

# Image Recognition Method Based on an Improved Convolutional Neural Network to Detect Impurities in Wheat

Yin Shen<sup>1,3</sup>, Yanxin Yin<sup>2</sup>, Chunjiang Zhao<sup>1,3</sup>, Bin Li<sup>2,3</sup>, Jun Wang<sup>4</sup>, Guanglin Li<sup>1</sup> and Ziqiang Zhang<sup>5</sup>

<sup>1</sup> College of Engineering and Technology, Southwest University, Chongqing, 400715 China

<sup>2</sup> National Research Center of Intelligent Equipment for Agriculture, Beijing 100097 China.

<sup>3</sup> Beijing Research Center for Information Technology in Agriculture, Beijing, 100097 China

<sup>4</sup> School of Electronics and Information Engineering, Sichuan University, Chengdu, 610064 China.

<sup>5</sup> Information Engineering College, Capital Normal University, Beijing 100048 China

Corresponding author: Chunjiang Zhao (e-mail: zhaojc@nercita.org.cn) and Bin Li (e-mail: lib@nercita.org.cn)

This work was supported by the National Key Research and Development Project of China (NO.2016YFD0702000).

**ABSTRACT** Impurities in wheat seriously affect wheat quality and food security. They are mainly produced during the operational process of combine harvesters. To solve the recognition and classification problems associated with impurities in wheat, a recognition method using an improved convolutional neural network is proposed in this article. A labeled dataset of normal wheat and five impurities is constructed, using which the Wiener filtering algorithm and the multi-scale Retinex enhancement algorithm are employed for image preprocessing. Based on network research using Inception\_v3, improvement and optimization are undertaken before designing the WheNet convolutional neural network, which is intended for automatic recognition of wheat images. Under the same conditions, comparative experiments using the WheNet, ResNet\_101 and Inception\_v3 networks are conducted. Indexes such as receiver operating characteristic, area under curve (AUC), and recall rate are adopted to evaluate the experimental outcomes. Experimental results indicate that the WheNet network achieved the most efficient results. It also shows a shorter training time, and its recognition accuracies for Top\_1 and Top\_5 of the test set are 98.59% and 99.98%, respectively. The mean values of both the AUC and recall rate of the network on the recognition of various images of impurities are higher than those of the ResNet\_101 and Inception\_v3 networks. Consequently, the WheNet network can be a useful tool in recognizing impurities in wheat. Furthermore, this method can be used to detect impurities in other fields.

**INDEX TERMS** Wheat images, impurity recognition, convolution neural network, WheNet network

## I. INTRODUCTION

Impurities in wheat are not only an important factor affecting the total grain output, but also an important index for measuring the harvesting quality of combine harvesters. As these impurities can seriously reduce the grain quality, recognition and testing of impurities in wheat is crucial.

Although researchers across the world have conducted recognition and monitoring of impurities in rice and maize by adopting manual means, the efficiencies are observed to be very low, the main reasons being that manual recognition is time-consuming, needs considerable efforts, and has certain subjectivity, which is prone to mistakes. In particular, when the backgrounds of images are complicated and the targets

themselves undergo dynamic changes, the difficulty in recognizing targets increases. Some researchers have identified impurities in rice and in cotton using image processing, hyperspectral imaging, and machine vision techniques [1]-[3], which have helped improve the accuracy and efficiency of impurity recognition. For example, Wallays et al. [4] conducted an online test of impurities in rice when using combine harvesters by adopting hyperspectral imaging, and realized classification of wheat and its impurities in the images. Mahirah et al. [5] adopted a visual system containing dual light sources to test for impurities and broken rice grains. However, their system is complex. Shatin et al. [6] used high light spectroscopy to detect impurities in Canadian wheat samples with an accuracy

rate of 90%. Singh et al. [7] used information from a 700-1100 nm hyperspectral band to classify normal wheat and mycelium-damaged wheat, and achieved good results. In most cases, traditional technology is characterized by acquiring features of targets such as color, shape, and graining before classifying the images through artificial neural network and support vector machine [8], [9]. Most of these methods require the production of samples, and the system structure is complex. Moreover, the methods are time-consuming, strenuous, and have poor recognition ability.

Currently, given their importance in the field of deep learning [10]-[13], convolutional neural networks (CNNs) have made great progress in aspects such as image recognition [14], [15], pest recognition [16], and face recognition [17], [18]. CNNs are being applied increasingly to agricultural production. For example, Krizhevsky et al. [19] achieved a breakthrough in the recognition of the dataset of ImageNet using a CNN. Peisen et al. [20] recognized the types of chrysanthemums using a CNN and succeeded in constructing a recognition system for the flower. Amara et al. [21] realized the classification of blades of banana trees infected with diseases using small samples and by adopting a CNN; however, the method provided poor image recognition performance for large samples. Using a CNN, Kun et al. [22] conducted image recognition of agricultural machinery, and realized the positioning and recognition of five types of common agricultural equipment. Using GoogLeNet CNN, Shuangping et al. [23] successfully realized accurate testing of seasonal febrile diseases of rice ears. Mohanty et al. [24] conducted the recognition of many images of crop diseases using a CNN, with accuracies reaching as high as 99.35%. Ferreira et al. [25] made use of the SLIC superpixel algorithm to generate superpixel images and conduct recognition processing using a CNN, and the resulting classification accuracy exceeded 98%.

These studies characterize CNNs as being able to independently learn images and provide excellent object recognition performance. Therefore, CNNs have become one of the most important methods of image classification and recognition. At present, CNNs are being studied with regard to damage caused by pests, fruit recognition, and plant recognition [26]-[29]. However, only a few studies have focused on the recognition of grain impurities, especially with regard to impurities in wheat. This is because the wheat impurity image distribution structure is complex. Although existing CNNs can achieve image feature extraction, the network structure is complex, the number of parameters is large, the training time is

long, and the recognition accuracy cannot be guaranteed [35], [36]. In addition, image recognition technology based on deep learning requires a training dataset that contains a large number of annotated images. Current open-source datasets, such as cifar10, cifar100, and ImageNet, provide data regarding general objects, such as pests, fruit, faces, and flowers. However, none of the open-source datasets recognize impurities in wheat. It is therefore necessary to construct an open-source dataset for recognizing impurities in wheat, and to perform studies on wheat impurities recognition using CNNs. To the best of our knowledge, our study is the first to recognize impurities in wheat using a CNN.

In this study, we collect many wheat samples, and begin by preprocessing the wheat images to reduce the influence of motion, shading, and differences in light before classifying and labeling them. Second, we propose the method for recognizing impurities based on a CNN, which is an improved Inception\_v3 network that analyzes the image characteristics of the impurities in wheat. The designed WheNet CNN adopts the optimized Adam algorithm to raise the efficiency of model training and optimizes the structure and parameters of the CNN so as to increase recognition accuracy. Finally, the results of a comparative experiment between the ResNet\_101 network and the Inception\_v3 network show that the proposed WheNet network is advantageous with regard to training time, recognition accuracy, area under curve (AUC), and recall rate. The feasibility and effectiveness of the WheNet network were thus verified. The proposed network can effectively overcome the drawbacks of traditional methods, and makes the recognition process more rapid and accurate.

## II. MATERIALS AND RELATED WORK

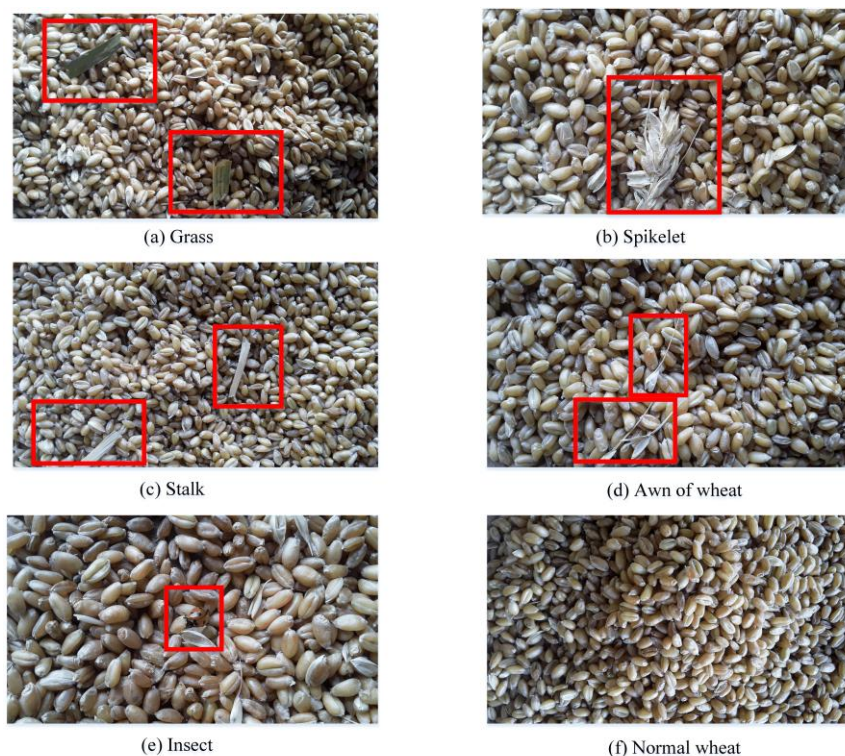
### A. IMAGE ACQUISITION AND ARRANGEMENT

The images were acquired in May 2019 from the Xiaotangshan Modern Agricultural Science and Technology Demonstration Park, Beijing, China (coordinates: 40°18'N and 116°46'E). This area has a mean annual precipitation of 575.5 mm and a mean annual temperature of 11.9°C. Note that the wheat samples and its impurities are more representative in a natural environment. A total of 6,000 images of wheat in the grain elevators of the combine harvesters were acquired during their operation using a MindVision color industrial camera accompanied with a 16-mm focal length lens. The image resolution was set as 2,000,000 pixels.

In accordance with the image characteristics of the wheat with impurities, the images were divided into six categories, namely

insects, stalks, grass, awns of wheat, spikelet, and normal wheat, as shown in Fig. 1. Due to the limited number of acquired image samples and the need to reduce the training time, samples of the training set were enlarged. By revolving the samples at seven angles—0°, 45°, 90°, 135°, 180°, 225°, and 270°—as well as the

horizontal and vertical flips in various directions, the original dataset for training was expanded to six times as large as the original; the final training dataset included a total of 36,000 images of wheat.



**FIGURE 1.** Examples of images from the partial dataset

In addition, the dataset was divided into two mutually independent sets, one for training and the other for validation. The sample set was composed of a number of images from each category and 70% of the samples were randomly chosen from the sample set for training, and the remaining 30% were used for

validation. The training set, which was used to train the model, comprised a total of 25,200 images. The validation set, which validated the model accuracy, comprised a total of 10,800 images. The distribution of the wheat images is listed in Table 1.

**TABLE 1.** Quantity distribution of wheat images

Category	Training set/image	Validation set/image	Sample size (width × height)
Insect	3,500	1,500	64 × 64
Stalk	4,200	1,800	64 × 64
Grass	3,500	1,500	64 × 64
Awn of wheat	4,900	2,100	64 × 64
Spikelet	5,600	2,400	64 × 64
Normal Wheat	3,500	1,500	64 × 64
Total	25,200	10,800	64 × 64



## B. IMAGE PREPROCESSING

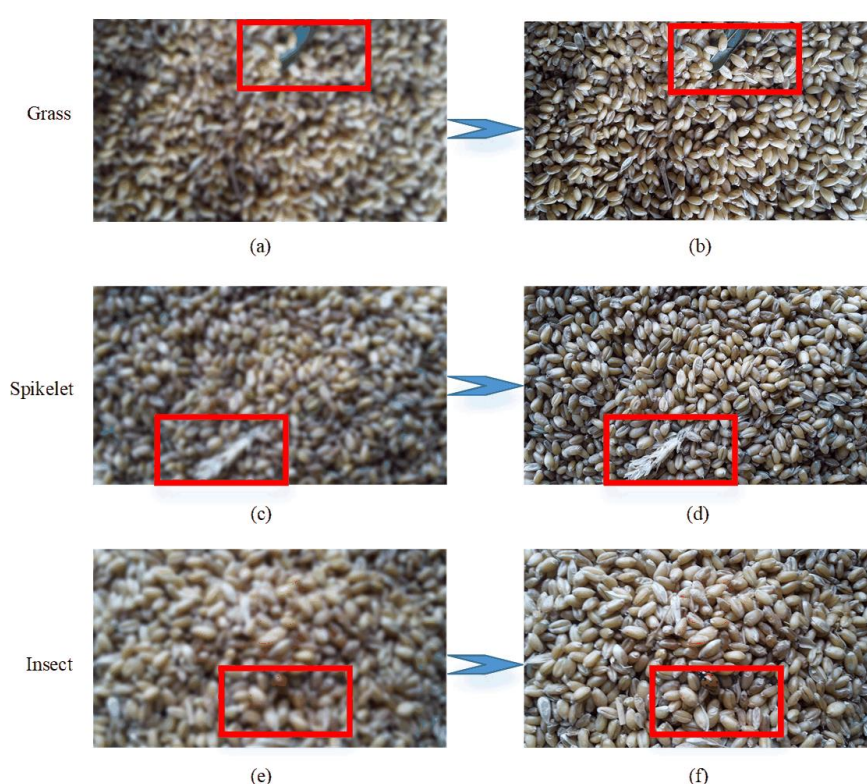
As the existence of lighting and shadows in the images of the wheat complicates image recognition, preprocessing of the original color images was imperative. The common methods of image preprocessing include image enhancement, deblurring, and binarization [30].

### 1) ELIMINATION OF MOTION BLUR

Images to be inputted in the CNN do not require complicated preprocessing. As long as the original images are inputted, the characteristics can be obtained through backpropagation (BP)

and layer-by-layer learning. In general, the images are adjusted for light, contrast, saturation, noise, and blur.

As motion blurring existed in a considerable number of the wheat images acquired for this experiment, recognizing impurities from the blurred images was critical. Motion blurring refers to image blurring caused by the relative motion between the camera and the scene to be shot. In this study, the Wiener filtering algorithm [31] is adopted to preprocess the wheat images with motion blurring. For example, the original images of grass (Fig. 2a), spikelet (Fig. 2c), and insect (Fig. 2e), which contained motion blurring, were processed, resulting in clear images (Fig. 2b, 2d, and 2f).



**FIGURE 2.** Comparison between the images of the impurities in wheat before and after the elimination of motion blurring

### 2) IMAGE BINARIZATION

The purpose of image binarization is to differentiate the foreground of image from its background. In this article, Multi-Scale Retinex with Color Restoration (MSRCR) is adopted for this purpose. In this method, first, a gray value is assumed before dividing the image into the foreground and background. Then, calculation and comparison are conducted through the Otsu algorithm to determine the optimal segmentation threshold before ensuring that the maximum difference exists between the background and foreground [32]. Fig. 3b is the acquired result of binarization by adopting the Otsu algorithm directly for Fig. 3a.

The outlines between the grains in the original image were not clear enough, and thus, we enhanced the original image before conducting binary segmentation. The result obtained after processing the original wheat images in Fig. 3a with the MSRCR algorithm is shown in Fig. 3c. The enhancement by the MSRCR algorithm increased the image contrast. As shown in Fig. 3d, the binarization image obtained by adopting the Otsu algorithm after image enhancement by the MSRCR algorithm shows a more complete foreground in comparison with that of the original image in Fig. 3b. Moreover, after undergoing suitable binary segmentation, clearer outlines among the grains were evident.

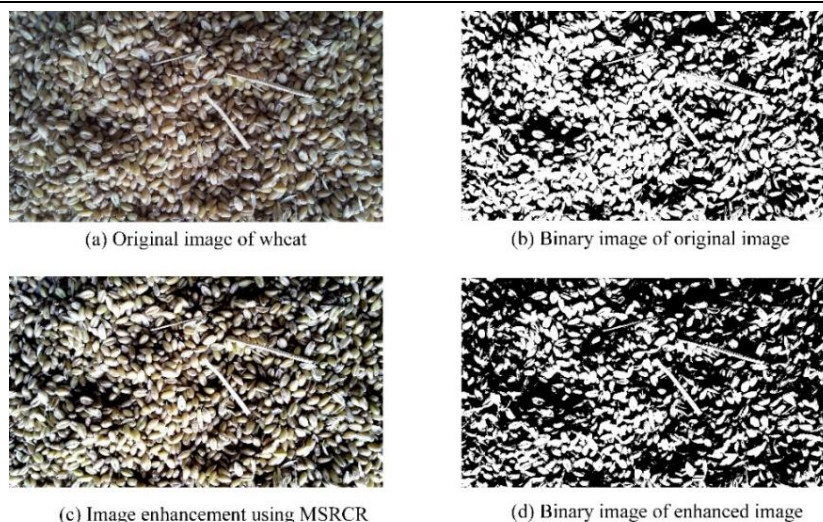


FIGURE 3. Image binarization

### C. LABELING OF THE IMAGE DATASET

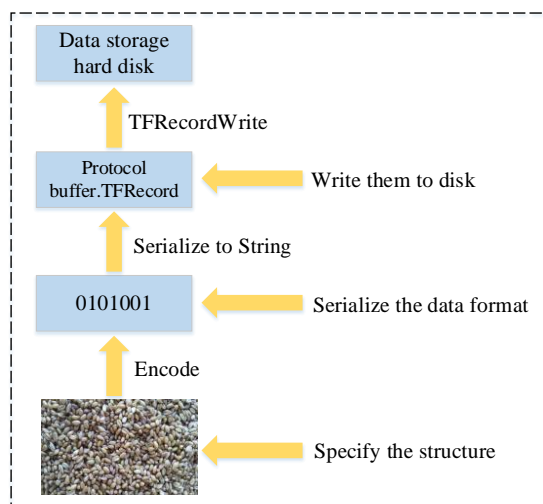


FIGURE 4. The wheat data set is converted into a TFRecord format

The CNN algorithm requires labeling many datasets. In this research, TensorFlow, a deep learning platform devised by Google, was adopted to label the datasets of the wheat images. First, the sorted training set and validation set were stored under two folders, with each folder containing the six categories of images seen in Table 1, and the labels of the corresponding folders were set up. Then, the built-in functions of TensorFlow were used to convert each original image into binary data, wherein the first byte indicated the category of the images and the remaining  $64 \times 64 \times 3$  bytes denoted the basic information of the images. Finally, we separately converted the training and validation sets into two independent binary documents before obtaining the dataset of the wheat impurity images whose labeling had been completed. In addition, the data reading mode of TensorFlow adopts TFRecord format. TensorFlow and

TFRecord have some supporting functions. TFRecord is a binary data storage format, which can speed up the processing of image data. Using TFRecord, the original image data is actually read first, then converted to the TFRecord format, and finally stored on the hard disk. The data processing process is shown in Fig. 4.

### III. IMAGE RECOGNITION ALGORITHM OF IMPURITIES IN WHEAT

#### A. NETWORK STRUCTURE OF INCEPTION\_V3

Compared with other CNNs such as AlexNet [19], VGGNet [14], and ResNet [10], the Inception\_v3 network [33] has been successfully applied in many fields due to its simplicity and excellent classification performance. The Inception module designed in the Inception\_v3 network has raised the utilization ratio of parameters. As shown in Fig. 5, the Inception module was designed as the  $1 \times 1$  and  $3 \times 3$  branches of the convolution kernel, and the structural characteristics of impurities at different positions were extracted and learned [34]. By receiving the input from the preceding layer, the Inception module realizes the fusion of multi-scale characteristics. With convolution kernels of different sizes added to the same layer, their basic structure includes four branches, each being applied via  $1 \times 1$  convolution. The addition of the  $1 \times 1$  convolution kernel in front of the  $3 \times 3$  convolution kernels attains dimensionality reduction that can reduce the amount of calculation [12]. With the Inception module connecting the nodes through  $1 \times 1$  and  $3 \times 3$  convolution kernels in the four branches as well as the  $3 \times 3$  maximum pooling layer, an efficient sparsity structure conforming to the Hebbian theory, an unsupervised learning method and the simplest neuron learning rule, has been

established.

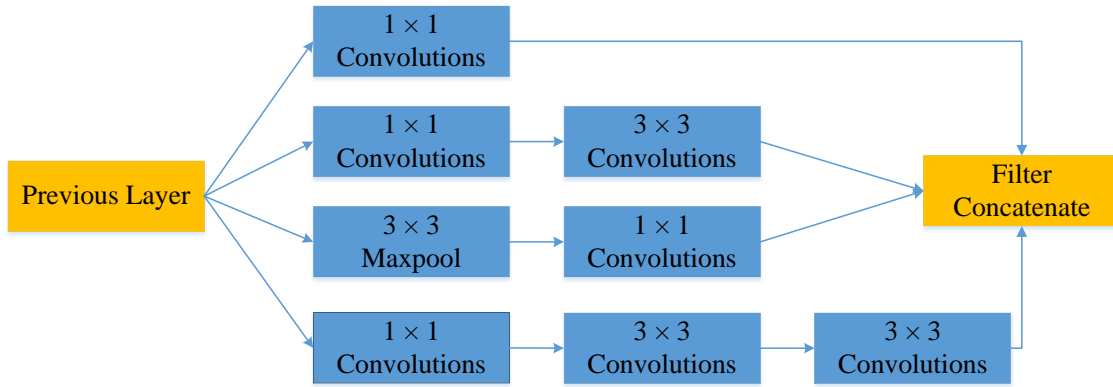


FIGURE 5. Inception Module

### B. STRUCTURE OF THE WheNet CNN

The complex distribution of the wheat impurity images resulted in a large amount of image data. Although mature deep CNNs can extract all the features of images, the complex structure and long training time can lower recognition accuracy slightly [35], [36]. Based on the Inception\_v3 network, improvement and optimization were undertaken, and the WheNet CNN was designed. The improved network structure is shown in Fig. 6, and the detailed design of the network parameters is listed in Table 2. As a type of multi-layer neural network, a CNN includes

multi-layer convolution and pooling layers, which are composed of the input layer, convolution layer, pooling layer, and output layer [37]. WheNet adopts the BP algorithm in terms of the cost function between the minimum training result and the true value, as represented in Formula (1):

$$L = -\frac{1}{|X|} \sum_{i=1}^{|X|} \ln(p(y^i|x^i)) \quad (1)$$

In Formula (1),  $|X|$  indicates the size of the training set, while  $y^i$  and  $x^i$  denote the  $i^{\text{th}}$  training sample and its corresponding category label, respectively.

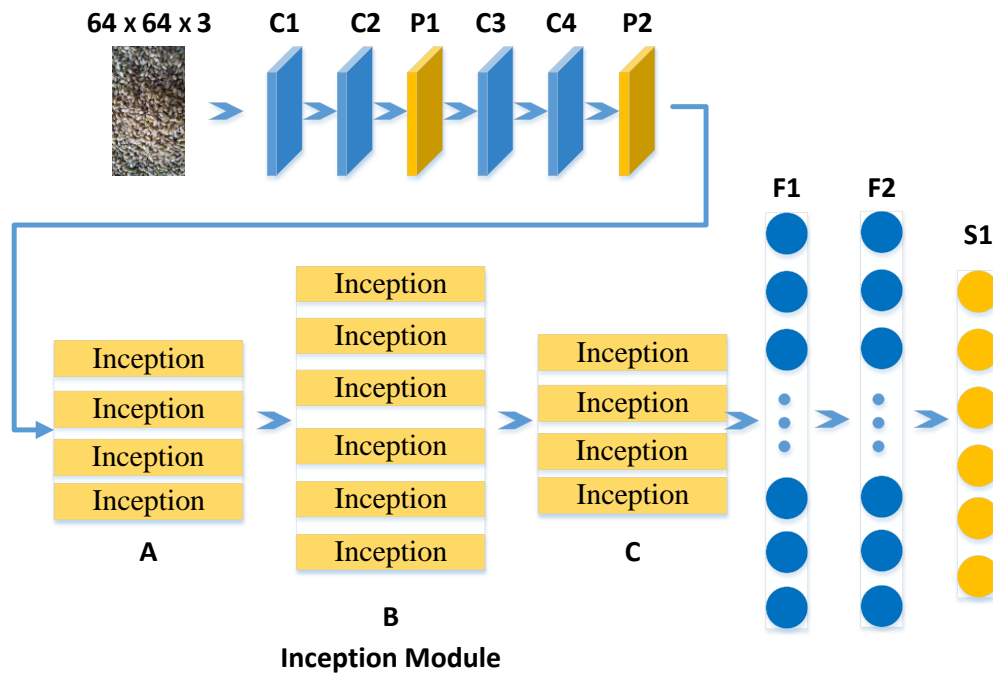


FIGURE 6. CNN structure



TABLE 2. Design of CNN parameters

No.	Layer	Size of convolution kernel	Type	Input size
C1	Convolution 1	$5 \times 5$	1	$64 \times 64 \times 3$
C2	Convolution 2	$5 \times 5$	1	$60 \times 60 \times 64$
P1	Pooling layer 1	$3 \times 3$	2	$29 \times 29 \times 64$
C3	Convolution 3	$5 \times 5$	1	$25 \times 25 \times 128$
C4	Convolution 4	$5 \times 5$	1	$21 \times 21 \times 256$
P2	Pooling layer 2	$3 \times 3$	2	$10 \times 10 \times 256$
A	Conv block 1	$4 \times$ Inception block 1		$10 \times 10 \times 288$
B	Conv block 2	$6 \times$ Inception block 1		$5 \times 5 \times 768$
C	Conv block 3	$4 \times$ Inception block 1		$5 \times 5 \times 1280$
F5	Fully connected 1	Logits		$2 \times 2 \times 2048$
F6	Fully connected 2	Logits		$1 \times 1 \times 2048$
S1	Softmax	Number of neurons: 6		$1 \times 1 \times 6$

To extract the characteristics of the wheat images as accurately as possible, the preprocessed images of the sample set were inputted into the WheNet network to receive training before generating the image recognition network [19]. To obtain better network parameters of the impurity images, we adjusted the number of convolutional layers and the width of the WheNet network. We considered a  $64 \times 64$  image as an example. The network included four convolutional layers (C1, C2, C3, and C4), two pooling layers (P1 and P2), three Inception modules (A, B, and C), two fully connected layers (F1 and F2), as well as one classification layer. The first two convolutional layers were connected with a pooling layer, and the last fully connected layer used the Softmax function for classification. The final number of categories was the same as the number of output categories of the classification layer.

The description of the kernel layers in the network appears below.

#### 1) INPUT LAYER

As the color and pattern of the wheat grains are important characteristics for identity recognition, the filtered negative and positive samples were converted to a  $64 \times 64 \times 3$  matrix through interpolation calculation. In addition, the positive and negative samples were labeled as “2” and “1,” respectively, and used as the input data for the convolutional network training.

#### 2) CONVOLUTIONAL LAYER

The convolutional layer involves convolving a convolutional kernel that can be learned and the preceding layer characteristic pattern that has undergone batch processing. As each convolutional kernel has its unique extracted characteristics, the larger the number of the convolutional kernels in a convolutional layer, the greater the number of the characteristics extracted. The

output characteristic pattern is obtained after combining many characteristic patterns that have been convolved and have undergone offset processing via an activation function.

$$x_j^l = f_{\max}(\sum_{i \in M_j} x_i^{l-1} \times k_{ij}^l + b_j^l, 0) \quad (2)$$

In Formula (2),  $l$  is the sequence number of a convolutional layer,  $x_j^l$  refers to the output of the  $j^{\text{th}}$  neuron of the convolutional layer  $l$ ,  $x_i^{l-1}$  indicates the output of the  $i^{\text{th}}$  neuron of the  $l-1^{\text{st}}$  layer,  $M_j$  denotes the set of input characteristic patterns,  $k_{ij}^l$  stands for convolutional kernels, and  $b_j^l$  represents the offset.

#### 3) POOLING LAYERS

To maintain the invariance of the revolution, translation, and elasticity of the original data, pooling technology is used to integrate the characteristic points in the small neighborhood and obtain new characteristics.

$$x_j^l = f(\alpha^l \text{pooling}(x_j^{l-1}) + b_j^l) \quad (3)$$

In Formula (3),  $\alpha$  is the weight coefficient,  $b_j^l$  denotes the offset, and  $x_j^l$  indicates the vector of  $x_j^{l-1}$  after its maximum value is pooled. The pooled kernel is  $3 \times 3$ , and the step length is  $2 \times 2$ .

The common pooling modes include *mean-pooling* and *max-pooling*. As maximum pooling can reduce errors and retain more image information, *max-pooling* was adopted in this experiment. If  $x_j$  denotes the pooling layer, the computational process of  $x_j$  can be described in Formula (4).

$$x_j = \max - \text{pooling}(x_{j-1}) \quad (4)$$

#### 4) FULLY CONNECTED LAYER

There can be several fully connected layers, which in fact are the

hidden layers of ordinary neural network layers. Each neuron in a fully connected layer is fully connected with all neurons in its preceding layer. The neural nodes in the same layer have no connection. Through weights on connecting lines, the neural nodes in each layer conduct forward propagation and calculation before obtaining the input of the neural nodes in the next layer.

After convolutional and pooling processing, a series of characteristic patterns are outputted via images, while the input received in the next layer is the output of the preceding layer. Therefore, the pixels of the characteristic patterns are fetched in a sequence, namely the output characteristic patterns  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_a$ . After the weighted summation of the vectors and the responses of the activation functions, the output formula of the fully connected layers is elicited as follows:

$$\mathbf{x}^l = \alpha^l \mathbf{x}^{l-1} + \mathbf{b}^l \quad (5)$$

In Formula (5),  $\mathbf{x}^l$  is the output vector of the fully connected layers,  $\mathbf{x}^{l-1}$  is the input vector,  $\alpha^l$  is weight coefficient, and  $\mathbf{b}^l$  is the offset item.

### C. INCREASING THE EFFICIENCY OF MODEL TRAINING

#### 1) LEARNING RATE AND ADAM OPTIMIZATION ALGORITHM

Learning rate is a very important parameter in the deep learning network. In this study, neural network training was carried out through the exponential decay learning rate [38], which not only makes the network rapidly approach a comparatively excellent solution in the earlier stage of training, but also ensures that the network is unlikely to undergo much fluctuation in the later stage of training, thus being closer to local optimum. The formula of learning rate is as follows:

$$\text{decayed\_learning\_rate} = \text{learning\_rate} \times \frac{\text{global\_step}}{\text{decay\_rate}^{\text{decay\_steps}}} \quad (6)$$

In Formula (6), *learning\_rate* is the initial learning rate, *global\_step* denotes the global step of the attenuation calculation, *decay\_rate* is the decay rate, and *decay\_steps* refers to the decay speed (i.e., how many rounds are needed to iterate all the training data). The training data in this case amount to 100,000 images, and each *batch\_size* is 64.

In this article, the gradient descent algorithm and Adam optimization algorithm [39] were adopted to conduct the network training. The Adam algorithm follows the method of gradient descent. The learning step length of the iteration parameters for each time has a fixed range, which dynamically

adjusts the learning rate of each parameter by making use of the first- and second-order moment estimations of the gradient [40]. The formulae of this algorithm are given below.

$$t = t + 1 \quad (7)$$

$$\mathbf{g}_t = \nabla_{\theta} f_t(\theta_{t-1}) \quad (8)$$

In Formulae (7) and (8),  $f_t(\theta)$  represents the gradient of  $\theta$ , that is, the partial derivative vector of  $f_t$  from  $\theta$  under the time step  $t$ , and the initial parameters  $t = 0$  and  $\theta_0 = 0$ .

$$m_t = \alpha m_{t-1} + (1-\alpha) \mathbf{g}_t \quad (9)$$

$$n_t = \beta n_{t-1} + (1-\beta) \mathbf{g}_t^2 \quad (10)$$

The biased first- and second-order moments were estimated in Formula (9) and (10), while  $\mathbf{g}_t$  is the current gradient, where  $\alpha$  and  $\beta$  represent the exponential decay rate and initial parameter of the moment estimation ( $p_0 = 0$ ,  $q_0 = 0$ ), respectively.

$$\widehat{m}_t = \frac{m_t}{1-\alpha^t} \quad (11)$$

$$\widehat{n}_t = \frac{n_t}{1-\beta^t} \quad (12)$$

In Formulae (11) and (12),  $\widehat{m}_t$  and  $\widehat{n}_t$  are the corrections of  $m_t$  and  $n_t$ , the Adam algorithm can make a dynamic adjustment according to the gradient, which imposes a dynamic restraint on the learning rate.

$$\Delta \theta_t = - \frac{\widehat{m}_t \cdot \gamma}{\sqrt{\widehat{n}_t + \varepsilon}} \quad (13)$$

The value of  $\theta$  was updated in Formula (13).  $\gamma$  is equivalent to the initial learning rate, and  $\varepsilon$  is a small constant with a stable value. Table 3 shows the values of the above parameters for the training of WheNet in this study.

TABLE 3. WheNet hyperparameters

Parameter	Value
<i>Learning_rate</i>	0.001
<i>Global_step</i>	100,000
<i>Decay_steps</i>	1,563
<i>Decay_rate</i>	0.95
<i>Batch_size</i>	64
$\alpha$	0.9
$\beta$	0.999
$\varepsilon$	$10^{-8}$

#### 2) LOCAL RESPONSE NORMALIZATION



Local response normalization (LRN) plays a role in conducting normalized processing for each element in accordance with the given coefficients so as to improve the accuracy of the classification on the part of the neural network [30], as shown in Formula (14):

$$b_{x,y}^i = a_{x,y}^i / \left[ k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^i)^2 \right]^\beta \quad (14)$$

In Formula (14),  $a_{x,y}^i$  represents the output of the  $i^{\text{th}}$  convolutional kernel at position  $(x, y)$ ,  $b_{x,y}^i$  refers to the output after local response normalization,  $N$  indicates the number of convolutional kernels, and  $k$ ,  $\alpha$ ,  $n$ , and  $\beta$  are constants.

### 3) LOSS FUNCTIONS

The objective of CNN training is to minimize the loss function. The computational formula of loss function is

$$L(W, b) = - \sum_{i=1}^N \sum_{j=1}^C I(\hat{y}_i = j) \log p_i^j \quad (15)$$

In Formula (15),  $W$  denotes weight,  $b$  stands for the offset items,  $\hat{y}_i$  refers to the expected value of the  $i^{\text{th}}$  training sample,  $j$  is the category of image samples, and  $I$  represents the indicator function. When  $\hat{y}_i = j$ , the numerical value of  $i$  is 1; otherwise, it is 0.  $p_i^j$  is the predicted probability of the  $i^{\text{th}}$  training sample in the  $j^{\text{th}}$  category.  $C$  is the number of the categories of the training samples.  $N$  is the total number of training samples. The iteration expression of  $W$  weight is

$$W_i = W_i - \gamma \frac{\partial L(W, b)}{\partial W} \quad (16)$$

$$b_i = b_i - \gamma \frac{\partial L(W, b)}{\partial b_i} \quad (17)$$

In Formulae (16) and (17),  $\gamma$  denotes the learning rate used to control the intensity of the BP.

## IV. RESULTS AND DISCUSSION

### A. COMPARISON WITH OTHER TECHNIQUES

TABLE 5. Performance comparison of the three CNNs

Network	Input image size	Training time (h)	Top_1 (Accuracy rate; %)	Top_5 (Accuracy rate; %)	Recognition efficiency (s/image)
ResNet_101	224 × 224	14	94.76	96.75	0.2
Inception_v3	299 × 299	15	95.84	97.70	0.3
WheNet	64 × 64	11	98.59	99.98	0.1

In this article, the network operated on a Ubuntu 16.04 System. The processing platform was a desk computer, the processor was Intel Core i5-7500, the dominant frequency was 3.40 GHz, the internal storage was 64 GB, the hard disk capacity was 4.1 TB, two 8-GB NVIDIA GeForce GTX1080 graphics cards were used, and the programming language was Python. The open-source framework of TensorFlow deep learning was adopted, and the software PyCharm was used to program the deep neural network layers, as shown in Table 4.

The dataset of the images of the wheat impurities was trained on ResNet\_101 and Inception\_v3 before being compared with WheNet for recognizing the impurities in the wheat. Table 5 presents the parameter configurations and training results of the three networks. According to the comparative analysis in Table 5, the network input of WheNet for recognizing impurities is characterized by a smaller number of input images and a shorter training time of 11 h. In addition, the testing efficiency of an image is higher than that of ResNet\_101 and Inception\_v3, and only 0.1 s are needed for each image. The numerical value of the loss function is relatively low. The recognition rates on the validation sets Top\_1 and Top\_5 are 98.59% and 99.98%, respectively.

TABLE 4. Hardware and software environment.

Configuration Item	Value
GPU	NVIDIA GeForce GTX1080 8GB
CPU	Intel Core i5-7500
Hard disk	4.1 TB
Memory	64 GB
Clock speed	3.40 GHz
Operating system	Ubuntu 16.04 (64-bit)

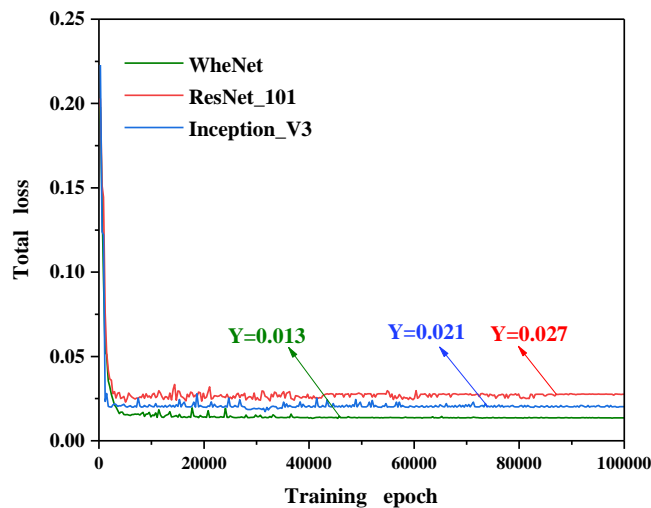


FIGURE 7. Loss rate curves

With the incessant increase in the number of iterations, the classification errors of the training set show a decreasing trend, as indicated in Fig. 7. When the serial number of training iterations of the WheNet network reaches 50,000, the training loss essentially stops and approximates to a stable value. The stable numerical value of mean loss rate is 0.013, which is lower than that on ResNet\_101 and Inception\_v3. To summarize, judging from the training time, recognition accuracy, and efficiency, the network structure of WheNet designed in this article is sound and provides a satisfactory training result.

## B. NETWORK TESTING EXPERIMENT AND RESULT ANALYSIS

Due to the considerable difference in various categories of wheat impurities in the training and validation sets, as well as the imbalance of the data samples, accuracy is not a sufficient metric to describe the practical application performance. Thus, to evaluate the recognition effect of the networks, in June 2019, we selected 300 images from each of the 5 different categories of impurities from the operating images offered by the combine harvesters. They served as a set for testing before comparing the relationship among WheNet, ResNet\_101, and Inception\_v3, and were used to evaluate the receiver operating characteristic (ROC) curve, AUC, confusion matrix, and recall rate. These evaluation functions can be accessed from the Scikit-learn library.

### 1) ROC CURVE AND AUC

For the ROC curve, the abscissa and ordinate are characterized by a false positive rate (FPR) and a true positive rate (TPR), respectively. The summation of the areas of the various parts below the curve is the AUC. In the evaluation accuracy indexes

of image recognition, we can quantize the ROC curve of the model by calculating the AUC below the curve. The closer the location of the ROC curve to the top left corner, the better its performance as a classifier. The larger numerical value of AUC indicates that this classifier performs better. The computation for the FPR and TPR are conducted as follows:

$$FPR = \frac{FP}{N} \quad (18)$$

$$TPR = \frac{TP}{P} \quad (19)$$

In Formula (18) and (19),  $P$  refers to the number of real positive samples,  $N$  indicates the number of real negative samples,  $TP$  represents the number of the positive samples that have been predicted by the classifier in  $P$  positive samples, and  $FP$  denotes the number of the positive samples that have been predicted by the classifier in  $N$  negative samples.

In most cases, as the ROC curve lies above the straight line ( $y = x$ ), the value ranges from 0.5 to 1. In addition, the ROC curve is formed by sequentially connecting the points whose coordinates are  $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ . Thus, AUC is calculated using the following formula (20):

$$AUC = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i) \cdot (y_i + y_{i+1}) \quad (20)$$

where the coordinate values  $x_i$  and  $y_i$  are the values of the above-mentioned FPR and TPR, respectively.

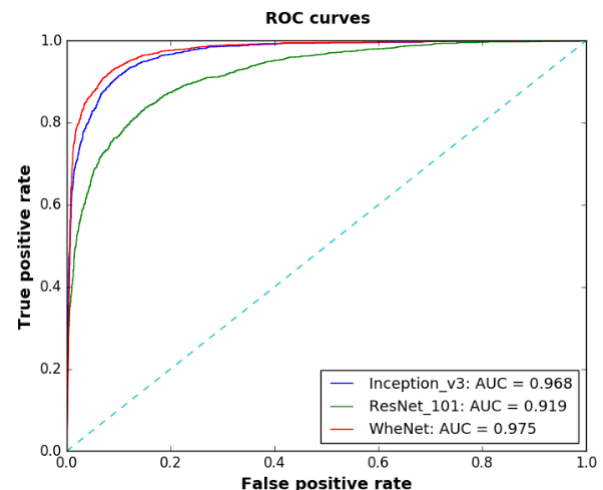


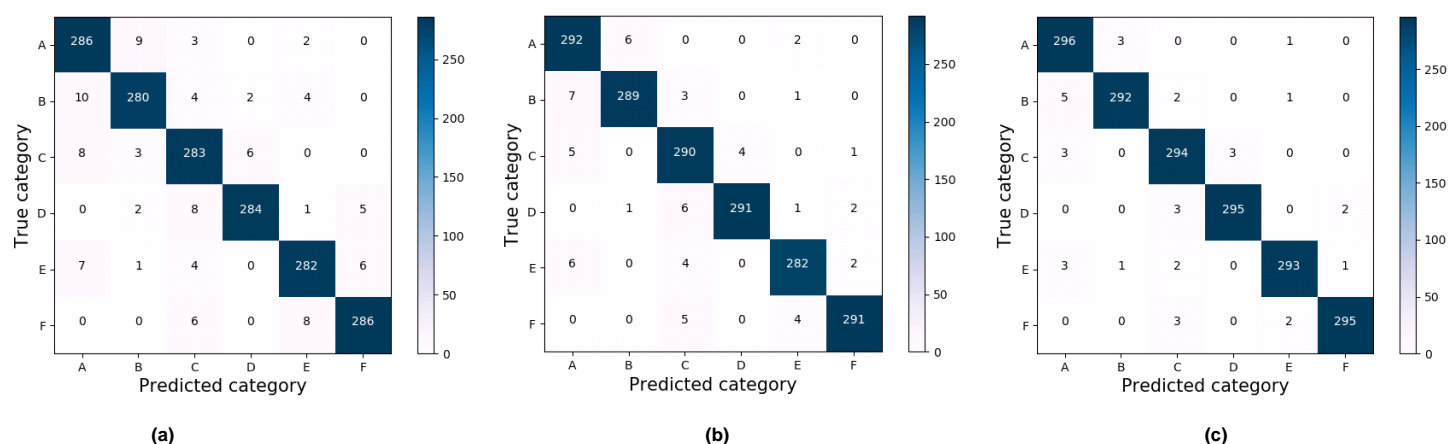
FIGURE 8. ROC curve of the image of the wheat impurities and the AUC

As shown in Fig. 8, when the images are trained in WheNet, ResNet\_101, and Inception\_v3, it is obvious that the AUC of the red curve is larger than that of the green curve, indicating that WheNet performs better in terms of accuracy. Its numerical value is 0.975, indicating that the network has a high accuracy rate for recognizing impurities in wheat. This result also indicates the good robustness and strong practicality of WheNet.

## 2) CONFUSION MATRIX AND RECALL RATE

While evaluating the accuracy of image recognition, the confusion matrix is mainly used for comparing the objective results and practical measured values. The recall rate is an index used to evaluate the performance of the classifiers. In Fig. 9, the confusion matrix of the three networks is calculated wherein the real categories (ordinates) are compared with the prediction categories (abscissas) to describe the individual classification

performance of each network. In the figure, A denotes normal wheat, B denotes insects, C denotes stalks, D denotes grass, E denotes awn of wheat, and F denotes spikelet. We can judge the classification result of recognition from the color distribution of the confusion matrix. The deeper the color, the larger the numerical value. It can be clearly seen that ResNet\_101 has the poorest classification effect and WheNet has the best classification rate, as indicated in Fig. 9(a) and Fig. 9(c), respectively.

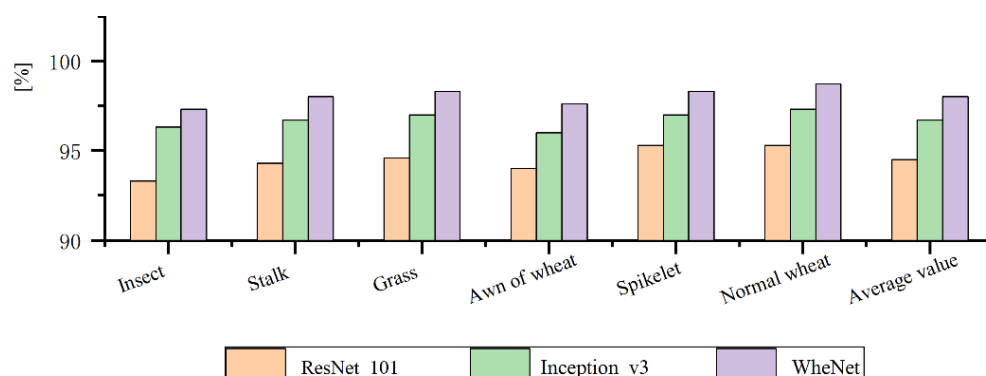


**FIGURE 9.** Test result of the confusion matrix. (a) Individual classification rate by ResNet\_101. (b) Individual classification rate by Inception\_v3. (c) Individual classification rate by WheNet.

To compare the recall rates among WheNet, ResNet\_101, and Inception\_v3, we tested the impurities separately. The calculation was conducted in accordance with Formula (19), and the calculated results are shown in Table 6. Judging from Table 6, the recall rate of WheNet is 98.0%, and the corresponding values of ResNet\_101 and Inception\_v3 are 94.5% and 96.7%, respectively, indicating that the WheNet recognition network is superior to ResNet\_101 and Inception\_v3. The mean F1-score is 98.0%, which indicates that the network has high accuracy, as shown in Table 7. The result also shows the advantage of the WheNet network in identifying impurities in wheat, as shown in Fig. 10.

In terms of the recall rates of grass and spikelet, the three networks all show high numerical values. The main reason is that there is a big difference between spikelet and grass on the one hand and normal wheat on the other in terms of shape and color.

These features are easily differentiated. The recall rate of insects is relatively low because insects are small-sized and their color is similar to that of wheat. One noteworthy point is that this network explicitly addresses the recognition of impurities in wheat and there is a possibility of misjudgment under certain conditions. We discuss this point next.



**FIGURE 10.** Performances of the three tested networks

**TABLE 6. Recall rate comparison of the three networks**

Model	Recall rate (%)						Average value
	Insect	Stalk	Grass	Awn of wheat	Spikelet	Normal wheat	
ResNet_101	93.3	94.3	94.6	94.0	95.3	95.3	<b>94.5</b>
Inception_v3	96.3	96.7	97.0	96.0	97.0	97.3	<b>96.7</b>
WheNet	97.3	98.0	98.3	97.6	98.3	98.7	<b>98.0</b>

**TABLE 7. F1-score comparison of the three networks**

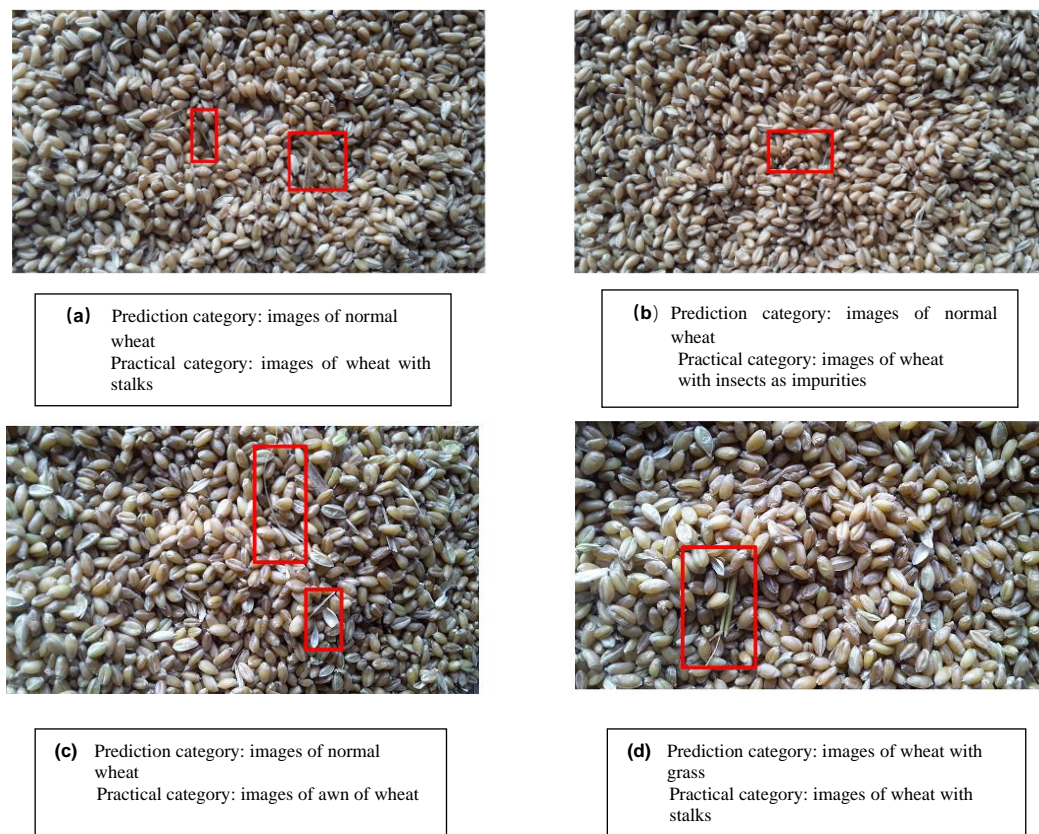
Model	F1-score (%)						Average value
	Insect	Stalk	Grass	Awn of wheat	Spikelet	Normal wheat	
ResNet_101	94.1	93.1	95.9	94.4	95.8	93.6	<b>94.5</b>
Inception_v3	98.5	95.4	97.8	96.9	97.6	95.7	<b>97.0</b>
WheNet	97.9	97.3	98.6	98.1	98.6	97.5	<b>98.0</b>

### 3) ERROR ANALYSIS

In this research, we arranged and analyzed the images of the impurities in wheat by centering on the recognition errors of the WheNet network, as shown in Fig. 11. The main reasons for the recognition errors are as follows:

The fact that partial impurities in wheat are similar to normal

wheat in shape and color leads to errors in image recognition. For example, in Fig. 11(a) and (b), the images of wheat with stalks and of wheat with insect impurities are recognized as images of normal wheat. We can solve this issue by increasing the number of learning samples or adopting the Terahertz imaging method.



**FIGURE 11. Images mistakenly recognized as those of normal wheat**



The existence of occlusion and overlapping between some impurities and wheat leads to recognition errors. For instance, in Fig. 11(c), the images of the impurities in the awns of wheat are mistakenly recognized as those of normal wheat. Again, in Fig.11(d), the images of wheat with stalks are incorrectly recognized as those of wheat with grass as impurities.

Judging from the error analysis, the network has insufficient local recognition capability for a few impurities, and it cannot recognize images correctly and effectively when a large surface area of wheat is occluded.

## V. CONCLUSION

The aims of this research were to address problems such as high labor intensity in traditional manual monitoring of wheat impurities and low recognition rate of current classification methods. Accordingly, we attempted to improve the performance of the Inception\_v3 network and designed a method for recognizing impurities in wheat characterized by WheNet CNN. The experimental results showed that our approach was able to recognize impurities from the backgrounds of images of wheat approximately 98% of the time when using WheNet CNN. This outcome emphasizes the considerable superiority of deep learning and its potential ability to detect impurities in wheat.

The main contributions of this paper can be summarized as below:

- 1) We constructed six datasets of labeled images of wheat, namely normal wheat, insects, grass, stalks, spikelets, and awns of wheat. The training set was composed of 25,200 images and the validation set consisted of 10,800 images. These datasets can be used to investigate automatic recognition applications and testing on wheat. These datasets also provide important insights into other grain impurities.

- 2) In line with practical application requirements and data characteristics, we adopted the Adam optimization algorithm to improve the training accuracy of WheNet. The resulting recognition efficiency reached 0.1 s per image. The recognition rates of both Top\_1 and Top\_5 on the validation set surpassed 98%. Compared with the classic networks of ResNet\_101 and Inception\_v3, the WheNet network is superior in terms of recognition accuracy and training time.

- 3) To test the performance of the network further, we randomly selected 300 images from each of the 5 categories as test sets and measured the mean value of the AUC as 0.975. The recall rate of WheNet was found to be 98.0%, which was higher than that of ResNet\_101 and Inception\_v3. This result proved that the proposed network demonstrates better classification performance.

In conclusion, the network proposed in this study can accurately recognize most impurities in wheat. However, for some impurities that are similar in color to normal wheat, certain occlusions, and overlapping impurities, the network is prone to recognition errors. In the future, to enable wider application, we will deepen the network structure and increase the types and number of learning samples. We also plan to adopt the Terahertz imaging method to enhance the recognition ability of the classifiers.

## ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Project of China (No.2016YFD0702000).

## REFERENCES

- [1] J. Chen, Y. Gu, Y. Lian, and M.N. Han, "Method for online identification of impurities and broken grains in rice based on machine vision," *Transactions of the Chinese Society of Agricultural Engineering.*, vol.34, no.13, pp.187-194, Jul. 2018.
- [2] J. X. Guo, Y. B. Ying, X. Q. Rao, J. W. Li, Y. X. Kang, and Z. Shi, "Hyperspectral image detection of carding inner impurities," *Chinese journal of agricultural machinery.*, vol.43, no.12, pp.197-203, Dec. 2012.
- [3] S. Serranti, D. Cesare, G. Bonifazi, et al, "The development of a hyperspectral imaging method for the detection of Fusarium-damaged, yellow berry and vitreous Italian durum wheat kernels," *Biosystems Engineering.*, vol.115, no.1, pp.20-30, May. 2013.
- [4] C. Wallays, B. Missotten, J. D. Baerdemaeker, et al, "Hyperspectral waveband selection for on-line measurement of grain cleanness," *Biosystems Engineering.*, vol.104, no.1, pp.1-7, Sep. 2009.
- [5] J. Mahirah, K. Yamamoto, M. Miyamoto, et al, "Monitoring harvested paddy during combine harvesting using a machine vision-Double lighting system," *Engineering in Agriculture Environment and Food.*, vol.10, no.2, pp.140-149, Apr. 2017.
- [6] M. A. Shahin, S. J. Symons, "Detection of fusarium damage in Canadian wheat using visible/near-infrared hyperspectral imaging," *Journal of Food Measurement and Characterization.*, vol.6, no.4, pp.3-11, Oct. 2012.
- [7] C. B. Singh, D. S. Jayas, J. Paliwal, et al, "Fungal Damage Detection in Wheat Using Short-Wave Near-Infrared Hyperspectral and Digital Colour Imaging," *International Journal of Food Properties.*, 2012, 15(1):11-24. vol.15 no.1, pp.11-7, Jan. 2012.
- [8] M. A. Ebrahimi, M. H. Khoshtaghaza, S. Minaei, et al,

- “Vision-based pest detection based on SVM classification method,” *Computers and Electronics in Agriculture*, vol.137, no.1, pp.52-58, May. 2017.
- [9] T. Liu, W. Chen, W. Wu, “Detection of aphids in wheat fields using a computer vision technique,” *Biosystems Engineering*, vol.141, no.1, pp.82-93, Jan. 2016.
- [10] K.M.He, X. Y. Zhang, S.Q. Ren, and J. Sun, “Deep residual learning for image recognition,” *In Proc. IEEE Conf. Comput. Vis. Pattern Recognit (CVPR)*, Aug. 2016, pp.770-778.
- [11] Y. H. Ng, M. Hausknecht, S. Vijayanarasimhan, et al, “Beyond short snippets: Deep networks for video classification,” *In Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, Aug. 2015, pp.4694-4702.
- [12] C. Szegedy, W. Liu, Y. Jia, et al, “Going deeper with convolutions,” *In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2014, pp. 1–9.
- [13] G. E. Hinton, S. Osindero, Y. W. The, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol.18, no.7, pp.1527-1554, May. 2014.
- [14] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *Computer Science- - Computer Vision and Pattern Recognition*, pp.1409-1556, 2014.
- [15] C. Szegedy, A. Toshev, D. Erhan, “Deep neural networks for object detection,” *Advances in Neural Information Processing Systems*, vol.26, no.3, pp.2553-2561, May. 2013.
- [16] W. Ding, G. Taylor, “Automatic moth detection from trap images for pest management,” *Computers and Electronics in Agriculture*, 2016, 123:17-28. vol.123, pp.17-28, Apr. 2016.
- [17] B. Mandal, L. Li, G. S. Wang, et al, “Towards detection of bus driver fatigue based on robust visual analysis of eye state,” *IEEE Transactions on Intelligent Transportation Systems*, 2017, 18(3): 545–557. vol.26, no.3, pp.2553-2561, May. 2013.
- [18] R. Singh, H. Om, “Newborn face recognition using deep convolutional neural network,” *Multimedia Tools and Applications*, vol.76, no.18, pp.19005-19015, Sep. 2017.
- [19] A. Krizhevsky, I. Sutskever, G. E. Hinton, “Image-Net classification with deep convolutional neural networks,” *Communications of the Acm.*, vol.60, no.6, pp.84-90, Jun. 2017.
- [20] P. S. Yuan, W. Li, S.G. Ren, H. L. Xu, “Identification of chrysanthemum flower types and varieties based on convolutional neural network,” *Transactions of the Chinese Society of Agricultural Engineering*, vol.34, no.5, pp.152-158, May 2018.
- [21] J. Amara, B. Bouaziz, A. Algergawy, et al, “A deep learning-based approach for banana leaf diseases classification,” *B. Mitschang , Bonn: Lecture Notes in Informatics*, pp.79-88, 2017.
- [22] K. Yang, H. Liu, P. Wang, Z. Meng and J. Chen, “Convolutional neural network-based automatic image recognition for agricultural machinery,” *International Journal of Agricultural and Biological Engineering*, vol.11, no.4, pp.200-206, Jul. 2018.
- [23] S. P. Huang, C. Sun, L. Qi, X. Ma, and W. J. Wang, “Detection method of rice blast based on deep convolutional neural network,” *Transactions of the Chinese Society of Agricultural Engineering*, vol.33, no.20, pp.169-176, Oct. 2017.
- [24] S. P. Mohanty, D.P. Hughes, M. Salathé, “Using Deep Learning for Image-Based Plant Disease Detection,” *Frontiers in Plant Science*, vol.7, pp.11419, Sep. 2016.
- [25] D.S.F. Alessandro, D. M. Freitas, G. D. S. Gercina, et al, “Weed detection in soybean crops using ConvNets,” *Computers and Electronics in Agriculture*, vol.143, pp.314-324, Oct. 2017.
- [26] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.
- [27] Z. Q. Lin, S. M. Mu, F. Huang, K. A. Mateen, M. J. Wang, W. L. Gao, and J. D. Jia, “A Unified Matrix-Based Convolutional Neural Network for Fine-Grained Image Classification of Wheat Leaf Diseases,” *IEEE ACCESS*, vol. 7, pp: 11570-11590, Jan. 2019.
- [28] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, “Deep learning for image-based cassava disease detection,” *Frontiers Plant Science*, vol. 8, pp. 1852, Oct. 2017.
- [29] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of Rice diseases using deep convolutional neural networks,” *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017.
- [30] Z. H. Huang, L. K. Huang, et al, “Framelet regularization for uneven intensity correction of color images with illumination and reflectance estimation,” *Neurocomputing*, vol.314, pp.154–168, Nov. 2018.
- [31] X. Zhang, Q. Yue, F. Meng, et al, “Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks,” *IEEE Access*, vol. 6, pp. 30370 -

30377, Jun 2018.

- [32] S.C. Satapathy, M. R. N. Sri, V. Rajinikanth, et al, "Multi-level image thresholding using Otsu and chaotic bat algorithm," *Neural Computing and Applications*., vol.29, no.12, pp.1285-1307, Jun. 2018.
- [33] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Science*., vol. 7, p.1419, Sep. 2016.
- [34] J. Peng, Y. H. Chen, B.Liu, D.J. He, and C.Q. 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. May. 2019.
- [35] K. Thenmozhi, U. Srinivasulu Reddy," Crop pest classification based on deep convolutional neural network and transfer learn," *Computers and Electronics in Agriculture*., vol: 164. pp: 0168-1699. Jul. 2019.
- [36] Y. Altuntaş, Z. Cömert, A. F. Kocamaz," Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach," *Computers and Electronics in Agriculture*., vol: 163. pp: 0168-1699. Jun. 2019.
- [37] M. M. Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*., vol.235, no. 26, pp. 228–235, Apr. 2017.
- [38] Smith, N. Leslie, "Cyclical Learning Rates for Training Neural Networks, " *IEEE Applications of Computer Vision*., pp. 464-472, 2017.
- [39] Y. Y. Sun, W. Zhang, H. Gu, C. Liu, S. Hong, and W. Xu, et al, "Convolutional Neural Network based models for improving super-resolution imaging," *IEEE ACCESS*., vol.7, pp. 43042-43051, Apr. 2019.
- [40] Kingma, P. Diederik, and J. Ba, "Adam: A Method for Stochastic Optimization," *Computer Science*., vol. 23, pp. 1412.6980, Jul. 2015.