CAPTCHA recongnition based on deep convolutional neural network

Github: https://github.com/how1215/CaptchaRecognition.git
YT:https://youtu.be/qauamsH-fGQ
周子豪
P76131717
國立成功大學

Abstract

Convolutional Neural Networks(CNN) was remarkable performance in some specific field like image recognition or image classification.Hence, Hence, research in recent years has also focused on improving results by using deeper neural networks.With the development of these models, image recognition can be implemented not only with higher accuracy but also across a wider range of applications.Therefore,this research will implement CAPTCHA recognition based on deep convolutional neural network including CNN or RESNET,and find a better one among all.Also will try different type of CAPTCHA as input to test the generalization of those models.Finally,optimize the model in order to improve accuracy as much as possible.

1 Introduction

Image recognition is an iconic problem in computer vision and is also widely used in real life such as facial recognition, medical diagnosis and so on.There is one interesting topic when it comes to image recognition:CAPTCHA recognition.

A CAPTCHA[1](Completely Automated Public Turing test to tell Computers and Humans Apart) is a type of challenge–response test used in computing to determine whether the user is human in order to deter bot attacks and spam.It is usually presented in the form of voice text or text image.Figure 1 shows a few samples of the common alpha-numerical CAPTCHAs and their types.

1.1 Motivation

The original intention of captch was to prevent any kind of non human access.However, with the development of AI technology and the reduction of the computing hardware cost, some type like text image of CAPTCHA can be easily cracked by well-trained image recognition model.In fact,recently studied[2] has shown that human beings perform significantly worse than robots in text image or other kind of CAPTCHA.This has led to the development of more different types of CAPTCHA which is more difficult for computers to recognize.Therefore,this research will focus on implementing a well perfrom image recognition to solve CAPTCHA with high accuracy.

Figure 1: *Example of different alphanumerical CAPTCHAs*

1.2 Problem

Common CAPTCHA on web pages are text images.Those text could be distorted or covered by lines and noise point to hinder non-human vision.Hence,one of the problem is how to effectively process these data to improve accuracy since the image recognition model may train better with clean text image .However,this can raise another question:Can we train a model that can recognize the CAPTCHA without doing preprocessing and can still maintain a high accuracy?

After data analysis and pre-processing ,next problem is to discuss which model is the best match for these dataset to produce the best possible results.Text image CAPTCHA nowadays have different variants,which means making a general recognition which had good performance on all kind of CAPTCHA is still a challenge.So in this research will focus on one type(Example is in Firgure **??**) of text images only.

1.3 Solution

Very Deep Convolutional Networks(VGG)[3] has made great progress in the field of image recognition.The innovation of VGG lies in its use of a relatively simple stacked convolution layer structure, where multiple small convolution filters are used to replace larger ones. This structure allows the network to be deeper, with more shared parameters, which helps in extracting richer features.There is a research uses it to implement CAPTCHA recognition[4] ,and achieved test accuracies of 95% on VGG-19.

Resent[7] is a major breakthrough in Deep Convolutional Networks.In the paper[7],they design a model with a reformulating layer using a residual learning block.This method enable them to reduce the difficulty of training deeper network, and also improved accuracy.Therefore,I think it will perform very well in CAPTCHA recognition by using the pre-trained resnet model.

There is also another research using dense convolutional neural network to implement the recognition.They improve and construct a new model based on DenseNet[5] called DFCR[6] on CAPTCHA recognition which gained accuracy of CAPTCHA with the background noise and character adhesion above 99.9 %.In this research will use ResNet[7] as based model to implement CAPTCHA recognition.

2 Method

2.1 Dataset

First, I used the built-in library 'ImageCaptcha' in Python to generate image of Captcha.The Captcha consists of four characters,and each character is either a number from 0 to 9 or a lowercase letter arranging with arbitrary direction and size .Images also have a crossing line and multiple dots as interference with computer vision .

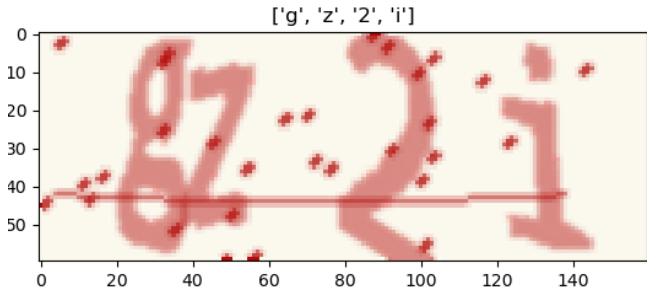

Figure 2: *Four character Captcha generated by the built-in CAPTCHA library in Python.*

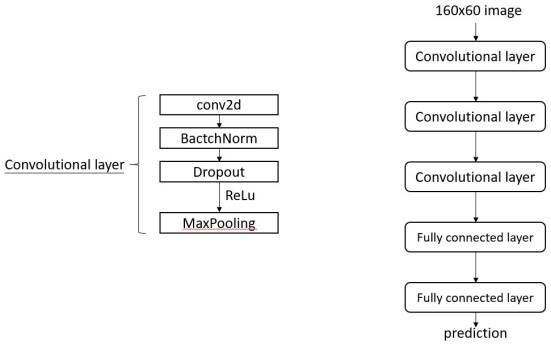

Figure 3: *The structure of CNN*

2.2 CNN

I tried using a simple Convolutional Neural Network constructing with three convolutional layer and two fully connected layer.(Figure 3)I converted the image input dataset from 3 color channel to 1 since the color of the image does little impact on image recognition,and also reduce the computational complexity.For each layer,I used a 2D convolutional layer with kernel size of 3 and padding of 1.Next is the batch normalization.I adopt dropout out to prevent overfitting right after the batch layer.Following up with activation and 2D pooling layer with a 2x2 kernel and a stride of 2.In the end, I modified the output of the last fully connected layer with 4x36 which is the numbers of characters times the range of them(0 to 9,a to z).

2.3 Fine-tuning Resnet-18

Next step, I implement a transfer learning model based on pre-trained model- resent-18.The reason I choose resent-18 to be my pre-trained model is because with fewer parameters compared to deeper models, resnet-18 requires less training time and converges faster.

Since the original model of resent-18 training weights I used was trained on the ImageNet dataset,I modified the last full connected layer output from the default 1000 class to 144 to meet the 4 characters output and each of the character is either lowercase letter or 0 to 9 number for each characters.Also, to satisfied the input of the pre-trained resnet-18,I pre-processed the input data by resizing the Captcha image to 224*224 and normalizing with mean equals to [0.485, 0.456, 0.406],standard deviation equals to [0.229, 0.224, 0.225].I performs affine transformations on data to prevent overfitting and improves the model's generalization capability.Below is a example of pre-processed Captcha

image.(Figure 4)

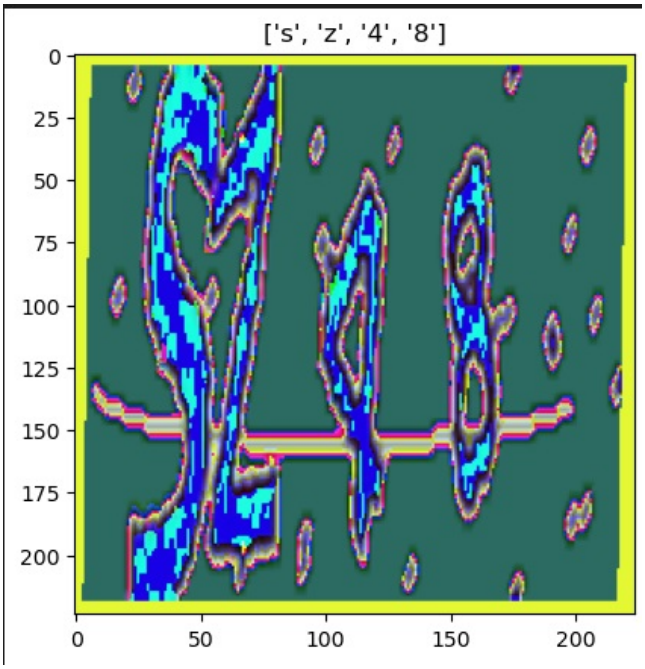

Figure 4: *Example of pre-processed Captcha image for resnet-18*

3 Experiment

The coding environment is python 3.10 and the training is running on NVIDIA GeForce GTX 1660Ti.

3.1 CNN

The built-in library 'ImageCaptcha' in Python randomly generates the training set with 10000 images and valid set for 2000 images.Each image is a 160 x 60 image which is converted to grayscale image.The training set is load with batch size of 128 while valid set is 256.The loss function I select multiLabelSoftMarginLoss to against mulit-label classification.The optimizer I use is SGD with momentum of 0.9 and weight decay of 0.0005.Moreover,I used plateau decay on learning rate to make sure the accuracy can converged in the end without any shocks.So the learning rate starts from 0.05 and turns into 0.001 after 300 epochs.After training 100 epochs,it drops to 0.0001 with and lasts for another 100 epochs.

After training on CNN,the accuracy of the model can reach 84.3% on the validation set(Figure 5),and the loss touch approximately 0.030.The final prediction I select some images in the testing set and compares to the ground truth.The result is on(Figure 6).However,it

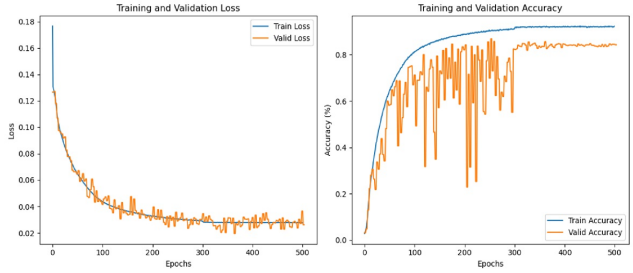

Figure 5: *Accuracy and loss graph for training and valid set*

will still predict a few errors with one of the letters.

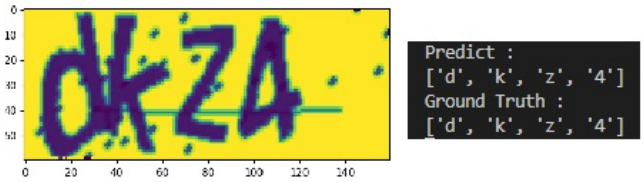

Figure 6: *Result of cnn*

3.2 Fine-tuning Resnet-18

I use the same amount of data splitting as CNN(10000 training set and 2000 validation set).The difference is this time I use batch size of 32 for training and validation set considering that the hardware limitations because the pre-trained model is large and will take plenty of time to train. I also use MultiLabelSoftMarginLoss as my loss function to address multi-label problem.But in fine-tuning resnet-18, I select Adam with learning rate by 0.001 to be my optimizer and it turns out to have better performance than SGD in results.With epoch,I use fixed 10 epoch .

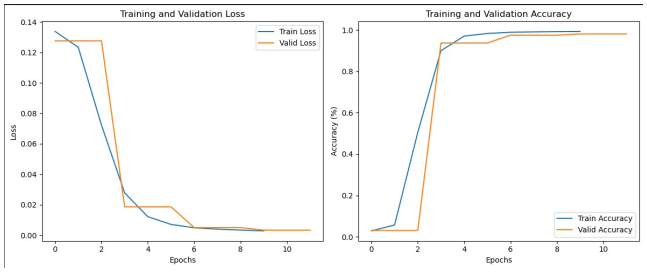

Figure 7: *Accuracy and loss graph for training and valid set*

After training,I randomly generate a Captcha image as a testing set and the prediction result is at below(Figure 8).

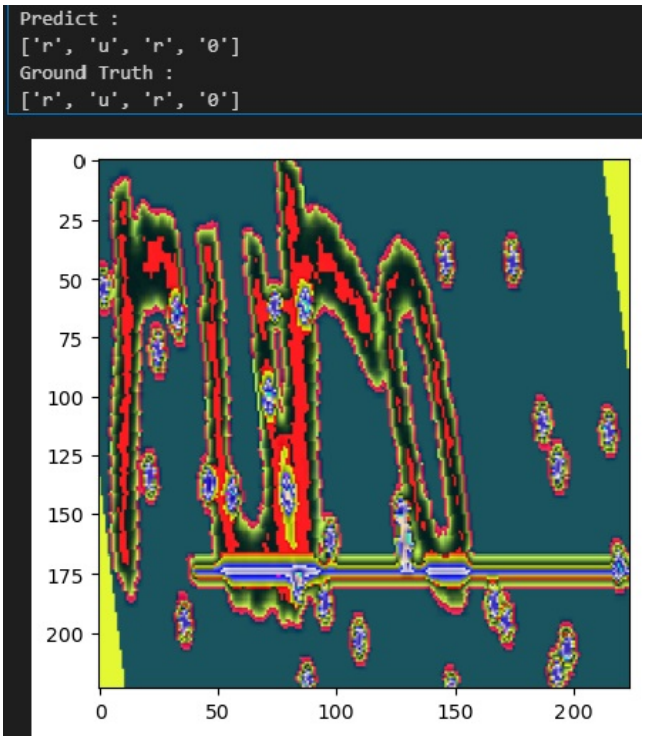

Figure 8: *Result of fine-tuning resnet*

3.3 Evaluate

See Table 1. CNN model has 84.3% accuracy and 0.030 loss on validation set.However,Fine-tuning Resnet-18 can reach 98.1 % accuracy and 0.003 loss on validation set.As the results,resnet-18 has a better performance than CNN.Thanks to the residual connections, it has the ability to train deeper neural network ,which means the more subtle features can be observed.Resnet-18 can effectively handle common distortions found in captchas, such as noise, blurring, and non-linear distortions.Thus,it is a perfect approach while implementing Captcha recognition.

Valid set	Accuracy	Loss
CNN	84.3%	0.030
Fine-tuning Resnet-18	98.1%	0.003

Table 1: *Comparison between two models*

References

[1] Ahn, L.V., Blum, M., Hopper, N., and Langford, J. (2003). CAPTCHA: Using Hard AI Problems for Security. International Conference on the Theory and Application of Cryptographic Techniques.

[2] Searles, A., Nakatsuka, Y., Ozturk, E., Paverd, A., Tsudik, G., and Enkoji, A. (2023). An empirical study and evaluation of modern CAPTCHAs. In 32nd USENIX Security Symposium (USENIX Security 23) (pp. 3081-3097). USENIX Association.

[3] Simonyan, K., and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

[4] Zhao, N., Liu, Y., and Jiang, Y. (2017). CAPTCHA breaking with deep learning. CS 229 Final Project, Stanford University, Tech. Rep.

[5] Huang, G., Liu, Z., and Weinberger, K.Q. (2016). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2261-2269.

[6] Wang, J., Qin, J., Xiang, X., Tan, Y., and Pan, N. (2019). CAPTCHA recognition based on deep convolutional neural network. Math. Biosci. Eng, 16(5), 5851-5861.

[7] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[8] Trong, N.D. and Huong, T.H.and Hoang, V.T. (2023)New Cognitive Deep-Learning CAPTCHA Sensors 2023, 23, 2338.