ECE 6310

LAB 2 - OPTICAL CHARACTER RECOGNITION

September 15, 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. 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.

2 Methods

2.1 Zero Mean Centered Template

The first step in this lab was to zero-mean center the template. The template was an image of a zoomed in "e" representing what we would be looking for in the parent image. The template was zero-meaned by determining the mean of all the values that the template was composed of. Once the mean was calculated each value in the original template was subtracted by the mean. It was important to make sure that the zero-mean version of the template was represented by integers to ensure that negative numbers could be handled.

2.2 Matched Spatial Filter Image

The second step in the lab was to convolve the zero-mean template with the original parent image. This convolution would ultimately create the MSF image. The convolution was performed by using the following equation:

$$MSF[r,c] = \sum_{dr=-Wr/2}^{+Wr/2} \sum_{dc=-Wc/2}^{+Wc/2} I[r+dr, c+dc] * T[dr+Wr/2, dc+Wc/2] \tag{1}$$

In equation 1 above the I represents the parent image and T represents the zero-mean centered template. It was again important in this step to save the MSF as a larger data type then a unsigned char. I chose to save this version of the image as a float to be able to handle the size of the numbers as well as the sign.

2.3 Normalized MSF Image

After the MSF image had been calculated it was important to normalize the image back to 8-bits. This would allow for the image to once again be represented by 8-bit characters from 0-255. In order to perform this normalization the following equation was performed on each value in the MSF image.

$$NormalizedImage_i = 255 * \frac{MSFImage_i - min}{max - min} \tag{2}$$

In equation 2 above the normalized value is calculated at any given location represented by i, which is dependent on the MSFImage at the corresponding position i. The max and min

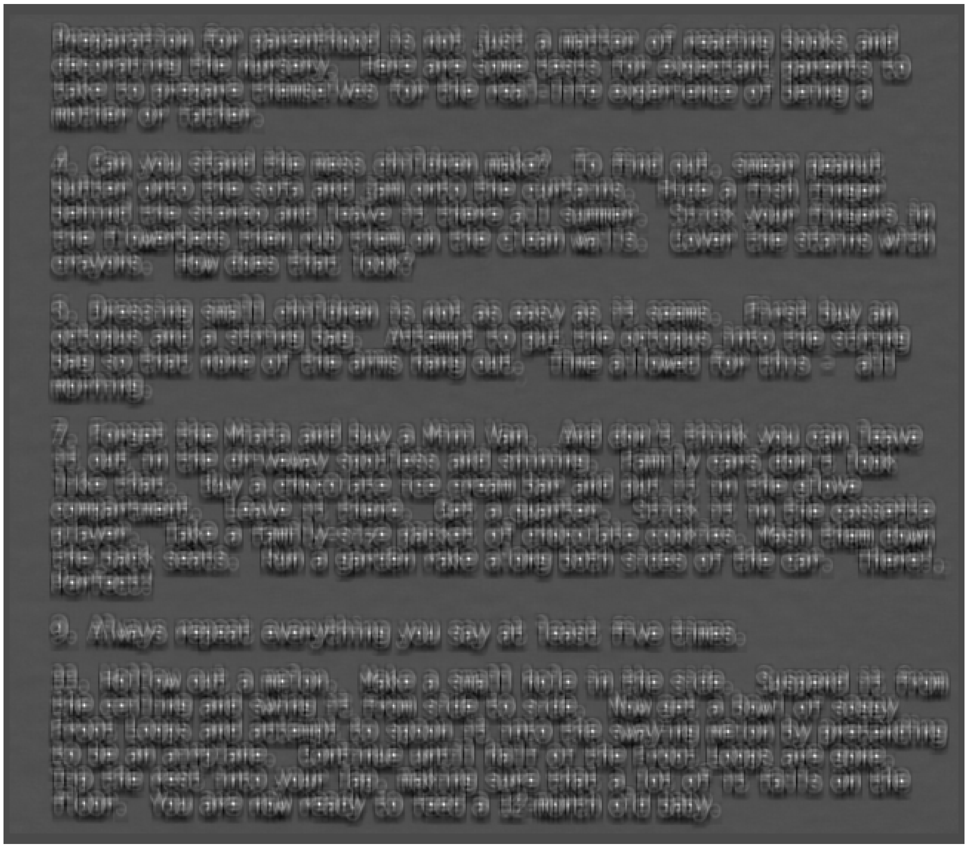

Figure 1: Normalized MSF Image

that were used in this equation represent the max and min of the original MSF image. The value of 255 is used to represent the range that the image is being normalized to (0-255). Figure 1 represents the MSF image after it has been normalized to a range of (0-255). In this image it is important to view the brighter spots or little white dots. These areas are points of interests because they represent locations with higher potential of containing an "e".

2.4 Threshold

The final step in the lab was to threshold the image to find actual matches. A range of thresholds from 0-255 were tested in order to view the effects of different values. This also allowed for the best threshold value to be determined. A binary version of the image was created based on the results of thresholding, which was created using the following equation:

$$BinaryImage[r,c] = \begin{cases} 1 & MSF[r,c] \geq Threshold \\ 0 & MSF[r,c] < Threshold \end{cases} \tag{3}$$

Once the thresholding was completed the binary image could be compared to the ground truth to categorize and count the detected letters.

2.5 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.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.txt','r');
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([-0.3 0.5 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 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 2 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.

4 Conclusion

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 be searched for will be identified and there will be small group of characters that

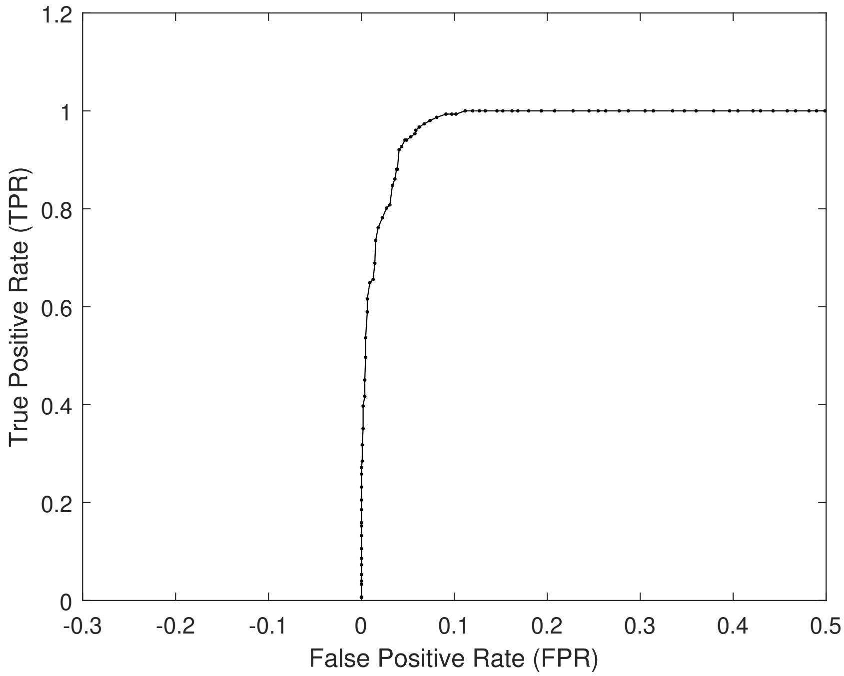

Figure 2: Receiver Operating Characteristics Curve

were falsely identified. After closely examining figure 2 I identified the optimal threshold to be 210. This threshold produced 143 true positives and 59 false positives. The true positive rate was about 95% and the false positive rate was about 5%. These results optimally balanced the trade off increasing the threshold to lower the FPR while maintaining a high value for TPR. 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.