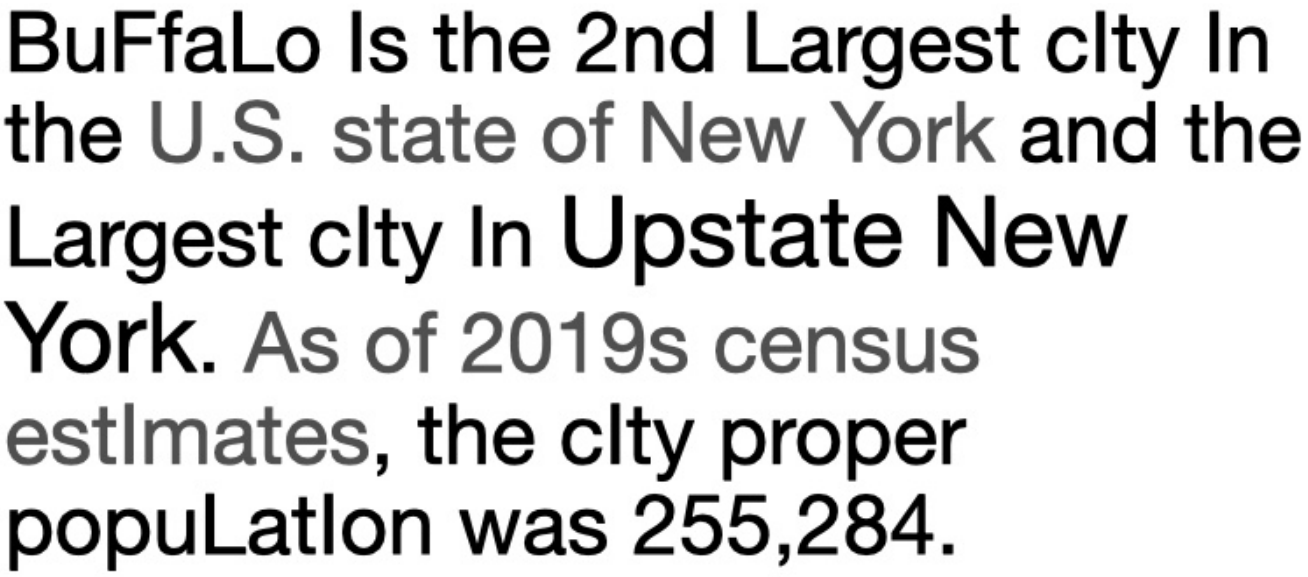
Optical Character Recognition Project

Rohith Reddy Kolla

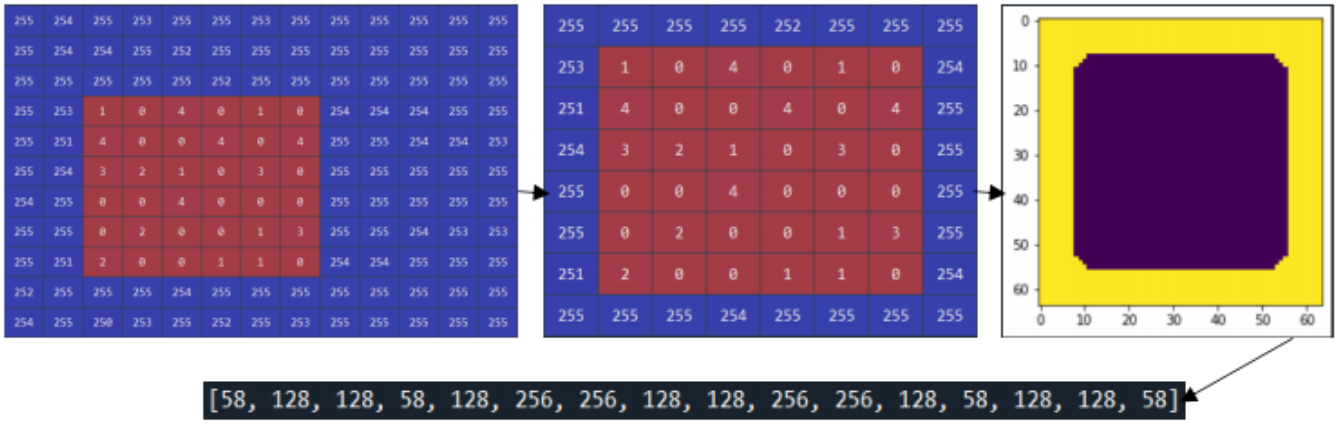
- The goal of the project is to identify features from the given images of characters and use these features to identify the characters from the test image.

Character Images

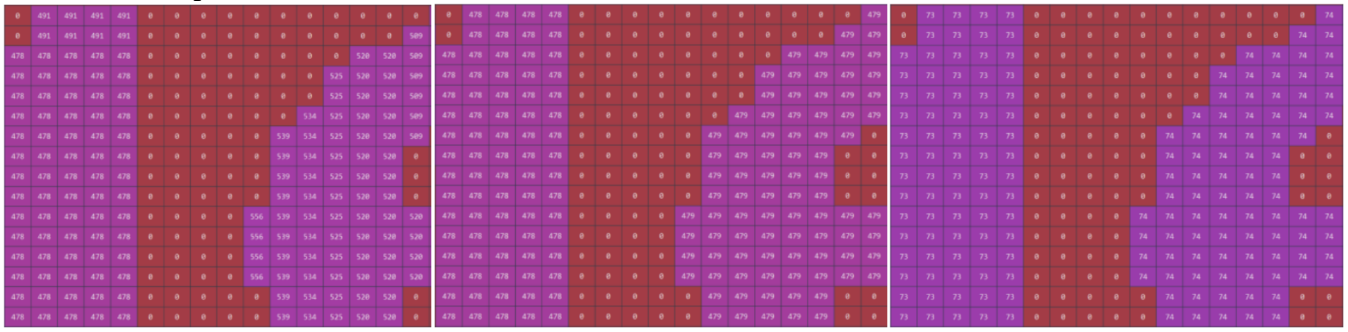


Test Image

- Computing Features – The process of computing the features from the given characters for enrollment occurs in several steps. Firstly, the array of the greyscale images is modified such that unnecessary whitespace is removed. Following this, the array is binarized based on a certain threshold (in this case anything less than 127 becomes 0 and anything greater or equal becomes 255). The array is then resized to the dimension 64x64 using the cv2.resize() function and is binarized again. The 64x64 array is then subdivided into 16 16x16 arrays from which the total number of elements equal to 0 is calculated and used to generate a 16x1 sized feature vector. This process is shown using the example of the "dot" character below. (Note - The third image is a plot instead of a spyder array like the two before since spyder is not able to display 64x64 array in one screen).



- Performing Detection – The detection is done using 3 passes over the test_img. Each pixel or element in the array is considered from the top left to the bottom right. If the element is a background pixel (value greater than 255 – threshold), it is given a label 0. If the element is not a background element, the element on top of it and the element to the left are considered. If the labels of these elements are 0, a new number is assigned as the label for the current element. If only one of these labels is non-zero, then the current element is given the same label as the non-zero element. If both the elements have non-zero labels, then the label with the lower number is assigned to the current element and the value of the two labels is stored in a list. At the end of the first pass, each non-background element is assigned a certain label and all the label couples that are in contact with each other are stored in a list. This list is then processed to generate a new list of lists where all the labels in contact with each other are stored in the sub lists. Over the second pass, the labels are modified such that all the elements connected to each other are given the same label based on the generated list. In the final pass, the labels are numerically ordered starting from 1. Finally, the labels are used to generate a list of all the bounding boxes for each connected component.



First Pass Second Pass Third Pass

- Performing Recognition – In order to perform recognition, features are gather as mentioned above for each of the connected components individually using the bounding boxes, similar to the method for enrollment characters. The 16x1 feature vector of each connected component is compared to the feature vector of each of the enrolled characters to generate a term which is equal to the sum of the square of the differences between each element of the vectors. If the feature vectors are similar, a lower number is generated and vice versa. If the number is lesser than 10,000 the connected component is recognized as the character whose feature vector was used. This method is able to recognize the characters from the test image with an F1 score of 1.0.

```
C:\Users\X\Desktop\Project1>python evaluate.py --preds results.json --groundtruth groundtruth.json
1.0
```