Comparing the performance of different convolution kernel sizes of fruit recognition based on CNN

1st Weixi Weng
school of computer science
Wuhan University
Wuhan, China
email:1366338385@qq,com

2nd Yifan Hu
school of computer science
Wuhan University
Wuhan, China
email:2284121004@qq.com

3rd Siwei Li
school of computer science
Wuhan University
Wuhan, China
email:1412245897@qq.com

4th Xiaoting Lu
school of network security
Wuhan University
Wuhan, China
email:2645034287@qq.com

5th Di Yang
school of computer science
Wuhan University
Wuhan, China
email:541294116@qq.com

***Abstract*—Fruit-360 is a high-quality dataset of images containing different types of fruits. In o rder to use this dataset to train our convolution neural network, we conducted data preprocessing and augmentation in advance. This article presents the results of some numerical experiment fo r training an efficient convolution neural network to detect fruits. This paper explored the influence that the convolution kernel siz e of the convolution layer has on the neural network structure. O ur experiment confirmed that replacing a convolution layer with a kernel size of 5*5 with two consecutive convolution layers with a kernel size of 3*3 will improve the accuracy and reduce the number of parameters of the neural network trained on the dataset fruit-360.**

***Keywords*—convolution neural network, convolution kernel, fruit classification, Fruit-360**

I. INTRODUCTION

The first aim of this paper is to find out whether we can use a more efficient convolution layer to replace the current convolution layer with a kernel size of 5*5 or not. Considering the principle and structure of a convolution neural network, we think that we can use two consecutive convolution layers with a kernel size of 3*3 to replace a convolution layer with a kernel size of 5*5, and prove it in the research below.

Another aim is to train a convolution neural network to detect fruits and presents the results of some numerical experiment based on the dataset Fruit-360. It can be used as part of some more complex project to obtain a classifier which can identify a wider range of objects from an image. . While several researches have been taken already, they focus on few species of fruits or vegetables. This could cause an error occurred in classifying some unusual fruits or vegetables. We will train a network that can classify a big variety of species of fruit ,which can be used in more scenarios.

The paper is structured as follows: in the first part we will shortly discuss some outstanding results of using deep learning for fruit recognition, and then introduce the concept of deep learning. In the second part, we will describe the Fruits-360 dataset, including how and why it was created and what it contains. In the third part, we will describe the TensorFlow framework used in this project and the reasons why we chose it. After the framework is introduced, we will detail the structure of the neural network we use. We also describe the use of training and test data and the performance gained. Finally, we will summarize the results of this project.

II. RELATED WORK

A. Relevant papers

In this part, we review some related works using neural networks and deep learning to recognize fruit.

The paper *Automatic fruit recognition and counting from multiple images*[1] presented a method to recognize and count fruits or vegetables from greenhouse with noisy background. The target vegetable is pepper, whose fruits are complex in shape and color, similar to the plant canopy. The purpose of the application is to locate and count the green and red pepper fruits of large, dense pepper plants growing in greenhouses. The training and validation data used in this paper consisted of 28,000 images of more than 1,000 plants and their fruits. The method uses a two-step method to locate and count the peppers: the first step is to locate the fruit in a single image, and the second step is to combine multiple views to improve the detection rate of the fruit. The method of finding pepper fruit in a single image is based on:

(1) finding points of interest;
(2) applying a complex high-dimensional feature descriptor to patches around points of interest;
(3) using the so-called word bag to classify patches.

With regard to the robot used in fruit farming mentioned in the introduction, the paper *Deep fruit detection in orchards*[2] shows a trained network to identify fruit in an orchard. The image contains many images of fruit trees, so it is difficult to identify them. The amount of fruit can even be large, with some images containing up to 1,500 pieces of fruit. Also, because the images were taken outside, there were many variations in brightness, fruit size, clustering, and perspective. This project utilizes the convolutional network based on Faster Region, which is introduced in

detail in the paper *Faster R-CNN: towards real-time object detection with region proposal networks*[3].

B. Deep Learning

Deep learning is a technology of non-linear transformation or representation learning of input with no less than two hidden layers of neural network. Through the construction of deep neural network, various analysis activities are carried out. Deep neural network consists of an input layer, several hidden layers and an output layer. Each layer has a number of neurons with connection weights between them. Each neuron mimics a biological nerve cell, and the connections between nodes mimic the connections between nerve cells. Deep learning practices have four key elements: computing power, algorithms, data, and application scenarios.

Convolutional Neural Networks (CNN) are a kind of Feedforward Neural Networks with deep structure including Convolutional computation. It is one of the representative algorithms of deep learning. Convolutional neural network has the ability of representation learning and can shift invariant classification of input information according to its hierarchical structure, so it is also known as "shift invariant artificial neural network".

(1) Convolution:
The most basic operation in CNN is convolution. To be more precise, the convolution used in basic CNN is a 2-D convolution. That is to say, kernel can only slide displacement on X and Y, and cannot carry out depth displacement. This can be understood according to the figure. For RGB images in the figure, three independent 2-D kernels were adopted, as shown in yellow, so the dimension of this kernel was **X*Y*3**

In different stages of basic CNN, the depth of kernel should be consistent, equal to the number of channels of the input image.
Convolution requires input of two parameters, which is essentially two-dimensional spatial filtering, and the nature of filtering is related to kernel selection. The convolution of CNN is between a 2-D kernel and the input 2-D input map, and each image channel in RGB is completed separately.

We assume that the spatial coordinates of the single channel input image are **(X,Y)**
The size of the convolution kernel is **p*q**
The kernel weight is **w**
The image brightness value is **v**
The convolution process is the sum of all kernel weight and the brightness of corresponding elements in the input image, which can be expressed as

(2)Activation:
After convolution, bias is usually added and nonlinear activation function is introduced. Here bias is defined as **b**, and activation function is **h()**
After the activation function, the result is

.y $$z_{x,y} = h(\sum_{i}^{p*q} w_i v_i + b)$$

Note here that the bias is not related to element position, only to layer. Mainstream activation functions include Linear rectifier unit (RELU), Sigmoid function and tanh function, etc.

(3)Pooling
Pooling is a subsampling operation whose main goal is to reduce the feature space of feature maps, or it can be considered to reduce the resolution of feature maps. Because the feature map has too many parameters, the image details are not conducive to the extraction of high-level features.Currently, the main pooling operations include:
Max pooling: As shown in the figure above, the 2 * 2 Max pooling is to take the maximum of the 4 pixel points and retain them
Average pooling: As shown in the figure above, the 2 * 2 Average pooling is to take the median value of the 4 pixel points and retain it
L2 pooling: namely, the mean square value is retained
The Pooling operation will reduce parameters and reduce the resolution of feature maps, but it is not certain whether such violence reduction is necessary under the condition of sufficient computing power. Currently some large CNN networks use pooling only occasionally.

The main method of deep learning of the experiment in this paper is CNN.

III. DATASET

Fruits-360 dataset contains 81210 images. Among them, there are 60486 pictures in the training set (each picture contains only one kind of fruit or vegetable), and 20618 pictures in the test set (each picture contains only one kind of fruit or vegetable), all of which are 100x100 pixels in size. In addition, there are 103 pictures with multiple fruits or vegetables, and their sizes are variable. The experiment our paper conducts only chooses the training set and the test set, not involving the use of 'test-multiple_fruits'.

According to Horea, those images were obtained by filming the fruits while they are rotated by a motor and then extracting frames. The raw pictures have the motor shaft and background is a piece of white paper(Figure 1). Both of them will have influence on the detecting. But these factors have been solved in the final processed image and the size is scaled down to 100 x 100 pixels(Figure 2).

Figure 1: raw image

Figure 2: processed image

Fruits-360 dataset contains 120 different fruits and vegetables, including the following categories:
Apples (different varieties: Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Beetroot Red, Blueberry, Cactus fruit, Cantaloupe (2 varieties), Carambula, Cauliflower, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Dates, Eggplant, Ginger Root, Granadilla, Grape (Blue, Pink, White (different varieties)), Grapefruit (Pink, White), Guava, Hazelnut, Huckleberry, Kiwi, Kaki, Kohlrabi, Kumsquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango (Green, Red), Mangostan, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine (Regular, Flat), Nut (Forest, Pecan), Onion (Red, White), Orange, Papaya, Passion fruit, Peach (different varieties), Pepino, Pear (different varieties, Abate, Forelle, Kaiser, Monster, Red, Williams), Pepper (Red, Green, Yellow), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum (different varieties), Pomegranate, Pomelo Sweetie, Potato (Red, Sweet, White), Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red, Yellow), Walnut.

To be noticed, one kind of fruit also has multiple varieties, such as apples. So each of them is considered as a separate object. For fruits whose scientific names cannot be found, they are labeled and differentiated with numbers like 'Apple Golden 1' (Figure 3), 'Apple Golden 2' (Picture 4) and 'Apple Golden 3'(Figure 5).

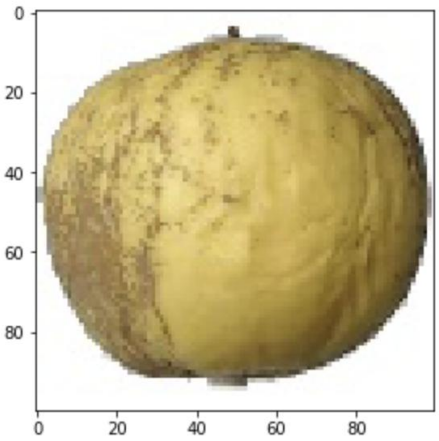

Figure 3 : Apple Golden 1

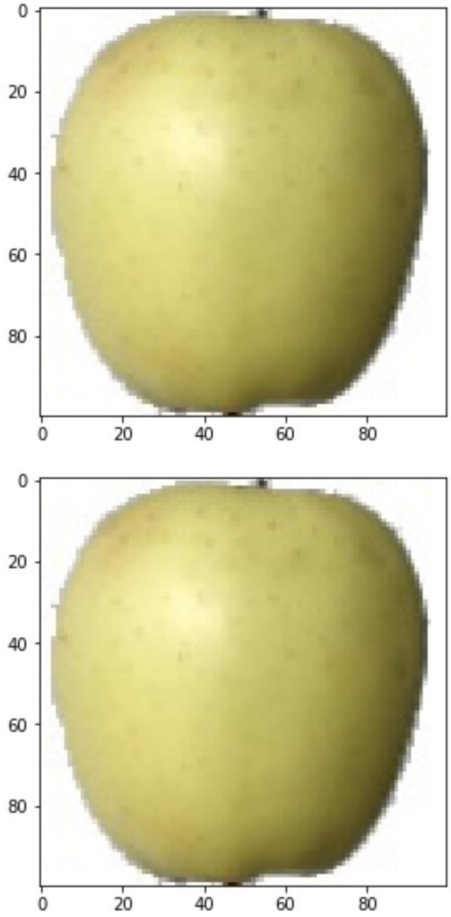

Figure 4: Apple Golden 2

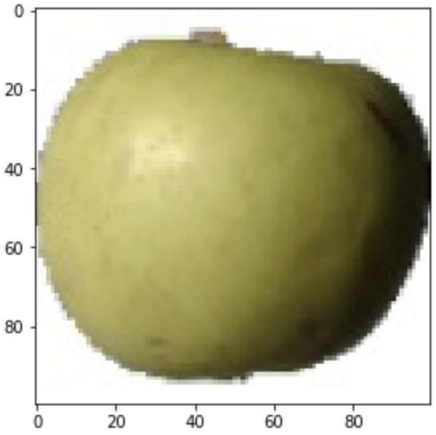

Figure 5 : Apple Golden 3

IV. Numeric Experiment

A. Experiment on different structure of CNN

For the experiments we used the 90380 images split in 2 parts: training set which consists of 67692 images of fruits and testing set which is made of 22688 images. The other 103 images with various kinds of fruits were not used in the training and testing of the network.

By calling the method ImageDataGenerator provided by the Keras library, we can easily generate a batch of image data and feed it into the neural network. The generator can randomly zoom in and out, shear and flip some of the images

in a batch of data to do data augmentation. This helps greatly lower the chance of using the same batch multiple times for training, which in turn improve the quality of the neural network. During training, the method will generate randomized input into the neural network in real time infinitely until the specified epoch times are reached. In Table 1 we present the parameters used in the ImageDataGenerator.

TABLE I. PARAMETERS USED

Parameters	**value**
rescale	1.0/255
shear_range	0.3
horizontal_flip	True
vertical_flip	False
zoom_range	0.3

An efficient convolution neural network is presented in the paper[1]. We analyzed the neural network structure of the paper, and find that the original neural network proposed in the paper[1] are composed of several convolution layer with the kernel size of 5*5. This article tries to explore whether two consecutive convolution layers with the kernel size of 3*3 can be used to replace a single convolution layer with the kernel size of 5*5, so we designed three different neural network structures and tested them together. In order to determine the best network configuration for classifying the images in our dataset, we took multiple configurations, used the train set to train them and then calculated their accuracy on the training set and the testing set. In Table 2 we present the results.

TABLE II. RESULTS OF DIFFERENT NETWORK STRUCTURE

Nr	**Configuration**			**Accuracy on testing set**	**Accuracy on training set**
1	Convolution	5*5	16	98.23%	93.39%
	Convolution	3*3	32		
	Convolution	3*3	32		
	Convolution	3*3	64		
	Convolution	3*3	64		
	Convolution	5*5	128		
	Fully Connected		1024		
	Fully Connected		256		
2	Convolution	5*5	16	97.68%	90.53%
	Convolution	5*5	32		
	Convolution	5*5	64		
	Convolution	5*5	128		
	Fully Connected		1024		
	Fully Connected		256		
3	Convolution	5*5	16	96.16%	89.54%
	Convolution	5*5	32		
	Convolution	3*3	64		
	Convolution	3*3	64		
	Convolution	5*5	128		
	Fully Connected		1024		
	Fully Connected		256		
4	Convolution	5*5	16	96.41%	92.58%
	Convolution	3*3	32		
	Convolution	3*3	32		
	Convolution	5*5	64		
	Convolution	5*5	128		
	Fully Connected		1024		
	Fully Connected		1024		

From Table 2 we can see that the best performance on the test set was obtained by configuration nr.1. The same configuration also obtained the best accuracy on the training set. This indicates that if a configuration of neural network obtained high accuracy on the train set, this will translate into a good performance on the test set. Compared to configuration nr.2 proposed in the paper[1], configuration nr.1 replaced some of the convolution layers with the kernel size of 5*5 with two consecutive convolution layers with the kernel size of 3*3 and had a better performance. However,

other experiments showed that this kind of replacement sometimes does not improve the accuracy on both the training set and the testing set. It reminds of us that we should consider this kind of replacement based on the problem we face and the actual experiment result.

B. The structure of our convolution neural network

CNN is one of the most established deep learning algorithm and is widely used in many fields such as image classification, object recognition and medical image examination. Considering its superior performance in classification, we chose CNN for our project. Usually, the architecture of CNN can be split into two sections: feature learning and classification. In our project, combined with the classical Convolution neural network model structure, we construct the model mainly using three types of layers: the convolution layer, the pooling layer and the fully connected layer, as shown as nr.1 in Table 2.

The first layer (Convolution 1) is a Convolution layer with 16 filters, each filter is in the size of 5*5 and has 3 color channels. These filters extract characteristics from a matrix of values. And carried out convolution operation between the filter and the image to detect the features. On this layer we apply max pooling with a filter of shape 2*2 with stride 2, which specifies that the pooled regions do not overlap (Max-Pool 1) At this level we use the activation function Relu. This layer output 32 filter maps. Other convolution layers are similar to the first layer, so we just discussed about the first one.

The seventh layer is a fully connected layer, the data from the former layer are flattened to the size of 512, and this layer feeds into another fully connected layer with 1024 inputs and 256 outputs.

The last layer is a softmax loss layer (Softmax) with 256 inputs. The number of outputs is equal to the number of classes. This layer is responsible for the classification part. Based on the one-dimensional vector from the input, it use the non-linear activation Softmax to make the final decision.

V. CONCLUSION

We used a new and complex dataset of images with fruits and do data augmentation on the dataset. Also we made some numerical experiments by using TensorFlow library in order to classify the images according to their content.

Through several experiments on the four different structures of convolution neural network we propose, we can use two consecutive convolution layers with a kernel size of 3*3 to replace a convolution layer with a kernel size of 5*5, and present an efficient structure of convolution neural network.

In the near future we plan to create a mobile application which takes pictures of fruits and labels them accordingly. Another objective is to expand the data set to include more fruits. This is a more time consuming process since we want to include items that were not used in most others related papers.

REFERENCES.

[1] Song, Y., Glasbey, C., Horgan, G., Polder, G., Dieleman, J., and van der Heijden, G. Automatic fruit recognition and counting from multiple images. Biosystems Engineering 118 (2014), 203 – 215.

[2] Bargoti, S., and Underwood, J. Deep fruit detection in orchards. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (May 2017), pp. 3626–3633.

[3] Ren, S., He, K., Girshick, R. B., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR54 abs/1506.01497 (2015).