Project 1: Optical Character Recognition

Name: Shreyash Singh
UBID: 550369166
UB Email Address: ssingh73@buffalo.edu

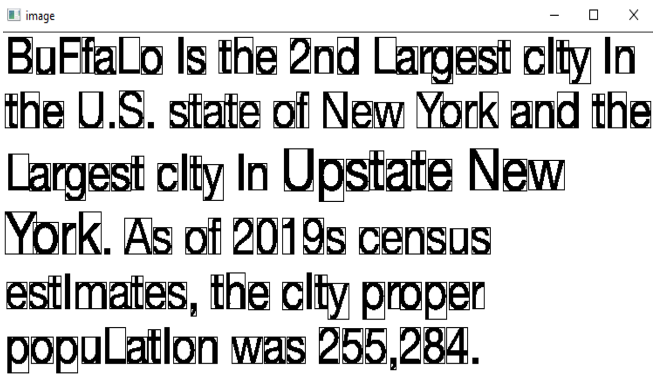
Goal: The Goal of this project is to implement an Optical Character Recognition System.

Steps:

1. Enrollment: Given a set of arbitrary number of characters, compute/extract features which will be used for Recognition
2. Detection: Given a gray scale test image containing characters to recognise, perform connected component labelling to detect various characters in the image.
3. Recognition: Using the features extracted in Step 1, implement a recognition or matching function which determines the identity of characters detected in Step 2, if it is in the enrollment set or UNKNOWN if it is not.

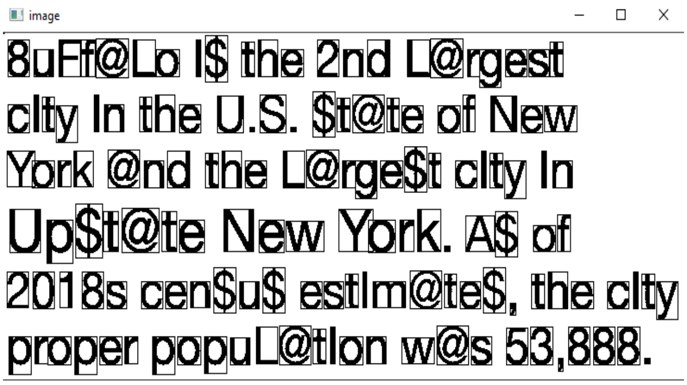
Code:

1. **Enrollment:**
   - For task 1, I used SIFT feature detector, which computes Key Points and Descriptors in each of the characters in the enrollment set and returns these features for recognition in step 3.
   - Features for each character (key points and descriptors), are stored into two lists containing features for each character and are returned by the enrollment function.
2. **Detection:**
   - For detection task, we were asked to implement connected component labelling to detect each candidate character.
   - I have implemented the 2-pass sequential connected component labelling algorithm, which iterates through each pixel in the test image.
   - In the 1st pass, the algorithm checks if a pixel is a foreground pixel or not. If it is a foreground pixel and has neighbours which are also foreground pixels, assign the label of any of the neighbours to the current pixel. If none of the neighbours are foreground pixels assign a new label and increment the label.
   - In the 2nd pass, after we have labelled each foreground pixel, we need to merge the equivalent regions together. We check if the neighbours have the same label, and if not assign the smallest label to the region and change all equivalent labels to the smallest label.
   - The Co-ordinates for each detected character are computed. (w, y, w, h)
   - Each detected character is stored as an individual character and their coordinates in respective lists and returned by the detection function for recognition.
   - Padding is added to each detected character.
   - Results of applying connected component labelling on the test images.



BuFfaLo Is the 2nd Largest clty In
the U.S. state of New York and the
Largest clty In Upstate New
York. As of 2019s census
estlmates, the clty proper
popuLatlon was 255,284.

Figure. 1 Detected characters in test_img.jpg



8uFf@Lo I$ the 2nd L@rgest
clty In the U.S. $t@te of New
York @nd the L@rge$t clty In
Up$t@te New York. A$ of
2018s cen$u$ estlm@te$, the clty
proper popuL@tlon w@s 53,888.

Figure .2 Detected characters for Evaluation Data test_img.jpg

3. **Recognition:**

- As the input arguments, the recognition function takes the features computed in the enrollment stage and the candidate characters and their coordinates computed in the detection stage.
- Each enrolled character is Upscaled and the candidate characters are resized to the size of each enrolled character to be matched to.
- Use SIFT feature extractor for extracting key points and descriptors in each candidate character.
- A FLANN based matcher is used for matching the features from each enrolled character to each of the detected candidate characters.
- Good matches are calculated by setting a threshold, and the maximum good matches for an enrolled character is used as the metric to check if it is a match and store the result in a python dictionary along with the coordinates of each candidate character.

Performance and Evaluation:

- The model gives an F-1 score of 0.73, for the given characters in the enrollment set and the test image provided in the .zip file, by running the evaluation.py file.
- Though SIFT may not be the best feature matching technique that can be used for Optical Character Recognition, it performs relatively well for the given data set.
- While it produces some incorrect matches, for the most part it gives rather good predictions on the training set.
- Some examples of good and bad matches:

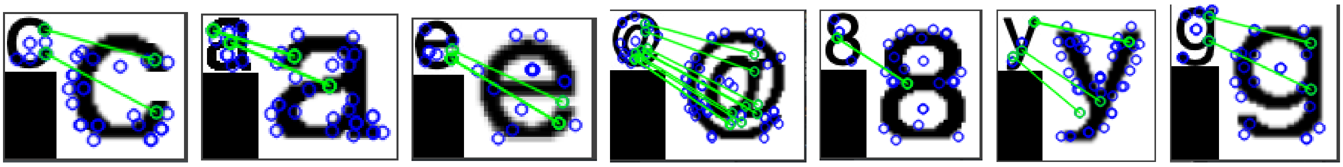


Figure. 3 Examples of Correct Predictions on test_img.jpg and evaluation data

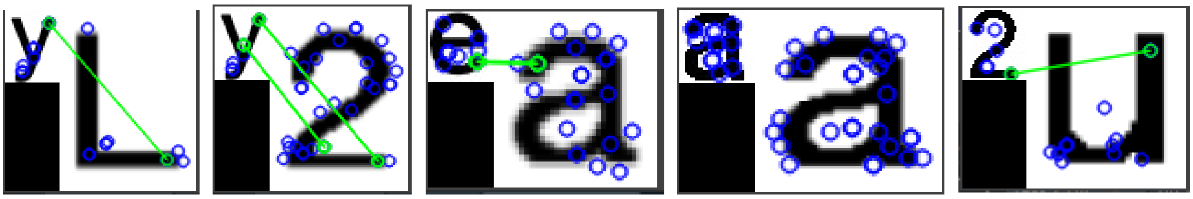


Figure. 4 Examples of Incorrect Predictions on test_img.jpg and evaluation data