Shashi Shivaraju
XID:C88650674

REPORT
ECE 6310 Lab #2 – Optical Character Recognition

Objective: To implement a Matched-Spatial Filter(normalized cross-correlation) to recognize letters in an image of text.

Implementation: Usage of Matched-Spatial Filter(MSF)to recognize a template in the input image follows the below steps:

a)Zero-mean center the template image.

Calculate the mean of the pixel values of the entire template image and subtract the mean from each pixel of the template image to obtain the zero mean template image.

[Code snippet]:

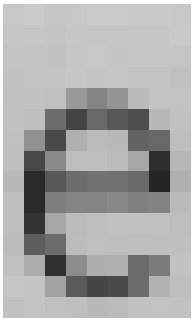
```
/*Calculate the average of the pixels of template image*/
     for(i = 0;i<row_template*col_template;i++)
     {
          sum = sum + template[i];
     }
     average = sum/(row_template*col_template);

/*Calculate the zero mean template*/
     for(i = 0;i<row_template*col_template;i++)
     {
          template_zero_mean[i] = template[i]-average;
     }
```

Here,
int row_template = height of the template image in pixels.
int col_template = width of the template image in pixels.
int sum = summation of the pixel values of the template image.
int average = mean of the pixel values of the template image
unsigned char* template is a pointer to memory storing template image data.
int* template_zero_mean is a pointer to memory storing zero mean template image.

Shashi Shivaraju
XID:C88650674

[Template Image]:

9 x 15 8bit Gray scale image of PPM format named parenthood_e_template.ppm



[Pixel value representation of the above Template Image]:

200	204	199	200	203	203	202	202	200
199	199	196	199	198	198	199	199	202
197	196	195	198	200	197	199	199	198
195	196	196	199	197	198	196	196	193
194	195	193	155	133	150	187	196	196
195	192	109	69	105	88	78	184	194
193	145	74	177	189	186	100	105	191
191	72	139	191	192	188	168	46	183
186	43	89	110	114	114	107	37	176
191	43	126	133	133	137	129	132	187
191	52	167	188	192	195	195	195	191
189	94	108	189	193	192	191	192	192
193	173	49	141	175	177	108	125	195
199	197	163	90	70	75	109	180	191
194	197	193	192	185	188	195	196	195

Template Image pixel value Sum is 22328; Average is 165

[Pixel value representation of the Zero Mean Template 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

Shashi Shivaraju
XID:C88650674

b)Preform convolution(cross-correlation)of the input image of text with the zero mean template to obtain matched-spatial filter(MSF) image.

[Code snippet]:

```
for(R=0;R<=row-(row_template);R++)
{
    for(C=0;C<=col-(col_template);C++)
    {
       sum = 0;
       for(r1=0;r1<row_template;r1++)
       {
         for(c1=0;c1<col_template;c1++)
         {
                sum = sum +
                image[col*(r1+R)+(c1+C)]*template_zero_mean[r1*col_template+c1];
         }
       }
    }
    MSF_image[col*(R+row_template/2)+(C+col_template/2)]=sum;
}
```

c)Normalize the MSF image to 8-bits

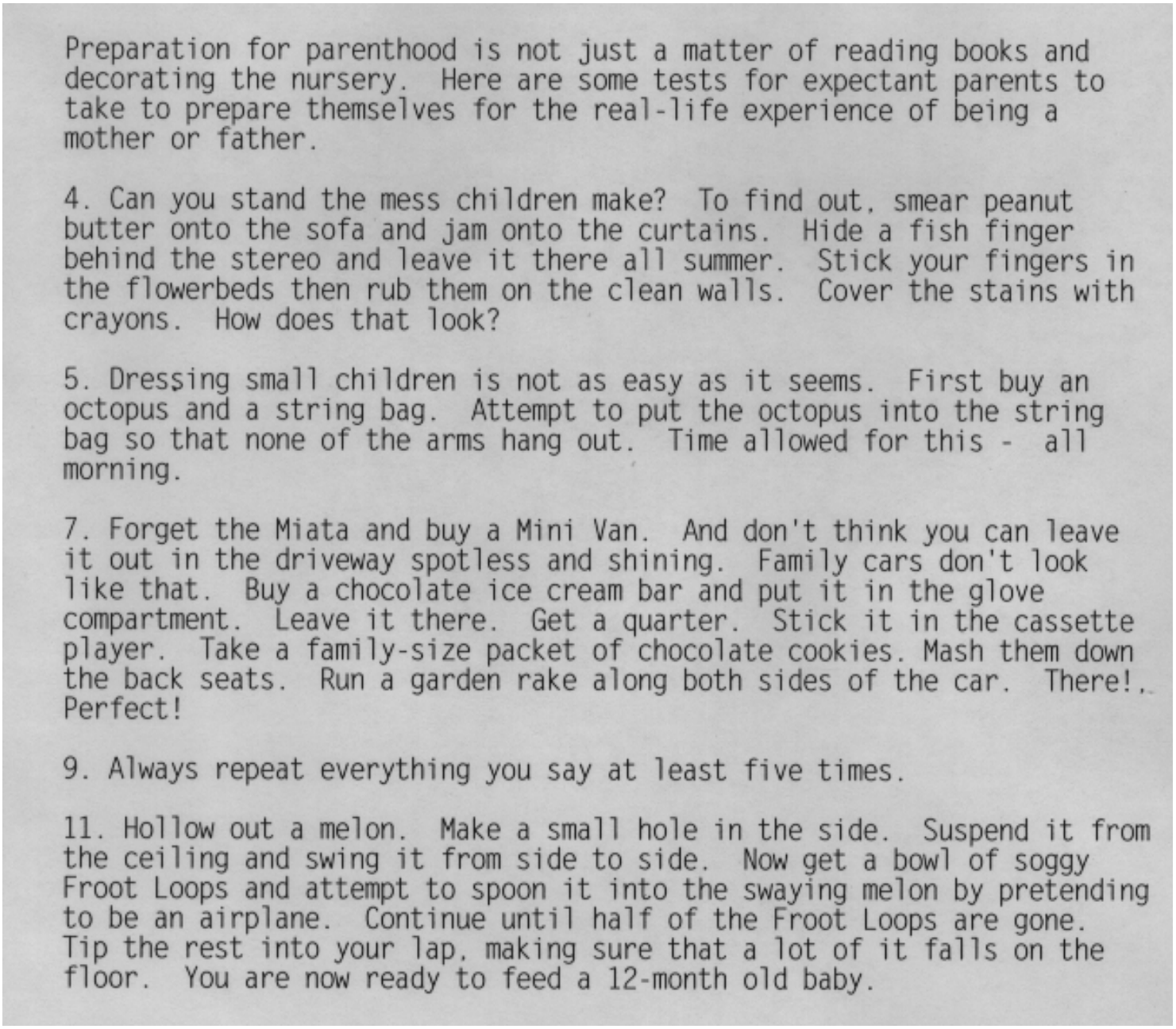
[Code snippet]:

```
for(i=0;i<row*col;i++)
{
      MSF_Pixel = (float)MSF_image[i];
      out = ((MSF_Pixel-min)/(max-min))*255;
      /*rounding the values to the next highest integer if value above 0.5  */
      if(out-(unsigned char)out > 0.5)
      {
            MSF_Normalized[i] = (unsigned char)(out+1);
      }
      else
      {
            MSF_Normalized[i] = (unsigned char)out;
      }
}
```

Here,
int row_template = height of the template image in pixels.
int col_template = width of the template image in pixels.
int row = height of the input image of text in pixels.
int col = width of the input image of text in pixels.
unsigned char* image is a pointer to memory storing input image of text.
int* template_zero_mean is a pointer to memory storing zero mean template image.
int* MSF_image is a pointer to memory storing matched-spatial filter image.
unsigned char* MSF_Normalized is a pointer to memory storing Normalized MSF image.
int min= minimum pixel value of the MSF_image.
int max= maximum pixel value of the MSF_image.

Shashi Shivaraju
XID:C88650674

[Input Image]:
649 x 567 8bit Gray scale image of PPM format named parenthood.ppm



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.

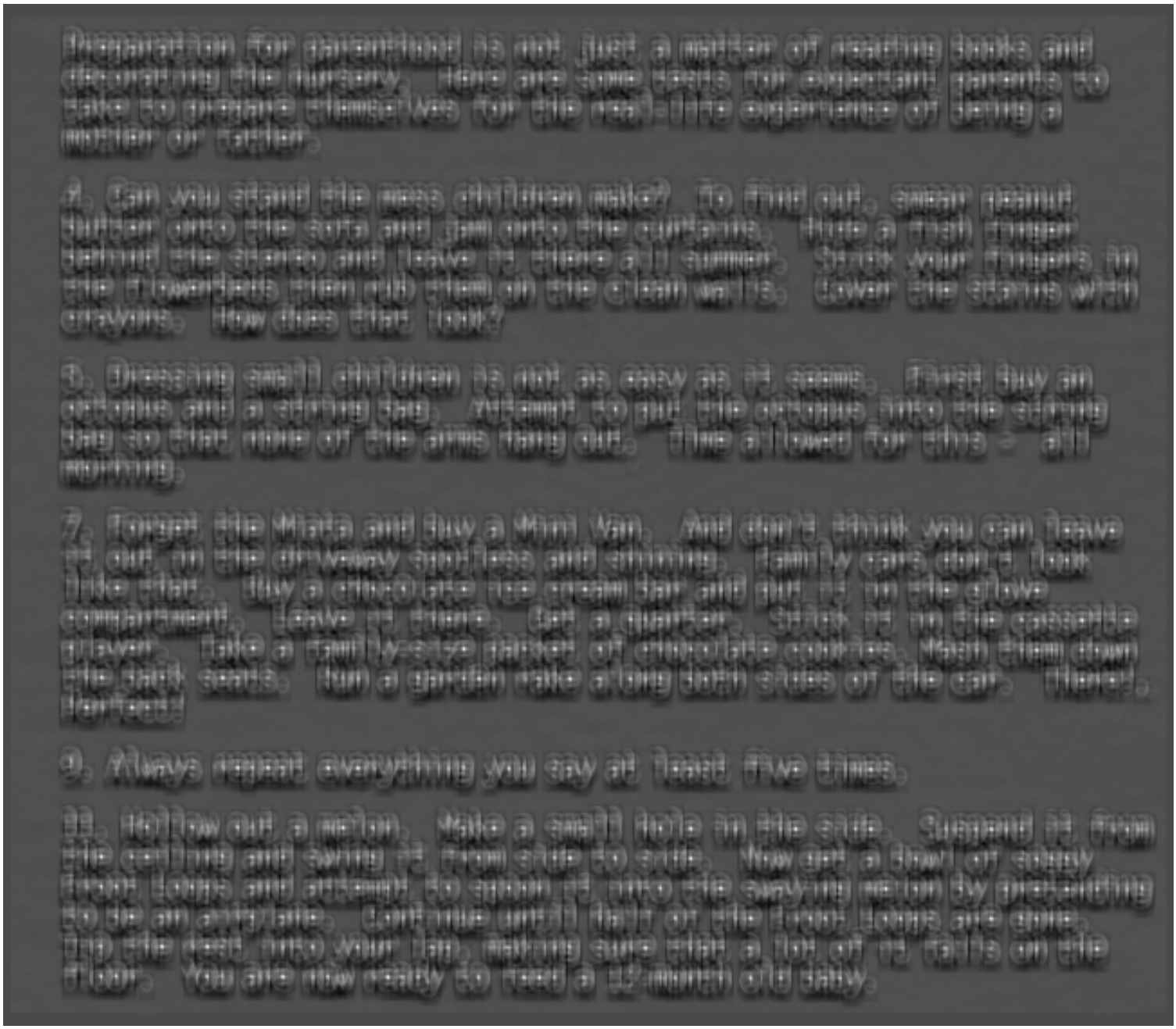
7. Forget the Miata and buy a Mini Van. And don't think you can leave it out in the driveway spotless and shining. Family cars don't look like that. Buy a chocolate ice cream bar and put it in the glove compartment. Leave it there. Get a quarter. Stick it in the cassette player. Take a family-size packet 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.

Shashi Shivaraju
XID:C88650674

[Normalized MSF image]:
649 x 567 8bit Gray scale image of PPM format named MSF_Normalized_output.ppm



Shashi Shivaraju
XID:C88650674

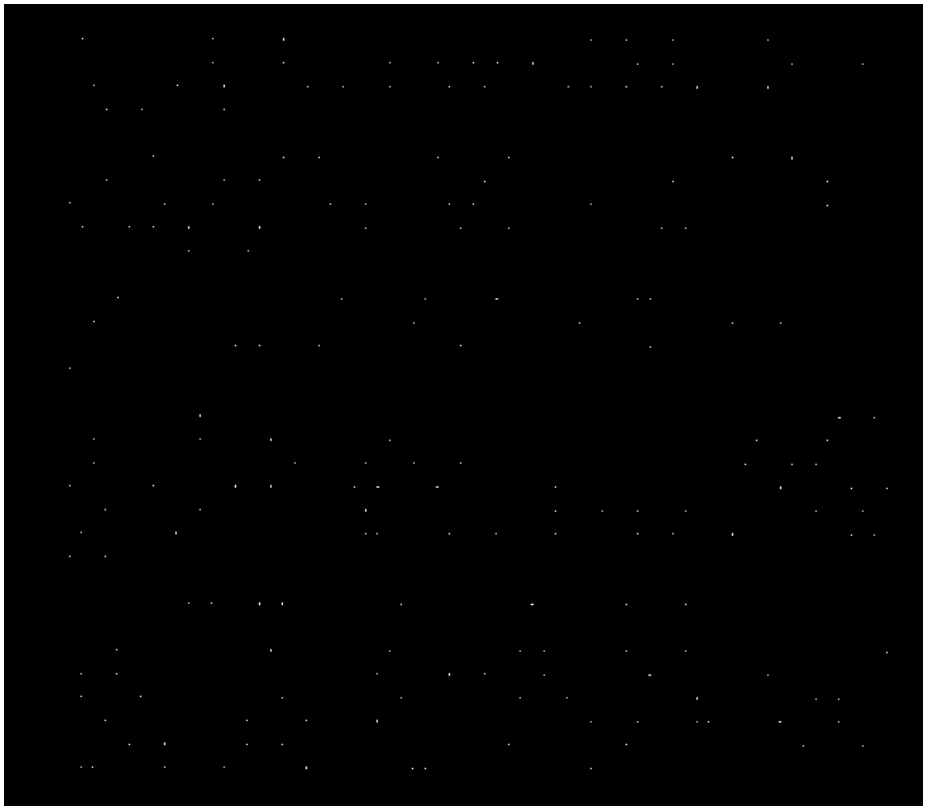
d)Threshold the normalized MSF image at pixel value T to create a binary image/dotted image to represent detections of template.

[Code snippet]:

```
for(i=0;i<row*col;i++)
{
         if(threshold < MSF_Normalized[i])
         {
            /*255 for representation by white dot or 1 for symbolic representation*/
             binary[i] = 255;
         }
         else
         {
             binary[i] = 0;
         }
}
```

Here,
int row = height of the input image of text in pixels.
int col = width of the input image of text in pixels.
int threshold= threshold pixel value above which represents the detected template.
unsigned char* MSF_Normalized is a pointer to memory storing Normalized MSF image.
unsigned char* binary is a pointer to memory of image representing the template detections.

[Dotted image representing the template detections]:
649 x 567 8bit Gray scale image of PPM format named Detection_output.ppm



Shashi Shivaraju
XID:C88650674

e)Verify the template detections obtained using the ground truth and calculate the True Positive Rate (TPR) and False Positive Rate (FPR).

For each location obtained from the ground truth file (parenthood_gt.txt),we check an area of the template dimension centered at the ground truth location in the Binary Image.

If any pixel in this area is 1,we consider that the template is detected.
Further if the input template matches the ground truth value then it is considered True Positive(TP) else it is considered False Positive(FP).

If none of the pixels in this area is 1,we consider the template is not detected.
Further if the input template matches the ground truth value then it is considered False Negative(FN) else it is considered True Negative(TN).

The True Positive Rate(TPR) and False Positive Rate(FPR) of the system is calculated as below:

- **TPR = TP/(TP+FN)**
- **FPR = FP/(FP+TN)**

[Code snippet]:

```
while(0<fscanf(fp,"%c %d %d ",&gt_char,&gt_col,&gt_row))
{
      detection_flag = 0;

      /*Check a 9 x 15 pixel area centered at the ground truth location
        of the Binary file to detect the character*/

      for(R=gt_row-(row_template/2);R<=gt_row+(row_template/2);R++)
      {
            for(C=gt_col-(col_template/2);C<=gt_col+(col_template/2);C++)
            {
              if(binary_image[R*col+C])
              {
                    detection_flag = 1;
                    break;
               }
            }

            if(detection_flag)
            break;
      }
```

Shashi Shivaraju
XID:C88650674

```
            if(detection_flag)
            {
                  if('e' == gt_char)
                        TP_Count++;  /* detected and it is template*/
                  else
                        FP_Count++;  /* detected but it is not template */
            }
            else
            {
                  if('e' == gt_char)
                        FN_Count++;  /* not detected but it is template */
                  else
                        TN_Count++;  /* not detected and the letter is not template*/
            }
            detection_flag = 0;
      }

TP_Rate  = ((float)TP_Count/(float)(TP_Count+FN_Count));
FP_Rate  = ((float)FP_Count/(float)(FP_Count+TN_Count));
```

Here,

```
int row_template = height of the template image in pixels.
int col_template = width of the template image in pixels.
int row          = height of the input image of text in pixels.
int col          = width of the input image of text in pixels.
unsigned char gt_char represents the ground truth value.
int gt_row,int gt_col represents the ground truth pixel location.
int FP_Count = instances of detected but the letter is not template.
int TP_Count = instances of detected and the letter is template.
int FN_Count = instances of not detected but the letter is template.
int TN_Count = instances of not detected and the letter is not template.
float TP_Rate = True Positive Rate(TPR)
float FP_Rate = False Positive Rate(FPR)
```

Shashi Shivaraju
XID:C88650674

System Analysis for Range of Threshold:

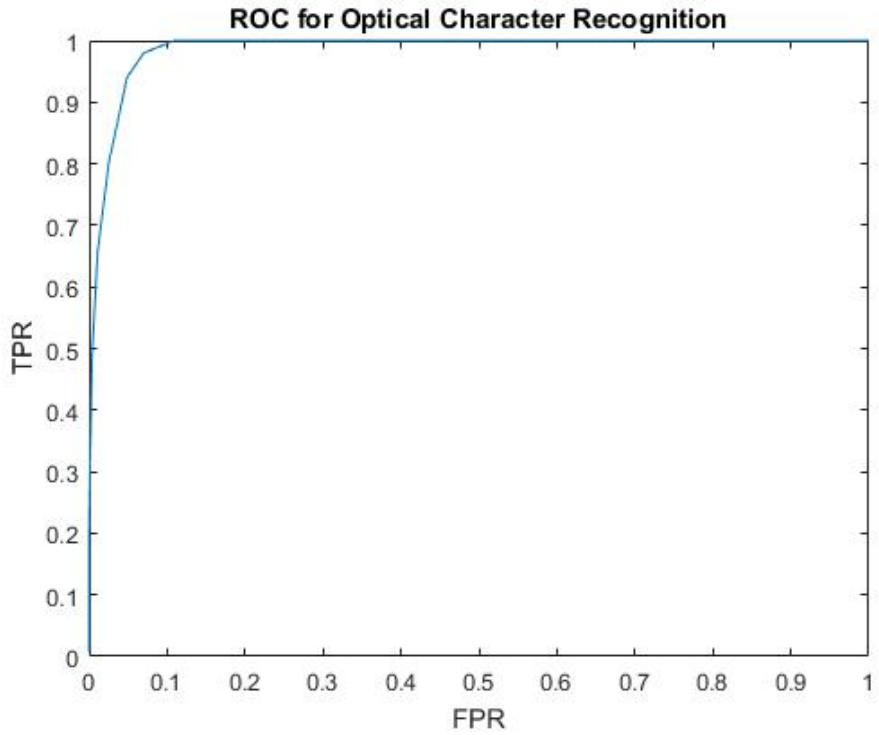
T (Threshold Pixel Value for Detection)	TPR (True Positive Rate)	FPR (False Positive Rate)	TP (True Positive Count)	FN (False Negative Count)	FP (False Positive Count)	TN (True Negative Count)
254	0.006623	0.000000	1	150	0	1111
250	0.026490	0.000000	4	147	0	1111
245	0.105960	0.000000	16	135	0	1111
240	0.192053	0.000000	29	122	0	1111
235	0.291391	0.000900	44	107	1	1110
230	0.483444	0.003600	73	78	4	1107
225	0.655629	0.010801	99	52	12	1099
220	0.801324	0.025203	121	30	28	1083
215	0.880795	0.038704	133	18	43	1068
210	0.940439	0.048605	142	9	54	1057
205	0.980132	0.070207	148	3	78	1033
200	1.000000	0.108911	151	0	121	990
195	1.000000	0.148515	151	0	168	946
190	1.000000	0.198920	151	0	221	890
180	1.000000	0.339334	151	0	377	734
170	1.000000	0.473447	151	0	526	585
160	1.000000	0.598560	151	0	665	446
150	1.000000	0.794779	151	0	883	228
140	1.000000	0.904590	151	0	1005	106
130	1.000000	0.963096	151	0	1070	41
120	1.000000	0.990099	151	0	1100	11
110	1.000000	0.999100	151	0	1100	1
100	1.000000	1.000000	151	0	1111	0
75	1.000000	1.000000	151	0	1111	0
50	1.000000	1.000000	151	0	1111	0
25	1.000000	1.000000	151	0	1111	0
5	1.000000	1.000000	151	0	1111	0

Shashi Shivaraju
XID:C88650674

Receiver Operating Characteristic (ROC):

ROC is a plot of True Positive Rate(TPR) vs False Positive Rate(FPR) as a function system variable and in this scenario the system variable is the pixel value Threshold(T) used to determine the template detections.

Below is the plot of ROC for the implemented Optical Character Recognition of template "parenthood_e_template.ppm" in image "parenthood.ppm"



Selection of Optimal T(Threshold Pixel Value for Template Detection):

As per the ground truth file "parenthood_gt.txt" provided there are 151 instances of the given template in the input image which this implementation should detect.

Thus an optimal T would be one in which we get high TP count(detected as template and the letter is template) and low FP count (detected as template but the letter is not template).From the ROC curve plot above, it can be observed the this condition is represented by the arc of the curve for which the range of T is [205,220].
Hence the Optimal T value should be selected form the range [205,220].