ECE 271A Statistical Learning HW2

a) In problem 2, we get the result of

$$P_k = \pi_k = \frac{C_j}{n}$$

Which means the relatives frequency of the two events. So we get,

P_Y(cheetah) = 250 / (250 + 1053) = 0.1919

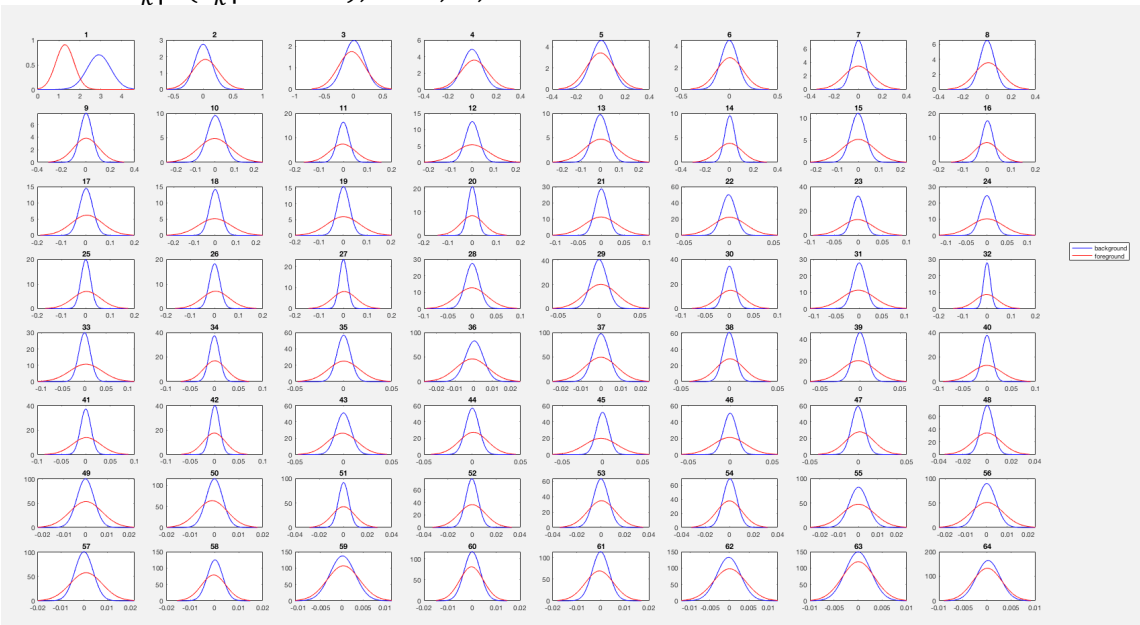
P_Y(grass) = 1053 / (250 + 1053) = 0.8081

Compare to the result in last week's assignment, they are the same.

P_Y(cheetah) = $\frac{\#\ of\ Cheetah\ samples}{\#\ of\ Cheetah\ samples + \#\ of\ grass\ samples}$ = 0.1919

P_Y(grass) = $\frac{\#\ of\ Cheetah\ samples}{\#\ of\ Cheetah\ samples + \#\ of\ grass\ samples}$ = 0.8081

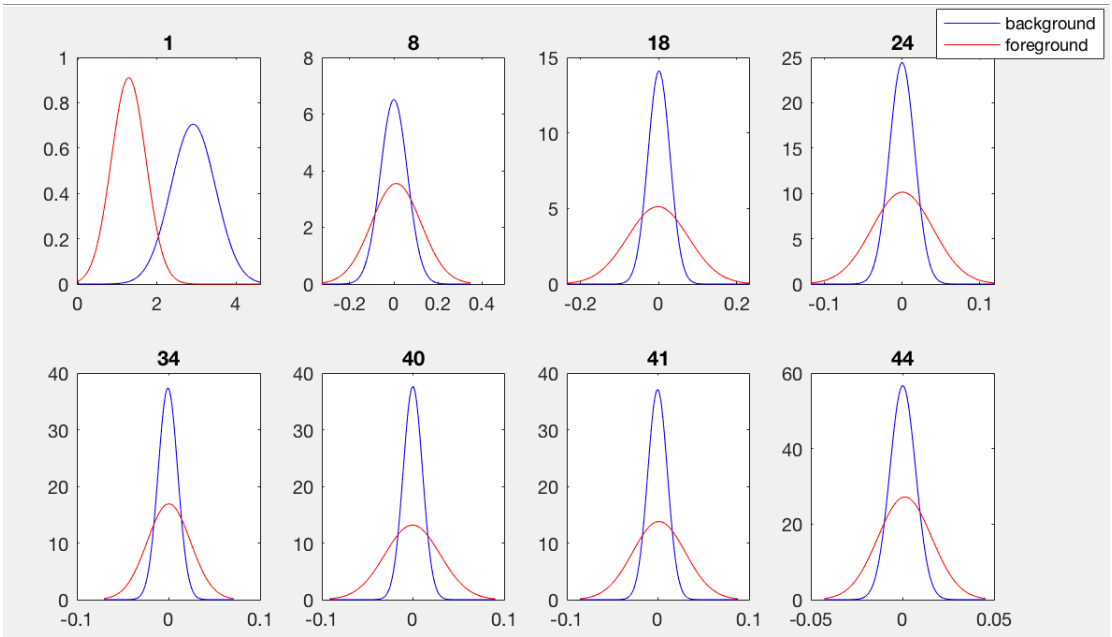
b) All 64 plots with the marginal densities for the two classes $PX_k|Y(x_k|grass)$ & $PX_k|Y(x_k|cheetah)$, k = 1, ..., 64



Blue Lines: $PX_k|Y(x_k|grass)$

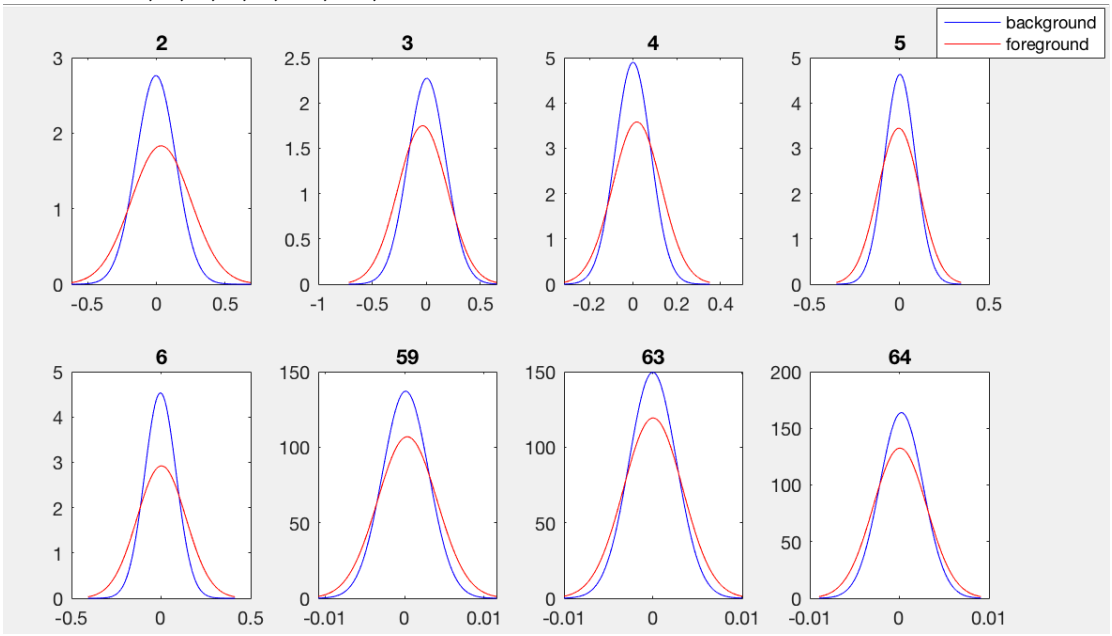
Red Lines: $PX_k|Y(x_k|cheetah)$

By visual inspection, best 8 plots with the marginal densities for the two classes $PX_k|Y(x_k|grass)$ & $PX_k|Y(x_k|cheetah)$, k = 1, …, 64
Number: 1, 8, 18, 24, 34, 40, 41, 44



I think these 8 is the best is because the two lines in the plot do not overlap too much that we see the mean and sigma are different. This means those feature can classify two classes really well.

By visual inspection, worst 8 plots with the marginal densities for the two classes $PX_k|Y(x_k|grass)$ & $PX_k|Y(x_k|cheetah)$, k = 1, …, 64
Number: 2, 3, 4, 5, 6, 59, 63, 64



I think these 8 is the worst is because the two lines in the plot overlap too much that we see the mean and sigma are mostly same. This means those feature cannot classify two classes really well.

c) In this problem we need to get the picture of the result that the computer learned to classify the background and the foreground.
As we know if $P_{Y|X}(\text{cheetah}|x) > P_{Y|X}(\text{grass}|x)$, the computer will classify the pixel as the "cheetah". However, we cannot compute $P_{Y|X}(\text{cheetah}|x)$ and $P_{Y|X}(\text{grass}|x)$ easily, so we use the Bayes rule to turn the problem into other variables that can easily be computed.
By Bayes rule:

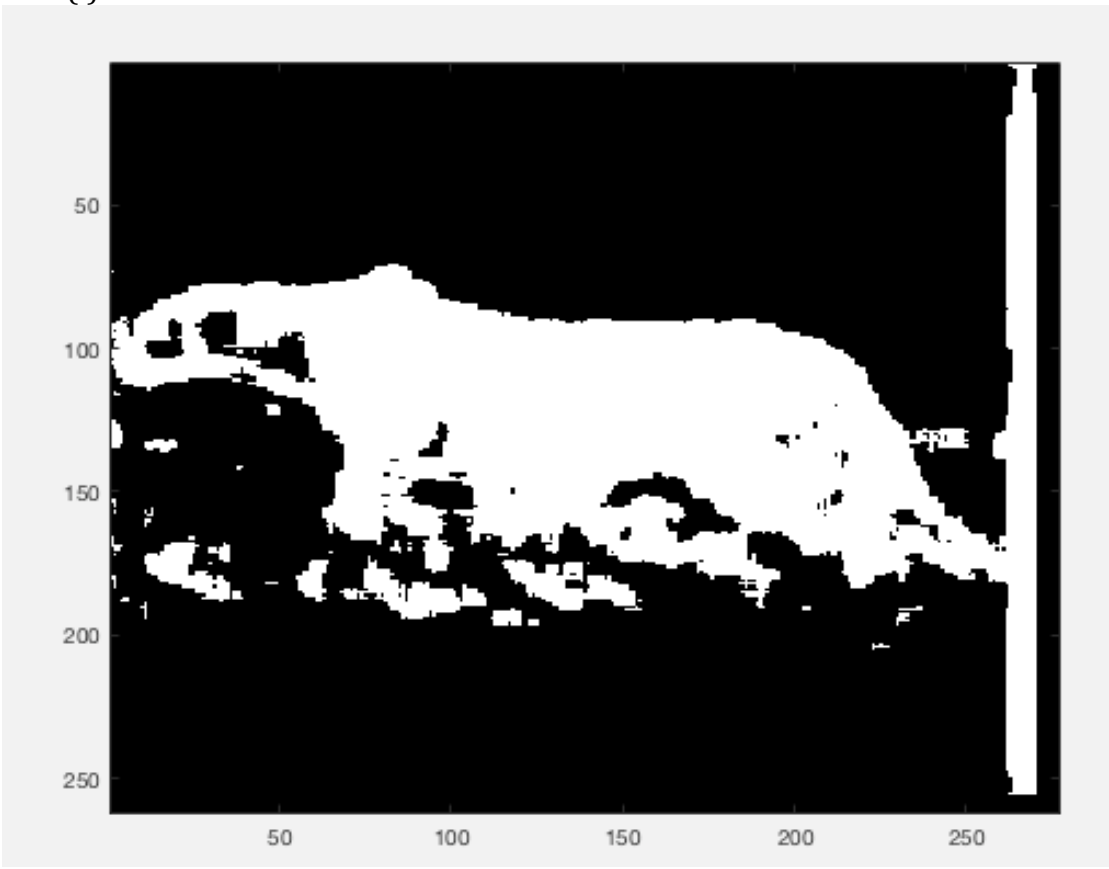
$$P_{Y|X}(\text{cheetah}|x) = \frac{P_{X|Y}(x|\text{cheetah}) * P_Y(\text{cheetah})}{P_X(x)}$$

So we if we want to compute $P_{Y|X}(\text{cheetah}|x) > P_{Y|X}(\text{grass}|x)$
We can rewrite it as below:

$$\frac{P_{X|Y}(x|\text{cheetah}) * P_Y(\text{cheetah})}{P_X(x)} > \frac{P_{X|Y}(x|\text{grass}) * P_Y(\text{grass})}{P_X(x)}$$

In this case, we can know that this pixel should be cheeta and set the mask value as 1.

(i) For 64-dimensional Gaussians



Probability of Error:

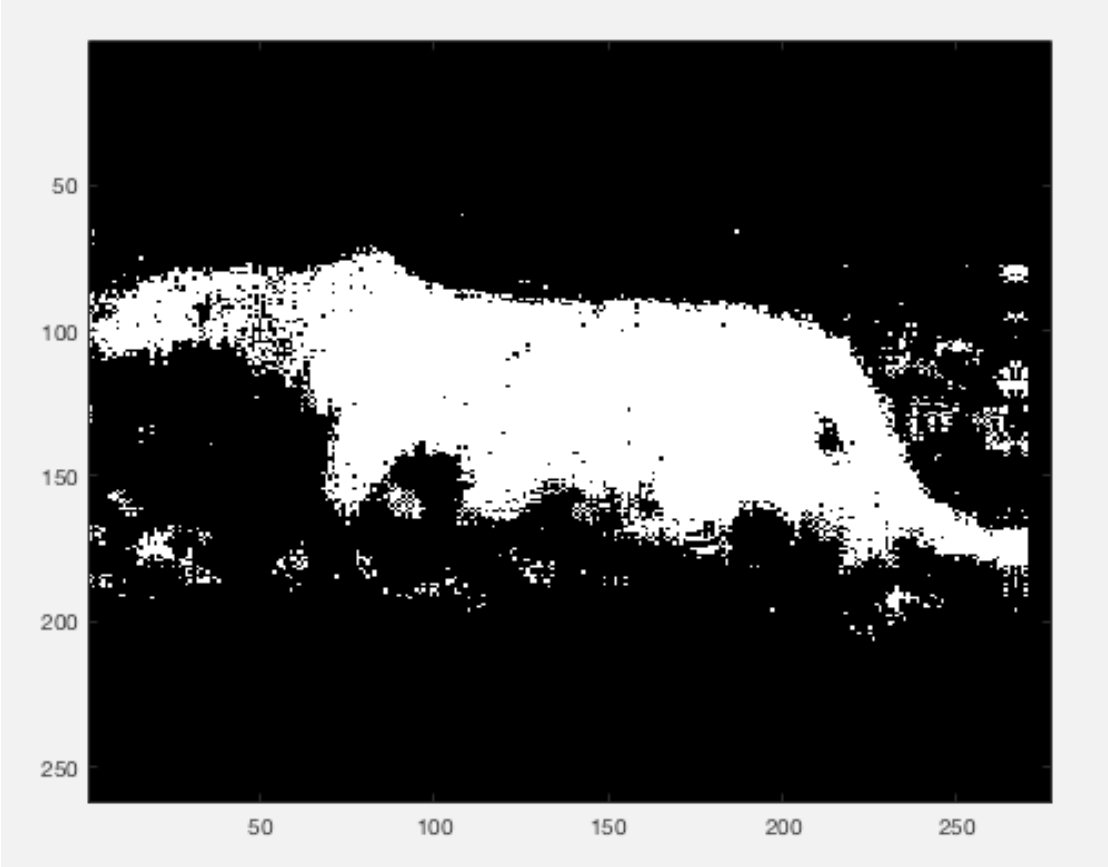
$$\text{error(FG)} = \frac{\#\text{FG pixels misclassified as BG}}{\#FG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ FG$$

Error of ground truth Foreground misclassified as Background,
its rate is 9.6 %.

$$\text{error(BG)} = \frac{\#\text{BG pixels misclassified as FG}}{\#BG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ BG$$

Error of ground truth Background misclassified as Foreground,
its rate is 1.56 %
The total error rate of 64-dimensional Gaussians (All 64 Features) is 9.6 % + 1.56 % = 11.16%.

(ii) For 8-dimensional Gaussians (best 8 features)



Probability of Error:

$$\text{error(FG)} = \frac{\text{\#FG pixels misclassified as BG}}{\#FG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ FG$$

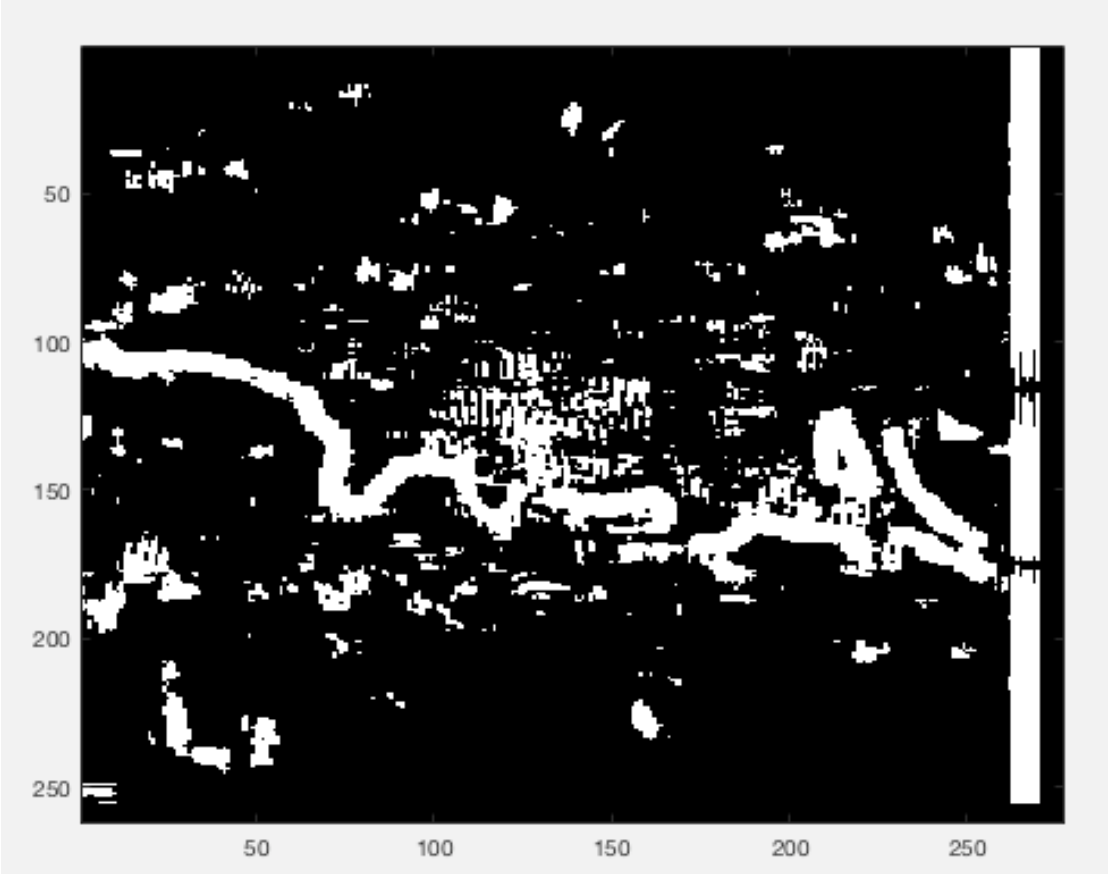
Error of ground truth Foreground misclassified as Background,
its rate is 3.64 %.

$$\text{error(BG)} = \frac{\text{\#BG pixels misclassified as FG}}{\#BG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ BG$$

Error of ground truth Background misclassified as Foreground,
its rate is 1.88 %

The total error rate of 64-dimensional Gaussians (All 64 Features) is 3.64 % + 1.88 % = 5.52 %.

Follow up:
For 8-dimensional Gaussians (worst 8 features)



Probability of Error:

$$\text{error(FG)} = \frac{\text{\#FG pixels misclassified as BG}}{\#FG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ FG$$

Error of ground truth Foreground misclassified as Background,
its rate is 7.70 %.

$$\text{error(BG)} = \frac{\text{\#BG pixels misclassified as FG}}{\#BG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ BG$$

Error of ground truth Background misclassified as Foreground,
its rate is 12.89 %

The total error rate of 64-dimensional Gaussians (All 64 Features) is 7.70 % + 12.89 % = 20.59 %.

Results Explanation:

Although the result of using Gaussian Classifier instead of taking out the 2nd largest coefficient from the DCT matrix is way more better, we still can exclude some bad classification result from the plot we did in (b) and only take 8 features that can easily classify the two classes. By taking out the best 8 features by observation, we can lower the error rate.

Code:

```
clear all;
%% Compute the priors Py(cheeta) & Py(grass)
load('TrainingSamplesDCT_8_new.mat');
p_background = size(TrainsampleDCT_BG,1) / (size(TrainsampleDCT_BG,1)
+ size(TrainsampleDCT_FG,1)); % PY(grass)
p_foreground = 1 - p_background; %PY(cheeta)

%% plot the 64 features in Guassian
mean_background = mean(TrainsampleDCT_BG);
sd_background = sqrt(var(TrainsampleDCT_BG));
mean_foreground = mean(TrainsampleDCT_FG);
sd_foreground = sqrt(var(TrainsampleDCT_FG));

figure(1);
for i = 1:64
    x_min = min(mean_background(1,i) - 3*sd_background(1,i),
mean_foreground(1,i) - 3*sd_foreground(1,i));
    x_max = max(mean_background(1,i) + 3*sd_background(1,i),
mean_foreground(1,i) + 3*sd_foreground(1,i));
    x = linspace(x_min,x_max);
    norm_background = normpdf(x, mean_background(1,i),
sd_background(1,i));
    norm_foreground = normpdf(x, mean_foreground(1,i),
sd_foreground(1,i));

    subplot(8,8,i);
    plot(x, norm_background,'B', x, norm_foreground,'R');
    title(i);
    hold on;
end
legend('background','foreground');



%% Best 8 features (1,8,18,24,34,40,41,44)
figure(2);
best = 1;
```

```
best_features = [1,8,18,24,34,40,41,44];
for i = best_features
    x_min = min(mean_background(1,i) - 3*sd_background(1,i),
mean_foreground(1,i) - 3*sd_foreground(1,i));
    x_max = max(mean_background(1,i) + 3*sd_background(1,i),
mean_foreground(1,i) + 3*sd_foreground(1,i));
    x = linspace(x_min,x_max);
    norm_background = normpdf(x, mean_background(1,i),
sd_background(1,i));
    norm_foreground = normpdf(x, mean_foreground(1,i),
sd_foreground(1,i));

    subplot(2,4,best);
    best = best + 1;
    plot(x, norm_background,'B', x, norm_foreground,'R');
    title(i);
    hold on;
end
legend('background','foreground');

%% Worst 8 features (2,3,4,5,6,59,63,64)
figure(3);
worst = 1;
worst_features = [2,3,4,5,6,59,63,64];
for i = worst_features
    x_min = min(mean_background(1,i) - 3*sd_background(1,i),
mean_foreground(1,i) - 3*sd_foreground(1,i));
    x_max = max(mean_background(1,i) + 3*sd_background(1,i),
mean_foreground(1,i) + 3*sd_foreground(1,i));
    x = linspace(x_min,x_max);
    norm_background = normpdf(x, mean_background(1,i),
sd_background(1,i));
    norm_foreground = normpdf(x, mean_foreground(1,i),
sd_foreground(1,i));

    subplot(2,4,worst);
    worst = worst + 1;
    plot(x, norm_background,'B', x, norm_foreground,'R');
```

```
    title(i);
    hold on;
end
legend('background','foreground');

%If Py|x(cheeta|x) > Py|x(grass|) ----> the pixel is cheeta
%same as Px|y(x|cheeta)Py(cheeta) > Px|y(x|grass)Py(grass)
%input bmp file & ZigZag pattern
A = im2double(imread('cheetah.bmp'));
Z = load('Zig-Zag Pattern.txt');

%% original size 255*270 ---> need to use padarray() to fill 255+7,
270+7
B = padarray(A,[7,7],'symmetric','post');
[q,l] = size(B);
matrix_zigzag=[];
mask = zeros(q,l);

%% make the picture into 8*8 pixels and use mask determine which
pixel is cheeta/grass
var_background = var(TrainsampleDCT_BG);
var_foreground = var(TrainsampleDCT_FG);

for i=1:q-7
    for j=1:l-7
        matrix_dct2 = dct2(B(i:i+7,j:j+7));
        matrix_zigzag(Z+ones(8,8)) = matrix_dct2; %store matrix_dct
into zigzag which is from 0~63 so plus a ones(8,8)
%         [V,I] = max(abs(matrix_zigzag(2:64)));
%%         compute class-conditioned Px|y(x|cheeta)
%         index_cheeta = zeros(1,64);
%         TrainsampleDCT_FG(:,1) = [];
%         for ci = 1: size(TrainsampleDCT_FG,1)
%             [VC,C] = max(abs(TrainsampleDCT_FG(ci,:)));
%             index_cheeta(C) = index_cheeta(C) + 1;
%         end
%         index_cheeta = index_cheeta / size(TrainsampleDCT_FG,1);
```

```
%%        compute class-conditioned Px|y(x|grass)
%        index_grass = zeros(1,64);
%        TrainsampleDCT_BG(:,1) = [];
%        for gi = 1: size(TrainsampleDCT_BG,1)
%            [VG,G] = max(abs(TrainsampleDCT_BG(gi,:)));
%            index_grass(G) = index_grass(G) + 1;
%        end
%        index_grass = index_grass / size(TrainsampleDCT_BG,1);

%% index_grass ----> Pxy_xgrass; index_cheeta -----> Pxy_xcheeta
   %Compute class-conditioned Px|y(x|cheeta) & compute class-
conditioned Px|y(x|grass)
       Pxy_xgrass= mvnpdf(matrix_zigzag, mean_background,
var_background);
       Pxy_xcheeta = mvnpdf(matrix_zigzag, mean_foreground,
var_foreground);
       if (Pxy_xcheeta*p_foreground) >  (Pxy_xgrass*p_background)
           mask(i,j) = 1;
       else% Pxy_xcheeta*p_foreground < Pxy_xgrass*p_background
           mask(i,j) = 0;
       end
    end
end
figure(4);
imagesc(mask);
colormap(gray(255));

%% Error Rate of all 64 features
diff_grass = 0;
diff_cheeta = 0;
count_grass = 0;
count_cheeta = 0;

groundtruth_mask = im2double(imread('cheetah_mask.bmp'));
for i=1:(q-7)
    for j=1:(l-7)
        if mask(i,j) == 1
            if groundtruth_mask(i,j) == 0
```

```
            diff_cheeta = diff_cheeta + 1;
        else
            diff_cheeta = diff_cheeta;
        end
    elseif mask(i,j) == 0
        if groundtruth_mask(i,j) == 1
            diff_grass = diff_grass + 1;
        else
            diff_grass - diff_grass;
        end
    end
 end
end

for i=1:(q-7)
    for j=1:(l-7)
        if groundtruth_mask(i,j) == 1
            count_cheeta = count_cheeta + 1;
        else
            count_grass = count_grass + 1;
        end
    end
end
error_cheeta64 = (diff_cheeta / count_cheeta)*p_foreground ;
error_grass64 = (diff_grass / count_grass)*p_background;
error_rate64 = error_cheeta64 + error_grass64;


%% Best Features
for i=1:q-7
    for j=1:l-7
        matrix_dct2 = dct2(B(i:i+7,j:j+7));
        matrix_zigzag(Z+ones(8,8)) = matrix_dct2; %store matrix_dct
into zigzag which is from 0~63 so plus a ones(8,8)
%% index_grass ----> Pxy_xgrass; index_cheeta -----> Pxy_xcheeta
    %Compute class-conditioned Px|y(x|cheeta) & compute class-
conditioned Px|y(x|grass)
        Pxy_xgrass = mvnpdf(matrix_zigzag(best_features),
```

```
mean_background(best_features), var_background(best_features));
       Pxy_xcheeta = mvnpdf(matrix_zigzag(best_features),
mean_foreground(best_features), var_foreground(best_features));
       if (Pxy_xcheeta*p_foreground) >  (Pxy_xgrass*p_background)
          mask(i,j) = 1;
       else% Pxy_xcheeta*p_foreground < Pxy_xgrass*p_background
          mask(i,j) = 0;
       end
    end
end
figure(5);
imagesc(mask);
colormap(gray(255));

%% Error Rate of 8 best features
diff_grass = 0;
diff_cheeta = 0;
count_grass = 0;
count_cheeta = 0;

groundtruth_mask = im2double(imread('cheetah_mask.bmp'));
for i=1:(q-7)
    for j=1:(l-7)
       if mask(i,j) == 1
          if groundtruth_mask(i,j) == 0
             diff_cheeta = diff_cheeta + 1;
          else
             diff_cheeta = diff_cheeta;
          end
       elseif mask(i,j) == 0
          if groundtruth_mask(i,j) == 1
             diff_grass = diff_grass + 1;
          else
             diff_grass - diff_grass;
          end
       end
     end
end
```

```
for i=1:(q-7)
    for j=1:(l-7)
        if groundtruth_mask(i,j) == 1
            count_cheeta = count_cheeta + 1;
        else
            count_grass = count_grass + 1;
        end
    end
end
error_cheetab = (diff_cheeta / count_cheeta)*p_foreground ;
error_grassb = (diff_grass / count_grass)*p_background;
error_rateb = error_cheetab + error_grassb;

%% Worst Features
for i=1:q-7
    for j=1:l-7
        matrix_dct2 = dct2(B(i:i+7,j:j+7));
        matrix_zigzag(Z+ones(8,8)) = matrix_dct2; %store matrix_dct
into zigzag which is from 0~63 so plus a ones(8,8)
%% index_grass ----> Pxy_xgrass; index_cheeta -----> Pxy_xcheeta
    %Compute class-conditioned Px|y(x|cheeta) & compute class-
conditioned Px|y(x|grass)
        Pxy_xgrass = mvnpdf(matrix_zigzag(worst_features),
mean_background(worst_features), var_background(worst_features));
        Pxy_xcheeta = mvnpdf(matrix_zigzag(worst_features),
mean_foreground(worst_features), var_foreground(worst_features));
        if (Pxy_xcheeta*p_foreground) >  (Pxy_xgrass*p_background)
            mask(i,j) = 1;
        else% Pxy_xcheeta*p_foreground < Pxy_xgrass*p_background
            mask(i,j) = 0;
        end
    end
end
figure(6);
imagesc(mask);
colormap(gray(255));
```

```
%% Error Rate of 8 worst features
diff_grass = 0;
diff_cheeta = 0;
count_grass = 0;
count_cheeta = 0;

groundtruth_mask = im2double(imread('cheetah_mask.bmp'));
for i=1:(q-7)
    for j=1:(l-7)
        if mask(i,j) == 1
            if groundtruth_mask(i,j) == 0
                diff_cheeta = diff_cheeta + 1;
            else
                diff_cheeta = diff_cheeta;
            end
        elseif mask(i,j) == 0
            if groundtruth_mask(i,j) == 1
                diff_grass = diff_grass + 1;
            else
                diff_grass - diff_grass;
            end
        end
     end
end

for i=1:(q-7)
    for j=1:(l-7)
        if groundtruth_mask(i,j) == 1
            count_cheeta = count_cheeta + 1;
        else
            count_grass = count_grass + 1;
        end
    end
end
error_cheetaw = (diff_cheeta / count_cheeta)*p_foreground ;
error_grassw = (diff_grass / count_grass)*p_background;
error_ratew = error_cheetaw + error_grassw;
```