

# CS771 : Assignment 3 : [Group 10]

## Group Members

Aakash [19111261]

Dhanish Kumar [170243]

Drashti Pathak [19111032]

Lt Cdr Dhurender [19111030]

Rohit Chouhan [170590]

Shreya Sharma [19111087]

## Question 1

### • Details Of Algorithms :

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

First we 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. For this 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 above.



Figure 1: Converting the image to simpler form

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

**Problem :** 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 just focus on contours. OpenCV has a built in findContours() function that we can use to detect these continuous regions.

So we will start with a captcha image given below [figure 2].



Figure 2: Image after background separation

And then we'll convert the image into pure black and white so that it will be easy to find the continuous regions.



Figure 3: Image in pure black and background pure white

And then, we will use OpenCV find Contours() function to detect the separate parts of the image that contain continuous blobs of pixels of the same color. At prediction time, we have counted these separate parts of image to predict total number of letters present in the captcha.

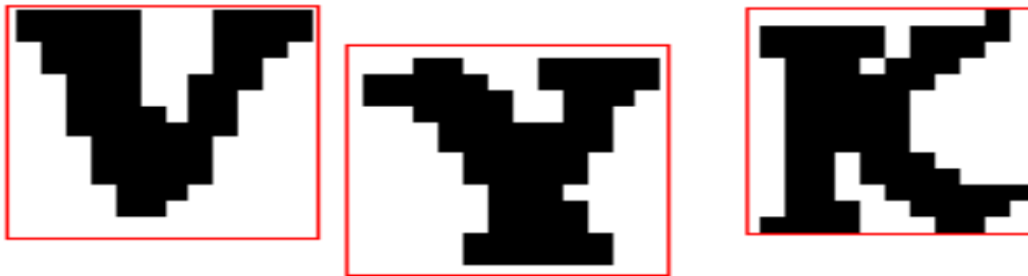


Figure 4: detecting separate character's

**New Problem :** Sometimes the captcha's have overlapping letters.

**Solution :** A simple idea here is that if a single contour area is a much wider than it is tall, that means we probably have two letters overlapping together. In that case, we will just split the joined letter in half down the middle and treat it as two separate letters.

finally we end up with individual letters.

### 3. Building and Training the Neural Network.

We will use a simple convolutional neural network architecture with two convolutional layers and two fully connected layers.

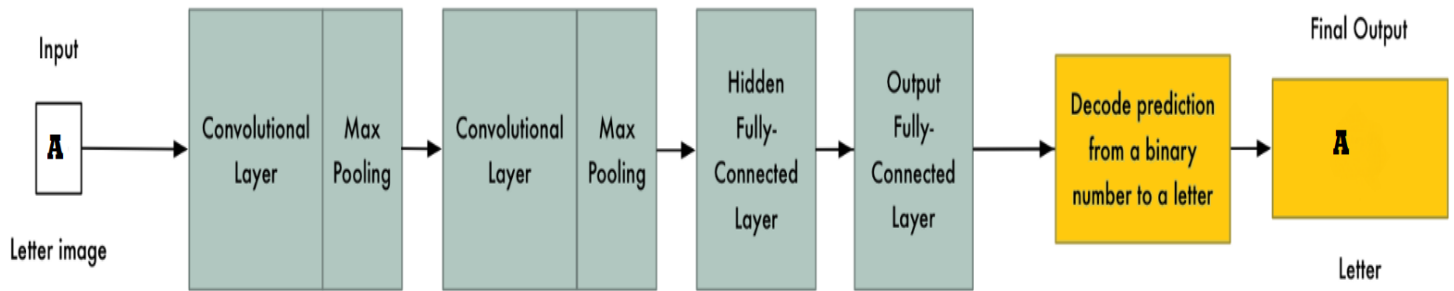


Figure 5: Identifying particular later

#### • Steps in Training Model :

1. initialize the data and labels
2. loop over the input images
  - (a) Load the image and convert it to grayscale
  - (b) Resize the letter so it fits in a 20x20 pixel box
  - (c) Add a third channel dimension to the image to make Keras happy
  - (d) Grab the name of the letter based on the folder it was in
  - (e) Add the letter image and it's label to our training data
3. Scale the raw pixel intensities to the range  $[0, 1]$  (this improves training)
4. Split the training data into separate train and test sets
5. Convert the labels (letters) into one-hot encodings that Keras can work with
6. Save the mapping from labels to one-hot encodings. (We'll need this later when we use the model to decode what it's predictions mean)
7. Build the neural network.
8. First and Second convolutional layer with max pooling
9. Ask Keras to build the TensorFlow model behind the scenes
10. Train the neural network
11. Finally Save the trained model to disk

**Important :** Hidden layer with 500 nodes and Output layer with 26 nodes (one for each possible letter we predict). Rectified Linear Unit activation function is used for hidden layer and for output layer, softmax activation function is used.

#### • References

1. <https://school.geekwall.in/p/S1CqSgRZ7/how-to-break-a-captcha-system-with-machine-learning>
2. <https://github.com/tharidu/breakingcaptcha>