Joseph Bell
ECE 271A

Homework 2 Computer Problem

a) The prior probabilities for the cheetah and grass were 0.1919 and 0.8081 respectively. These prior probabilities are the same as the last homework assignment. The prior probabilities for this homework assignment were calculated by maximum likelihood estimation.

The MLE for the prior distribution was calculated to be: $P(k) = \pi_k = \frac{c_k}{n}$, where π_k is the prior probability, c_k is the number of observations of the feature (i.e. cheetah or grass) and n is the total number of observations.

In homework 1 the priors were calculated by dividing the number of elements in the training data for both cheetah and grass by the total number of training data elements. This is the same thing as the maximum likelihood estimate for the prior probabilities.

b) The class conditional densities PX|Y (x|cheetah) and PX|Y (x|grass) were computed for the 64 DCT coefficients using MLE under the gaussian assumption. The MLE for the mean was $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x_i}$ and the MLE for the variance was $\boldsymbol{\sigma^2} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{\mu} - \boldsymbol{x_i})^2$. The area of intersection of the class conditional distributions was calculated and the 8 with the smallest area of intersection were selected as the "best 8" and the 8 with the largest area of intersection were selected as the "worst 8". The best 8 feature distributions can be seen in figure 1, and the worst 8 feature distributions can be seen in figure 2. Figure 3 displays all 64 feature distributions. The distribution for the cheetah is displayed in red and the distribution for the grass is displayed in green. The title of each subplot represents the coefficient index.

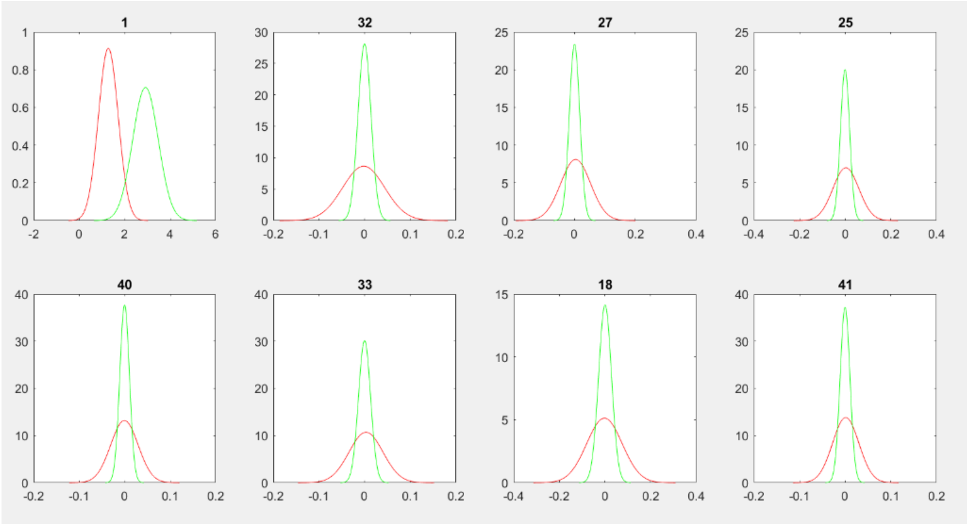

Figure 1: Best 8 features for classification purposes

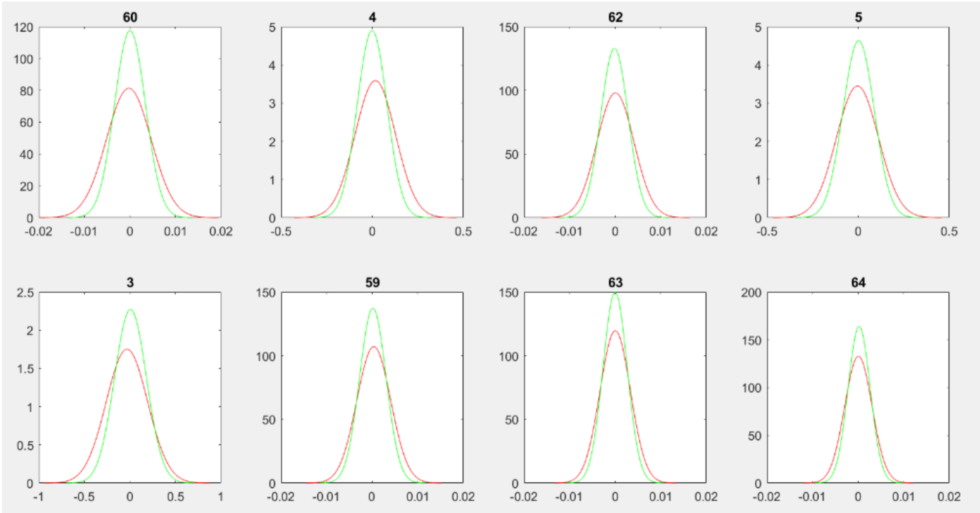

Figure 2: Worst 8 features for classification purposes

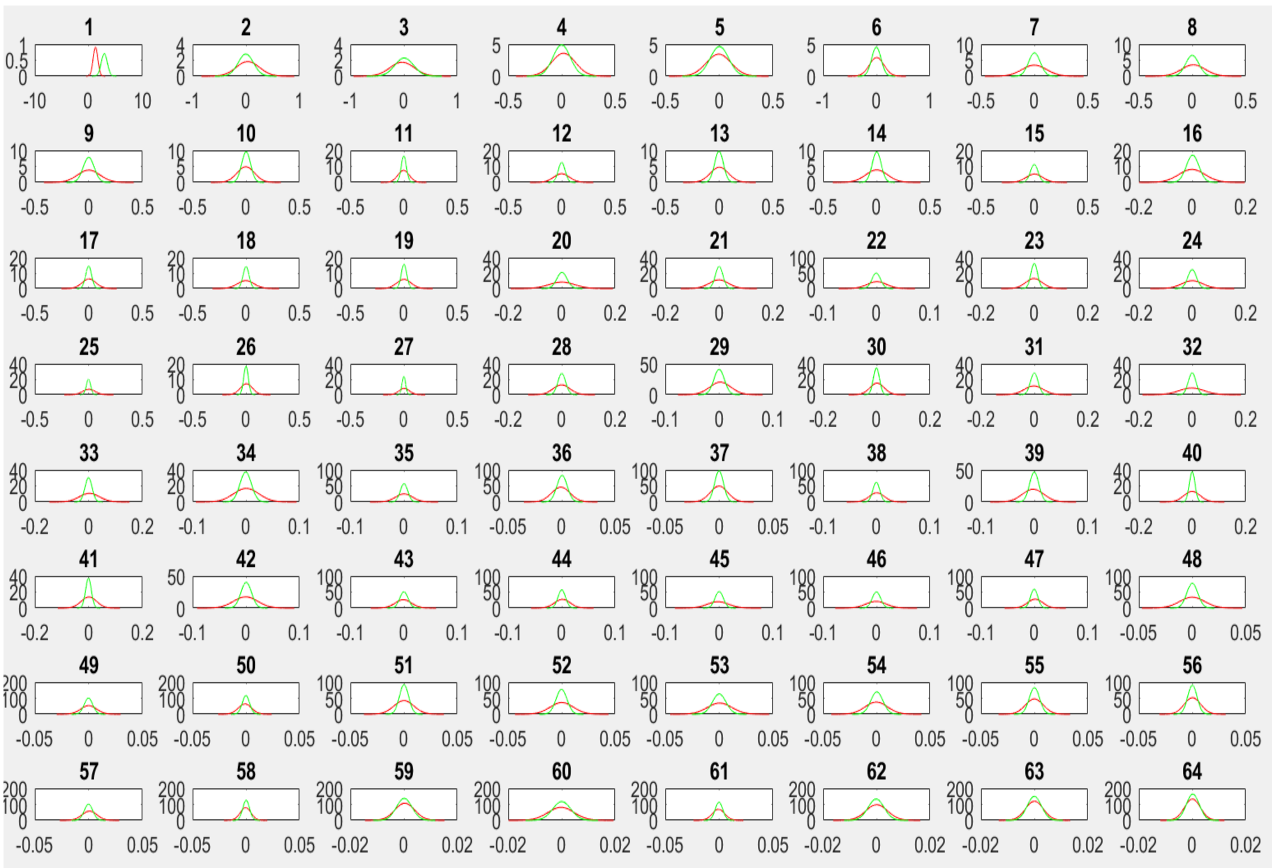

Figure 3: All 64 feature distributions

c) The Bayesian decision rule used for classification is shown in figure 4. The arguments in the first line are represented by the function g(x). The classification was done first using the 64-dimensional Gaussians and then using the best 8-dimensional Gaussians. The classification results are seen in figures 5 and 6 respectively. The probability of error for the 64-dimensional case was 8.72% and the probability of error for the 8-dimensional case was 6.48%. There were many feature distributions that had substantial overlap (e.g. the worst 8 features shown in figure 2), and the addition of these poor features do nothing but "confuse" the classifier and weaken its certainty for classification. They push the classifier to be "less strict" when choosing which pixels represent the cheetah. The "strict nature" of the 8-dimensional Gaussian classifier is represented visually by the entire neck of the cheetah being omitted. In the original image one can see that the neck area is in fact quite similar to the texture of the grass in comparison to the rest of the cheetah – too similar that the classifier didn't "take the chance" to classify it as a part of the cheetah. However, from a visual perspective, the 8-dimensional Gaussian classifier had far fewer false positives (i.e. identifying grass as cheetah) due to its "strict nature" of classifying which ultimately resulted in a smaller error percentage. I don't mean to anthropomorphize the classifier, but doing so aids in describing its behavior.

$$i^*(x) = \arg\min_i \left[d_i(x, \mu_i) + \alpha_i\right]$$

$$g_i(x) = x^T W_i\, x + w_i^T x + w_{i0}$$

$$W_i = \Sigma_i^{-1}$$

$$w_i = -2\Sigma_i^{-1}\mu_i$$

$$w_{i0} = \mu_i^T \Sigma_i^{-1}\mu_i + \log\left|\Sigma_i\right| - 2\log P_Y(i)$$

Figure 4: Bayesian decision rule used for classification

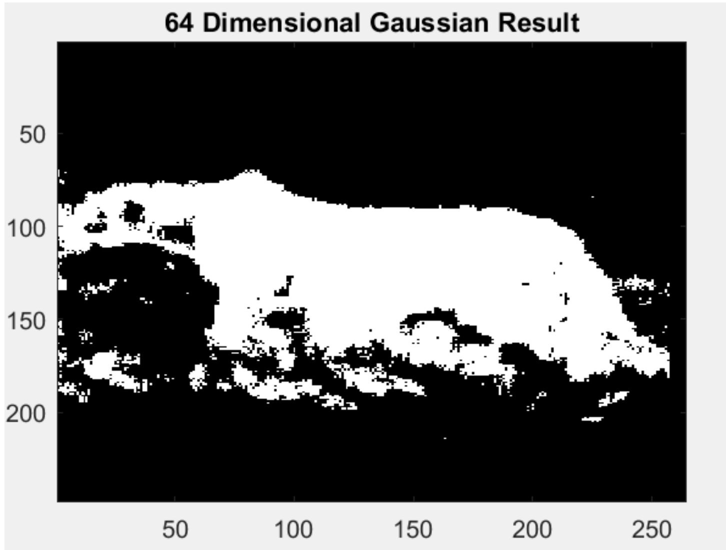

Figure 5: 64-dimensional Gaussian classification result

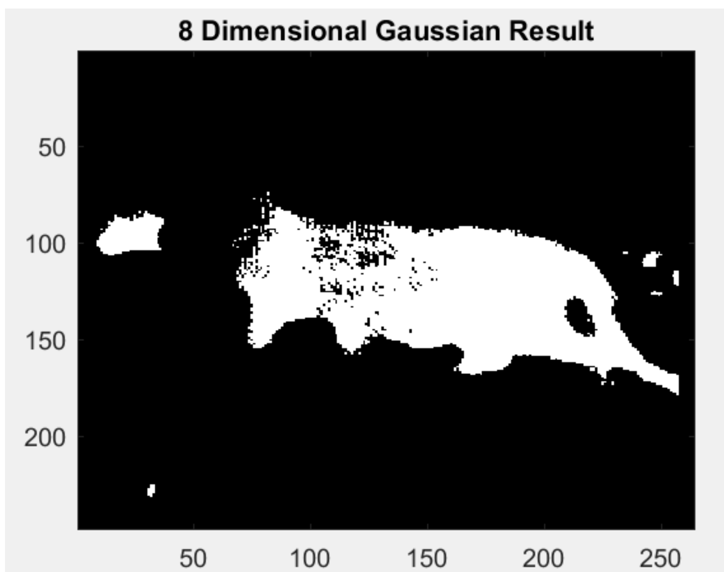

Figure 6: 8-dimensional Gaussian classification result

Appendix

```
%Joseph Bell
%ECE271A HW2

clc;
clear;
load('TrainingSamplesDCT_8_new.mat');

%%%%% CALCULATING PRIORS %%%%%
[rows_FG, cols_FG] = size(TrainsampleDCT_FG);
[rows_BG, cols_BG] = size(TrainsampleDCT_BG);

FG_training_elements = rows_FG*cols_FG;
BG_training_elements = rows_BG*cols_BG;
n = FG_training_elements + BG_training_elements;

% calculating prior of cheetah/foreground
z_ik = 0;
for i=1:FG_training_elements
    z_ik = z_ik + 1;
end

for i=1:BG_training_elements %this step is not necessary, but i'm doing it
    z_ik = z_ik + 0;         %to illustrate the maximum likelihood equation
end

prior_cheetah = z_ik/n; %0.1919
prior_background = 1 - prior_cheetah; %0.8081

%Priors are the same as last week. Reasoning explained in report.

%Calculating sample mean and variance using
%conclusion of maximum likelihood


cheetah_zigzag = zeros(64, rows_FG);
grass_zigzag = zeros(64, rows_BG);

%zigzagging each row and placing data in a column where each row is
%a DCT coefficient
for row=1:rows_FG
    cheetah_zigzag(:,row) = zigzag(TrainsampleDCT_FG(row,:));
end

for row=1:rows_BG
    grass_zigzag(:,row) = zigzag(TrainsampleDCT_BG(row,:));
end

means_variance_cheetah = zeros(64, 1, 2); %(:,:,1) = mean
```

```
means_variance_grass = zeros(64, 1, 2); %(:,:,2) = variance




%cheetah loop
for row=1:cols_FG
    N = rows_FG;
    sample_mean = 0;
    sample_variance = 0;

    sample = cheetah_zigzag(row,:); %grab each coefficient row

    for i=1:N
        sample_mean = sample_mean + sample(1,i);
    end
    sample_mean = sample_mean/N;

    for i=1:N
        sample_variance = sample_variance + (sample(1,i) - sample_mean)^2;
    end
    sample_variance = sample_variance/N;

    means_variance_cheetah(row,1,1) = sample_mean;
    means_variance_cheetah(row,1,2) = sample_variance;
end

%grass loop
for row=1:cols_BG
    N = rows_BG;
    sample_mean = 0;
    sample_variance = 0;

    sample = grass_zigzag(row,:); %grab each coefficient row

    for i=1:N
        sample_mean = sample_mean + sample(1,i);
    end
    sample_mean = sample_mean/N;

    for i=1:N
        sample_variance = sample_variance + (sample(1,i) - sample_mean)^2;
    end
    sample_variance = sample_variance/N;

    means_variance_grass(row,1,1) = sample_mean;
    means_variance_grass(row,1,2) = sample_variance;
end
```

```
overlap_areas = zeros(1,64);
num_of_points = 1000;
figure(1)
for i=1:8
    mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);

end

figure(2)
for i=9:16
    mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i-8)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);
end
```

```
figure(3)
for i=17:24
    mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i-16)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);
end
figure(4)
for i=25:32
   mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i-24)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);
end
figure(5)
for i=33:40
    mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
```

```
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i-32)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);
end
figure(6)
for i=41:48
    mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i-40)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);
end
figure(7)
for i=49:56
    mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i-48)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
```

```
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);
end

figure(8)
for i=57:64
    mean = means_variance_cheetah(i,1,1);
    variance = means_variance_cheetah(i,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i-56)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(i,1,1);
    variance = means_variance_grass(i,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(i);

    overlap_areas(1,i) = calculateOverlapArea(values1,values2,y1,y2,num_of_points);
end

figure(9)
[sorted_overlap_areas, indices] = sort(overlap_areas);
top_8 = [1 indices(1:7)];
bottom_8 = indices(57:end);
[r, c] = size(top_8);

for i=1:c
    index = top_8(i);
    mean = means_variance_cheetah(index,1,1);
    variance = means_variance_cheetah(index,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
```

```
    subplot(2,4,i)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(index,1,1);
    variance = means_variance_grass(index,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(index);

end

figure(10)
for i=1:c
    index = bottom_8(i);
    mean = means_variance_cheetah(index,1,1);
    variance = means_variance_cheetah(index,1,2);
    standard_deviation = sqrt(variance);
    values1 = linspace(mean-4*standard_deviation,mean+4*standard_deviation,
num_of_points);
    subplot(2,4,i)
    y1 = normpdf(values1, mean, standard_deviation);
    plot(values1, y1, 'r');
    hold on
    mean = means_variance_grass(index,1,1);
    variance = means_variance_grass(index,1,2);
    standard_deviation = sqrt(variance);
    values2 = linspace(mean-
4*standard_deviation,mean+4*standard_deviation,num_of_points);
    y2 = normpdf(values2, mean, standard_deviation);
    plot(values2, y2, 'g');
    title(index);

end

mu_cheetah = means_variance_cheetah(:,:,1);
mu_grass = means_variance_grass(:,:,1);
x_0 = (mu_cheetah+mu_grass)/2;
variance_cheetah = means_variance_cheetah(:,:,2);
variance_grass = means_variance_grass(:,:,2);

cheetah_covariances = zeros(64,64);
grass_covariances = zeros(64,64);

%calculating covariance matrix for cheetah
for i=1:64
```

```
    p1 = cheetah_zigzag(i,:); %grab coefficient row
    mu_1 = means_variance_cheetah(i,1,1);
    for j=1:64
        p2 = cheetah_zigzag(j,:); %grab coefficient row
        mu_2 = means_variance_cheetah(j,1,1);
        temp = 0;
        for k=1:250
            temp = temp + (p1(1,k) - mu_1)*(p2(1,k) - mu_2);
        end
        cheetah_covariances(i,j) = temp/249;
    end
end

%calculating covariance matrix for grass
for i=1:64
    p1 = grass_zigzag(i,:); %grab coefficient row
    mu_1 = means_variance_grass(i,1,1);
    for j=1:64
        p2 = grass_zigzag(j,:); %grab coefficient row
        mu_2 = means_variance_grass(j,1,1);
        temp = 0;
        for k=1:1053
            temp = temp + (p1(1,k) - mu_1)*(p2(1,k) - mu_2);
        end
        grass_covariances(i,j) = temp/1052;
    end
end



cheetah_img = imread('cheetah.bmp');
cheetah_img = im2double(cheetah_img); %converting to double values since training data
is of type double
[cheetah_rows, cheetah_cols] = size(cheetah_img);
cheetah_img = cheetah_img(1:8*floor(cheetah_rows/8),1:8*floor(cheetah_cols/8));
%modifying image so it can be split into 8x8 blocks
[cheetah_rows, cheetah_cols] = size(cheetah_img); %overwriting for modified dimensions

zz = load('Zig-Zag Pattern.txt');
zz = zz+1;
zz = zigzag(zz); %Credit to Alexey Sokolov from
https://www.mathworks.com/matlabcentral/fileexchange/15317-zigzag-scan
                 %for the zig zag code

%%%%% Block Window Sliding %%%%%
new_image64 = zeros(cheetah_rows, cheetah_cols);
new_image8 = zeros(cheetah_rows, cheetah_cols);
for i=1:cheetah_cols-7 %shift scan pointer over a column
    for j=1:cheetah_rows-7
```

```
        block = cheetah_img(j:7+j,i:7+i); %grab 8x8 block
        block_dct = dct2(block);
        zzblock_dct = zigzag(block_dct);


        %%%%% DO BAYESIAN DECISION RULE %%%%%
        [g_cheetah64, g_grass64] =
gaussianClassifier(transpose(zzblock_dct),means_variance_cheetah,means_variance_grass,
cheetah_covariances,grass_covariances,prior_cheetah,prior_background);
        [g_cheetah8, g_grass8] =
gaussianClassifier(transpose(zzblock_dct(1,top_8)),means_variance_cheetah(top_8,:,:),m
eans_variance_grass(top_8,:,:),cheetah_covariances(1:8,1:8),grass_covariances(1:8,1:8)
,prior_cheetah,prior_background);

        if g_cheetah64 < g_grass64
            new_image64(j:j,i:i) = 1;
        end
        if g_cheetah8 < g_grass8
            new_image8(j:j,i:i) = 1;
        end
    end
end

figure(12)
imagesc(new_image64);
colormap(gray(255));
title('64 Dimensional Gaussian Result')

figure(13)
imagesc(new_image8);
colormap(gray(255));
title('8 Dimensional Gaussian Result')

cheetah_mask = double(imread('cheetah_mask.bmp')/255);

counter_correct64 = 0;
counter_correct8 = 0;
total_pixels = cheetah_rows*cheetah_cols;
for i=1:cheetah_rows
    for j=1:cheetah_cols
        if cheetah_mask(i,j) ==  new_image64(i,j)
            counter_correct64 = counter_correct64 + 1;
        end
        if cheetah_mask(i,j) ==  new_image8(i,j)
            counter_correct8 = counter_correct8 + 1;
        end
    end
end
```

```
percent_correct64 = counter_correct64/total_pixels*100;
percent_correct8 = counter_correct8/total_pixels*100;

%function used to determine 8 best and 8 worst pdf combinations
function pct_area = calculateOverlapArea(x1,x2,y1,y2,num_points)
    y_overlap = [y1(y1<y2) y2(y2<y1)];

    lesser_x = x2(1);
    greater_x = x2(end);

    if x1(1) > x2(1)
        lesser_x = x1(1);
    end
    if x1(end) < x2(end)
        greater_x = x1(end);
    end

    values = linspace(lesser_x,greater_x,num_points);
    area_int  = trapz(values,y_overlap);
    total_area = trapz(x1,y1) + trapz(x2,y2);
    pct_area = area_int/total_area;
    plot(values, y_overlap,'b');
end

function [g_cheetah, g_grass] =
gaussianClassifier(x,means_variance_cheetah,means_variance_grass,cheetah_covariances,g
rass_covariances,prior_cheetah,prior_background)
    W_I_cheetah = inv(cheetah_covariances);
    W_I_grass = inv(grass_covariances);

    w_i_cheetah = -2*inv(cheetah_covariances)*means_variance_cheetah(:,:,1);
    w_i_grass = -2*inv(grass_covariances)*means_variance_grass(:,:,1);
    w_i_0_cheetah =
transpose(means_variance_cheetah(:,:,1))*inv(cheetah_covariances)*means_variance_cheet
ah(:,:,1) + log(det(cheetah_covariances))-2*log(prior_cheetah);
    w_i_0_grass =
transpose(means_variance_grass(:,:,1))*inv(grass_covariances)*means_variance_grass(:,:
,1) + log(det(grass_covariances))-2*log(prior_background);

    g_cheetah = transpose(x)*W_I_cheetah*x + transpose(w_i_cheetah)*x + w_i_0_cheetah;
    g_grass = transpose(x)*W_I_grass*x + transpose(w_i_grass)*x + w_i_0_grass;

end
```