Multimed Tools Appl (2019) 78:3613–3632
DOI 10.1007/s11042-017-5243-3

Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation

Yu-Dong Zhang[1,2] · **Zhengchao Dong**[3] · **Xianqing Chen**[4] · **Wenjuan Jia**[5] · **Sidan Du**[6] · **Khan Muhammad**[7] · **Shui-Hua Wang**[1]

Received: 20 June 2017 / Revised: 16 August 2017 / Accepted: 20 September 2017 /
Published online: 30 September 2017

Abstract Fruit category identification is important in factories, supermarkets, and other fields. Current computer vision systems used handcrafted features, and did not get good results. In this study, our team designed a 13-layer convolutional neural network (CNN). Three types of data augmentation method was used: image rotation, Gamma correction, and noise injection. We also compared max pooling with average pooling. The stochastic

Highlights • We proposed a 13-layer convolutional neural network, and validated the optimal number of convolution layers and pooling layers.
• We validated that the max pooling gives better slight performance than average pooling.
• Our method yielded an overall accuracy of 94.94%, better than five state-of-the-art approaches.
• We tested our method on imperfect images. The overall accuracy over background fruit images is 89.60%, over decay images is 94.12%, over unfocused images is 91.03%, and over occlusion image is 92.55%.
• We compared CPU and GPU computation, and found GPU can achieve a 177× acceleration on training data, and a 175× acceleration on test data.
• We used five different types of data augmentation methods, and compared the classification performance of using data augmentation and not using data augmentation.

✉ Yu-Dong Zhang
yudongzhang@ieee.org

✉ Khan Muhammad
khan.muhammad@ieee.org

✉ Shui-Hua Wang
shuihuawang@ieee.org

[1] School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, People's Republic of China

[2] Jiangsu Key Laboratory of Advanced Manufacturing Technology, Huaiyin, Jiangsu 223003, China

[3] Translational Imaging Division & MRI Unit, Columbia University and New York State Psychiatric Institute, New York, NY 10032, USA

[4] Department of electrical engineering, College of engineering, Zhejiang Normal University, Jinhua, Zhejiang 321004, China

gradient descent with momentum was used to train the CNN with minibatch size of 128. The overall accuracy of our method is 94.94%, at least 5 percentage points higher than state-of-the-art approaches. We validated this 13-layer is the optimal structure. The GPU can achieve a 177× acceleration on training data, and a 175× acceleration on test data. We observed using data augmentation can increase the overall accuracy. Our method is effective in image-based fruit classification.

Keywords Convolutional neural network · Fully connected layer · Softmax · Fruit category identification

1 Introduction

Fruit classification is a challenge since it is difficult to provide a definition of a type of fruit. Nevertheless, fruit classification can help in factory automatic fruit-packing and transportation [12], supermarket price determination [9], and dietary guidance [10]. At present, there are two types of fruit classification: one is to identify specific type of fruit, and the other is to classify multiple fruit categories.

In the past, scholars tend to use near-infrared imaging [28], gas sensor [27], high-performance liquid chromatography [25] devices to scan the fruit. Nevertheless, those methods need expensive devices (different types of sensors) and professional operators, and their overall accuracies are commonly lower than 85% [20].

Image-based fruit classification has attracted attention of scholars since their cheap device (only a digital camera) and excellent performance. For example, Wu [37] used principal component analysis (PCA) to reduce the color, texture, and morphological features. They introduced a kernel support vector machine (KSVM) as the classifier. Their overall accuracy reached 88.20%. Adak and Yumusak [2] used artificial bee colony (ABC)-based neural network (NN). Ji [14] replaced the KSVM with a fitness-scaled chaotic ABC (FSCABC). Tovar and Losada [35] used fuzzy logic block to develop a fuzzy fruit classification system. Wei [36] employed wavelet entropy (WE) and biogeography-based optimization. The overall accuracy of their method was 89.47%. Wu [38] combined BBO with feedforward neural network (FNN). Garcia et al. [11] extracted color chromaticity, texture and shape features. Lu [21] used fractional Fourier entropy (FRFE) as the features, and they used back propagation neural network (BPNN) as the classifier. Lu and Li [22] replaced BPNN with an improved hybrid genetic algorithm (IHGA). Their method received an overall accuracy nearly to 90%.

To further improve the performance of image-based fruit classifiers, we analyzed past literature, and believe the problems lies in following points: (i) Past methods used manual features designed by experts, but these authors may not be the optimal. (ii) The classifiers are of simple structure, and may fail to map the complicated features to the final identification result.

Convolutional neural network (CNN) can solve above two problems, and it is the hottest research topic. In CNN, the neuron-connectivity patterns are inspired by the visual cortex of

[5] School of Computer Science and Technology, Nanjing Normal University, Nanjing, Jiangsu 210023, China

[6] School of Electronic Science and Engineering, Nanjing University, Nanjing, Jiangsu 210046, China

[7] College of Software Convergence, Sejong University, Seoul, Republic of Korea

mammals. CNN firstly showed its outstanding ability in object recognition in the ImageNet competition [39]. Later, CNN has been widely applied to analyze medical images, such as choroid segmentation [29], abnormality detection [6], carcinoma nuclei grading [19], solitary cysts discrimination [17], vibrational spectroscopic data analysis [1], etc. To our knowledge, CNN has not been applied to fruit image. Therefore, we would like to investigate the performance of using CNN in fruit classification.

The structure of the remainder is organized as follows: Section 2 provides the materials, preprocessing, and data augmentation. Section 3 describes the basics of convolutional neural network (CNN). Section 4 contains the experiments and results. Section 5 gives concluding remarks.

2 Materials

2.1 Dataset and preprocessing

The fruit dataset was obtained through three months: (i) 6 months of on-site collecting via digital camera, (ii) download from http://images.google.com; (iii) download from http://images.baidu.com. Finally, we obtain a 3600-image dataset with 200 image for each fruit type.

A four-step preprocessing technique was used. First, we move the fruit into the center of the image. Second, the image was cropped and resized to a 256 × 256 matrix. Third, split-and-merge algorithm [8] was used to remove the background. Fourth, we label each image manually to one of the 18 fruit types.

Figure 1 shows the preprocessed sample images of our clean fruit image dataset. The 18 types from Fig. 1a–r are respectively Anjou pear, blackberry, black grape, blueberry, Bosc pear, cantaloupe, golden pineapple, granny Smith apple, green grape, green plantain, Hass

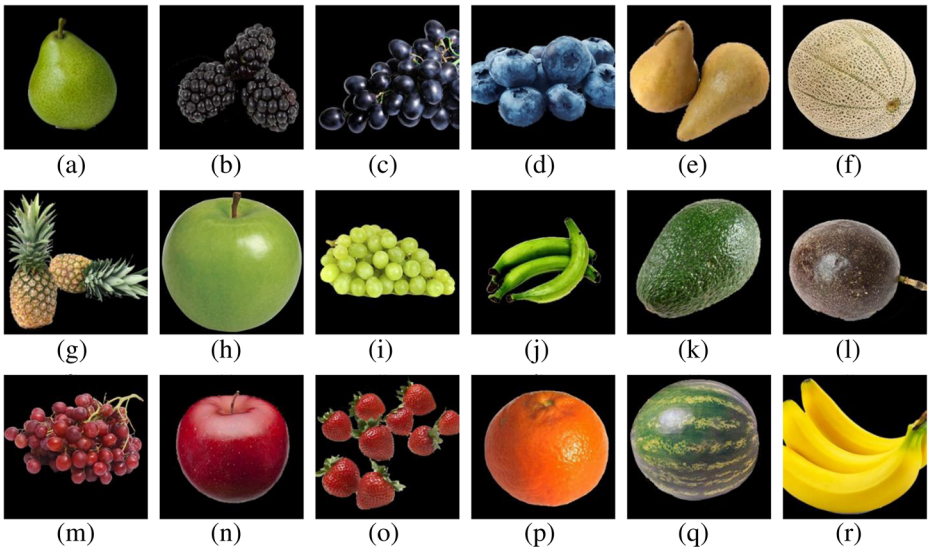


Fig. 1 Sample of our dataset of clean fruit images

Table 1 Original dataset

	Sample size of all classes
Training Set	1800
Test Set	1800
Total	3600

avocado, passion fruit, red grape, Rome apple, strawberry, tangerine, watermelon, and yellow banana.

2.2 Data augmentation

Small number of training samples may lead to overfitting [23]. One solution is to create fake data and add them to the training set. We divided original dataset into training set and test set randomly. The training and test set are of half size of original dataset as shown in Table 1. Usually less data is for training and more data for test, but considering we have a relatively large dataset, here we select 1800 for training set and the rest 1800 for test.

Afterwards, we created the fake samples based on the training set in following five means. The first data augmentation method is image rotation. The rotation angle θ is from −15° to 15° in step of 5°. Thus, we create new samples with size of 6 times of original training set. The second data augmentation method is gamma correction [31]. The gamma-value r varies from 0.7 to 1.3 with step of 0.1, again leading to new samples with size of 10 times of original training set. The third data generation method is noise injection [15]. We create 6 new noise-contaminated image for each image. The zero-mean Gaussian noise with variance of 0.01 was employed. The fourth is the scale transform, with scaling parameter S from 0.8 to 1.2 with increment of 0.05, generating 8 new images for each original image. Finally, the fifth is the affine transform, and we randomly generate eight different affine parameters for each original image. All the hyper-parameters selected here are following the experiences, which also be used in open published literature [13, 30].

3 Convolutional neural network

The convolutional neural network (CNN) with deep structures have gained tremendous success in text/non-text classification [4], human detection [16], ear detection [7], etc. Compared to traditional shallow neural networks, it has three important advantages: sparse

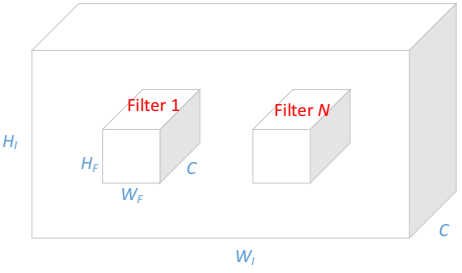


Fig. 2 Two-dimensional convolution operation

Table 2 Our CNN structure

Layer	Purpose	Filter	No. of filters	Stride	Padding	Weights	Bias	Activation
1	Image Input layer							256 × 256 × 3
2	Convolution + ReLU	7 × 7	40	[3 3]	[0 0]	7 × 7 × 3 × 40	1 × 1 × 40	84 × 84 × 40
3	Pooling	3 × 3		[3 3]	[0 0]			28 × 28 × 40
4	Convolution + ReLU	5 × 5	80	[3 3]	[2 2]	5 × 5 × 40 × 80	1 × 1 × 80	10 × 10 × 80
5	Pooling	3 × 3		[1 1]	[1 1]			10 × 10 × 80
6	Convolution + ReLU	3 × 3	120	[1 1]	[1 1]	3 × 3 × 80 × 120	1 × 1 × 120	10 × 10 × 120
7	Pooling	3 × 3		[1 1]	[1 1]			10 × 10 × 120
8	Convolution + ReLU	3 × 3	80	[1 1]	[1 1]	3 × 3 × 120 × 80	1 × 1 × 80	10 × 10 × 80
9	Pooling	3 × 3		[3 3]	[1 1]			4 × 4 × 80
10	Fully Connected					50 × 1280	50 × 1	1 × 1 × 50
11	Fully Connected					18 × 50	18 × 1	1 × 1 × 18
12	Softmax							1 × 1 × 18
13	Output							1 × 1 × 18

interaction, parameter sharing, and equivariance. It has shown significant gains over state-of-the-art classifiers, such as logistic regression, extreme learning machine, support vector machine and its variants, linear regression classifier, etc. A typical CNN will include convolution layer, nonlinear activation layer, and pooling layer.

3.1 Convolution layer

The convolution layer performs the two-dimensional convolution for three-dimensional input and three-dimensional filter. Suppose the size of the input is $H_I \times W_I \times C$, here H_I represents the height, W_I the width, and C the channels. Then, suppose the size of the filter is $H_F \times W_F \times C$, here HF and W_F represent the height and width of the filters. The channel size of both input

Table 3 CNN Training platform

Hardware	NVIDIA GeForce GTX 1050
	Compute capability = 6.1
	Clock rate = 1455 MHz
	Multiprocessors = 5
	Warp size = 32
	Registers per block = 65,536
	Threads per block = 1024
Software	Stochastic gradient descent with momentum
	minibatch size = 128
	Initial learning rate = 0.01
	Drop period = 10
	Drop rate factor = 0.1
	Momentum = 0.9
	Maximum epoch = 30
	Loss function = cross entropy

Table 4 Data augmentation of original dataset

	Sample size of new images	Sample size of each class
Data Augmented Training Set	63,000	3500
Original Training Set	1800	100
Image Rotation	10,800	600
Gamma Correction	10,800	600
Noise Injection	10,800	600
Scale Transform	14,400	800
Affine Transform	14,400	800
Test Set	1800	100
Total	64,800	3600

and filter should be equivalent; hence, the 2D convolution is implemented along the height and width directions (See Fig. 2).

Suppose the padding size at each margin is P, the stride size is S, we can calculate the height H_O and width W_O of output as

$$H_O = \frac{H_I - H_F + 2P}{S} + 1 \tag{1}$$

$$W_O = \frac{W_I - W_F + 2P}{S} + 1 \tag{2}$$

The output is also three dimensional with size of $H_O \times W_O \times N$, where N denotes the number of filters.

The neurons in the feature map after convolution layer will pass through a nonlinear activation function, such as a rectified linear unit (ReLU) layer, which carries out a ReLU function [5] as

$$\mathrm{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3}$$

3.2 Pooling layer

The pooling function replaces the outputs from the ReLU layer with a summary statistic of nearby outputs [3]. It has two advantages: (i) guarantee the representation become invariant to small translation of the input; (ii) help to reduce the computation burden.

(a) θ = -15 (b) θ = -10 (c) θ = -5 (d) θ = 5 (e) θ = 10 (f) θ = 15

Fig. 3 Generated images by image rotation

Fig. 4 Generated images by Gamma correction

Suppose the pooling region is R, the activation set A included in R is

$$A = \{a_i | i \in R\} \tag{4}$$

The max-pooling P_M [26] is the most popular pooling strategy. It is defined as

$$P_M = \max(A_R) \tag{5}$$

Average-pooling P_A [44] is another pooling technique defined as

$$P_A = \frac{\sum A_R}{|A_R|} \tag{6}$$

where |.| is the number of the elements in the set.

3.3 Network structure

The structure of our CNN is a 13-layer deep neural network. The details of each layer are shown in Table 2. The image input layer just inputs the preprocessed fruit image directly. The convolution layer, ReLU layer, and pooling layer are described before. We set the filter size and the number of filters by experiences.

The fully connected (FC) layer multiplies the input by a weight matrix and then adds a bias vector. The softmax layer used the softmax function, also known as the multiclass generalization of logistic regression. Suppose P(r) is the class prior probability, and $P(x|r)$ is the conditional probability of sample given class r. Then we can conclude that the probability of sample x belonging to class r is

$$P(r|x) = \frac{P(x|r)P(r)}{\sum_{k=1}^{R} P(x|k)P(k)} \tag{7}$$

Here R is the total number of classes. If we define A_r as

$$A_r = \ln(P(x, r)P(r)) \tag{8}$$

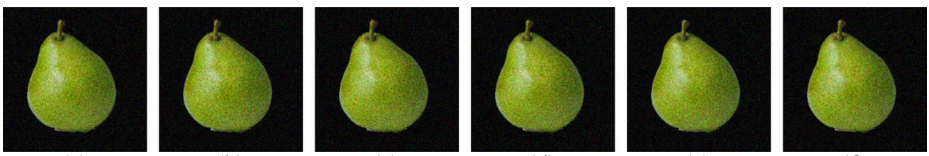


Fig. 5 Eight Generated images by noise injection

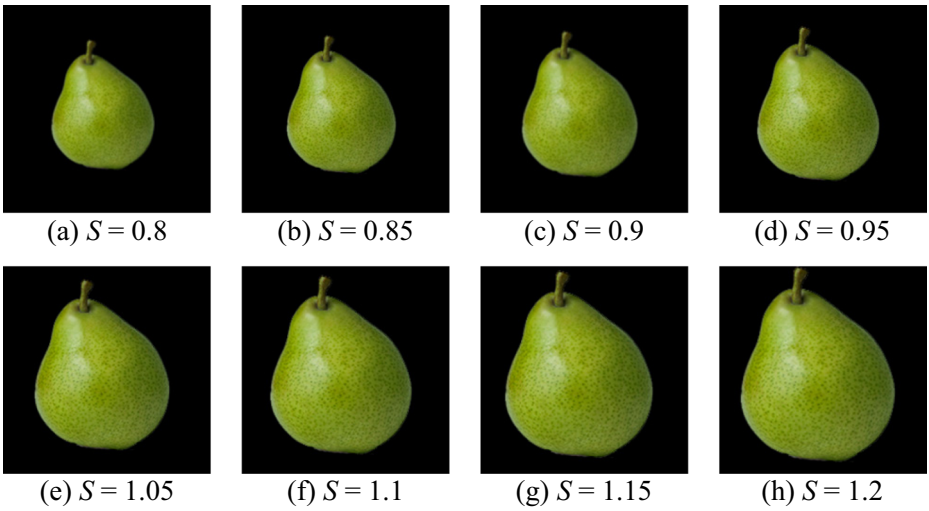


Fig. 6 Generated images by scale transform

Then we have

$$P(r|x) = \frac{\exp(A_r(x))}{\sum_{k=1}^{R} \exp(A_k(x))} \tag{9}$$

Finally, the output layer transforms the numerical result to categorical names with following rules: 1➔Anjou pear, 2➔blackberry, 3➔black grape, 4➔blueberry, 5➔Bosc pear, 6➔cantaloupe, 7➔golden pineapple, 8➔granny Smith apple, 9➔green grape, 10➔green plantain, 11➔Hass avocado, 12➔passion fruit, 13➔red grape, 14➔Rome apple, 15➔strawberry, 16➔tangerine, 17➔watermelon, and 18➔yellow banana.

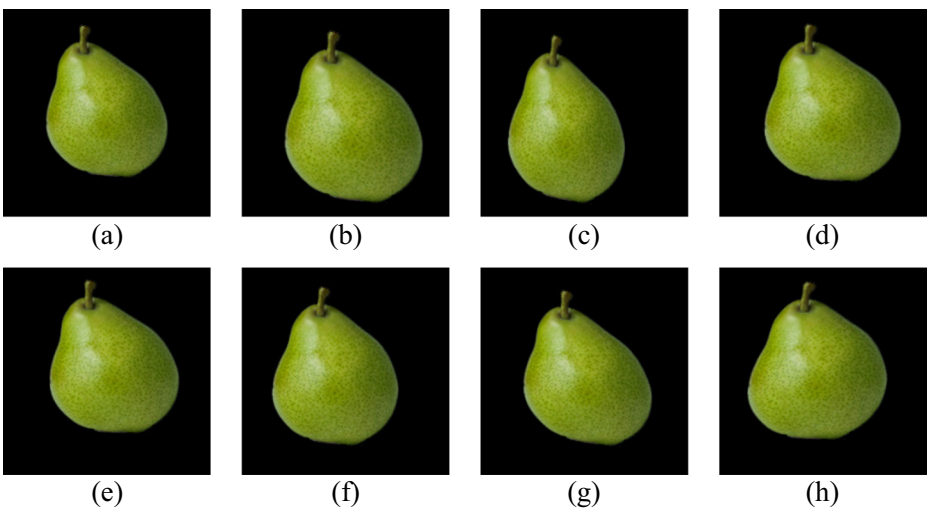
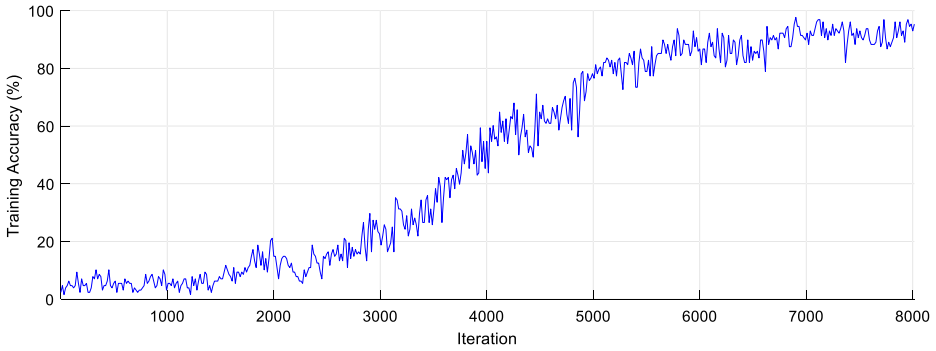
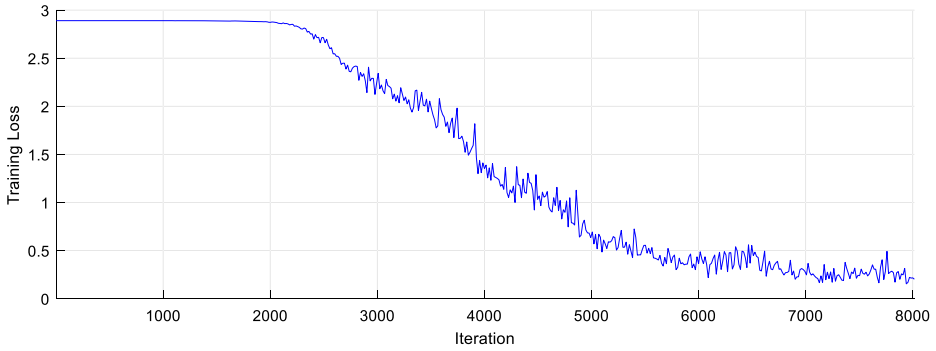


Fig. 7 Eight Generated images by randomly affine transform



(a) Training accuracy



(b) Training loss

Fig. 8 Training performance of CNN method

3.4 Training setting

The CNN Training was based on NVIDIA GeForce GTX 1050 with compute capability of 6.1, clock rate of 1455 MHz, and multiprocessors of 5. The training algorithm was stochastic gradient descent with momentum (SGDM). Here the "stochastic" represents the minibatch training method, in which we set its size to 128. The initial learning rate is set to 0.01, and was decreased by factor of 10 every 10 epochs. The momentum was set to 0.9. The maximum epochs was assigned with a value of 30. Cross entropy [24] was used as the loss function, since it is suitable for multiclass problem [18, 40]. Table 3 shows the hardware and software settings for training CNN. Here the hyperparameters are obtained by experiences, following the settings in open published literature [4, 7, 16]. Table 4 shows the amount of augmented training set and the test set.

4 Experiments and results

4.1 Illustration of data augmentation

In this experiment, we take the Anjou pear (Fig. 1a) as an example, and shows its generated images in below. Figures 3, 4, 5, 6, and 7 present the generated images by image rotation, gamma correction, noise injection, scale transform, and affine transform, respectively.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
C_1	94	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0
C_2	0	102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_3	0	10	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_4	0	1	0	92	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_5	0	0	0	0	93	0	3	0	0	0	0	0	0	0	0	0	0	0
C_6	0	0	0	0	0	87	7	0	0	0	0	1	0	0	0	0	0	0
C_7	0	0	0	0	0	0	105	0	0	0	3	0	0	0	0	2	0	0
C_8	9	0	0	0	0	0	1	88	4	2	0	0	0	0	0	0	0	0
C_9	2	0	0	0	0	0	0	0	93	0	0	0	0	0	0	0	0	0
C_{10}	1	0	0	0	0	0	0	1	0	104	0	0	0	0	0	0	0	0
C_{11}	0	0	0	0	0	0	1	0	0	0	111	0	0	0	0	0	1	0
C_{12}	0	0	9	0	0	0	0	0	0	0	0	74	0	0	1	0	0	0
C_{13}	0	0	0	0	0	0	0	0	0	0	0	2	92	0	3	0	0	0
C_{14}	0	0	0	0	0	0	0	0	0	0	0	0	1	87	0	0	0	0
C_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0	0
C_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	100	0	0
C_{17}	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	103	0
C_{18}	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	89

Fig. 9 Confusion matrix over test set

Using the data augmentation shown in Fig. 3 to Fig. 7, one fruit image can generate 34 new simulated images. In this way, the training dataset expands 34 + 1 = 35 times as large as original. This enlarged training set can help the deep learning learn more stable features than original training set.

Table 5 Performance of each class

Class	Sensitivity	Specificity	Precision	Accuracy	Class number
Anjou pear	91.3%	99.3%	88.7%	98.8%	103
Blackberry	100.0%	99.4%	90.3%	99.4%	102
Black grape	90.1%	99.5%	91.0%	98.9%	101
Blueberry	98.9%	100.0%	100.0%	99.9%	93
Bosc pear	96.9%	100.0%	100.0%	99.8%	96
Cantaloupe	91.6%	100.0%	100.0%	99.6%	95
Golden pineapple	95.5%	99.0%	86.1%	98.8%	110
Granny Smith apple	84.6%	99.4%	89.8%	98.6%	104
Green grape	97.9%	99.8%	95.9%	99.7%	95
Green plantain	98.1%	99.9%	98.1%	99.8%	106
Hass avocado	98.2%	99.2%	88.8%	99.1%	113
Passion fruit	88.1%	99.8%	96.1%	99.3%	84
red grape	94.8%	99.9%	98.9%	99.7%	97
Rome apple	98.9%	100.0%	100.0%	99.9%	88
Strawberry	100.0%	99.7%	95.4%	99.7%	104
Tangerine	99.0%	99.9%	98.0%	99.8%	101
Watermelon	90.4%	99.9%	99.0%	99.3%	114
Yellow banana	94.7%	100.0%	100.0%	99.7%	94
Average	94.94%	99.71%	95.34%	99.43%	100

Table 6 Pooling technique comparison

Pooling	Overall accuracy
Max-pooling	94.94%
Average-pooling	94.83%

4.2 Training of CNN

Here one "epoch" means a training over all samples, and "iteration" means a training over only 128 samples. The 30 epochs equal to 30 × 34,200/128 = 8015 iterations. The minibatch accuracy and loss versus iteration were shown in Fig. 8.

4.3 Confusion matrix

Using our data augmentation method, the confusion matrix over the test set was listed in Fig. 9. The sensitivity, specificity, precision, and accuracy over the test set was presented in Table 5. The overall accuracy of all classes is defined as the number of correctly identified fruit images divided by the number of the whole fruit images. The result of our overall accuracy is 94.94%. Note that overall accuracy is equal to the average sensitivity at the condition of equal or similar class numbers, which is our case as shown in Table 5.

Table 5 shows that the second class (blackberry) and fifteenth class (strawberry) can be identified perfectly with sensitivity of 100.0%. The fruit with worst performance is eighth class (Granny Smith apple). From the confusion matrix in Figs. 9, 9 Granny Smith apples were misclassified as Anjou pear, 1 Granny Smith apple was misclassified as Golden pineapple, four Granny Smith apples were misclassified as green grapes, and 2 Granny Smith apples were misclassified as green plantains.

4.4 Pooling technique comparison

In this experiment, we compared the max-pooling and average pooling technique. The results are shown below in Table 6.

The comparison results between max-pooling and average-pooling approaches in Table 6 showed max-pooling gives 0.11% better accuracy than average-pooling. Previous studies have shown that average pooling considered all elements in the region, hence, it will down-weight the strong activations. Nevertheless, the improvement of max-pooling in this study is slight. In the future, we shall test other pooling methods.

Table 7 CNN structure with different number of convolution layers

Number of combined layers	Overall accuracy
2	90.17%
3	94.06%
4 (Proposed)	94.94%
5	92.33%
6	92.00%
7	92.89%

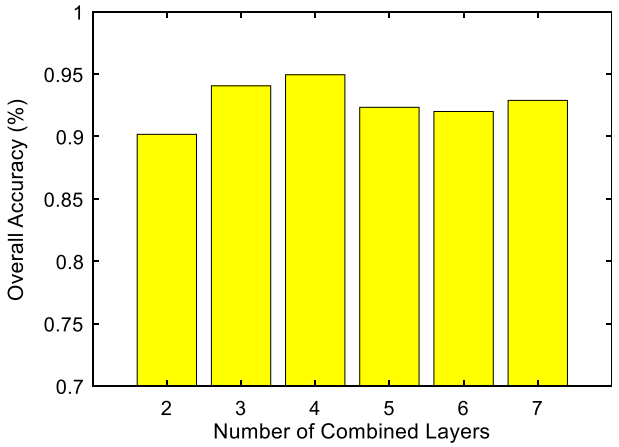


Fig. 10 Optimal number of convolution layers

4.5 Optimal structure of CNN

The convolution layer and pooling layer are the most important among all layers. Our CNN contains four combined layers (convolution layer and pooling layer) as shown in Table 2. In this experiment, we used grid searching method to find the optimal number of combined layers. The parameter setting here are the same as in Table 3. Finally, the overall accuracy results are presented in Table 7 and the corresponding bar-plot is pictured in Fig. 10.

We observed that CNN with 2, 3, 4, 5, 6, and 7 combined layers yielded an overall accuracy of 90.17%, 94.06%, 94.94%, 92.33%, 92.00%, and 92.89%, respectively. Hence, we select 4 combined layers in our proposed structure.

4.6 Comparison to state-of-the-art approaches

Finally, we compared our method with five state-of-the-art approaches: PCA + kSVM [37], PCA + FSCABC [14], WE + BBO [36], FRFE + BPNN [21], FRFE + IHGA [22]. The definition of those methods can be seen in the introduction. The comparison results are shown in Table 8.

The overall accuracy of our CNN method achieved 94.94% as shown in Table 8. It provides at least 5 percentage points higher than state-of-the-art approaches. For example, the PCA + kSVM [37] only obtains an accuracy of 88.20%, the PCA + FSCABC [14] yielded an accuracy of 89.11%, WE + BBO [36] yielded an accuracy of 89.47%, FRFE + BPNN [21] yielded an

Table 8 Comparison to state-of-the-art approaches

Approach	Overall accuracy
PCA + kSVM [37]	88.20%
PCA + FSCABC [14]	89.11%
WE + BBO [36]	89.47%
FRFE + BPNN [21]	88.99%
FRFE + IHGA [22]	89.59%
13-layer CNN (Our)	94.94%

Fig. 11 Imperfect samples of fruit images

accuracy of 88.99%, and FRFE + IHGA [22] yielded an accuracy of 89.59%. This superiority performance of CNN to traditional classifiers, again demonstrate the powerfulness of CNN.

The dataset was obtained by digital camera. A potential future direction is to use magnetic resonance imaging (MRI) [33, 34] to scan the fruit and obtain 3D volumetric image of fruits for identification. Another study is to use similarity measure [32] to help identify fruit categories.

4.7 Result on imperfect images

In this experiment, we tested our method on the imperfect images. We collected 173 fruit images with realistic complicated background, 136 fruit images with decay, 145 fruit images with camera not well focused, 161 fruit images with partially occluded by other stuffs. Those bad samples are shown in Fig. 11.

The overall accuracy using our method over these images are listed in Table 9. We can observe that the overall accuracy over background fruit images is 89.60%, over decay images is 94.12%, over unfocused images is 91.03%, and over occlusion image is 92.55%. We found that the overall accuracy over decay images is nearly the same as over our clean dataset. For the fruit image with complicated background, the performance deteriorates. In the future, we shall include those imperfect images to our dataset, so our trained CNN classifier can be generalized to identify them.

4.8 Time analysis

In this experiment, we compared the GPU-based computation with CPU-based computation. The CPU is Intel Core i5–3470 with frequency of 3.20GHz. We run the algorithm ten times, and calculate the average value. The computation time over training and test results are listed in Table 10.

For the training stage, the CPU costs 1,492,377.04 s, i.e., 414.55 h, while GPU costs 8415.75 s. The GPU reaches a 177× acceleration compared to CPU. Meanwhile, for the test stage, the CPU costs 92.48 s while GPU costs 0.53 s. Hence, GPU yields a 175× acceleration compared to CPU.

Table 9 Performance of our method over faulty image

Faulty type	Overall accuracy
Background	89.60%
Decay	94.12%
Unfocused	91.03%
Occlusion	92.55%

Table 10 Time analysis (Unit: second)

Training	Time (63,000 data augmented samples)
CPU	1,492,377.04
GPU	8415.75
Acceleration	177
Test	Time (1800 samples)
CPU	92.48
GPU	0.53
Acceleration	175

4.9 Effect of data augmentation

In this experiment, we checked the effect of data augmentation. Suppose "no augmentation (NA)" represents the training set contains only 1800 original image, and "data augmentation (DA)" represents the training set contains the 63,000 fruit images as presented in Table 4. The overall accuracies on different test set are shown below in Table 11 and Fig. 12.

From Table 11 and Fig. 12, we can observe that using data augmentation can significantly increase the classification performance in terms of overall accuracy, especially on imperfect images. The reason is the augmented image can help train the CNN to resist the attack of imperfection. Again, the fruit image with complicated background has the lowest overall accuracy of 84.39% when not using data augmentation.

5 Conclusion

In this study, our team developed a fruit classification method based on a 13-layer deep convolutional neural network. The experiments show that our CNN reached an overall accuracy of 94.94%, superior to five state-of-the-art approaches in terms of overall accuracy. Besides, the data augmentation expands the training data from 1800 to 63,000. The max pooling techniques performs slightly better than average pooling. We also validated the optimal structure of proposed CNN. We tested our classifier on imperfect images, and its overall accuracy decreased at most 5%. The time analysis showed GPU can achieve a 177× acceleration on training data, and a 175× acceleration on test data. We validated the effect of using data augmentation.

The shortcomings of our method are three-folds: (i) Our dataset is clean and hence it does not perform well on imperfect images. (ii) We validated the optimal combined layers of

Table 11 Effect of data augmentation

Test set	No. of image	Overall accuracy	
		DA	NA
Clean	1800	94.94%	93.72%
Background	173	89.60%	84.39%
Decay	136	94.12%	86.03%
Unfocused	145	91.03%	86.90%
Occlusion	161	92.55%	86.34%

(*DA* Data Augmentation, *NA* No Augmentation)

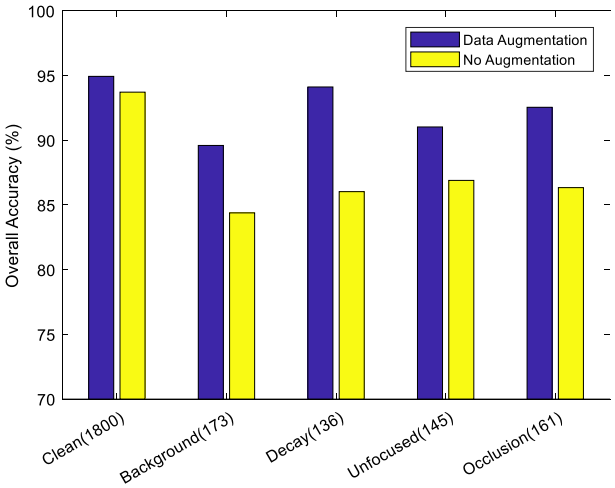


Fig. 12 Data augmentation versus no augmentation

convolution layer and pooling layer, but we did not test the optimal number of fully connected layers. (iii) The computation time can be accelerated if using FPGA.

In the future, we shall try to use realistic images obtained in supermarkets and factories. Besides, we shall test other advanced classification ideas, such as transfer learning, and use FGPA to accelerate the algorithm implementation. Our method may be combined with recommendation system [43], mobile computing [41], and big data [42].

Acknowledgments This study was supported by Natural Science Foundation of China (61602250), Natural Science Foundation of Jiangsu Province (BK20150983), Open fund of Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence (2016CSCI01).

Compliance with ethical standards

Conflict of interest We have no conflicts of interest to disclose with regard to the subject matter of this paper.

References

1. Acquarelli J, van Laarhoven T, Gerretzen J et al (2017) Convolutional neural networks for vibrational spectroscopic data analysis. Anal Chim Acta 954:22–31
2. Adak MF, Yumusak N (2016) Classification of E-Nose Aroma Data of Four Fruit Types by ABC-Based Neural Network. Sensors 16(3):13
3. Ahmad J, Mehmood I, Baik SW (2017) Efficient object-based surveillance image search using spatial pooling of convolutional features. J Vis Commun Image Represent 45:62–76
4. Bai X, Shi BG, Zhang CQ et al (2017) Text/non-text image classification in the wild with convolutional neural networks. Pattern Recogn 66:437–446
5. Chen Y (2016) Voxelwise detection of cerebral microbleed in CADASIL patients by leaky rectified linear unit and early stopping: a class-imbalanced susceptibility-weighted imaging data study. Multime Tools Appl. https://doi.org/10.1007/s11042-017-4383-9

6. Cicero M, Bilbily A, Dowdell T et al (2017) Training and Validating a Deep Convolutional Neural Network for Computer-Aided Detection and Classification of Abnormalities on Frontal Chest Radiographs. Investig Radiol 52(5):281–287
7. Cintas C, Quinto-Sanchez M, Acuna V et al (2017) Automatic ear detection and feature extraction using Geometric Morphometrics and convolutional neural networks. IET Biometrics 6(3):211–223
8. Dai-Ton H, Duc-Dung N, Duc-Hieu L (2016) An adaptive over-split and merge algorithm for page segmentation. Pattern Recogn Lett 80:137–143
9. Deliens T, Deforche B, Annemans L et al (2016) Effectiveness of pricing strategies on french fries and fruit purchases among university students: results from an on-campus restaurant experiment. PLoS One 11(11): 16 Article ID: e0165298
10. Di Cagno R, Filannino P, Cavoski I et al (2017) Bioprocessing technology to exploit organic palm date (*Phoenix dactylifera L. cultivar Siwi*) fruit as a functional dietary supplement. J Funct Foods 31:9–19
11. Garcia F, Cervantes J, Lopez A et al (2016) Fruit classification by extracting color chromaticity, shape and texture features: towards an application for supermarkets. IEEE Lat Am Trans 14(7):3434–3443
12. Getahun S, Ambaw A, Delele M et al (2017) Analysis of airflow and heat transfer inside fruit packed refrigerated shipping container: Part I - model development and validation. J Food Eng 203:58–68
13. Ghazi MM, Yanikoglu B, Aptoula E (2017) Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235:228–235
14. Ji G (2014) Fruit classification using computer vision and feedforward neural network. J Food Eng 143: 167–177
15. Jiang YL, Zur RM, Pesce LL et al (2009) A Study of the Effect of Noise Injection on the Training of Artificial Neural Networks. In International Joint Conference on Neural Networks (IJCNN), IEEE, Atlanta, pp 2784–2788
16. Kim JH, Hong HG, Park KR (2017) Convolutional neural network-based human detection in nighttime images using visible light camera sensors. Sensors (Basel) 17(5). https://doi.org/10.3390/s17051065
17. Kooi T, van Ginneken B, Karssemeijer N et al (2017) Discriminating solitary cysts from soft tissue lesions in mammography using a pretrained deep convolutional neural network. Med Phys 44(3):1017–1027
18. Lee CH, Chien JT (2016) Deep unfolding inference for supervised topic model. In International Conference on Acoustics, Speech And Signal Processing Proceedings, IEEE, Shanghai, pp 2279–2283
19. Li S, Jiang H, Pang W (2017) Joint multiple fully connected convolutional neural network with extreme learning machine for hepatocellular carcinoma nuclei grading. Comput Biol Med 84:156–167
20. Liu F, Snetkov L, Lima D (2017) Summary on fruit identification methods: A literature review. Adv Soc Sci Educ Hum Res 119:1629–1633
21. Lu Z (2016) Fractional Fourier entropy increases the recognition rate of fruit type detection. BMC Plant Biol 16(S2) Article ID: 10
22. Lu Z, 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, Institute of Industrial Applications Engineers, Japan, pp 293–299
23. Miki Y, Muramatsu C, Hayashi T et al (2017) Classification of teeth in cone-beam CT using deep convolutional neural network. Comput Biol Med 80:24–29
24. Oliva D, Hinojosa S, Cuevas E et al (2017) Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm. Expert Syst Appl 79:164–180
25. Pardo-Mates N, Vera A, Barbosa S et al (2017) Characterization, classification and authentication of fruit-based extracts by means of HPLC-UV chromatographic fingerprints, polyphenolic profiles and chemometric methods. Food Chem 221:29–38
26. Qian RQ, Yue Y, Coenen F et al (2016) Traffic sign recognition with convolutional neural network based on max pooling positions. In 2th International Conference on Natural Computation, Fuzzy Systems And Knowledge Discovery (ICNC-FSKD), IEEE, Changsha pp 578–582
27. Radi, Ciptohadijoyo S, Litananda WS et al (2016) Electronic nose based on partition column integrated with gas sensor for fruit identification and classification. Comput Electron Agric 121:429–435
28. Shao WH, Li YJ, Diao SF et al (2017) Rapid classification of Chinese quince (Chaenomeles speciosa Nakai) fruit provenance by near-infrared spectroscopy and multivariate calibration. Anal Bioanal Chem 409(1):115–120
29. Sui XD, Zheng YJ, Wei BZ et al (2017) Choroid segmentation from Optical Coherence Tomography with graph edge weights learned from deep convolutional neural networks. Neurocomputing 237:332–341
30. Tabik S, Peralta D, Herrera-Poyatos A et al (2017) A snapshot of image pre-processing for convolutional neural networks: case study of MNIST. Int J Comput Intellig Syst 10(1):555–568

31. Teh V, Sim KS, Wong EK (2016) Brain early infarct detection using gamma correction extreme-level eliminating with weighting distribution. Scanning 38(6):842–856
32. Thung KH, Paramesran R, Lim CL (2012) Content-based image quality metric using similarity measure of moment vectors. Pattern Recogn 45(6):2193–2204
33. Thung KH, Wee CY, Yap PT et al (2014) Neurodegenerative disease diagnosis using incomplete multi-modality data via matrix shrinkage and completion. NeuroImage 91:386–400
34. Thung KH, Wee CY, Yap PT et al (2016) Identification of progressive mild cognitive impairment patients using incomplete longitudinal MRI scans. Brain Struct Funct 221(8):3979–3995
35. Tovar MF, Losada HV (2016) Fuzzy systems: case study classification of fruit Mc Stipitata Vaug (Araza). Amazonia Investiga 5(9):45–56
36. Wei L (2015) Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization. Entropy 17(8):5711–5728
37. Wu L (2012) Classification of fruits using computer vision and a multiclass support vector machine. Sensors 12(9):12489–12505
38. Wu J (2016) Fruit classification by biogeography-based optimization and feedforward neural network. Expert Syst 33(3):239–253
39. Smirnov EA, Timoshenko DM, Andrianov SN (2014) Comparison of regularization methods for imagenet classification with deep convolutional neural networks. In 2nd Aasri Conference on Computational Intelligence And Bioinformatics (CIB). Elsevier Science Bv, South Korea, pp 89–94
40. Yaghoubi S, Noori S, Azaron A et al (2015) Resource allocation in multi-class dynamic PERT networks with finite capacity. Eur J Oper Res 247(3):879–894
41. Zhang Y (2016) GroRec: a group-centric intelligent recommender system integrating social, mobile and big data technologies. IEEE Trans Serv Comput 9(5):786–795
42. Zhang Y, Qiu M, Tsai CW et al (2015) Health-CPS: healthcare cyber-physical system assisted by cloud and big data. IEEE Syst J PP(99):1–8
43. Zhang Y, Chen M, Huang D et al (2017) iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. Futur Gener Comput Syst 66:30–35
44. Zhu SG, Du JP (2014) Visual tracking using max-average pooling and weight-selection strategy. J Appl Math 2014:828907

Yu-Dong Zhang received his Ph.D. degree from Southeast University at 2010. He worked as a postdoc from 2010 to 2012, and a research scientist from 2012 to 2013 at Columbia University. From 2013 to 2016 he worked a professor in Nanjing Normal University. He is now a professor in Henan Polytechnic University. He served as IEEE senior member and ACM senior member. He is included in “Most Cited Chinese researchers (Computer Science)” from 2015 to 2017. He won the “Emerald Citation of Excellence 2017”. He is elected as the “Bentham Ambassador”.

Zhengchao Dong was a tenured-track Associate Professor in Division of Translational Imaging, Columbia University, USA and New York State Psychiatry Institute, USA. He published over 20 papers on JAMA Psychiatry, Progress in Nuclear Magnetic Resonance Spectroscopy, Neuropsychopharmacology, Neuroimage, Human Brain Mapping, etc.

Xianqing Chen got his Ph.D. degree from Southeast University in 2013. Now he is a senior lecturer in Zhejiang Normal University. His research interest is image processing.

Wen-Juan Jia received B.S. in Yancheng Teachers University from the department of Mathematics & Statistics 2016. Now she is pursuing the M.S. in School of Computer Science & Technology, Nanjing Normal University. Her research interest is expert system.

Sidan Du received the Ph.D. degree in physics from Nanjing University, Nanjing, China, in 1997. She is currently a professor in the school of Electronic Science and Engineering, Nanjing University. Her research interests include in digital imaging processing and computer vision.

Khan Muhammad received his BS degree in computer science from Islamia College, Peshawar, Pakistan with research in information security. Currently, he is pursuing MS leading to PhD degree in digitals contents from College of Software Convergence, Sejong University, Seoul, Republic of Korea. He is working as a researcher at Intelligent Media Laboratory (IM Lab). His research interests include image and video processing, wireless networks, information security, data hiding, image and video steganography, video summarization, diagnostic hysteroscopy, wireless capsule endoscopy, and CCTV video analysis.

Shui-Hua Wang received a B.S. from Southeast University (2005–2008) and a M.S. from The City University of New York (2010–2012). She worked as a Research Assistant in Columbia University (2012–2014). She received her Ph.D. from Nanjing University (2014–2017). At present, she works as an Associate Professor. She published over 100 papers in SCI-indexed journals. She served as the editor of Journal of Alzheimer's disease from 2018, and the managing guest editor of Multimedia Tools and Applications (2017–2018).