Few-shot cotton leaf spots disease classification based on metric learning

Xihuizi Liang (liangxihuizi@alu.cau.edu.cn)
Suzhou Chien-Shiung Institute of Technology

Research

Keywords: Cotton leaf disease, few-shot learning, Support vector machine, Disease identification, Convolutional neural network

Posted Date: August 6th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-776867/v1

Few-shot cotton leaf spots disease classification based on metric learning

Xihuizi Liang[1*]
*Correspondence: liangxihuizi@sina.com
[1] Institute of intelligent manufacturing, Suzhou Chien-Shiung Institute of Technology, Jiangsu, China

Absrtact

Background: Cotton diceases seriously affect the yield and quality of cotton. The type of pest or disease suffered by cotton can be determined by the disease spots on the cotton leaves. This paper presents a small-sample learning framework that can be used for cotton leaf disease spot classification task, which using deep learning techniques is constructed based on a metric learning approach, to prevent and control cotton diseases timely. First, disease spots on cotton leaf's disease images are segmented by different methods, compared by using support vector machine (SVM) method and threshold segmentation, and discussed the suitable one. With segmented disease spot images as input, a disease spot dataset is established, and the cotton leaf disease spots were classified using a classical convolutional neural network classifier, the structure and framework of convolutional neural network had been designed, and the setting of relevant parameters and the detailed network structure configuration are analyzed according to the experimental environment. 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.

Results: To achieve the classification of cotton leaf spots by small sample learning, this paper constructs a metric-based learning method to extract cotton leaf spot features and classify the leaves. In the process of leaf spot extraction, image segmentation of the spots is performed by threshold segmentation and SVM, and comparative analysis is performed. In the process of leaf spot classification, the structural framework of leaf spot feature extractor and feature classifier is constructed, and the overall framework is built using the idea of two-way parallel convolutional neural network. A variety of excellent convolutional neural network feature extractors such as Vgg, DesenNet, and ResNet were used for feature extraction work, and a combination design based on the small sample classification framework was performed and compared. Experimentally, it is demonstrated that the classification accuracy is improved by nearly 7.7% on average for different number of samples in the case of using this optimizer. S-DesneNet have the highest accuracy. When n is 5, 10, 15 and 20, the accuracy is 58.63%, 84.41% ,92.51% and 91.75%, respectively, and the average accuracy is improved by nearly 7.7% compared with DenseNet.

Conclusions: To solve the problem of classification accuracy degradation due to small number of samples in small sample training tasks, a spatial structure optimizer (SSO) acting on the training process is proposed for this purpose.

Keywords: Cotton leaf disease, few-shot learning, Support vector machine, Disease identification, Convolutional neural network

Background

China is a major producer and consumer of raw cotton, at present, China produces about six million tons of raw cotton each year, the demand for raw cotton in about 10 million tons, according to the China National Reserve Grain Management Group, as the world's largest consumer of cotton and the second largest cotton producer, China's cotton production in 2020 is about 5.95 million ton. Xinjiang is the main production area of cotton in China, Xinjiang is the largest cotton production in China, long sunshine hours and frost-free period, especially suitable for the growth of cotton, with a unique geographical location, Xinjiang region is currently an important cotton production base in China, cotton production in Xinjiang in 2020 accounted for 87% of the national cotton production. With an area of 2,501.93 thousand hectares under cotton cultivation, the cumulative area of pests and diseases for the year 2020 is 1,459,900 hectares times, causing losses of up to 5.8%. With the increasing area of cotton in recent years, the crop layout is relatively single, cotton diseases and their control has become a big problem for the majority of cotton farmers. Therefore, identifying the types of cotton leaf's diseases accurately and taking corresponding preventive measures in time are of great significance for reducing the loss of cotton yield, improving the quality of cotton and increasing the income of cotton farmers.

In the past, the identification of cotton leaf's diseases mainly relies on farmers to conduct on-the-spot investigations and judge the categories of diseases based on experience[1]. Although some of the existing intelligent decision support systems are more accurate than those that used to rely solely on farmers' experience to determine the results of the disease, due to differences in the quality of individual agricultural producers and the influence of some human factors, agricultural producers are unable to make quantitative analysis and judgements on cotton diseases in conjunction with the actual disease situation of the plants, and the diagnostic criteria for plant diseases are ambiguous, resulting in unreasonable disease control. Therefore, in order to apply pesticides to cotton in an effective, rational and accurate manner, to ensure cotton yields, to reduce environmental pollution from pesticides and to reduce the dependence of agricultural production on labour, research and application of mechanised precision variable application methods and technologies is imperative. The use of machine vision technology for disease identification is a prerequisite for achieving this goal. It can be used to quickly and accurately obtain information about cotton diseases and to apply precise sprays according to the severity of the disease and the area of the cotton that has the disease. This will save pesticides, improve efficiency, reduce costs, reduce reliance on labour and greatly reduce pesticide pollution of the agro-ecological environment, which is very significant.

Now machine learning and image processing methods have been widely used for plant disease identification. Chaudhary et al. classified multi-class groundnut diseases by combining an improved random forest machine learning algorithm, an attribute evaluator method and an instance filter method[2]. Tetila et al. compared the performance of different classifiers, including sequential minimal optimization (SMO), adaboost, decision trees, K-NN, random forest, and naive Bayes, for the identification of soybean foliar diseases[3]. Ehsan et al. proposed a fuzzy logic classification algorithm to improve classification efficiency for healthy and disease infected strawberry leaves[4]. When these classical machine learning methods such as random forest, adaboost, decision trees and support vector machine are used to identify plant diseases, it is necessary to extract plant disease features which has a great influence on the identification accuracy. Since cotton infected by different diseases have minor difference in color and texture, the identification accuracy of cotton leaf's

diseases is low by using classical machine learning methods.

The deep learning method developed in recent years does not require manual extraction of target features when performing target identification. If enough training samples are available, the identification accuracy of deep learning method is high. Zhang et al. proposed the improved GoogLeNet and Cifar10 models based on deep learning for the identification of maize leaf diseases. These two improved models were used to test 9 kinds of maize leaf images[5]. Ferentinos developed a deep learning method to perform plant disease detection and diagnosis. The training samples came from an open database of 87,848 leave images of healthy and diseased plants[6].

Recently, convolutional neural networks in deep learning have become a major tool for solving image classification, image recognition and semantic segmentation problems with their outstanding performance in the field of computer vision. The application of deep learning methods to plant classification has performed extremely well, outperforming most of the semi-supervised learning methods based on manual feature extraction, and in particular, its excellent generalisation performance has achieved an important position in solving large-scale plant leaf classification problems. However, the disadvantage of using deep learning methods is also obvious. The prerequisite for obtaining a recognition network with high classification accuracy is that the network has sufficient supervised learning samples during the training phase[7]. Training the deep learning models above requires a large number of training samples. However, this is often difficult to accomplish, and in most cases we only have a small number of training samples of a certain object to be recognized[8, 9, 10]. The general deep learning convolutional neural network performs extremely poorly with a small number of learning training samples, because the network under-fits during the training learning process if there are not enough training samples[11].

Therefore, to solve this problem, the concept of few-shot learning was proposed. The aim of few-shot learning is to learn feature classification methods for these samples from few-shot learning of supervised training samples[12]. By learning quickly and accurately from a small number of samples, we can compare and analyse different learned samples in a metric-based way to make accurate judgements when faced with new samples[13]. We expect to train a deep learning classification network with a metric learning method that will allow it to perform few-shot learning tasks and apply it to the automatic classification of cotton pests and diseases, i.e. to determine the type of disease that a cotton suffers from by the color and texture of its leaves' disease spots.

This paper presents a low shot learning method for identification of cotton leaf's diseases by using support vector machine and deep learning networks, the method is evaluated by giving the experimental results. Support vector machine is used for low shot segmentation of disease spots on cotton leaf's disease images. Experimental results show that, SVM can segment disease spot images on the condition of low shot learning while retaining the edge information well, a k-nearest neighbor (kNN) classifier is used to classify leaves in the learned metric space. To solve the problem of classification accuracy degradation due to small number of samples in small sample training tasks, a spatial structure optimizer (SSO) acting on the training process is proposed for this purpose, and S- DesneNet have the highest accuracy.

The structure and framework of convolutional neural network that can be used for the small sample classification model in this paper. Besides that, the setting of relevant parameters and the detailed network structure configuration are analyzed according to the experimental environment.

The remainder of this paper is organized as follows. Section 2 describes materials and methodsused for the strategy in detail. Includding the Segmentation method of disease spots , and

the classification of disease in cotton leaf. Section 3 shows the results obtained by employing the proposed method, and discussed the effect among proposed method and other methods. Lastly, the conclusions are summarized in Section 4.

2. Material and methods

2.1 Image acquisition and material

Part of the cotton leaf's disease images used in this paper is taken in Xinjiang Agricultural Division 7 cotton at 44°25′27.61″ N, 84°57′27.15″ E, 464 m above sea level, in an area with aridity and low rainfall, annual sunshine hours of 2721-2818 h, annual precipitation of 125.0-207.7 mm, and average wind speed of 1.5 m/s, belonging to a typical temperate continental climate. The main cotton stems at the seedling stage were about 45 cm high and the number of leaves was around 13. Images were acquired using a Canon EOS 90D camera, the height of the camera from the cotton leaf is 30 mm, as shown in Fig.1. In this paper, the common and more serious diseases of cotton fields are Anthracnose, Verticillium and Ascochyta spot were acquired using the camera having a resolution of 640×480 pixels. In this paper, 50 pictures were selected for each of the three kinds of disease spots on cotton leaves, and a total number of 150 images were prepared for this study.

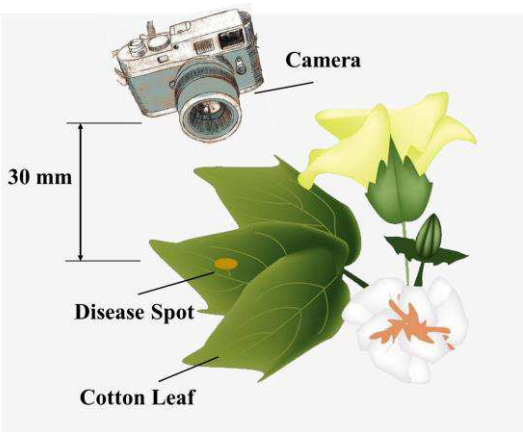


Fig.1 Camera photo diagram

In our experiment, 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.

2.2 Segmentation of disease spots

Cotton leaf's disease images are usually taken in the field with complex background which seriously interferes with the identification accuracy. In this paper, disease spots are segmented from cotton leaf's disease images to remove complex background.

2.2.1 Threshold segmentation

Determine the optimal segmentation threshold by iterative method. Select the median grayscale value of the grayscale map as the initial threshold[14, 15, 16]. The image is divided into two parts, the average gray value of the two parts before and after segmentation is calculated, and the median difference between the two average gray values is minimized as the goal, and the optimal segmentation threshold is determined by successive iterations, and the segmentation effect is

obtained by using the optimal threshold for image segmentation[15, 17]. The specific algorithm steps are as follows.

Step 1. Select the image to be segmented and record the median gray value of the image as T_i, and use T_i as the initial threshold for the iteration.

Step 2. Segmentation of the image into A and B regions by threshold T_i. Calculate the average grayscale value of area A as θ_A, and the average grayscale value of area B as θ_B.

Step 3. Calculate the new threshold value T_{i+1}, and the difference between the two thresholds, noted as Δ, which is is calculated as in Eq. 1 and Eq. 2.

$$T_{i+1} = \frac{1}{2}(\theta_A + \theta_B) \tag{1}$$

$$\Delta = T_{i+1} - T_i \tag{2}$$

Step 4. Step 2 and Step 3 are repeated until Δ is less than a given value, and then the optimal threshold value is obtained.

Step 5. Image segmentation is performed using the obtained optimal thresholds to obtain the final segmentation results.

2.2.2 SVM

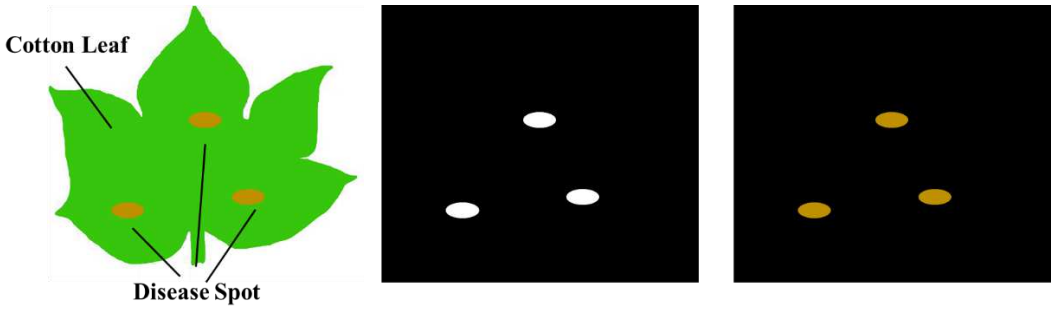
Support vector machine (SVM) is considered to be a classical binary classification algorithm[18, 19, 20, 21]. The basic model of SVM is the classifier with the largest interval in the feature space, and the core of training SVM classifiers is interval maximization, so it is suitable for high-dimensional, high-noise, few-shot learning. The specific algorithm steps are as follows.

Step 1. Extraction of features: Extraction of the 4-dimensional color features and the 6-dimensional texture features. The 4-dimensional color features include R, G, B three-channel pixel values and red-red index (2R-G-B) value, and the 6-dimensional texture features include grayscale co-occurrence matrix statistic for background and lesion regions measurement.

Step 2. Training SVM model: Cascade color features and texture features as spot features of diseased tea leaves, train SVM model, and obtain model parameters.

Step 3. Output category label: For the image of diseased tea leaves to be identified, the color features and texture features in the sliding window are extracted, and the cascaded features are input into the trained SVM model to label the diseased cotton leaf images (output label 1 for diseased areas and output label 0 for non-diseased background areas).

Diagram of these two different segmentation image effect of the disease spots are shown in Fig.2.



a b c

Fig.2 The diagram of the segmentation image effect of the spots; a The original images, b The threshold segmentation results, c SVM segmentation results.

The image effect shows that the segmentation results in preserving not only the edge contour of the lesion, but also the color and texture of the lesion. This paper uses SVM learning method to

segment disease spots in cotton leaf's disease images.

2.3 Classification of diseases

Images of diseased cotton leaves taken under natural scene conditions have complex backgrounds, which affect the accuracy of disease identification. For the segmented cotton leaf images, the disease spots are retained and the morphological characteristics of the spots are further used to carry out disease identification by deep learning methods for cotton leaf disease identification.

Deep learning method is a popular target identification method at present, but it has over-fit problem in the case of small training set size. In recent years, researchers classify sample learning models into three types: Metric Based, Mode Based and Optimization Based. Although with the development of machine learning techniques, small-sample learning has begun to experiment with applications in areas such as natural language processing, most of its research is still focused on computer vision. most of its research is still focused on computer vision.

Metric Learning in few-shot learning is admirably adapted to the leaf classification task, due to the fact that the differences between leaves differences in local features are more pronounced, thus making metric learning methods more effective compared to several other methods.

It refers to the use of a given pair of images to compute the distance between pairs of feature vectors. The degree of similarity between targets is measured and compared in some way The classification task can be accomplished by measuring and comparing the feature distances obtained through metric learning. In this paper, a parallel two-input model framework with a CNN feature extractor is used to build a metric network learning framework for small sample cases[22, 23, 24, 25]. In this study, all the different CNN structures are constructed according to the characteristics of the leaf classification task. In this study, all the different CNN structures are analyzed and experimentally validated.

The model framework is designed based on the theory of metric learning, firstly, the front-end of the structure is paired with different convolutional neural network feature extractors to automatically extract features from the dataset, and then a two-way parallel convolutional neural network architecture has been used to compare the input images and map their features into the same metric space.

The aim of metric learning is to be able to measure and compare the degree of similarity between targets in some way, which is the domain of machine learning.

Current Metric Learning methods automatically learn a metric distance function for a task depending on the target task, which reflects the difference in the degree of similarity between these targets for that task, and then form a specific metric space by learning the difference. The metric learning framework built in this paper is shown in Fig. 3.

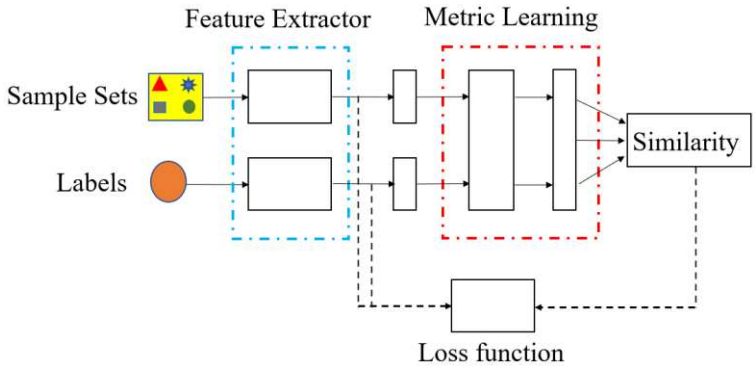


Fig. 3 Metric learning framework

In this paper, In the input side model, we have chosen the triplet label inputs $\{x^+, x^-, x_{\text{label}}\}$. Where when x^+ and x^- belong together, x_{label}=1, otherwise, where when x^+ and x^- belong to different classes, x_{label}=0.

The feature extractor part uses a convolutional neural network framework. The Euclidean distance function is used in the metric learning section to calculate the difference between the extracted features. The input image format is an RGB map and therefore needs to be calculated separately in each of its pixel channels. It is assumed that all pixel points are distributed in the first quadrant of the coordinate axes in their respective channels, and the positive half-axis and zero point of the X and Y axes, with initial pixel coordinates of (0,0), then its Euclidean distance is calculated as in Eq. 3.

$$L = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}\Bigg|_{p=2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

The second half of the pass and output of the feature values is directly chosen to relate to a fully connected layer containing only one neuron. As the Loss Function part is different from other convolutional neural networks, this paper uses a Comparison Loss Function to calculate the loss value of similarity. To facilitate the design of subsequent optimizers, we add a dimensional flag K to the formula, K = 0, 1, 2... In the absence of an optimizer K defaults to 0, i.e. it is not triggered, so the Loss Function is calculated as in Eq. 4.

$$F_{\text{loss}}(M, (X_1, X_2, Y), K) = \frac{1}{2N}\sum_{i=1}^{N} Y E_W^2 + (1 - Y)\max(m - E_W, 0)^2 \tag{4}$$

where E_w is calculated as in Eq. 5.

$$E_W(X_1, X_2) = \|X_1 - X_2\|_2 = \left(\sum_{i=1}^{N} \left(X_1^i - X_2^i\right)^2\right)^{\frac{1}{2}} \tag{5}$$

Where E_w represents the two-parametric Euclidean distance between the feature vectors X_1 and X_2 of the two samples, N represents the dimensionality of the sample feature vectors; Y represents the matching labels of the two samples, Y=1 means the samples belong to the same class, Y=0 means the samples belong to different classes, m parameter represents the upper threshold limit.

When the calculated Euclidean distance between the eigenvalues distance value exceeds the upper threshold value, the F_{loss} value is 0, indicating that the gap between samples is too large, and if the target label value is 1, the then it means that the F_{loss} value is increased, thus continuously adjusting the value of the similarity and loss function. The relationship between the Loss Value and the Euclidean Distance is shown in Fig.4. where red indicates similar samples and green indicates samples of different classes.

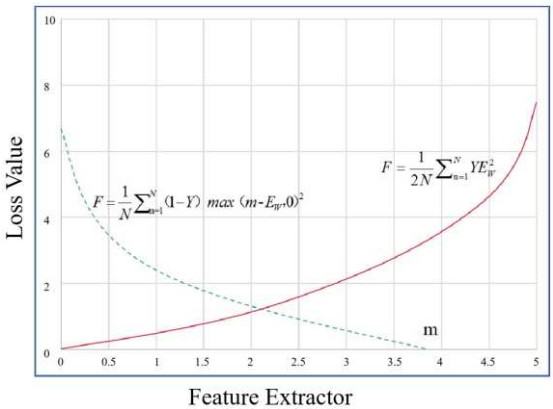


Fig.4 Loss Value and Euclidean Distance

2.3.1 Two-Way Parallel Network

Each time the input side of the model is trained, a training image pair is input, which consists of two images from the same or different leaf subclasses, and we designed a parallel neural network feature extractor structure for simultaneous input.

In the two-way parallel network structure, the weights within the feature extractor are identical and shared. Sharing the weights can reduce the model computation parameters by half and thus increase the processing speed and runnability of the network, and secondly, feature vectors of different dimensions can be mapped into the same space, facilitating the training process of metric learning.

The fully-connected layer was chosen to integrate the output of the feature extractor in the output layer of the extractor[26, 27]. The reason for choosing the fully-connected layer is to facilitate the use of batch training in the later training to speed up the learning and convergence of the network, and to facilitate the integration of the data under the same conditions. The structure is shown in Fig.5.

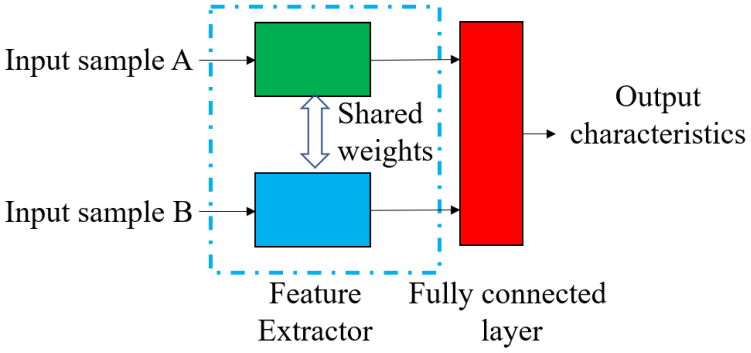


Fig.5 Structure of Two-Way Parallel Network

2.3.2 KNN

The trained metric space itself does not have a classification function, but only reflects the difference in similarity between the different types of samples in the space, so a separate classifier needs to be designed to classify the cotton leaf spots to be tested. In the framework of the model, the classification task is performed by the classification algorithm, which is selected according to the metric space.

The model used in the experiments approximates a two-dimensional metric space in the final plane, so we start with a planar search algorithm for the selection of classifiers.

The K-Nearest Neighbor (KNN) algorithm is one of the simplest and most effective classification algorithms in machine learning. In a fixed feature space, a sample is considered to belong to a certain sample if the nearest K samples around it belong to that sample. From the definition of the K-Nearest Neighbour algorithm It is clear from the definition of the K-Nearest Neighbour algorithm that the density and quality of the distribution of sample features in the metric space will determine the classification effect of the K-Nearest Neighbour algorithm, and hence the accuracy of the classified samples. This determines the accuracy of the classified samples.

The K-nearest neighbour algorithm in this study is determined by calculating the mean Euclidean distance value of the sample to be classified from other samples in the feature space. The class with the smallest mean Euclidean distance value is considered to be the class to which the sample to be tested belongs, and the principle is shown in Fig.6.

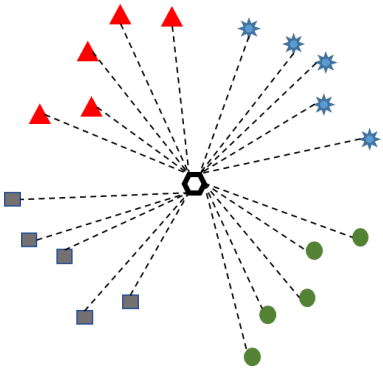


Fig.6 KNN classification of the test samples; five labeled samples per class, dotted lines represent European distances

Before making predictions, firstly, this paper needs to set up a supervised comparison sample. In this study, it is suggested that when the number of samples in the selected subclass is greater than 10, at least 30% of the samples are selected as supervised samples, and these supervised samples must belong to the same training set, and it is preferable to select samples with obvious population representativeness, which can be selected empirically, or by conducting experiments based on the results, etc.

When the sample to be classified, E, is extracted by the model's feature extractor, the Euclidean distance function is calculated. Once the Euclidean distance between sample E and a fixed number of supervised samples in the metric space is calculated, the value of the Euclidean distance between the sample to be classified and the other supervised samples is further calculated. The first five values that satisfy the condition and belong to the same category are the category to which the sample to be classified belongs, and the classification of the leaf is achieved. Therefore, the quality of the distribution of samples in the metric space directly affects the later classification results, and the sparser the distribution of different kinds of groups in the space, the better the KNN classifier will be.

To test the experimental results of all models under equal conditions, it is necessary to configure the same experimental environment for all the experiments in this paper.

The selection and construction of the feature extractor is the focus of this study. The common convolutional neural network framework cannot be applied under small sample conditions because there are too many parameters to be optimized in the network model, resulting in insufficient available feature values. There is no deep learning network model specifically applied to few-shot learning.

Therefore, we combined existing excellent deep learning network frameworks for experimental construction. In order to facilitate the loading and unloading of the model, the same fully connected layer is used in the output port of the convolutional neural network in this paper, and the number of parameters is adjusted according to the experimental environment. The number is set to 352 in this experiment, and although this may result in some networks not performing at their maximum, this is to facilitate This is to facilitate a cross-sectional comparison of different model characteristics. The overall identification process of the final model is shown in Fig.7.

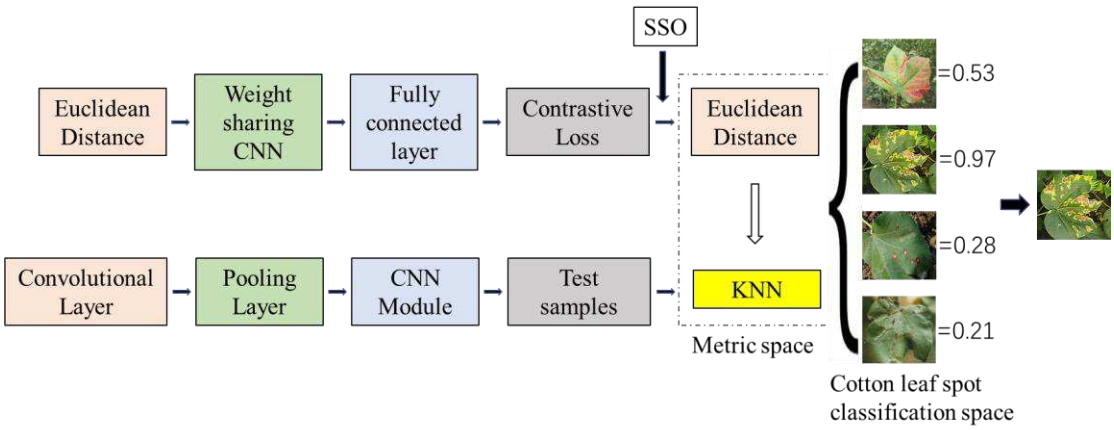


Fig.7 Overall experimental procedure framework.

In the experimental phase, the authors need to set up the training dataset and validation dataset reasonably according to the task scenario. A training dataset is constructed for different disease spots, and tab. 1 shows the number of positive samples and negative samples in the three subsets of the training datasets.

Tab.1 The number of positive samples and negative samples

Class	Number of positive samples	Number of negative samples
Dataset A	215	306
Dataset B	186	174
Dataset C	203	192
Dataset D	195	178

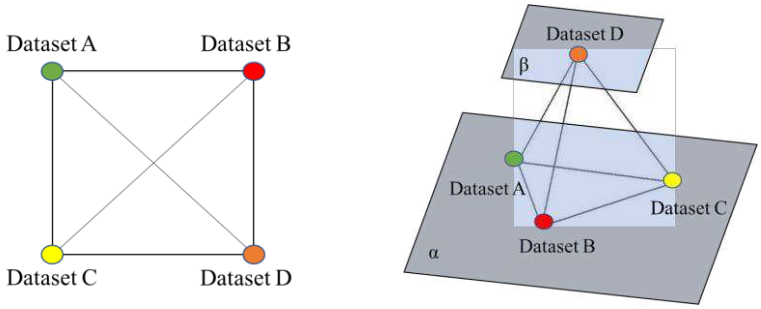
The focus of this study is to solve the task of cotton leaf spot classification in small samples. In this study, 10 or 15 categories are randomly selected for each spot in the dataset, and then a certain number of images are randomly selected from these categories to form the training dataset, and 5, 10, 15, and 20 images are selected in 5 increments to form different training datasets. In the experiments, 5, 10, 15, 20 images from each of the major categories of the dataset were selected as different training datasets in increments of 5. The images that are not selected from all the subcategories in a single test set will form the validation dataset, which will be used to evaluate the performance of the algorithm later[29].

When the number of samples to be extracted for features is insufficient, the convolutional neural network classifier can cause severe underfitting due to its own large number of parameters to be optimized. To solve this problem, this paper finds and builds a nonparametric optimization method in the framework of few-shot learning model.

The morphological similarity of leaf lesions will cause the model performance to degrade. For the four similar samples, if it is impossible to achieve this equal distribution of the four more similar samples in the plane space, in most cases the four similar samples will converge to a positive quadrilateral in the plane, and the equilibrium between their diagonal samples will be forced to break, which is shown in Fig. 8(a).

To achieve an even distribution of sample data in the space to ensure that each sample has a stable structure in the space, the average distribution is realized in space to ensure that each sample has a stable structure in space, and make sure that the oscillation of the model can be reduced, a spatial structure optimizer (SSO) is introduced in this paper for local optimization of the model, which is

shown in Fig. 8(b).



(a) Unstable sample space (b) Sample Space with SSO

Fig.8 Principle of the tetrahedron structure.

In the SSO, the distribution of these four samples in the four vertices of the positive tetrahedron can ensure the balance between the diagonal samples, whose distribution will form a relatively stable dynamic adjustment state. Thus leading to the convergence of the model to the best case, avoiding the repeated oscillations that will occur in the change of the loss function, and improving the performance of the model.

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 dispersion of the model's feature values at the beginning of training is random, which means that the reason for the high similarity between several samples appearing in the model's, beginning training phase may not be because their samples are more similar to each other, but that the model has not yet dispersed them well. Therefore the spatial structure optimizer is introduced in the middle and later stages of training and is used to improve the recognition accuracy of the model. The trigger condition is given by Eq. 6.

$$\frac{\sum_{i=0}^{n} f_i(d,a)}{n} \approx \frac{\sum_{i=0}^{n} f_i(d,b)}{n} \approx \frac{\sum_{i=0}^{n} f_i(d,c)}{n} \leq P \tag{6}$$

where n 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.

Due to the input-side limitation of the metric learning network structure in the experiment, each individual training sample consists of 2 images in a pair. 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. (7).

$$L_{loss} = -[f \times log(f_y) + (1-y) \times log(1-f_y) \tag{7}$$

where L_{loss} 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.

2.3.3 DenseNet

DenseNet is a method of training CNN models by building more dense pre-layer channels,

which is a method of training CNN models by building more dense front layer channels. The stacked channel structure has good performance from a feature perspective, as the features of the front layer are reused when they are equivalent to the features being renormalized. so even in DenseNet, it can achieve good loss convergence even when the BN layer is removed. The free combination of deep and shallow networks greatly reduces the chance of gradient disappearance during training, and the propagation of shallow features throughout the network allows The propagation of shallow features throughout the network allows all networks to access and utilize the original image features, which gives DenseNet a degree of adaptability in small sample tasks. This gives DenseNet the ability to adapt to small sample tasks.

The input to the classifier of most traditional CNN networks is the output of the last convolutional layer, i.e. the deepest feature map, whereas DenseNet can utilize the feature maps of all shallow layers, thus allowing for a smoother decision function with good generalization performance. With the number of samples increases, the improvement in classification accuracy is more moderate, which indicates that the network structure possesses good adaptability to the back-propagation process of the gradient This indicates that the network structure has a good ability to adapt to the backpropagation process of gradients. The DenseNet model is a suitable framework for few-shot learning feature extractors, provided that the memory space distribution can be optimized.

The free combination of deep and shallow networks greatly reduces the chance of gradient disappearance during training, and the propagation of shallow features throughout the network allows all networks to access and exploit the original image features, which gives DenseNet a degree of adaptability in small sample tasks. Moreover, DenseNet forms a wide network through dense connections, so that gradients are stable and converge quickly during training. To address the excessive memory consumption of DenseNet, the DenseNet tests used in this study were done on a computer with more memory.

A special structure in DenseNet is the Bottleneck layer. The principle is to adjust the number of channels by adding a 1×1 convolutional layer before the 3×3 convolutional layer in the Dense Block, to reduce the number of computational parameters and to integrate the individual channel features. The 1×1 convolutional layer has 4k channels, while the 3×3 convolutional layer has k, where k is called the growth rate and is set to 12 in this paper. The number of output channels of a Dense Block is equal to the number of input channels plus the growth rate k and the number of layers contained in the number of convolutional layers.

There are four Dense Block modules used in the model, and DenseNet uses direct connections in the connection method. DenseNet uses direct connections in the connection method, so it is necessary to ensure that the output feature map of each connection layer is the same size in the same Dense Block module.

In this paper, a lower growth rate was chosen when building DenseNet, thus reducing the number of parameters to be optimized and freeing up more memory space. The convolution module connecting the Dense Block does not use the original compression factor θ.

3 Results and Discussions

In this experiment, 40 images of each type of cotton diseases are selected. All 120 images are segmented to obtain disease spot images. 20 samples from each type of disease spot images are randomly selected as training samples, and the remaining 20 samples from each type are used as test samples.

3.1 Results for disease spot segmentation

Cotton leaf's disease images are usually taken in the field with complex background which seriously interferes with the identification accuracy. For image pre-processing the main objectives are: 1) to reduce irrelevant information in the image, e.g. by noise reduction, 2) to recover useful information and prevent information loss, 3) to make the information detectable and 4) to make the data simpler so that the reliability of recognition and detection is improved and thus the image can be better understood. In the process of crop disease identification In the process of crop disease identification, the quality of the image is degraded due to the influence of various factors in the acquisition of the image, resulting in noise bleeding into the image.

This method can be used as a numerical mapping of the corresponding images to remove invalid information, reduce the amount of data and improve the training efficiency. SVM works well for segmenting cotton leaf spot images with complex backgrounds. Better segmentation results on cotton leaf disease images with complex backgrounds.

Compared to threshold segmentation, under the same conditions, the threshold segmentation method and SVM segmentation method were compared, and the experimental results are shown in Figure 8.

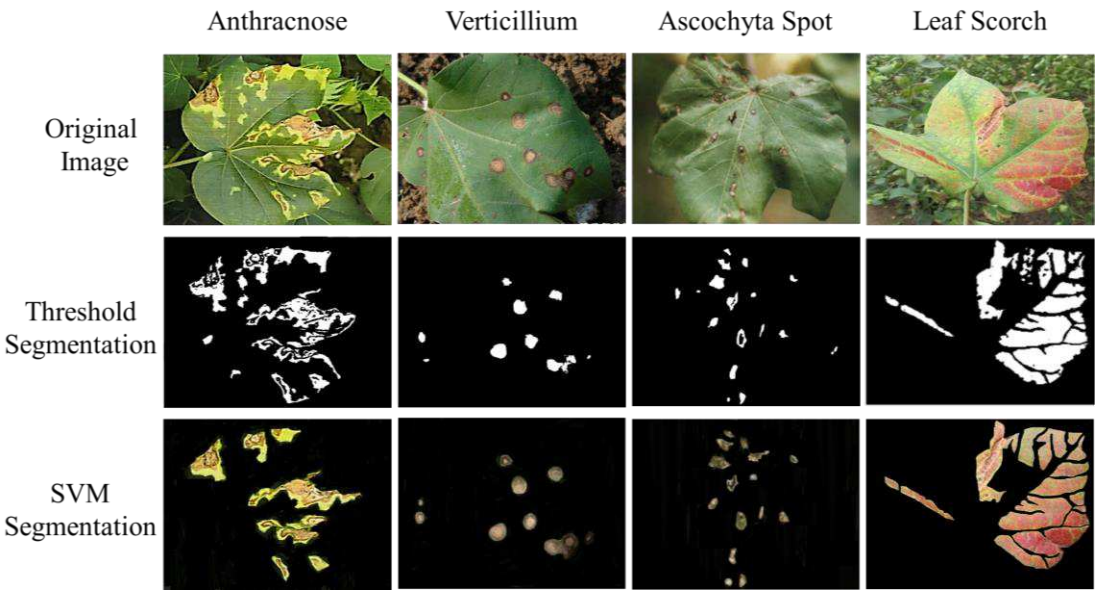


Fig. 8 Partial segmentation results of disease spots

The results show that SVM segmentation has the best effect on the segmentation of diseased cotton leaf images, except for the segmentation errors in the interference part, the disease spots are all segmented out. For the threshold segmentation method, the segmentation effect was good for the first and fourth pair of images with significant foreground background difference, and the segmentation effect was not good enough for the second and third images with small foreground background difference. According to the results, both threshold segmentation and SVM segmentation can segment out the lesions, but the SVM segmented lesions retain not only the traits of the lesions but also the color and texture of the lesions, preserving more features for further classification.

As can be seen in Figure 8, the area and morphology of the different spots on cotton leaves, as well as the density of the spots, vary, and the different characteristics are used to further classify the cotton leaf spots.

3.2 Results for cotton disease spots identification

We did two sets of experiments, one without SSO and the other with SSO, and then compared the results of these two sets of experiments.

In first experiment, the different convolutional neural network feature extractors used at the initial end of the model in this metric learning model building methods and final test effects in the learning classification framework. The selected convolutional neural network models are Vgg, DenseNet and its ResNeXt, respectively. 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. The best general structural framework is selected by comparing these models. The convolutional neural network is equivalent to a linear and non-linear transformation, and as the depth of the network increases the complexity of the non-linear transformation increases with the depth of the network. From the model test results, DenseNet has good resistance to underfitting in the absence of data features. The experimental results are shown in Fig. 9. As we can see from the figure, all the methods improve the accuracy when the number of supervised samples increases, but the DenseNet method improves faster, because the generalization ability of the DenseNet structure is better, and the width of the structure makes it possible to extract more features when the number of samples increases. So taking DenseNet as an example, when n is 5, 10, 15 and 20, the accuracy is 56.79%, 71.67%, 83.76% and 87.68%, respectively.

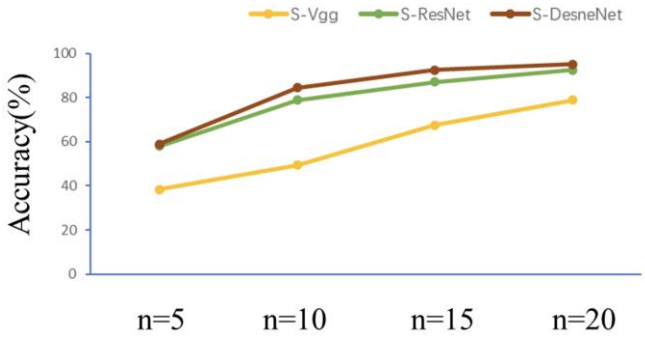


Fig. 9 Overall accuracy(%) of each model under the different sample size with SSO

In second experiment, the performance of Siamese network is used in each neural network frameworks, and Siamese add different network framworks called S-VGG, S-DenseNet and S-ResNet. The experimental results are shown in Fig. 10. The network with SSO is more accurate than the network without SSO structure, and S-DenseNet also obtain the best accuracy under different sample size conditions. Compared with other CNN frameworks, the adjusted S-DenseNet network can provide competitive results, and the results show that the S-DenseNet combination can achieve good accuracy when the training sample settings are appropriate. This finding is attributed to the advantages. This finding is attributed to the advantages of the DenseNet structure. When the number of samples is insufficient, the deep network gradient disappears seriously,and serious overfitting will occur. Taking S-DenseNet as an example, when n is 5, 10, 15 and 20, the accuracy is 58.63%, 84.41%, 92.51% and 95.13%, respectively, and the average accuracy is improved by nearly 7.7% compared with DenseNet.

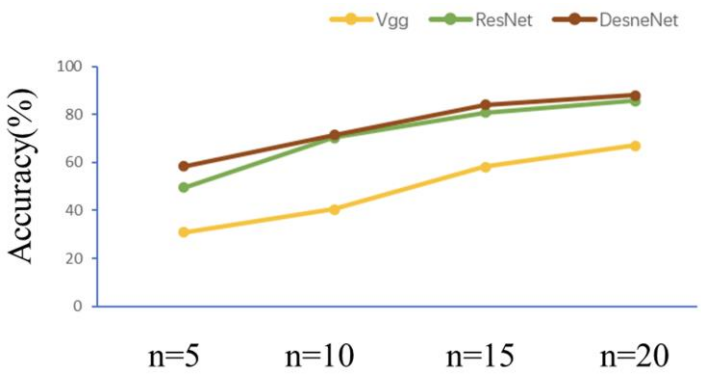


Fig. 10 Overall accuracy(%) of each model under the different sample size without SSO

Fig. 11 shows the variation in the loss curve with the SSO and without the SSO by b-spline curve. The average value of the 100 steps of loss training is calculated. As shown in the figure, the SSO loss curve converges faster and the descent process is smoother before the 15000 steps, and is more stable in the middle and late stages.

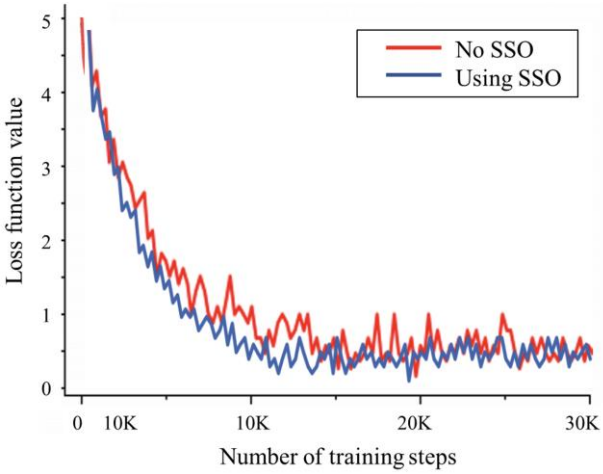


Fig. 10 Overall accuracy (%) from the same test dataset.

Fig. 12 shows the classification accuracy results of the kNN classifier on the same test dataset with and without the SSO. According to the histogram analysis, as the number of training steps increases, the network advantages of SSO training gradually emerge. After about 5K steps, SSO training accuracy takes a clear advantage, and a high classification accuracy is maintained in the later stages.

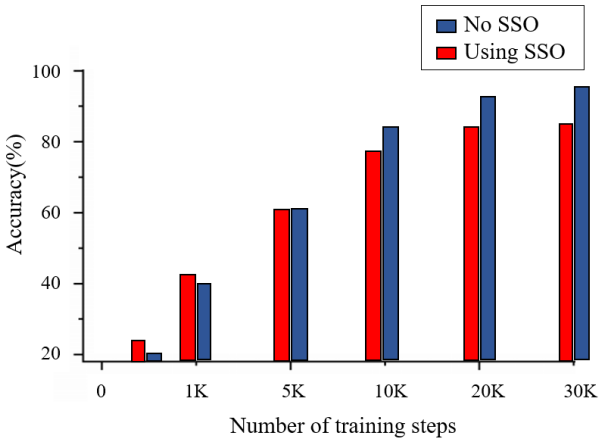


Fig. 11 Test dataset classification accuracy

To more intuitively compare the difference between the measurement space formed by whether using the SSO or not. In this paper, the feature maps of the last layer of the convolutional neural network of the model at the end of training are extracted and downscaled to a two-dimensional plane using principal component analysis. The results are shown in Figure 12 and Figure 13.

Fig. 12 is a metric space without the SSO. It can be seen from the graph that the distribution distances between disease spots 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.

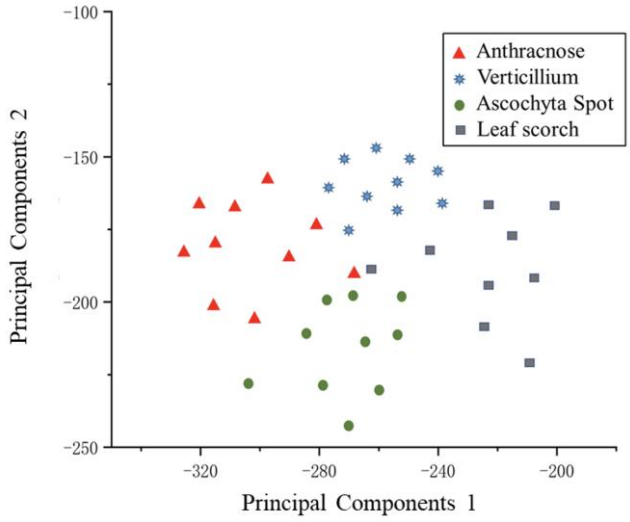


Fig. 12 Test dataset classification accuracy without SSO

Fig. 13 is a metric space formed using the SSO. As the distribution of the metric space tends to be more reasonable, the distributions of the same kind of disease spots is more concentrated, and the fault-tolerance rate is higher in the process of kNN classification, the quality of the distribution of samples in flat space is significantly improved.

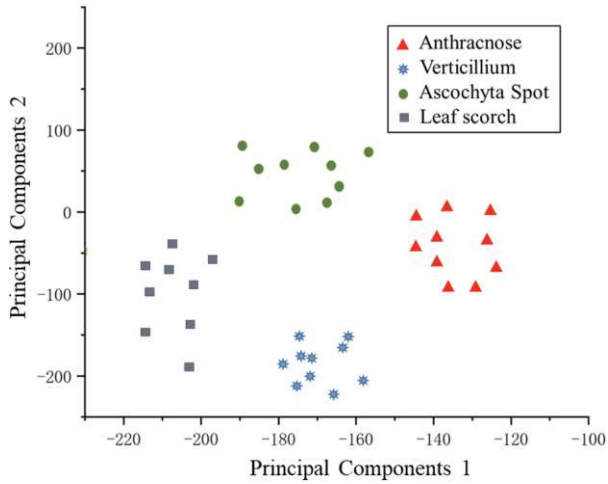


Fig. 13 Test dataset classification accuracy with SSO

5 Conclusions

In this paper, a small-sample learning framework that can be used for cotton leaf disease spot classification task using deep learning techniques is constructed based on a metric learning approach. Before performing the classification, the disease spots of cotton leaves are extracted by SVM

classification, and the disease spots are classified by the main features of classical convolutional neural network classifier, the structure and framework of convolutional neural network that can be used for the small sample classification model in this paper. Besides that, the setting of relevant parameters and the detailed network structure configuration are analyzed according to the experimental environment.

To solve the problem of classification accuracy degradation due to small number of samples in small sample training tasks, a spatial structure optimizer (SSO) acting on the training process is proposed for this purpose. Experimentally, it is demonstrated that the classification accuracy is improved by nearly 7.7% on average for different number of samples in the case of using this optimizer.

The experimental results show that when the number of training samples is 20, the classification accuracy of this method is much more higher, and S-DesneNet have the highest accuracy. When n is 5, 10, 15 and 20, the accuracy is 58.63%, 84.41% ,92.51% and 91.75%, respectively, and the average accuracy is improved by nearly 7.7% compared with DenseNet.

The next step is to find a better data generating method and a low shot learning method with strong generalization performance, so as to improve the robustness and accuracy of cotton leaf's disease identification with few training samples.

Acknowledgements

The authors would like to appreciate the China Agriculture University of for providing the facilities and financial support for this research.

Authors' contributions

Author have contributed during all stages of study. Author read and approved the final manuscript.

Funding

None applicable.

Availability of data and materials

The primary images that were acquired from cotton fields and the extracted features datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request. All the other data generated or analyzed during this study are included within this article.

Ethics approval and consent to participate

None applicable.

Consent for publication

None applicable.

Competing interests

The authors declared no competing interests.

References

1. Liu S, Wang R, Kang Y, Wan S, Hu W, Liu S. Salt Distribution and the Growth of Cotton Under Different Drip Irrigation Regimes in a Saline Area[J]. Agricultural Water Management, 2011, 100（1）: 58-69.

2. Kamal R, Chaudhary A, Kolhe S. An Improved Random Forest Classifier for Multi-Class Classification[J]. Information Processing in Agriculture, 2016, 3（4）: 215-222.

3. Tetila EC, Machado BB, Belete NA, Guimarães DA, Pistori H. Identification of Soybean Foliar Diseases Using Unmanned Aerial Vehicle Images[J]. IEEE Geosci. Remote Sensing Lett, 2017, 14（12）: 2190-2194.

4. Tofik M, Ehsan K. Identification of Plant Disease Infection Using Soft-Computing: Application to Modern Botany[J]. Procedia Computer Science, 2017, 120: 893-900.

5. M Zhang, Zhang X, Qiao Y. Identification of Maize Leaf Dis Eases Using Improved Deep Convolutional Neural Networks[J]. IEEE Access. 2018, 6: 30370-30377.

6. Konstantinos P. Ferentinos. Deep Learning Models for Plant Disease Detection and Diagnosis[J]. Computers and Electronics in Agriculture,2018, 145: 311-318.

7. Liu X, Li Y, Meng Q, Chen G. Deep Transfer Learning for Conditional Shift in Regression[J]. Knowledge-Based Systems, 2021, 227: 107216.

8. Li Y, Yang J. Meta-Learning Baselines and Database for Few-Shot Classification in Agriculture[J]. Computers and Electronics in Agriculture. 2021, 182: 106055.

9. Li Y, Yang J. Few-Shot Cotton Pest Recognition and Terminal Realization[J]. Computers and Electronics in Agriculture. 2020, 169: 105240.

10. Li M, Wang R, Yang J, Xue L, Hu M. Multi-Domain Few-Shot Image Recognition with Knowledge Transfer[J]. Neurocomputing. 2021, 442: 64-72.

11. Zhao P, Wu T, Zhao S, Liu H. Robust Transfer Learning Based On Geometric Mean Metric Learning[J]. Knowledge-Based Systems. 2021, 227: 107227.

12. Chao X, Zhang L. Few-shot Imbalanced Classification Based On Data Augmentation[J]. Multimedia Systems. 2021: 1-9.

13. Li Y, Chao X. ANN-Based Continual Classification in Agriculture[J]. Agriculture. 2020, 10（5）: 178.

14. Liang X, Chen B, Li M, Wei C, Wang J, Feng J. Dynamic Counting Method of Cotton Rows in Video Based On Centroid Tracking[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE). 2019, 35（2）: 175-182.

15. Jit Yan Lim, Kian Ming Lim, Shih Yin Ooi, Chin Poo Lee. Efficient-PrototypicalNet with Self Knowledge Distillation for Few-Shot Learning[J]. Neurocomputing. 2021.

16. Liang X, Chen B, Jiang Q, Zhu D, Yang M, Qiao Y. Detection Method of Navigation Route of Corn Harvester Based On Image Processing[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE). 2016, 32（22）: 43-49.

17. Liu G, Zhao L, Fang X. PDA: Proxy-Based Domain Adaptation for Few-Shot Image Recognition[J]. Image and Vision Computing. 2021, 110: 104164.

18. Liang X, Chen B, Li M, Wei C, Feng J. Method for Dynamic Counting of Cotton Rows Based On HOG Feature and SVM[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE). 2020, 36（15）: 173-181.

19. Chen G, Ge Z. SVM-tree and SVM-forest Algorithms for Imbalanced Fault Classification in

Industrial Processes[J]. IFAC Journal of Systems and Control. 2019, 8: 100052.
20. Wang R, Li W, Li R, Zhang L. Automatic Blur Type Classification Via Ensemble SVM[J]. Signal Processing: Image Communication. 2019, 71: 24-35.
21. Chaudhary A, Kolhe S, Kamal R. An improved random forest classifier for multi-class classification[J]. Information Processing in Agriculture. 2016, 3 (4), 215–222.
22. Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, Suneet Gupta. ToLeD: Tomato Leaf Disease Detection Using Convolution Neural Network[J]. Procedia Computer Science. 2020, 167: 293-301
23. Yang J, Wang C, Jiang B, et al. Visual perception enabled industry intelligence: state of the art, challenges and prospects[J]. IEEE Transactions on Industrial Informatics, 2020, 17(3): 2204-2219.
24. Li Y, Nie J, Chao X. Do we really need deep CNN for plant diseases identification?[J]. Computers and Electronics in Agriculture, 2020, 178: 105803.
25. Liu X, Hu C, Li P. Automatic Segmentation of Overlapped Poplar Seedling Leaves Combining Mask R-CNN and DBSCAN[J]. Computers and Electronics in Agriculture. 2020, 178: 105753
26. Avi Ben-Cohen, Eyal Klang, Stephen P. Raskin, Shelly Soffer, Simona Ben-Haim, Eli Konen, Michal Marianne Amitai, Hayit Greenspan. Cross-Modality Synthesis From CT to PET Using FCN and GAN Networks for Improved Automated Lesion Detection[J]. Engineering Applications of Artificial Intelligence. 2019, 78: 186-194.
27. Burks T. F, S. A. Shearer, J. R. Heath, K. D. Donohue. Evaluation of Neural-Network Classifiers for Weed Species Discrimination[J]. Biosystems Engineering. 2005, 91（3）: 293-304.
28. Li Y, Chao X. Semi-supervised few-shot learning approach for plant diseases recognition[J]. Plant Methods, 2021, 17(1): 1-10.