EAI Endorsed Transactions
on e-Learning

Research Article EAI.EU

# Recognition system for fruit classification based on 8-layer convolutional neural network

Jia-Ji Wang[1,*]

[1]School of Math and Information Technology, Jiangsu Second Normal University, Nanjing, Jiangsu 210016, P R China
E-mail: wongjiaji@126.com

## Abstract

INTRODUCTION: Automatic fruit classification is a challenging task. The types, shapes, and colors of fruits are all essential factors affecting classification.
OBJECTIVES: This paper aimed to use deep learning methods to improve the overall accuracy of fruit classification, thereby improving the sorting efficiency of the fruit factory.
METHODS: In this study, our recognition system is based on an 8-layer convolutional neural network(CNN) combined with the RMSProp optimization algorithm to classify fruits. It is verified through 10 times 10-fold crossover validation.
CONCLUSION: Our method achieves an accuracy of 91.63%, which is superior to the other four state-of-the-art methods.

**Keywords:** fruit classification, deep learning, convolutional neural network, RMSProp.

Received on 4 June 2021, accepted on 12 December 2021, published on 17 February 2022



doi: 10.4108/eai.17-2-2022.173455

*Corresponding author. Email:wongjiaji@126.com

## 1. Introduction

The traditional fruit sorting methods, which rely on manpower, consume a lot of time and labor. Fruits can be classified according to different internal structures and the origin of the fruit. In the factory, fruit classification can help employees improve the efficiency of fruit packaging and transportation[1]. In supermarkets, because customers are used to choosing different fruits themselves, fruit classification can help cashiers quickly determine the price of fruits without packaging and barcode scanning [2]. Nowadays, people attach great importance to their health. Eating fruits helps maintain health, and classifying fruits according to their effects can help people pick out the fruits that suit them, especially for patients who need conditioning.

Some researchers proposed some classic methods for the problem of fruit classification. Wei, L. [3] proposed to use biogeography-based optimization (BBO) to identify fruits. Tan, K. et al. [4] used histogram-oriented gradients and color features to identify blueberry fruits of different maturity in outdoor scenes. Nyarko, E. K. [5] proposed a k-nearest neighbor classifier based on convex detection for fruit recognition in RGB-D images. Aok, S [6] proposed a six-layer (6L) convolutional neural network for fruit classification. Li, Y. [7] proposed an improved hybrid genetic algorithm (IHGA) for fruit classification. Zhang, H. et al. [8] used volatile compounds in fruit peels as biomarkers to identify citrus species.Hassoon, I. M. [9] summarized the advantages and disadvantages of various shape-based feature extraction algorithms and techniques in fruit classification, classification and grading, and fruit quality evaluation. Ghazal, S. et al. [10] proposed a new fruit classification method combining Hue, Color-SIFT, Discrete Wavelet Transform, and Haralick features, which can better deal with the influence of rotation and light effect. SARI, A. C. [11] developed an app that can compare the quality of fruits by scanning real fruits and obtaining quality information and 3d images. Some successful image processing applications [12] [13] [14] [15] and artificial intelligence in other fields [16] [17] [18].

Although the methods above have achieved good results, they still have several flaws. For example, they need to collect features in advance and preprocess the images, which will waste lots of time. Secondly, the extracted features and selected indicators may not be accurate in different environments. With the emergence of convolutional neural network technology, some scholars began to use this method to solve the above problems and applied it to many fields, including face identification[19], cell segmentation [20], mechanical structural damage detection [21], and so on.

This paper builds a deep convolutional neural network to classify fruits. Compared with the traditional CNN model, we increased the number of convolution layers and used the ReLU activation function. By properly adjusting the parameters of the pre-trained model and the number of convolution layers, the accuracy and precision of image classification can be improved [22]. ReLU function can solve the problem of overfitting. The main contribution of the research is that the recognition system can identify images of different kinds of fruits mixed and further improve the accuracy of fruit classification. It reduces the classification time, thereby reducing the cost of classification.

The rest of this paper is organized as follows. Section 2 shows the dataset. Section 3 describes the standard convolutional neural network model. Section 4 discusses the CNN model we built and the experimental results. In Section 5, we summarize the research in this paper.

## 2. Dataset

The experimental dataset of this paper comes from the following three channels:

(i) http://images.google.com

(ii) http://images.baidu.com

(iii) Digital camera shooting.

After three months of collection and processing, we obtained 1,800 image datasets. The data set contains nine kinds of fruits, and the average data ratio of each fruit is 200. The nine fruits include Anjou pear, blackberry, black grape, blueberry, Bosque pear, cantaloupe, golden pineapple, Granny Smith apple, green grape. Figure 1 shows the sample of the fruit picture.

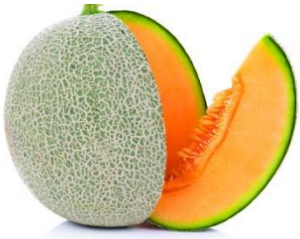
(a) Anjou pear

(b) blackberry

(c) black grape

(d) blueberry

(e) Bosque pear



(f) cantaloupe

(h) granny Smith apple

(i) green grape

**Figure 1.** The sample of fruit picture

## 3. Methodology

Deep learning and transfer learning are widely used in medical image analysis [23]. CNN is a technology that directly extracts features from images and then obtains accurate classification results by training and testing the extracted feature data. It is a feedforward neural network[24] with deep structure and convolution computation. CNN was first introduced in 1989, and until the 2012 ImageNet competition, it received more attention because of its excellent performance. The error rate was cut in half by applying CNN to a data set containing many images of different categories compared with the traditional best calculation method [25] [26] [27]. Our recognition system is based on CNN. The recognition process is shown in Figure 2.

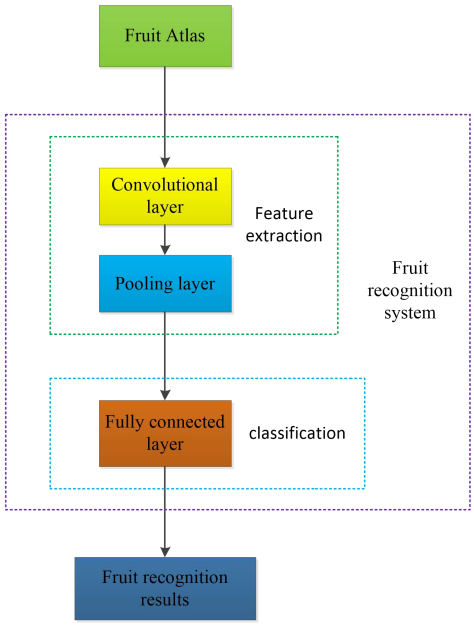



**Figure 2.** Fruit recognition process

CNN technology is the core of the fruit recognition system. Its standard architecture consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. The convolution layer is used to convolve the images. The pooling layer is used to reduce parameter dimensions and prevent overfitting. The fully connected layer maps all neurons in the previous layer [28]. However, according to the size of the data set, we can appropriately choose the number of layers of the CNN model. The CNN model shown in Figure 3 contains two convolutional layers, two pooling layers, and a fully connected layer.

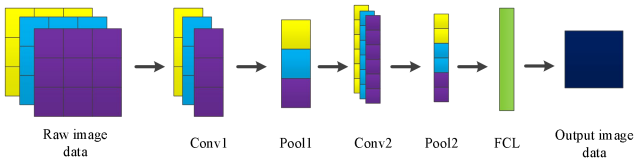



**Figure 3.** Multilayer convolutional neural network architecture

## 3.1. Convolution Layer

The role of the convolutional layer in the CNN model is extracting image feature information. Each convolutional layer has a different convolution kernel, and the images' characteristic data is convoluted by the convolution kernel [29]. The working principle of the convolution kernel is dividing the complete images into small pieces, which helps extract the feature pattern. The kernel uses a specific set of weights to convolve the images by multiplying its elements with the corresponding elements of the accepted domain [30].

The reasonable increase or decrease of the number of convolutional layers can not only adjust the amount of calculation in the training phase but also reduce the storage space requirement of the training model [31] [32] [33]. The basic operation of convolution is shown in Figure 4.

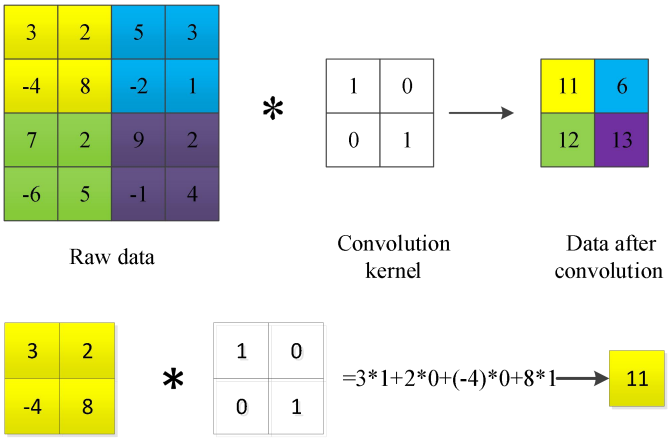



**Figure 4.** The basic operation of convolution

Suppose $R$ represents the original image data, and $K$ is the convolution kernel, then the convolution operation is as follows

$$P(x,y)=\sum_i\sum_j K(i,j)R(x-1,y-1) \qquad (1)$$

where $(i,j)$ is the size of the convolution kernel, and $(x,y)$ is the index of the original image.

## 3.2. Pooling

After the image data is passed into the pooling layer through convolution operation, the role of pooling is reducing the number of parameters in the training model, which can try to avoid overfitting [34] [35] [36] [37] [38]. The most commonly used pooling layers are the max-pooling and average pooling layers. The max-pooling operation returns the max value of the feature map, and the average pooling operation returns the average value. The basic operations of max pooling and average pooling are shown in Figure 5 and Figure 6.

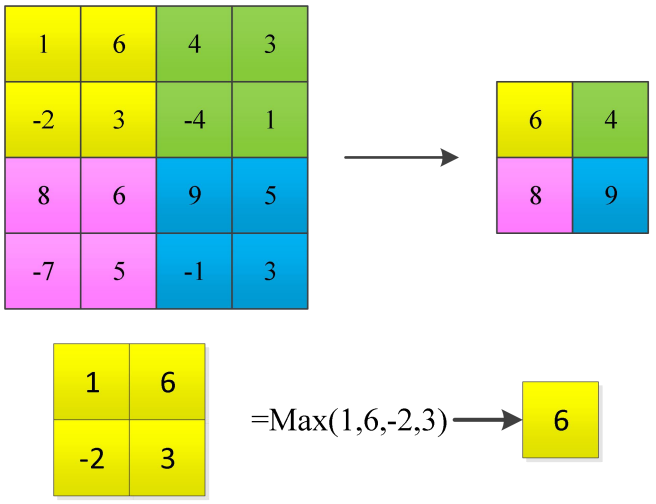



**Figure 5.** The basic operation of max pooling

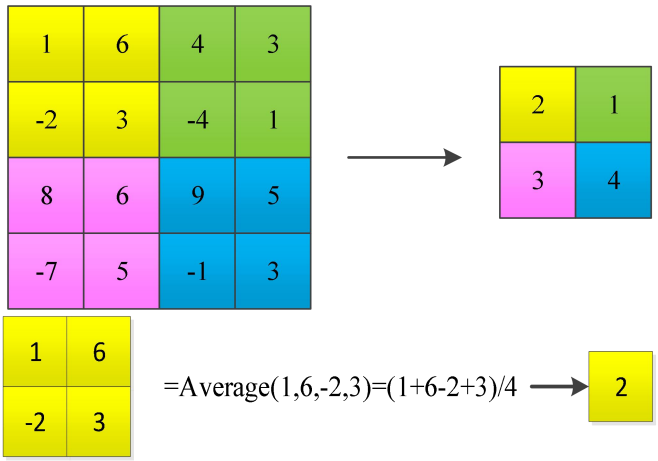



**Figure 6.** The basic operation of average pooling

If the pooling data area is $D$, then the operation results of max pooling and average pooling are as follows

$$\text{Pooling}_{Max} = Max\{D\} \tag{2}$$

$$\text{Pooling}_{Average} = \frac{\sum D}{|D|} \tag{3}$$

## 3.3. Fully Connected Layer

In the CNN model, convolution and pooling are used for feature extraction, and the purpose of the fully connected layer is image classification [39] [40] [41] [42]. Any neuron in the fully connected layer will be fully connected to the neurons in the adjacent layer. The structure of the fully connected layer is shown in Figure 7.

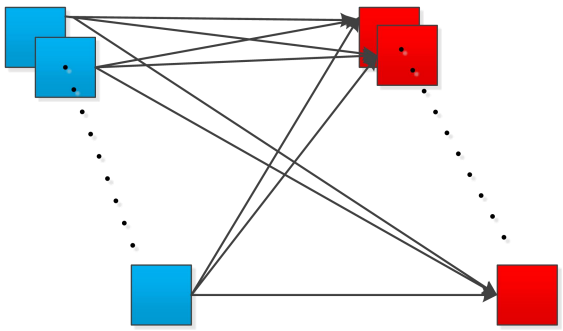


**Figure 7.** The basic operation of average pooling

## 3.4. ReLU

Rectified linear unit (ReLU) as an activation function in neural networks is generally used to solve the gradient descent problem in CNN models. Compared with leaky ReLU, Theoretically speaking, leaky ReLU has a better effect than the ReLU function, but a large number of practices have proved that its effect is unstable, so the application of this function is not much in practice. Due to inconsistent results from different functions applied in different intervals, it will be impossible to provide consistent relationship prediction for positive and negative numbers with the same input absolute values. The ReLU function has a sparse activation probability and can create a sparse representation of the data, which is very helpful for classification[43] [44] [45]. The calculation formula of ReLU is as follows, and the activation curve is shown in Figure 8.

$$\mathrm{ReLU}(\delta) = \begin{cases} \delta & \delta > 0 \\ 0 & \delta \le 0 \end{cases} \tag{4}$$

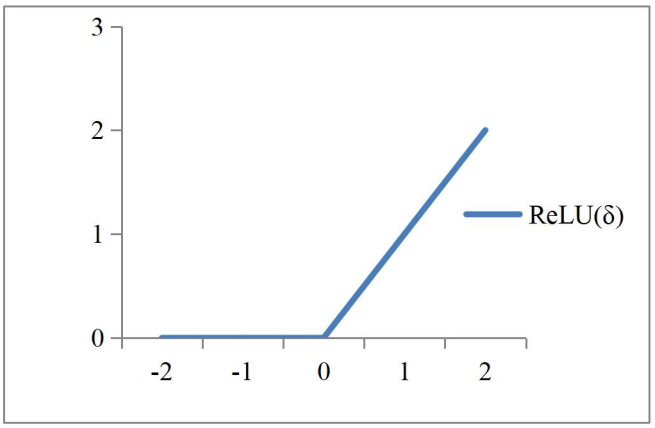



**Figure 8.** Activation curve of ReLU function

## 3.5. Training algorithms

Using the optimizer to update the parameters of each layer in the neural network helps optimize the deep convolutional neural network model to obtain the ideal experimental results. We use SGDM (Stochastic Gradient Descent with Momentum) and RMSProp (Root Mean Square Propagation) in the experiment to optimize the model and compare their effects.

The SGDM is a relatively popular deep learning optimization method. In experiments, the momentum term is usually set to 0.9, which helps to suppress oscillation [46]. The algorithm is defined as follows:

$$\omega_{t+1} = \omega_t - (xh_{t-1} + h_t)\varepsilon \tag{5}$$

where $h_t$ represents the gradient with respect to the current parameter at time $t$ and $\varepsilon$ is the learning rate, $x$ is a hyperparameter to control the moment.

RMSProp is very similar to momentum because it helps to eliminate the direction of large swings and allows a higher learning rate to accelerate the algorithm's learning [47] [48]. The algorithm is defined as follows:

$$\omega_{t+1} = \omega_t - \frac{\varepsilon}{\sqrt{X[h_t]^2 + \delta}} h_t \tag{6}$$

where, $X[h_t]^2$ is the exponentially decaying average of squared gradients, while $\delta$ is a vector of small numbers to avoid division by zero.

## 4. Experiment Design

Since the dataset in the experiment is not very large, the CNN model was tested by using 10-fold cross-validation. We divide the dataset into 10 equal parts. Nine of these datasets are used for training, and the remaining one is used for testing. The operation is shown in Figure 9. A total of 10 iterations are performed, and the final result is averaged over 10 iterations.

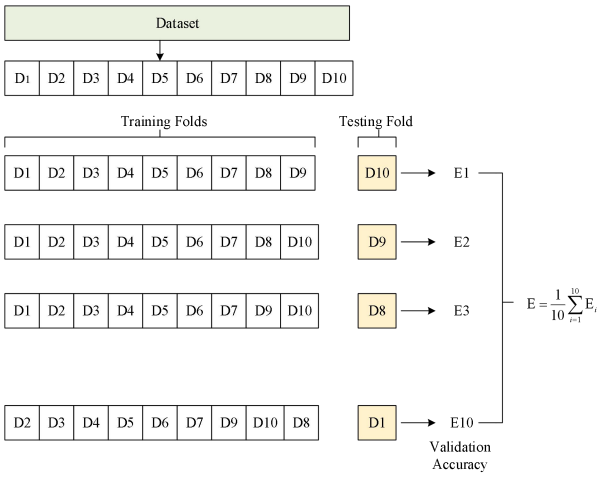



**Figure 9.** 10-fold cross validation operation

In the experiment, the 10-fold cross-validation will be run ten times, then the average of these ten times will be taken to detect the performance of the model.

We use overall accuracy (OA) as the performance index of the judgment model. The overall accuracy is the ratio between the number of correct predictions on all test sets and the overall number.

## 5. Experiment Results and Discussions

### 5.1. Training algorithms

In the experiment, the convolutional neural network model we built contains five convolutional layers and three fully connected layers. The convolutional and fully connected layers' parameters are shown in Table 1 and Table 2, respectively. The fill values of all layers are set to "same". The structure of the 8-layer convolutional neural network is shown in Figure 10.

Table 1. Parameters of the convolutional layer

| Layer | Filter size | #Channels | #Filters | Stride |
|---|---|---|---|---|
| Conv1 | 3×3 | 3 | 16 | 2 |
| Pool1 | 3×3 | | | 2 |
| Conv2 | 3×3 | 16 | 32 | 2 |
| Pool2 | 3×3 | | | 1 |
| Conv3 | 3×3 | 32 | 64 | 2 |
| Pool3 | 3×3 | | | 1 |
| Conv4 | 3×3 | 64 | 128 | 2 |
| Pool4 | 3×3 | | | 1 |
| Conv5 | 3×3 | 128 | 256 | 2 |
| Pool5 | 3×3 | | | 1 |

Table 2. Parameters of the fully connected layer.

| Layers | Weights | Bias |
|---|---|---|
| F1 | 50×8192 | 50×1 |
| F2 | 30×50 | 30×1 |
| F3 | 9×30 | 9×1 |

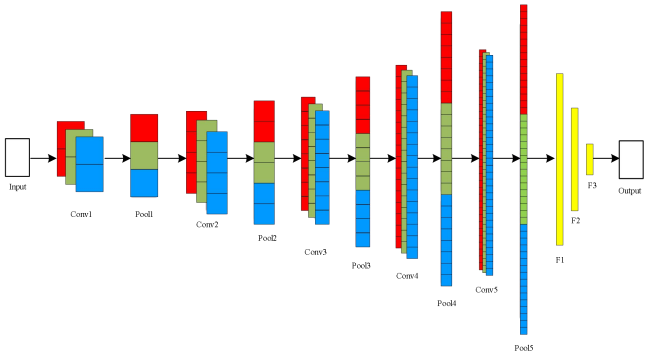



**Figure 10.** The structure of an 8-layer convolutional neural network

### 5.2. Training algorithms

We use this 8-layer CNN, then employ max pooling and RMSProp training algorithm. The results are shown in Table 3 and intuitively displayed in Figure 11. After 10 times 10-fold cross calculation, the overall accuracy of the CNN model reaches 92.52%. This is a very good result.

Table 3. Overall accuracy of 10x10-fold runs (using RMSProp)

| Run | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|
| 1 | 87.78 | 87.22 | 90.00 | 93.89 | 91.11 |
| 2 | 88.89 | 91.11 | 92.78 | 91.11 | 96.67 |
| 3 | 93.33 | 91.11 | 96.67 | 90.00 | 91.11 |
| 4 | 95.56 | 88.89 | 93.33 | 93.89 | 93.33 |
| 5 | 91.67 | 90.56 | 91.11 | 91.11 | 93.89 |
| 6 | 92.22 | 94.44 | 94.44 | 90.56 | 91.67 |
| 7 | 92.78 | 89.44 | 88.33 | 93.33 | 89.44 |
| 8 | 92.78 | 91.11 | 93.33 | 93.89 | 90.56 |
| 9 | 92.78 | 91.67 | 91.11 | 86.67 | 88.89 |
| 10 | 93.33 | 88.89 | 96.67 | 91.67 | 95.56 |
| Mean+SD | 92.11 ±2.25 | 90.44 ±1.97 | 92.78 ±2.72 | 91.61 ±2.29 | 92.22 ±2.57 |

Refer to Table 3(continued)

| Run | F6 | F7 | F8 | F9 | F10 |
|---|---|---|---|---|---|
| 1 | 93.33 | 91.11 | 92.22 | 91.67 | 92.22 |
| 2 | 93.33 | 95.00 | 95.00 | 90.56 | 86.11 |
| 3 | 91.11 | 91.11 | 91.67 | 88.33 | 89.44 |
| 4 | 93.33 | 93.89 | 91.67 | 91.11 | 95.00 |
| 5 | 90.56 | 91.11 | 89.44 | 89.44 | 88.89 |
| 6 | 94.44 | 89.44 | 93.89 | 91.11 | 89.44 |
| 7 | 86.11 | 93.33 | 90.56 | 84.44 | 90.56 |
| 8 | 93.89 | 89.44 | 88.89 | 91.67 | 95.00 |
| 9 | 96.11 | 93.33 | 91.67 | 95.00 | 92.22 |
| 10 | 92.22 | 88.33 | 93.89 | 92.22 | 87.78 |
| Mean+SD | 92.44 ±2.74 | 91.61 ±2.20 | 91.89 ±1.96 | 90.56 ±2.77 | 90.67 ±2.94 |

Refer to Table 3(continued)

| Run | Total |
|---|---|
| 1 | 91.06 |
| 2 | 92.06 |
| 3 | 91.39 |
| 4 | 93.00 |
| 5 | 90.78 |
| 6 | 92.17 |
| 7 | 89.83 |
| 8 | 92.06 |
| 9 | 91.94 |
| 10 | 92.06 |
| Mean+SD | 91.63 ± 0.89 |

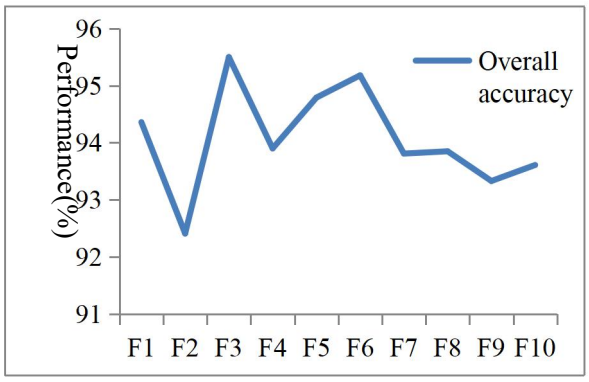



**Figure 11.** The structure of an 8-layer convolutional neural network

## 5.3. Pooling comparison

At the same time, we also try to use average pooling and the RMSProp training algorithm in the model. The results are shown in Table 4 and intuitively displayed in Figure 12. The overall accuracy achieved by the CNN model combined with average pooling is 91.57%, which proves that using max-pooling in the model can achieve better results than using average pooling.

Table 4. Overall accuracy of 10x10-fold runs (using average pooling)

| Run | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|
| 1 | 87.78 | 90.56 | 92.22 | 90.00 | 93.89 |
| 2 | 88.89 | 94.44 | 88.33 | 92.22 | 88.89 |
| 3 | 92.22 | 85.56 | 88.89 | 90.00 | 89.44 |
| 4 | 91.11 | 91.67 | 90.00 | 90.56 | 90.56 |
| 5 | 92.78 | 90.00 | 89.44 | 93.89 | 90.00 |
| 6 | 90.00 | 89.44 | 90.00 | 92.22 | 90.56 |
| 7 | 87.22 | 88.33 | 90.56 | 91.67 | 88.89 |
| 8 | 89.44 | 93.89 | 92.22 | 92.22 | 91.11 |
| 9 | 92.78 | 91.67 | 91.11 | 90.00 | 93.33 |
| 10 | 85.56 | 93.33 | 95.56 | 91.11 | 91.67 |
| Mean+SD | 89.78 ±2.47 | 90.89 ±2.72 | 90.83 ±2.10 | 91.39 ±1.29 | 90.83 ±1.72 |

Refer to Table 4(continued)

| Run | F6 | F7 | F8 | F9 | F10 |
|---|---|---|---|---|---|
| 1 | 92.78 | 92.78 | 91.67 | 86.11 | 92.78 |
| 2 | 92.22 | 93.89 | 88.89 | 88.89 | 91.67 |
| 3 | 91.67 | 86.67 | 93.89 | 89.44 | 93.33 |
| 4 | 92.22 | 91.11 | 95.00 | 90.56 | 95.56 |
| 5 | 91.11 | 87.78 | 90.56 | 89.44 | 93.89 |
| 6 | 88.89 | 86.67 | 90.00 | 86.67 | 91.67 |
| 7 | 91.67 | 90.56 | 89.44 | 94.44 | 92.78 |
| 8 | 91.11 | 85.56 | 87.78 | 96.11 | 88.33 |
| 9 | 93.33 | 90.56 | 86.11 | 91.11 | 87.78 |
| 10 | 93.33 | 87.22 | 93.89 | 96.11 | 92.78 |
| Mean+SD | 91.83 ±1.31 | 89.28 ±2.87 | 90.72 ±2.89 | 90.89 ±3.59 | 92.06 ±2.39 |

Refer to Table 4(continued)

| Run | Total |
|---|---|
| 1 | 91.06 |
| 2 | 90.83 |
| 3 | 90.11 |
| 4 | 91.83 |
| 5 | 90.89 |
| 6 | 89.61 |
| 7 | 90.56 |
| 8 | 90.78 |
| 9 | 90.78 |
| 10 | 92.06 |
| Mean+SD | 90.85 ±0.72 |

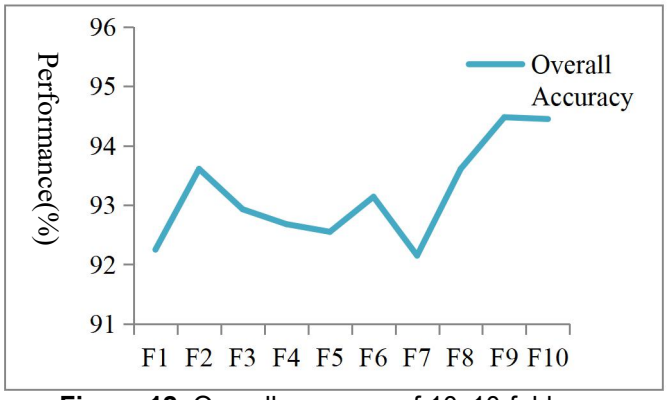



**Figure 12.** Overall accuracy of 10x10-fold runs (using average pooling)

## 5.4. Training algorithm comparison

In order to judge the RMSProp training algorithm and SGDM training algorithm, which can provide the best optimization model, we employ the SGDM training algorithm and max-pooling in the CNN model then perform ten times 10-fold cross calculation. The results are shown in Table 5 and intuitively displayed in Figure 13. Although the overall accuracy reached 91.21%, the RMSProp training algorithm has better overall accuracy in optimizing the 8-layer CNN model used in this experiment.

Table 5. Overall accuracy of 10x10-fold runs (using average pooling)

| Run | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|
| 1 | 87.22 | 91.67 | 92.22 | 90.56 | 91.11 |
| 2 | 87.78 | 86.11 | 90.00 | 93.33 | 87.78 |
| 3 | 85.00 | 89.44 | 92.78 | 88.33 | 92.22 |
| 4 | 86.11 | 85.56 | 87.78 | 90.56 | 89.44 |
| 5 | 90.56 | 94.44 | 87.22 | 89.44 | 90.56 |
| 6 | 88.89 | 92.78 | 87.78 | 90.56 | 93.33 |
| 7 | 90.56 | 83.89 | 91.11 | 87.78 | 88.33 |
| 8 | 90.56 | 89.44 | 87.22 | 89.44 | 93.33 |
| 9 | 87.22 | 90.00 | 88.89 | 91.67 | 88.33 |
| 10 | 89.44 | 93.89 | 92.78 | 93.89 | 92.22 |
| Mean+SD | 88.33 | 89.72 | 89.78 | 90.56 | 90.67 |
| | ±1.98 | ±3.61 | ±2.30 | ±1.98 | ±2.11 |

Refer to Table 5(continued)

| Run | F6 | F7 | F8 | F9 | F10 |
|---|---|---|---|---|---|
| 1 | 92.22 | 88.89 | 96.11 | 92.78 | 92.22 |
| 2 | 89.44 | 85.00 | 91.67 | 96.11 | 88.33 |
| 3 | 84.44 | 91.11 | 88.33 | 86.67 | 95.56 |
| 4 | 93.89 | 93.89 | 90.56 | 87.22 | 87.78 |
| 5 | 92.78 | 93.33 | 91.11 | 92.78 | 88.33 |
| 6 | 88.89 | 88.89 | 87.78 | 87.22 | 91.11 |
| 7 | 88.89 | 87.22 | 87.22 | 83.33 | 86.11 |
| 8 | 93.33 | 87.78 | 86.67 | 92.22 | 85.56 |
| 9 | 87.78 | 91.11 | 91.11 | 88.33 | 87.78 |
| 10 | 91.67 | 92.78 | 90.00 | 92.78 | 91.11 |
| Mean+SD | 90.33 | 90.00 | 90.06 | 89.94 | 89.39 |
| | ±2.97 | ±2.92 | ±2.78 | ±3.93 | ±3.07 |

Refer to Table 5(continued)

| Run | Total |
|---|---|
| 1 | 91.50 |
| 2 | 89.56 |
| 3 | 89.39 |
| 4 | 89.28 |
| 5 | 91.06 |
| 6 | 89.72 |
| 7 | 87.44 |
| 8 | 89.56 |
| 9 | 89.22 |
| 10 | 92.06 |
| Mean+SD | 89.88 |
| | ±1.33 |

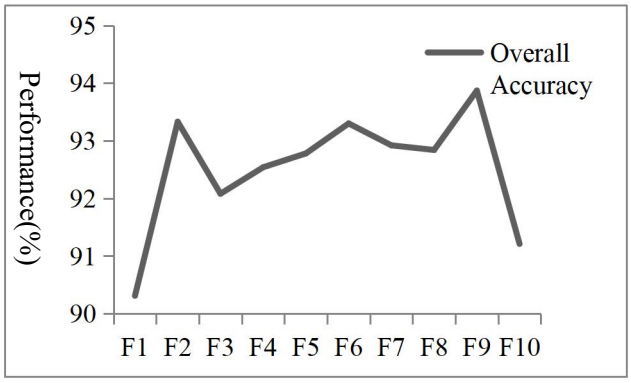



**Figure 13.** Overall accuracy of 10x10-fold runs (using SGDM)

## 5.5. Comparison of Different Number of Conv Layers

Based on the 3 fully connected layers, we set different numbers of convolutional layers in the model for experimental comparison. The results are shown in Table 6, Table 7, and Figure 14. Experiments show that using 5 convolutional layers in the model can obtain better detection results.

Table 6. Comparison of Different Number of Conv Layers

| Structure | Conv layers | FCL layers | Overall Accuracy |
|---|---|---|---|
| Setting I | 3 | 3 | 90.77± 0.69 |
| Setting II | 4 | 3 | 91.13± 1.12 |
| Ours | **5** | **3** | 91.63± 0.89 |
| Setting III | 6 | 3 | 91.05± 1.42 |

Table 7. The overall accuracy of different number of Conv Layers

| Run | Setting I | Setting II | Ours | Setting III |
|---|---|---|---|---|
| 1 | 91.61 | 92.33 | 91.06 | 88.44 |
| 2 | 91.39 | 92.06 | 92.06 | 89.56 |
| 3 | 91.00 | 89.06 | 91.39 | 92.61 |
| 4 | 90.22 | 90.78 | 93.00 | 91.94 |
| 5 | 89.72 | 91.28 | 90.78 | 92.89 |

Refer to Table 7(continued)

| Run | Setting I | Setting II | Ours | Setting III |
|---|---|---|---|---|
| 6 | 91.72 | 92.17 | 92.17 | 90.78 |
| 7 | 90.67 | 89.78 | 89.83 | 92.83 |
| 8 | 90.11 | 91.06 | 92.06 | 90.22 |
| 9 | 90.17 | 90.56 | 91.94 | 90.89 |
| 10 | 91.06 | 92.28 | 92.06 | 90.33 |
| Mean+SD | 90.77 ± 0.69 | 91.13 ± 1.12 | 91.63 ± 0.89 | 91.05 ± 1.42 |

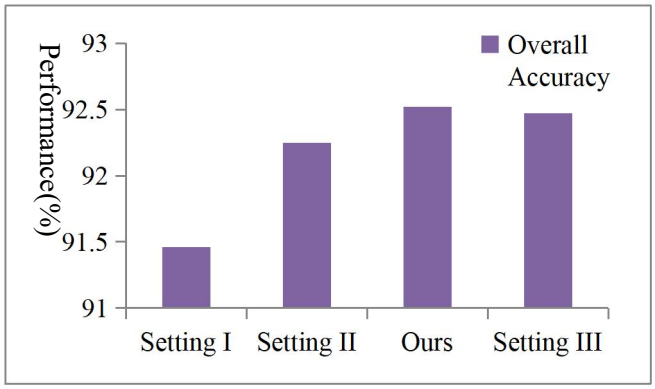



**Figure 14.** Overall accuracy of different number of Conv Layers

## 5.6. Comparison of different number of FCLs

In the same case where the number of convolutional layers is 5, we set up different numbers of fully connected layers for experimental comparison. The results are shown in Table 8 and Table 9 and are clearly shown in Figure 15. We found that the effect achieved by using three fully connected layers in the model is the best.

Table 8. Comparison of Different Number of FCLs

| Structure | Conv layers | FCL layers | Overall Accuracy |
|---|---|---|---|
| Setting IV | 5 | 1 | 88.61± 0.96 |
| Setting V | 5 | 2 | 90.87± 0.68 |
| Ours | **5** | **3** | 91.63± 0.89 |
| Setting VI | 5 | 4 | 91.08± 0.76 |
| Setting VII | 5 | 5 | 89.82± 1.28 |

Table 9. Overall accuracy of different number of FCLs

| Run | Setting IV | Setting V | Ours | Setting VI | Setting VII |
|---|---|---|---|---|---|
| 1 | 88.89 | 91.61 | 91.06 | 90.39 | 86.78 |
| 2 | 88.22 | 91.00 | 92.06 | 91.56 | 89.39 |
| 3 | 89.44 | 91.00 | 91.39 | 91.72 | 91.11 |
| 4 | 87.56 | 90.61 | 93.00 | 90.94 | 89.78 |
| 5 | 88.89 | 91.78 | 90.78 | 91.33 | 90.78 |
| 6 | 89.28 | 91.28 | 92.17 | 91.44 | 90.28 |
| 7 | 89.00 | 90.56 | 89.83 | 91.44 | 91.33 |
| 8 | 87.22 | 91.17 | 92.06 | 92.06 | 89.33 |
| 9 | 87.44 | 89.56 | 91.94 | 90.33 | 89.56 |
| 10 | 90.11 | 90.11 | 92.06 | 89.61 | 89.83 |
| Mean+SD | 88.61 ± 0.96 | 90.87 ± 0.68 | 91.63 ± 0.89 | 91.08 ± 0.76 | 89.82 ±1.28 |

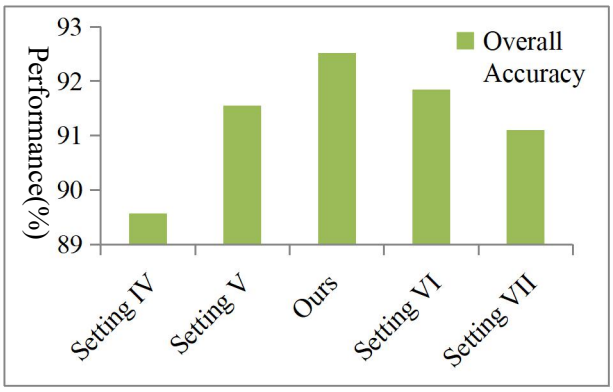



**Figure 15.** The overall accuracy of different numbers of FCLs

## 5.7. Comparison to State-of-the-art Approaches

In the experiment, we compared our method with four state-of-the-art methods: kSVM [2], 6-layer CNN[6], BBO [3], and IHGA [7]. The results are shown in Table 10 and intuitively expressed in a histogram in Figure 16. It can be seen that our method has high accuracy. Based on the same dataset, the classification effect of our proposed 8-layer CNN is the best. The overall accuracy reached 91.63%.

Table 10. Comparison to State-of-the-art Approaches

| Approach | No. of classes | Overall Accuracy |
|---|---|---|
| kSVM [2] | 18 | 88.20% |
| 6-layer CNN [6] | 9 | 91.44% |
| BBO [3] | 18 | 89.47% |
| IHGA [7] | 18 | 89.59% |
| 9L-CNN (Ours) | 9 | 91.63% |

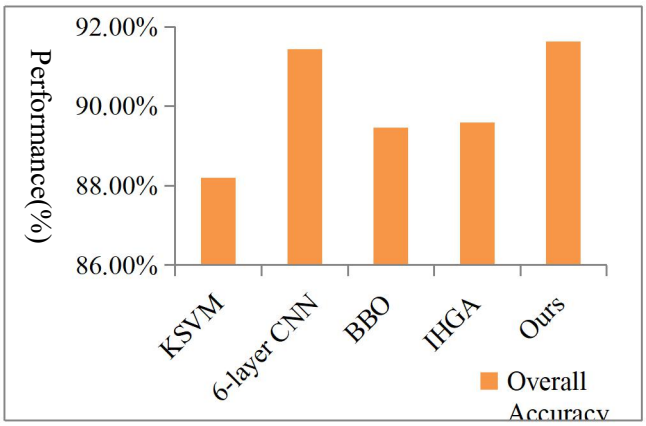



**Figure 16.** Comparison to State-of-the-art approaches

## 6. Conclusions

The paper built an eight-layer convolutional neural network model combined with the RMSProp optimization algorithm

for the fruit classification recognition system. The main contributions are as follows:

(i) We first used eight-layer CNN for the fruit classification recognition system, improving classification accuracy.

(ii) The RMSProp optimization algorithm can improve the stability of the CNN model.

(iii) Our proposed CNN model is superior to the four latest methods.

In future work, we will conduct fruit classification experiments based on more fruit pictures and further improve the detection accuracy by building an ideal model.

## References


[1] ZHANG, Y.-D., et al. (2017). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications* **78**, 3613-3632.

[2] WU, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. *Sensors* **12**, 12489-12505.

[3] WEI, L. (2015). Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization. *Entropy* **17**, 5711-5728.

[4] TAN, K., et al. (2018). Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes. *Biosystems Engineering* **176**, 59-72.

[5] NYARKO, E.K., et al. (2018). A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces. *Expert Systems with Applications* **114**, 454-466.

[6] AOK, S. (2018). Fruit classification based on six layer convolutional neural network. In *23rd International Conference on Digital Signal Processing (DSP)* IEEE, Shanghai, China, 1-5.

[7] LI, Y. (2017). A Fruit Sensing and Classification System by Fractional Fourier Entropy and Improved Hybrid Genetic Algorithm. In *5th International Conference on Industrial Application Engineering (IIAE)*, Kitakyushu, Japan, 293-299.

[8] ZHANG, H., et al. (2019). Volatile Compounds in Fruit Peels as Novel Biomarkers for the Identification of Four Citrus Species. *Molecules* **24**.

[9] HASSOON, I.M. (2021). Shape Feature Extraction Techniques for Fruits: A Review. *Iraqi Journal of Science*, 2425-2430.

[10] GHAZAL, S., et al. (2021). Analysis of visual features and classifiers for Fruit classification problem. *Computers and Electronics in Agriculture* **187**, 106267.

[11] SARI, A.C., et al. (2021). Fruit classification quality using convolutional neural network and augmented reality. *Journal of Theoretical and Applied Information Technology* **99**.

[12] WU, L.N. (2008). Improved image filter based on SPCNN. *Science In China Series F-Information Sciences* **51**, 2115-2125.

[13] WU, L.N. (2008). Pattern Recognition via PCNN and Tsallis Entropy. *Sensors* **8**, 7518-7529.

[14] WU, L.N. (2009). Segment-based coding of color images. *Science In China Series F-Information Sciences* **52**, 914-925.

[15] YAO, X. AND HAN, J. (2021). COVID-19 Detection via Wavelet Entropy and Biogeography-Based Optimization **60**, 69-76.

[16] YAN, J. (2010). Find multi-objective paths in stochastic networks via chaotic immune PSO. *Expert Systems with Applications* **37**, 1911-1919.

[17] WEI, G. (2010). Color Image Enhancement based on HVS and PCNN. *SCIENCE CHINA Information Sciences* **53**, 1963-1976.

[18] HUO, Y. (2010). Feature Extraction of Brain MRI by Stationary Wavelet Transform and its Applications. *Journal of Biological Systems* **18**, 115-132.

[19] DANG, L., et al. (2018). Deep Learning Based Computer Generated Face Identification Using Convolutional Neural Network. *Applied Sciences* **8**, 2610.

[20] MBOGBA, M.K., et al. (2018). The application of convolution neural network based cell segmentation during cryopreservation. *Cryobiology* **85**, 95-104.

[21] WANG, G. (2018). Design of damage identification algorithm for mechanical structures based on convolutional neural network. *Concurrency and Computation: Practice and Experience* **30**, e4891.

[22] GANDO, G., et al. (2016). Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs. *Expert Systems with Applications* **66**, 295-301.

[23] HAN, J. AND HOU, S.-M. (2020). A multiple sclerosis recognition via hu moment invariant and artificial neural network trained by particle swarm optimization. In *International Conference on Multimedia Technology and Enhanced Learning* Springer, 254-264.

[24] HAN, J. AND HOU, S.-M. (2019). Multiple sclerosis detection via wavelet entropy and feedforward neural network trained by adaptive genetic algorithm. In *International Work-Conference on Artificial Neural Networks* Springer, 87-97.

[25] LV, Y.-D. (2018). Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling. *Journal of Medical Systems* **42**.

[26] HAN, L. (2018). Identification of Alcoholism based on wavelet Renyi entropy and three-segment encoded Jaya algorithm. *Complexity* **2018**.

[27] TANG, C. (2018). Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform. *Multimedia Tools and Applications* **77**, 22821-22839.

[28] PUNYANI, P., et al. (2019). Neural networks for facial age estimation: a survey on recent advances. *Artificial Intelligence Review* **53**, 3299-3347.

[29] KHATAMI, A., et al. (2020). A weight perturbation-based regularisation technique for convolutional neural networks and the application in medical imaging. *Expert Systems with Applications* **149**, 113196.

[30] KHAN, A., et al. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*.

[31] COTRIM, W.D.S., et al. (2020). Short convolutional neural networks applied to the recognition of the browning stages of bread crust. *Journal of Food Engineering* **277**, 109916.

[32] CHENG, H. (2018). Multiple sclerosis identification based on fractional Fourier entropy and a modified Jaya algorithm. *Entropy* **20**.

[33] PAN, C. (2018). Abnormal breast identification by nine-layer convolutional neural network with parametric

rectified linear unit and rank-based stochastic pooling. *Journal of Computational Science* **27**, 57-68.

[34] ER, M.J., et al. (2016). Attention pooling-based convolutional neural network for sentence modelling. *Information Sciences* **373**, 388-403.

[35] PAN, C. (2018). Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU. *Journal of Computational Science* **28**, 1-10.

[36] HUANG, C. (2018). Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network With Batch Normalization, Dropout, and Stochastic Pooling. *Frontiers in Neuroscience* **12**.

[37] ZHANG, Y.-D. (2021). A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. *Machine Vision and Applications* **32**.

[38] ZHANG, Y.D. (2020). A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling. In *IEEE Sensors Journal*, 1-1.

[39] VOKHIDOV, H., et al. (2016). Recognition of Damaged Arrow-Road Markings by Visible Light Camera Sensor Based on Convolutional Neural Network. *Sensors (Basel)* **16**.

[40] ZHAO, G. (2018). Polarimetric synthetic aperture radar image segmentation by convolutional neural network using graphical processing units. *Journal of Real-Time Image Processing* **15**, 631-642.

[41] GOVINDARAJ, V.V. (2019). High performance multiple sclerosis classification by data augmentation and AlexNet transfer learning model. *Journal of Medical Imaging and Health Informatics* **9**, 2012-2021.

[42] CHEN, Y. (2020). Cerebral micro - bleeding identification based on a nine-layer convolutional neural network with stochastic pooling. *Concurrency and Computation: Practice and Experience* **31**, e5130.

[43] LIEW, S.S., et al. (2016). Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing* **216**, 718-734.

[44] SANGAIAH, A.K. (2020). Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. *Neural Computing and Applications* **32**, 665-680.

[45] WANG, S.-H. (2020). DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. *ACM Transactions on Multimedia Computing Communications and Applications* **16**.

[46] BERA, S. AND SHRIVASTAVA, V.K. (2019). Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *International Journal of Remote Sensing* **41**, 2664-2683.

[47] JIANG, X. (2020). Fingerspelling identification for Chinese sign language via AlexNet-based transfer learning and Adam optimizer. *Scientific Programming* **2020**;WANG, S.-H. (2021). Covid-19 Classification by FGCNet with Deep Feature Fusion from Graph Convolutional Network and Convolutional Neural Network. *Information Fusion* **67**, 208-229.

[48] WANG, S.-H. (2021). COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis. *Information Fusion* **68**, 131-148.