ECE 271A: Statistical Learning I

Homework 3

Name	:	Sanchit Gupta
PID	:	A59010276
Date of Submission	:	14th November 2022

Solution for Question 4

The parameters required for performing predictive distribution, MAP and MLE procedures are described in the steps below:

a) The maximum likelihood estimate for the prior probabilities is calculated by taking a ratio of the length of foreground and background training data with the total length of the training data respectively. Foreground corresponds to cheetah and background corresponds to grass.
   - P_Y (cheetah) = Prior probability for foreground
   - P_Y (grass) = Prior probability for background

b) The ML estimates of mean and covariance are obtained from the given training data sets.
   - $\mu_{ML} = ML\ estimate\ of\ mean$
   - $\sigma^2 = ML\ estimate\ of\ covariance$

c) The Gaussian prior for mean μ_0 is provided in the file Prior_1.mat for strategy 1 and in file Prior2.mat for strategy 2. The Gaussian prior for covariance Σ_0 is calculated using the weights w_i and values of α provided using the equation shown below:

$$(\Sigma_0)_{ii} = \alpha w_i$$

d) The posterior mean μ_1 and covariance Σ_1 of

$$P_{(\mu|T)}(\mu \mid D_1) = G(\mu, \mu_1, \Sigma_1)$$

are calculated using the following equations:

$$\mu_n = \frac{n\sigma_0^2}{\sigma^2 + n\sigma_0^2}\mu_{ML} + \frac{\sigma^2}{\sigma^2 + n\sigma_0^2}\mu_0$$

$$\frac{1}{\sigma_n^2} = \frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}$$

e) The parameters of predictive distribution

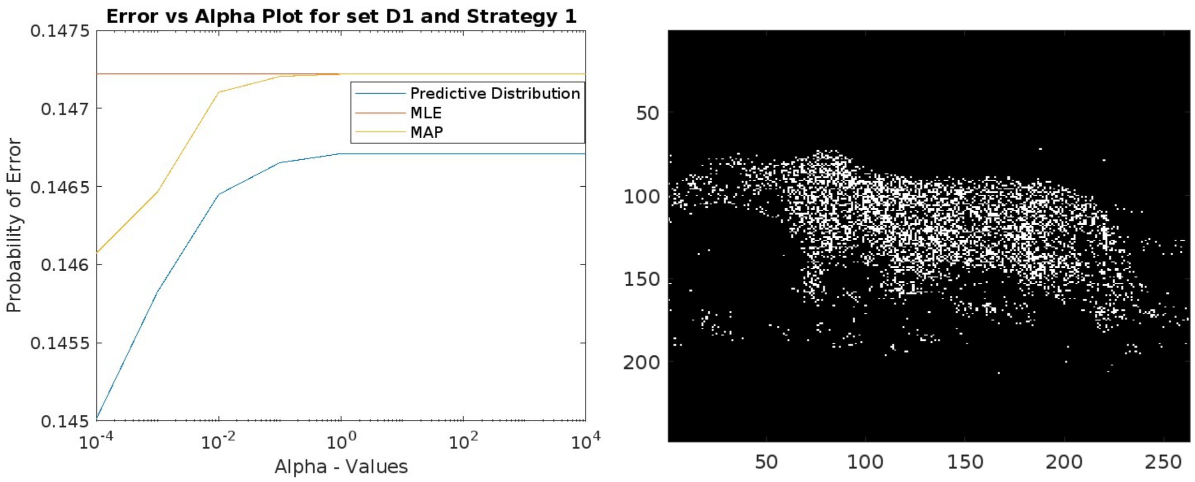
$$P_{(x|T)}(x|D_1)$$

are calculated using the following equation

$$P_{Bayes} = P_{ML} + P_{Prior}$$

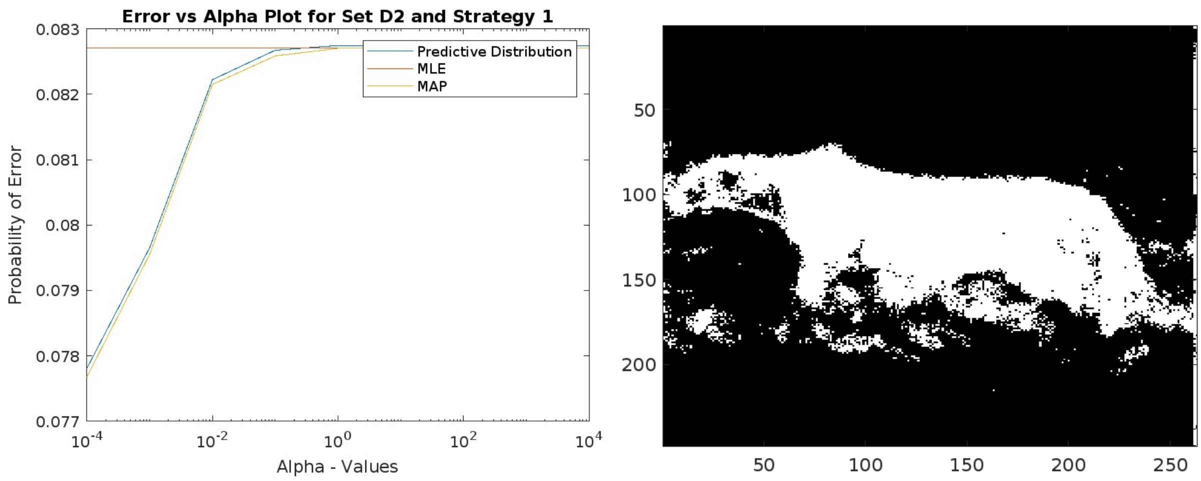
$$\sigma_{Bayes}^2 = \sigma_{ML}^2 + \sigma_n^2$$

These parameters are plugged into the Bayesian decision rule to classify the cheetah image using (i) predictive distribution, (ii) MLE estimate and (iii) MAP estimate procedures. Following classification of image, the probability of error is measured with respect to ground truth mask for each of the three procedures. The plots for probability of error vs values of α showing comparison between the three procedures for datasets $D_i, i = 1, 2, 3, 4$ using strategy 1 are shown in Figures 1 to 4. The mask of cheetah image obtained using predictive distribution procedure for each of the four datasets is also shown along with the plots. The mask corresponds to the one obtained using value of α with least probability of error.



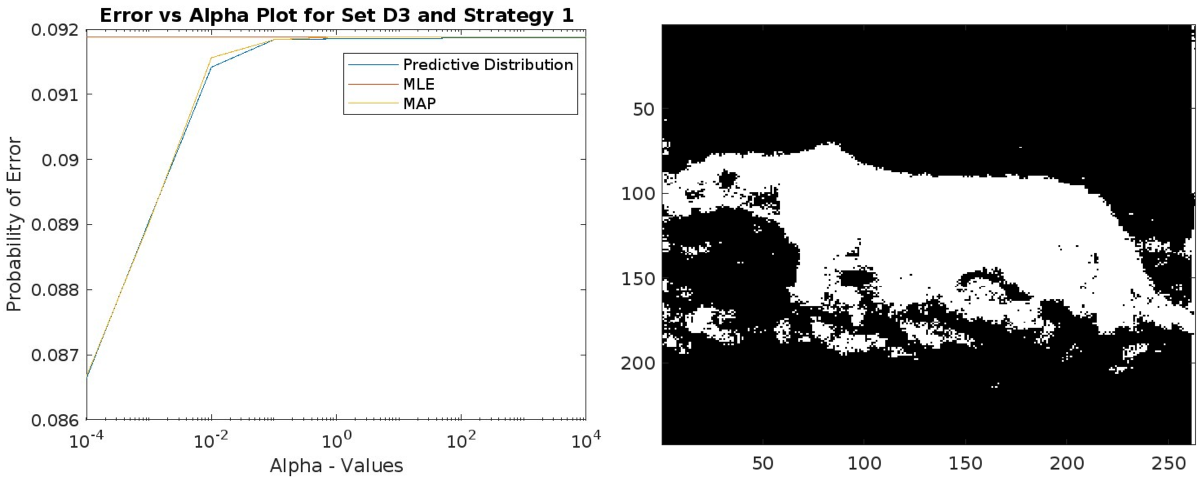
(a) (b)

Figure 1: (a) Plot for probability of error vs values of α for dataset D1 using strategy 1, (b) Mask of the cheetah with least probability of error



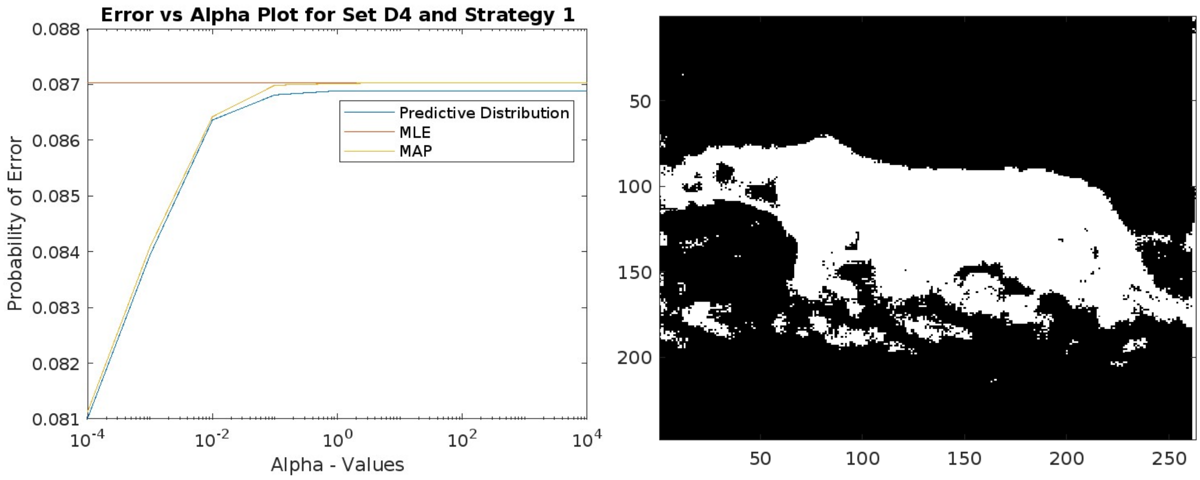
(a) (b)

Figure 2: (a) Plot for probability of error vs values of α for dataset D2 using strategy 1, (b) Mask of the cheetah with least probability of error



(a) (b)

Figure 3: (a) Plot for probability of error vs values of α for dataset D3 using strategy 1, (b) Mask of the cheetah with least probability of error



(a) (b)

Figure 4: (a) Plot for probability of error vs values of α for dataset D4 using strategy 1, (b) Mask of the cheetah with least probability of error

From the plots shown in Figures 1 to 4, it can be seen that the probability of error increases as the value of α increases. The Gaussian prior for covariance is a function of α and hence, as α increases, the prior becomes unreliable. An unreliable prior does not perform good in classification of the image, leading to higher probability of error. The prior calculated using lower values of α help in better classification of the image with lower probability of error. It is also seen from the plots that as α increases, the probability of error for the predictive distribution procedure converges to that of MLE solution.

In the case of MLE procedure, the mean and covariance estimates are not a function of α, thus, the classification of the cheetah image does not vary with the varying values of α. The probability of error in this case remains constant. Thus, a straight-line plot is obtained for MLE solution for each of the four datasets.

In the case of MAP procedure, the mean estimate is same as that of predictive distribution procedure, whereas the covariance estimate is the same as MLE procedure. Thus, for MAP procedure as well, the probability of error increases as the values of α increases.

The general trend between the probability of errors for three approaches using Strategy 1 in all the four datasets is as follows:

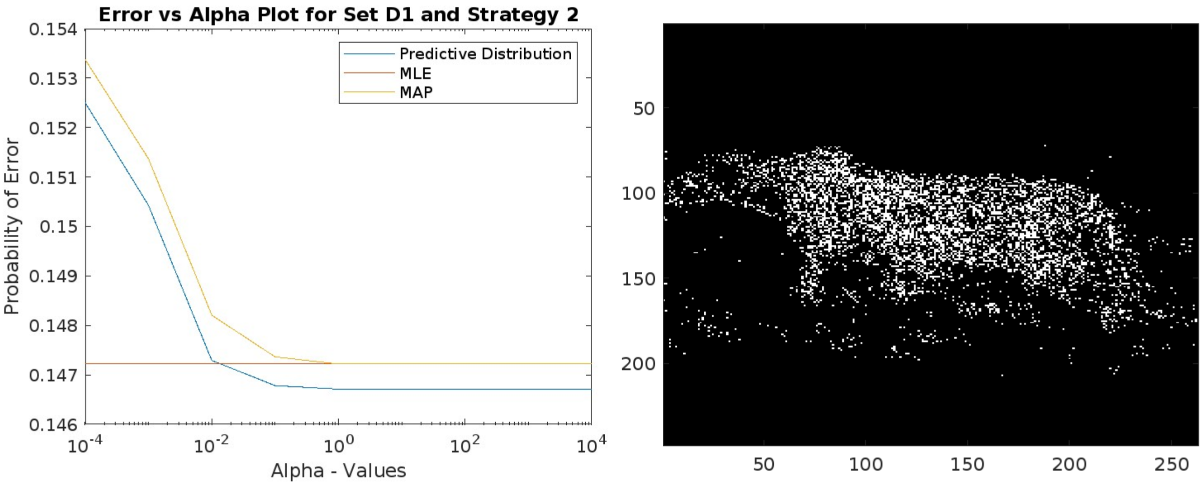
Error of Predictive Distribution < Error of MAP Solution < Error of MLE Solution

Further, the trend for probability of error between four datasets for predictive distribution solution is as follows. Even though the size of dataset increases from D1 to D4, the trend seen in the probability of error values is different in comparison to the size of datasets.

Error in D2 < Error in D4 < Error in D3 < Error in D1

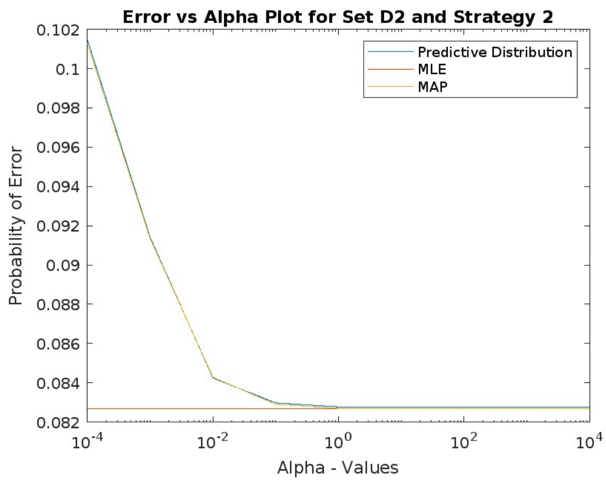
Shifting to Strategy 2

The plots for probability of error vs values of α showing comparison between the three procedures for datasets $D_i, i = 1, 2, 3, 4$ using strategy 2 are shown in Figures 5 to 8. The mask of cheetah image obtained using predictive distribution procedure for each of the four datasets is also shown along with the plots. The mask corresponds to the one obtained using value of α with least probability of error.

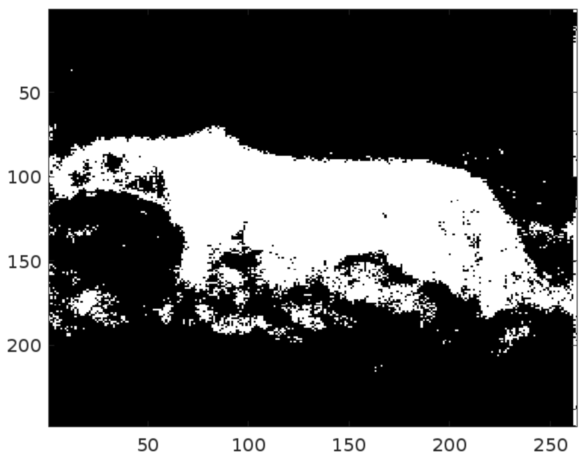


(a) (b)

Figure 5: (a) Plot for probability of error vs values of α for dataset D1 using strategy 2, (b) Mask of the cheetah with least probability of error

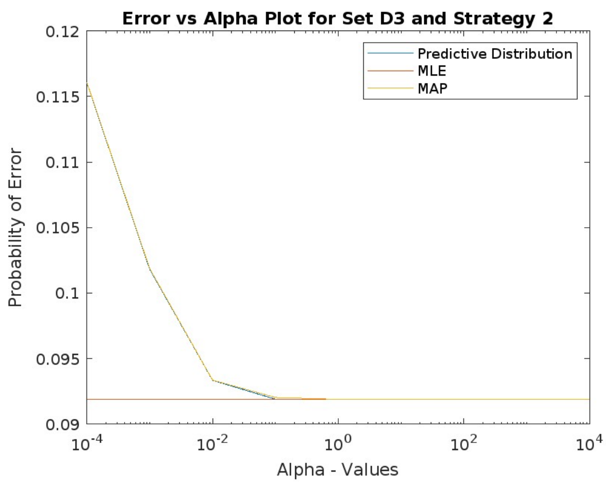


(a)

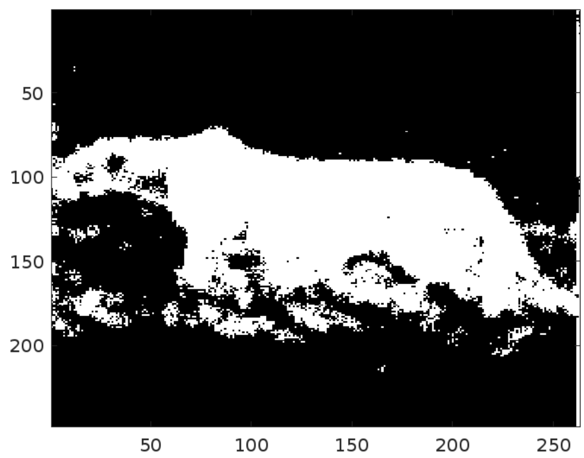


(b)

Figure 6: (a) Plot for probability of error vs values of α for dataset D2 using strategy 2, (b) Mask of the cheetah with least probability of error

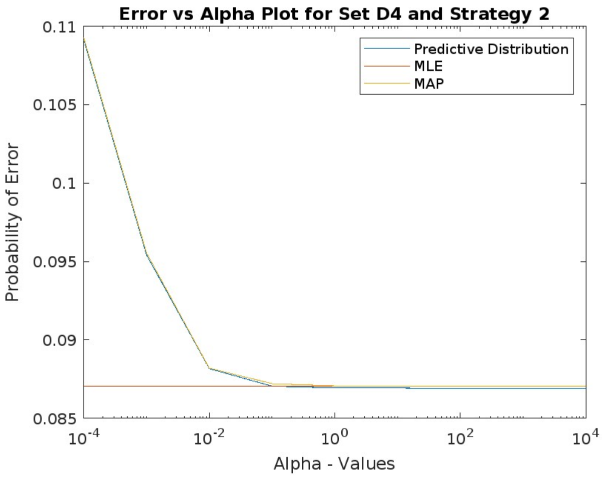


(a)

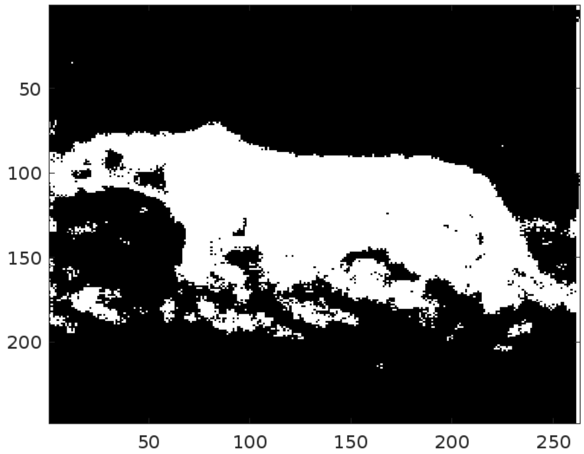


(b)

Figure 7: (a) Plot for probability of error vs values of α for dataset D3 using strategy 2, (b) Mask of the cheetah with least probability of error



(a)



(b)

Figure 8: (a) Plot for probability of error vs values of α for dataset D4 using strategy 2, (b) Mask of the cheetah with least probability of error

In case of Strategy 2, the Gaussian prior mean μ_0 for both the classes is equal. From the plots shown in Figures 5 to 8, it can be seen that the probability of error decreases as the value of α increases. The Gaussian prior for covariance is a function of α. The posterior mean is a function of Gaussian prior for mean and covariance. Since μ_0 is same for both the classes, it is trusted more for low values of α, which gives higher probability of error. As α increases, the prior mean μ_0 is trusted less which improves and decreases the probability of error. This shows that the strategy of giving equal prior mean μ_0 to both foreground and background classes is not a good strategy.

It is also seen from the plots that as α increases, the probability of error for the predictive distribution procedure converges to that of MLE solution.

Similar to Strategy 1, in the case of MLE procedure, the mean and covariance estimates are not a function of α, thus, the classification of the cheetah image does not vary with the varying values of α. The probability of error in this case remains constant for each of the four datasets.

In the case of MAP procedure, the mean estimate is same as that of predictive distribution procedure, whereas the covariance estimate is the same as MLE procedure. Thus, for MAP procedure as well, the probability of error decreases as the values of α increases.

The general trend between the probability of errors for three approaches using Strategy 2 in all the four datasets is as follows:

Error of MLE Solution < Error of Predictive Distribution < Error of MAP Solution

Further, the trend for probability of error between four datasets for predictive distribution solution is as follows.

Error in D2 < Error in D4 < Error in D3 < Error in D1

This trend remains the same as seen in Strategy 1. Even though the datasets used for Strategy 1 and Strategy 2 are same, there are differences in classification of image and probability of error obtained in both the cases. The differences mainly arise from different choices of Gaussian prior for means in both the strategies. This highlights the importance of choosing a good estimate for prior means to obtain good classification results.

Full code written to solve the quiz is shown below.

```
clc; clear; close all;

load("Alpha.mat")
load("Prior_1.mat")
load("Prior_2.mat")
load("TrainingSamplesDCT_subsets_8.mat")

% Strategy Declaration
mu0_BG(1,1)                = 2;
mu0_FG(1,1)                = 2;

% Initialize Variables
error_Pred_Dist            = zeros(max(size(alpha)), 1);
error_MAP                  = zeros(max(size(alpha)), 1);
error_MLE                  = zeros(max(size(alpha)), 1);
num_Features               = 64;

% Training Sample Set Declaration
Trainsample_FG             = D1_FG;
Trainsample_BG             = D1_BG;
length_TrainSampleFG       = length(Trainsample_FG);
length_TrainSampleBG       = length(Trainsample_BG);

% Calculation of Prior Probabilities
P_cheetah                  = length_TrainSampleFG / (length_TrainSampleFG +
length_TrainSampleBG);
P_grass                    = length_TrainSampleBG / (length_TrainSampleFG +
length_TrainSampleBG);

% Calculation of mean for both the datasets
mean_FG                    = mean(Trainsample_FG);
mean_BG                    = mean(Trainsample_BG);

% Calculation of covariance for both the datasets
cov_FG                     = cov(Trainsample_FG);
cov_BG                     = cov(Trainsample_BG);
det_cov_FG                 = det(cov_FG);
det_cov_BG                 = det(cov_BG);
```

```
% Load Input Image and Declare Variables
inputImg     = imread("cheetah.bmp");
inputImg     = im2double(inputImg);
img_Size     = size(inputImg);
img_Width    = img_Size(1);
img_Height   = img_Size(2);
winSize      = 8;
A            = zeros(img_Width - winSize + 1, img_Height - winSize + 1);

fileID       = fopen('Zig-Zag Pattern.txt','r');
global zigzag
zigzag       = fscanf(fileID, '%d');

% Part a: Solution using parameters of predictive distribution

for k = 1:max(size(alpha))
    % Calculation of mu_n and sigma_n
    cov0_FG                = diag(W0) * alpha(k);
    cov0_BG                = diag(W0) * alpha(k);

    mu_n_FG                = ( (length_TrainSampleFG * cov0_FG) / (cov_FG +
(length_TrainSampleFG * cov0_FG)) ) * mean_FG' + ( cov_FG / (cov_FG +
(length_TrainSampleFG * cov0_FG)) ) * mu0_FG';
    mu_n_BG                = ( (length_TrainSampleBG * cov0_BG) / (cov_BG +
(length_TrainSampleBG * cov0_BG)) ) * mean_BG' + ( cov_BG / (cov_BG +
(length_TrainSampleBG * cov0_BG)) ) * mu0_BG';

    cov_n_FG               = (cov_FG * cov0_FG) / (cov_FG + (length_TrainSampleFG
* cov0_FG));
    cov_n_BG               = (cov_BG * cov0_BG) / (cov_BG + (length_TrainSampleBG
* cov0_BG));

    det_cov_n_FG           = det(cov_n_FG);
    det_cov_n_BG           = det(cov_n_BG);

    % Calculation of sigma_combined
    cov_combined_FG        = cov_FG + cov_n_FG;
    cov_combined_BG        = cov_BG + cov_n_BG;

    det_cov_combined_FG    = det(cov_combined_FG);
    det_cov_combined_BG    = det(cov_combined_BG);

    for j = 1:img_Height - winSize + 1
        for i = 1:img_Width - winSize + 1
            block                  = inputImg(i:i+winSize-1, j:j+winSize-1);
            block_DCT              = dct2(block);
            dct_Vector             = matrix_to_zigzag_vector(block_DCT);
            P_x_FG                 = exp( -0.5*( (dct_Vector - mu_n_FG') *
inv(cov_combined_FG) * (dct_Vector - mu_n_FG')' ) ) / (sqrt(
((2*pi)^num_Features)*det_cov_combined_FG ) );
            P_x_FG                 = log(P_x_FG) + log(P_cheetah);
```

```
            P_x_BG                  = exp( -0.5*( (dct_Vector - mu_n_BG') *
inv(cov_combined_BG) * (dct_Vector - mu_n_BG')' ) ) / (sqrt(
((2*pi)^num_Features)*det_cov_combined_BG ) );
            P_x_BG                  = log(P_x_BG) + log(P_grass);

            if P_x_FG > P_x_BG
                A(i,j)  = 1;
            else
                A(i,j)  = 0;
            end
        end
    end
    imagesc(A)
    colormap(gray(255))
    filename = sprintf('%d.png', k);
    saveas(gcf,filename)

    % Calculation of probability of error
    ground_Truth_Mask   = imread("cheetah_mask.bmp");
    ground_Truth_Mask   = im2double(ground_Truth_Mask);

    error               = sum( abs(A - ground_Truth_Mask(1:img_Width - winSize +
1, 1:img_Height - winSize + 1)), "all" );
    error_Pred_Dist(k)  = error / (img_Width * img_Height);
end

% Part b: Solution using ML Procedure

for k = 1:max(size(alpha))
    for j = 1:img_Height - winSize + 1
        for i = 1:img_Width - winSize + 1
            block                   = inputImg(i:i+winSize-1, j:j+winSize-1);
            block_DCT               = dct2(block);
            dct_Vector              = matrix_to_zigzag_vector(block_DCT);
            P_x_FG                  = exp( -0.5*( (dct_Vector - mean_FG) *
inv(cov_FG) * (dct_Vector - mean_FG)' ) ) / (sqrt(
((2*pi)^num_Features)*det_cov_FG ) );
            P_x_FG                  = log(P_x_FG) + log(P_cheetah);
            P_x_BG                  = exp( -0.5*( (dct_Vector - mean_BG) *
inv(cov_BG) * (dct_Vector - mean_BG)' ) ) / (sqrt(
((2*pi)^num_Features)*det_cov_BG ) );
            P_x_BG                  = log(P_x_BG) + log(P_grass);

            if P_x_FG > P_x_BG
                A(i,j)  = 1;
            else
                A(i,j)  = 0;
            end
        end
    end
```

```
    % Calculation of probability of error
    error                = sum( abs(A - ground_Truth_Mask(1:img_Width - winSize +
1, 1:img_Height - winSize + 1)), "all" );
    error_MLE(k)         = error / (img_Width * img_Height);
end

% Part c: Solution using MAP estimate

for k = 1:max(size(alpha))
    % Calculation of mu_n
    cov0_FG                  = diag(W0) * alpha(k);
    cov0_BG                  = diag(W0) * alpha(k);

    mu_n_FG                  = ( (length_TrainSampleFG * cov0_FG) / (cov_FG +
(length_TrainSampleFG * cov0_FG)) ) * mean_FG' + ( cov_FG / (cov_FG +
(length_TrainSampleFG * cov0_FG)) ) * mu0_FG';
    mu_n_BG                  = ( (length_TrainSampleBG * cov0_BG) / (cov_BG +
(length_TrainSampleBG * cov0_BG)) ) * mean_BG' + ( cov_BG / (cov_BG +
(length_TrainSampleBG * cov0_BG)) ) * mu0_BG';

    for j = 1:img_Height - winSize + 1
        for i = 1:img_Width - winSize + 1
            block                    = inputImg(i:i+winSize-1, j:j+winSize-1);
            block_DCT                = dct2(block);
            dct_Vector               = matrix_to_zigzag_vector(block_DCT);
            P_x_FG                   = exp( -0.5*( (dct_Vector - mu_n_FG') *
inv(cov_FG) * (dct_Vector - mu_n_FG')' ) ) / (sqrt(
((2*pi)^num_Features)*det_cov_FG ) );
            P_x_FG                   = log(P_x_FG) + log(P_cheetah);
            P_x_BG                   = exp( -0.5*( (dct_Vector - mu_n_BG') *
inv(cov_BG) * (dct_Vector - mu_n_BG')' ) ) / (sqrt(
((2*pi)^num_Features)*det_cov_BG ) );
            P_x_BG                   = log(P_x_BG) + log(P_grass);

            if P_x_FG > P_x_BG
                A(i,j)  = 1;
            else
                A(i,j)  = 0;
            end
        end
    end

    % Calculation of probability of error
    error                = sum( abs(A - ground_Truth_Mask(1:img_Width - winSize +
1, 1:img_Height - winSize + 1)), "all" );
    error_MAP(k)         = error / (img_Width * img_Height);
end

figure;
plot(alpha, error_Pred_Dist)
hold on
```

```
plot(alpha, error_MLE)
hold on
plot(alpha, error_MAP)
hold off
set(gca, 'XScale', 'log')
xlabel('Alpha - Values')
ylabel('Probability of Error')
title('Error vs Alpha Plot for Set D1 and Strategy 2')
legend('Predictive Distribution', 'MLE', 'MAP')
saveas(gca,'Plot_D1_S2.jpg')

function dct_vector = matrix_to_zigzag_vector(img_dct_block)
    dct_vector  = zeros(1,64);
    global zigzag
    for i = 1:8
        for j = 1:8
            index = zigzag( (i-1)*8 + j ) + 1;
            dct_vector(index) = img_dct_block(i,j);
        end
    end
end
```