhanced. (2) When the particles are general in the swarm, f_i is better than f_{avg} but worse than f'_{avg}. At this time, the correction formula of the inertia weight is given by:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \frac{1 + \cos(T-1)\pi/(\max ST - 1)}{2} \tag{11}$$

where, the maximum inertia weight at the beginning of the search is the minimum inertia weight at the end of the search, and T is the number of iteration steps, which is the maximum number of iteration steps allowed. (3) When the particles are poor in the swarm, then the adjustment formula of the inertia weight is:

$$\omega = 1.5 - \frac{1}{1 + k_1 \times \exp(-k_2 \times (f_{avg} - f'_{avg}))} \tag{12}$$

where k_1, k_2 are usually the constants greater than 1. The value of $f_{\text{avg}} - f'_{\text{avg}}$ can be used to evaluate the degree of premature convergence of the particle swarm. The smaller the value of $f_{\text{avg}} - f'_{\text{avg}}$, the more likely the particle swarm tends to prematurely converge.

However, the CPSO algorithm lacks flexibility to some extent. When the particle swarm approaches the local optimal state, the particles' update speed is usually becoming smaller or even stopped. In this case, the inertia weight needs to be artificially changed to make it effectively jump out local optimal to avoid premature convergence. Therefore, the basic steps of the ACPSO algorithm (ACPSO) are as follows.

Step1: Initialize the particle swarm, set the maximum number of iterations allowed, inertia weight ω, learning factors c_1 and c_2, and chaotic perturbation range $[-\beta, \beta] = [-0.5, 0.5]$.

Step2: Initialize the chaotic particle swarm. Randomly generate a D-dimensional vector $z = (z_1, z_2, \cdots, z_D)$ with each component value between 0 and 1, and use Logistic mapping iteration to generate N vectors $z_1, z_2, \cdots, z_1, z_N$, and take each component carrier to the corresponding range of particle swarms, calculate the fitness value of the particle swarm, select M solutions with better performance from the initial swarm as the initial solution and randomly generate M initial speeds.

Step3: Evaluate the fitness of each particle. If the particle fitness f_i is better than the individual extremum or the global extremum, the individual and global extremum are updated accordingly.

Step4: Update the position and speed of the particles according to the formulae.

Step5: Perform chaotic optimization in the optimal position. The chaotic variable sequence is generated according to the Logistic chaotic mapping equation, and is returned to the original solution space through inverse mapping. The fitness value f_i of each feasible solution of chaotic variables is calculated in the original solution space, and the best feasible solution is obtained.

Step6: Replace the position of the particles in the current swarm with the best possible solution.

Step7: Determine whether the condition is met. If it is satisfied, stop and output the optimal position. Otherwise, adopt the corresponding adaptive strategy to adjust the inertia weight ω according to the specific fitness value of f_i, and go to Step 3 to continue the calculation.

2) SUPPORT VECTOR MACHINE OPTIMIZATION METHOD BASED ON ADAPTIVE CHAOTIC PARTICLE SWARM OPTIMIZATION

In this paper, the CPSO algorithm is applied to the feature selection and parameter setting of SVM, and it is applied to the classification of plant leaf images. The specific steps are as follows:

Step1: Extract features from CNN and convert them into feature vectors, including textures, shapes, etc., which is D-dimension in total. Since the features cover different values, to prevent the features with a larger range of values from weakening those with a smaller range of values, the feature data must be normalized.

Step2: Initialize ACPSO, perform chaotic initialization on the speed and position of each particle in the swarm.

Step3: Train the SVM classifier, select feature subsets and parameters, get corresponding penalty factor and kernel parameter according to the latest position of the particle, and use it to train the SVM classifier.

Step4: Update the swarm, calculate the fitness of the particles, adaptively adjust according to the fitness value, adjust the optimal position and global optimal position of the individual, update the speed and position of each particle, and optimize the optimal position. A feasible solution with good performance is obtained to replace the position of any particle in the current swarm with a feasible solution.

Step 5: Termination condition. When the termination condition is met, the training of the SVM classifier is completed, and the optimized parameter combination and feature subset are obtained. Otherwise, continue to return to the Step3.

Step6: Retrain the SVM classifier according to the optimized parameters and feature subset, and classify the test data.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL DATA COLLECTION AND PREPROCESSING

The leaf images used for pre-training is the Folio [46] leaf image dataset of UCI Machine Learning Repository [47], which was released by the Center for Machine Learning and Intelligent Systems at the University of California, Irvine. The image dataset contains 32 types of plant leaf samples, with 20 images for each type. After pre-processing operations such as brightness enhancement, vertical flipping, and horizontal rotation, the number of images is expanded to 9, 200. The pre-processed dataset is used for the training of the network model to obtain the weights. Some of the leaf images are shown in Figure 7.

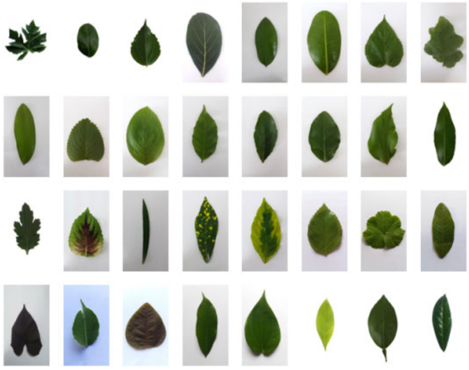


FIGURE 7. Plant leaf image dataset of Folio.

The images obtained by preprocessing one of the images are shown in Figure 8.

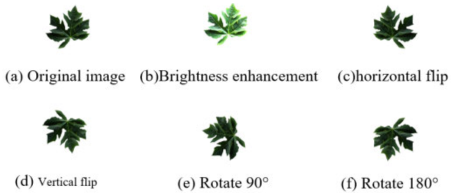

(a) Original image (b)Brightness enhancement (c)horizontal flip
(d) Vertical flip (e) Rotate 90° (f) Rotate 180°

FIGURE 8. The images obtained after preprocessing an image.

Besides, Canon HD cameras were used to capture and collect image data at Huangfengqiao State-owned Forest Farm in Youxian County, Zhuzhou City, Hunan Province. The collected data set was uploaded to the Chinese Plant Image Library [48] for expert certification, which is the largest library of plant classification in China. The dataset contains high-definition images taken in different field environments with different complex backgrounds, overlapping objects, different illuminations, etc. The database contains the images of leaves, flowers, and fruits of the plants. We screened out the leaves, sorted and labeled the leaves, and named the collated dataset as CSUFT20, which contains 20 types of leaves of southern China plants, a total of 100, 000 images.

The plant species and quantities contained in the two datasets are shown in Table 2, and the plant species in the two datasets are not substantially duplicated.

TABLE 2. Image dataset of plant leaves.

Data set	Plant species	Numbers
Folio Data Set	32	9200
CSUFT20	20	99413

In the experiment, all image sizes were uniformly processed into a size of 224 × 224. Then, 60% of the dataset was randomly selected as the training set, 20% as the verification set, and the remaining 20% was used as the testing set. At the same time, the leaf images were subjected to pre-processing operations such as background enhancement, noise addition, flipping and rotation, image shifting, and zooming.

The images after pre-processing of an image in the CSUFT20 are shown in Figure 9.

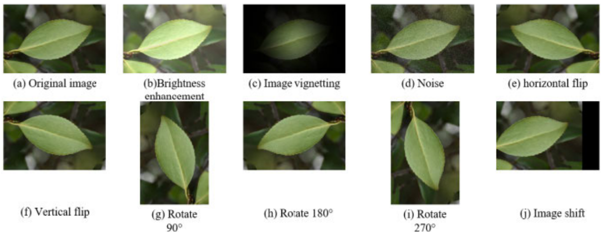
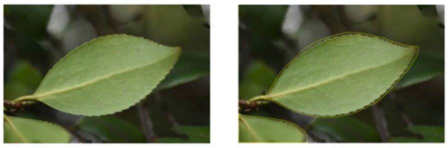

(a) Original image (b)Brightness enhancement (c) Image vignetting (d) Noise (e) horizontal flip
(f) Vertical flip (g) Rotate 90° (h) Rotate 180° (i) Rotate 270° (j) Image shift

FIGURE 9. The images after pre-processing of an image in the CSUFT20.

Mask R-CNN needs to train the data before performing object detection and instance segmentation on an image. Data training requires an annotated dataset, so the dataset needs to be labeled first. In this experiment, 1000 images were selected from the training set of CSUFT20 as the training set of Mask R-CNN, 1000 images from the verification set of CSUFT20 as the verification set of Mask R-CNN, and all testing sets of CSUFT20 as the testing set of Mask R-CNN. Besides, 60 epochs were trained. This experiment did not use a large amount of images for the training of Mask R-CNN but used migration learning to select the training model from the pre-trained weights provided in the COCO dataset [49], meaning that it is not necessary to train the model from scratch. Instead, the early stages of using CNN as a feature extractor were ignored to directly accelerate the later stages of the Mask R-CNN model.

The image annotation tool selected in this paper was VIA [50](VGG Image Annotator), which can be used by simply downloading and opening a single HTML file in a web browser. The VIA tool saves the annotated annotations in a JSON file, which generates a set of polygon points for each object in the image, and stores the coordinates in the form of (x, y). The network needs a 'mask image' when training, and this was temporarily generated from the JSON file loading polygon coordinate points. An image is selected for the training of Mask RCNN and is labeled, as shown in Figure 10.



(a) Original image (b)The annotated image

FIGURE 10. Labeled image.

B. EVALUATION INDICATORS AND EXPERIMENTAL RESULTS

1) EVALUATION INDICATORS

The evaluation indicators in this paper consist of two parts:

(1) For the ISC-MRCNN model, the mean average precision (mAP) and the *Precision-Recall* (P-R) curve under different thresholds were used. *Precision* and *Recall* are two important metrics in the field of information retrieval and statistical classification to evaluate the quality of certain results. *Precision* is the proportion of samples that are correctly predicted in all samples. *Recall* is the proportion of samples that are positively predicted to be positive. To solve the problem of the single point value limitation of P and R, mAP is used, which is defined as the mean of the average precision at different points of Recall, as shown in Eq (13).

$$mAP = \int_0^1 P(R)dR \tag{13}$$

To obtain an indicator that can reflect the global performance, Recall is used as the horizontal axis, Precision as the vertical axis to draw the P-R curve. The larger the area enclosed by the curve and the x and y axes, the better the performance.

2) For the CNN model, the evaluation indicators include the loss of verification and the comparison of recognition precision. The verification loss is used to judge whether the model has over-fitting and the generalization ability in the iterative process, and the precision is used to judge the performance of the model itself. The curves in this paper were all drawn from the data obtained by Python's drawing library Matplotlib, which were used to analyze the convergence of CNN and the precision of network recognition.

2) EXPERIMENTAL RESULTS AT THE IMAGE PREPROCESSING MODULE

In the ISC-MRCNN model, the ResNet101 network model was used as the feature extractor, and the feature pyramid network was also used together to form the backbone network. The ResNet101 network model is divided into five stages of C1∼C5, and the partial feature map is shown in Figure 11.

In the model training process, the weight distribution histogram was detected to know the change of the weights of the model during the training process. The partial weight histogram is shown in Figure 12, where the horizontal axis indicates the weight value, the vertical axis represents the number of weight values, and the weight of values around 0 has the largest number, indicating the excellent performance of the model.

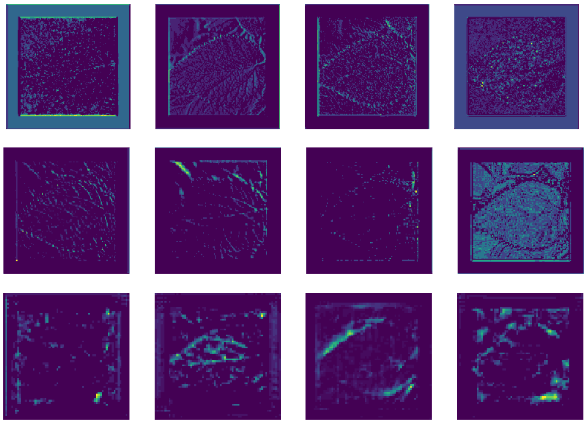


FIGURE 11. The feature map of the backbone network.

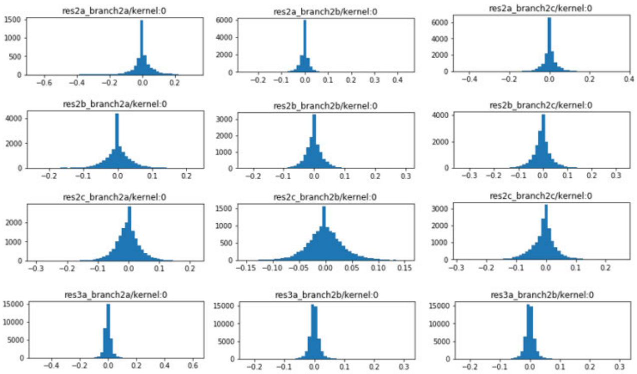


FIGURE 12. Weight distribution histogram.

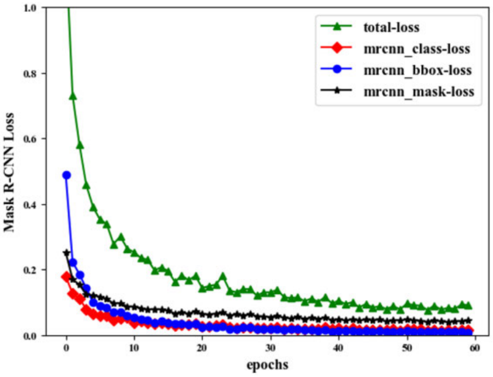


FIGURE 13. Various losses in the training of ISC-MRCNN model.

The loss of ISC-MRCNN is composed of the category prediction branch, the box prediction branch, and the loss of the segmentation mask branch. The training loss of the ISC-MRCNN model is shown in Figure 13.

The loss of RPN is composed of category prediction and box prediction, and the loss of training is shown in Figure 14, where the horizontal axis represents the number of training batches and the vertical axis represents the loss.

When the ISC-MRCNN model is trained and verified, the image needs to be input into the network, and the

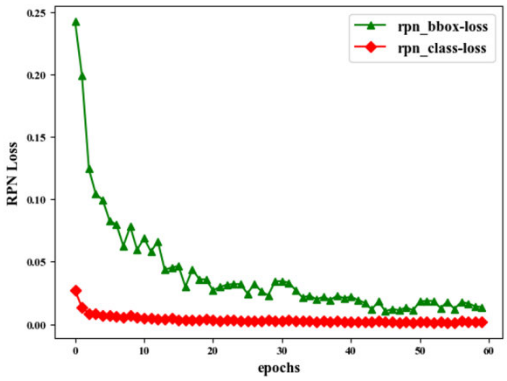


FIGURE 14. RPN loss in the training of ISC-MRCNN model.

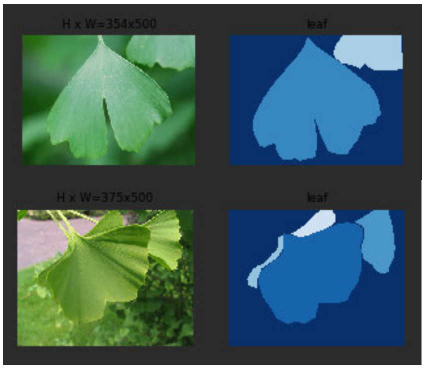
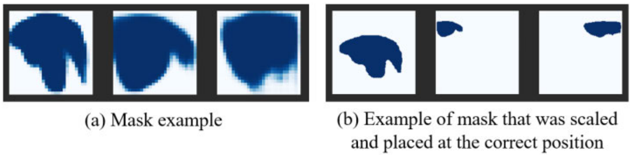


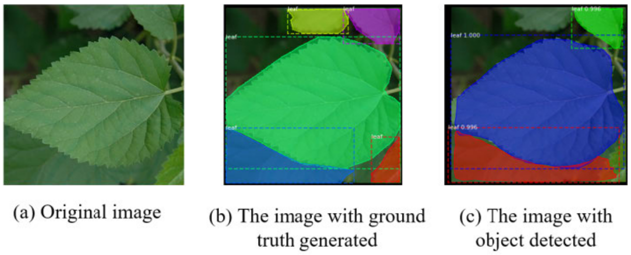
FIGURE 15. Example of applying a mask.



(a) Mask example

(b) Example of mask that was scaled and placed at the correct position

FIGURE 16. Example of mask for training and verification.

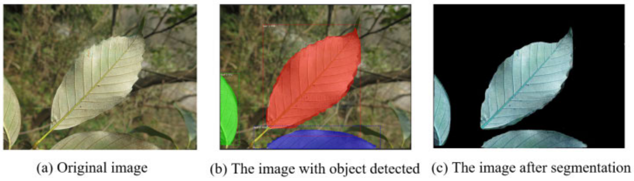


(a) Original image

(b) The image with ground truth generated

(c) The image with object detected

FIGURE 17. The ground truth image and the image with the object detected.



(a) Original image

(b) The image with object detected

(c) The image after segmentation

FIGURE 18. An example of instance segmentation by ISC-MRCNN.

corresponding annotation information that needs to be loaded is equivalent to the label of the convolutional neural network. All the annotation information of the image was stored in a JSON file, and the serial number of the loaded image was used to find the boundary coordinates of the object areas contained in the image. An image may contain multiple objects. Each object corresponds to a set of coordinates, and an example image of the generated mask is shown in Figure 15, where the left part is the original image, and the right part is the example image after the mask is applied.

In the mask branch of ISC-MRCNN, the pooled feature map was obtained from RoI, and a segmentation mask was generated for each instance in the feature map, that was, one instance for one mask. An example diagram is shown in Figure 16, where Figure 16(a) is an enlarged mask diagram. And Figure 16(b) is a mask diagram which was scaled and placed at the correct position.

In the training and verification of the ISC-MRCNN model, according to the sequence number of the image shown in Figure 17(a), the ground truth data of the image was obtained, and the ground truth data specifically includes:

1) Return the image, with its form being like [height, width, 3], where height and width are the height and width of an image, respectively.

2) Return the original shape of the image before resizing and cropping.

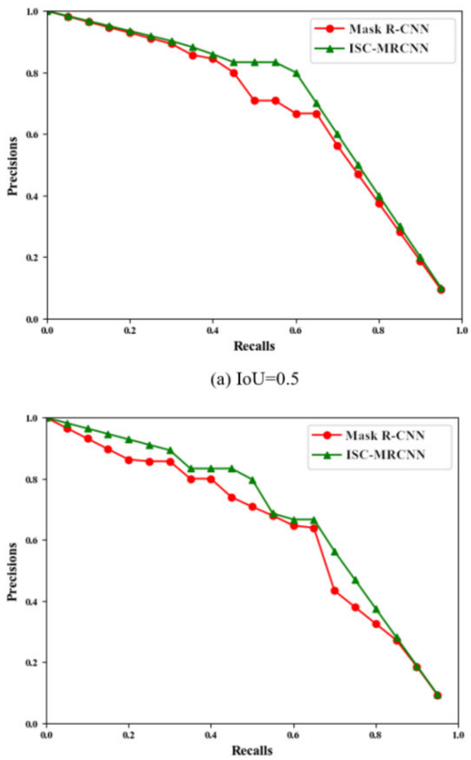
3) Return the category number.

4) Return the ground truth boxes, with its form being like [instance_count, (y_1, x_1, y_2, x_2)], where instance_count represents the number of objects in the image, and (y_1, x_1, y_2, x_2) represents the upper left and lower right corners of the box.

5) Return the mask, with its form being like [height, width, instance_count], where height and width are the height and width of the image. Based on the obtained ground truth data, a ground truth image was generated, as shown in Figure 17(b), and each object had a mask, a box, and a category. In the RPN, 9 detection boxes were generated on a reference size of 16 × 16 according to the ratios of 0.5, 1, and 2. The scores of these detection boxes were calculated by ground truth boxes, so that the subsequent detection boxes were filtered. The best box for each object was obtained, as shown in Figure 17(c).

Based on object detection and instance segmentation, a color filter was constructed to segment the background of an image. Firstly, a template image of the same size as the original image was constructed. For example, a black image with three channels of RGB was multiplied with the pixels of the mask at the corresponding position. The mask generated was the matrix that stores True and False. The matrix was transformed into a binary matrix composed of 0 and 1, and the computed pixels were assigned to the template image. An image was selected from the testing set of Mask R-CNN for instance segmentation. The segmentation effect is shown in Figure 18.

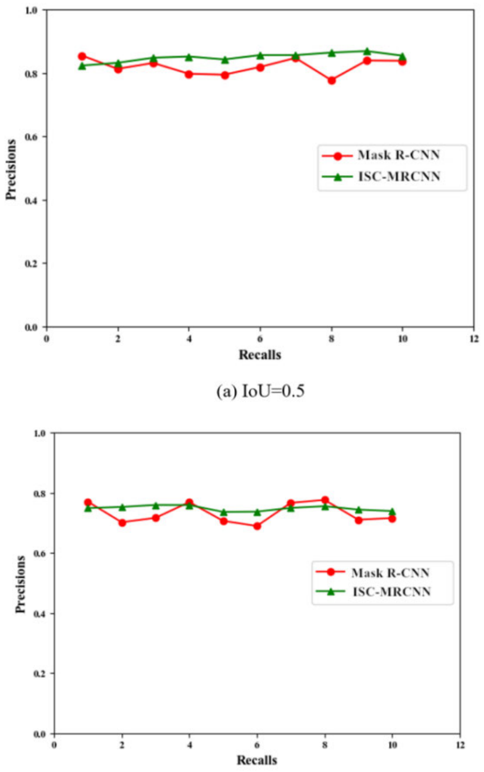
When the ISC-MRCNN model performed object detection on plant leaf images, the precision and recall under different



(a) IoU=0.5

(a) IoU=0.75

FIGURE 19. P-R curves of Mask R-CNN and ISC-MRCNN at different thresholds.



(a) IoU=0.5

(b) IoU=0.75

FIGURE 20. Average precision of Mask R-CNN and ISC-MRCNN at different thresholds.

TABLE 3. Comparison of mAP results of instance segmentation.

Method	Backbone	AP	AP_{50}	AP_{75}
MNC[51]	ResNet-101-C4	24.6	44.3	24.8
FCIS[36]	ResNet-101-C5	29.2	49.5	-
Mask R-CNN[19]	ResNet -101-FPN	35.7	58.0	37.8
TensorMask[52]	ResNet-50-FPN	35.4	57.2	37.3
TensorMask[52]	ResNet-101-FPN	37.1	59.3	39.4
YOLACT[23]	ResNet-101-FPN	31.2	50.6	32.8
SOLO[24]	ResNet-50-FPN	36.8	58.6	39.0
SOLO[24]	ResNet-101-FPN	37.8	59.5	40.4
PolarMask[26]	ResNet-101-FPN	30.4	51.9	31.0
PolarMask[26]	ResNeXt-101-FPN	32.9	55.4	33.8
CondInst[28]	ResNet-50-FPN	37.8	59.1	40.5
CondInst[28]	ResNet-101-FPN	39.1	60.9	42.0
SOLOv2[29]	ResNet-50-FPN	38.8	59.9	41.7
SOLOv2[29]	ResNet-101-FPN	39.7	60.7	42.9
Mask R-CNN(ours)	ResNet-101- FPN	60.35	82.13	73.23
ISC-MRCNN(ours)	ResNet-101- FPN	62.24	84.57	74.38

IoU ratios were calculated in the process of classification. When IoU = 0.5, the Recall and Precision were calculated using the Mask R-CNN model and the ISC-MRCNN model, respectively, and the calculated Recall was taken as the horizontal axis, and the Precision as the vertical axis to draw the P-R curve, as shown in Figure 19(a). When IoU = 0.75, the P-R curve was plotted as shown in Figure 19(b). It can be seen from the analysis of Figure 19 that the area enclosed by the P-R curve and the coordinate axes became larger under different IoU thresholds due to the application of the proposed method, indicating that the overall performance of the network was improved by the proposed method.

When IoU = 0.5, the average precision after 10 times of calculation using Mask R-CNN and ISC-MRCNN are shown in Figure 20(a). When IoU = 0.75, the corresponding result was also obtained and shown in Figure 20(b), where the horizontal axis represents the number of calculations and the vertical axis represents the average precision. As can be seen from Figure 20, after the improvement by the NMS algorithm, the detection effect of the network on the overlapping objects has been improved, and the overall average precision has been further improved.

As shown in Table 3, P_{AP} is the average precision from a total of 10 thresholds T_{IoU} from 0.5 to 0.95 with 0.05 as the step size. Similarly, P_{AP50} and P_{AP75} are the average precision values for the thresholds of 0.5 and 0.75, respectively. Table 3 shows that when the object overlapped very much, the score was usually set to zero and deleted, while the proposed method could reduce the score, which improved the detection of overlapping objects, thus making the overall average precision increase by 1.89%.

Meanwhile, the proposed method is compared with the latest instance segmentation methods. Analysis of Table 3 shows

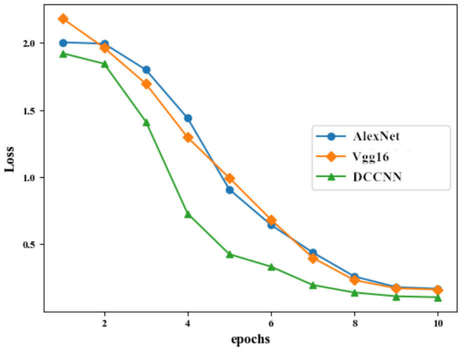


FIGURE 21. The verification loss of three CNN models.

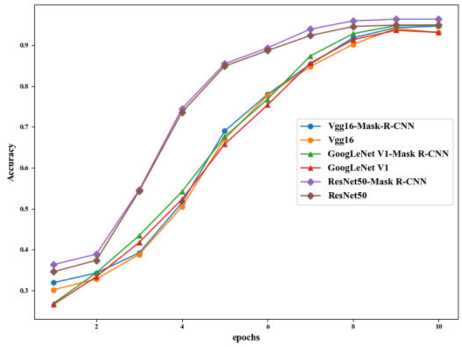


FIGURE 22. Comparison of recognition precision by three CNN models.

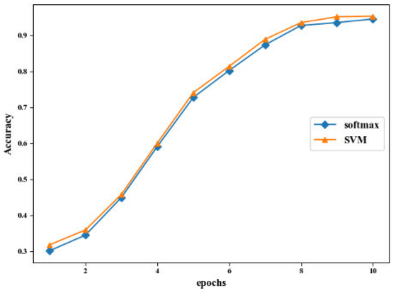


FIGURE 23. Comparison of SVM and softmax recognition accuracy.

that these methods use different backbones for feature extraction, and are all verified on public data sets, showing excellent performance. Because the proposed method is an experiment on a data set constructed by ourselves, we can not reflect the superiority of the proposed method from the front, only from the side that the proposed method has excellent performance and has a certain adaptability scenes.

3) EXPERIMENTAL RESULTS AT THE IMAGE CLASSIFICATION MODULE

This paper selects three CNN models, i.e. AlexNet, Vgg16, and DCCNN, for classification and recognition. The loss of verification is shown in Figure 21. Analysis of Figure 21 shows that the above three CNN models have no over-fitting phenomenon and show good generalization ability, among which DCCNN has the fastest convergence rate, followed by Vgg16 and AlexNet.

TABLE 4. Recognition precision of different CNN models.

Method	Accuracy	Accuracy(With ISC-MRCNN)	Percentage Increase
AlexNet	0.9308	0.9467	1.59%
Vgg16	0.9313	0.9488	1.75%
DCCNN	0.9494	0.9637	1.43%

TABLE 5. Comparison with other schemes.

Scheme	Accuracy
Mutil-feature+SVM[1]	0.9200
Mutil-feature+PCA+SVM[5]	0.9525
AlexNet+transfer learning[3]	0.9531
GoogLeNet V3+transfer learning[3]	0.9540
k-nearest neighbou[36]	0.8730
CNN[53]	0.9460
PCNN[54]	0.9667
AlexNet	0.9460
Vgg16	0.9488
DCCNN	0.9637

And the recognition precision of these three CNNs after applying the ISC-MRCNN network is compared, as shown in Figure 22. Figure 22 compares the recognition precision of the above three CNNs after applying ISC-MRCNN.

As shown in Figure 23, the horizontal axis is the training batch and the vertical axis is the accuracy of recognition. Training 10 batches, in each batch, the recognition accuracy of SVM is slightly higher than that of softmax, but the difference between them is not great. It can be seen from the analysis of Figure 23 that after building the two-channels convolutional neural network, the performance of SVM instead of softmax classifier is still excellent, which shows the feasibility of SVM instead of softmax. At the same time, the difference between the two is not very big, which shows that the training of the two-channels convolutional neural network is appropriate or the data set is clean after the preprocessing operation, which makes the network easy to identify.

As shown in Table 4, three CNN models, i.e. AlexNet, Vgg16, and DCCNN, were selected to classify and recognize the plant leaf images. Precision represents the recognition precision of image preprocessing without using the ISC-MRCNN method. Precision (With ISC-MRCNN) represents the recognition precision using ISC-MRCNN. Table 4 shows that after using ISC-MRCNN, the recognition precision of the three CNNs generally increased by 1%∼2%, and the overall average increased by 1.59%. It can be seen that when ISC-MRCNN was used, the background information of plant leaf images was eliminated, and the network model paid more attention to the object area, so that the recognition precision was improved.

As shown in Table 5, the three models used in this paper are compared with other models. Table 5 shows that the convolutional neural network model is used to identify the pre-processed data set, and its recognition accuracy is generally improved, and the recognition accuracy is higher than other models, and the recognition rate of DCCNN is significantly higher than other CNNs. model. It is indicated that extracting more feature information and performing post-fusion recognition can help improve the recognition accuracy.

VI. CONCLUSION

This paper proposes an instance segmentation and classification method for plant leaf images based on ISC-MRCNN and APS-DCCNN. Firstly, an ISC-MRCNN based plant leaf image segmentation method is proposed, which is used to preprocess the plant leaf images to remove the background interference. In the ISC-MRCNN method, the following three problems have been solved. 1) Based on the original framework of feature pyramid network, a layer of detection step is added to the bottom layer, so that the network model learns more detailed information; the FPN is used to generate the feature maps of different sizes and construct the bottom-up reverse side connection; finally, the multi-scale feature map generated by the reverse side connection is merged, so that the FPN makes full use of the multi-layer feature map to reduce the information loss in high-layer feature map processing and achieve better detection results. 2) The original NMS algorithm for Mask R-CNN cannot effectively detect overlapping objects. To solve this problem, the fusion is performed for improvements. By reducing the score of the detection box and manually setting the appropriate threshold confidence, the performance of detecting overlapping objects has been improved. 3) In the process of RoIPooling, the pixels of each grid region are calculated by using the continuous function to calculate the integral, and the existing bilinear interpolation method is abandoned, which reduces the pixel's loss in the mapping process. Finally, this paper proposes a plant leaf image classification algorithm based on APS-DCCNN. In this method, the DCCNN network model is constructed and the last layer of the fully-connected layer of the DCCNN network and the softmax classifier are replaced with SVM. Through the training and testing on the dataset, the experimental results show that compared with the Mask R-CNN, the average precision of object detection is increased by 1.89% after applying ISC-MRCNN. After applying the ISC-MRCNN method, the precision of classification by the DCCNN network model has increased by 1.43%, and the overall classification using three CNN models has improved by about 1.59%, which shows the effectiveness and robustness of the proposed method.

The SNMS-MRCNN method proposed in this paper has greatly improved the performance in detecting overlapping objects, but it still has the problem that the threshold confidence needs to be manually selected and that the appropriate thresholds are selected through multiple experiments. The future research focuses on further improving the performance in detecting the overlapping objects based on the improved NMS algorithm, and it is expected to achieve the effect of intelligently selecting the threshold confidence. We aim to continuously improve the detection performance on the plant leaf images, and further improve the average precision.

ACKNOWLEDGMENT

Useful suggestions given by Xin Zhang and Xiaohu Qiang of Central South University of Forestry and Technology are also acknowledged. The references are arranged according to the citations in the article.

REFERENCES

[1] L. Wang, Y. Huai, and Y.-C. Peng, "Identification of foliage plant species based on multi-feature fusion of leaf images," *J. Beijing Forestry Univ.*, vol. 37, no. 01, pp. 55–61, 2015, doi: 10.13332/j.cnki.jbfu.2015.01.006.

[2] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: A review," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7562–7573, Jun. 2012, doi: 10.1016/j.eswa.2012.01.073.

[3] Y. Zheng and L. Zhang, "Plant leaf image recognition method based on transfer learning with convolutional neural networks," *Trans. Chin. Soc. Agricult. Machinery*, vol. 49, no. 1, pp. 354–359, 2018, doi: 10.6041/j.issn.1000-1298.2018.S0.047.

[4] M. Ingrouille and S. Laird, "A quantitative approach to oak variability in some north London woodlands," *London Naturalist*, vol. 65, pp. 35–46, Oct. 1986, doi: 10.6041/j.issn.1000-1298.2017.03.004.

[5] Y. Zheng, "Method of leaf identification based on multi-feature dimension reduction," *Trans. Chin. Soc. Agricult. Mach.*, vol. 48, no. 03, pp. 30–37, 2017, doi: 10.6041/j.issn.1000-1298.2017.03.004.

[6] Bo Fu, Zhang Yang, Xilin Zhao, and, "Plant leaves recognition method based on dimension reduction local binary pattern and shape features of leaves," *Comput. Eng. Appl.*, vol. 54, no. 02, pp. 173–176 and 187, 2018, doi: 10.3778/j.issn.1002-8331.1608-0031.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/b:visi.0000029664.99615.94.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2005, pp. 886–893, doi: 10.1109/cvpr.2005.177.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, p. 7, doi: 10.1109/cvpr.2008.4587597.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/cvpr.2016.91.

[11] W. Liu, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988, doi: 10.1109/iccv.2017.324.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587, doi: 10.1109/cvpr.2014.81.

[15] 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, doi: 10.1109/cvpr.2015.7298965.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/tpami.2015.2389824.

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

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99, doi: 10.1109/tpami.2016.2577031.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969, doi: 10.1109/iccv.2017.322.

[20] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6409–6418, doi: 10.1109/cvpr.2019.00657.

[21] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4004–4012, doi: 10.1109/cvpr.2016.434.

[22] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," 2017, *arXiv:1708.02551*. [Online]. Available: http://arxiv.org/abs/1708.02551

[23] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9157–9166, doi: 10.1109/iccv.2019.00925.

[24] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," 2019, *arXiv:1912.04488*. [Online]. Available: http://arxiv.org/abs/1912.04488

[25] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636, doi: 10.1109/iccv.2019.00972.

[26] E. Xie, P. Sun, X. Song, W. Wang, D. Liang, C. Shen, and P. Luo, "PolarMask: Single shot instance segmentation with polar representation," 2019, *arXiv:1909.13226*. [Online]. Available: http://arxiv.org/abs/1909.13226

[27] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "BlendMask: Top-down meets bottom-up for instance segmentation," 2020, *arXiv:2001.00309*. [Online]. Available: http://arxiv.org/abs/2001.00309

[28] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," 2020, *arXiv:2003.05664*. [Online]. Available: http://arxiv.org/abs/2003.05664

[29] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOv2: Dynamic, faster and stronger," 2020, *arXiv:2003.10152*. [Online]. Available: http://arxiv.org/abs/2003.10152

[30] T. L. I. Sugata and C. K. Yang, "Leaf app: Leaf recognition with deep convolutional neural networks," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 273, Nov. 2017, Art. no. 012004, doi: 10.1088/1757-899x/245/1/012004.

[31] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995, doi: 10.1162/neco.1995.7.6.1129.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 2012, pp. 1097–1105.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[35] 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. 1–9, doi: 10.1109/cvpr.2015.7298594.

[36] 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.

[37] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. workshop Comput. Learn. Theory*, 1992, pp. 144–152, doi: 10.1145/130385.130401.

[38] Y. Tang, "Deep learning using linear support vector machines," 2013, *arXiv:1306.0239*. [Online]. Available: http://arxiv.org/abs/1306.0239

[39] Y. Chen, G. Tao, H. Ren, X. Lin, and L. Zhang, "Accurate seat belt detection in road surveillance images based on CNN and SVM," *Neurocomputing*, vol. 274, pp. 80–87, Jan. 2018, doi: 10.1016/j.neucom.2016.06.098.

[40] H. Zhang and H. Mao, "Feature selection for the stored-grain insects based on PSO and SVM," in *Proc. 2nd Int. Workshop Knowl. Discovery Data Mining*, Jan. 2009, pp. 586–589, doi: 10.1109/wkdd.2009.69.

[41] H. Gao, M. K. Mandal, and J. Wan, "Classification of hyperspectral image with feature selection and parameter estimation," in *Proc. Int. Conf. Measuring Technol. Mechatronics Autom.*, Mar. 2010, pp. 783–786, doi: 10.1109/icmtma.2010.765.

[42] Zhijun Ren, Suzhen Lin, Dawei Li, and, "Mask R-CNN Object detection method based on improved feature pyramid," *Laser Optoelectron. Prog.*, vol. 56, no. 4, pp. 174–179, 2019, doi: 10.3788/LOP56.041502.

[43] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5561–5569, doi: 10.1109/iccv.2017.593.

[44] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 784–799, doi: 10.1007/978-3-030-01264-9_48.

[45] J. Kennedy, "Particle swarm optimization," in *Proc. Encyclopedia Mach. Learn.*, 2010, pp. 760–766, doi: 10.1007/978-0-387-30164-8_630.

[46] T. Munisami, M. Ramsurn, S. Kishnah, and S. Pudaruth, "Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers," *Procedia Comput. Sci.*, vol. 58, pp. 740–747, Oct. 2015, doi: 10.1016/j.procs.2015.08.095.

[47] *Center for Machine Learning and Intelligent Systems*, Univ. California, Irvine, Irvine, CA, USA, 1965.

[48] *Plant Photo Bank of China, PPBC*. Accessed: Sep. 30, 2019. [Online]. Available: http://ppbc.iplant.cn

[49] T.-Y. Lin, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[50] A. Dutta and A. Zisserman. *The VGG Image Annotator (VIA)*. Accessed: Apr. 1, 2019. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2019arXiv190410699D

[51] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3150–3158, doi: 10.1109/cvpr.2016.343.

[52] X. Chen, R. Girshick, K. He, and P. Dollar, "TensorMask: A foundation for dense object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2061–2069, doi: 10.1109/iccv.2019.00215.

[53] C. Zhang, P. Zhou, C. Li, and L. Liu, "A convolutional neural network for leaves recognition using data augmentation," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.*, Oct. 2015, pp. 2143–2150, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.318.

[54] Z. Wang, X. Sun, Y. Zhang, Z. Ying, and Y. Ma, "Leaf recognition based on PCNN," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 899–908, May 2016, doi: 10.1007/s00521-015-1904-1.

XIAOBO YANG received the B.Sc. degree from the Central South University of Forestry and Technology, in 2017, where he is currently pursuing the M.Sc. degree. His main research interests include image processing and deep learning.

AIBIN CHEN received the B.Sc. degree from Lanzhou University, in 1994, and the M.Sc. and Ph.D. degrees from Central South University, China, in 2004 and 2010, respectively. He is currently a Professor with the Central South University of Forestry and Technology. His main research interests include artificial intelligence and forest information engineering.

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.

JIANWU WANG was born in 1970. He graduated from the Hunan Forestry College, in 1993. He is currently with the Huangfengqiao State-Owned Forest Farm, Youxian, Hunan, specializing in forestry management and forestry planning management.

WENJIE CHEN received the B.Sc. degree majored in the Internet of Things (IoT) from Nanchang Hangkong University, in 2018. She is currently pursuing the degree with a major of computer technology with the Computer and Information School, Central South University of Forestry and Technology. She has been studying in the application of the Artificial Intelligence Laboratory. Her main research interests include graphics and image processing.

YUAN GAO received the B.Sc. degree from the Central South University of Forestry and Technology, in 2017, where he is currently pursuing the M.Sc. degree. His main research interests include image processing and deep learning.

RUNDONG JIANG received the B.Sc. degree from Huaihua University, in 2017. He is currently a 2017 Graduate Student majoring in computer technology with the Central South University of Forestry and Technology. His main research interests include graphics and image processing.