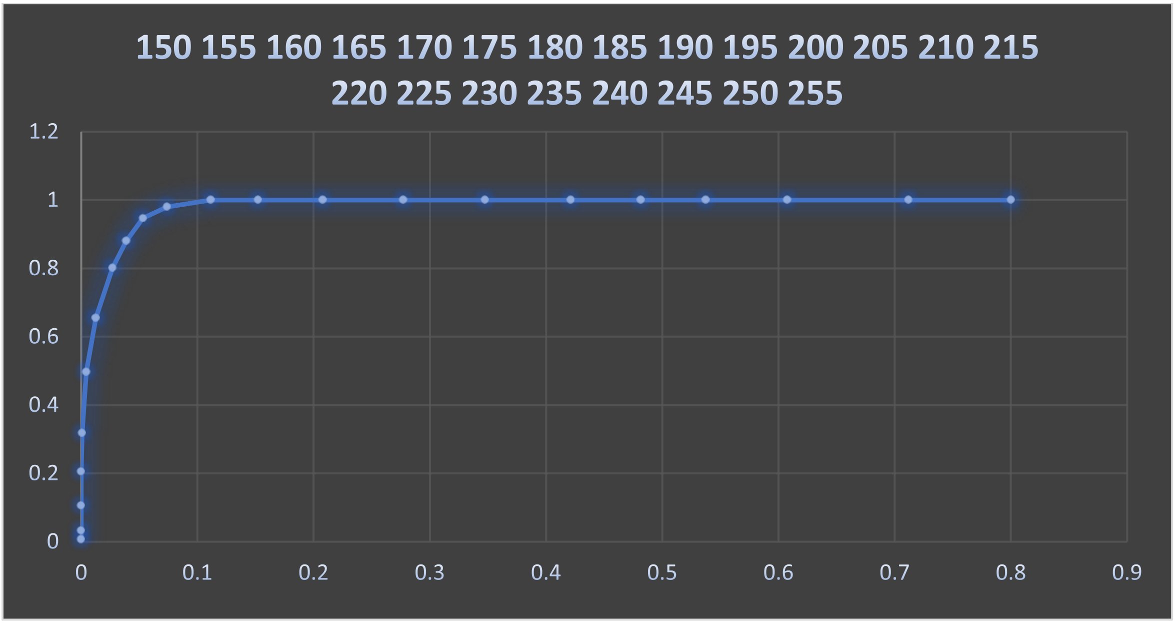
ECE 4310/6310 Introduction to Computer Vision

Lab #2 – Optical Character Recognition

Name- Pinak Kelkar

First the MSF filter was calculated and with which you can see below.

After that the we start putting the filter on the real image and then I have used the maximum value to normalize the image in 8-bit format by using this formula (Value-min/ Max-min) *255. I have used a in Array to hold all the large numbers and then normalize hem in 8-bit format

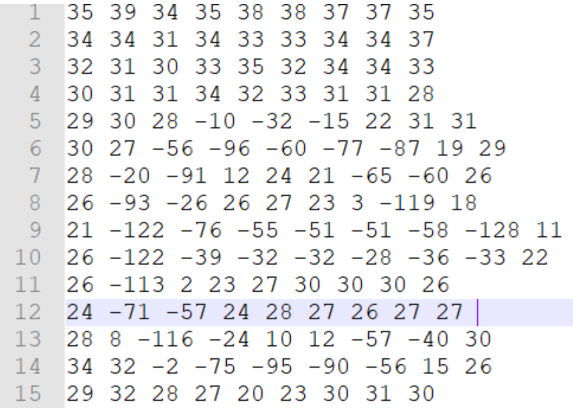


This is the result table True Positive, False Positive True Negative and False Negative Calculation for all the thresholds from 150 to 255. So according to the ROC curve we can see that the optimal threshold for this image for detecting 'e' would be between 200 to 220. According to the truth table there are 151 'e' in the image, we can detect all of them at a threshold of 200 but with a false negative of 124 detections.

Result Table

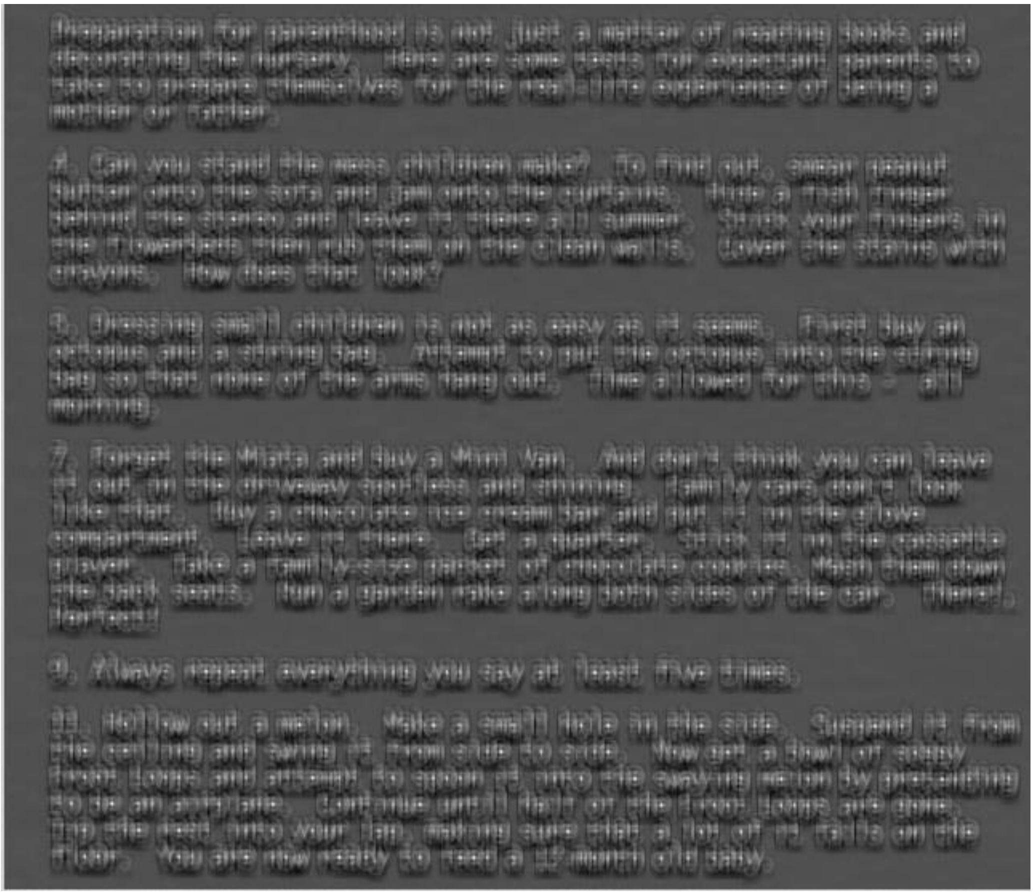
Threshold	TP	FP	TN	FN	TPR	FPR
150	151	889	222	0	1	0.80018
155	151	791	320	0	1	0.711971
160	151	675	436	0	1	0.607561
165	151	597	514	0	1	0.537354
170	151	535	576	0	1	0.481548
175	151	468	643	0	1	0.421242
180	151	386	725	0	1	0.347435
185	151	308	803	0	1	0.277228
190	151	231	880	0	1	0.207921
195	151	169	942	0	1	0.152115
200	151	124	987	0	1	0.111611
205	148	82	1029	3	0.980132	0.073807
210	143	59	1052	8	0.94702	0.053105
215	133	43	1068	18	0.880795	0.038704
220	121	30	1081	30	0.801325	0.027003
225	99	14	1097	52	0.655629	0.012601
230	75	5	1106	76	0.496689	0.0045
235	48	1	1110	103	0.317881	0.0009
240	31	0	1111	120	0.205298	0
245	16	0	1111	135	0.10596	0
250	5	0	1111	146	0.033113	0
255	1	0	1111	150	0.006623	0

MSF FILTER IMAGE



```
35 39 34 35 38 38 37 37 35
34 34 31 34 33 33 34 34 37
32 31 30 33 35 32 34 34 33
30 31 31 34 32 33 31 31 28
29 30 28 -10 -32 -15 22 31 31
30 27 -56 -96 -60 -77 -87 19 29
28 -20 -91 12 24 21 -65 -60 26
26 -93 -26 26 27 23 3 -119 18
21 -122 -76 -55 -51 -51 -58 -128 11
26 -122 -39 -32 -32 -28 -36 -33 22
26 -113 2 23 27 30 30 30 26
24 -71 -57 24 28 27 26 27 27
28 8 -116 -24 10 12 -57 -40 30
34 32 -2 -75 -95 -90 -56 15 26
29 32 28 27 20 23 30 31 30
```

MSF Image



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.

Code for C

```
#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#include <string.h>

int main (int argc, char *argv[])

{

            if (argc!=2)

   {

       printf("Enter Threshold");

       return 0;

   }

FILE *fpt2,*fpt,*ftr;

unsigned char *template;

unsigned char *image;

int *MSF;
int *MSF_Image;
char header[80];
int ROWS,COLS,BYTES;
char T_header[80];
int T_ROWS,T_COLS,T_BYTES;
int i,j,Threshold=atoi(argv[1]);
int r=0,c=0;
int sum=0,total=0;
int max=0,min=0;
int counter=0;
int r1=0,c1=0;

char * line = NULL;
size_t len = 0;
ssize_t read;

fpt2=fopen("parenthood_e_template.ppm","r");
fpt=fopen("parenthood.ppm","r");
ftr=fopen("parenthood_gt.txt","r");
if (fpt2 == NULL || fpt == NULL) {
  printf("Unable to open for reading\n");
  exit(0);
  }

 i=fscanf(fpt2,"%s %d %d %d",T_header,&T_COLS,&T_ROWS,&T_BYTES);
 j=fscanf(fpt,"%s %d %d %d",header,&COLS,&ROWS,&BYTES);
/* allocate dynamic memory for image */
template=(unsigned char *)calloc(T_ROWS*T_COLS,sizeof(unsigned char));
image=(unsigned char *)calloc(ROWS*COLS,sizeof(unsigned char));
MSF=(int *)calloc(T_ROWS*T_COLS,sizeof(int));
MSF_Image=(int *)calloc(ROWS*COLS,sizeof(int));
```

```
if (template == NULL || image==NULL || MSF==NULL) {
  printf("Calloc Failed\n");
  exit(0);
  }
  T_header[0]=fgetc(fpt2);
  fread(template,1,T_ROWS*T_COLS,fpt2);

  header[0]=fgetc(fpt);
  fread(image,1,ROWS*COLS,fpt);
fclose(fpt2);
 fclose(fpt);

 ///creating MSF Image
  for(r=0;r<T_ROWS;r++){
    for(c=0;c<T_COLS;c++){
    sum+=template[r*T_COLS+c];
    }
  }
 total=(T_COLS*T_ROWS);
sum=sum/total;
   for(r=0;r<T_ROWS;r++){
    for(c=0;c<T_COLS;c++){
    MSF[r*T_COLS+c]=template[r*T_COLS+c]-sum;
    }
}

  i=(T_ROWS/2);
  j=(T_COLS/2);
  sum=0;

   for(r=i;r<ROWS-i;r++){
```

```
  for(c=j;c<COLS-j;c++){


   int i1=0,j1=0;

   for(r1=-i;r1<=i;r1++){

    j1=0;

    for(c1=-j;c1<=j;c1++){


      sum+=(MSF[i1*T_COLS+j1]*image[(r+r1)*COLS+(c+c1)]);

    j1++;

    }

               i1++;

   }

       if(max<sum){

       max=sum;

       }

       if(min>sum){

       min=sum;

       }

              MSF_Image[r*COLS+c]=sum;

              sum=0;

       }

  }



  for(r=i;r<ROWS-i;r++){

  for(c=j;c<COLS-j;c++){


   image[r*COLS+c]=((((double)MSF_Image[r*COLS+c]-min)/(double)(max-min))*255);

              }

       }


int TP=0,FP=0;
int TN=0,FN=0;

 while ((read = getline(&line, &len, ftr)) != -1) {
 int pixelc=0;
 int pixelr=0;
       char *token = strtok(line, " ");
  token = strtok(NULL, " ");
       pixelc=atoi(token);
       token = strtok(NULL, " ");
       pixelr=atoi(token);

 int Flag=0;

 for(r=-i;r<=i;r++){
  for(c=-j;c<=j;c++){
              if(image[(pixelr+r)*COLS+(pixelc+c)]>=Threshold){
                     if(line[0]=='e'){
                     TP++;
                     Flag=1;
                     }
                     else{
                     FP++;
                     Flag=1;
                     }

                     break;
              }
       }
        if(Flag==1){break;}  ///Leave if Flag is set and a value above threshold is found
}
        ///Enter if no no value is above Threshold
       if(Flag==0){
```

```
                        if(line[0]=='e'){

                        FN++;

                        }

                        else{

                        TN++;

                        }

 }

         Flag=0;

 }///End While

        printf("TP-%d FP-%d TN-%d FN-%d  Total-%d Threshold-%d\n",TP,FP,TN,FN,TP+FP+TN+FN,Threshold);
```

```
/* write out brighter.ppm, the output result */

fpt=fopen("brighter.ppm","w");

fprintf(fpt,"P5 %d %d 255\n",COLS,ROWS);

fwrite(image,1,ROWS*COLS,fpt);

fclose(fpt);

}
```

Shell used to run the program from 150 to 255 threshold.

```
gcc -o lab2 lab2.c

for i in {150..255..5}

 do

   a=./lab2

         a+=' '$i

         $a

 done
```