IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 11, 2019, accepted October 4, 2019, date of publication October 15, 2019, date of current version October 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947510

Plant Leaves Classification: A Few-Shot Learning Method Based on Siamese Network

BIN WANG AND DIAN WANG
School of Technology, Beijing Forestry University, Beijing 100089, China
Corresponding author: Dian Wang (wangdian@ bjfu.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2015ZCQ-GX-01.

ABSTRACT In recent years, the method of plant leaf classification by deep learning has gradually become mature. However, training a leaf classifier based on deep learning requires a large number of samples for supervised training. In this paper, a few-shot learning method based on the Siamese network framework is proposed to solve a leaf classification problem with a small sample size. First, the features of two different images are extracted by a parallel two-way convolutional neural network with weight sharing. Then, the network uses a loss function to learn the metric space, in which similar leaf samples are close to each other and different leaf samples are far away from each other. In addition, a spatial structure optimizer (SSO) method for constructing the metric space is proposed, which will help to improve the accuracy of leaf classification. Finally, a k-nearest neighbor (kNN) classifier is used to classify leaves in the learned metric space. The average classification accuracy is used as a performance measure. The open access Flavia, Swedish and Leafsnap datasets are used to evaluate the performance of the method. The experimental results show that the proposed method can achieve a high classification accuracy with a small size of supervised samples.

INDEX TERMS Leaf classification, few-shot learning, convolutional neural network, Siamese network.

I. INTRODUCTION

Plants are widely distributed in the natural environment, participate in the material cycle of the ecosystem, and play an important role in protecting the earth's ecosystem. At present, the global climate is gradually changing, the natural environment is being destroyed by human beings and the continuous expansion of human cities, resulting in a sharp decline in plant species and numbers [1]. Therefore, it is particularly important to protect the biodiversity of plants. Unfortunately, the conservation of plant species requires the ability to artificially classify their species, a skill that comes from intensive learning and experience [2]. It is almost impossible for ordinary people to identify traditional plant species, and even for practitioners who come into contact with plants every day, such as horticulturists, farmers and landscape architects have difficulty classifying plant species. This issue is called the taxonomic crisis in the field of related research [3]. Therefore, botanists believe that technological of plant image retrieval can greatly reduce the gap in plant classification skills of researchers from different fields [4].

In past studies, features for plant identification were usually selected from plant organs such as leaves, flowers, fruits, and stems, among which the leaves of plants are the most representative and easiest to obtain. Generally, most plants can be well identified by processing the image of plant leaves with a semisupervised method [5] in machine learning, but this method is time-consuming and laborious, which makes it difficult to popularize and apply further. At present, with the development of image technology, using computer technology to automatically classify leaf characters after feature extraction has become the mainstream method [6]. In the research of related fields, the characteristics of leaves are usually extracted by hand, For example, leaves are often classified by using the shape differences between different leaves [7], [8], Leaf edges, as an important feature, are often used as extraction targets [9], In addition, using the technology of leaf vein texture feature detection [10], singular value decomposition (SVD) and sparse representation (SR) are combined to process dimensionally reduced plant images [11], the moment invariant method for multicomponent shapes [12] and artificial neural network with support vector machine [13] have also been successful to some extent.

However, it should be noted that the above methods rely on feature selection and manual processing, while the

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

processing of advanced features and multiscale features will significantly increase the complexity and workload, which leads to the problem of reduced generalizability of these methods after increasing the number and types of identified plants. Recently, due to the excellent performance of deep learning convolutional neural networks in the field of computer vision, they has become the main means to solve the problems of image classification, image recognition and semantic segmentation [14]–[16]. The application of deep learning methods in plant classification has achieved a good performance, and their comprehensive performance is better than that of most manual feature extraction classification methods, especially their excellent generalization performance [17]–[19].

However, the disadvantages of deep learning are also obvious. The premise of high classification accuracy is that the network has sufficient supervised learning samples, which is usually very difficult. In most cases, we can only obtain a small number of learning samples, and general deep learning neural networks perform extremely poorly when encountering a small number of learning samples. Therefore, the concept of few-shot learning is proposed. It aims to learn the classification methods of these samples from a small number of supervised samples. Faced with the same problem, humans can quickly and accurately master classification methods with few samples. When new samples are introduced, humans can make accurate judgments by comparing them through measurements. We expect to make them master a measurement method through the training network so that they can learn from small samples and apply it to the automatic classification of leaves. Therefore, inspired by the small-sample learning method and the metric space [20] in the prototype network, we constructed a structure based on a Siamese network [21]–[23] to extract the characteristics of plant leaves and classify them.

The main contributions of this paper are as follows.

1) A method based on the Siamese network structure is proposed to construct a metric space for leaf classification, where similar samples are close to each other and dissimilar samples are far away from each other.
2) A spatial structure optimizer is proposed to improve the speed and performance of measuring the spatial formation process.
3) Experimental verification with the Flavia [24], Swedish [25] and Leafsnap [26] datasets shows that this method can effectively classify leaves with a small number of supervised samples.

The rest of this article is structured as follows. The overall structure and algorithm of the method are presented in section II. Section III shows the experimental method and the principle of spatial structure optimization. The validation results of this method in related data sets are presented in section IV. Finally, we conclude and summarize the results in section V.

TABLE 1. Performance measures comparison within the same training environment.

Method	Top-1 Accuracy	FLOPs	Params
VGG-16	84.47%	1.56X10^{10}	138.38M
Inception-v4	94.11%	5.22X10^{9}	14.62M
ResNet-50	92.24%	4.12X10^{9}	25.60M
ResNeXt-50	93.49%	4.25X10^{9}	25.00M
SENet	92.55%	4.19X10^{9}	28.70M
DenseNet-121	91.63%	2.83X10^{9}	7.89M
DenseNet-201	93.83%	3.79X10^{9}	12.79M

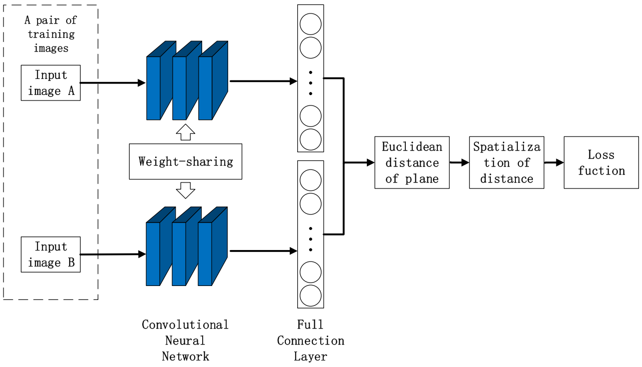
A pair of training images
Input image A
Weight-sharing
Input image B
Convolutional Neural Network
Full Connection Layer
Euclidean distance of plane
Spatialization of distance
Loss fuction

FIGURE 1. Structure flow chart of training feature extraction.

II. PROPOSED CNN STRUCTURE

A. INITIAL CNN ANALYSIS

Generally, classification accuracy is an important performance index for evaluating convolutional neural networks. However, in practical applications, we need to consider the complexity of the model and the number of calculations, which are measured by floating-point operations (FLOPs) and parameters (params), respectively. For example, for ResNet and other networks, as the number of layers increases, the classification accuracy will increase, but at the same time, the complexity and computational complexity of the model will also increase. Too many parameters will lead to too many computations of the metric function, making it difficult to form the metric space.

When the output parameters of the model are close to each other, reasonable layers are set for different CNNs, and experiments were carried out on the Flavia datasets. The results are shown in Table 1. According to the experimental results, this paper proposes using the inception-v4 structure. Although a deep DenseNet has better performance in model complexity, DenseNet consumes much memory, which is not conducive to the implementation of the project.

B. EXPERIMENTAL STRUCTURE AND ALGORITHMS

The training structure for each batch is shown in Fig. 1. Inspired by the Siamese network structure, this paper proposes a structure combining multilayer convolutional neural networks and few-shot learning methods to classify plant leaf

TABLE 2. Algorithm for the proposed work.

```
Algorithm Iterative Training Method. $N_p$ denotes the total number of positive samples, $N_q$ denotes the total number of negative samples, RandomTrainingSet(Q) denotes randomly choosing Q elements from set $N_p$ or $N_q$, $P_i$ denote the sample label, S denote the total number of training steps.
Input: Training data set $T = \{(x_1,y_1, P_1),\ldots,(x_n,y_n, P_n)\}$, $n \le N_p + N_q$
Output: The Euclidean distance and loss for a batch-size training
  $R(x_i,y_i,P_i) \leftarrow$ RandomTrainingSet $Q(x_i,y_i,P_i)$, $Q \in T$
  for i in $\{1, \ldots, n\}$ do
    $R_1 \leftarrow x_i$,
    $R_2 \leftarrow y_i$,
    $R_p \leftarrow P_i$
  end for
 $D = f(R_1, R_2)$                                   ►Computing the Euclidean distance
  $L_0 \leftarrow 0$
  for k in $\{1, \ldots, S\}$ do
    $L_k \leftarrow L_{k-1} + [-(R_p \times \mathrm{Log}(D) + (1 - R_p) \times \mathrm{Log}(1 - D)]$     ►Update loss
  end for
```

TABLE 3. Structure of the training dataset.

Class	Number of Species	Number of Images	Total Number of Images
Training Dataset A	10	20	200
Training Dataset B	10	15	150
Training Dataset C	10	10	100
Training Dataset D	10	5	50

species. In the design of feature extractors, we refer to and improve the Inception structure to increase the adaptability of the network to the input image scale, and reduce the phenomena of gradient disappearance and overfitting, thus reducing the adverse impact caused by the sample itself. The last part of the structure uses the logistic regression loss function to measure the similarity between input image pairs. Table 2 shows the proposed steps of the algorithm.

III. MATERIALS AND METHODS

A. DATASET

In the proposed work, it is necessary to select the appropriate dataset in the training and evaluation stages to evaluate the performance of the algorithm. We choose the Flavia, Swedish and Leafsnap datasets for the training and test sets. Since the main purpose of this paper is to solve the problem of leaf classification in the case of small samples, the numbers of training images for each supervised sample are different (5-20, and the increment is 5). All other images that are not selected as monitoring samples will constitute a verification set to evaluate the algorithm. It is worth noting that a single training sample consists of two pictures. If they belong to the same category, it is called a positive sample, and the sample is labeled with a 1; if they belong to different categories, it is called a negative sample, and the sample is labeled with a 0. Fig. 2 shows some samples of the Flavia, Swedish and Leafsnap datasets. Table 3 shows the composition of the four subsets required for each dataset experiment.

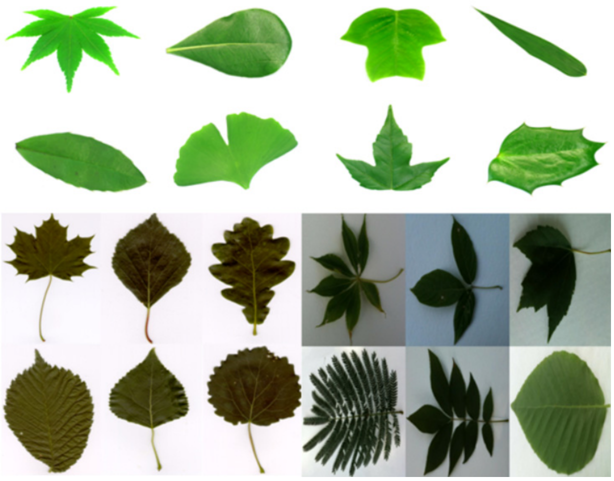

FIGURE 2. Samples of images.

B. PREPROCESSING OF IMAGES

To improve the performance of the convolutional neural network in image feature extraction, image preprocessing is required before training, and it is necessary to reconstruct the image size. In this study, the sizes of all the images in the dataset were uniformly adjusted to 112×112 pixels by central clipping, which was automatically completed by the computer through the OpenCV framework and Python script. Eq. (1) gives the center square clipping method for image scaling.

$$def\ SquareResize(Img, New\_height, New\_width) \tag{1}$$

C. FEW-SHOT LEARNING AND DISTANCE TRAINING STRATEGY

Generally, when the number of features is insufficient, the use of neural network classifiers for optimization will lead to serious overfitting because the neural network classifier has a large number of parameters to be optimized. It is necessary to construct a nonparametric optimization method and construct a classifier suitable for few-shot learning under the framework of meta-learning. The Siamese network structure can map the similarity relationship between different images into a metric space so that the samples belonging to the same category can be as close as possible, and the samples belonging to different categories can be as far away as possible. The method used in this paper is trained in a supervised way, and the samples are extracted by a two-way convolution neural network. Then, the Euclidean distance between features is calculated by a metric-based method: the closer the distance is, the more similar the samples are. However, errors may occur in the formation of the measurement space, as shown in the large fluctuation in the loss function and the slow convergence rate. The reason is that there are several similar types of leaves in the training samples, which makes it difficult to form a relatively stable measurement space. For example, three similar samples will be in the metric space mapping plane and form a stable distribution of an equilateral triangle,

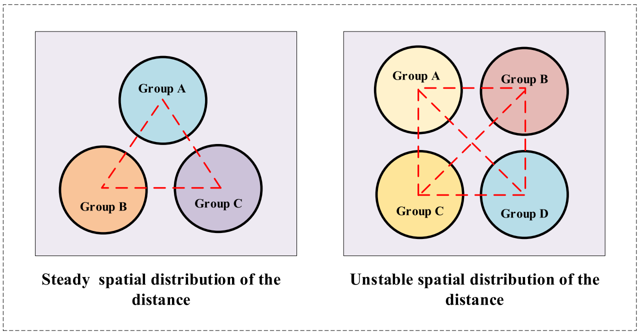
Group A
Group B
Group C
Steady spatial distribution of the distance
Group A
Group B
Group C
Group D
Unstable spatial distribution of the distance

FIGURE 3. Reasons for plane errors.

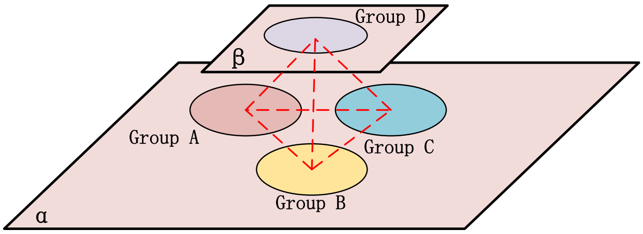

FIGURE 4. Principle of the tetrahedron structure.

but the four similar samples will endanger the stability of the original mapping plane, causing its distribution to be a square, which can force the distances between samples in the diagonal positions to increase, as shown in Fig. 3 (the red dotted line represents the average distance between the sample classes). This situation will cause the oscillation of the loss function in the training process, which will slow down the convergence process and eventually lead to a decrease in precision.

Therefore, we propose the SSO that acts on the process of metric space formation. The spatial distribution of the distance is achieved by using the stability of the spatial structure of a regular tetrahedron to accelerate the training convergence speed and improve the accuracy, as shown in Fig. 4. The β plane is independent of the α plane. Under normal circumstances, the metric space plane mapping of the distance is distributed in the α plane. When sample D meets the SSO condition, the distance of sample D will be mapped into the β plane. In the subsequent training, only the distances between samples A, B and C related to it will be trained. In the k-nearest neighbor classifier, the distance between sample D and other unassociated samples E is replaced by formula eq. (2).

$$distance(DE) = mean(distance(AE) + distance(BE) + distance(CE)) \quad (2)$$

It should be noted that when multiple SSO conditions are triggered, the distance is not calculated between all samples distributed in the beta plane during training.

The trigger condition of the SSO structure is calculated by eq. (3). The formula only works for four samples satisfying the conditions: SSO does not trigger when there are more than four different samples, this is because, in the early stage of network training, a large number of samples meets the requirements of sufficient samples. Note that to distinguish samples in different planes, it is necessary to mark the mapping plane of the samples in the training process, which is expressed as eq. (4) in the program.

$$\frac{\sum_{i=0}^{k} f_i(d,a)}{k} \approx \frac{\sum_{i=0}^{k} f_i(d,b)}{k} \approx \frac{\sum_{i=0}^{k} f_i(d,c)}{k} \ll P \quad (3)$$

where k denotes the minimum number of sample distances calculations to be met, f(d, a), f(d, b), and f(d, c) are the Euclidean distance functions between samples, and P is the Euclidean distance value that satisfies the trigger condition

$$f_y = (Distance, n) \quad (4)$$

where *Distance* is the Euclidean distance function of the samples, and n is the distribution plane.

With the Euclidean distance, stochastic gradient descent training can be used to motivate the loss function. The logistic regression loss function does not have a perfect predictive functional performance, but here, it is very good for generating a metric space, making similar samples, close to the same sample. The loss function is shown in eq. (5).

$$L = -[f \times \log(f_y) + (1-y) \times \log(1-f_y)] \quad (5)$$

where L is the loss function and f is the label of the input pair., if the input images are from the same class, $f = 1$,; otherwise, $f = 0$,; f_y is the European distance for the training pair.

D. CONVOLUTIONAL NEURAL NETWORKS

After AlexNet [27], proposed by Alex Krizhevsky et al, won the ImageNet competition in 2012 by an absolute majority, deep learning models have gradually become a major force in various fields such as computer vision and natural language processing. Moreover, with the rapid development of high-performance computing devices dominated by graphics processing units (GPUs), deep learning networks with deeper layers and wider widths continue to emerge. Active in the field of computer vision, convolutional neural networks play an important role in target detection, image classification, image segmentation, and other fields. Convolutional neural networks can extract target features on a large scale and conduct complex calculations.

Generally, this architecture is composed of five parts: the input layer, convolution layer, pooling layer, full connection layer, and output layer. In some systems, the convolution layer is used to replace the pooling layer. Through the flexible design of the convolutional layer and pooling layer structure, modification of the linear and nonlinear activation functions, the addition of auxiliary structures or parameters, etc., the convolutional neural network models such as GoogLeNet, ResNet, and DenseNet, are formed. Fig. 5 shows our proposed CNN model for extracting and processing leaf features. This model is inspired by the structure of GoogLeNet.

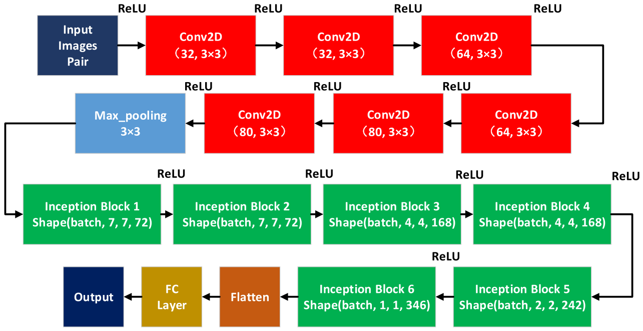
Input Images Pair
ReLU
Conv2D (32, 3×3)
ReLU
Conv2D (32, 3×3)
ReLU
Conv2D (64, 3×3)
ReLU
Conv2D (64, 3×3)
ReLU
Conv2D (80, 3×3)
ReLU
Conv2D (80, 3×3)
ReLU
Max_pooling 3×3
Inception Block 1 Shape(batch, 7, 7, 72)
ReLU
Inception Block 2 Shape(batch, 7, 7, 72)
ReLU
Inception Block 3 Shape(batch, 4, 4, 168)
ReLU
Inception Block 4 Shape(batch, 4, 4, 168)
ReLU
Inception Block 5 Shape(batch, 2, 2, 242)
ReLU
Inception Block 6 Shape(batch, 1, 1, 346)
Flatten
FC Layer
Output

FIGURE 5. The architecture of the proposed CNN.

GoogLeNet can allocate computing resources better and extract more features than other models under the same computation amount, and GoogLeNet can solve the gradient disappearance, gradient explosion and other problems caused by the ultradeep network. GoogLeNet introduced the concept of the "Inception module", the idea of which is to use relatively dense components to approximate the optimal local sparse structure. In this paper, the structure consists of six convolution layers and one pooling layer, followed by six "Inception modules". Finally, the characteristic parameters are transferred through a fully connected layer. A random gradient descent method was used to train the CNN. A ReLU nonlinear activation function is added in each layer to reduce the probability of gradient disappearance and improve the speed of the backpropagation calculation. Moreover, the ReLU function increases the sparsity of the model. The maximum pooling layer and the average pooling layer are introduced to reduce the number of calculations and optimize the calculation space. The maximum pooling layer can reduce the error of the estimated mean deviation caused by the parameter error of the convolution layer and retain more texture features. The average pooling layer can reduce the error caused by the increase in the estimated variance caused by the limitation of the neighborhood size and retain more background features. The "Inception module" uses an asymmetric convolution kernel to replace the conventional convolution kernel to reduce the computational burden. Details of the proposed convolutional neural network are as follows:

1) This model is a one-way transmission, the input layer is responsible for extracting the training images of a special standard size (112 × 112 in this paper), and the function of each layer is to extract more image features from the previous layer and pass them to the next layer for processing.
2) The 1st layer reduces the images and increases the number of features through 32 convolution kernels of size 3 × 3 and sets the step size. The no-padding filling method is adopted to reduce the edge feature quantity.
3) Compared with the previous layer, the step size is set to 1 to further reduce the edge feature quantity.
4) The number of convolution kernels in the 3 layers is increased to 64; starting with this layer, the padding filling method is selected to better preserve the edge features
5) Layers 4 to 6 are adjusted by the above method.
6) The 7th layer is the maximum pooling layer with a step size of 2, and the size will be reduced to half of the original one when the number of feature graphs remains unchanged.
7) The "Inception module" of layers 8 and 9 have the same structure, with 4 parallel convolution channels and feature combinations at the end of the module, among them, channel 2 contains a convolution kernel of size 5 × 5, channel 3 contains two convolution kernels of size 3 × 3, and channel 4 contains an average pooling layer.
8) The third "Inception module" reduces a parallel convolution channel. Channel 3 selects the maximum pooling layer.
9) The fourth "Inception module" readopts the 4-channel pattern, where the 2nd and 3rd channels start to use an asymmetric convolution kernel, and the 4th channel uses an average pooling layer.
10) The last two "Inception modules" restore the 3 channels: channel 2 uses a symmetric convolution kernel, and channel 3 uses an averaging pooling.
11) The last layer is the fully connected layer and involves weight sharing, which is responsible for integrating all the extracted characteristics and passing them onto other structures for processing.

E. NEAREST NEIGHBOR CLASSIFICATION

Before making predictions, we first need to build a supervised sample. It has been suggested that 50% of the training samples be randomly selected as supervised samples when the number of training samples for each type of image is greater than 10 and label them according to the type of samples. As shown in Fig. 7, during the test phase, the tested sample and the supervised sample are extracted by the convolutional neural network.

Then, the Euclidean distance between them is calculated. After training, the network will keep similar samples close to each other and dissimilar samples far away from each other. The correlation method [28] proves that the kNN classifier is essential when there are few supervised samples. Therefore, through a simple kNN classifier, the classification task can be completed by comparing and analyzing the Euclidean distance between the samples to be tested and different kinds of supervised samples, as shown in Fig. 8.

F. OVERALL EXPERIMENTAL PROCEDURE

The overall experimental procedure is shown in Fig. 6. First, combined with the training image pairs constructed in the data set, the constructed two-way convolution neural network is used for feature extraction, and then the contrast loss function is used to form the metric space with the SSO. When the image is input, the network calculates the Euclidean distance between the sample to be tested and the known species in

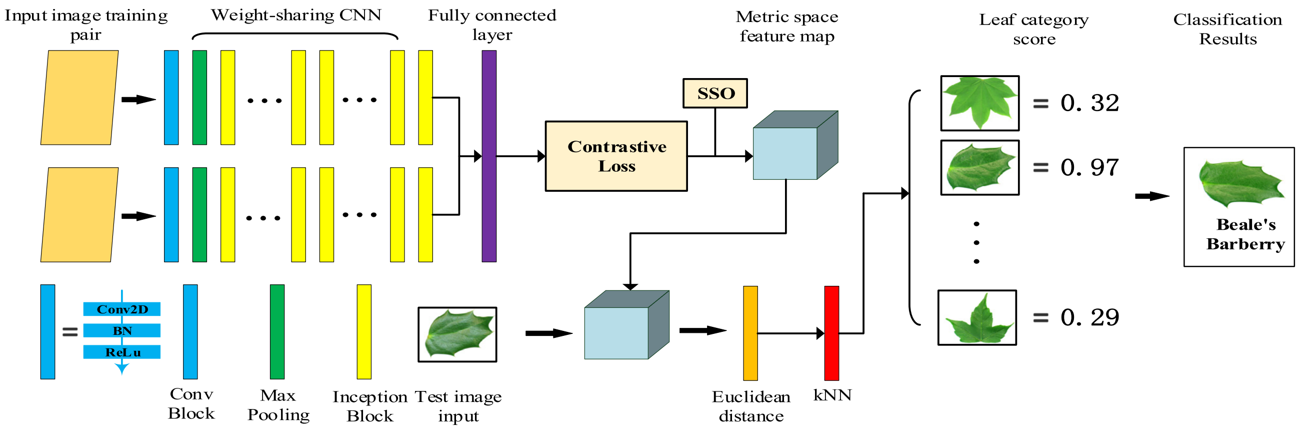

FIGURE 6. Overall experimental procedure based on a Siamese network framework.

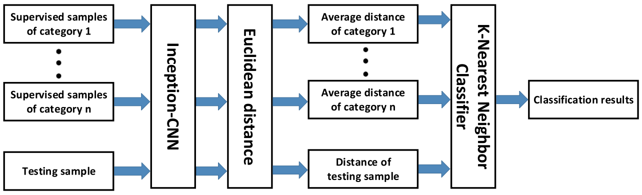
Supervised samples of category 1
Supervised samples of category n
Testing sample
Inception-CNN
Euclidean distance
Average distance of category 1
Average distance of category n
Distance of testing sample
K-Nearest Neighbor Classifier
Classification results

FIGURE 7. Flowchart of leaf classification for the test dataset.

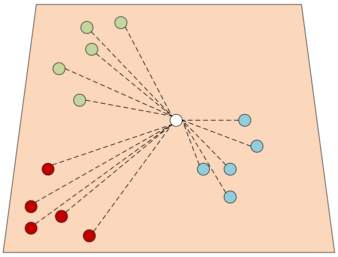

FIGURE 8. KNN classification of the test samples(five labeled samples per class, dotted lines represent European distances).

the metric space, and outputs the similarity score through the kNN classifier: the higher the similarity between samples, the higher the score is. The highest score category is the prediction result.

IV. EXPERIMENTS AND RESULTS

All the experiments were conducted on a laptop with an Intel Core i7-6700HQ processor (2.6 GHz) and an Nvidia Geforce GTX 1060 6 GB graphics card. The laptop has 16 GB of memory. The training and testing work was implemented using the open-source software framework TensorFlow. The recommended parameters for the CNN were set as follows: the learning rate was set to 0.001, the dropout rate was set to 0.5, the training step length was set to 30000, and the batch size was set to 8.

A. EXPERIMENTAL DATASET AND SSO VERIFICATION

Initially, four subsets of the training sets were generated from the three datasets. According to the Siamese network structure, the input image training pair is constructed. Training samples from the same type constitute positive training samples, while training samples from different types constitute negative training samples. It should be noted that the number of negative training samples is larger than that of the positive ones, so we need to randomly remove some negative training samples. Table 4 shows the number of positive samples and negative samples in the four subsets of the Flavia, Swedish and Leafsnap datasets.

In addition, to prove the positive effect of the SSO, we set up two special training sets, which are composed of samples satisfying the trigger condition of the SSO by observing several training results of subset C (n=10) of the Flavia training dataset. Fig. 9 shows four leaves that meet the requirements of a particular training set: in some ways, it is difficult for nonprofessionals to classify these leaves accurately.

Fig. 10 shows the variation in the loss curve with the SSO and without the SSO. The average value of the 100 steps of loss training is calculated, and the b-spline curve is used for drawing. As shown in the figure, the SSO loss curve converges faster and the descent process is smoother before the 20000 steps. In addition, at the end of training, the stability

TABLE 4. The number of positive samples and negative samples.

Class	Number of positive samples	Number of negative samples
Training Dataset A	4000	12000
Training Dataset B	2250	6750
Training Dataset C	1000	3000
Training Dataset D	250	750

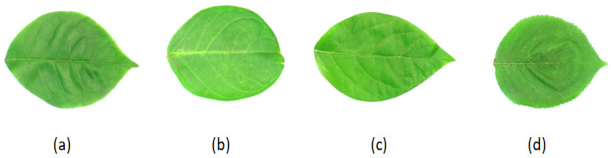

FIGURE 9. Images of four plant samples. (a) Big-fruited holly. (b) Crepe myrtle. (c) Wintersweet. (d) Japanese flowering cherry.

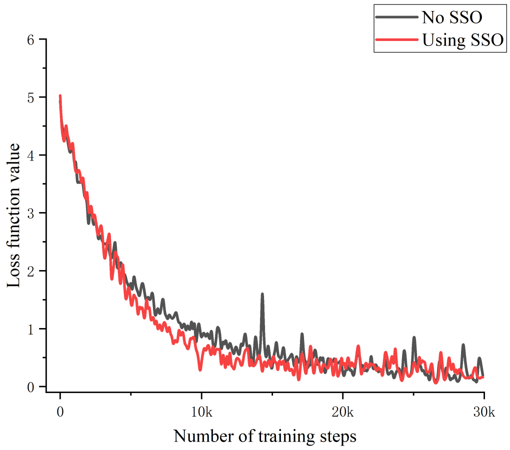

FIGURE 10. **Curves of the loss function from the same test dataset.**

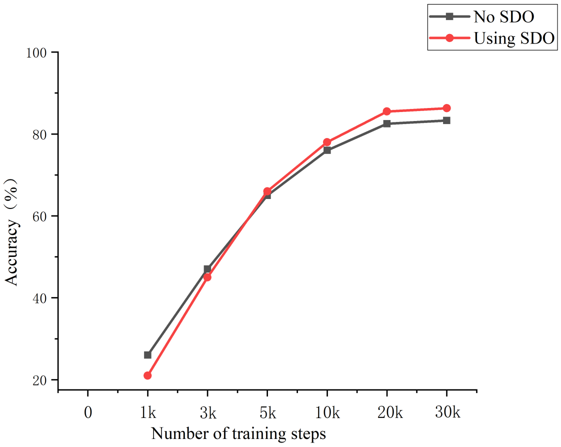

FIGURE 11. **Overall accuracy (%) from the same test dataset.**

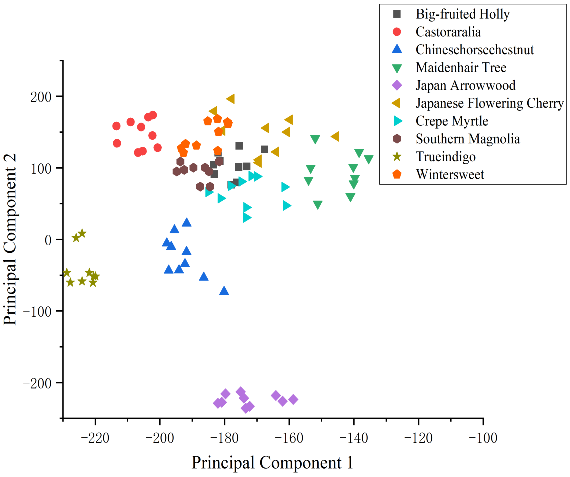

FIGURE 12. **Metric spaces formed without SSO on special training datasets.**

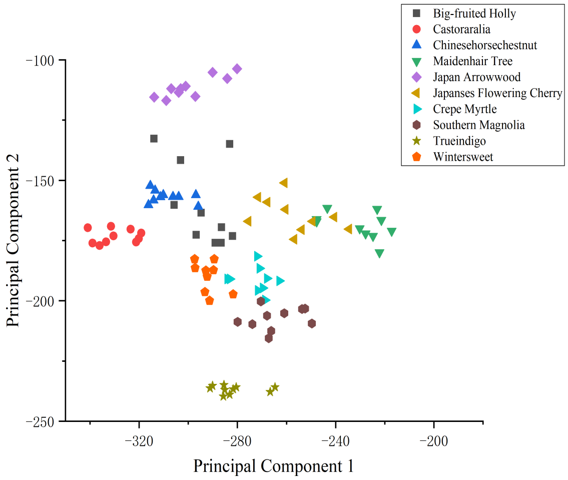

FIGURE 13. **Metric spaces formed by the SSO on special training datasets.**

of the SSO loss curve is better. Fig. 11 shows the classification accuracy results of the kNN classifier on the same test dataset with and without the SSO. According to the curve analysis, as the number of training steps increases, the network advantages of SSO training gradually emerge, and a high classification accuracy is maintained in the later stages.

To more intuitively compare the difference between the measurement space formed by using the SSO and by not using the SSO, we extracted the output layer of the CNN, the last layer of the special training set, and used PCA method for visualization. Fig. 12 is a metric space without the SSO. It can be seen from the graph that the leaves are densely distributed around big-fruited holly, and the distribution distances between some species with higher similarity are close. Even the phenomenon of staggering distribution exists, which leads to a low fault tolerance rate in the process of kNN classification, which leads to a decline in the accuracy. Fig. 13 is a metric space formed using the SSO. Although big-fruited holly's distribution is scattered, it is well separated from the similar Japanese flowering cherry, crepe myrtle and wintersweet samples. After ignoring big-fruited holly, the distribution of the metric space tends to be more reasonable, the distributions of the same kind of leaves is more concentrated, and the fault-tolerance rate is higher in the process of kNN classification, so the classification accuracy is higher.

B. COMPARISON WITH OTHER CNN FRAMEWORKS

In this section, the performance of Siamese network with other convolutive neural network frameworks, such as VGG, ResNet, ResNeXt, SENet and DenseNet, is tested. Note that for a fair comparison, it is necessary to uniformly input images of the same size and to fine-tune these networks, especially the number of layers in the network. Table 5, Table 6 and Table 7 lists the test results. Compared with other CNN frameworks, the adjusted Siamese + Inception (S-Inception) network can provide competitive results, and the results show that the S-Inception combination can achieve good accuracy when the training sample settings are appropriate. This finding is attributed to the advantages

TABLE 5. **Overall accuracy (%) of the different CNN methods for the Flavia dataset.**

Method	n = 5	n = 10	n = 15	n = 20
S-VGG	33.75	52.60	71.33	82.20
S-Inception	59.20	**85.24**	**92.34**	**95.32**
S-ResNet	57.41	81.18	89.69	93.84
S-ResNeXt	58.10	82.35	90.27	94.24
S-SENet	57.62	81.85	90.22	94.01
S-DenseNet	**60.73**	83.91	92.32	94.78

n is the number of supervised samples per category in the training set; bold values represent the best accuracy among these methods in each case.

TABLE 6. **Overall accuracy (%) of the different CNN methods for the Swedish dataset.**

Method	n = 5	n = 10	n = 15	n = 20
S-VGG	29.20	47.85	66.97	74.33
S-Inception	52.75	77.24	**88.15**	**91.67**
S-ResNet	49.55	72.47	85.05	88.76
S-ResNeXt	49.50	73.96	86.24	90.00
S-SENet	50.45	73.65	85.94	89.60
S-DenseNet	**54.72**	**78.10**	86.71	90.44

n is the number of supervised samples per category in the training set; bold values represent the best accuracy among these methods in each case.

TABLE 7. **Overall accuracy (%) of the different CNN methods for the Leafsnap dataset.**

Method	n = 5	n = 10	n = 15	n = 20
S-VGG	38.30	49.47	67.55	78.86
S-Inception	**60.95**	84.37	**92.55**	**95.75**
S-ResNet	59.41	79.98	90.03	93.21
S-ResNeXt	59.50	80.75	91.70	94.67
S-SENet	58.65	80.35	90.57	94.35
S-DenseNet	58.85	**84.41**	92.50	95.10

n is the number of supervised samples per category in the training set; bold values represent the best accuracy among these methods in each case.

of the Inception structure. When the number of samples is insufficient, the deep network gradient disappears seriously, and serious overfitting will occur. The Inception structure increases the utilization of parameters and uses smaller convolution blocks instead of larger ones, thus increasing the nonlinear expression ability of the model and effectively alleviating these two phenomena. It is noteworthy that in these test datasets, when the number of training samples is small (n = 5,10), S-DenseNet achieves the highest accuracy and performs better in other monitoring samples, as determined by DenseNet's structural characteristics. When there are fewer samples, the dense connection can preserve the feature maps of each layer well and reduce overfitting, which is similar to the auxiliary classifiers in Inception structure. The S-DenseNet combination after precise optimization may exceed the S-Inception combination. Although it performs well, the extremely high operation cost and poor operation efficiency are not suitable for few-shot learning. Overall, the test results show that it is good to combine the Siamese network structure with different CNN frameworks to realize the few-shot learning classification method of leaves, which further proves the effectiveness of the method. The accuracy is computed by eq. (6).

TABLE 8. **Overall accuracy (%) of the different semisupervised methods for the Flavia dataset.**

Method	n = 5	n = 10	n = 15	n = 20
S-Inception	59.20	85.24	92.34	95.32
SSLDP	32.20	44.14	58.72	74.61
SFFD	42.81	77.75	83.21	85.77
SS-HCNN	41.83	69.10	87.12	93.50

n is the number of supervised samples per category in the training set.

TABLE 9. **Overall accuracy (%) of the different semisupervised methods for the Swedish dataset.**

Method	n = 5	n = 10	n = 15	n = 20
S-Inception	52.75	77.24	88.15	91.67
SSLDP	31.74	42.50	55.67	73.79
SFFD	39.62	73.55	80.85	83.12
SS-HCNN	38.54	66.47	84.95	92.05

n is the number of supervised samples per category in the training set.

$$\text{Accuracy} = \frac{\text{number of correctly classified samples}}{\text{total number of samples}} \times 100\% \tag{6}$$

C. COMPARISON WITH SEMISUPERVISED METHODS

In this section, the proposed method is compared with several semisupervised methods, including SSLDP [29], SFFD [30], SS-HCNN [31], and the unified use of the kNN classifier. Additionally, L is set to a maximum of 6 in SSLDP for maximum performance. Table 8, Table 9 and Table 10 show the results of the comparative tests. As we can see from the table, all the methods improve the accuracy when the number of supervised samples increases, but the S-Inception method improves faster, because the generalization ability of the S-Inception structure is better, and the width of the structure makes it possible to extract more features when the number of samples increases. The proposed method performs well when the supervisory sample is small (n=5). When the number of training samples increases, the accuracy of SS-HCNN improves rapidly. This is because when the total number of training samples is small, there is a serious overfitting phenomenon. The increase in the training samples will be

TABLE 10. Overall accuracy (%) of the different semisupervised methods for the Leafsnap dataset.

Method	n = 5	n = 10	n = 15	n = 20
S-Inception	60.95	84.37	92.55	95.75
SSLDP	31.74	43.64	61.87	77.45
SFFD	43.00	79.97	82.59	86.81
SS-HCNN	47.74	73.80	88.88	94.05

n is the number of supervised samples per category in the training set.

alleviated. At the same time, the multiscale characteristics of SS-HCNN increase the utilization of features, so the accuracy improves. The experimental results show that this method performs better than some semisupervised methods.

V. CONCLUSION AND FUTURE WORK

In this paper, an improved convolutional neural network structure is proposed to solve the problem of leaf classification in the case of small samples, which is of great significance for solving the problem of sparse samples or various types of classification tasks. The key to this method is to extract image features by using a convolutional neural network and construct the metric space by using the concept of similarity between different image features. The quality of the metric space and the selection of the supervised samples determine the classification accuracy of the nearest neighbor classifier. Moreover, through the loss function curve and accuracy, it is proven that a structure optimization device that acts on the metric space is effective. The experimental results also show that when the number of training samples is 20, the classification accuracy of this method is the highest, 95.32%, 91.37% and 91.75% accuracies are obtained from the Flavia, Swedish and Leafsnap datasets, respectively. These results are competitive in the deep learning classification field.

Although this method has achieved good results on Flavia, Swedish and Leafsnap datasets, further research is needed to improve the generalization ability of the model. In the future, our method will be applied to more datasets and few-shot classification tasks.

REFERENCES

[1] S. L. Pimm, C. N. Jenkins, R. Abell, T. M. Brooks, J. L. Gittleman, L. N. Joppa, P. H. Raven, C. M. Roberts, and J. O. Sexton, "The biodiversity of species and their rates of extinction, distribution, and protection," *Science*, vol. 344, no. 6187, May 2014, Art. no. 7246752. doi: 10.1126/science.1246752.

[2] J. Wäldchen, M. Rzanny, M. Seeland, and P. Mäder, "Automated plant species identification—Trends and future directions," *PLoS Comput. Biol.*, vol. 14, no. 4, Apr. 2018, Art. no. e1005993. doi: 10.1371/journal.pcbi.1005993.

[3] B. Dayrat, "Towards integrative taxonomy," *Biol. J. Linnean Soc.*, vol. 85, no. 85, pp. 407–415, Jul. 2005. doi: 10.1111/j.1095-8312.2005.00503.x.

[4] Y. Sun, Y. Liu, G. Wang, and H. Zhang, "Deep learning for plant identification in natural environment," *Comput. Intell. Neurosci.*, vol. 2017, May 2017, Art. no. 7361042. doi: 10.1155/2017/7361042.

[5] V. Narayan and G. Subbarayan, "An optimal feature subset selection using GA for leaf classification," *Int. Arab J. Inf. Technol.*, vol. 11, no. 5, pp. 447–451, Sep. 2014. doi: 10.1109/TIT.2014.2344251.

[6] H. Qi, T. Shuo, and S. Jin, "Leaf characteristics-based computer-aided plant identification model," *J. Zhejiang Forestry College*, vol. 20, no. 3, pp. 281–284, 2003. doi: 10.1023/A:1022289509702.

[7] C. Zhao, S. S. F. Chan, W.-K. Cham, and L. M. Chu, "Plant identification using leaf shapes—A pattern counting approach," *Pattern Recognit.*, vol. 48, no. 10, pp. 3203–3215, Oct. 2015. doi: 10.1016/j.patcog.2015.04.004.

[8] A. Aakif and M. F. Khan, "Automatic classification of plants based on their leaves," *Biosyst. Eng.*, vol. 139, pp. 66–75, Nov. 2015. doi: 10.1016/j.biosystemseng.2015.08.003.

[9] J. R. Kala and S. Viriri, "Plant specie classification using sinuosity coefficients of leaves," *Image Anal. Stereol.*, vol. 37, no. 2, pp. 119–126, 2018. doi: 10.5566/ias.1821.

[10] H. Kolivand, B. M. Fern, T. Saba, M. S. M. Rahim, and A. Rehman, "A new leaf venation detection technique for plant species classification," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3315–3327, Apr. 2019. doi: 10.1007/s13369-018-3504-8.

[11] S. Zhang, C. Zhang, Z. Wang, and W. Kong, "Combining sparse representation and singular value decomposition for plant recognition," *Appl. Soft Comput.*, vol. 67, pp. 164–171, Jun. 2018. doi: 10.1016/j.asoc.2018.02.052.

[12] M. B. H. Rhouma, J. Žunić, and M. C. Younis, "Moment invariants for multi-component shapes with applications to leaf classification," *Comput. Electron. Agricult.*, vol. 142, pp. 326–337, Nov. 2017. doi: 10.1016/j.compag.2017.08.029.

[13] S. J. Kho, S. Manickam, S. Malek, M. Mosleh, and S. K. Dhillon, "Automated plant identification using artificial neural network and support vector machine," *Frontiers Life Sci.*, vol. 10, no. 1, pp. 98–107, 2017. doi: 10.1080/21553769.2017.1412361.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. doi: 10.1038/nature14539.

[15] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016. doi: 10.1109/TMI.2016.2528162

[16] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019. doi: 10.1109/ACCESS.2019.2907383.

[17] J. Hu, Z. Chen, M. Yang, R. Zhang, and Y. Cui, "A multiscale fusion convolutional neural network for plant leaf recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 853–857, Jun. 2018. doi: 10.1109/LSP.2018.2809688.

[18] S. A. Pearline, V. S. Kumar, and S. Harini, "A study on plant recognition using conventional image processing and deep learning approaches," *J. Intell. Fuzzy Syst.*, vol. 36, no. 3, pp. 1997–2004, 2019. doi: 10.3233/JIFS-169911.

[19] T. K. N. Thanh, Q. B. Truong, Q. D. Truong, and H. H. Xuan, "Depth learning with convolutional neural network for leaves classifier based on shape of leaf vein," in *Proc. Asian Conf. Intell. Inf. Database Syst. (ACIIDS)*, 2018, pp. 575–585. doi: 10.1007/978-3-319-75417-8_53.

[20] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087. [Online]. Available: https://arxiv.org/abs/1703.05175.

[21] Z.-Y. Gao, H.-X. Xie, J.-F. Li, and S.-L. Liu, "Spatial-structure siamese network for plant identification," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 11, Nov. 2018, Art. no. 1850035. doi: 10.1142/S0218001418500350.

[22] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 539–546. doi: 10.1109/CVPR.2005.202.

[23] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural netwoRK," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 669–688, 1993. doi: 10.1142/S0218001493000339.

[24] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Dec. 2007, pp. 11–16. doi: 10.1109/ISSPIT.2007.4458016.

[25] O. Soderkvist, "Computer vision classification of leaves from Swedish trees," Teknik Och Teknologier, Tech. Rep., 2010.

[26] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 502–516. doi: 10.1007/978-3-642-33709-3_36.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, no. 2, pp. 1097–1105. doi: 10.1145/3065386.

[28] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019. doi: 10.1109/TGRS.2018.2872830.

[29] S. Zhang, Y.-K. Lei, and Y.-H. Wu, "Semi-supervised locally discriminant projection for classification and recognition," *Knowl.-Based Syst.*, vol. 24, no. 3, pp. 341–346, Mar. 2011. doi: 10.1016/j.knosys.2010.11.002.

[30] L. Longlong, J. M. Garibaldi, and H. Dongjian, "Leaf classification using multiple feature analysis based on semi-supervised clustering," *J. Intell. Fuzzy Syst.*, vol. 29, no. 4, pp. 1465–1477, Oct. 2015. doi: 10.3233/IFS-151626.

[31] T. Chen, S. Lu, and J. Fan, "SS-HCNN: Semi-supervised hierarchical convolutional neural network for image classification," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2389–2398, May 2019. doi: 10.1109/TIP.2018.2886758.

BIN WANG received the B.E. degree from Yantai University. He is currently pursuing the M.S. degree with Beijing Forestry University. His current research interests include forestry image recognition and processing, computer vision, and deep learning.

DIAN WANG received the B.S. degree in electrical engineering from Northeast Forestry University, Harbin, China, in 2006, the M.S. degree in power electronic from the Harbin University of Science and Technology, Harbin, in 2009, and the Ph.D. degree in forest engineering from Beijing Forestry University, Beijing, China, in 2012.

From 2012 to 2015, he was an Assistant Professor with the College of Technology, Beijing Forestry University, and he held a postdoctoral position with Changlin Construction Machinery Group. Since 2016, he has been an Associate Professor with the Institute of Forestry Equipment. He was a Visiting Scholar with Auburn University, in 2018. He was named as the Director of the Vehicle Engineering Department, Beijing Forestry University. His research interest includes forestry equipment and automation related area. In 2018, he was a recipient of the China Forestry Society and the Liang Xi Forestry Science and Technology Award for Excellence.