MP#1 is implementing the sequential algorithm for connect component labeling

All images are binary images. In the first scanning:

Scan the image from left to right, top to bottom. If the result is 1, meaning the pixel needs to be labeled. There is total four cases:

L_u: label of the upper pixel

L_l: label of the left pixel

L(u,v): label at position (u,v)

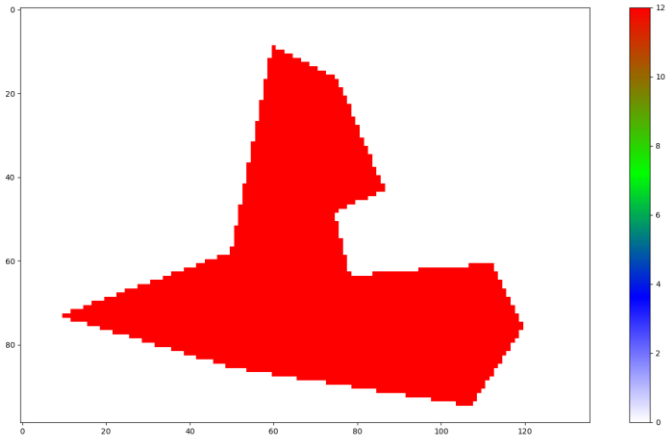
<table>
<tr><td></td><td>L_u = 0</td><td>L_u != 0</td></tr>
<tr><td>L_l = 0</td><td>L(u,v) = L+1</td><td>L(u,v) = max(L_u, L_l)</td></tr>
<tr><td rowspan="2">L_l != 0</td><td rowspan="2">L(u,v) = max(L_u, L_l)</td><td>L_u = L_l</td></tr>
<tr><td>L_u != L_l (E_table)</td></tr>
</table>

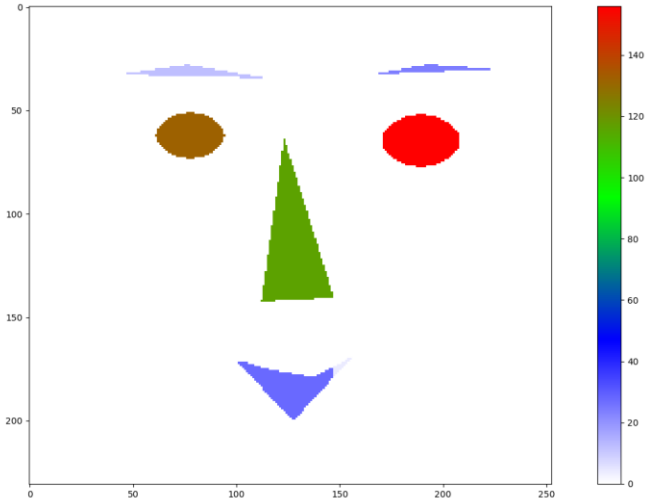
If the labels of the upper pixel and left pixel are both zero, current pixel needs a new label. Thus, L(u,v) = L+1. If one label of the upper pixel or left pixel is not zero, the current pixel does not need a new label. Thus, L(u,v) = max(L_u, L_l). If the labels of the upper pixel and left pixel are both zero means, the current pixel does not need a new label. For this case, there are two possibilities. First, if L_u = L_l, L(u,v) = L_u or L_l. Second, if L_u != L_l, L(u,v)=min(L_u, L_). Also, L_u and L_l should be the same label, so a E_table will be used. I use Dictionary to build the E_table.
In the second run, I will renumbering the labels by using the E_table I built.
For gun.bmp, there is some noise in the picture. To avoid that, I applied a size filter. I count the total number of each label. If the number is smaller than 500, this label will be ignored (set to 0).

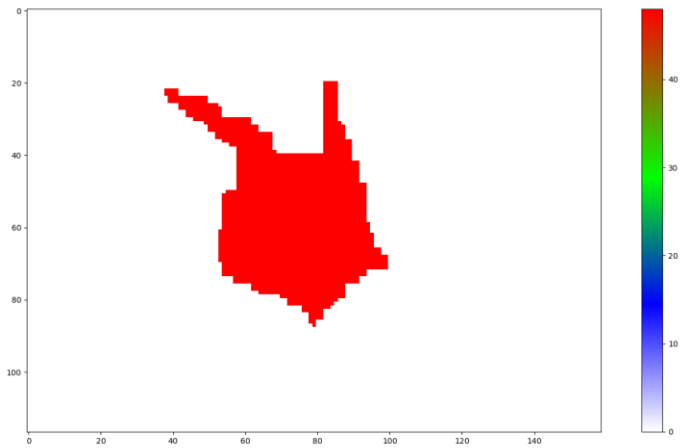
Results Analysis:

All three test cases look good. For picture 2, different parts of the face will be shown in different colors means the labels for each part are different. For picture 3, noise will be ignored successfully.

Results:


0
20
40
60
80
0
20
40
60
80
100
120
12
10
8
6
4
2
0


0
50
100
150
200
0
50
100
150
200
250
140
120
100
80
60
40
20
0


0
20
40
60
80
100
0
20
40
60
80
100
120
140
40
30
20
10
0