Name: Neil Chen
PID: A59013393

November 14, 2021

ECE 271A Homework 2

1. This week we will continue trying to classify our cheetah example. Once again we use the decomposition into 8 x 8 image blocks, compute the DCT of each block, and zig-zag scan. However, we are going to assume that the class-conditional densities are multivariate Gaussians of 64 dimensions.

   (a) Using the training data in **TrainingSamplesDCT 8.mat** compute the histogram estimate of the prior $P_Y(i), i \in \{cheetah, grass\}$. Using the results of problem 2 compute the maximum likelihood estimate for the prior probabilities. Compare the result with the estimates that you obtained last week. If they are the same, interpret what you did last week. If they are different, explain the differences.

Solution.
We obtained the value of $P_Y(cheetah)$ and $P_Y(grass)$, which is 0.192 and 0.808 respectively (the histogram is shown in Fig. 1). The result aligns with the result from last week. In my reckoning, since the mean of the estimator $\hat{\pi}_i$ is equal to π_i, which is an unbiased estimator, hence $\hat{\pi}_i$ will converge to π_i as the number of observations n goes to a large number, in this case n = 64, which is large enough.

Matlab Code

```
clear
clc
clf

% import data
load("HW2\TrainingSamplesDCT_8_new.mat");
M_cheetah = TrainsampleDCT_FG;
M_grass = TrainsampleDCT_BG;

% calculate the number of appearance
[m_cheetah, ~] = size(M_cheetah);
[m_grass, ~] = size(M_grass);

% calculate the probabilities
```

```
P_Y_cheetah = m_cheetah / (m_grass + m_cheetah);
P_Y_cheetah = round(P_Y_cheetah, 2);
P_Y_grass = m_grass / (m_grass + m_cheetah);
P_Y_grass = round(P_Y_grass, 2);

% plot the histogram
X = categorical({'Cheetah', 'Grass'});
Y = [P_Y_cheetah P_Y_grass];
bar(X, Y)
text(1:length(Y),Y,num2str(Y'),'vert','bottom','horiz','center
    ');
```

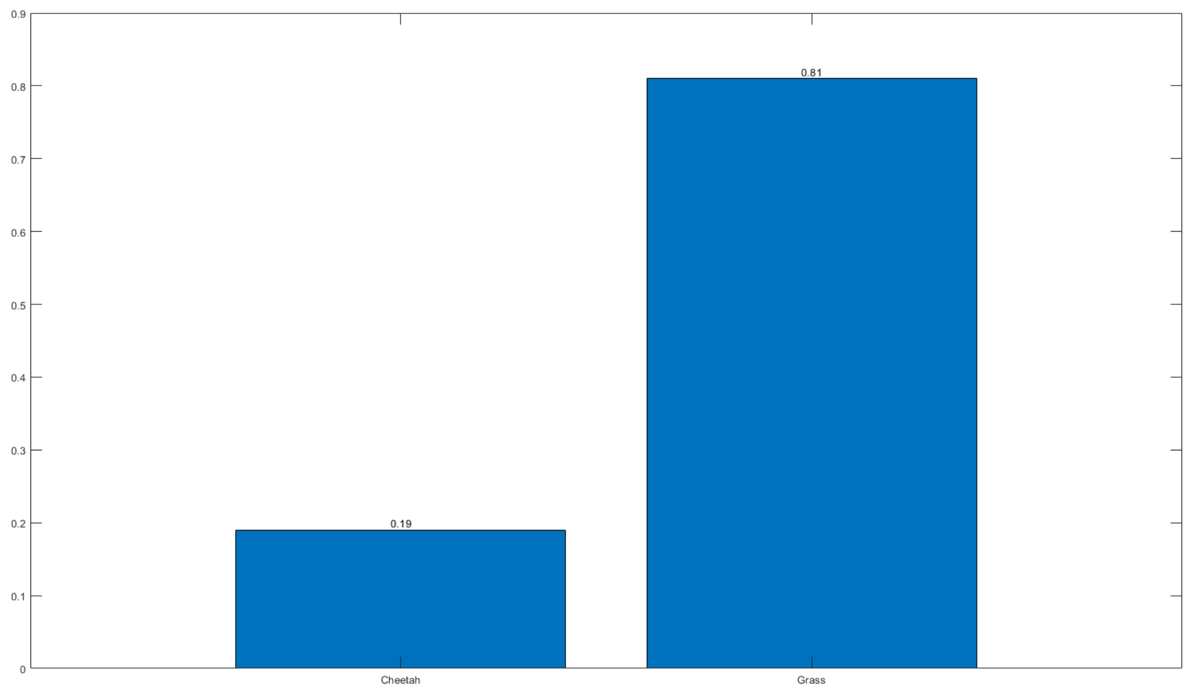

Figure 1: 64 Features

(b) Using the training data in TrainingSamplesDCT 8.mat, compute the maximum likelihood estimates for the parameters of the class conditional densities $P_{X|Y}(x \mid cheetah)$ and $P_{X|Y}(x \mid grass)$ under the Gaussian assumption. Denoting by $X = \{X_1, X_2, X_3, ..., X_n\}$ the vector of DCT coefficients, create 64 plots with the marginal densities for the two classes $P_{X_k|Y}(x_k \mid cheetah)$ and $P_{X_k|Y}(x_k \mid grass)$, $k = 1, 2, ..., 64$ - on each. Use different line styles for each marginal. Select, by visual inspection, what you think are the best 8 features for classification purposes and what you think are the worst 8 features (you can use the subplot command to compare several plots at a time). Hand in the plots of the marginal densities for the best-8 and worst-8 features (once again you can use subplot, this should not require more than two sheets of paper). In each subplot indicate the feature that it refers to.

Solution.
My code is in the following page. The result of those 64 features is shown on Fig. 2. Based on my observation, **Feature No. 1** to **Feature No. 8** are the best 8 features since the mean values for each data (i.e., the mean of *cheetah* and the mean of *grass*) are distinctive (See Fig. 3). As for the worst 8 features, I believe that **Feature No. 37**, **Feature No. 40**, **Feature No. 51**, **Feature No. 59**, **Feature No. 60**, **Feature No. 62**, **Feature No. 63**, and **Feature No. 64** are the worst since the means are indistinguishable from each other (See Fig. 4).

Matlab Code

```
clear
clc
clf

load("HW2\TrainingSamplesDCT_8_new.mat");

% creat vectors for means and variances
mean_cheetah = zeros(1, 64);
var_cheetah  = zeros(1, 64);
mean_grass   = zeros(1, 64);
var_grass    = zeros(1, 64);

% calculate means and variances of cheetah and grass
for indx = 1 : 64
    col = TrainsampleDCT_FG(:,indx);
    mean_cheetah(1, indx) = mean(col);
```

```
    var_cheetah(1, indx) = var(col);
    col = TrainsampleDCT_BG(:,indx);
    mean_grass(1, indx) = mean(col);
    var_grass(1, indx) = var(col);
end

% create the x-axis and y-axis for cheetah and grass
x_cheetah = -1 : 0.01 : 5;
x_grass = x_cheetah;

deno_cheetah = (sqrt(var_cheetah(1) * 2 * pi));
deno_grass = (sqrt(var_grass(1) * 2 * pi));
pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(1)).^2 / var_cheetah(1));
pow_grass = -0.5 * ((x_cheetah - mean_grass(1)).^2 / var_cheetah(1));
vec_cheetah = exp(pow_cheetah) / deno_cheetah;
vec_grass   = exp(pow_grass) / deno_grass;
subplot(8, 8, 1);
plot(x_cheetah, vec_cheetah);
hold on
plot(x_grass, vec_grass);
delta = mean_cheetah(1) - mean_grass(1);
text = "\delta = " + round(abs(delta), 4);
subtitle(text)

for indx = 2 : 64
    x_cheetah = mean_cheetah(indx) - 0.25 : 0.001 : mean_cheetah(indx) + 0.25;
    x_grass = x_cheetah;
    co_cheetah = 1 / (sqrt(var_cheetah(indx) * 2 * pi));
    co_grass = 1 / (sqrt(var_grass(indx) * 2 * pi));
    pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(indx)).^2 / var_cheetah(indx));
    pow_grass = -0.5 * ((x_cheetah - mean_grass(indx)).^2 / var_cheetah(indx));
    vec_cheetah = exp(pow_cheetah) * co_cheetah;
    vec_grass = exp(pow_grass) * co_grass;
    subplot(8, 8, indx);
```

```
    plot(x_cheetah, vec_cheetah);
    hold on
    plot(x_grass, vec_grass);
    delta = mean_cheetah(indx) - mean_grass(indx);
    text = "\delta = " + round(abs(delta), 4);
    subtitle(text)
end
sgtitle('64 Features')
figure

% extract the best 8 features
cnt = 2;
desire_best = 1 : 1 : 8;
for i = 2 : 8
    indx = desire_best(i);
    x_cheetah = mean_cheetah(indx) - 0.5 : 0.001 : mean_cheetah(indx) + 0.5;
    x_grass = x_cheetah;
    co_cheetah = 1 / (sqrt(var_cheetah(indx) * 2 * pi));
    co_grass = 1 / (sqrt(var_grass(indx) * 2 * pi));
    pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(indx)).^2 / var_cheetah(indx));
    pow_grass = -0.5 * ((x_cheetah - mean_grass(indx)).^2 / var_cheetah(indx));
    vec_cheetah = exp(pow_cheetah) * co_cheetah;
    vec_grass = exp(pow_grass) * co_grass;
    subplot(4, 2, cnt);
    plot(x_cheetah, vec_cheetah);
    hold on
    plot(x_grass, vec_grass);
    delta = mean_cheetah(indx) - mean_grass(indx);
    text = "\delta = " + round(abs(delta), 4);
    subtitle(text)
    cnt = cnt + 1;
end
x_cheetah = -1 : 0.01 : 5;
x_grass = x_cheetah;

deno_cheetah = (sqrt(var_cheetah(1) * 2 * pi));
```

```
deno_grass = (sqrt(var_grass(1) * 2 * pi));
pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(1)).^2 /
    var_cheetah(1));
pow_grass = -0.5 * ((x_cheetah - mean_grass(1)).^2 /
    var_cheetah(1));
vec_cheetah = exp(pow_cheetah) / deno_cheetah;
vec_grass   = exp(pow_grass) / deno_grass;
subplot(4, 2, 1);
plot(x_cheetah, vec_cheetah);
hold on
plot(x_grass, vec_grass);
delta = mean_cheetah(1) - mean_grass(1);
text = "\delta = " + round(abs(delta), 4);
subtitle(text)
sgtitle('Best 8 Features')
figure

% extract the worst 8 features
cnt = 1;
desire_worst = [37 40 51 59 60 62 63 64];
for i = 1 : 8
    indx = desire_worst(i);
    x_cheetah = mean_cheetah(indx) - 0.05 : 0.001 :
        mean_cheetah(indx) + 0.05;
    x_grass = x_cheetah;
    co_cheetah = 1 / (sqrt(var_cheetah(indx) * 2 * pi));
    co_grass = 1 / (sqrt(var_grass(indx) * 2 * pi));
    pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(indx)).^2
        / var_cheetah(indx));
    pow_grass = -0.5 * ((x_cheetah - mean_grass(indx)).^2 /
        var_cheetah(indx));
    vec_cheetah = exp(pow_cheetah) * co_cheetah;
    vec_grass = exp(pow_grass) * co_grass;
    subplot(4, 2, cnt);
    plot(x_cheetah, vec_cheetah);
    hold on
    plot(x_grass, vec_grass);
    delta = mean_cheetah(indx) - mean_grass(indx);
    text = "\delta = " + round(abs(delta), 4);
```

```
    subtitle(text)
    cnt = cnt + 1;
end
sgtitle('Worst 8 Features')
```

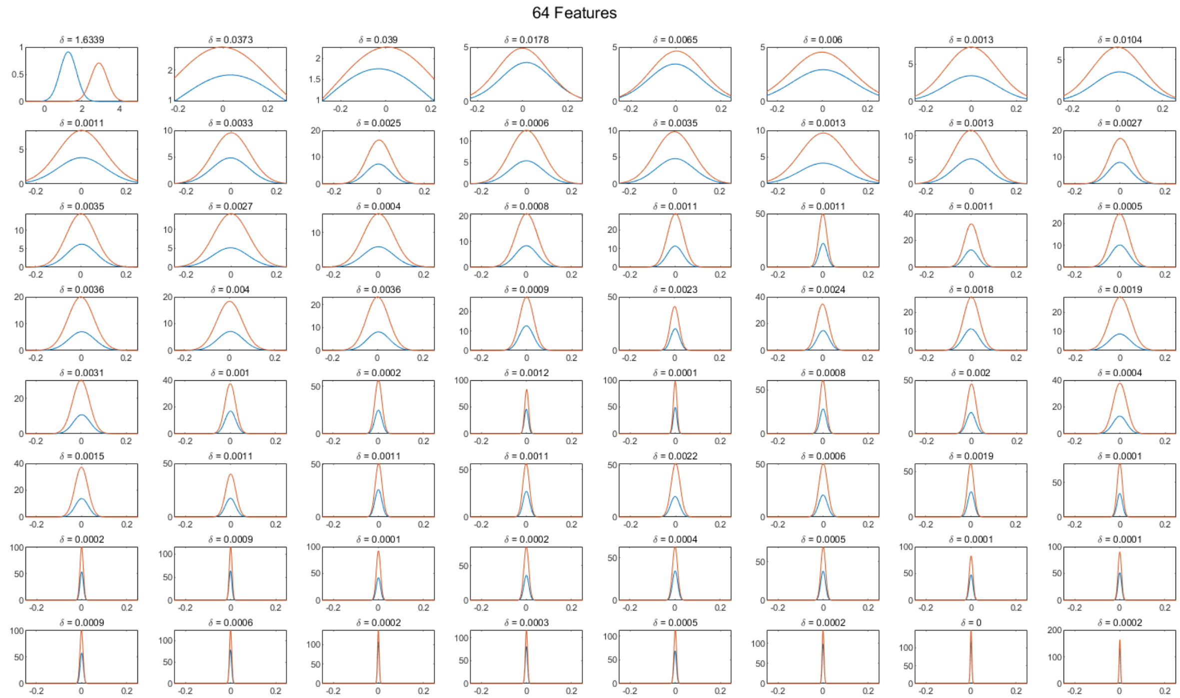

Figure 2: 64 Features

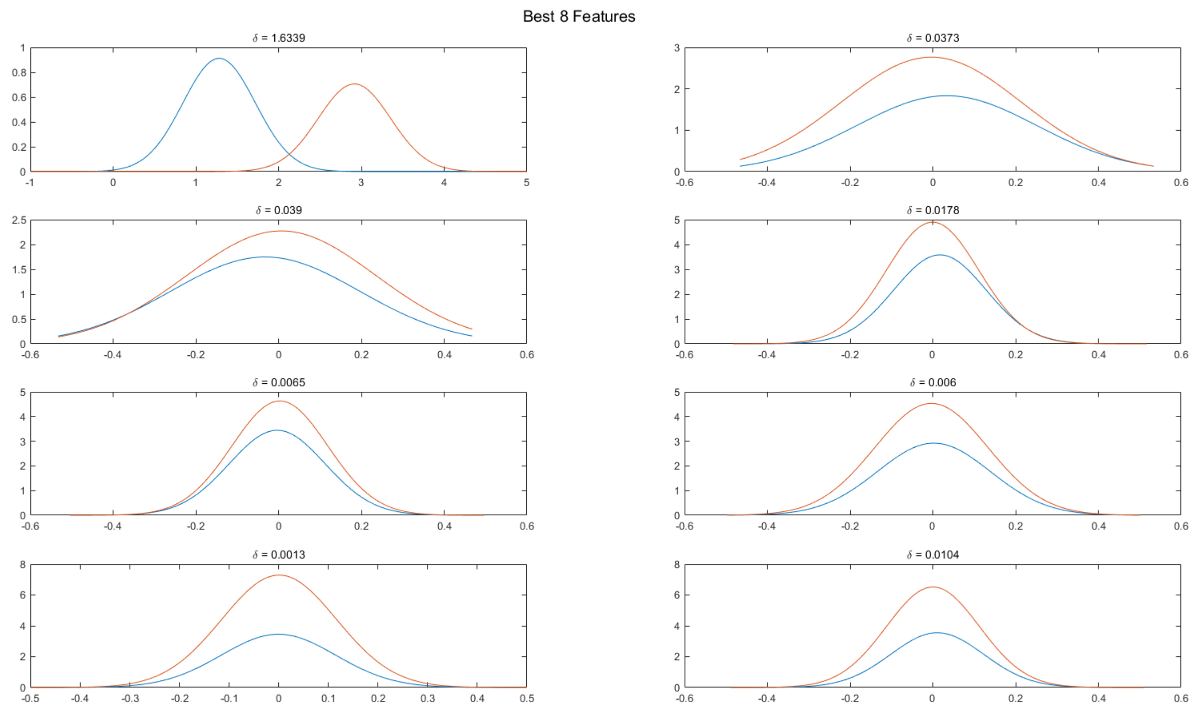

Figure 3: Best 8 Features

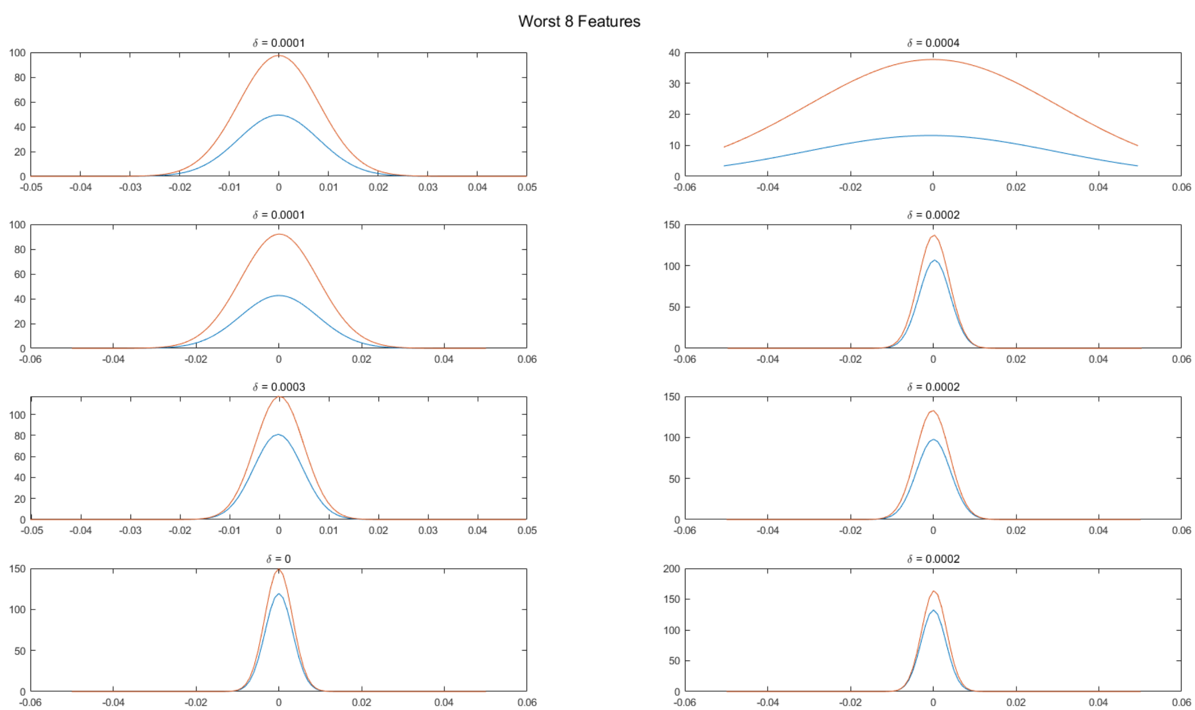

Figure 4: Worst 8 Features

(c) Compute the Bayesian decision rule and classify the locations of the cheetah image using (i) the 64-dimensional Gaussians, and (ii) the 8-dimensional Gaussians associated with the best 8 features. For the two cases, plot the classification masks and compute the probability of error by comparing with cheetah **mask.bmp**. Can you explain the results?

Solution.
My code is in the following page. The result from using 64 features is shown in the middle figure of Fig. 5 and the result form using 8 features is shown on the right figure of Fig. 5. The error rate of the first result and the second result is 9.7% and 7% respectively. And the side-by-side comparison between three mask is shown in Fig. 6. I used δ to represent the difference between the mean of cheetah and the mean of grass. If the value of δ is large, then we can easy distinguish **cheetah** from **grass**; however, if the value of δ is small, then we cannot tell them apart.

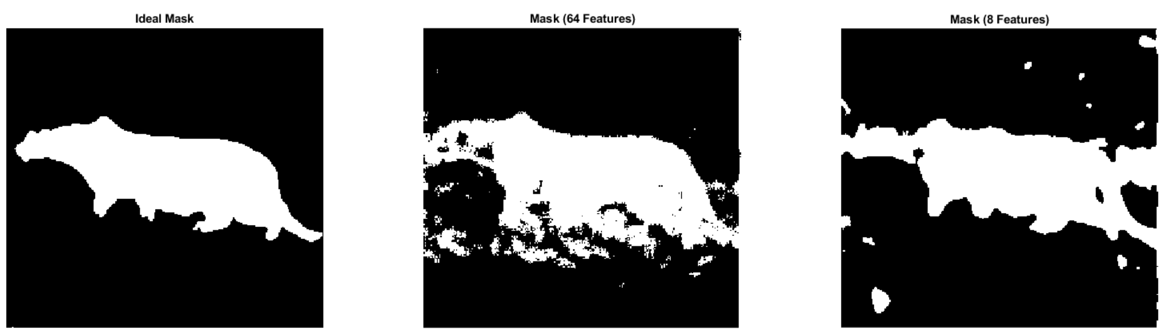

Figure 5: Side-by-side comparison between three masks

Matlab Code

```
clear
clc

load("HW2\TrainingSamplesDCT_8_new.mat");
img_cheetah = imread("cheetah.bmp");
mask_ans = imread("cheetah_mask.bmp");
img_cheetah = im2double(img_cheetah);
mask_ans = imbinarize(mask_ans);

subplot(1, 3, 1);
imshow(mask_ans);
hold on
title('Ideal Mask')

% creat vectors for means and variances
mean_cheetah_64D = zeros(1, 64);
var_cheetah_64D  = zeros(1, 64);
mean_grass_64D   = zeros(1, 64);
var_grass_64D    = zeros(1, 64);

% calculate means and variances of cheetah and grass
for i = 1 : 64
    col = TrainsampleDCT_FG(:,i);
    mean_cheetah_64D(1, i) = mean(col);
    var_cheetah_64D(1, i) = var(col);
    col = TrainsampleDCT_BG(:,i);
    mean_grass_64D(1, i) = mean(col);
    var_grass_64D(1, i) = var(col);
end

% DCT (64D)
[m, n] = size(img_cheetah);
mask_64D = zeros(m - 7, n - 7);

% padding the image
I = zeros(m + 7, n + 7);
I(1:m, 1:n) = img_cheetah;
```

```
% the zigzag map
ZigZagM = [0    1    5    6   14   15   27   28;
     2    4    7   13   16   26   29   42;
     3    8   12   17   25   30   41   43;
     9   11   18   24   31   40   44   53;
     10   19   23   32   39   45   52   54;
     20   22   33   38   46   51   55   60;
     21   34   37   47   50   56   59   61;
     35   36   48   49   57   58   62   63];
ZigZagM = ZigZagM + 1;  % index in MATLAB starts with 1
ZigZagV = ZigZagM(:);   % 64 x 1
ZigZagV = ZigZagV.';    % 1 x 64

% calculate the sigmas and sigma^-1s
sigma_cheetah_64D    = cov(TrainsampleDCT_FG);
sigma_grass_64D      = cov(TrainsampleDCT_BG);
sigmaInv_cheetah_64D = inv(sigma_cheetah_64D);
sigmaInv_grass_64D   = inv(sigma_grass_64D);

% calculate the coefficients
deno_cheetah = (sqrt(((2 * pi)^64) * det(sigma_cheetah_64D)));
deno_grass   = (sqrt(((2 * pi)^64) * det(sigma_grass_64D)));

% calculate DCT2 (64D)
m = m - 7;
n = n - 7;
for i = 1 : m
    for j = 1 : n
        Block = I(i : i + 7, j : j + 7);
        Block_DCT = dct2(Block, 8, 8);
        V = Block_DCT(:).';
        X = zeros(1, 64);
        % mapping
        for k = 1 : 64
            X(ZigZagV(k)) = V(k);
        end
        % calculate the probabilities
        P_X_cheetah = exp(-0.5 * (X - mean_cheetah_64D) * (
```

```
sigmaInv_cheetah_64D) * (X - mean_cheetah_64D).') / deno_cheetah;
        P_X_grass   = exp(-0.5 * (X - mean_grass_64D)   * (sigmaInv_grass_64D)   * (X - mean_grass_64D).')   / deno_grass;
        % based on the results, decide whether the pixel is grass or cheetah
        if (P_X_grass > P_X_cheetah)
            mask_64D(i, j) = 0;
        else
            mask_64D(i, j) = 1;
        end
    end
end

% compare my mask with ans
e_64D = 0;
for i = 1 : m
    for j = 1 : n
        if (mask_ans(i, j) ~= mask_64D(i, j))
            e_64D = e_64D + 1;
        end
    end
end
error_rate_64D = e_64D / (m * n);

% plot the result
subplot(1, 3, 2);
imshow(mask_64D);
hold on
title('Mask (64 Features)')

% creat vectors for means and variances (8D)
mean_cheetah_8D = zeros(1, 8);
var_cheetah_8D  = zeros(1, 8);
mean_grass_8D   = zeros(1, 8);
var_grass_8D    = zeros(1, 8);

% calculate means and variances of cheetah and grass
```

```
for i = 1 : 8
    col = TrainsampleDCT_FG(:,i);
    mean_cheetah__8D(1, i) = mean(col);
    var_cheetah__8D(1, i) = var(col);
    col = TrainsampleDCT_BG(:,i);
    mean_grass_8D(1, i) = mean(col);
    var_grass_8D(1, i) = var(col);
end

% DCT (64D)
[m, n] = size(img_cheetah);
mask_8D = zeros(m - 7, n - 7);

% padding the image
I = zeros(m + 7, n + 7);
I(1:m, 1:n) = img_cheetah;

% calculate the sigmas and sigma^-1s
sigma_cheetah_8D    = cov(TrainsampleDCT_FG(:, 8));
sigma_grass_8D      = cov(TrainsampleDCT_BG(:, 8));
sigmaInv_cheetah_8D = inv(sigma_cheetah_8D);
sigmaInv_grass_8D   = inv(sigma_grass_8D);

% calculate the coefficients
deno_cheetah_8D = (sqrt(((2 * pi)^8) * det(sigma_cheetah_8D)))
    ;
deno_grass_8D   = (sqrt(((2 * pi)^8) * det(sigma_grass_8D)));

% calculate DCT2 (64D)
m = m - 7;
n = n - 7;
for i = 1 : m
    for j = 1 : n
        Block = I(i : i + 7, j : j + 7);
        Block_DCT = dct2(Block, 8, 8);
        V = Block_DCT(:).';
        X = zeros(1, 64);
        % mapping
        for k = 1 : 64
```

```
                X(ZigZagV(k)) = V(k);
            end
            % calculate the probabilities
            P_X_cheetah = exp(-0.5 * (X(1:8) - mean_cheetah_8D) * (sigmaInv_cheetah_8D) * (X(1:8) - mean_cheetah_8D).') / deno_cheetah_8D;
            P_X_grass   = exp(-0.5 * (X(1:8) - mean_grass_8D)   * (sigmaInv_grass_8D)   * (X(1:8) - mean_grass_8D).') / deno_grass_8D;
            % based on the results, decide whether the pixel is grass or cheetah
            if (P_X_grass > P_X_cheetah)
                mask_8D(i, j) = 0;
            else
                mask_8D(i, j) = 1;
            end
        end
end

% compare my mask with ans
e_8D = 0;
for i = 1 : m
    for j = 1 : n
        if (mask_ans(i, j) ~= mask_8D(i, j))
            e_8D = e_8D + 1;
        end
    end
end
error_rate_8D = e_8D / (m * n);

% plot the result
subplot(1, 3, 3);
imshow(mask_8D);
hold on
title('Mask (8 Features)')
figure

imshow(mask_64D);
title('Mask (64 Features)')
```

```
figure
imshow(mask_8D);
title('Mask (8 Features)')
```