Assignment 3

Indian Institute of Information, Kanpur
Department of Computer Science
CS771 Introduction to Machine Learning
Instructor: Purushottam Kar

Submitted: November 26, 2022

Ambati Madhav
160098

Kudupudi Mohan Kumar
22105037

Abhinav Kuruma
22111401

Gnanendra Sri Phani Sai Channamsetty
22105032

Rahul Aggarwal
22111403

Contents

1 Answer 1

1.1 Problems and patterns identified with the given training data

- Background color is always light shade as seen in Fig. 1
- Obfuscating lines are always dark shade as seen in Fig. 1
- Three Characters have not been segmented as seen in Fig. 1
- The reference images have been rotated, potentially decreasing the prediction accuracy of the model, as seen in Fig. 1

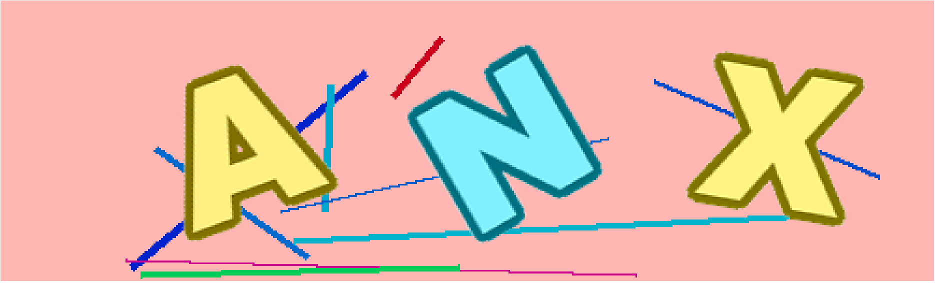

Fig. 1 Training Data Sample

1.2 Approach to Solution

The solution method ultimately selected includes a pipeline consisting of pre-processing features and SVM Solver, as described in 8 steps in section 1.2.1 to 1.2.8.

1.2.1 Step 1 : Segmentation of image into three individual pieces

Identification of background pixel color helped with the process of segmenting each training image into three individual pieces, , each containing a character as seen in Fig. This was achieved by looking for vertical columns of pixels that contain at-least some 50 non-background pixels. Three such sets containing minimum number of non-background pixels were obtained and used this information to get the indices for segmentation of images. Refer function "GetIndices" in code and see Fig. 2 .

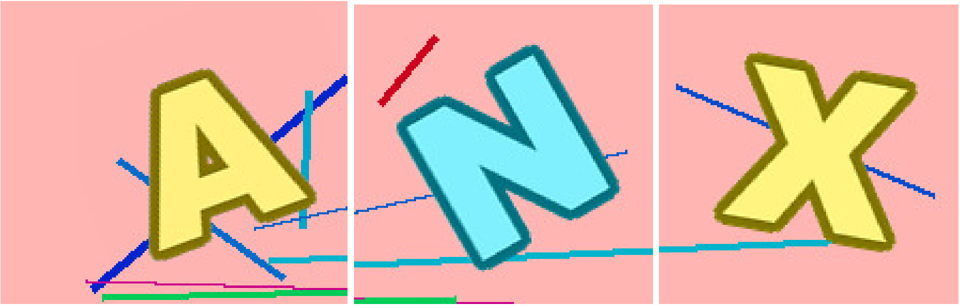

Fig. 2 After Segmentation Step

1.2.2 Step 2 : Pre-processing of Features through identification of pixels that represent background color

Background color is always light shade while obfuscating lines are always dark shade. However, the corners of the the images are almost always background color. Pixel Color has been represented in Greyscale format instead of the more ubiquitous RGB format to better identify the background pixel color. Background pixel colour has been identified from top-left corner pixel. Refer function "initialpreprocess" in code and see Fig. 3 .

1.2.3 Step 3 : Pre-processing of Features through identification of pixels that represent obfuscating lines

Obfuscating lines have been mostly eliminated through erosion technique, with the help of functions and kernels from OPENCV library. The erosion parameter with Structured Kernel, iteration =1 has been identified to be performing well on sample of test images. Hence, these values have used them during erosion. Refer function "initialpreprocess" in code and see Fig. 3 .

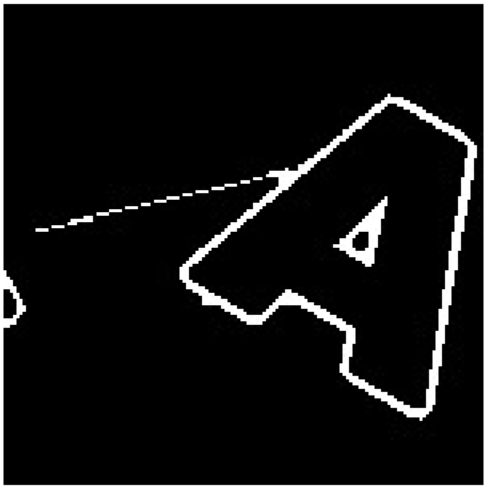

Fig. 3 After elimination of Obfuscating Lines and Background color through "initialprocess" function

1.2.4 Step 4 : Translation and bringing character to centre

Within each segmented individual piece, the character is off-centre. Translation is used to move the desired foreground to the center of the image. Chose 150 x 150 as a standard size for all the images. Center of mass of the images was found and later moved the image to the frame’s center i.e H/2, W/2 where H, W are height and width of the frame. The whole image is cut and resized in such a way that it maintains the dimensions of 150 x 150 and the required foreground is in the center of the frame. To get the character almost in the center of segmented image we averaged the vertical columns position in the set and obtained equal number of vertical columns on left and right of that average position. Refer function "GetIndices" in code and see Fig. 4.

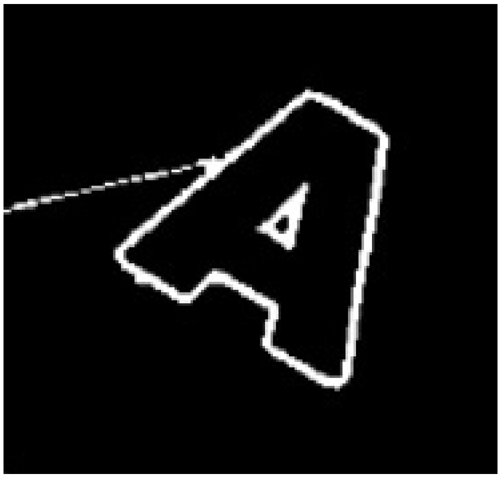

Fig. 4 After Translation Step

1.2.5 Step 5 : Rotate character to it's correct position

Within each segmented individual piece, the character has also been rotated.Translated image is Rotated. Firstly,the contours in the image were found and filtered out the required countours based on the area.The contours needs to be arranged in a matrix with size n x 2, where n is the number of data points . Then PCA analysis has been performed. The calculated mean (i.e. center of mass) is stored in the "cntr" variable and the eigenvectors and eigenvalues are stored in the corresponding variables. Thetangent angle between first two eigen vectors is calculated to get the angle a contour/image of interest is rotated. This is done in function "getOrientation" in code. Later the image is rotated along the center about the angle obtained above opposite to its default direction to correct the rotation . See Fig. 5.

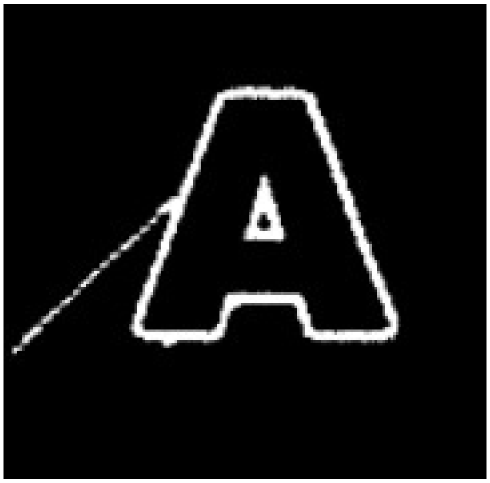

Fig. 5 After Rotation Step

1.2.6 Step 6 : Extracting feature map of each character

Used hog filter in sklearn image library to get the feature map out of those pre-processed images and sent that to SVM Solver.

1.2.7 Step 7 : Implementation of three solver Methods and Selection of Solver Method

Three solver methods have been attempted.Their methodology names and shorthand have been given below.

1) Support Vector Machine((SVM)
2) Decision Tree(DT)
3) Convolutional Neural Networks(CNN)

SVM has been selected as the final solver used in the code. The reasons have been provided in Section 1.2.8

1.2.8 Step 8 : Three Advantages of SVM Solver Method adopted over other attempted implemented methods

1) Decreased Prediction Time for 1000 samples[421.8 ms for SVM vs 452.6 ms for DT vs 127 ms for CNN] as seen in Fig 6
2) Decrease in Model Size[10.5 MB for SVM vs 15.7 MB vs 333.3 MB for CNN] as seen in Fig 7
3) Increase in Prediction Accuracy[97% for SVM vs 95% for DT vs 95% for CNN] as seen in Fig 8

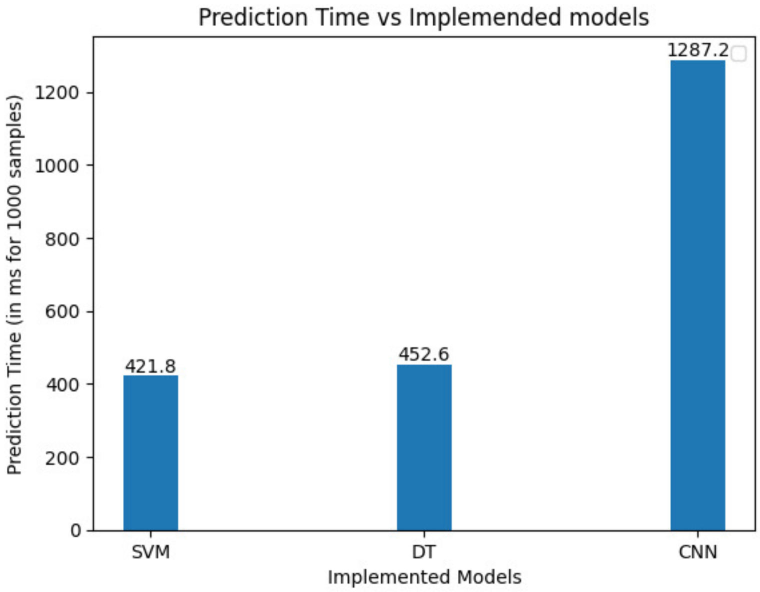

Fig. 6 Prediction Time of Implemented Models

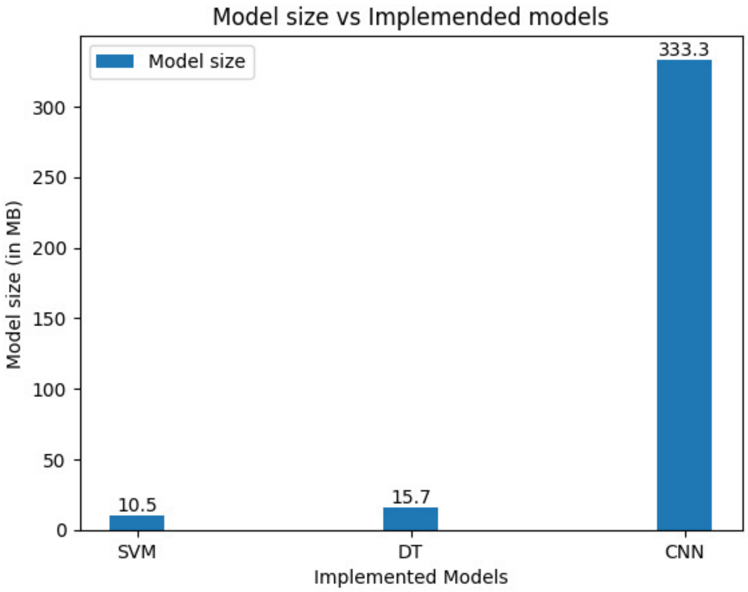

Fig. 7 Model Size of Implemented Models

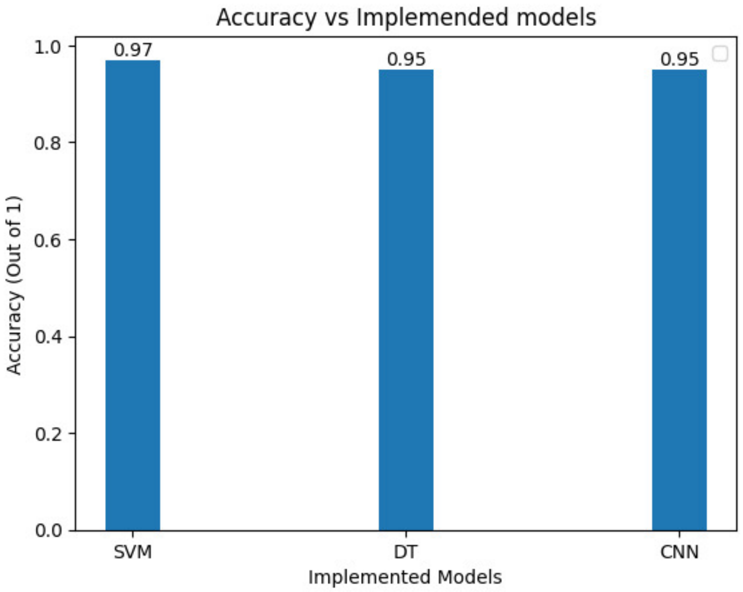

Fig. 8 Accuracy of Implemented Models

2 Answer 2

<code link> : https://cse.iitk.ac.in/users/rahulag/Assn3/submit.zip

3 References

[1] https://deepdatascience.wordpress.com/2017/04/22/dense-word-vector-representation/

[2] https://web.cse.iitk.ac.in/users/purushot/courses/ml/2022-23-a/discussion.html

[3] https://docs.opencv.org/3.4/d1/dee/tutorial$_{i}ntroduction_{t}o_{p}ca.html$