CS771 Introduction to Machine Learning Assignment 3

Bharat
22111074
bharat22@iitk.ac.in

Pradeep Chalotra
22111045
pchalotra22@iitk.ac.in

Pulkit Sharma
22111048
Pulkits22@iitk.ac.in

Sarthak Neema
22111079
sarthakn22@iitk.ac.in

Sumit Kumar Chaudhary
22111060
sumitkc22@iitk.ac.in

1 Pre – Processing Phase :

The provided images were in RGB format with obfuscation lines, making it difficult to separate individual characters. To make the image acceptable for segmentation, the following steps were taken:

- Convert the RGB image to GRAYSCALE format.
- Learned the color of the background and subtract the color from the image to effectively nullify the background pixels.
- The image was then eroded using the cv2.erode() function with a 5x5 kernel over the course of 6 iterations. By selecting 30 random images for iterations 2, 3,4,5,6,7,8,9 of a normal 5x5 kernel, we were able to determine the number of erosion iterations. From that we found 6 to be suitable for the removal of obfuscation lines.
- We then dilate the image using cv2.dilate() function using a 5x5 kernel for 1 iteration. This step is done to gradually enlarge the boundaries of regions of foreground pixels.

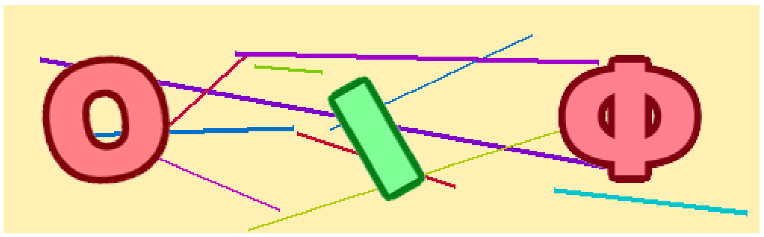


Figure 1: Image before preprocessing

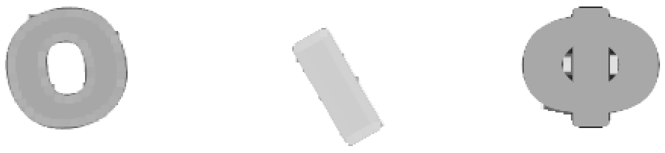


Figure 2: Image after preprocessing

2 Segmentation Phase :

- After carefully examining the provided CAPTCHA training examples, we found that despite the images being tilted around a pivot, the letter boundaries do not cross.

- By scanning vertically, which would help separate neighbouring characters, we can separate the characters in the image.

- By sweeping the entire image with a vertical line from the left end to the right end, the characters from the provided CAPTCHAs can be separated (column vector).

- When we found a bright pixel in the sweeping column vector, we inferred that the pixel was the beginning of a contour. This was only possible because a vertical line divided the characters. We further thought that once every pixel in the sweeping vector had expired, we had completed sweeping over a significant character. In order to correctly scan the final character of the CAPTCHA, either all of the pixels on the sweeping vector turned black or we reached the end of the image.
- Then we get the number of characters present in the image as well as the individual segments containing a character each.

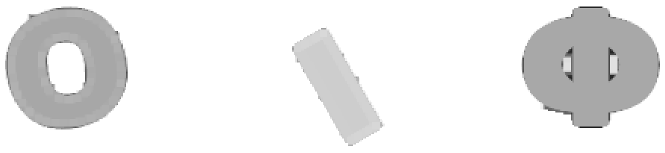


Figure 3: Image before segmentation

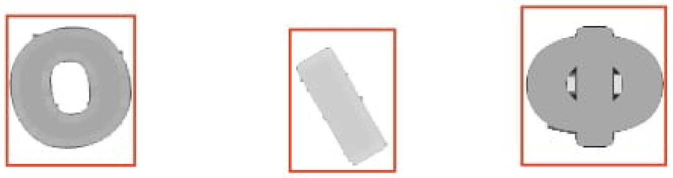


Figure 4: Image after segmentation

3 Classification Phase :

- After getting the separated characters we feed them to a SVM classifier to find which characters are present in the segmented image. First, all the pixels were resized to 30x30 pixels for input to SVM classifier and the output labels were mapped using a dictionary (in the dictionary we have keys as all the unique 24 characters) to numerical value. Then we dumped the file which is to used during prediction.

- We used scikit learn train test split of 75-25 and the scikit-learn GridSearch function to tune the hyperparameter ‘C’.

- For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

- GridSearchCV tries all the combinations of the values passed in the dictionary and evaluated the model for each combination using the Cross-Validation method.

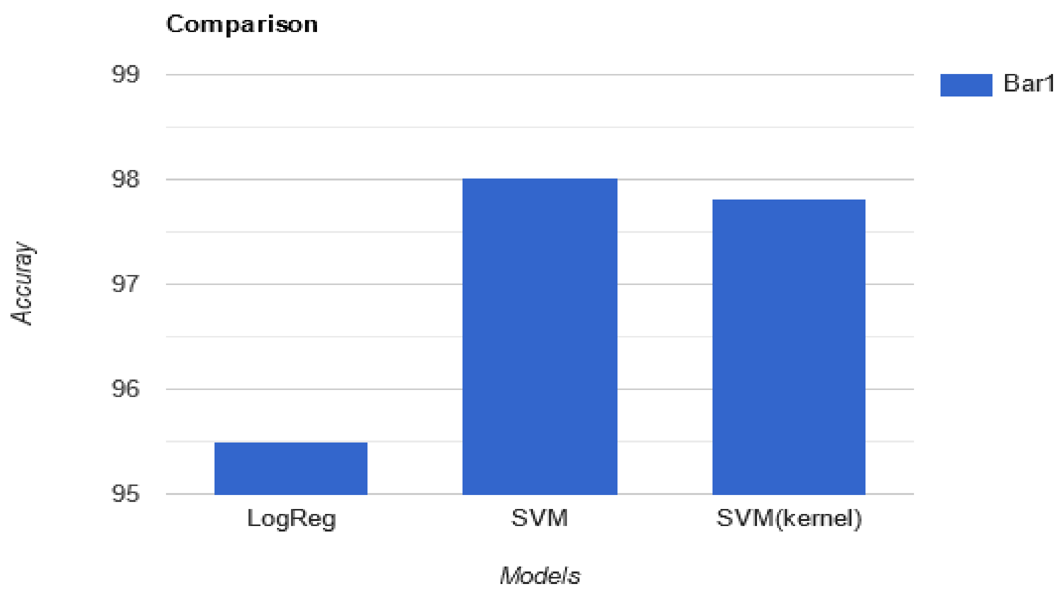
Why SVM(linear) as classifier?

- For the image classification we have used Logistic regression, SVM(kernel=’rbf’) and SVM(linear).

- For Logistic Regression as classifier, we’re great very space efficient model but is less than as compared to SVM(linear).

- For SVM(kernel=’rbf’) as classifier, we’re getting very good accuracy as compared to SVM(linear) but the model size is quite high as compared to SVM(linear).

That’s why we choose SVM(linear) as our classifier.


Comparison
Accuray
Bar1
LogReg
SVM
SVM(kernel)
Models