Hindawi
Complexity
Volume 2021, Article ID 6868592, 12 pages
https://doi.org/10.1155/2021/6868592

WILEY | Hindawi

*Research Article*

# A Fruit Tree Disease Diagnosis Model Based on Stacking Ensemble Learning

**Honglei Li**,[1] **Ying Jin**,[1] **Jiliang Zhong**,[2] **and Ruixue Zhao**[2]

[1]*Liaoning Normal University, Dalian, China*
[2]*Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing, China*

Correspondence should be addressed to Honglei Li; lhl@lnnu.edu.cn and Ruixue Zhao; zhaoruixue@caas.cn

Received 13 July 2021; Revised 26 August 2021; Accepted 28 August 2021; Published 15 September 2021

Academic Editor: Sampath Pradeep



Fruit tree diseases have a great influence on agricultural production. Artificial intelligence technologies have been used to help fruit growers identify fruit tree diseases in a timely and accurate way. In this study, a dataset of 10,000 images of pear black spot, pear rust, apple mosaic, and apple rust was used to develop the diagnosis model. To achieve better performance, we developed three kinds of ensemble learning classifiers and two kinds of deep learning classifiers, validated and tested these five models, and found that the stacking ensemble learning classifier outperformed the other classifiers with the accuracy of 98.05% on the validation dataset and 97.34% on the test dataset, which hinted that, with the small- and middle-sized dataset, stacking ensemble learning classifiers may be used as cost-effective alternatives to deep learning models under performance and cost constraints.

## 1. Introduction

In recent years, due to the influence of global climate and environmental changes, crop disasters around the world occur more frequently than ever, which results in a significant decline of the yield and quality of agricultural products, especially of fruit products. For example, the loss rate of fruit yield in the United States is about 20%, and that of some other countries is even up to 50% [1]. Crop disease has been the major reason that causes the yield loss of agricultural production, which limits the high-quality, high-efficiency, and sustainable development of the world's agriculture [1, 2]. However, most of the farmers have not master efficient and effective methods to identify fruit disease by themselves.

In the early 20th century, the traditional disease recognition methods were mainly based on biological experiments. Professionals used electronic microscopes and other equipment to observe bacterial changes, such as enzyme-linked immunosorbent Assays, DNA probe technology, PCR technology, and other biological methods [3–5]. However, those recognition methods cannot be widely practiced due to the large investment of instruments and equipment, and high cost of time and labors. Since the 1970s, a large number of traditional expert systems have been used to diagnose crop diseases. For example, PLANT/DS, as a kind of expert system, was developed to diagnose soybean diseases and insect pests [6]. In 1982, PLANT/CD was developed to diagnose corn borer pests. In 1990s, intelligent expert systems were developed to treat with agricultural problems. The various intelligent technologies were introduced to expert systems to improve the accuracy, intelligence, and practicability of disease diagnosis. However, expert systems are still in the role-based reasoning mode, which is considered much difficult to maintain and evolve. In recent 10 years, machine learning, especially deep learning, helps with plant disease diagnosis based on images recognition. This paper aims at proposing a machine learning-based model for fruit disease diagnosis.

## 2. Related Studies

Related studies have recently focused on image segmentation, feature extraction, and model training of diagnosis models of plant diseases. Jaisakthi et al. proposed a grape disease system, which can segment leaves from background

images and segment the ill areas based on global threshold processing and semisupervision technology. The systems with classification models were, respectively, trained with support vector machine, AdaBoost, and random forest machine learning algorithms [3]. Chakraborty et al. used Otsu thresholding algorithm and histogram equalization to preprocess images for recognition of black rot and cedar apple rust. They separated the image segmentation region of the infected part, and the accuracy of the improved multiclass SVM model was up to 96% [4]. Hossain et al. proposed a k-nearest neighbor (KNN) classifier to detect and classify black spots, anthracnose, bacterial wilt, leaf spots, and canker of various plants, which mainly depended on the extraction of color and texture features of ill leaves. The classifier was validated with the final accuracy of 96.76% [5]. To identify apple diseases, Zhang et al. extracted 38 features of color, texture, and shape of leaves and combined genetic algorithm with complete-fair-scheduler algorithm to extract the main features. They claimed that the recognition rate based on support vector classifier reached more than 90% [7]. Mohamed et al. carried out the research to identify the disease detection of four kinds of grape disease leaves, which included four stages: image enhancement with stretch method, image segmentation with *K*-means, texture feature extraction, and classification based on multi-SVM and Bayesian classifiers. The average accuracy was nearly 100% in their validation experiment [8]. For the diagnosis of four common alfalfa leaf diseases, Qin et al. extracted 129 features of texture, color and shape based on *K*-mean clustering algorithm, and linear discriminant analysis. After screening important features, a disease identification model was established based on SVM. The results showed that the SVM model built with the most important 45 features selected from 129 features was the final optimal model. For this SVM model, the recognition accuracies on the training set and the testing set were 97.64% and 94.74% [9].

In recent years, deep learning has attracted the attention of agronomic experts. Because of significant advantages of feature extraction and easy-to-use, deep learning technologies have effectively promoted the development of agricultural intelligent mechanical applications [10]. The related studies are mainly conducted in data enhancement and model improvement. For example, to identify five common apple leaf diseases, Jiang et al. constructed 26377 apple leaf disease samples through data enhancement and image annotation technology and proposed a deep CNN model by introducing GoogleNet Inception and Rainbow concatenation. The model achieved 78.80% in mean average precision [11]. Liu et al. proposed an architecture of deep CNN model based on AlexNet to detect diseases of apple leaves. Using 13689 ill leaf images as the sample dataset, the recognition rate of the model reached 97.62% in model test [12]. Based on more than 7000 pear disease images, Yang et al. established models using deep learning neural network models including VGG16, Inception V3, ResNet50, and ResNet101 to explore the relationship between key influencing factors and severity of pear disease. The recognition rate of diagnosis models was proved from 97.67% to 99.44% [13]. To identify types of maize leaf disease, Agarwal et al. improved the model from four aspects of enhanced convolution neural network (ECNN), fusion of extended convolution layer, one-dimensional convolution layer, and ECNN motivation. They established the ECNN model and achieved better performance than AlexNet, GoogleNet in Precision, Recall, and Accuracy [14]. Zhang et al. proposed multiscale fusion convolutional neural network (MSF-CNNS) for segmentation of cucumber ill leaf images. The method of gradual adjustment of transfer learning was adopted to speed up the training speed. By introducing multilevel parallel structure and multiscale connection, multiscale features of crop ill leaf images were extracted. The final average accuracy rate was 93.12%. Compared with Fully Convolutional Networks (FCNs), SegNet, U-NET, and DenseNet, the accuracy of the proposed model was increased by 13.00%, 10.74%, 10.40%, 10.08%, and 6.40%, respectively, and the training time was reduced by 0.9 hours [15].

Ensemble learning has also been introduced to image-based crop disease diagnosis. Ensemble learning aims at constructing a powerful classifier by using simple base classifiers. Ensemble learning successfully avoids the high training cost and large dataset demand of deep learning. For example, Rehman et al. proposed a hybrid contrast stretching method for apple ill leaves to increase the visual impact of the image, which used the pretrained CNN model for feature extraction. They achieved 96.6% recognition rate on the ensemble subspace discriminant analysis (ESDA) classifier [16]. To identify the three disease categories of corn leaves, Bhatt et al. collected the image features with CNN and used the boosting ensemble learning method with decision tree classifiers to train the features from CNN. The experiment showed that the accuracy of the model was up to 98% [17]. Azim et al. proposed a model to detect three kinds of rice leaf diseases. By removing the background, segmenting the disease area, extracting color, shape, and texture features, they used eXtreme gradient boosting (XGBoost) to enhance the recognition performance. The result showed that the accuracy of 86.58% was achieved [18].

## 3. The Data Source and Feature Engineering

*3.1. The Data Source.* In this study, we selected samples for four common fruit tree diseases including pear black spot, pear rust, apple mosaic and apple rust. These diseases are the most common diseases for apple and pear trees. The data for model training and validation are from the fruit tree disease image library of the Agricultural Knowledge Service System of Chinese Academy of Agricultural Sciences (AKSS), which contains 10,000 leaf images of pear black spot, pear rust, apple mosaic, and apple rust diseases. There are 2,500 pictures of each disease. The pictures were collected by agronomists during the fruit tree growth period. As shown in Figure 1, each leaf picture is separated from the panorama with pure white background and the color temperature between 5200 and 5500. The resolution of the picture is $2816 \times 2112$.

We also used Baidu image search engine (https://image.baidu.com) with the disease names as keywords to gather the fruit leaf images for the model test. As a result, 500 images

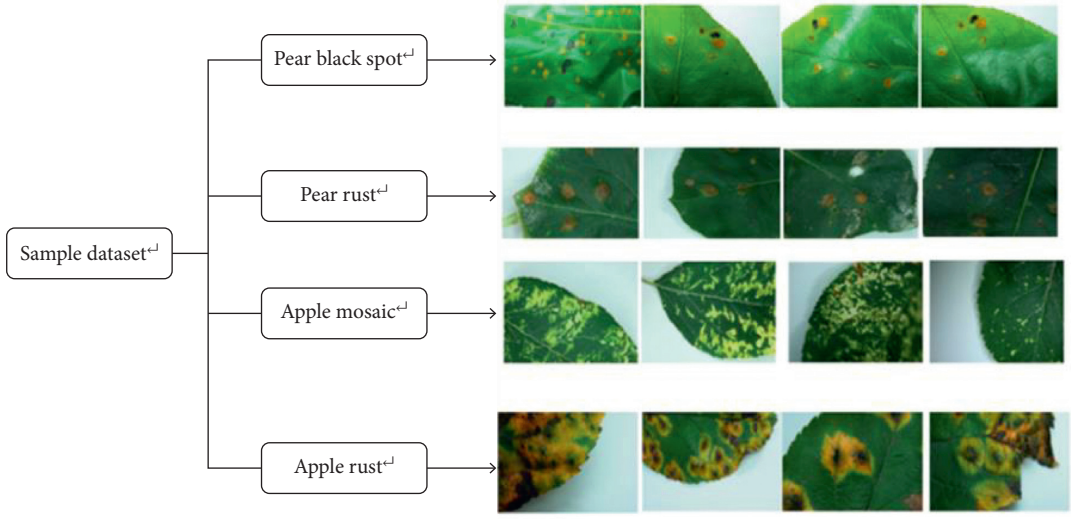


Figure 1: Images in training and validation dataset.

were finally selected into the test dataset by agronomic experts. As shown in Figure 2, the pictures for the model test are mixed with pictures in different quality levels and background, which is reasonable for generalization ability evaluation.

*3.2. The Feature Extraction.* The feature extraction is the process of extracting invariant features from images to solve practical problems. Before building the fruit tree disease diagnosis model, the features of ill leaves should be extracted. Theoretically, it is necessary to integrate multidisciplinary knowledge such as mathematics and physics to define the features of images. Technically, it is necessary to combine digital image processing and computer vision techniques to depict digital image features [19]. In practice, the features about color, shape, texture, and number of disease spots of leaves are usually used to recognize the plant disease.

*3.2.1. Color Feature Extraction of Ill Leaves.* CMYK, HSV, RGB, bitmap, and grayscale contribute to the representation of color attributes of pictures. In this study, RGB is used to define the color feature. RBG uses the change of red ($R$), green ($G$), and blue ($B$) color channels and their superposition to express a variety of colors. As one of the most widely used color systems, RGB system almost includes all colors that human vision can perceive. Since the color and size of the disease spots are clearly different from the healthy parts of the leaf and different from those of different diseases, the statistical description of RGB data contributes more to the recognition of leaf diseases. We defined the following indexes to describe the color feature of fruit leaves with RGB system.

As shown in Equation (1), $L_i$, the first moment of color data, denotes the general level of the color in channel $i$, where $P$ is the number of pixels in $R$, $G$, and $B$ channels, and $i$ is the channel ID. $X_{ij}$ is the color brightness value of channel $i$.

$$L_i = \frac{1}{P}\sum_{j=1}^{P} X_{ij}, \quad i = \{R, G, B\}. \tag{1}$$

In Equation (2), $\sigma_i$ is the second moment of color data, which uses the standard deviation (Std.) value to reflect the fluctuation degree of leaf colors.

$$\sigma_i = \left[\frac{1}{P}\sum_{j=1}^{P}\left(X_{ij} - L_i\right)^2\right]^{1/2}. \tag{2}$$

In Equation (3), $R_i$ denotes the range of color values in channel $i$, which reflects the extreme difference of colors in a channel.

$$R_i = \max\left(R_i\right) - \min\left(R_i\right). \tag{3}$$

In addition, since the mean value cannot objectively reflect the overall level of color in a channel when the data are not in normal distribution, we took the median value $M_i$ of channel $i$ as a supplement to $L_i$.

All color features of the dataset are shown in Table 1.

*3.2.2. Texture Feature Extraction of Ill Leaves.* As one important visual feature of pictures, texture refers to properties inherent to the surface of an object and optical properties, microgeometric features, and other information of the object surface that is closely related to it. In this study, through the observation of ill leaves of four kind of fruit trees, we found that spots of pear black spot, pear rust, and apple rust were scattered on the surface of ill leaves, while spots of apple mosaic leaf disease were irregular and spread in a continuous way. Therefore, the texture feature is an important factor to distinguish the different ill leaves.

As a powerful tool to extract texture features of pictures, the gray-level cooccurrence matrix (GLCM) statistically characterizes the cooccurrence level of gray-level pixels [19]. This kind of texture context information is adequately specified by the matrix of relative frequency $P(i, j \mid d, \theta)$, in

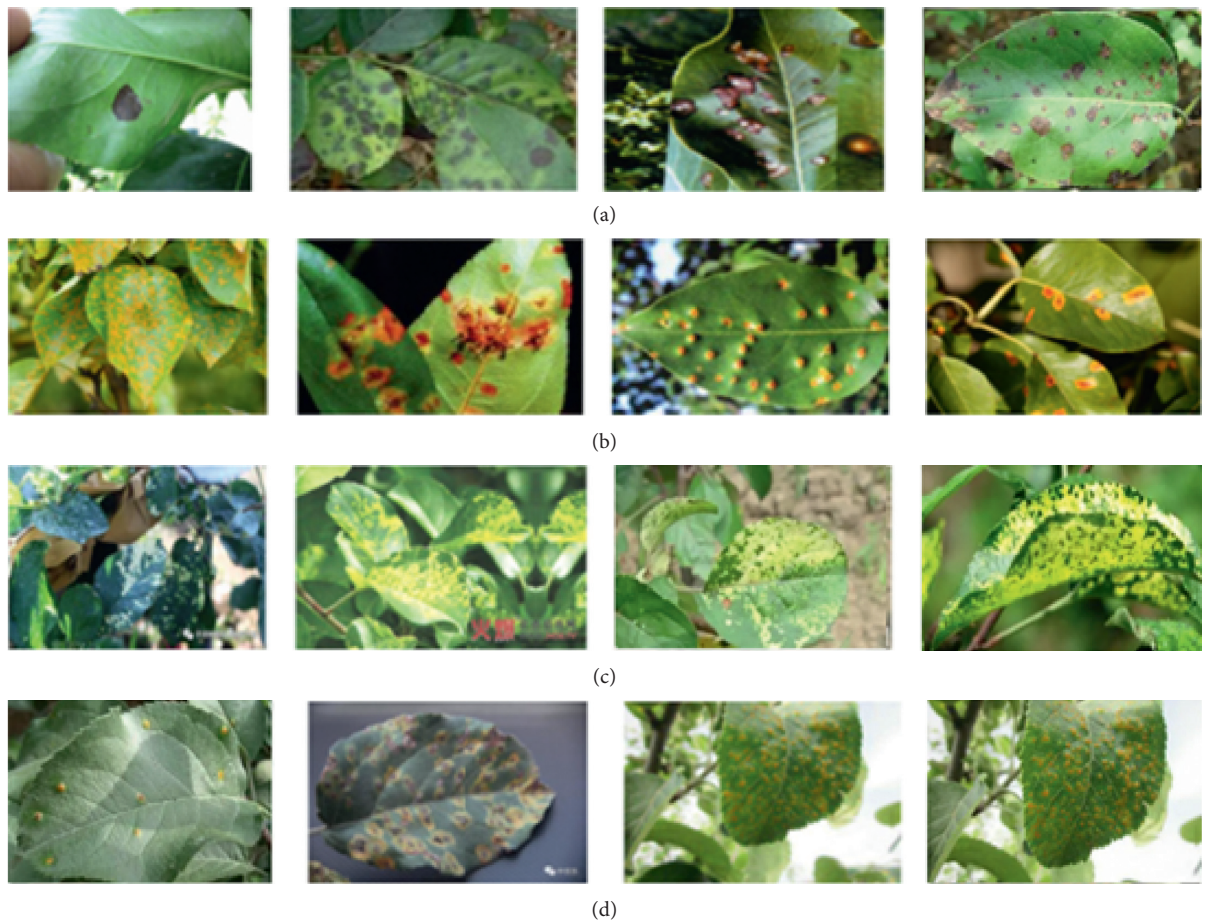

Figure 2: Fruit tree leaf images in test dataset. (a) Pear black spot. (b) Pear rust. (c) Apple mosaic. (d) Apple rust.

Table 1: Image color feature of disease images.

| Disease | $L_r$ | $\sigma_r$ | $R_r$ | $M_r$ | $L_g$ | $\sum_g$ | $R_g$ | $M_g$ | $L_b$ | $\Sigma_b$ | $R_b$ | $M_b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pear black spot | 78.6 | 3743.56 | 247.48 | 58.25 | 78.6 | 2312.51 | 226.46 | 138.03 | 94.47 | 2757.37 | 235.2 | 79.51 |
| Pear rust | 77.03 | 3255.78 | 223.28 | 56.25 | 77.03 | 2456.03 | 202.39 | 83.07 | 73.84 | 2580.88 | 200.37 | 54.74 |
| Apple mosaic | 85.6 | 5114.08 | 231.46 | 58.52 | 85.6 | 3412.91 | 210.34 | 97.25 | 90.12 | 3713.35 | 211.57 | 72.37 |
| Apple rust | 83.44 | 6153.76 | 224.45 | 52.58 | 83.44 | 3634.51 | 213.72 | 106.58 | 108.56 | 3231.94 | 206.13 | 101.41 |

which two neighboring resolution cells are separated by a distance $d$ occurs on the image, one with gray tone $i$ and the other with gray tone $j$ at the angle of $\theta$ (see Equation (4)), where $N$ is the gray level:

$$P(i, j \mid d, \theta) = \{(x, y) \mid f(x, y) = i, f(x + \mathrm{d}x, y + \mathrm{d}y) = j; x = 1, 2, \ldots, N, y = 1, 2, \ldots, N\}. \quad (4)$$

Such matrices of gray-tone spatial dependence frequencies are a function of the angular relationship between the neighboring resolution cells, as well as a function of the distance between them. $\theta$ is usually set to 0, 45, 90, and 135. Figure 3 illustrates a GLCM example with $d = 1$ and $\theta = 0$. The gray level of the image is 8.

With GLCM, Haralick et al. proposed 14 indexes to illustrate the texture of pictures, which includes angular second moment (ASM), contrast (CON), correlation, sum of squares, inverse difference moment (IDM), sum average, sum variance, sum entropy, entropy (ENT), difference variance, difference entropy, 2 information measures of correlation, and maximal correlation coefficient [20]. Due to the diversity of leaf image textures, 14 statistical indexes are all used in this study, and texture features are traded off by dimensionality reduction operations before model training. Table 2 shows the calculation results of some important texture indexes.

*3.2.3. Shape Feature Extraction of Disease Spots.* For a typical fruit disease, the shape of the disease spots in leaves is always more stable. However, the shape features of different disease spots are often different. Therefore, the shape features of disease spots are essential in the recognition of fruit diseases. Because the shape of the disease spots is often smaller and irregular, it is difficult to describe the shape

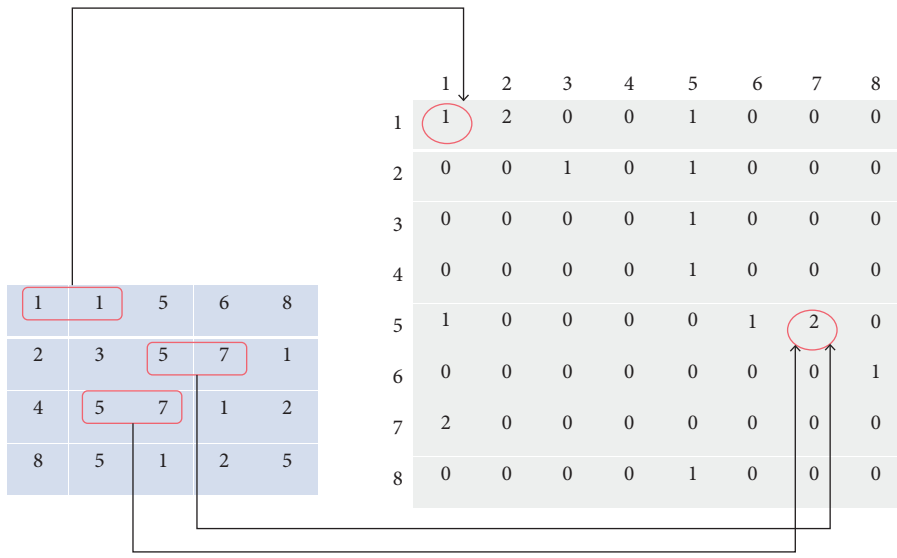


Figure 3: The gray-level cooccurrence matrix.

Table 2: Partial texture features of fruit tree ill leaves.

| Disease name | ASM | CON | ENT | IDM |
|---|---|---|---|---|
| Black spot | 0.072366 | 0.341840 | 3.044625 | 0.874390 |
| Pear rust | 0.206153 | 0.197791 | 2.138423 | 0.914096 |
| Apple mosaic | 0.096281 | 0.248845 | 2.856540 | 0.893278 |
| Apple rust | 0.113039 | 0.234975 | 2.739097 | 0.906248 |

features on the contour of disease spots. In this study, fractal dimension methods are introduced to extract the shape features of disease spots.

At present, the methods to calculate the fractal dimension of irregular objects include box counting method, perimeter area method, variable method, and radius method. Among them, the box counting method is popular and easy to use. It is available whether the object is a curve or a surface surrounded by a curve, and it has little to do with the physical nature of the object. The counting dimension value $D$ used in box counting method is defined in the following equation:

$$D = \lim_{r \longrightarrow 0} \frac{\ln N(A, r)}{\ln r}, \tag{5}$$

where $N(A, r)$ is the number of pixels in all square grids with the width $r$, and $A$ is the binary image matrix. Figure 4 shows a spot in the grid-like background.

In practice, for ease of computation, the linear fitting coefficient of $N(A, r)$ and $r$ is often used as the approximate value of $D$. The coefficient is easy to get by ordinary least squares (OLS) method. In this study, the counting dimension values of different fruit diseases are shown in Table 3.

*3.2.4. The Number of Feature Extraction of Disease Spots.* For different diseases, the number of disease spots differs to some extent. By observing the ill leaf pictures, we found that (1) there are a few black spots for pear black spot and a large number of yellow spots for pear rust; (2) there is a large area of light colored patches for apple rust, and the color of the spots has obvious variability. So, we adopted SimpleBlobDetector (SBD) [21, 22] to count the disease spots.

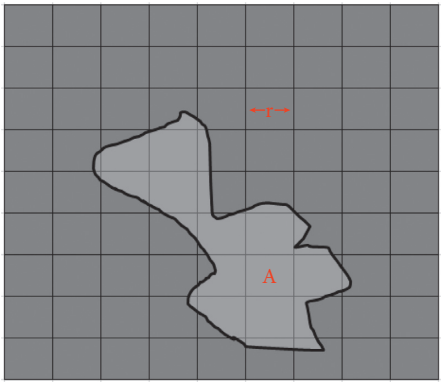


Figure 4: The box counting dimension by grids.

Table 3: The counting dimension values of different fruit diseases.

| Diseases | Mean | Std. | Median | Min. | Max. |
|---|---|---|---|---|---|
| Black spot | 1.5625 | 0.0987 | 1.5716 | 1.0103 | 1.7722 |
| Pear rust | 0.9976 | 0.2072 | 0.9563 | 0.3428 | 1.5793 |
| Apple mosaic | 1.1968 | 0.1789 | 1.2028 | 0.7519 | 1.6091 |
| Apple rust | 1.1333 | 0.2181 | 1.0907 | 0.4072 | 1.6416 |

SBD is a kind of image segmentation methods based on topological and morphological theories. This algorithm is good at handling weak edge information and has good ability to connect grayscale edges. Meanwhile, the catch basin concept effectively preserves the regional features of the image. Therefore, SBD is suitable for image segmentation. The flowchart of SBD is shown in Figure 5.

The main parameters of SBD are set as follows:

(i) minThreshold = 10

(ii) maxThreshold = 250

(iii) minCircularity = 0.3

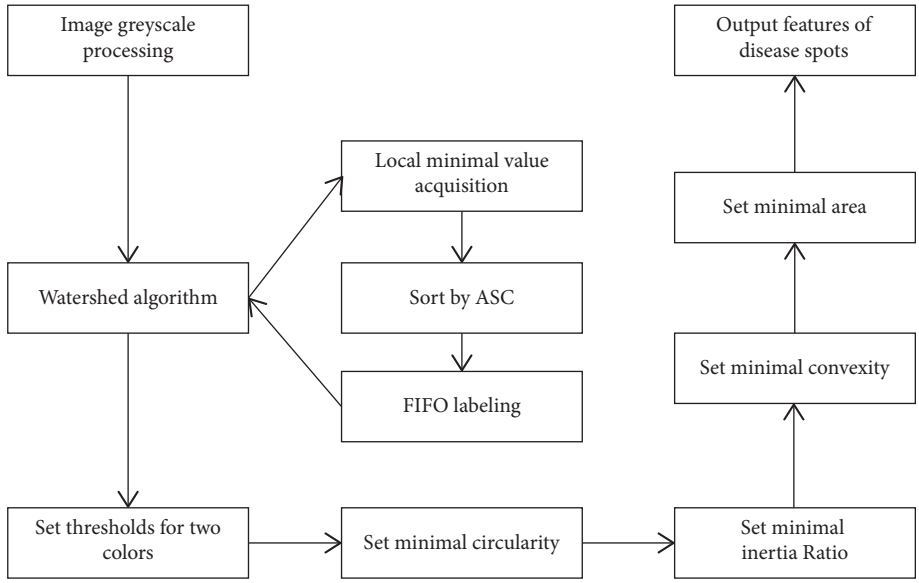


Figure 5: Flowchart of simple blob detector.

(iv) minInertiaRatio = 0.1

(v) minConvexity = 0.5

(vi) minArea = 100

Since the color brightness of the spots varies from dark to gray, we capture white blobs and black blobs separately. The blobColor parameter is set to 255 to count white spots and 0 to count black spots. The number features of disease spots are described in Tables 4 and 5.

*3.3. Data Standardization and Dimensionality Reduction.* Through the feature extraction, we got 33 features to describe the leaf pictures. These features are grouped into 12 color features, 14 texture features, 2 number features, and 5 shape features. Since the values of different features vary in different ranges, before the model training, we conducted the data standardization. Each feature is transformed by the following equation:

$$v' = \frac{v - \mu}{\sigma}, \tag{6}$$

where $\mu$ is the mean of feature samples, and $\sigma$ is the standard deviation.

To simplify the diagnosis model and improve the generalization performance of the model, we also reduced the dimensionality of the 33-feature dataset. Principal component analysis (PCA) was used to conduct the task. PCA algorithm has only one parameter, $n$_Components, which is used to determine the dimension after dimensionality reduction or the proportion of information retained after dimensionality reduction. The parameter is usually set according to experience rather than definite rules. To ensure the rationality of dimensionality reduction, $n$_Components were validated from 2 to 33. The reduced datasets with different dimensions were tested by a classifier such as logistic regression classifier. The dimensions with the best $f$1_score were selected. The test results are shown in Figure 6.

Table 4: The number features of disease spots (white).

| Diseases | Mean | Std. | Median | Min. | Max. |
|---|---|---|---|---|---|
| Black spot | 11.66 | 18.28 | 5 | 0 | 116 |
| Pear rust | 2.28 | 2.52 | 2 | 0 | 23 |
| Apple mosaic | 34.38 | 24.24 | 49.25 | 0 | 146 |
| Apple rust | 3.94 | 4.17 | 3 | 0 | 35 |

Table 5: The number features of disease spots (black).

| Diseases | Mean | Std. | Median | Min. | Max. |
|---|---|---|---|---|---|
| Black spot | 48.81 | 75.65 | 16 | 0 | 454 |
| Pear rust | 1.74 | 3.27 | 1 | 0 | 31 |
| Apple mosaic | 8.91 | 10.62 | 5 | 0 | 68 |
| Apple rust | 12.26 | 8.99 | 10 | 0 | 65 |

According to Ockham's Razor, the dataset with 6 dimensions was taken as the dataset for model training and validation. The data after dimensionality reduction is shown in Table 6.

## 4. Model Training and Selection

As is known in machine learning, deep learning is the most popular technology for image recognition. However, if the training dataset is in small and middle size, the performance of the deep learning model is not certainly guaranteed. In this study, we, respectively, conducted ensemble learning and deep learning tactics to find the best model. The tactics are compared by the metrics defined in Section 4.1.

*4.1. Model Evaluation Metrics.* We used $f$1_score to evaluate machine learning models. $f$1_score is defined in Equations (7) and (8). As a deliberately designed metric, $f$1_score fairly measures the bias and variance of the model.

$$f1\_\text{score} = \frac{2 * P * R}{P + R}, \tag{7}$$

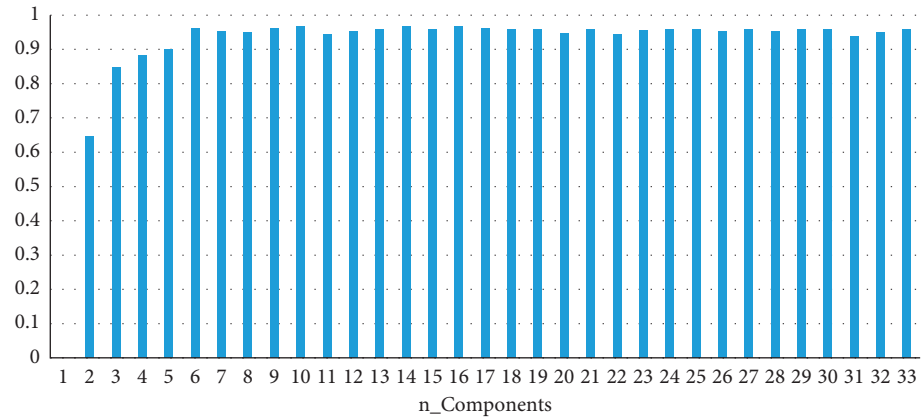


Figure 6: Dimensionality reduction results with different *n*_components.

Table 6: The dimensionality reduction result.

| ID | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | −6.40276 | −4.32052 | 3.007755 | 1.928639 | −0.50985 | −0.90559 |
| 1 | −5.70798 | −3.92368 | 2.348488 | 0.807734 | 0.048524 | −0.78388 |
| 2 | −5.32771 | −2.8948 | 1.272052 | −0.02776 | 0.214765 | −0.33611 |
| 3 | −4.89776 | −3.548 | 0.525648 | −1.50622 | 1.545405 | −0.42973 |
| 4 | −0.82885 | −1.73172 | −0.94987 | −0.66864 | 0.219924 | −3.6E-05 |
| ... | | | | | | |

where

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \quad (8)$$

In the above equations, true positive (TP) refers to the number of positive classes identified by the model; true negative (TN) refers to the number of negative classes identified by the model; false positive (FP) refers to the number of false positive classes identified by the model; false negative (FN) refers to the number of false negative classes identified by the model. Precision ($P$) refers to the proportion of true positive classes in the set of predicted positive classes. Recall ($R$) is the ratio of true positive classes in the predicted results to all actually true positive classes.

*4.2. Ensemble Learning.* Ensemble learning is a powerful way to integrate many weak classifiers for better prediction. In practice, ensemble learning classifiers show better performance than a unique classifier, even almost better than deep learning ones on small- and middle-sized datasets. According to different ensemble tactics, ensemble learning is divided into 3 branches: bagging-based ensemble learning, boosting-based ensemble learning, and retraining-based ensemble learning.

*4.2.1. Bagging Ensemble Learning-Based Model Training.* In bagging ensemble learning, all base classifiers are trained concurrently, so the efficiency of the training is much higher than other ensemble learning algorithms. If the sampling of features is also different from other base classifiers, the generalization ability is further improved. The output of the bagging ensemble learning model is usually decided by majority vote. The flowchart of the training of bagging ensemble learning is shown in Figure 7.

In this study, we chose random forest algorithm, which is proved to be one of the best ensemble learning algorithms [23, 24]. Since random forest uses classification and regression tree (CART) and feature sampling to train base classifiers, the main hyperparameters including max_depth, max_features, min_samples_leaf, min_samples_split, and n_estimators are required to be determined before the model training. We used GridSearchCV method of Scikit-learn to optimize the hyperparameters and got the final random forest-based diagnosis model. The final parameters were determined with the following values:

{"max_depth": 40, "min_samples_split": 2, "min_samples_leaf": 1, "*n*_estimators": 100, "max_features": 0.6}.

The $f1$_score was 0.9249, and the train time was 38.4 minutes.

*4.2.2. Boosting Ensemble Learning-Based Model Training.* Boosting is one of the most important developments in model training methodology. Boosting works by sequentially applying a classification algorithm to reweighted versions of the training data and then taking the majority voting or weighted mean of the sequence of classifiers thus produced. That is, after a base classifier is trained, the latter base classifier is trained on the validation result of the former base classifier. The weight of the false predicted samples will be adjusted to improve the latter classifier's accuracy. As a result, the bias of the latter classifier is decreased. Generally, the final outputs of ensemble classifiers are averaged with different weights [25]. The

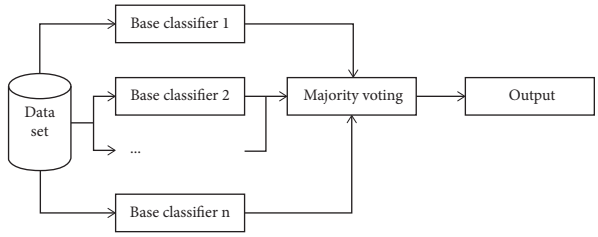


Figure 7: Bagging ensemble learning process.

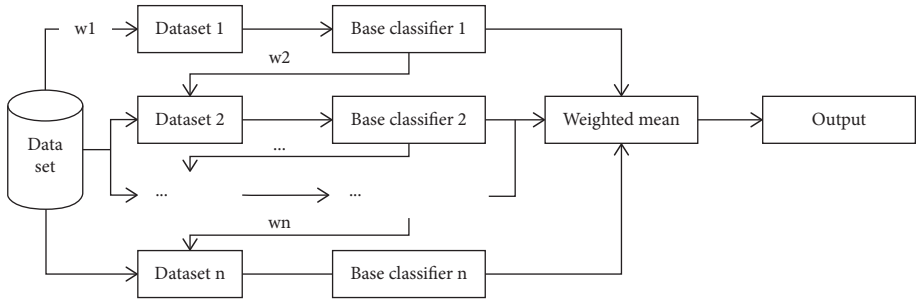


Figure 8: Boosting ensemble learning process.

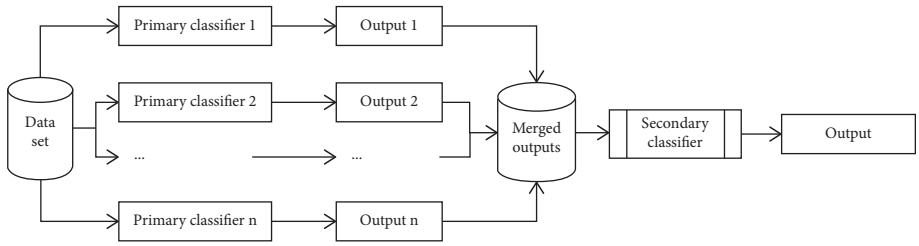


Figure 9: Stacking ensemble learning.

flowchart of the training of boosting ensemble learning is shown in Figure 8.

Boosting ensemble learning algorithm family has some famous members such as AdaBoost, GBDT, lightGBM, and XGBoost. Compared with AdaBoost and other algorithms, XGBoost uses the same sampling methods as random forest, which is proved to decrease the variance effectively. We also used GridSearchCV method to optimize the hyper-parameters and got the final XGBoost-based diagnosis model. The best parameter sets are as follows:

{"subsample": 0.7, "learning_rate": 0.1, "max_depth": 8, "colsample_bytree": 0.5, "*n*_estimators": 200}.

The *f*1_score is 0.9398, and the train time is 42.6 minutes.

*4.2.3. Retraining Ensemble Learning-Based Model Training.* Retraining ensemble learning uses the primary classifiers' outputs as inputs to train the secondary classifier. The typical algorithm is the stacked generalization (also called stacking ensemble) algorithm [26, 27]. The algorithm mainly consists of primary classifiers and secondary classifier. The number and type of primary classifiers are not limited. However, for the sake of efficiency and generalization ability, the simple classical and heterogeneous classifiers are preferred. Figure 9 shows the flowchart of the stacking ensemble learning process.

In this study, we explored 4 classical simple classifiers as primary classifiers and random forest as the secondary classifier to create the stacking ensemble model. The 6-dimension dataset was used to train these classifiers. In the first training stage, all primary classifiers were trained by GridSearchCV to get the high-accuracy classifiers and their outputs. Then, in the second training stage, the outputs were merged as the training data to train the secondary random forest classifier. Table 7 shows the test results.

If the secondary classifiers are trained concurrently, the total training time can be estimated to be about 54.8 minutes, which is the sum of the time of support vector classifier in first training stage and random forest in the second stage.

*4.3. Deep Learning-Based Model Training.* Convolutional neural network (CNN) is the typical deep learning technology for image recognition. CNN usually consists of an input layer, convolutional layers, pooling layers, a fully connected layer, and an output layer. Convolutional layers are used to extract features. In convolutional layers, activation functions such as ReLu and Sigmoid function are

Table 7: Test results of classifiers in the stacking ensemble diagnosis model.

| Classifiers | Type | $P$ | $R$ | $f1$_score | Time (Mins.) |
|---|---|---|---|---|---|
| K-nearest neighbors [28] | Primary | 0.9362 | 0.9496 | 0.9429 | 20.6 |
| Logistic regression [29] | Primary | 0.8503 | 0.9124 | 0.8803 | 18.2 |
| Support vector classifier [30] | Primary | 0.9444 | 0.8947 | 0.9189 | 24.5 |
| Naive Bayesian [31] | Primary | 0.9213 | 0.9286 | 0.9249 | 23.7 |
| Random forest | Secondary | 0.9786 | 0.9825 | 0.9805 | 30.3 |

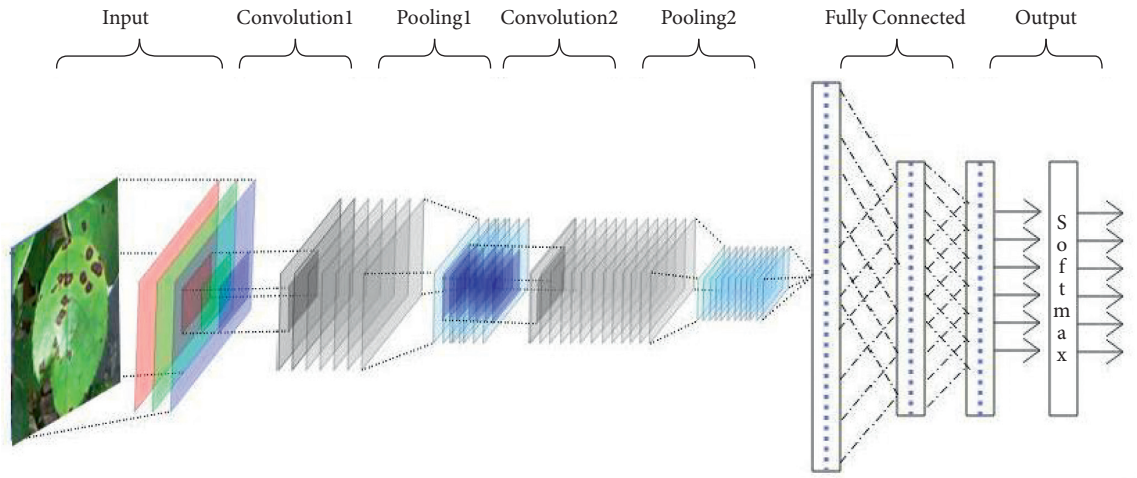


Figure 10: The process of convolutional neural network.

Table 8: Test results of CNN-based diagnosis models.

| Classifiers | $P$ | $R$ | $f1$_score | Time (hours.) |
|---|---|---|---|---|
| ResNet-101 | 0.9517 | 0.9565 | 0.9541 | 113.7 |
| DenseNet-121 | 0.9737 | 0.9613 | 0.9675 | 100.3 |

used to express complex features. After feature extraction, features are fed into pooling layers for feature selection and information filtering. Consequently, the high-dimension data are significantly condensed before being fed into the fully connected layer. The process is shown in Figure 10.

CNN is a family of many deep learning algorithms. In this family, there are many famous algorithms including LeNet [32], AlexNet [33], ZF Net [34], GoogLeNet [35], VGGNet [36], ResNet [37], and DenseNet [38]. In this study, ResNet-101 and DenseNet-121, as two popular CNN algorithms, were selected to create diagnosis models. We conducted the model training in TensorFlow [39] on the 33-dimension dataset. Test results of two models are shown in Table 8.

## 5. Results and Discussion

Figure 11 shows test results of different models in Sections 4.2 and 4.3. The stacking ensemble-based model gets the top $f1$_score 0.9805, then the DenseNet-121 model (0.9675), the ResNet-101 model (0.9541), the XGBoost model (0.9398), and the random forest model (0.9249).

Obviously, the stacking ensemble-based model outperformed other models on small- and middle-sized dataset, and the time-consuming (nearly 60 minutes) is acceptable. On the contrary, two deep learning models also show better scores. However, the training time is much more than that of the stacking-based model. This hints that even if deep learning algorithms usually showed better performance than other algorithms in image recognition, their performance may not be as good as the performance of simple machine learning algorithms when datasets are not large enough and diverse enough.

To further evaluate the above models, we used the test dataset introduced in Section 3.1 to test the models. Since the test data are not used in model training, we can evaluate the generalization ability of all models. The $f1$_scores of the models are 93.88% (random forest), 94.65% (XGBoost), 97.34% (stacking), 95.21 (ResNet-101), and 96.27% (DenseNet-121). The stacking-based model was still the best one.

We observed the outputs of all models. 57 out of 500 test samples were predicted with inconsistent values, among which the stacking-based model has the most right prediction values. Table 9 shows the difference.

We also observed the accuracy of the models on different diseases. Table 10 shows that the stacking-based model is still better than other models.

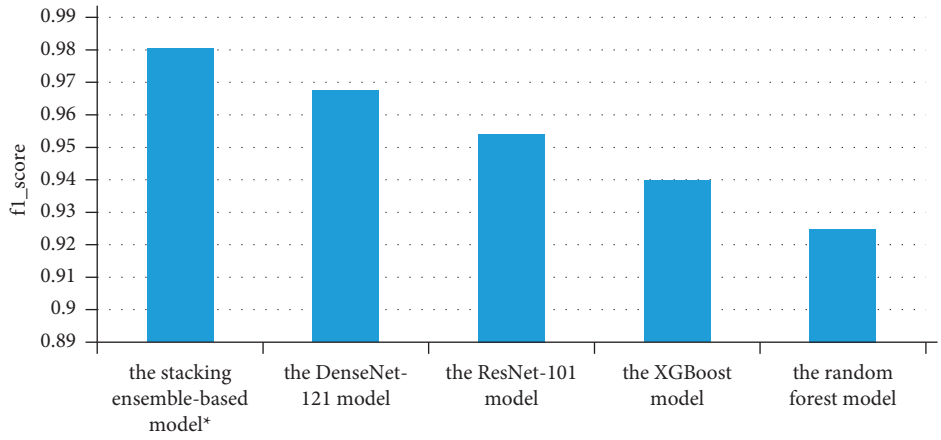


Figure 11: The $f1$_scores of diagnosis models.

Table 9: Comparison of inconsistent results of all models.

| Sample ID | Predicted values | | | | | Original values |
|---|---|---|---|---|---|---|
| | XGBoost | Random forest | Stacking | ResNet-101 | DenseNet-121 | |
| 3 | 3 | 0 | 0 | 3 | 0 | 3 |
| 10 | 2 | 2 | 2 | 1 | 2 | 2 |
| 22 | 1 | 2 | 0 | 2 | 2 | 2 |
| 37 | 3 | 2 | 3 | 3 | 2 | 3 |
| 43 | 2 | 0 | 0 | 0 | 0 | 0 |
| 84 | 1 | 3 | 1 | 1 | 1 | 1 |
| 102 | 2 | 1 | 1 | 2 | 1 | 2 |
| 156 | 3 | 3 | 3 | 3 | 2 | 3 |
| ... | | | | | | |
| 495 | 2 | 3 | 3 | 2 | 3 | 2 |

Table 10: The prediction accuracy of models with different diseases ($f1$_score).

| Diseases | XGBoost (%) | Random Forest (%) | Stacking (%) | ResNet-101 (%) | DenseNet-121 (%) |
|---|---|---|---|---|---|
| Pear black spot | 95.79 | 94.27 | 98.01 | 96.11 | 97.33 |
| Pear rust | 95.47 | 95.73 | 98.36 | 96.03 | 96.18 |
| Apple mosaic | 93.62 | 92.89 | 96.82 | 94.52 | 96.21 |
| Apple rust | 93.73 | 92.61 | 96.16 | 94.17 | 95.36 |

According to the results demonstrated above, the stacking ensemble-based model is selected as the final model for the diagnosis of fruit tree disease.

## 6. Conclusions and Future Studies

To automatically identify fruit tree diseases with leaf pictures, we trained the machine learning models with ill leaf pictures to create the diagnosis model. Since the size of the dataset is not large enough to implement reliable deep learning models, we trained 3 kinds of ensemble learning models and compared the accuracy of ensemble learning models with 2 deep learning-based models. The results showed that the stacking ensemble-based model outperformed other kinds of models. This study also hinted that when the dataset is in small and middle size, the accuracy of the deep learning models may not be satisfactory. The ensemble learning models, especially the stacking ensemble-based model, would be a high cost-effective solution with the help of high-quality feature engineering. Some studies tried ensemble learning of deep learning classifiers and implemented high accuracy of prediction [40]. However, the cost of the model training was heavily increased, while the efficiency of the model was decreased. It hinted that stacking ensemble learning classifiers may be used as cost-effective alternatives to deep learning models under performance and cost constraints.

It should be noted that the study has limitations in feature engineering and test data collection. (1) As was discussed in Section 3.2, we only tried RGB color scheme to extract the color features and box counting method to extract the shape features, which inevitably led to incomplete and inaccurate feature expression. (2) The test dataset to evaluate and select the final model was limited to the size and diversity, which may lead to inaccurate evaluation and choice of the best model. Therefore, in future studies, we will improve our work

on feature engineering and high-quality dataset collection to develop better models for fruit tree disease diagnosis and extend the model to the diagnosis of other crop diseases.

## Data Availability

The training dataset was downloaded from the database (http://agri.ckcest.cn/specialtyresources/list29-1.html).

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

The authors would like to thank Liaoning Normal University for the lab facilities and the necessary technical support and funding of China Knowledge Center for Engineering Sciences and Technology. This research was funded by China Knowledge Center for Engineering Sciences and Technology Project (Grant no. CKCEST-2020-1-20).

## References

[1] M. Dutot, L. M. Nelson, and R. C. Tyson, "Predicting the spread of postharvest disease in stored fruit, with application to apples," *Postharvest Biology and Technology*, vol. 85, pp. 45–56, 2013.

[2] D. Louro and D.-E. Lesemann, "Use of protein A-gold complex for specific labelling of antibodies bound to plant viruses I. Viral antigens in suspensions," *Journal of Virological Methods*, vol. 9, no. 2, pp. 107–122, 1984.

[3] S. M. Jaisakthi, P. Mirunalini, D. Thenmozhi, and Vatsala, "Grape leaf disease identification using machine learning techniques," in *Proceedings of 2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pp. 1–6, Chennai, India, February 2019.

[4] S. Chakraborty, S. Paul, and M. D. Rahat-uz-Zaman, "Prediction of apple leaf diseases using multiclass support vector machine," in *Proceedings of the 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 147–151, Dhaka, Bangladesh, January 2021.

[5] E. Hossain, M. F. Hossain, and M. A. Rahaman, "A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier," in *Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–6, IEEE, 2019.

[6] R. S. Michalski, "Designing extended entry decision tables and optimal decision trees using decision diagrams," 1978, https://mars.gmu.edu/handle/1920/1543.

[7] C. Zhang, S. Zhang, Y. Jucheng, S. Yancui, and C. Jia, "Apple leaf disease identification using genetic algorithm and correlation based feature selection method," *International Journal of Agricultural and Biological Engineering*, vol. 10, no. 2, pp. 74–83, 2017.

[8] K. K. Mohammed, A. Darwish, and A. E. Hassenian, "Artificial intelligent system for grape leaf diseases classification," *Artificial Intelligence for Sustainable Development: Theory, Practice and Future Applications*, Springer, Cham, Switzerland, pp. 19–29, 2021.

[9] F. Qin, D. Liu, B. Sun, L. Ruan, Z. Ma, and H. Wang, "Identification of alfalfa leaf diseases using image recognition technology," *PLoS One*, vol. 11, no. 12, Article ID e0168274, 2016.

[10] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[11] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[12] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, p. 11, 2018.

[13] F. Yang, F. Li, K. Zhang, W. Zhang, and S. Li, "Influencing factors analysis in pear disease recognition using deep learning," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1816–1828, 2021.

[14] R. Agarwal and H. Sharma, "Enhanced convolutional neural network (ecnn) for maize leaf diseases identification," *Advances in Intelligent Systems and Computing*, vol. 1168, pp. 297–307, 2021.

[15] S. Zhang, Z. Wang, and Z. Wang, "Method for image segmentation of cucumber disease leaves based on multi-scale fusion convolutional neural networks," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 36, no. 6, pp. 149–157, 2020.

[16] Z. U. Rehman, M. A. Khan, F. Ahmed et al., "Recognizing apple leaf diseases using a novel parallel real-time processing framework based on mask RCNN and transfer learning: an application for smart agriculture," *IET Image Processing*, vol. 15, no. 10, pp. 2157–2168, 2021.

[17] P. Bhatt, S. Sarangi, A. Shivhare, D. Singh, and S. Pappula, "Identification of diseases in corn leaves using convolutional neural networks and boosting," in *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods*, pp. 894–899, Prague, Czechia, February 2019.

[18] M. A. Azim, M. K. Islam, M. M. Rahman, and F. Jahan, "An effective feature extraction method for rice leaf disease classification," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 463–470, 2021.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[21] Simple Blob Detetor. Open CV documents, https://docs.opencv.org/master/javadoc/org/opencv/features2d/SimpleBlobDetector.html.

[22] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Computer Architecture Letters*, vol. 13, no. 06, pp. 583–598, 1991.

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] L. Breiman and A. Cutler, "Random forests-classification description," 2007, https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[25] J. Friedman, T. Hastie, and R. Tibshirani, "Special invited paper. additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, no. 2, pp. 337–374, 2000.

[26] L. Breiman, "Stacked regressions," *Machine Learning*, vol. 24, no. 1, pp. 49–64, 1996.
[27] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
[28] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1953.
[29] J. S. Cramer, *The Origins of Logistic Regression*, Social Science Electronic Publishing, Rochester, NY, USA, 2003.
[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[31] L. Zhang and H. Guo, *Introduction to Bayesian Networks*, China Science Publishing House, Beijing, China, 2006.
[32] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1097–1105, Curran Associates Inc., Lake Tahoe Nevada, December 2012.
[34] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013, https://arxiv.org/abs/1311.2901.
[35] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," 2014, https://arxiv.org/abs/1409.4842.
[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, May 2015, https://www.robots.ox.ac.uk/~vgg/publications/2015/Simonyan15.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
[38] H. Gao, L. Zhuang, V. D. M. Laurens, and Q. W. Kilian, "Densely connected convolutional networks," 2016, https://arxiv.org/abs/1608.06993.
[39] Google. Tensor Flow, https://github.com/tensorflow.
[40] H. Qi, Y. Liang, Q. Ding, and J. Zou, "Automatic identification of peanut-leaf diseases based on stack ensemble," *Applied Sciences*, vol. 11, no. 4, p. 1950, 2021.