# Captcha Recognition
# Machine Learning EL-GY 6132

Mingdi Han mh4928
Mo Wu mw3774

May 17, 2019

## 1 Introduction

**Background** Verification code (CAPTCHA) is an effective tool to distinguish whether a user is a computer or a human. This process can be considered as an image identification problem identify the number or letter from image. Some common methods use image processing and support vector machine (SVM) for verification codecaptcha) recognition,however, the effect of this method depends heavily on the result of image cutting. These methods do not give good results when there are sticking between captcha characters.

As shown in the two figures below. The captcha in figure.1 is a sample one, this type of codes can be recognized by performing character segmentation and identification (SVM). Due to the distortion of and sticking shown in figure.2, a rectangular window cannot be used to split characters.

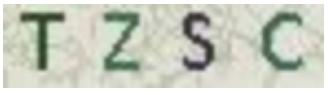


Figure 1: Sample Captcha

Figure 2: Captcha with characters sticking

**Description** In this project, we are going to implement the CAPTCHA recognition by building and training a machine learning model without image

cutting. The model we trained will take the whole image as input, and predict the content of the captcha image.

## 2 Model

In this project, we identify the captcha by convolutional neural network model with tenserflow.

**Dataset** Without training individual characters, the model uses the whole image as training data. Therefore, we need sufficient amount of captcha images to complete the model training. Through the import of the *captcha* library, we were able to generate the captcha images and labels through a few lines of python code. It will generate a branch of captcha images of a combination of 4-bit random uppercase/lowercas letters and numbers, with some distortion and noise processing.

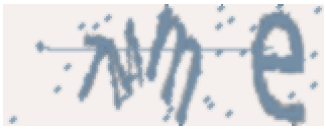

Figure 3: Captcha generated by python code

**Model Structure** We implement a three-layer CNN model in this project. The kernel size of convolution is $3 * 3$, the sampling include Maxpooling and Droupout ,and the activation function is ReLU.

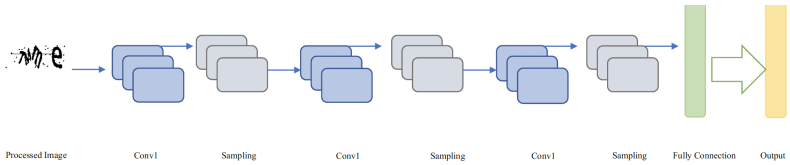


Figure 4: 3 layer CNN model

## 3 Result

We set the branch size as 64, because of the large quantity of images generated during the training, we only save the first 3000 images.
The accuracy of the model stable at about 0.93 after 33000 steps.

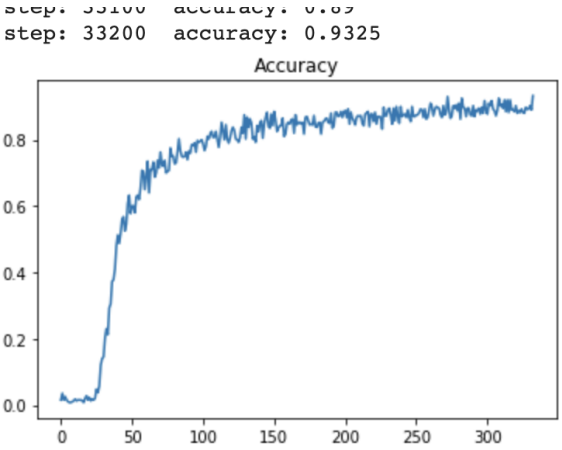


Figure 5: Accuracy

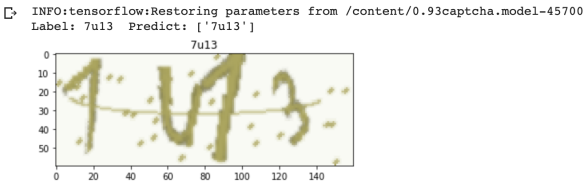


Figure 6: Predict value

# References

[1] 'CAPTCHA Recognition with Active Deep Learning', Fabian Stark, Caner Hazirbas, Rudolph Triebel, and Daniel Cremers, 2015

[2] 'Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks',Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet

[3] From dukn's GitHUb https://github.com/dukn/Captcha-recognition-Keras

[4] http://matthewearl.github.io/2016/05/06/cnn-anpr/