

# Assignment

Sayak Kundu

Since the images had noise in them and the digits were of different width and some were occluded, using just OpenCV wasn't enough. It took around 12 hours to first create the dataset for each and every image in the training set. The training set had 5 columns. Given below is an example showing data annotation for number 00012 -

0	0.285185	0.528571	0.111111	0.600000
0	0.396296	0.528571	0.111111	0.542857
0	0.500000	0.457143	0.111111	0.571429
1	0.607407	0.514286	0.118519	0.571429
2	0.725926	0.557143	0.118519	0.600000

It contains the digit, x-coordinate of centre, y-coordinate of centre, width and height respectively. The measurements were in pixels.

```
python3 captcha-single.py --image test.jpg #for  
single image  
python3 captcha.py #for all images in test folder
```

Also there were instances of datasets which were labelled wrongly in both test and train dataset. So I had to go through the dataset again for checking that if annotations are correct or not. I used pretrained model (darknet) for faster learning along with YOLOv3. For using YOLOv3, I created a config file and the final weights are also saved in the submission. The accuracy on the test data is **98.16%**. The result file contains the file name, output, time taken to calculate and if the predicted value matches with the true value or not.