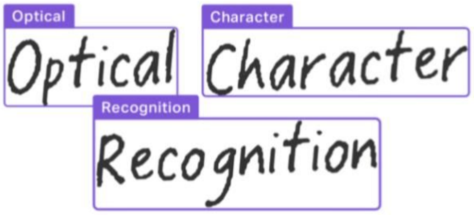
University at Buffalo
Department of Computer Science and and Engineering
CSE 473/573 - Computer Vision and Image Processing

Project #1
Optical Character Recognition

Naman Pundir
50373843
namanpun@buffalo.edu
Date: 03/09/21

OCR Stands for "Optical Character Recognition." OCR is a technology that recognizes text within a digital image. It is commonly used to recognize text in scanned documents, but it serves many other purposes as well.

OCR processes a digital image by locating and recognizing characters, such as letters, numbers, and symbols. Some OCR software will simply export the text, while other programs can convert the characters to editable text directly in the image. Advanced OCR software can export the size and formatting of the text as well as the layout of the text found on a page.

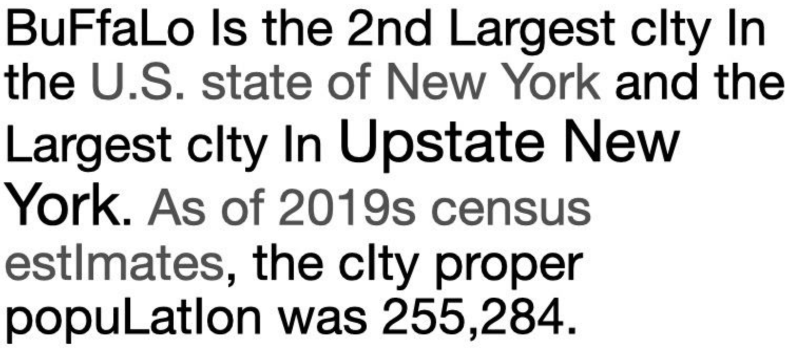


The OCR system will contain three parts, **Enrollment, Detection and Recognition**.

Dataset: We are provided with a set of characters to be recognized from the main image.

Characters: 1. 2 2. e 3. c 4. a 5. ▪

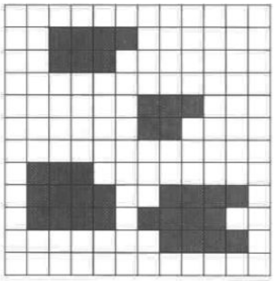
Main Image:



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

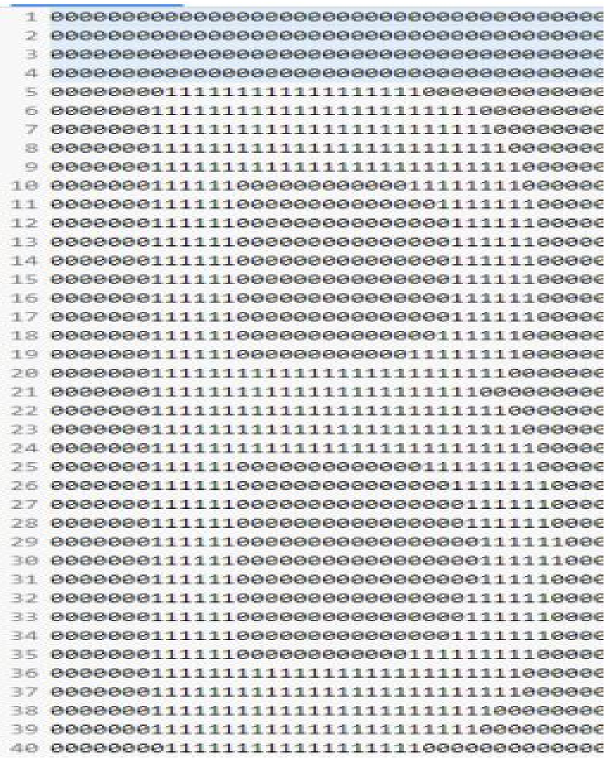
1. **Enrollment:**
   In the task-1 of enrollment we are using Scale-invariant feature transform (SIFT) to extract features from the character images.

2. **Detection:**
   We have used Connected Component Labeling for the detection of the characters in the image.
   Connected Component Labeling: A component labeling algorithm finds all connected components in an image and assigns a unique label to all points in the same component.



		1	1	1	1						
		1	1	1							
						2	2	2			
						2	2				
	3	3	3								
	3	3	3	3			4	4	4	4	
	3	3	3	3		4	4	4	4		
							4	4	4	4	

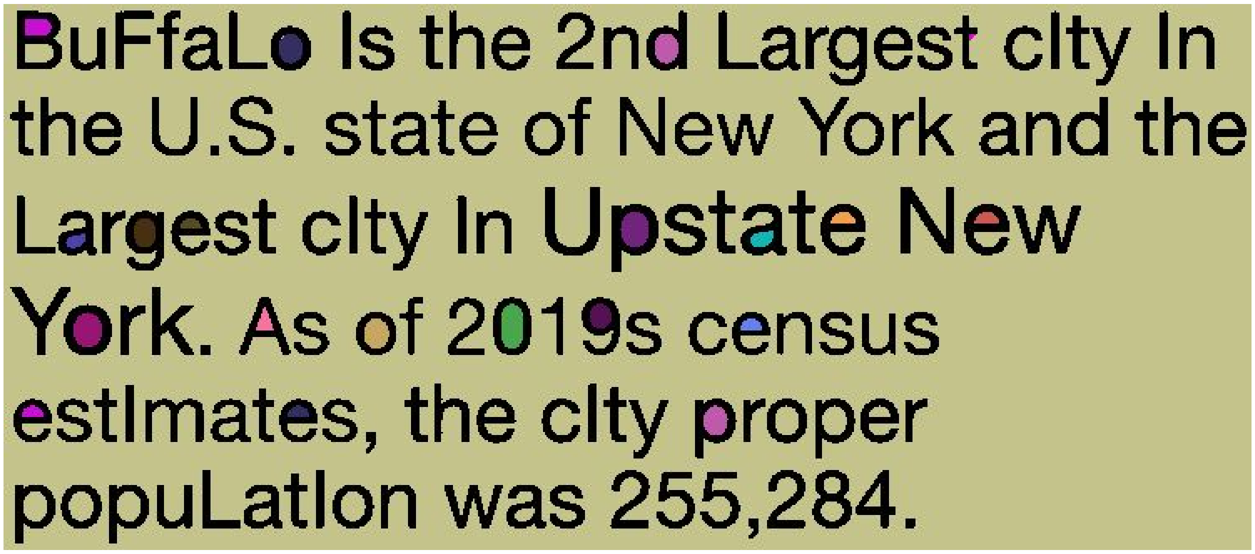
Image after CCL:



```
000000000000000000000000000000000000000
000000000000000000000000000000000000000
000000000000000000000000000000000000000
000000000000000000000000000000000000000
000000001111111111111111110000000000000
000000011111111111111111111111000000000
000000011111111111111111111111100000000
000000011111111111111111111111110000000
000000011111111111111111111111111000000
000000011111100000000000011111111000000
000000011111100000000000000111111000000
000000011111100000000000000011111000000
000000011111100000000000000011111000000
000000011111100000000000000011111000000
000000011111100000000000000011111000000
000000011111100000000000000011111000000
000000011111100000000000000011111000000
000000011111100000000000000111110000000
000000011111100000000000011111110000000
000000011111111111111111111111110000000
000000011111111111111111111111000000000
000000011111111111111111111111110000000
000000011111111111111111111111110000000
000000011111111111111111111111111000000
000000011111100000000000001111111000000
000000011111100000000000000011111100000
000000011111100000000000000011111000000
000000011111100000000000000001111100000
000000011111100000000000000000111110000
000000011111100000000000000000111110000
000000011111100000000000000000111100000
000000011111100000000000000001111100000
000000011111100000000000000001111100000
000000011111100000000000000011111100000
000000011111100000000000011111111000000
000000011111111111111111111111111100000
000000011111111111111111111111111100000
000000011111111111111111111111110000000
000000011111111111111111111111100000000
000000001111111111111111110000000000000
```

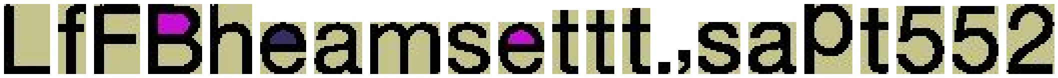
First character

Main Image after CCL:



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 estImates, the clty proper popuLatIon was 255,284.

3. **Recognition:**
   For the recognition part first we have made the bounding box in the CCL image shown above.

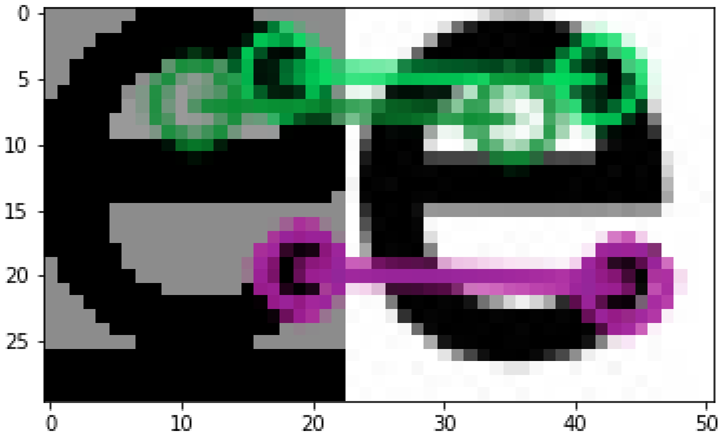


With the help of a bounding box we can determine the position of each character in the image.

```
"bbox": [540, 309, 6, 6], "name": "dot"}, {"bbox": [446, 309, 6, 12],
"name": "dot"}
```

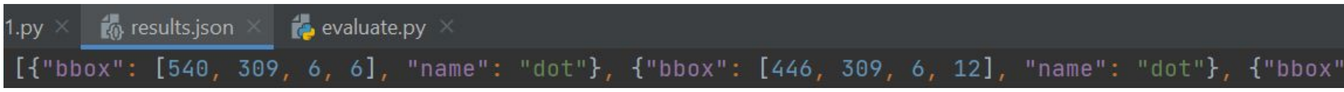
Where: "bbox": [x,y,h,w]

Then we have performed template matching with the enrollment image and each character in the bounding box. If they matched we put the "name" as the character detected and if they don't match we put the "name" as "UNKNOWN".



Result.json:

All the matching and position of the bounding box is then dump into json file "Result.json".

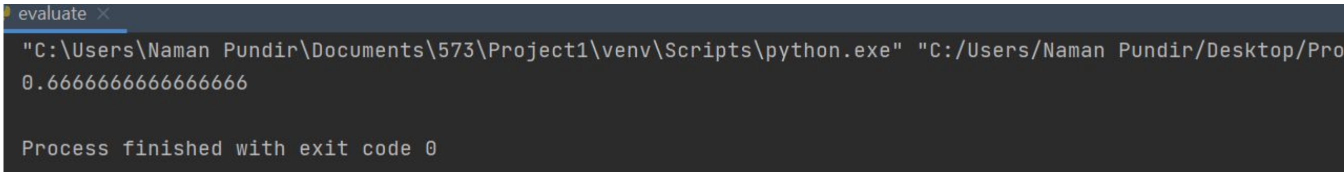


```
[{"bbox": [540, 309, 6, 6], "name": "dot"}, {"bbox": [446, 309, 6, 12], "name": "dot"}, {"bbox"
```

Evaluate.py:

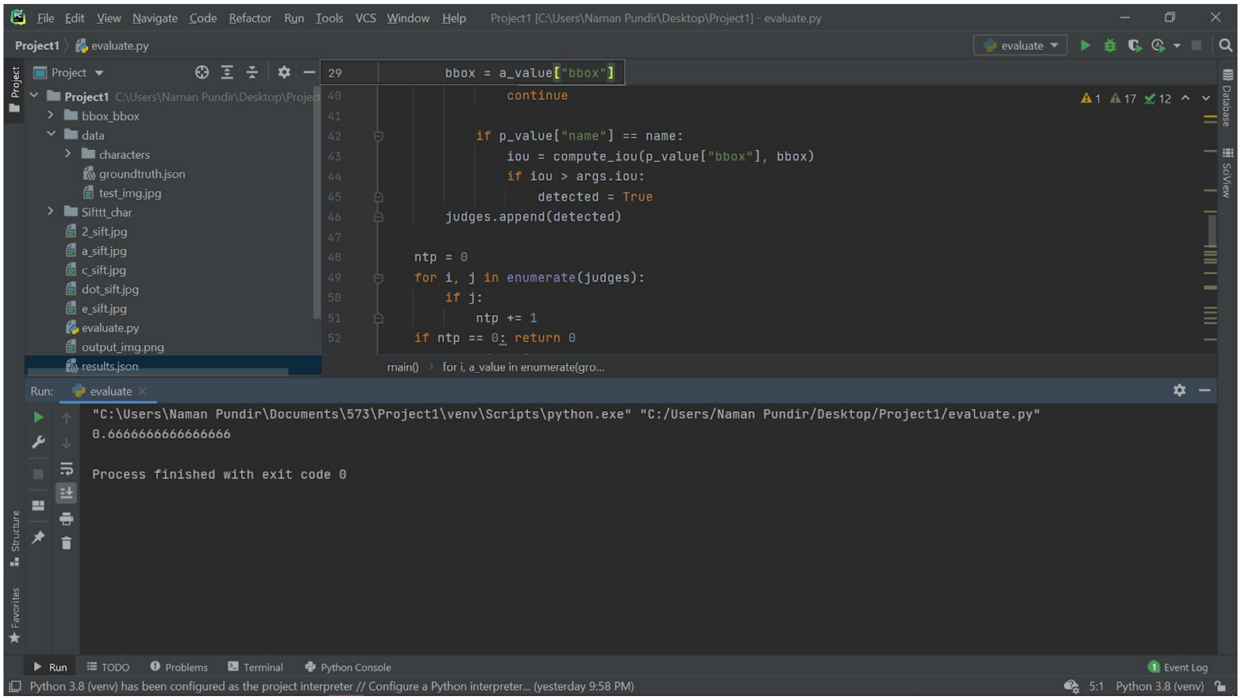
Parsing result.json file intro evaluate.py to check the F1 measure.

F1 Measure = 0.6666666



```
"C:\Users\Naman Pundir\Documents\573\Project1\venv\Scripts\python.exe" "C:/Users/Naman Pundir/Desktop/Pro
0.6666666666666666

Process finished with exit code 0
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Project1 [C:\Users\Naman Pundir\Desktop\Project1] - evaluate.py
Project1 › evaluate.py
evaluate
Project
Project1 C:\Users\Naman Pundir\Desktop\Project
bbox_bbox
data
characters
groundtruth.json
test_img.jpg
Sifttt_char
2_sift.jpg
a_sift.jpg
c_sift.jpg
dot_sift.jpg
e_sift.jpg
evaluate.py
output_img.png
results.json
main() › for i, a_value in enumerate(gro...
Run: evaluate
Run TODO Problems Terminal Python Console Event Log
Python 3.8 (venv) has been configured as the project interpreter // Configure a Python interpreter... (yesterday 9:58 PM)
5:1 Python 3.8 (venv)

```
        bbox = a_value["bbox"]
                continue

            if p_value["name"] == name:
                iou = compute_iou(p_value["bbox"], bbox)
                if iou > args.iou:
                    detected = True
        judges.append(detected)

    ntp = 0
    for i, j in enumerate(judges):
        if j:
            ntp += 1
    if ntp == 0: return 0
```

```
"C:\Users\Naman Pundir\Documents\573\Project1\venv\Scripts\python.exe" "C:/Users/Naman Pundir/Desktop/Project1/evaluate.py"
0.6666666666666666

Process finished with exit code 0
```

xxxxxxxxxxxxxxxxxxxxxxxxx