Captchas

UOA

***Abstract*—**

I. Introduction

II. RELATED WORK

III. HYPOTHESIS

IV. METHODOLOGY

A. Experiment 1

The original CNN model from Ye et al. [1] has 5 convolution2D layers so that the model is more generally applicable to predict the label of text-base CAPTCHA of a wide range of modern CAPTCHA schemes. According to LeNet-5 [2] model, we have modified the model slightly with 3 Conv2D layers and compared the performance between these two models. In our experiment, we used the CAPTCHA scheme from JD.com which contains uppercase alphabets and digits (some of more likely confusing characters are excluded). It has taken 43 hours to train the 3 Conv2D layers CNN model with 100,000 generated images and 200 epochs on a 8xCPUs machine (no GPU), and 2 more hours are needed to train a 5 Conv2D layers CNN. The mean validation accuracy are 14.9% and 8.8% for 3 Conv2D layers and 5 Conv2D layers models respectively. Figure 1 displays the performance comparison between the original [1]'s model and simpler model by our modification. We can find that the loss is flatten after 50 epochs and their performance are quite similar on average.

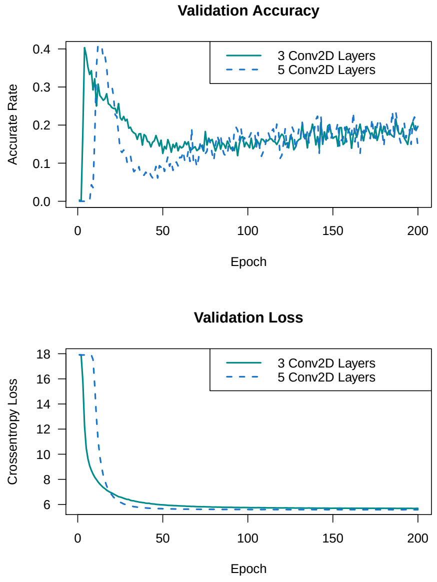

Fig. 1. CNN Validation Performance on Ye et al. Solver [1]

B. Experiment 2

A lot of CNN models built to break CAPTCHAS assume that the number of characters in the CAPTCHA image is fixed or is already known. To recognize label of CAPTCHAS with variable length of text, we have created a CNN model to identify the number of letters and numbers in a single CAPTCHA image. The CNN model contains 3 convolution2D layers with sub-sampling (pooling) layers, followed by 2 fully connected layers. The python code is based on Keras library, input data (images) is pre-processed by Pillow library (such as re-size and gray-scale), and the trained model is saved into JSON (for CNN structure) and hdf5 (for weights).

Our experiment is to identify the number of characters in an image which may contains 2 to 6 letters or numbers. We used 25,000 images in total, with 5000 images for each length respectively. It takes about 10 hours to train the CNN model for 64 epochs on a machine without GPU. The overall validation accuracy is as high as 96.4% after 30 epochs. Figure 2 shows the accuracy and categorical cross-entropy loss during the training process. We can see that after 30 epoch, the validation loss and accuracy become flat.

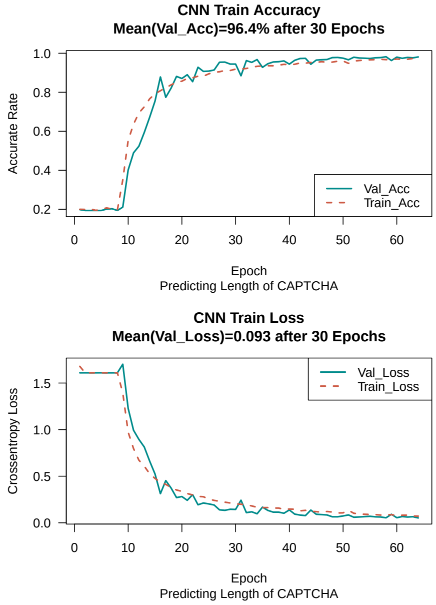

Fig. 2. CNN Training Performance

When using this model to predict the length of 2000 CAPTCHAs, the overall accuracy is 98.2%, and the mean time of prediction an image is only 30 milliseconds. Figure 3 indicates the prediction accurate rate for each length group and they all have over 95% correctness. By this experiment, we have presented the feasibility to predict the length of CAPTCHA image using CNN model with a very high accuracy rate.

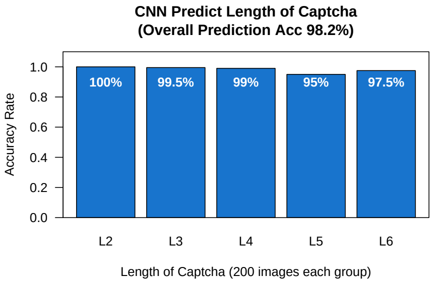

Fig. 3. CNN Prediction for Each Length Group

V. CONCLUSION

REFERENCES

[1] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng, P. Xu, X. Chen, and Z. Wang, "Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, (New York, NY, USA), pp. 332–348, ACM, 2018. event-place: Toronto, Canada.

[2] Y. LeCun, L. Bottou, Y. Bengio, and others, "LeNet-5, convolutional neural networks," 2016. http://yann.lecun.com/exdb/lenet/.