ECE 6310

LAB 3 - OPTICAL CHARACTER RECOGNITION WITH SKELETONIZATION

September 28, 2021

Tim Driscoll
Clemson University
Department of Electrical and Computer Engineering
tdrisco@clemson.edu

1 Introduction

This lab report concerns the problem of optical character recognition with advanced techniques. A matched filter technique was used to to recognize the letter "e" in a sample image of text. Specifically, the technique that was used was matched spatial filtering. This was implemented by zero-mean centering the template, convolving the template with the image, normalizing the convolved image, then using thresholds to find matches. This method was further improved upon by thinning the image representing each desired character identified by the MSF filter. If the skeleton of the image had the desired number of branchpoints and endpoints it was confirmed to be the desired character. With the added algorithm of skeletonization we were able to reduce the number of false positives identified.

2 Methods

The methods used in lab three built upon those that were used to complete lab two. In lab two a matched spatial filter was used to identify the occurrence of the letter 'e' in a given template image. The matched spatial filter was thresholded at a series of values to determine which threshold would produce the optimal character recognition. Lab three takes this technique a step further by thinning the letters detected by the MSF filter and determining if they have the correct number of branchpoints and endpoints.

2.1 Letter Thresholding

After using the original matched spatial filtering algorithm, the letters that were identified as 'e' were further processed to verify the character. A 9x15 window from the original image was copied at the location of the letter. This 9x15 letter image was thresholded at a value of 128 to clearly seperate the background from the letter itself. An example of this thresholded image can be observed in figure 1. This image shows a clear separation between the background in black (0) and the letter in white (255).

2.2 Image Thinning

After the image was thresholded the main process of thinning could be performed. The process of thinning is where pixels representing the letters are iteratively deleted. Pixels are deleted until the representation of the letter is a single pixel wide outline. In order to determine if pixels should be deleted three conditions need to be met. The first condition is that only one edge to non-edge transition occurs around the entire pixel. The second condition is that the pixel has between 2 and 6 edge neighbor pixels. The final condition is that at least one of the north, east, or (west and south) neighbors are not edge pixels. If all three of these conditions are met a template image is marked with a one to represent a deletion at the given location. Once the entire image was parsed through the deletions were performed on the copy, thinning the image. This entire process is run until the complete parse of the copy image results in no more new deletions. It is common for it to take multiple iterations to completely thin the image. A thinned version of figure 1 can be observed in

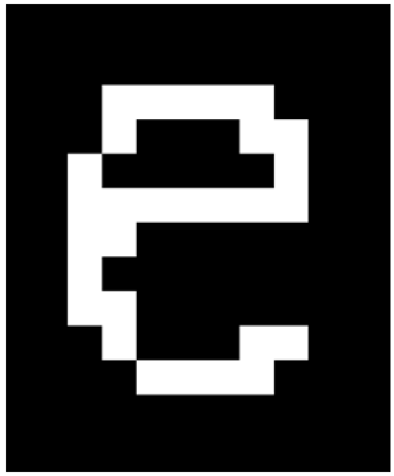

Figure 1: Letter copy after thresholding.

figure 2. It can be observed that the thinned image is a one pixel wide outline of the entire letter. Comparatively, the image that is only thresholded has multiple areas were the letter outline is more than one pixel wide.

2.3 Counting Branchpoints and Endpoints

Finally, for this algorithm to improve upon the matched spatial filter it was important to count the number of branch points and endpoints. For the letter we were identifying, 'e', there should only be one endpoint and one branchpoint. Endpoints and branchpoints are determined by counting the number edge to nonedge transitions at each edge pixel location. At a given edge pixel if there is exactly one edge to nonedge transition this represets an endpoint. If there is more than two transitions than this represents a branchpoint. The locations of the branchpoint and endpoint on the letter 'e' can be observed in figure 3. The branchpoint is represented by the grey pixel towards the top of the 'e' and the end point is represented by the grey pixel at the bottom left of the 'e'.

2.4 ROC Curve

Lastly, to view the results an ROC curve was produced. In order to plot this curve data was outputted from the c code to a text file (Roc_lab3.txt). This data included the false positive rate (FPR) and the true positive rate (TPR). The FPR would be plotted on the x-axis and the TPR on the y-axis to create the ROC curve. The following MATLAB script was used to plot the curve.

```
clear
clc
close all

fileID = fopen('Roc_lab3.txt','r');
```

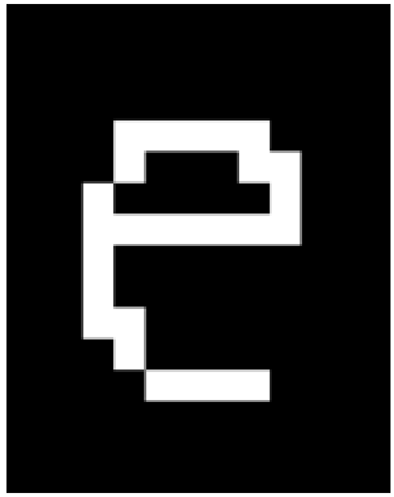

Figure 2: Letter copy after thinning.

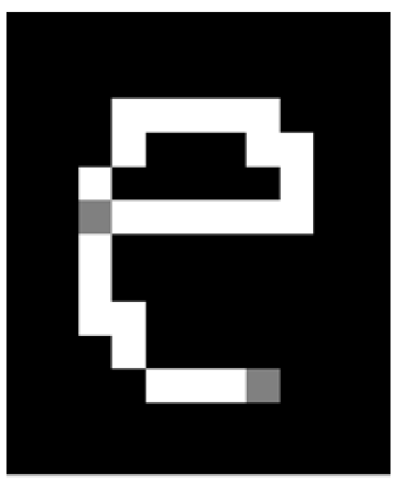

Figure 3: Thinned letter copy with branchpoint and endpoint identified.

```
data = fscanf(fileID,'%f %f',[2,Inf]);
data = data';

xdata = data(:,1);
ydata = data(:,2);

figure('Color','w');
plot(xdata,ydata,'ko','markerfacecolor','k','markersize',1);
hold on
plot(xdata,ydata,'k-');
axis([-0.08 0.15 0 1.2]);
xlabel("False Positive Rate (FPR)");
ylabel("True Positive Rate (TPR)");
```

3 Results

The results for the optical character recognition implementing a matched filter and skeletonization algorithm can best be viewed using a receiver operation characteristics curve. This curve plots the true false positive rate versus the false positive rate. Both these values are scaled from 0-1, so ideally we would like to find a threshold where the TPR is 1 and the FPR is 0. Looking at figure 4 a trade off can observed when moving from right to left across the curve. As you move to the left and the threshold increases we can observe a decrease in FPR. The FPR does not become minimized until the TPR starts to rapidly decrease creating the trade off. The ROC curve in figure 4 does not have a rapid fall off. The lack of rapid fall off allows for us to determine a threshold were the TPR is very high and the FPR is close to 0. It is also important to note that the maximum TPR was about 95%. The reason this was the maximum is because there was about 7 occurrences where the thinning algorithm failed to detect an "e". This failure was because pieces of neighboring letters from the original image were appearing in the reduced 9x15 window. These unwanted letter pieces caused for additional endpoints not associated with the 'e'.

4 Conclusion

Similar to lab two, it often depends on the scenario to determine what the optimal threshold value is. Given the case of optical character recognition I have chosen the case of locating a threshold at the knee of the ROC curve. Finding a threshold in this range will give be a very high true positive rate, and smaller false positive rate. In the case of character recognition almost all the characters being searched for will be identified and there will be a small group of characters that were falsely identified. After closely examining figure 4 I identified the optimal threshold to be 203. This threshold produced 143 true positives and 15 false positives. The true positive rate was about 95% and the false positive rate was about 1.3%. These results optimally balanced the trade off of increasing the threshold to lower the FPR while maintaining a high value for TPR. With the addition of the skeletonization

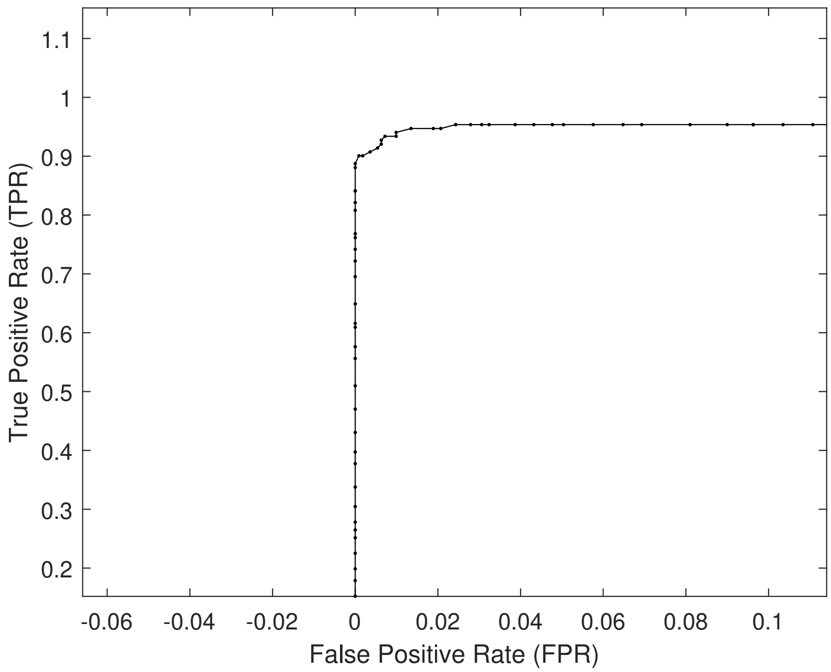

Figure 4: Receiver Operating Characteristics Curve

algorithm the false positive rate decreased and the true positive rate stayed the same. These were the expected results because the MSF regulated the maximum amount of 'e's that were detected at any give threshold. The skeletonization allowed us to further refine the group of detected characters decreasing the number of false positives that occurred. Even in optical character recognition there may be cases where this isn't the optimal threshold. There could be instances where it is more important correctly identify all characters regardless of the FPR or instances where it is more important for the FPR to be zero.