

Summary Report

Summary details of file retrieval:

I used a java jar which opens the csv file and uses java-nio package to download and save the response as a .png

Timing: It took relatively the same time on Pi as on the local. The bandwidth on the pi was high, and running the jar showed no difficulties at all. There's not too much difference between read/write file speeds of Raspberry Pi 4 and local machine (macbook)

Summarized Details of Preparation/Pre-Processing:

Approach 1:

Used padding; Padded the file with a char – space in this case and generated the images with the padded name by changing Generate.py.

Generate one set of dataset for training the model with 36000 images, and another set of 3600 images for validation keeping a 90-10 ratio.

Approach 2:

Same Pre-Processing as above and used keras.Regularizer.l1 and l2 and attempted penalizing the extra space parameters.

Approach 3: Used Max Pooling with the set value of (2,2), along with strides of (2,2) and (3,3)

Approach 4: Used a Gray image, inverted the hues to create a black and white inverted image with numbers in white and background in black.

Approach 5: Segregated the images into different size lengths (brute force, did manually)

(Note: Can be automated using object detection algorithms like YOLO which builds on Sliding Window Object Detection in the way that instead of running each window through a CovNet, you run a CovNet on the whole image and then at different places in the net, you run final object classification and bounding box regression.)

Then trained each model for specific length and generated output.

Timings: Only when the generation size is greater than 10000 it takes time to create the images. Timings were not measured, but it usually takes less than a minute in both the cases.

Summarized Details on Training Set Creation:

To cover all possible cases for a length of 6, the possible arrangements from a unique symbol set of 60 is : $60 \times 59 \times 58 \times 57 \times 56 \times 55$. This is a huge number. Instead I generated 36000 and 40000 images for the training set for length 6 case.

Timings: For length 6 it takes roughly 14-17 minutes per epoch to train the above set. For 10 epochs it takes roughly 140-170 minutes. It will definitely take more time on the pi as it lacks a GPU. Considering a 100% increment, we can assume a 28-34min per epoch training time.

Summarized Details on Validation Set Creation:

Validation set is ~10% of the size of training set. (I wanted to check if cross-validation would've improved the accuracy, but couldn't implement that)

Timings: Generation takes the same time. There's not too much difference between read/write speeds of Raspberry Pi 4 and local machine.

Summarized Details on Submittity Solving:

Finally the output.csv was generated individually for each case, which I collated and used Excel to sort the image names (column 1) from A-Z. This sorts it in the required format of 0-9a-f.

Timings: Not applicable

Useful Links:

Complete Deep Learning Course – CNN explained in a few videos.

<https://www.youtube.com/playlist?list=PLZoTAELRMXVPGU70ZGscrMdr0FteeRUi>

How CNN Algorithm works:

<https://www.youtube.com/watch?v=oI2rvjbzVml>

Captcha recognition using CNN

<https://medium.com/@manvi./captcha-recognition-using-convolutional-neural-network-d191ef91330e>

Novel Captcha Solver Implementation

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9044336/>

Supplements:

captcha:

- 1) <https://towardsai.net/p/deep-learning/deep-learning-based-automatic-captcha-solver>
- 2) <https://medium.com/@ageitgey/how-to-break-a-captcha-system-in-15-minutes-with-machine-learning-dbebb035a710>

training ideas:

- 1) <https://www.dlology.com/blog/how-to-train-a-keras-model-to-recognize-variable-length-text/>

segmentation and object detection ideas:

- 1) <https://stackoverflow.com/questions/42364513/how-to-locate-multiple-objects-in-the-same-image>

improve accuracy ideas:

- 1) <https://stackoverflow.com/questions/44043544/how-to-increase-low-accuracy-on-captcha-recognition>
- 2) <https://stackoverflow.com/questions/53993955/keras-cnn-training-to-recognize-captcha-get-low-loss-and-get-low-accuracy>
- 3) <https://stackoverflow.com/questions/69536569/how-to-get-this-ocr-model-to-work-with-variable-length-examples>

Github:

- 1) https://github.com/krishna1633/Captcha-Detection/blob/master/train_model.py
- 2) https://github.com/DrMahdiRezaei/Deep-CAPTCHA/blob/master/Captcha_Solver_numerical.ipynb
- 3) <https://github.com/dukhn/Captcha-recognition-Keras/blob/master/Train.py>