Brandon Shumin
ECE 6310
Lab 3

Reading Input Data, Template Image, Ground Truth File and Lab2 Normalized MSF Files:

For this step, most code from the read-in part of Lab 2 was recycled, with the inclusion of the Normalized MSF image, called msf_e.ppm, also being read in. The template image was used to get the dimensions for the 9x15 area to be used checking ground truth locations dynamically as opposed to hard-coding, and the MSF image was used to save time and skip generating this image again as it would be identical to the MSF image from Lab 2.

Copy Image in 9x15 Area Around Ground Truth and Threshold:

At each ground truth location, a 9x15 area around the pixel in question is copied to a separate array from the original image. Then, this 9x15 image is ran through a threshold of 128, meaning all pixels with values above 128 are set to 255 while all other pixels are set to 0. By performing this thresholding, a binary image of all values 255 and 0 is generated to be used in future steps for thinning. An example of the resulting image is shown in Figure 3.

Thin the Binary 9x15 Image:

The 9x15 binary image centered at a ground truth location is then thinned until thinning no longer removes pixel on the image. Thinning is accomplished by checking that three constraints are met for a every detected, or value of 0, pixel in the image. All pixels are first checked for a detection, and if they are detected, then a 3x3 area around them is copied, with out-of-bounds cases being counted as non-edges, and checked for their number of edge neighbors, number of clockwise edge-to-non-edge transitions, and if the North or East or South and West pixels relative to the center of the 3x3 image are not edges. If all three of these conditions are satisfied, then the pixel in the center of the 3x3 is marked for removal after all detected pixels in the 9x15 area are checked. This process is then repeated until no more pixels are removed.

Find Branchpoints and Endpoints:

As the image is being thinned, I determine how many branchpoints and endpoints are in the image for use after thinning is completed on the final run, and resetting the count of the points if more thinning is required. The amount of branchpoints and endpoints is used to help determine if the image detected is correctly identified as an ‘e’ or not, since an ‘e’ should have one endpoint and one branchpoint, shown below as identified by my program in Figure 6. These branchpoints are then used in the below section to determine if the predictions for the letter are True Positives, True Negatives, False Positives, or False Negatives.

Brandon Shumin
ECE 6310
Lab 3

Compare Detections with Ground Truth:

The final step for the program is to check which detections generated in the previous step constitute a True Positive (TP), False Positive (FP), True Negative (TN), or False Negative (FN). To do this, a different approach from Lab 2 must be taken. To get TP and FP, you check if the letter was counted as detected using the MSF and if it has exactly one branchpoint and endpoint. If it meets these criteria, then it is counted as a detected 'e', otherwise it was not detected. Then this value is compared against the Ground Truth value at that point to determine if it was a TP or FP. If the image was not detected using the MSF then if the Ground Truth stated it should be an 'e' it is a FN, otherwise it is a TN.

ROC Curve:

Using the outputs generated in Figure 1 below, a table of data, shown in Figure 5, was used to generate the ROC Curve shown in Figure 6. This curve shows that for my program and given the thresholds that I checked, the most appropriate T, or threshold values, would be 210, as in lab 2, which would mean the TP count would be 136 and the FP count would be 60. The optimal threshold is 210 because it is the closest threshold to having a TPR of 1 and FPR of 0, which would be a perfect model.

Problems with This Method:

One edge case that makes detecting all 151 TP cases using this method possible is that if the image around the Ground Truth location clips another letter adjacent to it, extra endpoints or branchpoints could be calculated by accident. An example of this can be seen in Figure 8. This problem was brought up to Dr. Hoover in class and he stated there was no way to fix this problem without using more advanced methods such as segmentation. Therefore, having my total TP max out at 144 even at low thresholds makes sense, as this most likely means the other 7 'e' characters are being affected by this clipping.

Brandon Shumin
ECE 6310
Lab 3

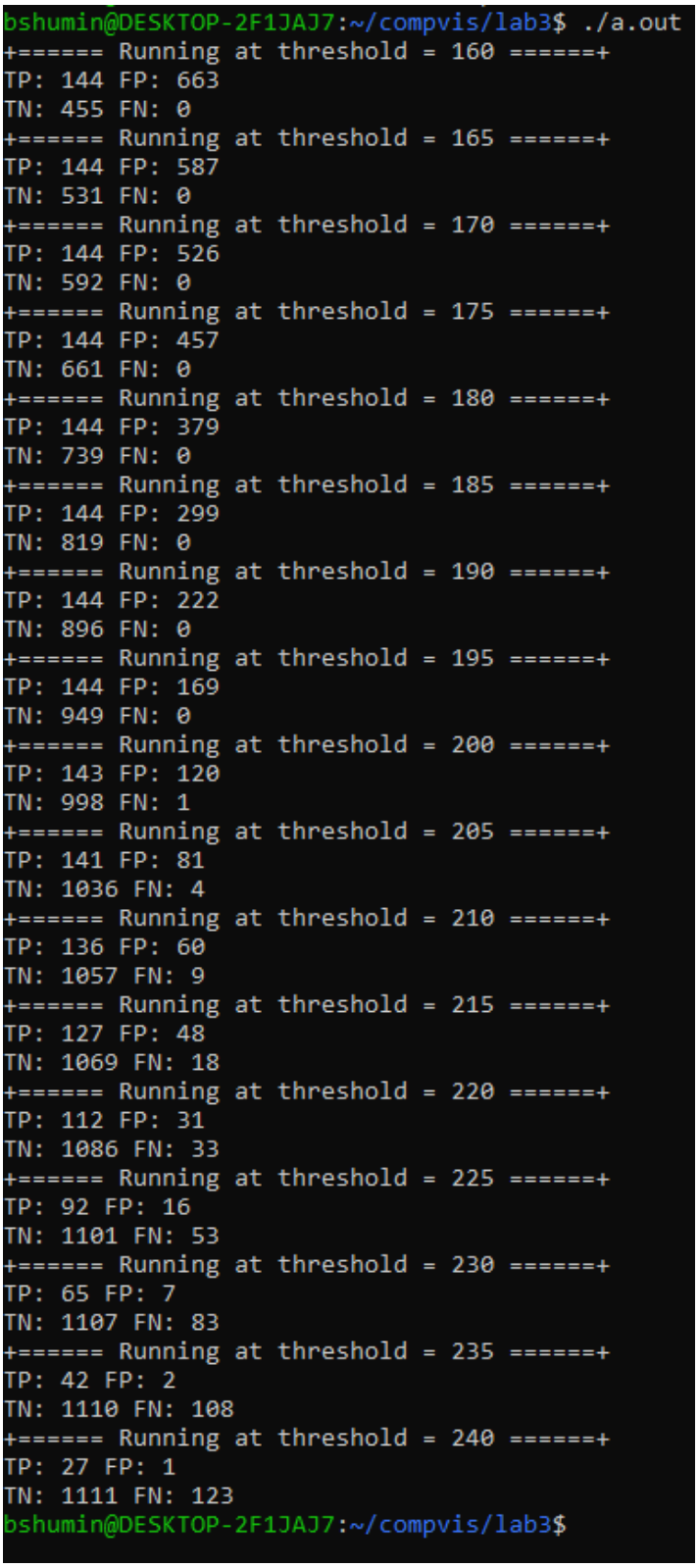

Figure 1: Example of running this code and outputs

Brandon Shumin
ECE 6310
Lab 3

Figure 2: Normalized MSF image from Lab 2

Brandon Shumin
ECE 6310
Lab 3

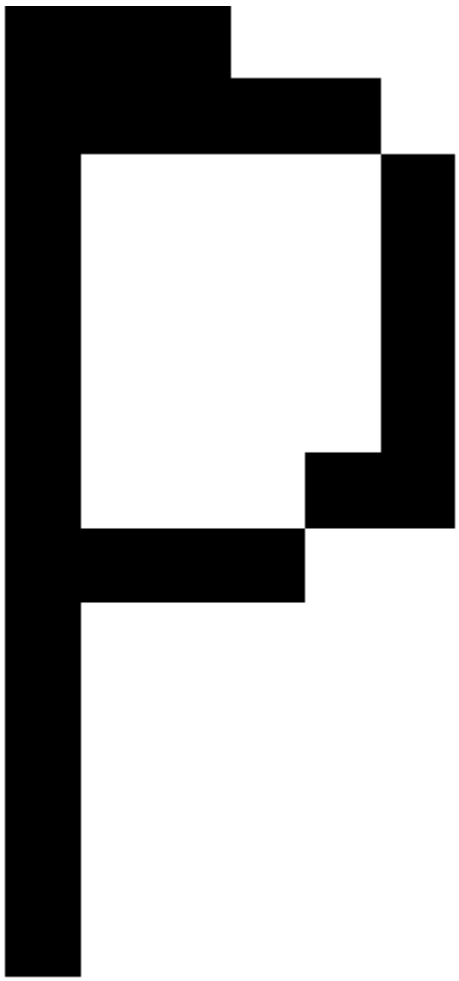

Figure 3: Example of binary 9x15 image generated by thresholding Ground Truth area at 128

Brandon Shumin
ECE 6310
Lab 3

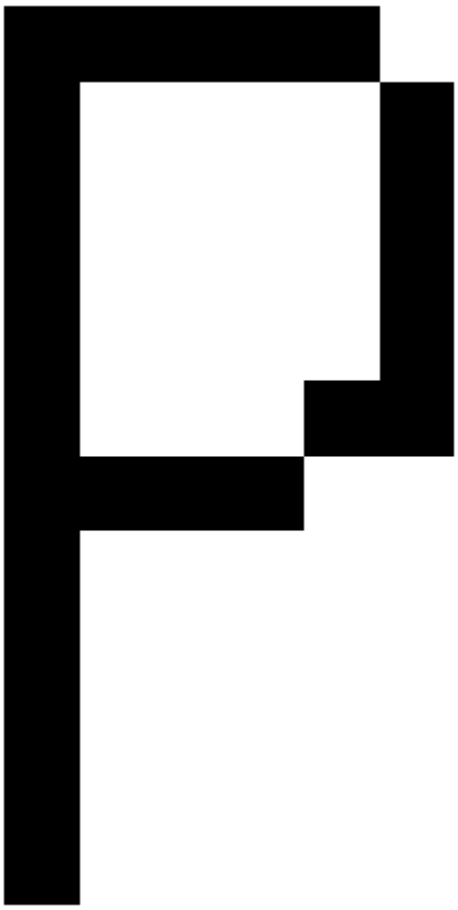

Figure 4: Example of thinned image after Figure 3 is thinned in my code

Brandon Shumin
ECE 6310
Lab 3

Threshold	TP	FP	TN	FN		FPR	TPR
160	144	663	455	0		0.593023	1
165	144	587	531	0		0.525045	1
170	144	526	592	0		0.470483	1
175	144	457	661	0		0.408766	1
180	144	379	739	0		0.338998	1
185	144	299	819	0		0.267442	1
190	144	222	896	0		0.198569	1
195	144	169	949	0		0.151163	1
200	143	120	998	1		0.107335	0.993056
205	141	81	1036	4		0.072516	0.972414
210	136	60	1057	9		0.053715	0.937931
215	127	48	1069	18		0.042972	0.875862
220	112	31	1086	33		0.027753	0.772414
225	92	16	1101	53		0.014324	0.634483
230	65	7	1107	83		0.006284	0.439189
235	42	2	1110	108		0.001799	0.28
240	27	1	1111	123		0.000899	0.18

Figure 5: Table of TP, FP, TN, FN, TPR, and FPR

Brandon Shumin
ECE 6310
Lab 3

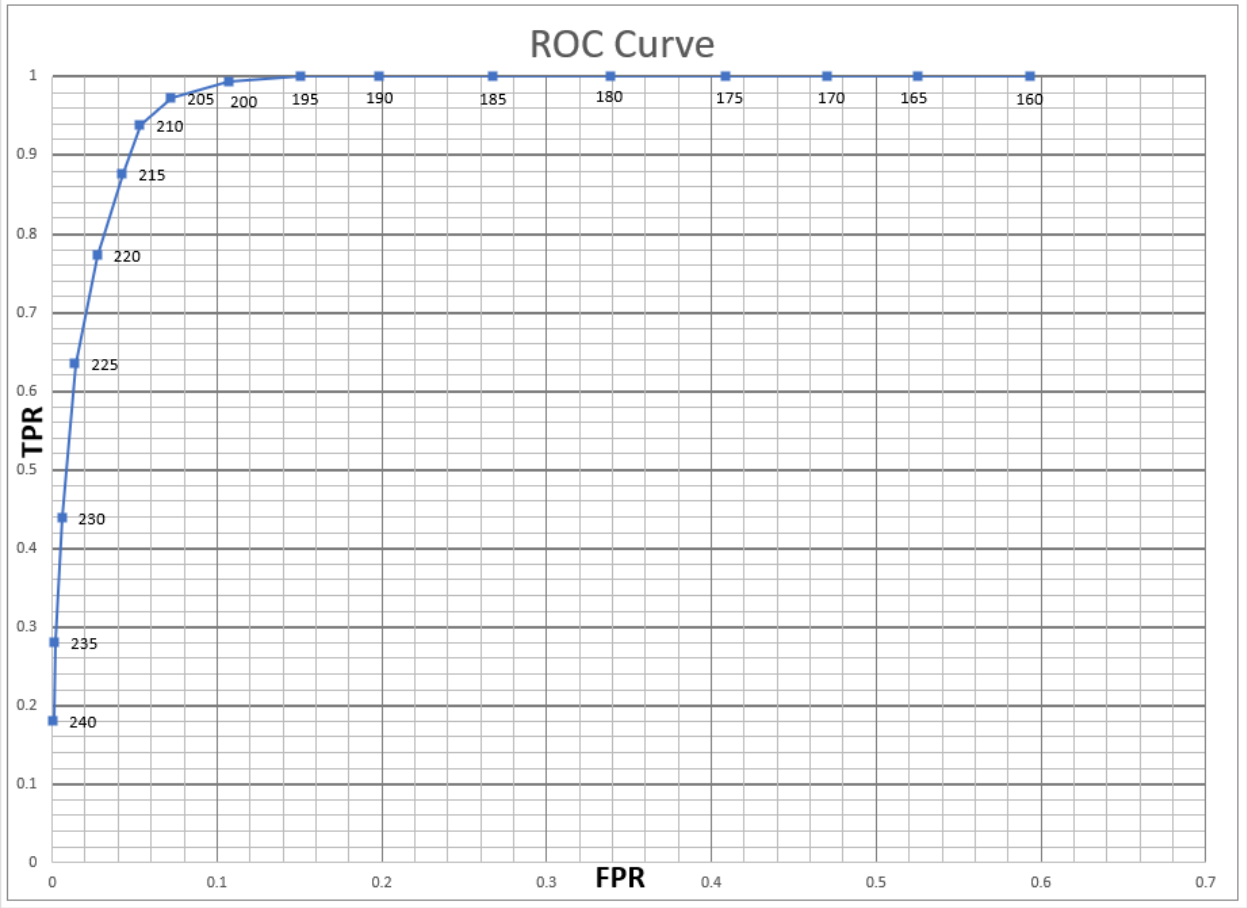

Figure 6: ROC Curve generated from data in Figure 5

Brandon Shumin
ECE 6310
Lab 3

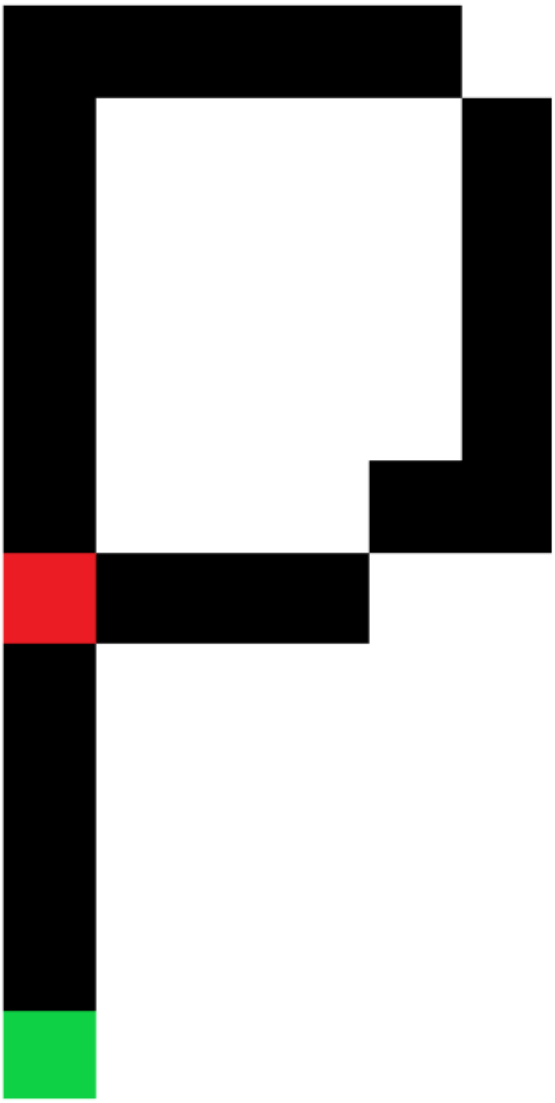

Figure 7: Example of where a branchpoint (Red) and endpoint (Green) would be detected using my code on Figure 4.

Brandon Shumin
ECE 6310
Lab 3

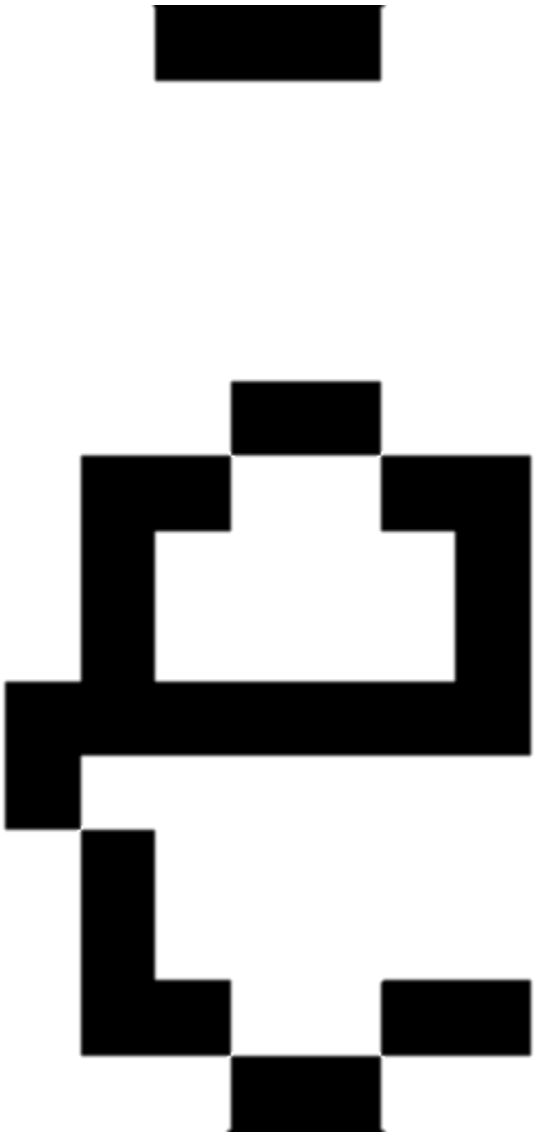

Figure 8: Example of an image clipping another letter above it, resulting in 3 endpoints and 1 branchpoint