CAPTCHA Recognition based on LeNet-5

Yu Liu
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, PA 15261
YUL189@pitt.edu

Zong-lin Li
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, PA 15261
ZOL3@pitt.edu

***Abstract*—**CAPCHA is used as tests to tell apart computers and humans. The aim of it is to be solved by humans easily and unsolvable by computers. With traditionary pattern recognition methods, we could pass the tests automatically. However, it may encounter some limits, especially for CAPTCHA with distorted formation. Hence, we propose CNN to overcome the weakness of traditional methods by avoiding segmenting the CAPTCHA. We will discuss the recognition performance of two differently formed CAPTCHA and the possible methods to improve the performance.

Keywords—CNN, CAPTCHA Recognition, LeNet-5

I. INTRODUCTION

Many websites have adopted CAPTCHA technology to prevent users from automatically registering, logging in and commenting. However, because the CAPTCHA requires artificial input, it increases the user's workload to a certain extent, and increases the operation time at the same time, which affects the work efficiency. Through the application of the authentication code, it requests users to share the problem of network security, which caused great dissatisfaction among the users, especially for the industry which has high requirement on efficiency and speed of network. It is time to break through the bottleneck. So, to make the users convenient, the automatic CAPTCHA identification product has received the consistent high praise. In this project, what we focus is based on the premise of the legitimate use, so that it can avoid illegal malicious use. If the CAPTCHA identification technology can be used legally, it can bring great convenience to the user.

In real life, the application demand of computer character recognition is quite high, and the field of automatic recognition of computer is becoming more and more extensive. With the development of the recognition technology, computer word recognition technology will be widely used in the field of automatic input of documents, image text processing and automatic reader. In this project, the recognition technology of CAPTCHA is also used for the computer to be able to replace the characters in the automatic recognition code of human eyes, which can be considered as an application of character recognition technology.

II. METHODS

A. Pre-processing

A.1 Resize

To complete the recognition more efficiently, we will adjust the size of the CAPTCHA to the uniform size. The size we set in this project is 30*100.

A.2 Grayscale

Computer vision detection image processing system are often required to convert color images to gray image, the current CAPTCHA images are in the majority with color images, therefore we need to do original image gray processing first. Grayscale is an image that shows only brightness information without color information. Grayscale refers to the process of grayscale images with brightness and color image programming. Grayscale processing is very important in many image processing, and the result of grayscale is the basis of subsequent processing, so it is very important to seek a suitable grayscale algorithm.

A.3 Reshape

In the following steps we will place the grayscale values of each pixel in a row in the grayscale image.

A.4 Summary

The purpose of Pre-processing is to improve the quality of image data, prevent the deformation and enhance the certain feature in the subsequent processing. This chapter introduced how to solve the problems which caused by CAPTCHA image itself and laid the foundation for the subsequent recognition

B. LeNet-5[1]

B.1 Convolution neural network

Convolutional neural network is an effective recognition method which has been developed in recent years. Now, the convolutional neural network has become one of the highlights in the many fields of science, especially in the field of pattern classification, due to the network can reduce the pre-processing about the image, thus has been widely used[2].

Convolution neural network achieves displacement recognition, scaling and distortion invariance by combining three methods: local receptive field, weight sharing and subsampling. Local receptive field refers to the connection between neurons in each network layer only in a small neighborhood of the upper layer, through the local receptive field, each neuron can extract primary visual features, such as direction lines, endpoints, and corners. Weight sharing makes convolutional neural networks have fewer parameters and requires relatively few training data. Subsampling reduces the resolution of features and achieves invariance to displacement, scaling and other forms of distortion.

Generally, there is a subsampling layer after the convolution layer to reduce computation time and to establish spatial and structural invariance.

B.1.1 Convolution layer[3]

In the convolution layer, the characteristic graph of the first layer is convoluted with a learning kernel, and the convolution results form the characteristic graph of the layer after the output of the activation function. Each output characteristic graph may be related to the convolution of several characteristic graphs in the previous layer. Generally, the form of the convolution layer is shown in:

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * K_{ij}^l + b_j^l)$$

Among them, l represents the number of layers, k is the convolution kernel, and M_j represents one of the input feature maps. Each output graph has a bias b .

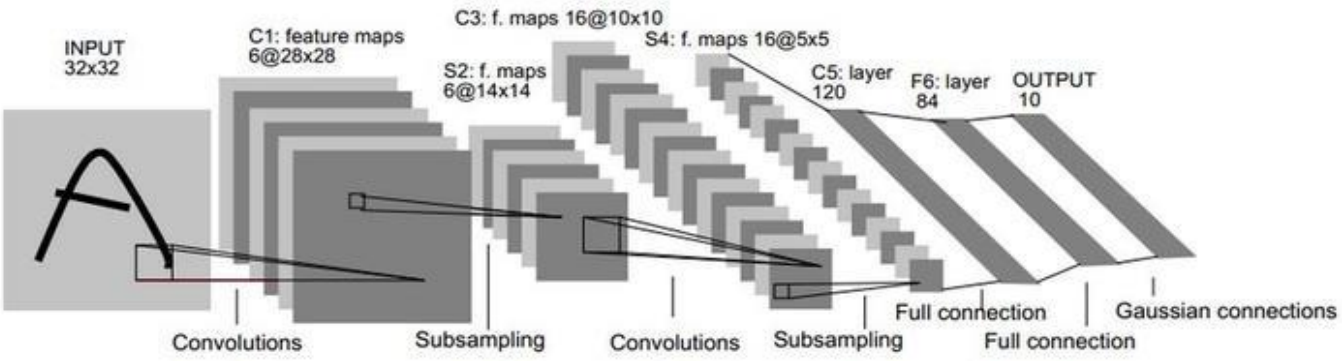

Fig. 1. LeNet-5

TABLE I. THE WAY TO CONNECT S2 AND C3

	0	1	2	3	4	5	6	7	8	9			10	11	12	13	14	15
0	×				×	×	×				×	×	×	×			×	×
1	×	×				×	×	×				×	×	×		×		×
2	×	×	×				×	×	×					×		×	×	×
3		×	×	×			×	×	×	×					×		×	×
4			×	×	×			×	×	×	×				×	×		×
5				×	×	×			×	×	×	×				×	×	×

B.1.2 Subsampling

A subsampling layer is used for sampling the input. If the input feature graph is n, then the number of the feature graph is still n after the subsampling layer, but the output feature graph is smaller (for example, half of the original). The general form of the subsampling layer is shown in:

$$x_j^l = f(\beta_j^l down(x_j^{l-1}) + b_j^l)$$

Among them, $down(\cdot)$ is a subsampling function. The subsampling function generally summits the region of a $n*n$ size of the input image of the layer. The size of the output image is $\frac{1}{n}$ of the size of the input image. Each output feature graph has its own β and b .

B.2.1 LeNet-5

The structure of the convolution neural network LeNet-5 is shown in Figure. The input image is normalized by size, and the input of each neuron comes from a local neighborhood of the previous layer and is added to a weight determined by a set of weights. The extracted features form a higher-level feature at the next level. The neurons of the same characteristic graph share the same set of weights, and the subsampling level averages the upper layer.

Convolution neural network LeNet-5 does not include input, consisting of 7 layers, each layer includes trainable parameters (weights). The input of the network is 32 * 32 image, in which the C layer is a network layer composed of a layer of layer neurons, and the S layer is a network layer composed of sub sampling neurons.

The network layer C1 is a convolution layer consisting of 6 characteristic graphs. Each neuron is connected to a 5 * 5 neighborhood of the input image, so the size of each feature is 28 * 28.

The network layer S2 is composed of 6 feature maps with a size of 14 * 14. It is sampled by the C1 layer. Each neuron in the feature graph is connected to the C1 layer by a neighborhood of 2 * 2.

The network layer C3 consists of 16 volumes of 10 * 10 characters. Each neuron of the feature graph connects with some characteristic graphs' 5 * 5 neighborhood of S2 network layer. Table 1 shows how the feature graph of the S2 layer to form each feature map of C3, each of column represents the characteristic graph of the S2 layer which forms a feature map of C3. For example, from the first column, we know that if we combine the 0th, 1st, 2nd character graph of S2, we will get the 0th character graph of C3.

The network layer S4 is a subsampling layer composed of 16 characteristic graphs with a size of 5 * 5. Each neuron in the feature graph is connected to a 2 * 2 size neighborhood in the C3 layer.

The network layer C5 is a convolutional layer composed of 120 feature graphs. Each neuron is connected to a 5*5 size neighborhood of all the features in S4 network layer.

The network layer F6, including 84 neurons, is fully connected with the network layer C5.

Finally, the output layer has 10 neurons, which are composed of radial basis function units (RBF). Each neuron in the output layer corresponds to a character class. The calculation method of the output y_i of the RBF unit is shown in:

$$y_i = \sum_j (x_j - w_{ij})^2$$

B.2.3 Improvement of LeNet-5

The convolution neural network (LeNet-5) is originally used in handwritten digital recognition. The number of categories of the output is 10. Compared with the handwritten digital recognition, CAPTCHA recognition is much more character to be classified. In addition to 10 Arabia figures, there are 26 upper English letters and 26 lower English letters[4].

The improvement of the LeNet-5 is:

The output layer of the traditional LeNet-5 is changed from 10 neurons to 62 neurons. So, the number of output neurons is changed to 76.

B.2.3 Experimental platform

Hardware:

CPU: Inter I7-7700HQ

GPU: NVidia GTX 1070 8G

RAM: 16G DDR4 2400MHz

Software:

OS: Window 10

Complier: Python 3.6.5

Support Package: TensorFlow-GPU 1.7.0

III. Experimental Results and Discussion

We present the results and discussion of our testing based on data recorded in Tensor board.

A. *Basic Information of Dataset*

Before presenting test results, we should introduce some basic information of our datasets. The standard captcha dataset was captured by spider from Discuz Service. And the distorted captcha dataset is generated by python captcha library. For each dataset, it is divided into three parts. Training set contains fifty thousand pictures of captcha and validation set owns ten thousand pictures. The rest set is test set, which has the size of one thousand pictures.

B. *Performance of Standard Captcha Recognition without dropout[5]*

With consideration that there is no clear evidence of overfitting, we choose to make keep-possibility of dropout equal to one which means there will be no features being dropped during the process. To pursue as higher accuracy as possible, we set the one hundred and ten thousand steps as the break point. After 110000 steps, the model of standard captcha recognition is saved and could be evaluated by the testing set. Based on the curves extracted from Tensor board, it will be found no obvious evidence showing the training is overfitting. The training accuracy could be up to 96.25% and the validation accuracy is 91.60%, which have no obvious difference between each other. The loss is continuously decreasing until end 0.0102. And the following test resulted to the accuracy ratio of 86.17%, which follows our expectation. Notice that the initial parts of two curves are flat. It is due to each our batch only contains 150 figures and the whole dataset for training and validation is up to 60000 figures.

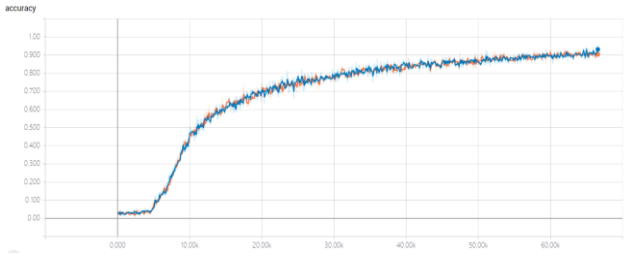

Fig. 2. Accuracy of standard CAPTCHA Recognition without drop-out

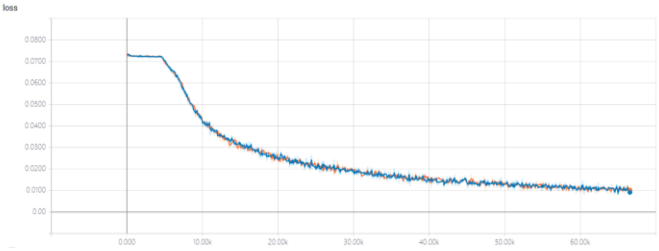

Fig. 3. Loss of standard CAPTCHA Recognition without drop-out

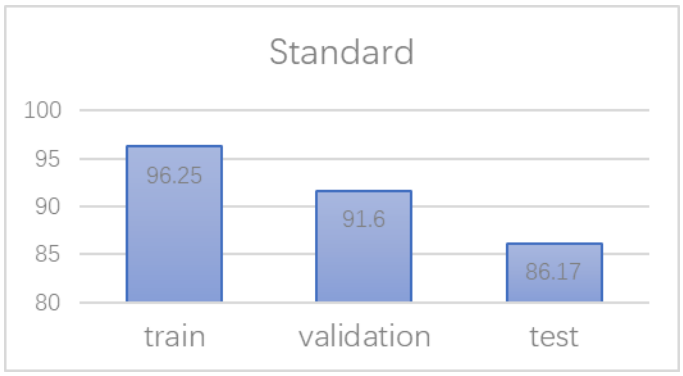

Fig. 4. Result of standard CAPTCHA recognition without drop out

C. *Discussion of Performance of Distorted Captcha Recognition without dropout[5]*

Since the wonderful performance of standard CAPTCHA recognition without dropout. The keep possibility parameter of distorted CAPTCHA is also set up as one. Considering the complexity of distorted captcha, to determine the number of steps training needs, the break point is set up as accuracy ratio of 95%. And in practical training process, the number is 27164. Thus, in the following test, the break point is set up as 25500th step. From the two curves extracted from Tensor board, the loss curve and accuracy curves are all satisfying, without any clues of overfitting. Training accuracy and validation accuracy are both monotonically increasing until the points of 93.13% and 91.20%. At the same time, loss is going to 0.0901. However, the test accuracy of distorted captcha without dropout is far from our expectation, which is only 65.63%.

To find the reason of such difference between training accuracy and test accuracy, the possibility of overfitting comes to the first consideration. Although the data recorded in Tensor board does not show any evidence of overfitting, this possibility still could not be excluded. So that the keep possibility of dropout is changed from 1into 0.4, which means only forty percent of the original will be kept after dropping process. Another possibility is that the validation method has something wrong as well, resulting to the consequence that the overfitting does not reflect on the validation accuracy curve. At the beginning, the fixed validation set was used. To make the result more accurate, it is replaced by cross-validation method.

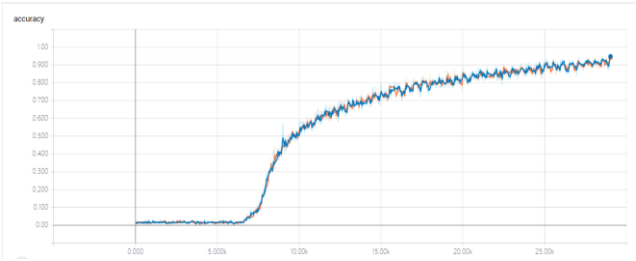

Fig. 5. Accuracy of distorted CAPTCHA Recognition without drop-out

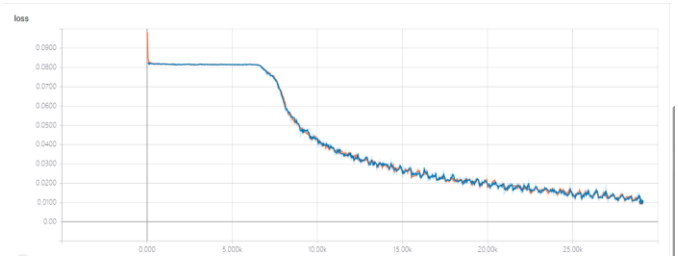

Fig. 6. Loss of distorted CAPTCHA Recognition without drop-out

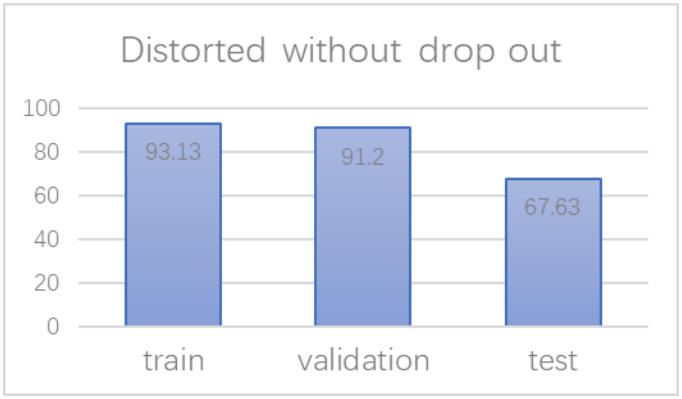

Fig. 7. Result of distorted CAPTCHA Recognition without drop-out

D. *Discussion of Performance of Distorted Captcha Recognition with dropout[5]*

After the modification of relative parameters, the training costs would cost much more steps to finish training. In this trial, the break point is 26500th step. When training finished, the training accuracy is 90.70% and the validation accuracy is up to 92.37%. Those two curves are satisfying as the last trial. However, as shown in the above, there is some improvement but not obviously. The update result is 69.93%, which is not as good as standard captcha.

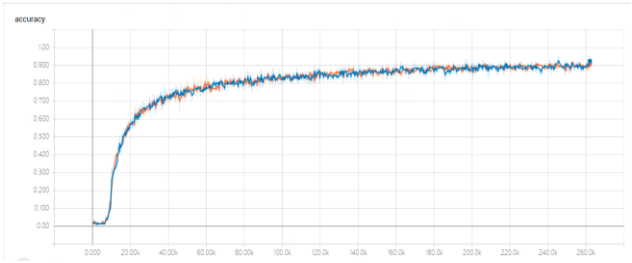

Fig. 8. Accuracy of distorted CAPTCHA Recognition with drop-out

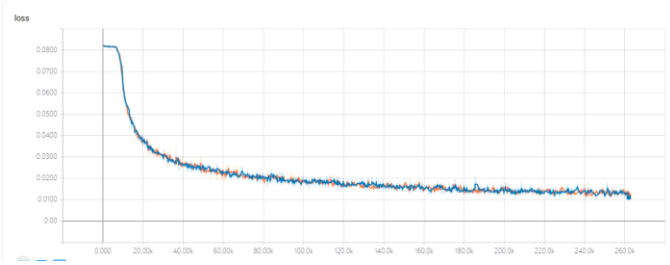

Fig. 9. Loss of distorted CAPTCHA Recognition with drop-out

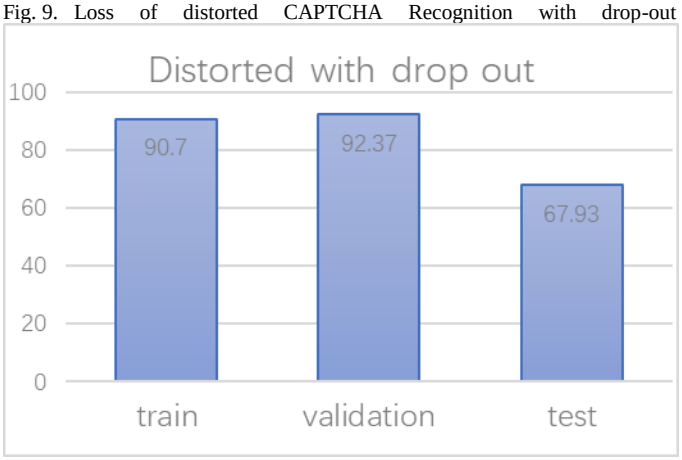

Fig. 10. Result of distorted CAPTCHA Recognition with drop-out

E. Performance of switch model

Another idea which is examined in this project is that whether the trained model of complex captcha could be used to predict the captcha with simple formation. Thus, there will be no need to train corresponding model for each kind of captcha. The figures show results of switching model, which is too bad and the complex model has no use to recognize simply formed captcha. It is verified that features machine learned is far different than people's thinking. And it explains why the captcha could be designed easy for human and hard for computers.

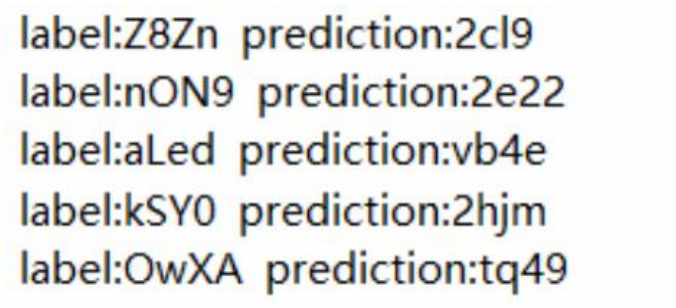

Fig. 11. Result of test between standard Captcha and distorted Captcha model

IV. Conclusion

We propose a CAPTCHA solving technique based on a simple CNN network—LeNet5 and discuss the performance for different kinds of captcha. In the next, we will focus on improving the distorted captcha recognition accuracy and training efficiency by using Midian Filter to deal with the original distorted captcha and modifying the construction of network. At the meanwhile, we would find a way to train a general model through mixed dataset.

V. Contribution

Yu Liu: Final Report, Codes of LeNet-5 Construction
Zong-lin Li: Final Report, Presentation PPT, Test Scripts.

References

[1] Y. Lecun et al, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, (11), pp. 2278-2324, 1998.

[2] E. Bursztein, M. Martin and J. Mitchell, "Text-based CAPTCHA strengths and weaknesses," in 2011, . DOI: 10.1145/2046707.2046724.

[3] C. Chen, Y. Wei and W. Fang, "A study on captcha recognition," in 2014, . DOI: 10.1109/IIH-MSP.2014.105

[4] Y. Lv et al, "Chinese character CAPTCHA recognition based on convolution neural network," in 2016, . DOI: 10.1109/CEC.2016.7744412.

[5] Stark F, Hazırbas C, Triebel R, et al. Captcha recognition with active deep learning[C]//Workshop New Challenges in Neural Computation 2015. Citeseer, 2015: 94.