Machine Learning Internship | Navia Life Care

Name	Uma T V
Email	uma.tv1699@gmail.com
Roll No	ME17B170
Institute	Indian Institute of Technology Madras

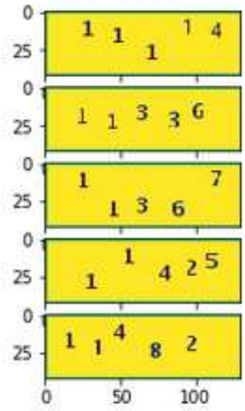
Github Link for the code: https://github.com/Uma1699/ML_Internship_Navia_Life_Care.git

The task of predicting the captcha images provided is carried out in 3 steps:

01 Splitting_Component_Numbers_From_Tagged_Captchas

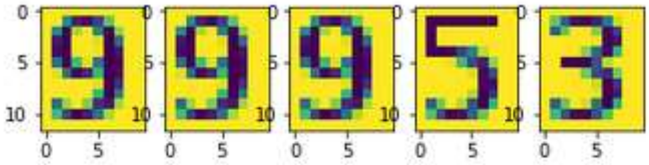
First of all, the electoral-tagged images and filenames are imported and split into the component numbers. The pixel information of the split numbers is stored along with the labels of the split numbers, thus providing a training dataset for individual number identification.

- It was noticed that the captchas are quite simple, in the sense that the numbers are neither rotated nor is there noise in the images. The numbers are merely translated, with sufficient gap between one another. Hence, we did not need to use neural networks. We could easily split the numbers using black pixel detection and classify the numbers separately.



The Captchas Provided are quite simple

- Another reason for not using neural networks is the insufficient number of training images (only 940), and large number of test images (1500). Hence, it is best to split the numbers (then we will get 940*5 = 4200 training images which can be easily learnt by machine learning algorithms.



The Split number images

- Splitting is done based on vertical segments of black pixels separated by vertical segments of white spaces. The standard size of individual numbers is taken to be 12*10.

02 Training_Split_Numbers_PCA_SVM

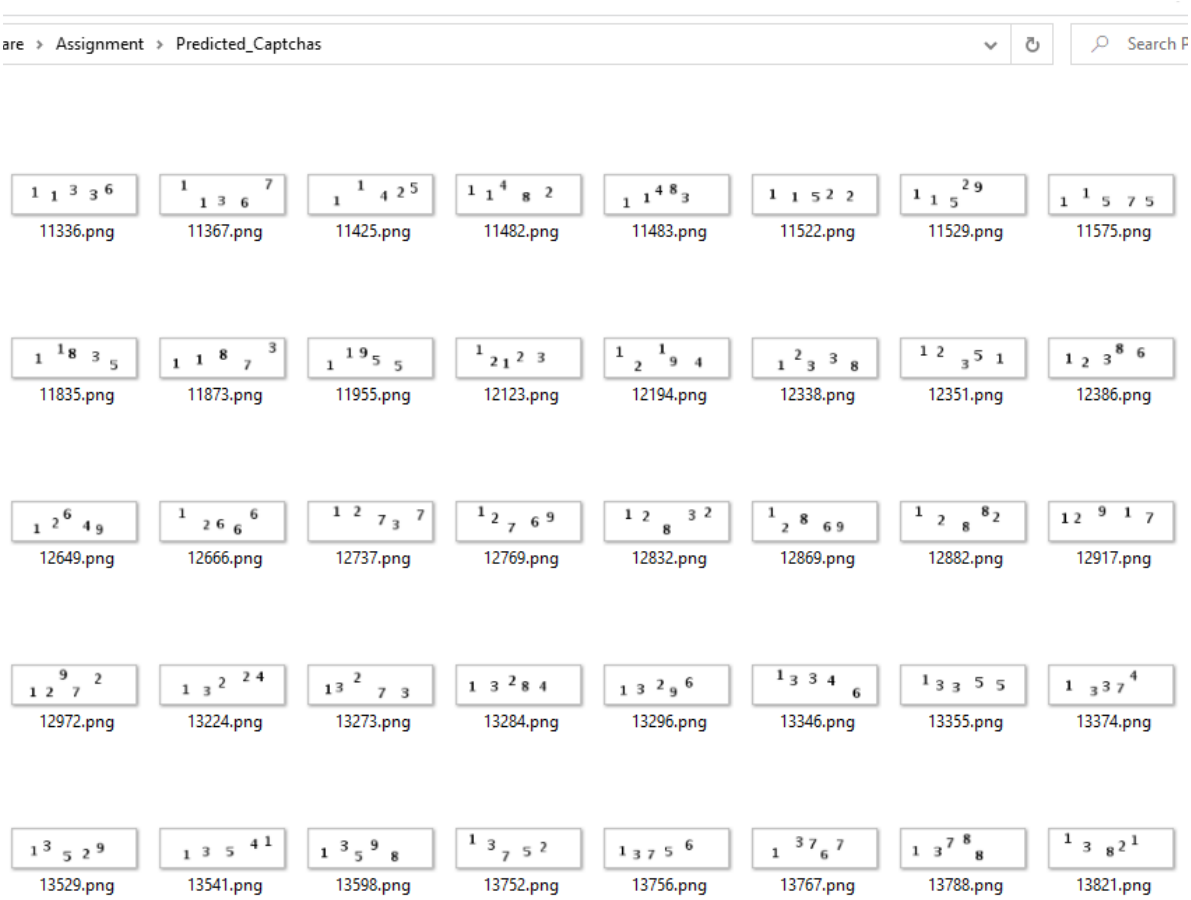
The split processed individual numbers information (output of program 1) is loaded and used to learn and predict numbers from split number images.

03 Predicting Test Tagnames

The most efficient model from program 2 is used to predict the tag names of the test images given in electoral-captchas, through individual number splitting analysis. The predicted values are stored as filenames to the test images in another folder created, called “Predicted_Captchas”.

The explanation of individual steps is provided as markdowns and comments in the codes.

Result :



11336.png 11367.png 11425.png 11482.png 11483.png 11522.png 11529.png 11575.png

11835.png 11873.png 11955.png 12123.png 12194.png 12338.png 12351.png 12386.png

12649.png 12666.png 12737.png 12769.png 12832.png 12869.png 12882.png 12917.png

12972.png 13224.png 13273.png 13284.png 13296.png 13346.png 13355.png 13374.png

13529.png 13541.png 13598.png 13752.png 13756.png 13767.png 13788.png 13821.png

The predicted tag names are mostly almost hundred percent accurate on manual inspection

The training accuracy of individual numbers recognition is 0.9970215027209504. The test “electoral-captchas” images Predictions are very accurate and no errors were found through manual inspection.