



Classification of jujube defects in small data sets based on transfer learning

Jianping Ju^{1,2} · Hong Zheng¹ · Xiaohang Xu¹ · Zhongyuan Guo¹ · Zhaojun Zheng^{1,3} · Mingyu Lin²

Received: 29 September 2020 / Accepted: 5 January 2021

© The Author(s) 2021

Abstract

Although convolutional neural networks have achieved success in the field of image classification, there are still challenges in the field of agricultural product quality sorting such as machine vision-based jujube defects detection. The performance of jujube defect detection mainly depends on the feature extraction and the classifier used. Due to the diversity of the jujube materials and the variability of the testing environment, the traditional method of manually extracting the features often fails to meet the requirements of practical application. In this paper, a jujube sorting model in small data sets based on convolutional neural network and transfer learning is proposed to meet the actual demand of jujube defects detection. Firstly, the original images collected from the actual jujube sorting production line were pre-processed, and the data were augmented to establish a data set of five categories of jujube defects. The original CNN model is then improved by embedding the SE module and using the triplet loss function and the center loss function to replace the softmax loss function. Finally, the depth pre-training model on the ImageNet image data set was used to conduct training on the jujube defects data set, so that the parameters of the pre-training model could fit the parameter distribution of the jujube defects image, and the parameter distribution was transferred to the jujube defects data set to complete the transfer of the model and realize the detection and classification of the jujube defects. The classification results are visualized by heatmap through the analysis of classification accuracy and confusion matrix compared with the comparison models. The experimental results show that the SE-ResNet50-CL model optimizes the fine-grained classification problem of jujube defect recognition, and the test accuracy reaches 94.15%. The model has good stability and high recognition accuracy in complex environments.

Keywords Defect detection · Deep transfer learning · Sample imbalance · Loss function · Deep transfer learning

1 Introduction

The quality test of jujube includes size and defect test, which is to test the external quality characteristics of jujube and classify and identify according to the standard. At

present, research on jujube size nondestructive testing based on machine vision and machine learning has achieved good results, but the research on jujube defect detection is facing greater challenges.

The performance of jujube defect detection depends mainly on feature extraction and the classifier used. There are many varieties of jujube and there are huge differences between species; even the same variety of jujube has many types of defects and the color and texture of the defects will change as the season, planting environment and storage time change, planting environment and storage time; the environment of actual detection and grading is changeable, and the variation of debris, dust and light will also bring challenges. Due to the diversity of the jujube materials

✉ Hong Zheng
gjdxjjp@whu.edu.cn

¹ School of Electronic Information, Wuhan University, Hubei 430072, China

² School of Artificial Intelligence, Hubei Business College, Hubei 430000, China

³ Department of Public Courses, Wuhan Railway Vocational College of Technology, Hubei 430205, China

mentioned above and the variability of the testing environment, most of the hierarchical testing models established under ideal laboratory conditions based on traditional manual feature extraction methods have obvious limitations, and the robustness and repeated testing stability often fail to meet the requirements of practical application.

Compared with the traditional machine learning method, the deep learning method can dig deeper the deep structure of data the convolutional neural network (CNN) is the first deep Neural network that has been successfully trained. CNNs show excellent performance in image classification applications when the network structure is complex and the number of training samples is sufficient. Some scholars have applied CNNs to the field of agricultural product classification and classification, crop classification, and beef texture classification.

To detect red jujube in real applications, although we can collect a large number of jujube samples, the defect samples are very limited and unevenly distributed. The data received is actually a small set of data. Transfer learning can transfer the learned rules in the big data set to the new data, expand the application field of learning, and constantly improve the data model. Therefore, it is feasible and necessary to establish a pre-training model based on convolutional neural network and model adjustment based on transfer learning method to establish a defect classification and recognition model oriented to small data sets of jujubes.

At present, there are few practical testing equipment for the quality of jujube, but under the premise of ensuring the detection accuracy, optimizing the execution efficiency of the algorithm and reducing the complexity of the system are the key to the practicability of the grading detection technology.

2 Related works

2.1 Jujube defect and feature extraction

As shown in Fig. 1, the jujubes involved in this paper included normal jujubes and four kinds of defective jujubes. Healthy type jujube is brightly colored, fruit shape is full, and the surface is smooth. The rotted defect is the grayness of the jujube stem and head, or the darkening of

the rind and the darkening of the flesh. The split type defect was that the jujube began to crack from the fruiting part, and more than 1/10 of the break appeared in the direction of the long head, and the break was not discolored. The peeling type defect was that the skin was broken and part of the flesh was exposed. In the actual production process, the proportion of healthy dates is much higher than that of other types. The image collection of Russet defects is more difficult. The actual samples collected are the fewest, and the imbalance between the samples is very obvious.

Relevant scholars have studied methods for detecting jujube defects based on hyperspectral or near-red light imaging [1–4], and the cost of such methods is much higher than that based on computer vision. Feature extraction and selection are the key to surface defect detection of jujube based on machine vision. Common feature extraction methods include color feature extraction and texture feature extraction.

The color features are not sensitive to the rotation, translation, and scale changes of the image, and show strong robustness. By using color features, we can count the pixel points in the area of interest in jujubes, extract the disease area by hue difference, and extract the mean value and mean square error of H (hue) from HIS color space as the characteristic values [5, 6].

Common features of defect detection include: texture features, shape features, and color features. The texture features have rotation invariance, strong antinoise interference ability, extract the texture characteristics of jujube, and select the contrast and difference variance according to the characteristic parameters to realize the texture feature classification of jujube [7].

In the above research and exploration of jujube defects based on traditional machine vision, certain results have been achieved, but the research methods for jujube defects often only involve unilateral defects, and the accuracy and efficiency of the sorting are not high, and they are not yet satisfactory. The actual application needs of the project.

2.2 Application of deep learning in agriculture

At present, the application of deep learning in agriculture mainly includes weed identification, land cover classification, plant identification, fruit count, and crop type classification [8].

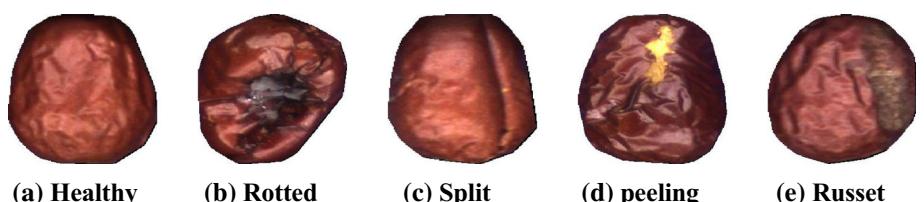


Fig. 1 Different examples of jujube samples

On the basis of VGGNet, Zhou, Yuncheng et al. Optimized the structure to design an 8-layer network for the extraction and expression of tomato main organ features, and applied various data augmentation techniques to train the network [9].

Yang Guoliang et al. proposed a parameter exponential nonlinearity (PENL) function to improve the residual network, using navel orange leaf image as a sample, and performing can training to distinguish lesions, deficiencies, normal, and non-species. Type [10].

Sharada P. Mohanty [11] used the AlexNet and GoogLeNet networks to test 14 crops and 26 diseases in 38 categories of 54,306 images in the PlantVillage dataset. Ramcharan et al. [12] used the Inceptionv3 network to treat three diseases of cassava Identification of two species of pests.

2.3 Transfer learning

With the development of deep learning methods, more and more researchers use deep neural networks for transfer learning. Transfer learning is a learning process that utilizes the similarity between data, tasks, and models to apply the knowledge (model parameters) learned from one environment (old domain) to the learning tasks in the new environment, so as to accelerate and optimize the Learning efficiency of the model [13].

Model transfer using the idea of transfer learning can transfer the model trained on big data to our task. By fine-tuning the task, we can have a good model for training on big data. Further, we can adaptively update these models for our tasks to achieve better results [14].

2.4 Style of learning transfer

There are three ways to implement transfer learning: (1) transfer learning, freeze all convolutional layers of the pre-training model, and only train your own custom connection layer; (2) extract feature vector, first calculate the convolution layer pair of the pre-training model The eigenvectors of all training and test data, then abandon the pre-training model and train only the self-customized fully connected network; (3) finetune, freezes the partial convolution layer of the pre-training model (usually the majority of the convolutional layer close to the input), training the remaining convolutional layer (usually a partial convolutional layer close to the output) and the fully connected layer [15–17].

2.5 Resnet and se-block

When building a convolutional network, the higher the depth of the network, the richer the feature hierarchy that

can be extracted. When using deeper network structures to obtain higher-level features, you will encounter problems with gradient disappearance, explosion, and network degradation. The above is not caused by overfitting, but by the redundant network layer learning parameters that are not identical mappings (the input and output of the layer are identical). In order to solve the above problems, He Kaiming et al. proposed a deep residual network (ResNet) constructed by a residual block, which is to replace the volume base layer (or fully connected layer) in the network with a residual block. Convert the original identity mapping function into learning and optimization of the residual function between input and output[18].

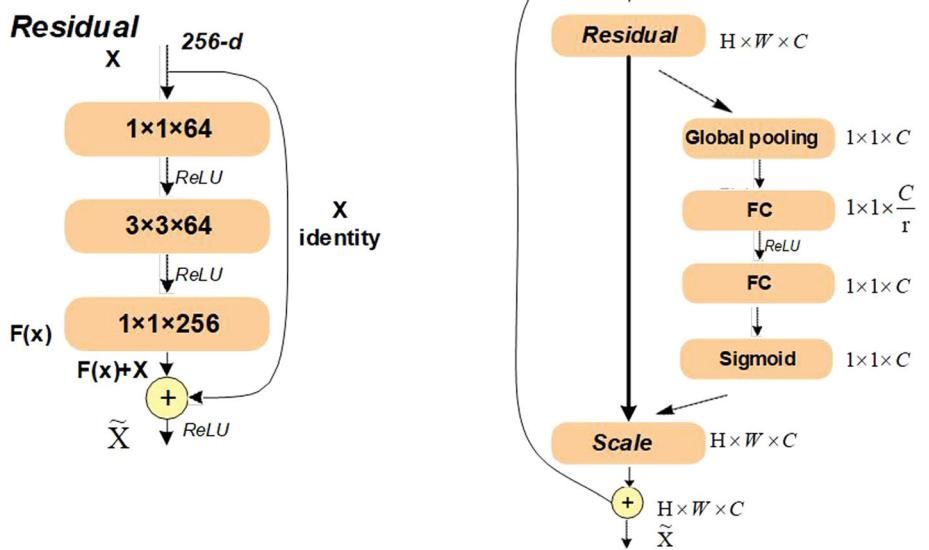
For different network models of layers, the residual block can be divided into two types: building block and bottleneck design. Bottleneck design is shown in Fig. 2b. This module is typically used in ResNet50/101/152 networks with a large number of layers. Bottleneck design connects the input data directly to the third layer through the shortcut connection (the curve part in the figure), which is equivalent to simply performing the equivalent mapping, without generating additional parameters and increasing computational complexity. In order to reduce the number of parameters, the first 1×1 convolutional layer is used to reduce the 256-dimensional data to 64 dimensions, and then at the end by the 1×1 convolution to recover [19].

The basic network model chosen in this paper is ResNet-50, which is mainly composed of three parts: one $7 \times 7 \times 64$ input convolutional layer conv1, one FC layer for classification, and conv2_x, conv3_x, conv4_x, respectively. The convolution module of conv5_x contains $3 + 4 + 6 + 3 = 16$ building blocks, each block is 3 layers, so there are a total of $1 + 16 \times 3 = 49$ convolutional layers [20, 21].

Squeeze-and-excitation networks (SENet) is Hu Jie et al. modeling the channel correlation between convolutional neural network channels, learning feature channel weights, using the “feature recalibration” strategy, selective by learning global information. Enhancements contain features of useful information and suppress useless features, adaptively adjust and enhance the representation of convolutional neural networks. The SE module is independent of the network module of the specific network structure, and can be embedded in the existing CNN network model with only a small increase in computational consumption, thereby improving the network training performance of the network model.

As shown in Fig. 2a, the SE module mainly includes two sub-modules, squeeze and excitation. Squeeze is used to convert local spatial features on each channel into global spatial features, and excitation is used to learn the correlation between feature channels.

Fig. 2 Bottleneck design and se-resnet module



Squeeze uses the global average pooling to obtain the statistics of each channel after obtaining the feature map of each channel. The global spatial features are obtained, and the global spatial features are obtained. The combined compression forms a channel descriptor.

The SE module can be embedded in almost all network structures, including embedded in the module network structure with shortcuts, which can effectively improve the generalization ability of the network. When the SE module is added to the shallower layer of the network structure, the quality of the low-level feature extraction can be enhanced, while for deeper layers, the SE module can improve the generalization of different data sets.

The SE module is embedded in the ResNet network and is called SE-ResNet-50 [22, 23].

3 Data set and approach

The deep learning method is used to detect red jujube. The original image is collected in the actual jujube inspection production line and then the image is pre-processed to obtain a set of jujube defect data. On this basis, a training model is constructed and the training data and label data are entered. Defects of red dates. On this basis, the training model is constructed, and the training data and label data are input into the convolutional neural network model for training. Finally, the network model with better effect is obtained, and the prediction and classification are carried out based on this, so as to realize the automatic detection of jujube defects.

3.1 Datasets and setup

The collection of jujube defects images is as shown in Fig. 3. Fig. 3a is the actual collection of the production line and the working principle of the acquisition, and Fig. 3b, c is the actual image sample collected. In actual production, the jujubes are placed between the blue rollers, and are moved forward with the feeding conveyor belt, and the conveyor belt contains a plurality of channels. Each channel has three color industrial cameras located directly above the conveyor belt. When the jujube moves directly below the industrial camera, the camera captures the image by the position sensor. In order to ensure that the image of the surface of the jujube can be collected completely, the roller rotates in the opposite direction of the advancing direction while moving along with the conveyor belt. Due to the frictional force, the jujube is still spinning while advancing. By adjusting the belt speed and the roller rotation rate, a complete image of the jujube surface can be collected by different cameras on the same channel. Since it is defect detection, if defects are detected in any image, when the corresponding jujubes are sent from the conveyor belt to the corresponding graded air nozzle, the graded air nozzle will spray high-speed airflow of a certain speed to blow it to the baffle and fall to the date collection box to complete the grading process.

3.2 Image preprocessing

After the image is acquired, image preprocessing is performed. In order to establish the jujube data set, the

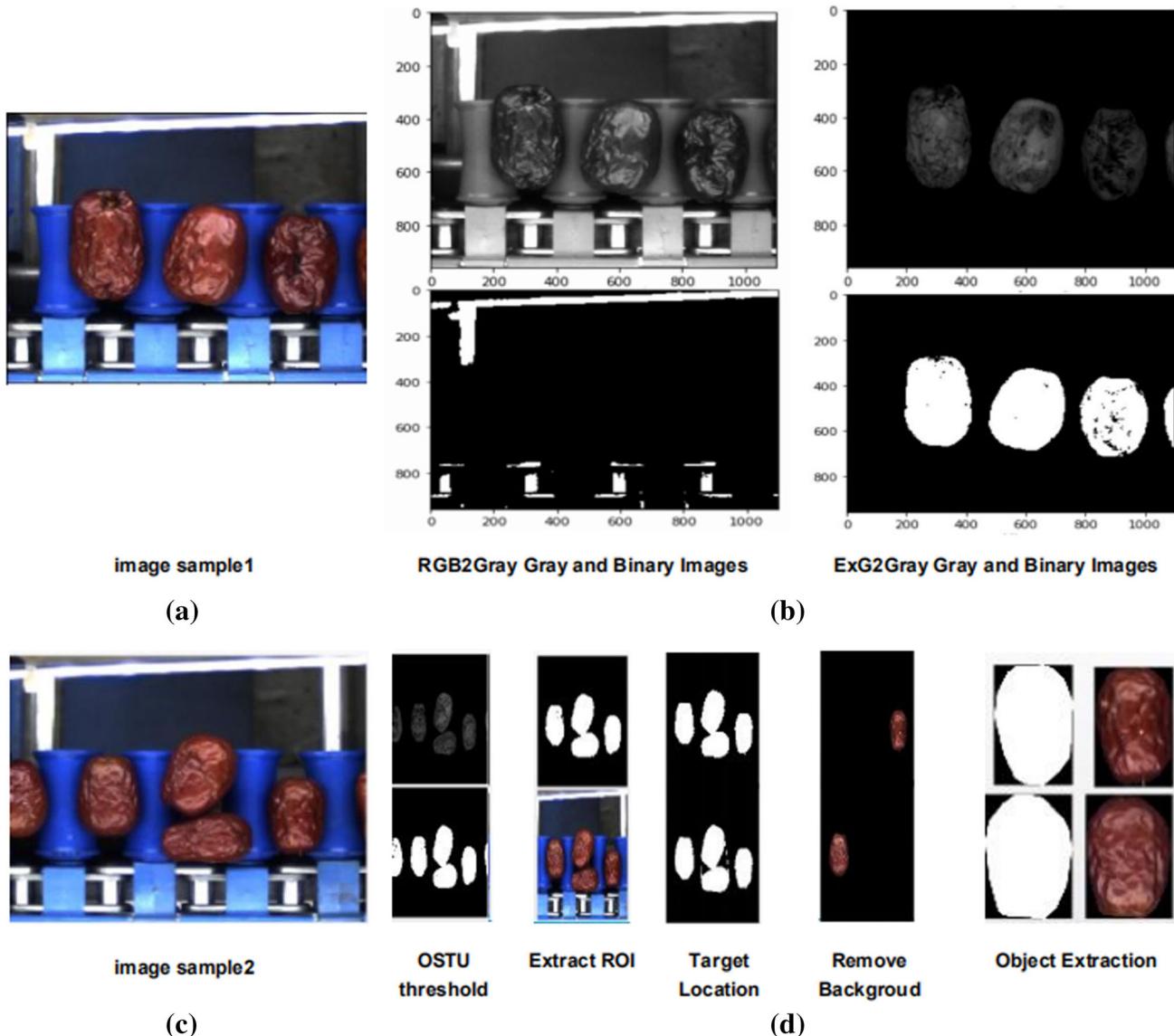


Fig. 3 Image segmentation process

image segmentation is mainly performed, and the jujube object is separated from the complex background.

Graying the collected color image is the first step of image segmentation. Through the analysis of the collected jujube images, it can be seen that the fruit tray is blue, the production line chain is made of stainless steel, the bottom background of the production line is dark blue or black, only the jujube is red, and the color tone will not change even if the light intensity changes. Therefore, if the weight of the red channel can be increased and the contrast with the non-red background can be increased, the purpose of segmenting the image and extracting the red jujube area can be better achieved.

By analyzing the above scene characteristics and verifying by experiment, this paper designs the following grayscale factors based on RGB color space:

$$E \times G = 1.4R - G \quad (1)$$

As shown in Fig. 4, compared with the default grayscale processing method of OpenCV, the gradation factor designed in this paper can effectively filter out the background and glare interference, and accurately segment the jujube contour.

The steps of segmenting the collected jujube images are as shown in Fig. 4c, d. The threshold segmentation method of gray image adopts the OTSU threshold method based on the optimal criterion of the smallest variance within the class after segmentation, and performs dynamic threshold

Operation	Original image	Rotated [-20°, 20°]	Scaled [80%, 120 %]	Brightened [0, 20)	Darkened (-20, 0]	Noise-added	Dynamic-blur-added
Probability		0.80	0.80	0.15	0.15	0.05	0.03
Examples							

Fig. 4 Different operations and corresponding probabilities in jujube simulation based augmentation

segmentation to obtain the corresponding binary image [24–26].

In order to accurately find the interest threshold with complete jujubes, the binarized image is used for vertical projection [27]. When the gray level mean of a column of an image changes, it is reflected in the vertical integral projection value. The vertical projection map is divided into two parts, and the projection minimum value is, respectively, searched as the boundary of the extraction interest threshold, thereby extracting the ROI area.

For extracting the binary image of the ROI region, a morphological closing operation is performed to eliminate the black dot, and then, contour extraction is performed to obtain the position information of the corresponding jujubes. In order to prevent the presence of upper and lower jujube in the collected image, the deformation coefficient U is defined to distinguish:

$$U = \frac{S_{DP}}{S_{Hull}} \times 100\%$$

In this expression, S_{DP} is the area of the outer contour of the target contour, and S_{Hull} is the area of the outer concave polygon of the target contour. If the deformation coefficient value is less than 90%, the target is considered to be the upper and lower jujube, and the corresponding target is discarded and will not be processed.

After determining that the extracted target is a normal jujube, in order to facilitate subsequent processing, for each extracted jujube, the outer section of the outline of the truncated polygon is filled with black to remove the background. In order to facilitate the subsequent processing, in the color image with the background removed, the affine transformation corrects the skewed target, and finally intercepts the target with the smallest outer cut rectangle.

3.3 Simulation based image augmentation method

The establishment of jujube datasets requires priority to cover all types of defects, while also considering the

different sampling conditions, lighting conditions, and even the effects of different varieties, origins, planting environments, and seasons. However, some of these categories have lower probability of occurrence and are more difficult to sample. In order to ensure the richness of training samples, improve the generalization ability of the model, and suppress over-fitting, it is necessary to use data augmentation method to expand the sample size and type [28].

This paper uses a simulation-based data augmentation method (simulation based augmentation, SimAug), based on the prior knowledge of the probability that different samples may occur in the real environment, and assigns the corresponding execution probabilities of different augmentation operations.

As shown in Fig. 4, different sample processing methods are given different execution probabilities based on the experience accumulated in previous studies on jujube production. For example, in the sampling environment, the displacement and rotation that often occur in jujubes will increase the frequency of corresponding operations when expanding. The image noise caused by motion blur and network transmission instability is less likely to occur. The frequency of processing will also be reduced accordingly. There are five types of samples in the jujube data collection in this paper. The data set before and after the data augmentation method is shown in Table 1. Finally, we normalize the size of the sample data, the size is unified to

Table 1 Jujube dataset sample statistics

Categories	Original	Simaug
Healthy	598	3600
Rotted	340	1160
Split	347	1225
Peeling	205	920
Rust	203	880
Samples sum	1693	7785

224×224 , and the data are converted to lmdb format, with sample tags for subsequent deep network training.

3.4 Network model design

The convolutional neural network uses the loss function to evaluate the training model. The loss function is used to estimate the degree of inconsistency between the predicted value and the real value of the model. It is a non-negative real-valued function. Different loss functions have different meanings. The most commonly used loss function is the softmax loss function using cross entropy loss (logloss). The Softmax loss function has good separability, but the Softmax loss function does not limit the features learned between samples of different classes. Therefore, when identifying similar samples, it is easy to be classified into the same category, and the trained model is panned. The ability and robustness are not high. For the classification and classification of jujube defects, under different scenes such as illumination, background and occlusion, it is easy to appear that the intra-class distance is large and the distance between classes is small, which causes the intra-class distance to be larger than the inter-class distance and cause misclassification [29].

In order to reduce or limit the distance between similar samples, and increase the distance between different categories of samples, to achieve fine-grained classification and recognition of images, Florian Schroff et al. used the Triplet loss function instead of the Softmax loss function to map image features to the Euclidean space defines the distance relationship between the homogeneous sample and the heterogeneous sample. The distance between the similar samples is reduced by using the triplet to calculate the distance between the samples, and the heterogeneous sample spacing is increased. The definition is as shown in Eq. (3) [30, 31].

$$L_{TL} = \sum_i^n \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ \quad (3)$$

$f(x_i^a)$, $f(x_i^p)$, and $f(x_i^n)$ are three samples that map to Euclidean Spaces. Through repeated learning, the distance between feature expression of x^a and x^p should be as small as possible, and the distance between feature expression of x^a and x^n should be as large as possible, and there should be a minimum interval α between the two distances.

When the distance between feature expression of x^a and x^n is less than the sum of α and the distance between feature expression of x^a and x^n , the value in square brackets is greater than zero, and a loss is incurred. Otherwise the loss is zero. When the loss is not zero, the entire network is

adjusted by a back propagation algorithm to optimize the feature extraction model.

The method of calculating distance by triples is used to optimize the network, so that the acquired image features can be more fine, and the fine-grained recognition can be completed. However, compared with the softmax loss function, the construction of the triple is very important. Otherwise, the neural network converges slowly and is more likely to over-fitting. Moreover, the method is mainly for balancing data and does not deal with the data imbalance. The Center loss function proposed by Wen Y et al. learns one Center of each type of depth feature, punishes the distance between depth feature and its class center by measuring learning, so as to reduce the difference within the class and effectively increase the difference between classes [33, 34]. Therefore, Center loss is generally used in combination with Softmax, as shown in Eq. (4).

$$L_{CL} = L_S + \lambda L_C \\ = - \sum_{i=1}^m \lg \frac{e^{w_{y_i}^T x_i} + b_{y_i}}{\sum_{j=1}^n e^{w_j^T} + b_j} + \frac{\lambda}{2} \sum_{i=1}^m \|x_i - c_{y_i}\|_2^2 \quad (4)$$

In Eq. (4), $c_{y_i} \in \mathbb{R}^d$ is the depth feature y_i class center. Where parameters Λ used to balance the weight of the two loss functions.

3.5 Deep transfer learning

The purpose of transfer learning is to solve the problem encountered in another environment by acquiring knowledge in one environment, that is, transfer learning can transfer the already trained model parameters to the new model. Due to this ability to transfer learning, more and more scholars are now combining deep learning with transfer learning. Deep learning based transfer learning refers to training the already trained model on the new target again to obtain a CNN model suitable for the current sample.

The fine-tune of deep networks is a common method of deep network transference. Fine-tune uses an already trained network to make adjustments for specific tasks. Fine-tune saves time by not having to train the network from scratch for new tasks [32].

We can also use deep networks to train raw data, rely on the network to extract richer and more expressive features, and then use these features as input to traditional machine learning methods. This avoids complicated manual feature extraction and automatically extracts more expressive features.

The classification of jujube defects can be viewed as the problem of fine-grained image classification (FGIC) [35]. Due to the small difference between defect images and the high signal-to-noise ratio caused by the interference

between similar images, the effective classification of defect images usually requires the help of the sparse and local features in the images.

In view of the task of classification and detection of jujube defects, this paper combines the depth model with transfer learning, first obtains the pre-training model generated by pre-training on the larger sample dataset, and then trains on the jujube defect image samples to make the pre-training model parameters. The parameter distribution of the jujube defect image is fitted. Finally, the parameter distribution is transferred to the jujube defects data set, and the local sparse feature of the image is extracted to complete the model transference.

4 Experiments and results

4.1 Experimental setup

The computer hardware configuration used in this experiment is: Intel E3-1230V2 \times 2 (3.30 GHz) CPU, based on Pascal architecture NVIDIA GeForce GTX1080Ti graphics card, 11 GB memory, 16 GB DDR3 Memory, hard disk 256 GB SSD. In terms of software, the operating system is Ubuntu 16.04 LTS and the programming environment is PyCharm. In the experiment, the algorithm preprocessing and data enhancement are implemented in OpenCV and Python. The CNN training and testing are implemented in the deep learning open source framework Caffe environment.

In the experiment of this paper, the data of the jujube dataset are divided into three parts: training set, verification set and test set, which are used for training, verification and testing of the transfer learning model. In the data set partitioning, if the data set is divided into fixed training set, verification set and test set, the training set image cannot be tested, and the test set image cannot be used to train the network, thus affecting the performance of the model ability. Therefore, in this experiment, the pre-processed sample images were randomly divided into five equal numbers and tested separately. To ensure that the data fits the model's input size, the image is cropped to 224×224 .

On the jujube dataset, we conducted four sets of experiments using ResNet50 and SE-ResNet50 as the basic network: (1) Transfer learning classification based on the classical residual network using Softmax as the objective function (denoted as ResNet50-Softmax and SE-ResNet50-Softmax) (2) The residual network combined with the traditional SVM classification (denoted as ResNet50-SVM and SE-ResNet50-SVM); (3) The residual network combined with the classification of the Triplet loss objective

function (denoted as ResNet50-TL and SE-ResNet50-TL); (4) The residual network is combined with the classification of the Center loss objective function (denoted as ResNet50-CL and SE-ResNet50-CL). The specific experimental process design is as follows:

Pre-training model. Using ResNet-50 and SE-ResNet-50 as the basic network, the ImageNet database is used for pre-training, and the weight parameters of the neural network are initialized to obtain the pre-training model of the transfer learning.

A network model based on the Softmax classification layer. The residual network layer of the target model is initialized using the weight parameters of the pre-training model, and then, fine-tuned training on the jujube data set to obtain the final models ResNet50-Softmax and SE-ResNet50-Softmax, and finally the performance of the model is tested on the test set. Since the classical residual network does not contain a hidden FC layer, the original layer is replaced by a feature layer (2048 dimensions), and a convolution kernel is added as 1×1 , and the output is a 5-dimensional convolution layer.

The network model of the residual network combined with the SVM. The feature layer $1 \times 1 \times 2048$ dimension vector in (2) is extracted as the extraction feature, and the feature vector is normalized according to the L1-norm standard. The feature vector features of all training images were classified and trained by SVM, and the test images were classified by SVM modules obtained by training.

The network model of the residual network combined with the Triplet loss objective function. The softmax layer in the pre-training model is removed, the layer of Triplet loss is added, and fine-tuning training is performed on the jujube data set to obtain models ResNet50-TL and SE-ResNet50-TL.

The residual network combines the network model of the Center Loss objective function. Similar to (4) the models ResNet50-CL and SE-ResNet50-CL are trained.

This experiment uses accuracy, precision, and loss as the evaluation index of the model. The correct rate is divided into Top-1 accuracy (AccTop-1) and Top-5 (AccTop-5). The former predicts the correct number of samples/the total number of samples, and only the class with the highest probability of prediction is correct. The category is judged to be correct for prediction; the latter assumes that the highest five of the predicted label probability values contain the true label, which is the correct classification.

In terms of hyperparameter setting, learning rate and momentum are important parameters in the process of convolutional neural network training. Set the initial learning rate to 0.001, and reduce the learning rate by 1/10 every 3 periods, with a momentum of 0.5.

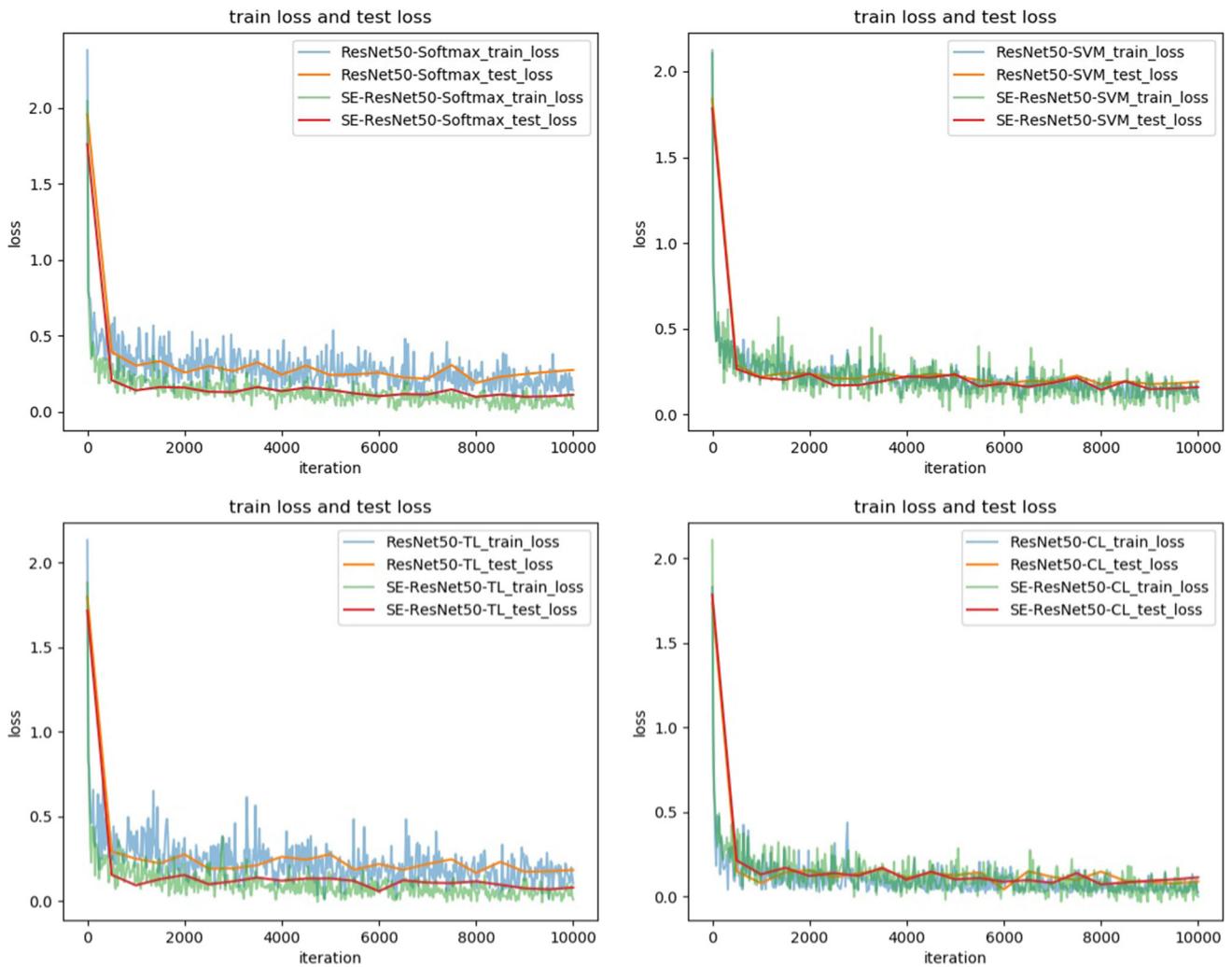


Fig. 5 Comparison of the progression of train loss and validation accuracy through the training period

4.2 Results and evaluation of experiments

Figure 5 shows the downward trend of the loss function of each model on the jujube dataset, where the horizontal axis represents the number of iterations and the vertical axis is the Loss value. The train_loss curve represents the average loss value of an iteration in the training set. The val_loss curve refers to the iteration of an epoch, that is, the average loss value of all the training sets after training and the test on the verification set.

As can be seen from the comparison, it can be seen that as the number of iterations increases, the model gradually converges and the loss value eventually decreases to a fixed range. Except for all models except ResNet50-CL and SE-ResNet50-CL, the network model embedded in the SE module is lower than the original model's Loss value, which indicates that the network embedded in the SE module can better train the deep model. When the ResNet50-CL and SE-ResNet50-CL models are iterated to

1500 iterations, they begin to converge, and no over-fitting occurs. When iterating to 3000 iterations, it finally stabilizes at around 0.1. Although the ResNet50-CL and SE-ResNet50-CL models have little difference in the stability of the Loss value, the convergence rate of the train_loss curve in the ResNet50-CL model is slower than that of the SE-ResNet50-CL model, and the curve oscillation is more obvious.

After the above trained model was tested on the test set, test results such as Table 2 and Fig. 6 were obtained.

As can be seen from Table 2, compared with the original ResNet50-Softmax network, the Top-1 accuracy and average correct rate of the SE-ResNet50-CL network on the jujube dataset increased by 5.89% and 3.63%, respectively. This shows that the application of the powerful classification network built by the SE module to the detection of jujube defects significantly improves the recognition rate of defects. This is mainly because the SE-shortcut structure introduces the original information into the deep layer,

Table 2 Top-1 and Top-5 accuracy of jujube classification on test set across different networks

Models	Top-1(%)	Top-5(%)	Average accuracy
ResNet50-Softmax	82.42	100.00	88.64
SE-ResNet50-Softmax	84.53	100.00	89.56
ResNet50-SVM	84.11	100.00	89.27
SE-ResNet50-SVM	85.00	100.00	90.50
ResNet50-TL	84.56	100.00	91.48
SE-ResNet50-TL	87.14	100.00	93.17
ResNet50-CL	85.21	100.00	92.16
SE-ResNet50-CL	88.31	100.00	94.15

suppresses the degradation of the information, and then pools and expands the receptive field, and fuses the shallow information with the deep information at multiple angles, so that the combined output contains multiple levels of information. Enhanced the ability to express feature maps. The SE module adaptively recalibrates the characteristic response of the channel by explicitly modeling the inter-dependencies between the channels, thereby further improving the generalization capability of the network and enhancing the image recognition performance of the network.

Then we analyze the impact of using different loss functions on the classification accuracy of the network

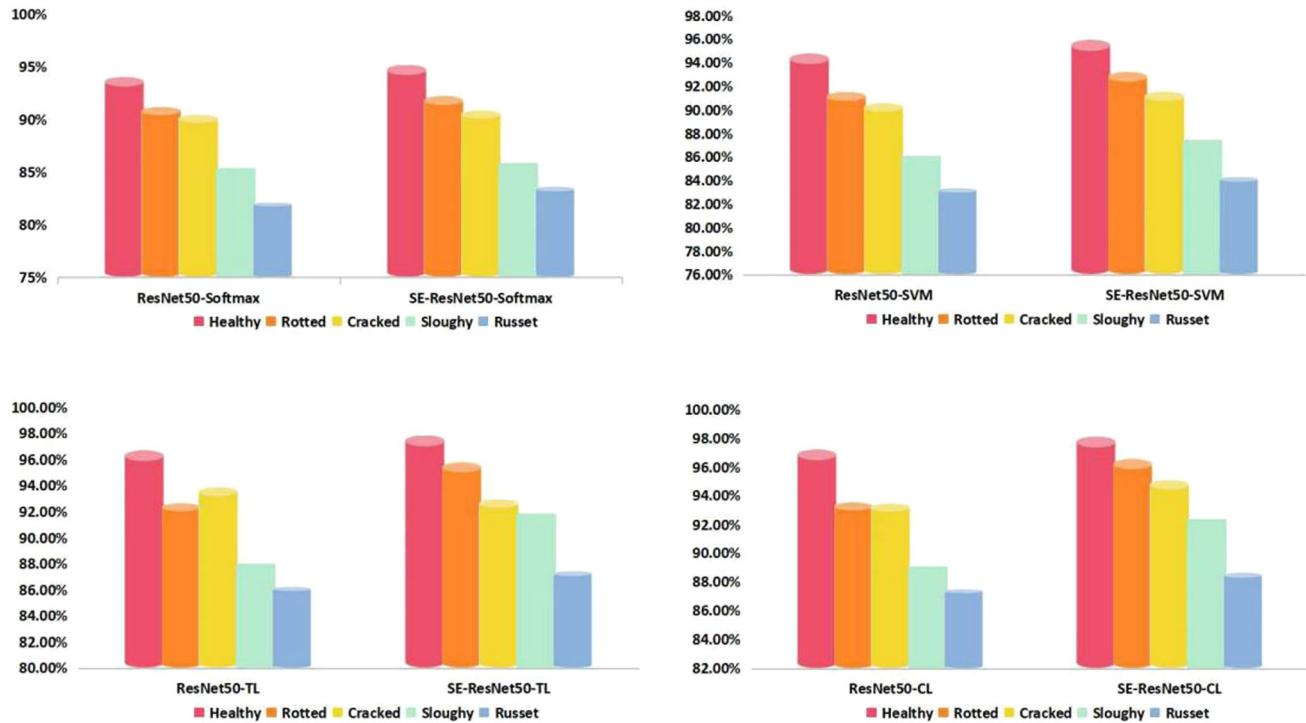
model. From the results, the classification performance is the lowest when using the traditional Softmax method. The network model using the Triplet loss and Center loss methods is better than the model using the SVM method.

Compared with the Triplet loss method, the accuracy of the Center loss method is improved, and the method is simple to calculate. It avoids the selection of complex training data input pairs in the model training of the Triplet loss method, and improves the discriminating ability of the model output features.

The experimental results show that the Center loss method can better meet the high accuracy and real-time requirements of the actual jujube defect detection and quality sorting.

In order to further analyze, the performance of the Center Loss method, the ResNet50-CL and SE-ResNet50-CL models were tested on the test samples using the confusion matrix, and the results were described and analyzed. The confusion matrix is a visual classification effect diagram, which can be used to describe the relationship between the real category attribute of the sample data and the recognition result [36].

It can be seen from Fig. 7 that the classification accuracy of different defects of the test samples of the jujube data set and the types of defects and false positives that are misjudged. Observing the defect categories that are easy to misjudge, we can find that: (1) Whether the ResNet50-CL model or the SE-ResNet50-CL model, the probability of

**Fig. 6** Comparison of classification accuracy of different models

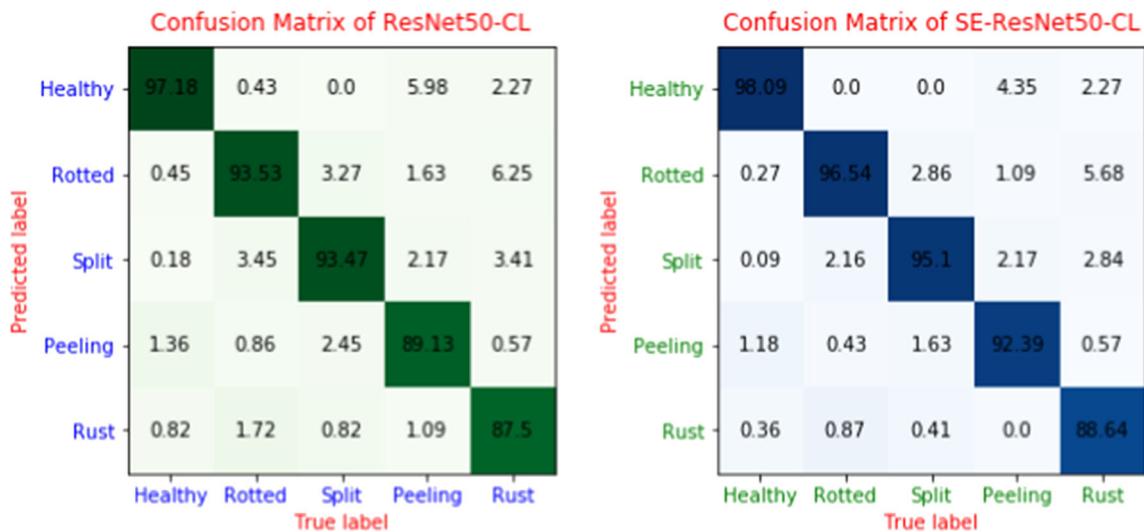


Fig. 7 The confusion matrix of center loss method

the healthy class being misjudged is very low, and it may be misjudged as the peeling class; The aspect is because the number of healthy samples is the largest, the CNN model extracts such sample features adequately, and on the other hand, it may be that the healthy class samples are less similar to other categories. (2) The rust class has the highest probability of being misjudged, and is mainly misjudged as the rotted class and the split class; this aspect is because the sample distribution is uneven, and the Russet defect sample is relatively small; On the other hand, there is a great similarity between different defects. This is mainly due to the fact that the SE shortcut structure inserts the initial information into the deep layer, suppresses the information degradation and then expands the receptive field by aggregating and merging the shallow information with the deep information from many angles so that the combined output contains multiple levels of information. Enhance the expressiveness of feature maps.

In order to explain the classification results of the model to the sample, the features learned by the deeper convolutional layer in the CNN model are visualized. Figure 8 uses the Grad-CAM visualization method to visualize the output of the last convolutional layer of the SE-ResNet50-CL model. It visually shows which areas play a crucial role in identifying the classification. Grad-CAM is improved based on the CAM (Class Activation Map) method. Grad-CAM uses the global average of the gradient to calculate the weight of the feature map. After obtaining the weight of the category for all the feature maps, the weighted sum is obtained. A thermogram can be obtained [37, 38]. In Fig. 7, the Grad-CAM visualization shows the importance of the different features extracted when class discrimination. The closer the color is to the red feature, the more important the color is. The closer the color is to the blue

feature, the less important the classification result is. The Guided Backprop visualization can get all the features that work for the classification, highlighting the fine-grained details, but not the importance of the different features. Guided Grad-CAM is a visualization that combines the two to create a level of detail that combines fine-grained detail features with different features.

From the results in Fig. 8, it can be seen that except for the healthy sample, the areas in the other visualizations that play a more significant role in predicting the classification of the model are the areas where the jujube defects are located, which proves that the model does extract the corresponding features from the image. From the visualization results, some reasons for misclassification of categories can also be analyzed: the surface of the red sample of the healthy sample is smooth under the illumination of the external light source, and the peeling sample also exhibits a highlight characteristic due to the damage of the red jujube skin. These two types of features are similar, resulting in misclassification; similarly, the features extracted by the rotted sample and the split sample are similar, and may also cause misclassification; for the rust sample, the area occupied by the heat map area is significantly smaller than the actual fruit rust area, showing only the fruit rust in the right half area, while the fruit rust in the left half area has less effect on the classification. The main reason for this phenomenon may be: in the Grad-CAM algorithm, the heat map is generated by features from the upper layers of the network. This feature preserves high semantic features but loses part of the spatial information. This makes the final visual result have a certain error in space. Of course, probably because there are too few training samples in the rust category, the trained model

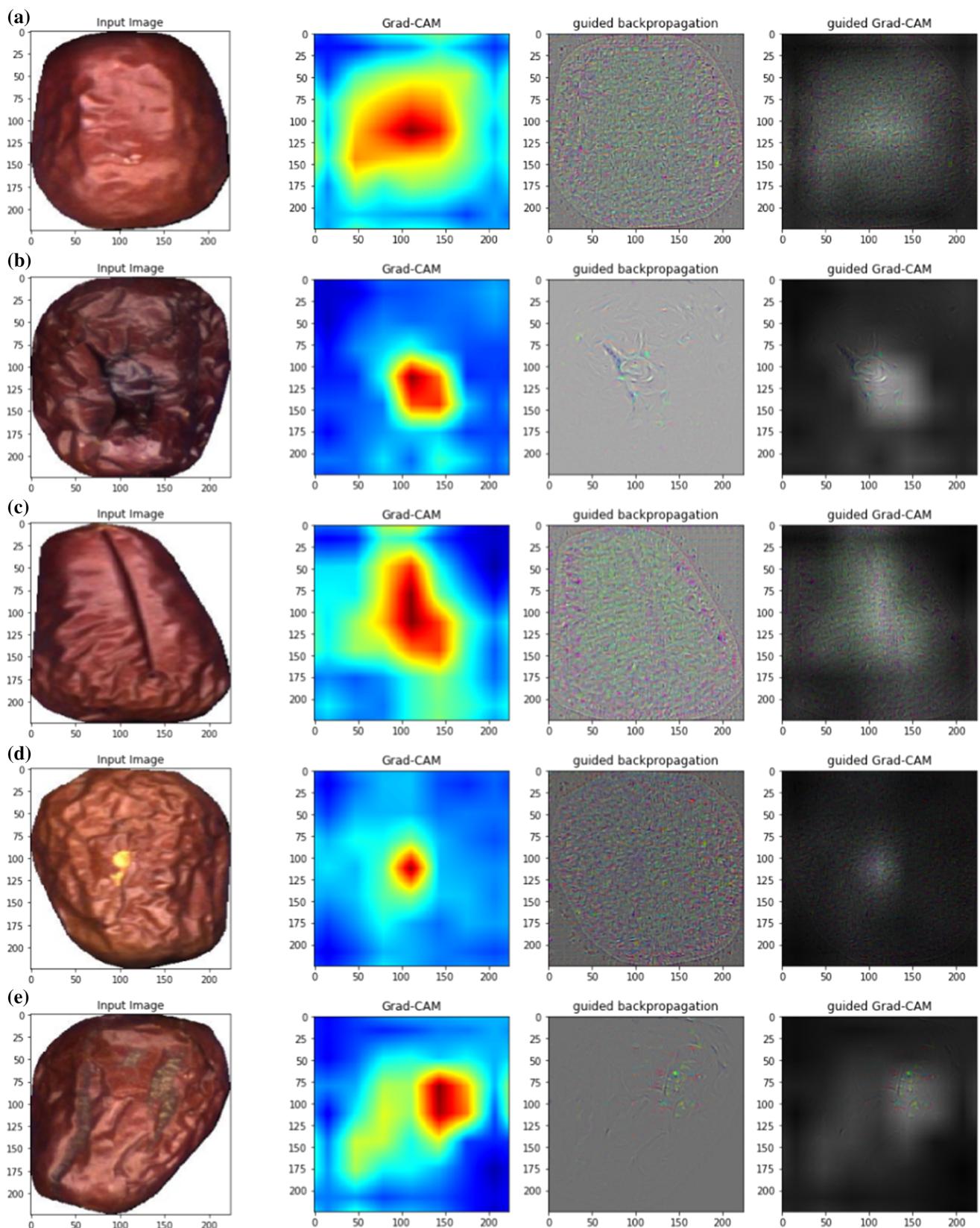


Fig. 8 Heatmap of SE-ResNet50-CL.27. Figure **a**, **b**, **c**, **d**, and **e**, respectively, present healthy, rotted, split, peeling, and rust samples

does not fully extract the features used to identify the rust defects.

5 Conclusions and future work

In this article, oriented to the classification requirements of dry red date defect detection, a small data set detection model for red date classification based on the convergent neural network and migration learning is proposed. The original images collected from the actual jujube production line were pre-processed and a small data set containing five categories of jujube defects was created. Triplet loss function and Center loss function were used to replace softmax loss function and embed SE module, and the SE-ResNet50-TL and SE-ResNet50-CL models were designed. The experiment shows that the SE-ResNet50-CL model optimizes the fine-grained classification problem of jujube defects identification, and the test accuracy reaches 94.15%. The model has good stability and high identification accuracy.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No.61771352), the Science and Technology Innovation Team Project of Hubei Province (Grand No. T201838).

Compliance with ethical standards

Conflict of interest These no potential competing interests in our paper. And all authors have seen the manuscript and approved to submit to your journal. We confirm that the content of the manuscript has not been published or submitted for publication elsewhere.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Wang K, Nakano S, Ohashi S (2011) Detection of external insect infestations in jujube fruit using hyperspectral reflectance imaging. *Biosys Eng* 108:345–351
- Wang K, Nakano S, Ohashi S (2010) Nondestructive evaluation of jujube quality by visible and near-infrared spectroscopy. *LWT - Food Sci Technol* 44(4):1119–1125
- Wu L, He J, Liu G, Wang S, He X (2016) Detection of common defects on jujube using Vis-NIR and NIR hyperspectral imaging. *Postharvest Biol Technol* 112:134–142
- Lee D, Schoenberger R, Archibald J (2008) Development of a machine vision system for automatic date grading using digital reflective near-infrared imaging. *J Food Eng* 86:388–398
- Li Y, Zhang Q, Chen H (2016) Detection of diseases and cracks of semi-dried dates based on machine vision. *J Agric Mech Res* 38:120–125
- Zhao J (2008) Recognition of defect chinese dates by machine vision and support vector machine. *Trans Chin Soc Agric Mach* 39(3):113–117
- Zhang LT, San-Min S (2016) Study on appearance quality classification based on detection of the stripes of red jujube in Southern Xinjiang. *Acta Agric Zhejiangensis* 28:1089–1093
- Dastjerdi AV, Buyya R (2016) Fog computing: helping the internet of things realize its potential. *Computer* 49(8):112–116
- Zhou Y (2017) Classification and recognition approaches of tomato main organs based on DCNN. *Trans Chin Soc Agric Eng* 33(15):219–226
- Yang G, Xu N, Kang L, Gong M, Hong Z (2018) Identification of navel orange lesions leaves based on parametric exponential non-linear residual neural network. *Acta Agric Zhejiangensis* 30(6):1073–1081
- Mohanty P, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci* 7:1419
- Charles T (2017) Photogrammetric computer vision: statistics, geometry, orientation and reconstruction. *Photogram Eng Remote Sens* 83(10):661–662
- Hu J, Lu J, Tan YP (2015) Deep transfer metric learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 325–333
- Zhuang FZ, Luo P, He Q, Shi ZZ (2015) Survey on transfer learning research. *Ruan Jian Xue Bao/J Softw* 26(1):26–39
- Rodionov S, Potapov A, Latapie H, Fenoglio E, Peterson M (2018) Improving deep models of person re-identification for cross-dataset usage. *Artif Intell Appl Innovations*. https://doi.org/10.1007/978-3-319-92007-8_7
- Luo Y, Wen Y, Duan L-Y, Tao D (2018) Transfer metric learning: Algorithms, applications and outlooks. *arXiv preprint arXiv:1810.03944*
- Wei X, Chen Y, Su J (2018) Domain adaptation via identical distribution across models and tasks. *Lecture Notes Comput Sci*. https://doi.org/10.1007/978-3-030-04167-0_21
- Huang Z, Zhu Q (2016) Detection of red region of Fuji Apple based on RGB color model. *Laser Optoelectron Progress* 53(4):041001
- Peiyong C (2009) A novel OSTU segmentation algorithm for image threshold. *Comput Appl Softw* 2009(5):227–232
- Zheng Z, Ma Y, Zheng H, Ju J, Lin M (2018) UGC: real-time, ultra-robust feature correspondence via unilateral grid-based clustering. *IEEE Access* 6:55501–55508
- Sun M, Si J, Zhang S (2007) Research on embedding and extracting methods for digital watermarks applied to QR code images. *NZ J Agric Res* 50(5):861–867
- Chen Y, Lin Z, Zhao X, Wang G, Gu Y (2014) Deep learning-based classification of hyperspectral data. *IEEE J Sel Top Appl Earth Observ Remote Sens* 7(6):2094–2107
- He K (2016) Deep residual learning for image recognition. *IEEE conference on computer vision & pattern recognition*.
- Perkins C, Moideen AN, Ahuja S (2017) Return to activity and sports following posterior correction and fusion for adolescent idiopathic scoliosis. *The Spine J* 17(11):S329–S330
- Ma W (2019) Achieving super-resolution remote sensing images via the wavelet transform combined with the recursive res-net. *IEEE Trans Geosci Remote Sens* 57(6):3512–3527
- Chen H (2017) VoxResNet: deep voxelwise residual networks for brain segmentation from 3D MR images. *Neuroimage* 170:S1053811917303348

27. Jie H (2017) Squeeze-and-excitation networks. *IEEE Trans Pattern Anal Mach Intell*, pp 99
28. Zhou FY, Jin LP, Dong J (2017) Review of convolutional neural network. *Chin J Comput* 40:1229–1251
29. Liu W (2016) Large-margin softmax loss for convolutional neural networks
30. Schroff F, Kalenichenko D, Philbin J (2015) Face net: a unified embedding for face recognition and clustering. In: IEEE conference on computer vision and pattern recognition
31. Zhu F (2017) Part-based deep hashing for large-scale person re-identification. *IEEE Trans Image Process* 26(10):4806–4817
32. Lei H (2018) A deeply supervised residual network for HEp-2 cell classification via cross-modal transfer learning. *Pattern Recognit*, 79.
33. Wen Y, Zhang K, Li Z (2016) A discriminative feature learning approach for deep face recognition. In: ECCV
34. He X (2018) Triplet-center loss for multi-view 3D object retrieval. In: IEEE/CVF conference on computer vision and pattern recognition (CVPR).
35. Zhang L (2015) Detecting densely distributed graph patterns for fine-grained image categorization. *IEEE Trans Image Process* 25(2):553–565
36. Zhuang FZ, Luo P, He Q, Shi ZZ (2015) Survey on transfer learning research. *Ruan Jian Xue Bao/J Softw* 26(1):26–39
37. Zhou B, Khosla A, Lapedriza A, Oliva (2016) Learning deep features for discriminative localization. In: IEEE conference on computer vision and pattern recognition, pp.2921–2929.
38. Selvaraju RR (2016) Grad-CAM: visual explanations from deep networks via gradient-based localization. In: International conference on computer vision and pattern recognitions

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.