Identification of internal cracks in corn seed using convolutional neural networks

Ziliang Huang[1,2,3], *Rujing* Wang[1,3], *Liusan* Wang[1,3,*], *Yue* Teng[1,2,3], and *Shijian* Zheng[3,4]

[1]Institute of Intelligent Machines, Hefei Institute of Physical Science, Chinese Academy Sciences, Hefei 230031, China
[2]University of Science and Technology of China, Hefei 230026, China
[3]Intelligent Agriculture Engineering Laboratory of Anhui Province, Hefei 230031, China
[4]Southwest University of Science and Technology, Mianyang 621010, China

Abstract. The identification of seed quality is very important for which the quality of seed is crucial to the yield and quality of crops. There are two main problems with the acquisition and identification of cracks inside corn seed. One is that most of the methods of near-infrared spectroscopy or X-ray are used to obtain images of cracks inside the seed, the acquisition equipment is expensive and the operation is complicated. The other is the identification of crack images, and the traditional image processing method is usually used which requires professionals to design different model parameters each time, resulting in poor model robustness and low model accuracy. In this study, we originally proposed a simple but effective method to obtain the picture of corn seed internal cracks, which is combined with visible light transmission and ordinary camera acquisition method. We also proposed using the transfer learning methods not only solving the problem of the small scale of our corn seed internal cracks dataset but also avoiding extracting features manually. Our proposed method achieved a promising result, which is able to correctly identify the cracked and intact corn seed 100% in our training stage and testing stage.

Keywords: Corn seed, Internal cracks, Convolutional neural networks, Transfer learning.

1 Introduction

Corn is the most important food crop in the world and one of the main commodities in global food trade. In order to improve the quality of corn seed, it is necessary to strengthen the identification of corn seed quality. With the improvement of agricultural mechanization, mechanized agricultural tools are mainly used for corn harvesting, and mechanized operations often cause damage to corn seed. Corn with cracks is easy to be broken in the process of transportation, easy to be mildew in the process of storage, and low germination

[*]Corresponding author: lswang@iim.ac.cn

rate in the process of planting, which ultimately affects the yield (D Patrício and R Rieder, 2018; C Ni et al., 2019; N Kussul et al., 2017).

At present, China lacks convenient and fast recognition methods in corn crack recognition. It mainly relies on manual judgment or traditional image processing methods. Traditional image processing methods have poor repeatability, weak resistance to complex environmental variables in the detection process, and high requirements for the position and direction of the identified object. In order to solve the problems, this study focuses on corn crack recognition technology, and proposes a corn seed internal crack recognition method based on convolutional neural networks, making full use of the powerful image feature extraction ability and the learning ability of convolutional neural networks, which are fast and accurate to determine whether there are cracks in the corn.

2 Material and methods

2.1 Convolutional neural network

Convolutional neural network is a kind of feed-forward neural network that includes convolution calculation and has a deep structure. A basic convolutional neural network consists three major parts, such as convolution operation, activation operation and pooling operation. The essence of convolution is the process of realizing two-dimensional spatial filtering, and the output after convolution is called feature map, which is the information carrier circulated in the neural network.

2.2 Batch normalization

Batch normalization is a normalization method proposed by (S Ioffe and C Szegedy, 2015). Ioffe et al. believe that in deep learning, if the data is not normalized, the model training speed will be slow or even not converged due to the scattered sample feature distribution. Further experimental studies by Santurkar et al. (S Santurkar et al., 2018) show that the Batch Normalization changes the Lipschitz properties of the loss function through re-parameterization, which is to make the gradient of the loss function more Lipschitz.

2.3 Transfer learning

For computers, transfer learning is a technology that allows existing model algorithms to be slightly adjusted to be applied to a new domain and function. For deep learning models, it is the training of different data sets by using the weight matrix trained on other data sets after fine-tuning. Using transfer learning solves the problem of the low training performance of the model caused by the small amount of data in the target domain, and also relieves the problem of the poor generalization ability of the existing deep learning model in the case of different training sets.

3 Experiment and analysis

3.1 Experimental setup and dataset

Our experiments were implemented using an Intel Core i9-9900k CPU with a single NVIDIA Titan RTX GPU in Ubuntu 16.04 operation system. All experiments were using Keras deep learning framework and choosing Tensorflow as the backend.

We divided our dataset into three parts as training set, validation set and testing set in the performance calculation of each model we set. The detail of the dataset was listed in table 1.

Regardless of training from scratch or using transfer learning, the settings during the training process remained the same. In order to accelerate the training process and reduce the amount of computation, all images were normalized when the data set is input into the model and all picture sizes were set to 224*224 pixels. The batch size during training was set to 16, the optimizer used SGD, the learning rate was set to 2.5e-3, the decay was set to 3e-4, and the momentum was set to 0.9. In order to avoid over-fitting, the early stopping mechanism was adopted, and the patience was set to 10. The models used in our experiments were shown in table 2.

Table 1. The detail of our dataset.

Training set		**Validation set**		**Testing set**	
Intact	Cracked	Intact	Cracked	Intact	Cracked
602	592	173	170	86	85

Table 2. The CNN models with properties.

Model	**Input Size**	**Parameters(M)**	**Depth**
InceptionV3	224-by-224	21.8	47
ResNet50	224-by-224	23.6	50
ResNet101	224-by-224	42.6	101
Inception-ResNetV2	224-by-224	54.3	58
Xception	224-by-224	22.8	36
DenseNet121	224-by-224	7.04	121

3.2 Exprimental results

As mentioned above, the transfer learning weight adopted in this experiment was the weight matrix trained on the ImageNet data set, and then the final model was obtained through fine-tuning training on the data set. We chose several non-pre-trained original models, namely InceptionV3 (C Szegedy et al., 2016), ResNet50 (K M He et al., 2016), ResNet101 (K M He, et al., 2016), Inception-ResNetV2 (C Szegedy et al., 2016), Xception (F Chollet, 2017), DenseNet121 (G Huang et al., 2017).

Firstly, the training results of the original model were presented, and the training accuracy and loss curves were shown in figure 1.

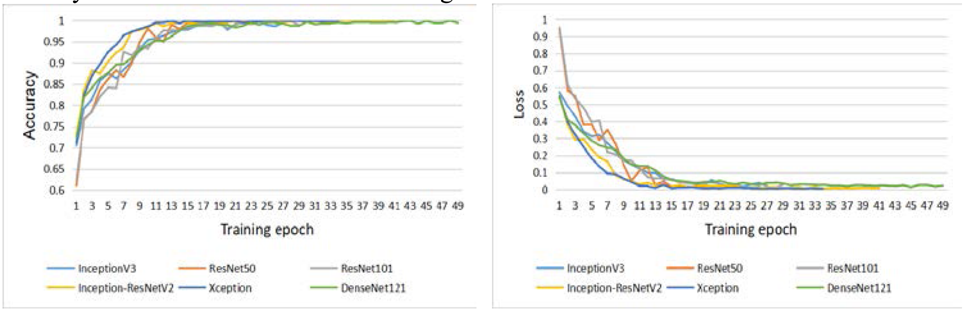

Fig. 1. Model accuracy and loss curves.

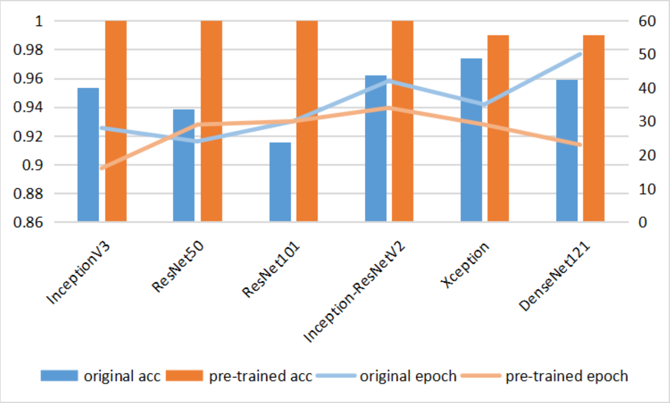

Fig. 2. Comparison of the pre-trained models and the original models.

We pre-trained the selected model and trained it again on our dataset. This method is called fine-tuning, which is mainly to retrain the final classification layer, softmax layer and fully connected layer. Since our research focused on the presence or absence of corn cracks, it was only divided into two categories. The comparison of the training results between the pre-trained model and the original model was shown in figure 2. As can be seen from figure, 2 compared with the original model, the training accuracy of the pre-trained model was improved by 2.4 percentage points to 8.5 percentage points, with an average increase of 4.6 percentage points. In terms of convergence speed, the training epoch of the pre-trained model was smaller than that of the original model except Resnet50.

The pre-trained model reduced the overall training time of the model. In order to study the speed of each epoch in the training process of the model, the time spent by each epoch in the training process of the pre-trained model and the original model was also shown in figure 3. It can be found that all the other pre-trained models except Resnet50 spent more time on each epoch than the original model. In summary, using the pre-training model will increase the time spent in training each epoch, but the used of the pre-trained method reduced the total number of epochs for training, and reduced the training time in terms of overall time.

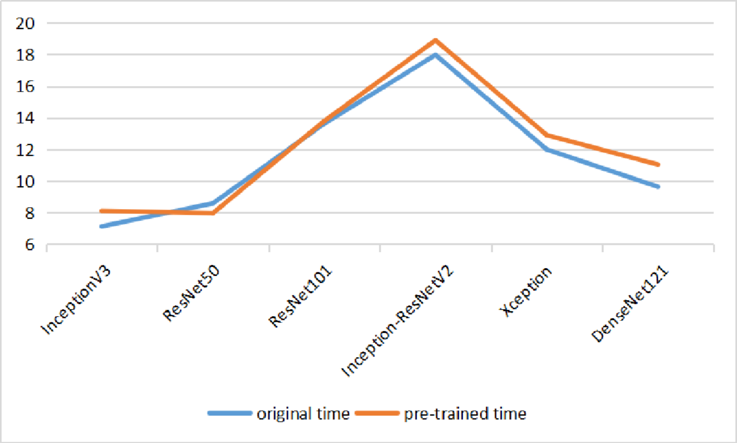

Fig. 3. Comparison of the time spent in each epoch between the pre-trained models and the original models.

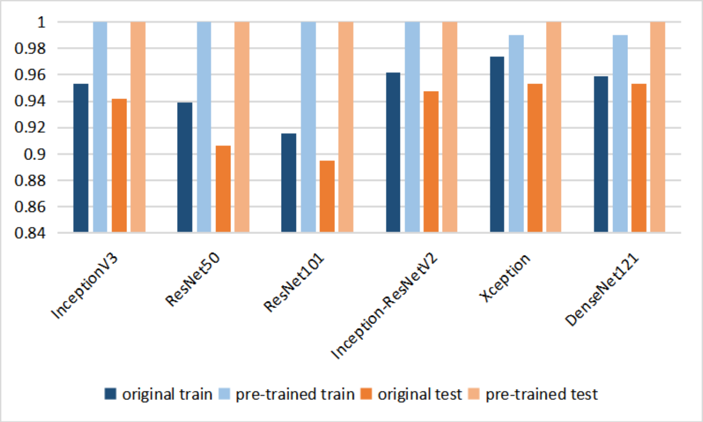

Fig. 4. Comparison of training and test results between pre-trained models and original models.

Table 3. Details of experimental results.

Model Accuracy	Original train	Pre-trained train	Increase (%)	Original test	Pre-trained test	Increase (%)
InceptionV3	0.953	1.0	4.7	0.941	1.0	5.9
ResNet50	0.938	1.0	6.2	0.906	1.0	9.4
ResNet101	0.915	1.0	8.5	0.894	1.0	10.6
Inception-ResNetV2	0.962	1.0	3.8	0.947	1.0	5.3
Xception	0.973	0.997	2.4	0.953	1.0	4.7
DenseNet121	0.959	0.996	3.7	0.953	1.0	4.7

After the model finished training, we experimented the model on the testing set, the results of training set and testing set of the models were obtained in figure 4 and table 3. It can be seen that the accuracy of all the original models in the training set exceeded 90%, among which Xception reached 97.3% with the highest accuracy. The accuracy of all pre-trained models was close to 100%, which was an improvement of 2.4 percentage points to 8.5 percentage points over the original model. In terms of accuracy on the testing set, the lowest and highest accuracy of the original model was 89.4% and 95.3%, which was 4.7 percentage points to 10.6 percentage points different from that of the pre-trained model. From the experimental results, it can be concluded that the pre-trained model had greatly improved the model accuracy, and the model performance from the training stage to the test stage remained excellent.

4 Conclusion

In this study, we first proposed to adopt a convenient, fast and low-cost method for obtaining internal cracks in corn seed based on visible light transmission which is easy to implement in both the laboratory environment and the actual application environment. We also introduced the method of convolutional neural network to identify the crack of corn seed, and used the powerful automatic feature extraction ability of convolutional neural network to learn the corresponding features from the samples and avoided the complex feature extraction process. At the same time, in order to address the problem of the small scale of our data set, we adopted the method of pre-trained model to improve the recognition accuracy and convergence speed of the model. From the experimental results, we can find that using the pre-trained model is beneficial for the model training in this experiment. The parameters learned on the large-scale dataset through the pre-trained model were used as the initialization parameters of the original model, which improved the initial performance of the original model and at the same time increased the convergence speed when training the original model. Finally, the model obtained by training is more integrated and robust.

In the future, we will continue to use convolutional neural networks combined with seed internal and external quality and other related work to carry out our research, and strive to apply relevant theoretical research work to practical application.

This work was supported by National Key Research and Development Program of China (2018YFD0101004) and Major Scientific and Technological Innovation Projects in Shandong Province (2019JZZY010730).

References

1. Patrício D and Rieder R. 2018. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Comput. Electron. Agric. 153: 69-81. doi:10.1016/j.compag.2018.08.001.
2. Ni C, Wang D Y, Vinson R, Holmes M and Tao Y. 2019. Automatic inspection machine for maize kernels based on deep convolutional neural networks. Biosyst. Eng. 178: 131-144. doi:10.1016/j.biosystemseng.2018.11.010.
3. Kussul N, Lavreniuk M, Skakun S and Shelestov A. 2017. Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. IEEE Geoscience and Remote Sensing Letters PP: 1-5. doi:10.1109/LGRS.2017.2681128.
4. Ioffe S and Szegedy C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.
5. Santurkar S, Tsipras D, Ilyas A and Madry A. 2018. How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift)in arXiv: 1805.11604.
6. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z and Ieee. 2016. Rethinking the Inception Architecture for Computer Vision. 2016 Ieee Conference on Computer Vision and Pattern Recognition. Ieee, New York. p. 2818-2826.
7. He K M, Zhang X Y, Ren S Q, Sun J and Ieee. 2016. Deep Residual Learning for Image Recognition. 2016 Ieee Conference on Computer Vision and Pattern Recognition. Ieee, New York. p. 770-778.
8. Szegedy C, Ioffe S, Vanhoucke V and Alemi A. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. in arXiv:1602.07261.
9. Chollet F. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. 30th Ieee Conference on Computer Vision and Pattern Recognition. Ieee, New York. p. 1800-1807.
10. Huang G, Liu Z, van der Maaten L and Weinberger K. 2017. Densely Connected Convolutional Networks. in arXiv:1608.06993.