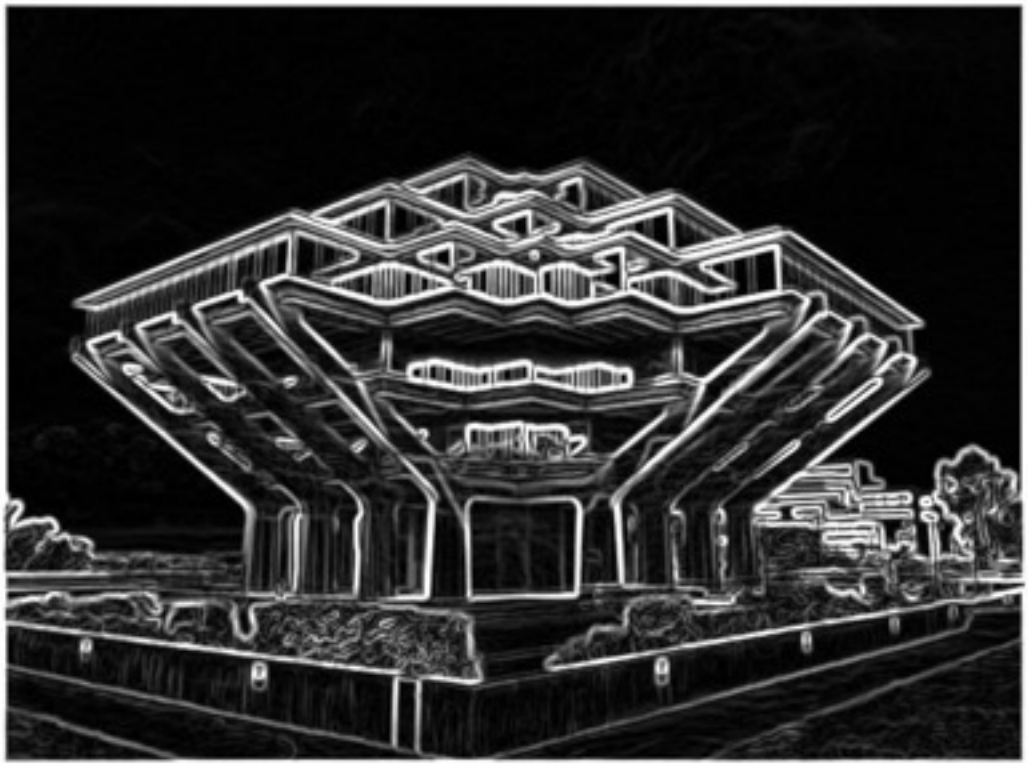
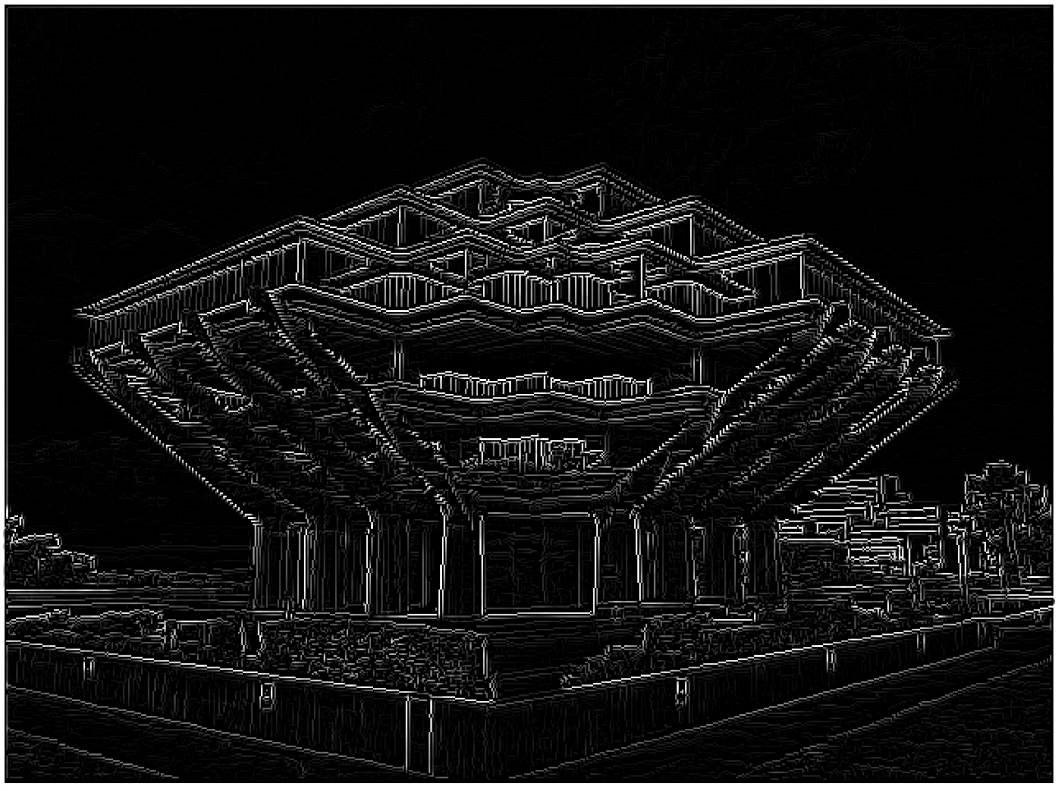
ECE172A HW4
Ruomei Ye
A99074215

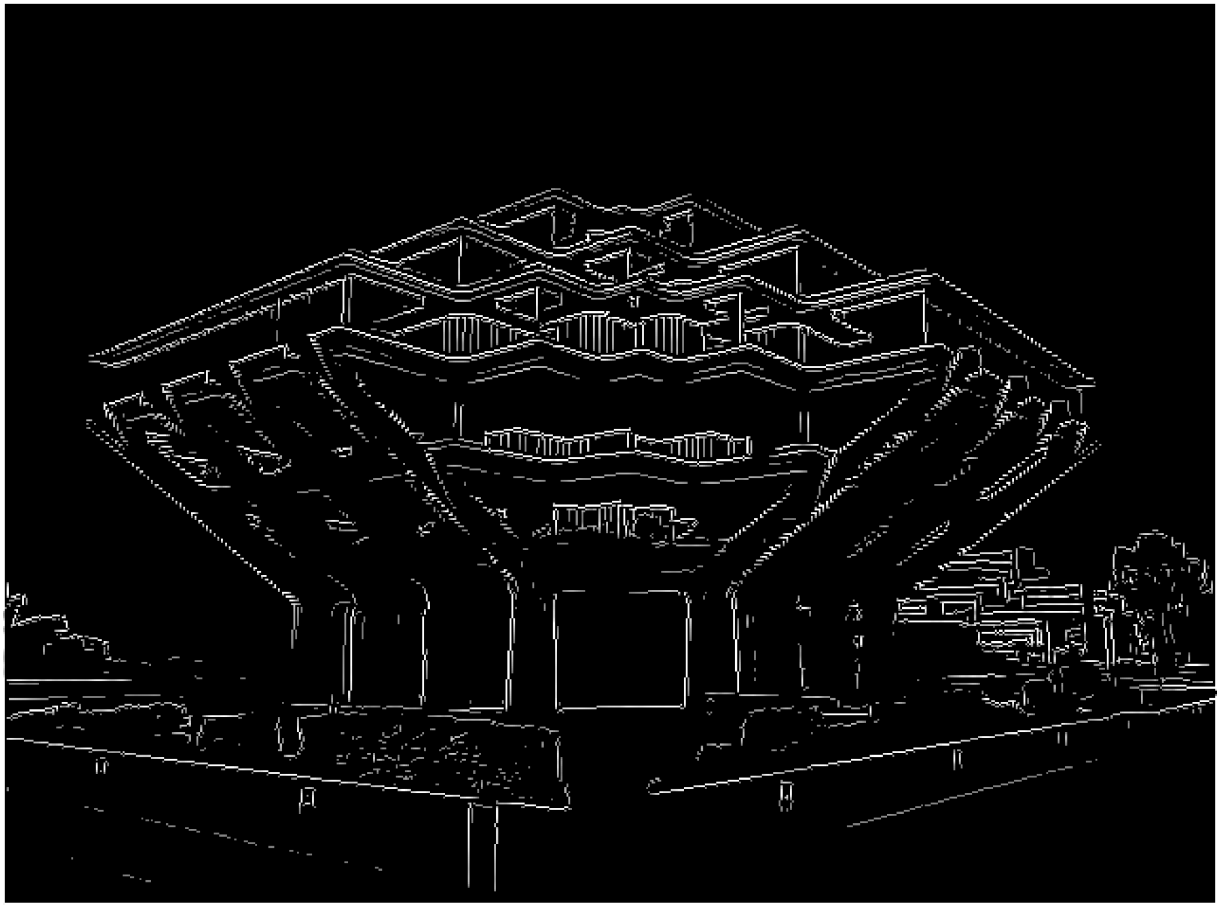
Problem1
original gradient magnitude picture:



After non-maximum suppression:



final edge image:



The threshold I chose for final edge image is 120.
MATLAB code please see the appendix.

Problem 2

original image:

win_size =33:

win_size
=65:

win_size=129:

histeq():

- How does the original image qualitatively compare to the images after AHE and HE respectively?

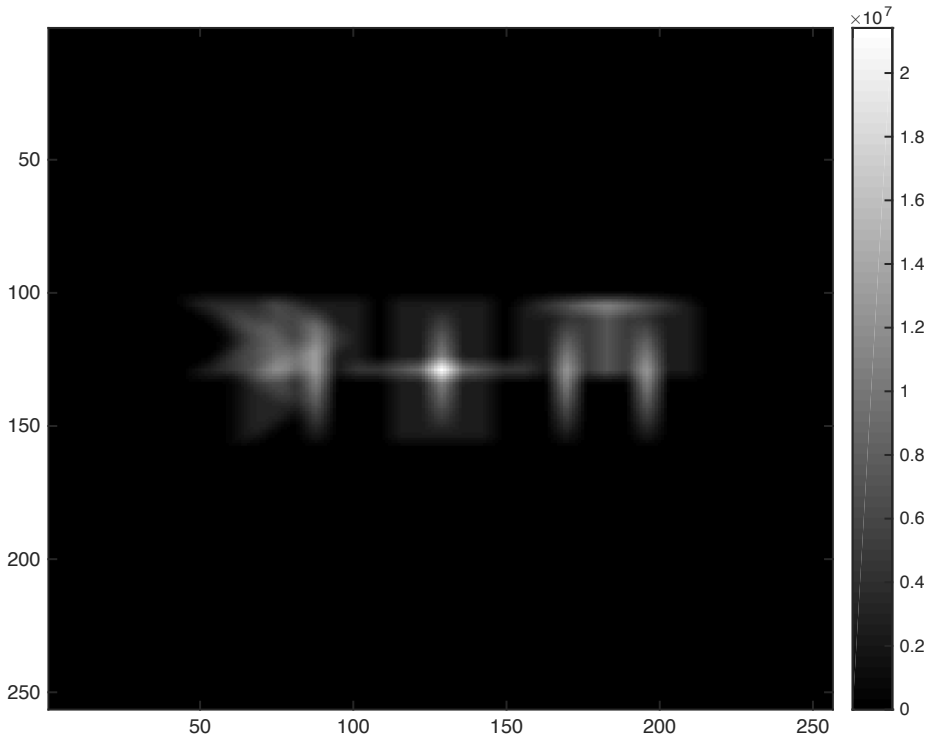
The histogram equalization (HE) will increase the contrast of the whole picture by getting the cumulative histogram and then re-distributing the intensity.
The adaptive histogram equalization(AHE) is similar to HE, but instead of taking all the histogram from the image, it calculates different histograms from several sections and then redistributes the intensity.
Therefore, the AHE could adjust the local contrast of the original image rather than just adjust the global contrast like HE does.

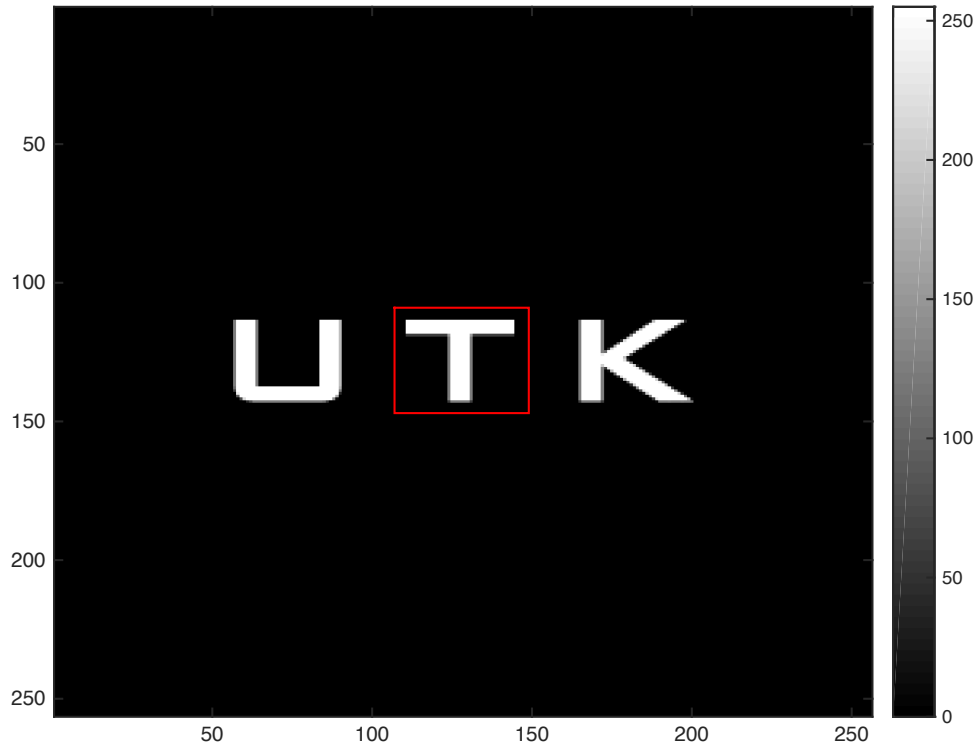
- Which strategy (AHE or HE) works best for beach.png and why? Is this true for any image in general?

AHE is better for beach.png. The sky in the picture is very bright while the house is dark. Its histogram is not very close contrast and therefore AHE which adjusts local contrasts is better than HE.

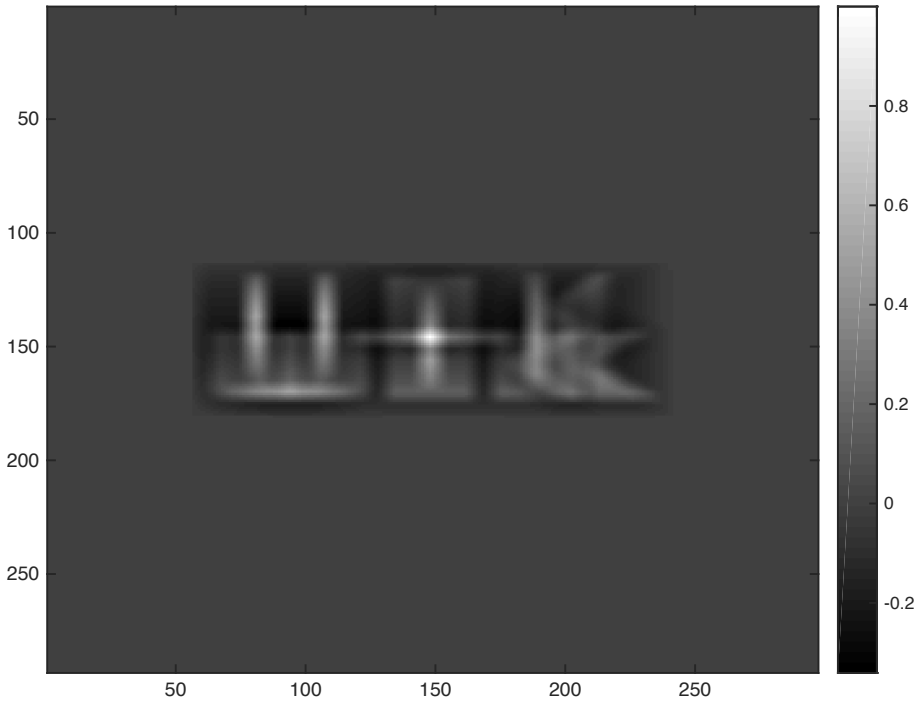
This is not true any image. For those close contrast images, whose histograms are very concentrated, HE is very useful.

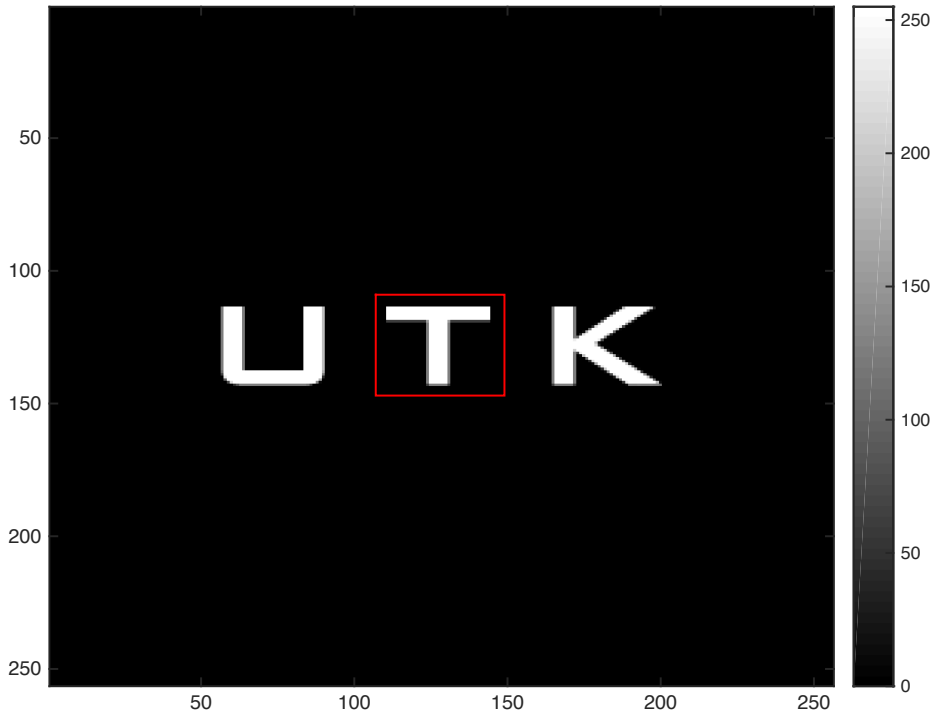
Problem 3
(i). convolution:





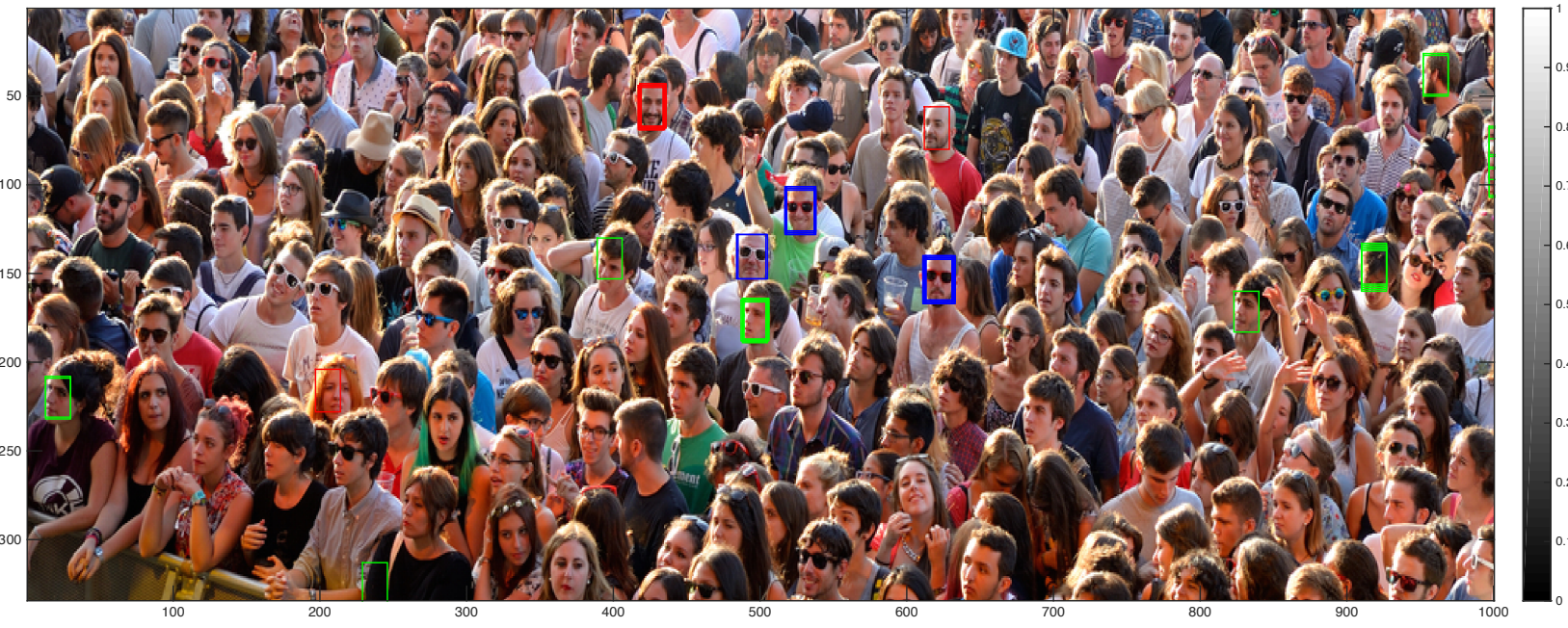
(ii). NCC:


50
100
150
200
250
0.8
0.6
0.4
0.2
0
-0.2


U T K
50
100
150
200
250
250
200
150
100
50
0

(iii). the threshold I chose is 0.59

(iv). the threshold for template1 is 0.59(red), template2 is 0.75(green), template3 is 0.74(blue)



Appendix:

Problem1:

```
function Gmag = HW4_pro1(filename,threshold)

geisel = imread(filename);
geisel = double(rgb2gray(geisel));

k_gaussian = 1/159 * [2 4 5 4 2;4 9 12 9 4;5 12 15 12 5;4 9 12 9 4;2 4 5 4 2];
smoothImage = conv2(geisel,k_gaussian,'same');
figure
imshow(uint8(smoothImage));
title('smoothImage')
hold on



kx = [-1 0 1;-2 0 2;-1 0 1];
ky = [-1 -2 -1;0 0 0;1 2 1];
Gx = conv2(smoothImage,kx,'same');
Gy = conv2(smoothImage,ky,'same');
Gmag = sqrt(Gx.^2+Gy.^2);
Gtheta = atan2d(Gy,Gx);
Gtheta = round(Gtheta./45)*45;

figure
imshow(uint8(Gmag));
title('gradient magnitude iamge')
hold on

[height,width]=size(Gmag); %h=476,w=640
for i=2:height-1
    for j=2:width-1
        if Gtheta(i,j)==0 || Gtheta(i,j)==180 || Gtheta(i,j)==-180
            W = Gmag(i,j-1);
            E = Gmag(i,j+1);
            if Gmag(i,j)<=W||Gmag(i,j)<=E  %Gmag(i,j)<max(W,E)
                Gmag(i,j)=0;

            end
        end
        if Gtheta(i,j)==45 || Gtheta(i,j)==-135
            SW = Gmag(i+1,j-1);
            NE = Gmag(i-1,j+1);
            if Gmag(i,j)<=SW || Gmag(i,j)<=NE
                Gmag(i,j)=0;
            end
        end
        if Gtheta(i,j)==135||Gtheta(i,j)==-45
            NW = Gmag(i-1,j-1);
            SE = Gmag(i+1,j+1);
```

```
        if Gmag(i,j)<=NW ||Gmag(i,j)<=SE
            Gmag(i,j)=0;
        end

    end
    if Gtheta(i,j)==90||Gtheta(i,j)==-90
        N = Gmag(i-1,j);
        S = Gmag(i+1,j);
        if Gmag(i,j)<=S ||Gmag(i,j)<=N
            Gmag(i,j)=0;
        end
    end

  end
end

Gmag = Gmag.*(Gmag>threshold);
figure
imshow(uint8(Gmag))
title('non-maximumn supression image with threshold')
```

Problem2:

```
function newImage = HW4_pro2(filename,win_size)


beach = imread(filename);
im = double(beach);
newImage = im;
%this part simply using built in function histeq() to get the HE image
figure
 im2 = imread('beach.png');
 im2 = histeq(im2);
 imshow(im2)
title('HE with built in function')

hold on
pad_size = round(win_size/2);
[h,w]=size(im);

im = padarray(im,[pad_size pad_size],'symmetric','both');

for x= pad_size :h+pad_size
   for y= pad_size :w+pad_size
       rank = 0;
        region = im(x-(round(win_size/2)-1):x+(round(win_size/2)-1),y-(round(win_size/2)-1):y+(round(win_size/
2)-1));
        largerPixel = im(x,y)>region;
        rank = sum(sum(largerPixel));
```

```
        newImage(x,y) = rank*255/(win_size*win_size);
    end
end
newImage = newImage(pad_size:h+pad_size,pad_size:w+pad_size);
figure
imshow(uint8(newImage))
```

Problem3:
(i).

```
letters = imread('Letters.jpg');
template = imread('LettersTemplate.jpg');

letters = double(letters);
template = double(template);

original = letters;
[temph,tempw] = size(template);

letters = flip(letters,1);
letters = flip(letters,2);


convolvedIm = conv2(letters,template,'same');

figure
imagesc(convolvedIm);
colormap gray
colorbar
hold on



maxVal = max(convolvedIm(:));
[maxr,maxc] = find(convolvedIm==maxVal);

figure
imagesc(original)
colormap gray;
colorbar
```

```
rectangle('Position',[maxc-tempw/2-1 maxr-temph/2-1 tempw temph],'EdgeColor','r');
```

(ii).

```
letters = imread('Letters.jpg');
template = imread('LettersTemplate.jpg');

letters = double(letters);
template = double(template);

original = letters;
[temph,tempw] = size(template);



corrIm = normxcorr2(template,letters);

figure
imagesc(corrIm)
colormap gray
colorbar
hold on

maxVal = max(corrIm(:));
[maxr,maxc] = find(corrIm==maxVal);

figure
imagesc(original)
colormap gray;
colorbar
rectangle('Position',[maxc-size(template,2)+1 maxr-size(template,1)+1 tempw temph],'EdgeColor','r');
```

(iii).

```
crowd = imread('crowd.jpg');
template = imread('face1.jpeg');
original = crowd;
crowd = double(rgb2gray(crowd));
template = double(rgb2gray(template));


[temph,tempw] = size(template);


corrIm = normxcorr2(template,crowd);

maxVal = 0.59* max(corrIm(:));
[maxr,maxc] = find(corrIm>maxVal);

imagesc(original)
colormap gray; hold on
```

```
colorbar
for i = 1:numel(maxc)
rectangle('Position',[maxc(i)-size(template,2)+1 maxr(i)-size(template,1)+1 tempw temph],'EdgeColor','r');
end
```

(iv).

```
crowd = imread('crowd.jpg');
template = imread('face1.jpeg');
template2 = imread('face2.jpeg');
template3 = imread('face3.jpeg');
original = crowd;
crowd = double(rgb2gray(crowd));
template = double(rgb2gray(template));
template2 = double(rgb2gray(template2));
template3 = double(rgb2gray(template3));

[temph,tempw] = size(template);
[temph2,tempw2] = size(template2);
[temph3,tempw3] = size(template3);

corrIm = normxcorr2(template,crowd);
corrIm2 = normxcorr2(template2,crowd);
corrIm3 = normxcorr2(template3,crowd);

threshold = 0.59* max(corrIm(:));
threshold2 = 0.75* max(corrIm2(:));
threshold3 = 0.74* max(corrIm3(:));

[maxr,maxc] = find(corrIm>threshold);
[maxr2,maxc2] = find(corrIm2>threshold2);
[maxr3,maxc3] = find(corrIm3>threshold3);


imagesc(original)
colormap gray;hold on
colorbar

for i = 1:numel(maxc)
rectangle('Position',[maxc(i)-size(template,2)+1 maxr(i)-size(template,1)+1 tempw temph],'EdgeColor','r');
end

for j = 1:numel(maxc2)
rectangle('Position',[maxc2(j)-size(template2,2)+1 maxr2(j)-size(template2,1)+1 tempw2
temph2],'EdgeColor','g');
end

for k = 1:numel(maxc3)
rectangle('Position',[maxc3(k)-size(template3,2)+1 maxr3(k)-size(template3,1)+1 tempw3
temph3],'EdgeColor','b');
end
```