Jeremy Midvidy, jam658
EECS 332, Fall 2018
MP#1 Report

About My Code:

My code for MP1 can be found in this folder, in the python file, `MP1.py`. The file is callable from the command line with the command `$ MP1.py PATH_TO_IMAGE`.

After running, it will print the number of labels found to the command line. Additionally, it will write a result image named "`PATH_TO_IMAGE_result` " of the classified labels to the folder named "results." When called from the command line, MP1.py will print relevant information like the number of classification labels, the location of the saved image file, the equivalences assigned on the first pass, the union-find of the equivalences, and the colors assigned to each label.

I wrote my code in Python 3.7, and it assumes python packages `numpy`, `imageio`, `collections`, `matplotlib`, and `sys` are installed.

Results:

I ran my code on the three given image files, test.bmp, gun.bmp, and face.bmp. In short, my code performed correctly on all three test cases. First, my code uses the `imageio.imread()` function to translate the .bmp image into a 2-D matrix of RGB values. Then, I implemented the sequential labeling algorithm to go through and label row-major neighbors.

On the first pass, for each non-background cell, the algorithm would check to see if the cell above was labeled and if the cell to the left was labeled. If cell-up and cell-left were labeled, the algorithm labels the current cell min(up,left) and adds an equivalence [up,left]. If only cell-left was labeled, it assigned the current cell the label from cell-left. Likewise, if only cell-up was labeled, it assigned the current cell the label from cell-up. If neither cell-up or cell-left were labeled, it assigned the next smallest label from the pool of unused labels, and continued to the next cell.

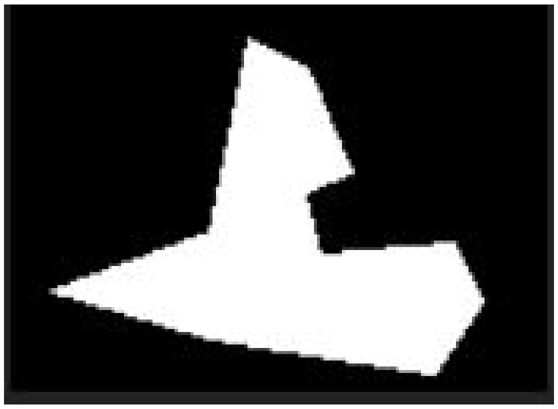
On the second pass, the algorithm resolved the equivalences and merged all neighboring cells into the same number. Resolving the equivalences was more complicated and nuanced than I had anticipated, but ultimately in involved both forward chaining and backward chaining by implementing a data-structure called union-find. In short, for each possible label, the second pass found the smallest possible label that could be chained to get to the current label. Then it mapped all labels with the same smallest chained label to the smallest chain label. Then it iterated through the 2-D matrix and reassigned the correct label to each cell.

Here are the results from my code. It is kind of hard to see in images of this size, but if you go into the "results" folder contained within this submission, and open the results in of each test (every unique label is represented by its own color in the result image). You will see the correct labels. Likewise, if you run the python script from the command line with the test files as inputs (or other appropriate .bmp inputs), it will return the correct files with labels to the results folder.

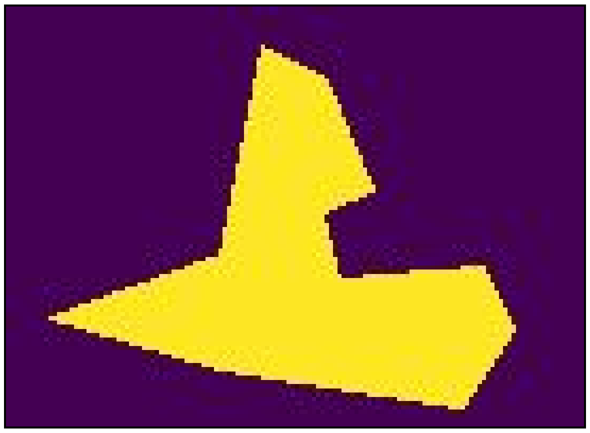
Results Analysis for test.bmp, gun.bmp, face.bmp

---

test.bmp:



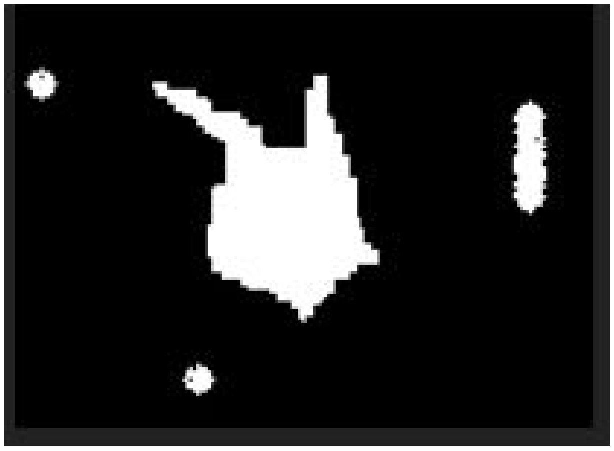
→



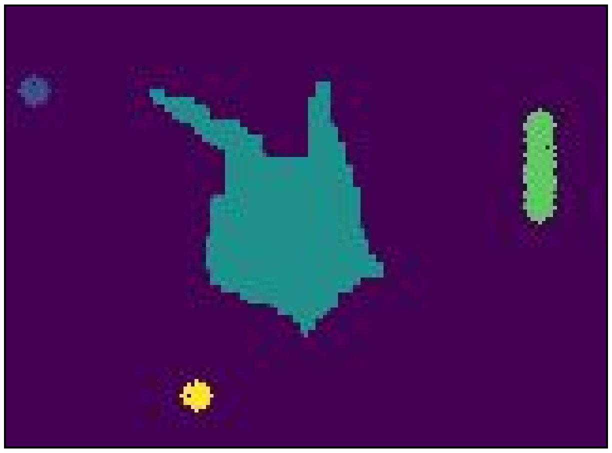
Number of labels assigned on first pass:	29
Equivalences on first pass	[[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11], [11, 12], [12, 13], [13, 14], [14, 16], [15, 17], [16, 18], [17, 19], [18, 20], [1, 19], [20, 21], [21, 22], [22, 23], [23, 24], [24, 25], [25, 26], [26, 27], [27, 28], [28, 29]]
Union-find of equivalences	{1: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29}}
Label colors	{1: [255]}
Number of total labels	1

---

gun.bmp:



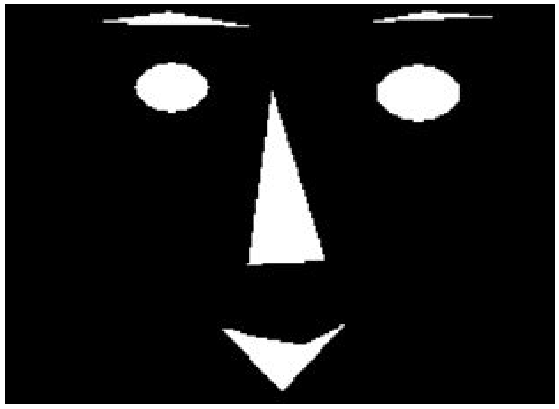
→



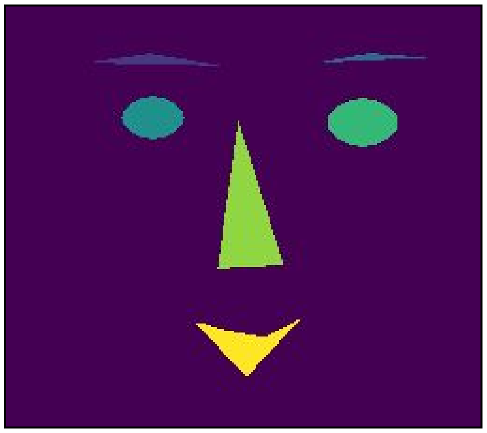
Number of labels assigned on first pass:	23
Equivalences on first pass	[[1, 2], [2, 3], [3, 5], [4, 7], [8, 9], [9, 10], [10, 11], [10, 12], [10, 13], [6, 7], [4, 6], [10, 14], [10, 15], [6, 16], [16, 17], [10, 18], [17, 19], [21, 22], [20, 21], [22, 23]]
Union-find of equivalences	{1: {1, 2, 3, 5}, 2: {4, 6, 7, 16, 17, 19}, 3: {8, 9, 10, 11, 12, 13, 14, 15, 18}, 4: {20, 21, 22, 23}}
Label colors	{1: [63], 2: [126], 3: [189], 4: [252]}
Number of total labels	4

---

face.bmp:



→



Number of labels assigned on first pass:	52
Equivalences on first pass	[[1, 3], [2, 4], [3, 5], [4, 6], [5, 7], [6, 8], [7, 9], [8, 10], [11, 12], [12, 14], [13, 15], [14, 16], [15, 17], [16, 18], [17, 19], [18, 20], [19, 21], [20, 22], [21, 23], [22, 24], [23, 25], [24, 26], [25, 27], [27, 28], [28, 29], [26, 30], [31, 32], [32, 33], [33, 34], [34, 35], [35, 36], [36, 37], [37, 38], [38, 39], [39, 40], [40, 41], [41, 42], [43, 44], [44, 46], [46, 47], [47, 48], [48, 49], [49, 50], [50, 51], [51, 52], [45, 52], [45, 51], [45, 50], [45, 49]]
Union-find of equivalences	{1: {1, 3, 5, 7, 9}, 2: {2, 4, 6, 8, 10}, 3: {11, 12, 14, 16, 18, 20, 22, 24, 26, 30}, 4: {13, 15, 17, 19, 21, 23, 25, 27, 28, 29}, 5: {32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 31}, 6: {43, 44, 45, 46, 47, 48, 49, 50, 51, 52}}
Label colors	{1: [42], 2: [84], 3: [126], 4: [168], 5: [210], 6: [252]}
Number of total labels	6