CS771 : Assignment 3 : [Group 10]

Aakash [19111261] **Dhanish Kumar [170243]**

Drashti Pathak [19111032] **Lt Cdr Dhurender [19111030]**

Rohit Chouhan [170590] **Shreya Sharma [19111087]**

QUESTION NO. 1

- **Details Of Algorithm Used : -**

  1. We have undertaken the Background modification to get the letters in a state such that we can apply Neural Network onto it.

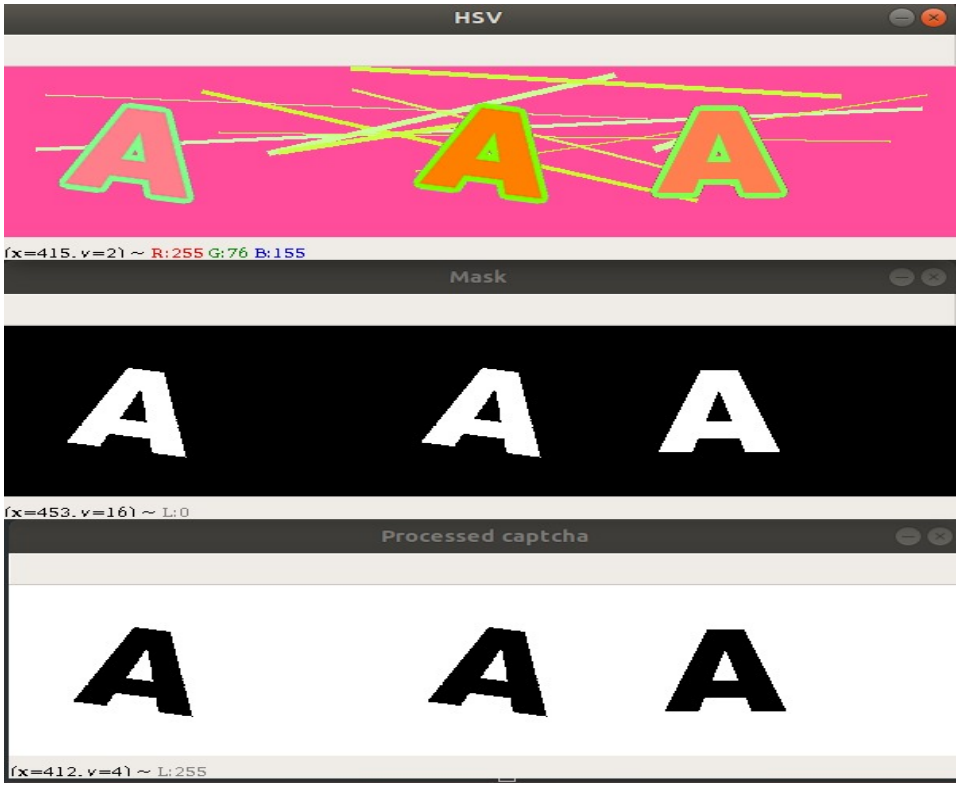

Figure 1: Converting the image to simpler form

Firstly, we have converted the BGR(.png) image to HSV(Hue, Saturation, and Value), so that we can use it to extract the letters from the given images. To undertake this

Preprint. Under review.

task, we used the OpenCV Library for it. We also made sure that the color difference between the letters and the background is high so that the Neural Network works smoothly.

By doing so we also got rid of the lines in the background as can be see from the second image of [Figure 1].

2. Then we split the image so that each letter is a separate image in itself, then we only have to train the neural network model to recognize a single letter at a time.

   Problem encountered : - : - We can't just split the images into three or four equal size images because the Captcha randomly places the letters in different horizontal locations.

   Solution : - Like in image processing, we need to detect blobs of pixels that have the same color. so we did the same thing, we have just focused on the contours. OpenCV has a built in findContours() function that we have used to detect these continuous regions.

   Approach followed : -

3. First, we have started with a captcha image given below [figure 2].

Figure 2: Image after background separation

4. And then we have converted the image into pure black and white form, so that it will be easy to find the continuous regions [Figure 3].

Figure 3: Image in pure black and background pure white

5. Finally, we have used Contours() function from OpenCV library to detect the separate parts of the image that contain continuous blobs of pixels of the same color [Figure 4]. At prediction time, we have counted these separate parts of image to predict total number of letters present in the Captcha.

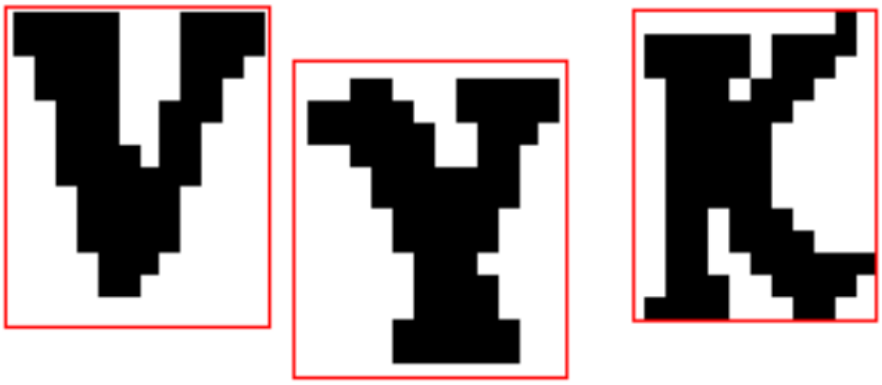

Figure 4: detecting separate character's

6. **Building and Training of the Neural Network**

   We have used a simple Convolutional neural network architecture with two Convolutional layers and two fully connected layers. The flow of NN is displayed in [Figure 5].

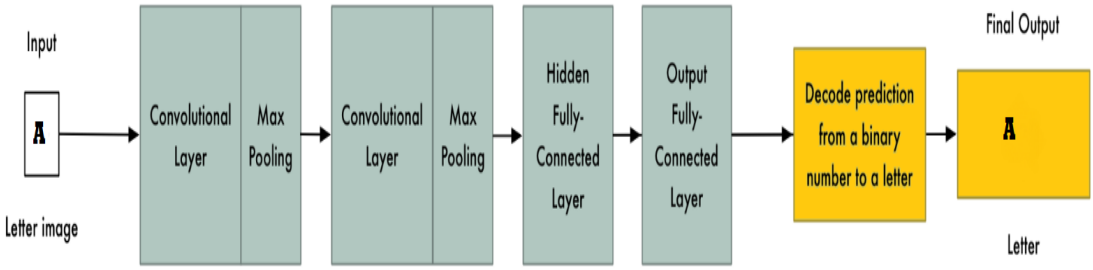

Figure 5: Identifying particular later

- **Steps followed while Training the Model : -**

  1. Initialize the data and labels.
  2. loop over the input images
     (a) Loaded the image and converted it into grayscale.
     (b) Resized the letter such that it fits in a 20x20 pixel box.

(c) Added a third channel dimension to the image to make Keras happy.

(d) Grabed the name of the letter based on the folder it was copied in.

(e) Added the letter image and it's label to our training data.

3. Scaled the raw pixel intensities to the range [0, 1] (this has improved training).

4. Divided the training data into separate train and test sets.

5. Converted the labels (letters) into one-hot encoding, which were used by Keras to work with.

6. Saved the mapping from labels to one-hot encoding. (We have utilised it, later, when we have used the model to decode what it's predictions mean).

7. Builded the neural network.

8. Performed the Max Pooling in the first and second Convolutional layer.

9. Asked Keras to build the TensorFlow model behind the scenes.

10. Trained the neural network.

11. Finally Saved the trained model to disk.

Important : We have implemented using Hidden layer with 500 nodes and Output layer with 26 nodes (one for each possible letter we predicted). Rectified Linear Unit activation function is used for hidden layer and for output layer, softmax activation function was used.

REFERENCES: -

1. https://school.geekwall.in/p/S1CqSgRZ7/how-to-break-a-captcha-system-with-machine-learning by Adam Geitgey, Software Engineer / Consultant, writer of Machine Learning is Fun!

2. https://s3-us-west-2.amazonaws.com/mlif-example-code/solving$_c aptchas_c ode_e xamples.zip$