# Captcha recognition based on CNN

JUN ZHAO, YING LU

JXZ1395, YXL3073

# 1 ABSTRACT

Captcha, a security wall that prevents computer programs from attempting brute force password cracking, mass account registration and other malicious attacks on websites. The research of CAPTCHA recognition needs to be applied to several fields of content, therefore, the research of CAPTCHA can be beneficial to a certain extent to promote the research of image processing, pattern recognition and other fields. The research on CAPTCHA recognition technology can also verify the security of the CAPTCHA used by popular online information platforms to protect information, serve as a means of evaluating the reliability of websites, and contribute to the improvement of the performance of CAPTCHA in terms of its complexity and practicality.

The traditional way to recognize CAPTCHA has four steps: preprocessing, localization, segmentation, and recognition, and each step is independent and unrelated to each other. The problem brough by this method is that once an error occurs in the step before a certain step, it is not only difficult to find the error, but also makes the result recognition rate lower. The deep neural network in the neural network has many network layers, and the use of CNN can be realized by setting different parameters and tasks for each layer to learn different features of the image respectively, and then finally train the network, which can finally simplify the recognition process of CAPTCHA and achieve a higher recognition accuracy rate.

# 2 BACKGROUND

## 2.1 INTRODUCTION TO CAPTCHA

CAPTCHA is an abbreviation for Completely Automated Public Turing test to tell Computers and Humans Apart. It is program or system designed to distinguish between human and machine input, often as a way to block spam and automatically extract data from websites.

In our project, we use the captcha library to generate the CAPTCHA to train our model.

## 2.2 INTRODUCTION TO TENSORFLOW

Currently, machine learning is becoming more and more complex, and deep learning models continue to grow. Many model graphs require distributed training to iterate in a reasonable timeframe, and developers often want to deploy the developed models to multiple platforms. The advantages of using the tensorflow library in recognizing CAPTCHA are portability, optimizability and distributed execution

## 2.3 ENVIRONMENT

Table 1 Environment Information

| Language & library | Versions |
|---|---|
| Python | 2.7 |
| Numpy | 1.16.6 |
| Tensorflow | 1.13.1 |
| Captcha | 0.4 |
| tqdm | 4.64.0 |

# 3 MODEL

## 3.1 MODEL CONSTRUCTION

### 3.1.1 Environment setup

This article is based on a TensorFlow study, starting with the installation of the Anaconda environment.

Because Anaconda is currently a relatively complete data science computing environment, it contains a large number of tools and data packages that do not require tedious testing and tuning. Then, download TensrFlow in the Anaconda environment and run python to view the runtime environment.

### 3.1.2 Captcha generation

The captcha library is used to generate the captcha images for subsequent use as training set, validation set and test set, where *gen_captcha()* is the method to generate the captcha, which is set to check if the captcha image already exists before writing to avoid duplicate generation.

Then read the generated image data and tags, and set these tags as image file names.

## 3.2 CNN MODEL

A convolutional neural network can be seen as a combination of many convolutional layers, which is intuitively understood as a superposition and combination of neural networks.

In this project, the system uses three layers of convolution, and the filter_size is 5, and the general structure is shown in Figure 1, and the specific configuration items are shown in the Table 2.

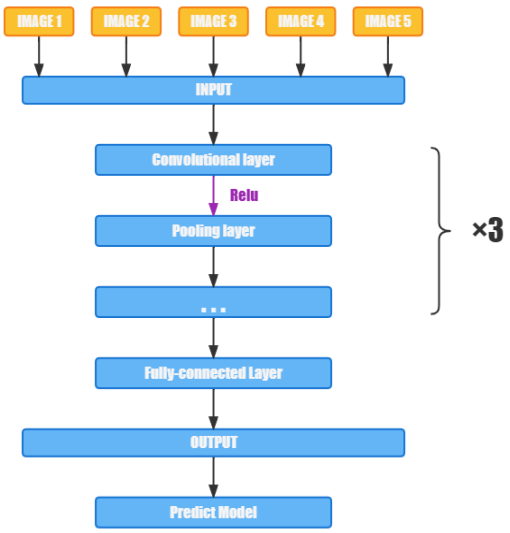



Figure 1 CNN model structure

Table 2 configuration

| Config Option | |
|---|---|
| **Name** | **Parameters** |
| Captcha_width | 160 |
| Captcha_height | 60 |
| Number of characters | 4 |
| Range of characters | From 0 to 9 |
| Learning rate | 1e-3 |
| Epoch | 100 |
| Batch_size | 64 |
| Dropout_ratio | 0.5 |

### 3.2.1 Input layer

The information is passed into the neural network through the input layer. In the case of image files, the image files need to be decoded before being passed into the input layer, and the image information is processed and converted into a matrix before being passed into the input layer.

### 3.2.2 Convolutional layer

Using the convolution kernel with artificially set size to move a set length each time to extract useful features, it can be set smaller so that neighboring neurons in the output feature map can share most of the input, thus ensuring that some regions in the input feature map will not be ignored, and the structure will become deeper after the convolution layer processing, and the performance will be substantially improved, which is more conducive to accurate recognition.

### 3.2.3 Pooling layer

The pooling operation can abstract the concrete information into a fuzzy one, which can greatly reduce the computation of parameters and, to some extent, prevent overfitting. Although the pooling operation cannot change the depth vertically, it can change its size horizontally to reduce the lateral size, which is also a very important and essential part of the convolutional neural network structure.

### 3.2.4 Fully-connected layer

An iterative convolution and pooling operation are used first, and then a small number of fully-connected layers are usually added later to show the classification results. And the above operations can be seen as convolution and pooling to condense the extracted information, abstract the concrete information on the image into data information that can be processed with a computer, and finally still need to use fully connected layers for classification.

The dropout operation is added after the fully connected layer to prevent overfitting and to improve the model generalization ability.

```
# Fully connected layer
W_fc1 = self.weight_variable([20 * 8 * 64, 128])
b_fc1 = self.bias_variable([128])
h_mp3_flat = tf.reshape(h_mp3, [-1, 20 * 8 * 64])
h_fc1 = tf.nn.relu(tf.matmul(h_mp3_flat, W_fc1) + b_fc1)
h_fc1_drop = tf.nn.dropout(h_fc1, self.keep_prob)
```

Figure 2 Fully-connected layer

During the training process, we added a special mechanism: if the model does not improve in 1000 steps, then the model will stop training (because in a sense, there is no possibility of subsequent improvement).

# 4 RESULTS

## 4.1 TRAINING PROCESS

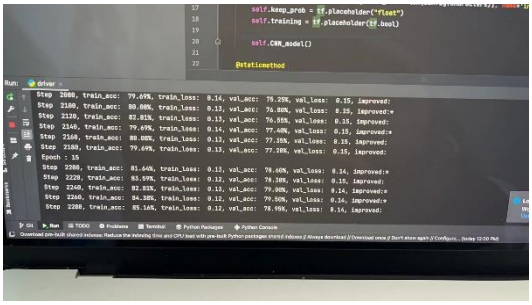


Figure 3 Training process

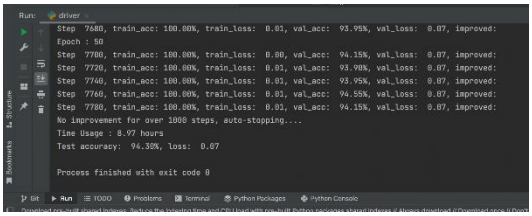
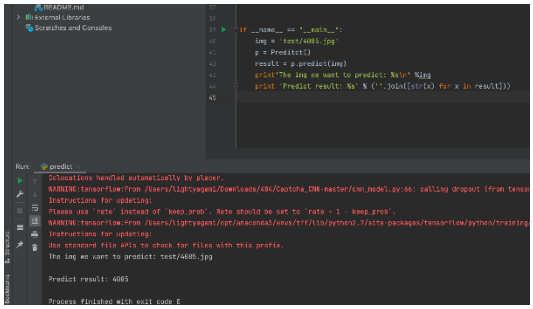


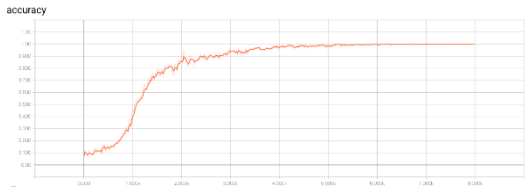
Figure 4 Training result

## 4.2 TRAINING RESULTS

- Validation:
  - Input: 
  - Our output:




- Tensorboard
  - Accuracy




  - Loss


# 5 SUMMARY

The CAPTCHA studied in this paper contains ten characters from 0 to 9, so the number of training set is 10,000 images, and the CAPTCHA recognition work is successful and accurate to some extent.

In this case, the limitations are obvious.

- No CAPTCHA images containing upper and lower case letters A to Z were generated;
- the model is able to recognize only the images generated by the system (i.e. of a specific size and pixels);
- the type of CAPTCHA recognized in the image is too homogeneous.

Future work could be directed towards further research, such as encoding a total of 36 characters in sequence, including upper and lower case letters A to Z and numbers 0 to 9, or another type of algorithmic CAPTCHA commonly used today, such as 2+8=10 (in which case special symbols are obviously included).

# 6 ACKNOWLEDGE

We would like to thank Professor Wyatt, who led us to the door of computional intelligence, and put forward a lot of valuable comments on our research, so that our researchwork has a goal and direction. At the same time, Newman profound knowledge, rigorous scholarship attitude also makes us very admire, is our modelof learning and work. Through this project, we have a deeper understanding of cnn network and tensor flow, I believe that they can also be applied to the code in the future. Thank the teacher for our care and attention, in this to express oursincere gratitude.

# 7 PROJECT LINK

https://github.com/yagami-light95/484-final