ECE 6310- INTRODUCTION TO COMPUTER VISION

LAB 2: OPTICAL CHARACTER RECOGNITION

Submitted To:

Dr. Adam Hoover

Ashit Mohanty

C13582787

amohant@g.clemson.edu

Date of submission: 09-15-2020

Mean Spatial Filter Image and Zero mean template image

The main agenda of this lab assignment was to implement a matched spatial filter to recognize the letter e in an image of text. The program was written using the language C. The user must input only threshold limits, lower and upper threshold respectively, between which the program runs to generate the total false positives and the total true positives for every threshold.

The first step was to generate a Mean Spatial Filter image of the original image. To perform this objective, first the zero mean template had to be generated of the original template image. This was generated by subtracting the average of all pixels in the template image, parenthood_e_template.ppm, from every individual pixel in the template image. This template was then used to perform convolution on the original input image, parenthood.ppm. The image that is generated is not an 8-bit image. This image is then normalized to 8-bits using the formula,

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin$$

Where,

I = Image pixel after convolution, but not normalized

Min = Minimum pixel value in the MSF image, but not normalized

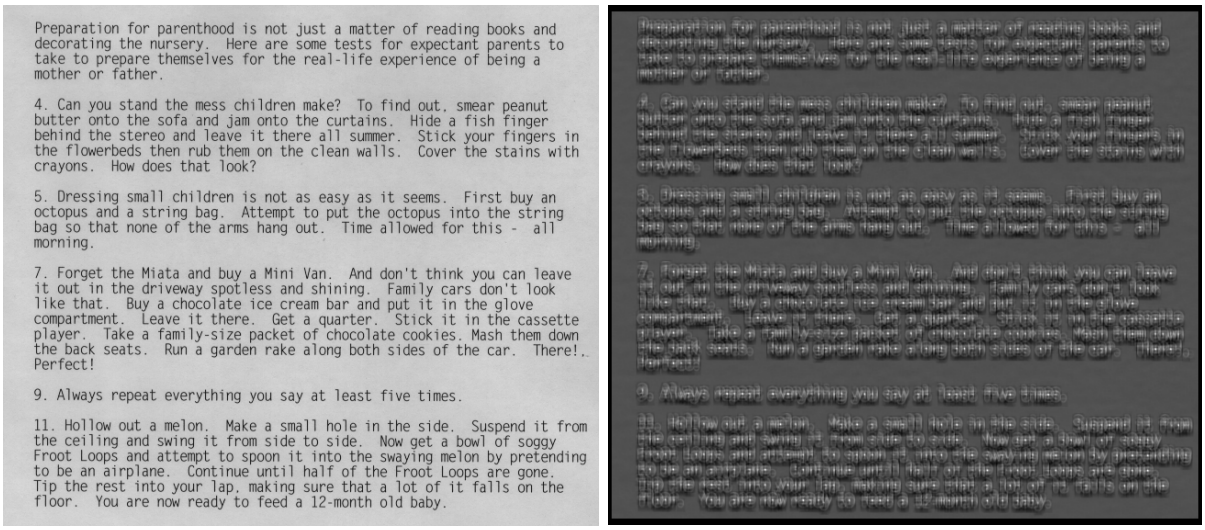
Max = Maximum pixel value in the MSF image, but not normalized

$newMin$ = New Minimum pixel value in the normalized MSF image, in this case 0

$newMax$ = New Maximum pixel value in the normalized MSF image, in this case 255

I_N = Image pixel after convolution and after being normalized

The original image and the MSF image after being normalized are shown underneath.



Preparation for parenthood is not just a matter of reading books and decorating the nursery. Here are some tests for expectant parents to take to prepare themselves for the real-life experience of being a mother or father.

4. Can you stand the mess children make? To find out, smear peanut butter onto the sofa and jam onto the curtains. Hide a fish finger behind the stereo and leave it there all summer. Stick your fingers in the flowerbeds then rub them on the clean walls. Cover the stains with crayons. How does that look?

5. Dressing small children is not as easy as it seems. First buy an octopus and a string bag. Attempt to put the octopus into the string bag so that none of the arms hang out. Time allowed for this - all morning.

7. Forget the Miata and buy a Mini Van. And don't think you can leave it out in the driveway spotless and shining. Family cars don't look like that. Buy a chocolate ice cream bar and put it in the glove compartment. Leave it there. Get a quarter. Stick it in the cassette player. Take a family-size packet of chocolate cookies. Mash them down the back seats. Run a garden rake along both sides of the car. There!. Perfect!

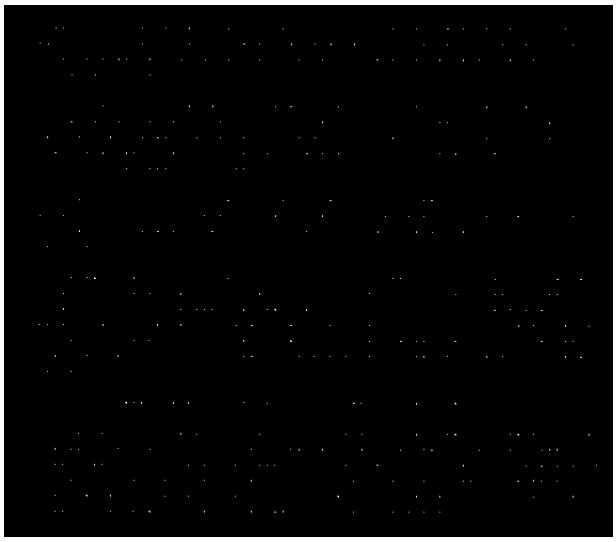
9. Always repeat everything you say at least five times.

11. Hollow out a melon. Make a small hole in the side. Suspend it from the ceiling and swing it from side to side. Now get a bowl of soggy Froot Loops and attempt to spoon it into the swaying melon by pretending to be an airplane. Continue until half of the Froot Loops are gone. Tip the rest into your lap, making sure that a lot of it falls on the floor. You are now ready to feed a 12-month old baby.

The image on the left is the original image, and as can be seen, is very clear and easy to read. The image on the right is that of the MSF normalized image.

Detecting letter ‘e’

The next part of the lab assignment was to detect the letter e from the MSF normalized image. This was done by iterating a set of operations through the entire range of thresholds, that would be entered by the user. In this case, I entered the threshold range of 175-235. For every threshold value, every pixel of the normalized MSF image would be checked if its value is greater than the threshold value or not. If the value of the pixel turned out to be greater than threshold, the value 255 would be assigned to another new image file at that same position. If it were smaller, the value of 0 would be assigned. This way the entire binary image would have only black pixels (value 0) and white pixels (value 255), thereby creating a binary image. An example of the binary image is attached underneath,



This binary image is of when the threshold was 195. So all the pixels whose value were lesser than this threshold value of 195, were given a value of 0, thereby the black pixels, and the pixels whose value were greater than 195, were given a value of 255, thereby the few white pixels.

These white pixels now needed to be analyzed using the ground truth locations from the given file, parenthood_gt.txt. The ground truth locations were read from the file, and at every ground truth location in the binary image, every pixel in a 9x15 window were scanned, centered at the ground truth location. If any pixel in the 9x15 window, were a white one, i.e. a value of 255, then the value of a variable “detected” would be changed to 1. This means that the letter e has been detected. If the letter scanned from the ground truth file was also an e, this would mean that it is a true positive. If the letter in the ground truth file was not an e, this would mean that the detected letter is a false positive. If the no pixel in the 9x15, were a white one, i.e. all black pixels, means that no letter e is in the area. However, if from the ground truth file, the letter scanned were an e, this means that this is a false negative. And if the letter in the ground truth location were not an e, verified from the ground truth file, this would indicate a true negative. For every threshold, the count of the true positives and the false positives were printed. The output for every run of the program is shown in the image as below.

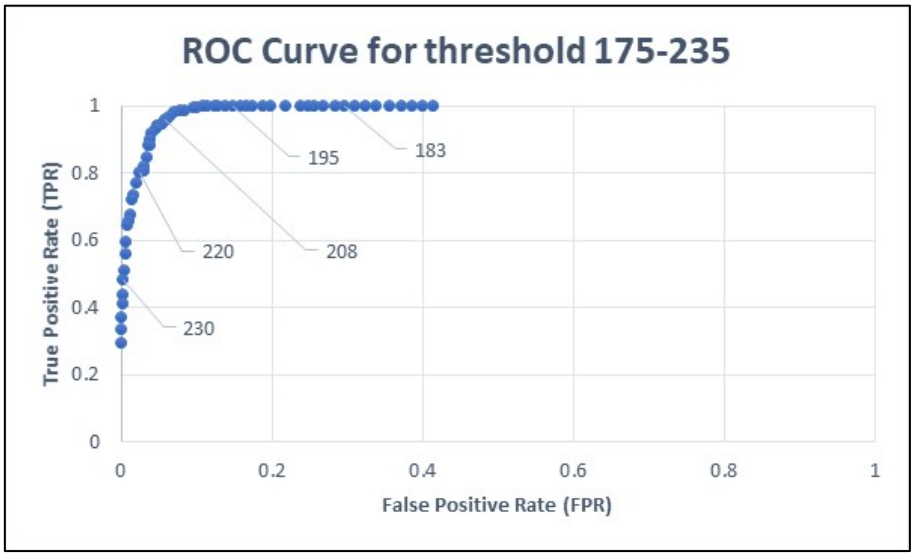
```
kimir@LAPTOP-G6LMKVC2 MINGW64 /d/Clemson/Clemson_Fall_2020/ECE6310_Computer_Visi
on/Lab_projects/Lab_2
$ ./lab2 175 235
Threshold:175 TP:151 FP:471
Threshold:176 TP:151 FP:461
Threshold:177 TP:151 FP:445
Threshold:178 TP:151 FP:429
Threshold:179 TP:151 FP:413
Threshold:180 TP:151 FP:395
Threshold:181 TP:151 FP:377
Threshold:182 TP:151 FP:361
Threshold:183 TP:151 FP:344
Threshold:184 TP:151 FP:330
Threshold:185 TP:151 FP:316
Threshold:186 TP:151 FP:298
Threshold:187 TP:151 FP:286
Threshold:188 TP:151 FP:277
Threshold:189 TP:151 FP:264
Threshold:190 TP:151 FP:242
Threshold:191 TP:151 FP:221
Threshold:192 TP:151 FP:210
Threshold:193 TP:151 FP:193
Threshold:194 TP:151 FP:185
Threshold:195 TP:151 FP:176
Threshold:196 TP:151 FP:165
Threshold:197 TP:151 FP:154
Threshold:198 TP:151 FP:143
Threshold:199 TP:151 FP:138
Threshold:200 TP:151 FP:128
Threshold:201 TP:151 FP:121
Threshold:202 TP:150 FP:112
Threshold:203 TP:150 FP:107
Threshold:204 TP:149 FP:95
Threshold:205 TP:149 FP:87
Threshold:206 TP:148 FP:78
Threshold:207 TP:146 FP:71
Threshold:208 TP:145 FP:68
Threshold:209 TP:145 FP:65
Threshold:210 TP:143 FP:62
Threshold:211 TP:142 FP:54
Threshold:212 TP:142 FP:54
Threshold:213 TP:141 FP:51
Threshold:214 TP:139 FP:45
Threshold:215 TP:136 FP:43
Threshold:216 TP:133 FP:43
Threshold:217 TP:133 FP:40
Threshold:218 TP:128 FP:38
Threshold:219 TP:124 FP:35
Threshold:220 TP:122 FP:34
Threshold:221 TP:121 FP:28
Threshold:222 TP:116 FP:23
Threshold:223 TP:111 FP:19
Threshold:224 TP:109 FP:16
Threshold:225 TP:102 FP:15
Threshold:226 TP:99 FP:12
Threshold:227 TP:97 FP:9
Threshold:228 TP:90 FP:7
Threshold:229 TP:84 FP:7
Threshold:230 TP:77 FP:5
Threshold:231 TP:73 FP:4
Threshold:232 TP:66 FP:4
Threshold:233 TP:62 FP:3
Threshold:234 TP:56 FP:2
Threshold:235 TP:50 FP:1

kimir@LAPTOP-G6LMKVC2 MINGW64 /d/Clemson/Clemson_Fall_2020/ECE6310_Computer_Visi
on/Lab_projects/Lab_2
$
```

ROC Curve

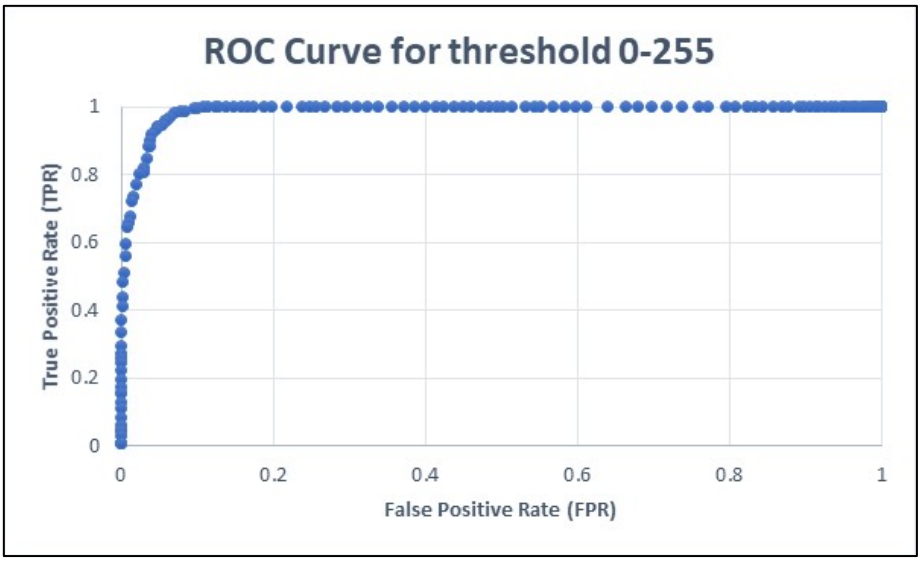
The values of true positives (TP), true negatives (TN), false positives(FP), and false negatives(FN) that were generated for the entire range of thresholds, as entered by the user, were written into a file output_lab2.txt, along with the corresponding threshold value. This would be used to generate the ROC curve and the optimal threshold value.

The ROC or receiver operating characteristic curve is plot between the True Positive Rate and the False Positive Rate. These rates are calculated by the formulae,

$$TPR = \frac{TP}{TP + FN} \quad FPR = \frac{FP}{FP + TN}$$



The ROC curve is plot between the TPR and FPR for the range of thresholds, 175 to 235. The data labels in the graph show the value of thresholds at the data points. As can be seen from the ROC curve when the threshold is lower the number of False Positives rise. This is because at a lower threshold all the letters are being detected and the system says all are the letter e. When the threshold is very high, the number of true positives also decrease because the system can detect very few letters.



This can be verified also when the thresholds are taken for the entire range of the grayscale pixel value, as shown in the image above.

The optimal value would be a tradeoff between the two. The optimal value of threshold would be the knee of the curve. The knee of the curve is the value of the curve that is closest to the desired value. The best-case scenario in this problem statement would be to have detected all the letters e and have detected no other letters. This means that TPR should be 1 and FPR should be 0. So, the knee of the curve would be the point that is closest to the ideal point of TPR=1 and FPR=0. This value is found by computing the Euclidean distance between the plot points to the ideal point. This value of threshold was found to be 208 with a TPR value of 0.9602 and an FPR value of 0.0612. The corresponding values at this threshold were, TP=145 and FP=65.

Therefore,

Optimal Threshold = 208

TP = 145

FP = 68