

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

```
1 clear
2 clc
3 clf
4
5 % import data
6 load('HW2\TrainingSamplesDCT_8_new.mat');
7 M_cheetah = TrainsampleDCT_FG;
8 M_grass = TrainsampleDCT_BG;
9
10 % calculate the number of appearance
11 [m_cheetah, ~] = size(M_cheetah);
12 [m_grass, ~] = size(M_grass);
13
14 % calculate the probabilities
```

```

15 P_Y_cheetah = m_cheetah / (m_grass + m_cheetah);
16 P_Y_cheetah = round(P_Y_cheetah, 2);
17 P_Y_grass = m_grass / (m_grass + m_cheetah);
18 P_Y_grass = round(P_Y_grass, 2);
19
20 % plot the histogram
21 X = categorical({'Cheetah', 'Grass'});
22 Y = [P_Y_cheetah P_Y_grass];
23 bar(X, Y)
24 text(1:length(Y), Y, num2str(Y'), 'vert', 'bottom', 'horiz', 'center
    ');

```

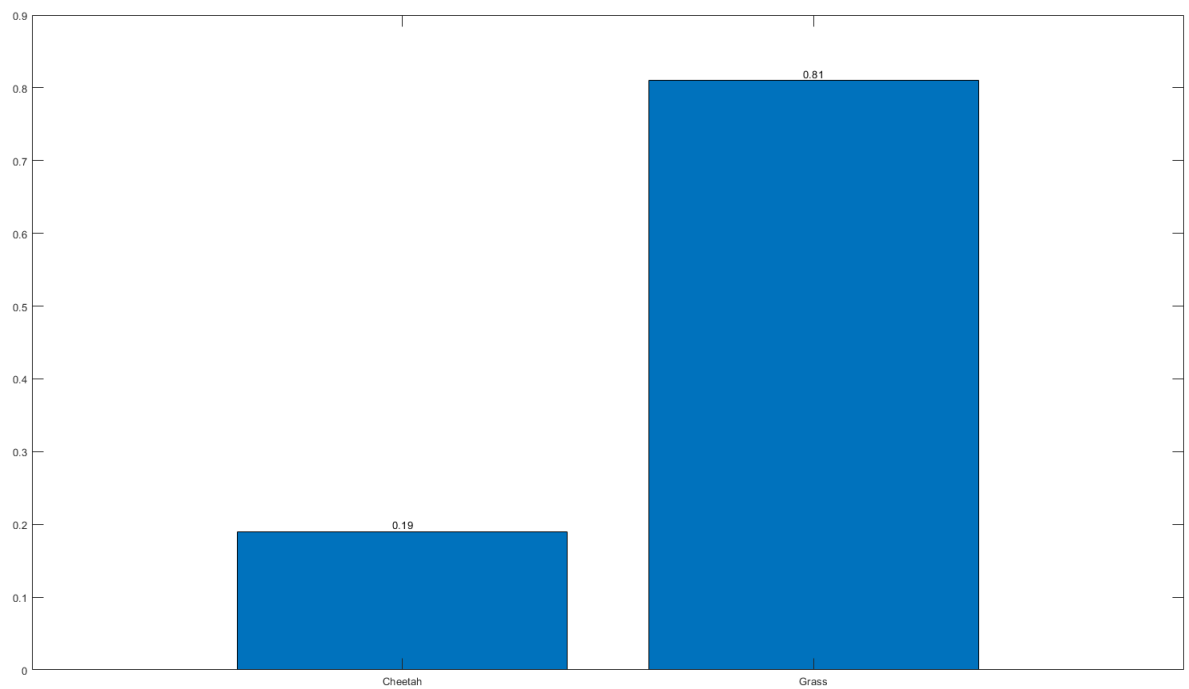


Figure 1: 64 Features

- (b) Using the training data in TrainingSamplesDCT 8.mat, compute the maximum likelihood estimates for the parameters of the class conditional densities $P_{X|Y}(x | cheetah)$ and $P_{X|Y}(x | grass)$ under the Gaussian assumption. Denoting by $X = \{X_1, X_2, X_3, \dots, X_n\}$ the vector of DCT coefficients, create 64 plots with the marginal densities for the two classes $P_{X_k|Y}(x_k | cheetah)$ and $P_{X_k|Y}(x_k | grass)$, $k = 1, 2, \dots, 64$ - 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

```
1 clear
2 clc
3 clf
4
5 load("HW2\TrainingSamplesDCT_8_new.mat");
6
7 % creat vectors for means and variances
8 mean_cheetah = zeros(1, 64);
9 var_cheetah = zeros(1, 64);
10 mean_grass = zeros(1, 64);
11 var_grass = zeros(1, 64);
12
13 % calculate means and variances of cheetah and grass
14 for indx = 1 : 64
15     col = TrainsampleDCT_FG(:, indx);
16     mean_cheetah(1, indx) = mean(col);
```

```

17     var_cheetah(1, indx) = var(col);
18     col = TrainsampleDCT_BG(:,indx);
19     mean_grass(1, indx) = mean(col);
20     var_grass(1, indx) = var(col);
21 end
22
23 % create the x-axis and y-axis for cheetah and grass
24 x_cheetah = -1 : 0.01 : 5;
25 x_grass = x_cheetah;
26
27 deno_cheetah = (sqrt(var_cheetah(1) * 2 * pi));
28 deno_grass = (sqrt(var_grass(1) * 2 * pi));
29 pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(1)).^2 /
    var_cheetah(1));
30 pow_grass = -0.5 * ((x_cheetah - mean_grass(1)).^2 /
    var_cheetah(1));
31 vec_cheetah = exp(pow_cheetah) / deno_cheetah;
32 vec_grass = exp(pow_grass) / deno_grass;
33 subplot(8, 8, 1);
34 plot(x_cheetah, vec_cheetah);
35 hold on
36 plot(x_grass, vec_grass);
37 delta = mean_cheetah(1) - mean_grass(1);
38 text = "\delta = " + round(abs(delta), 4);
39 subtitle(text)
40
41 for indx = 2 : 64
42     x_cheetah = mean_cheetah(indx) - 0.25 : 0.001 :
        mean_cheetah(indx) + 0.25;
43     x_grass = x_cheetah;
44     co_cheetah = 1 / (sqrt(var_cheetah(indx) * 2 * pi));
45     co_grass = 1 / (sqrt(var_grass(indx) * 2 * pi));
46     pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(indx)).^2
        / var_cheetah(indx));
47     pow_grass = -0.5 * ((x_cheetah - mean_grass(indx)).^2 /
        var_cheetah(indx));
48     vec_cheetah = exp(pow_cheetah) * co_cheetah;
49     vec_grass = exp(pow_grass) * co_grass;
50     subplot(8, 8, indx);

```

```

51     plot(x_cheetah, vec_cheetah);
52     hold on
53     plot(x_grass, vec_grass);
54     delta = mean_cheetah(indx) - mean_grass(indx);
55     text = "\delta = " + round(abs(delta), 4);
56     subtitle(text)
57 end
58 sgtitle('64 Features')
59 figure
60
61 % extract the best 8 features
62 cnt = 2;
63 desire_best = 1 : 1 : 8;
64 for i = 2 : 8
65     indx = desire_best(i);
66     x_cheetah = mean_cheetah(indx) - 0.5 : 0.001 :
        mean_cheetah(indx) + 0.5;
67     x_grass = x_cheetah;
68     co_cheetah = 1 / (sqrt(var_cheetah(indx) * 2 * pi));
69     co_grass = 1 / (sqrt(var_grass(indx) * 2 * pi));
70     pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(indx)).^2
        / var_cheetah(indx));
71     pow_grass = -0.5 * ((x_cheetah - mean_grass(indx)).^2 /
        var_cheetah(indx));
72     vec_cheetah = exp(pow_cheetah) * co_cheetah;
73     vec_grass = exp(pow_grass) * co_grass;
74     subplot(4, 2, cnt);
75     plot(x_cheetah, vec_cheetah);
76     hold on
77     plot(x_grass, vec_grass);
78     delta = mean_cheetah(indx) - mean_grass(indx);
79     text = "\delta = " + round(abs(delta), 4);
80     subtitle(text)
81     cnt = cnt + 1;
82 end
83 x_cheetah = -1 : 0.01 : 5;
84 x_grass = x_cheetah;
85
86 deno_cheetah = (sqrt(var_cheetah(1) * 2 * pi));

```

```

87 deno_grass = (sqrt(var_grass(1) * 2 * pi));
88 pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(1)).^2 /
    var_cheetah(1));
89 pow_grass = -0.5 * ((x_cheetah - mean_grass(1)).^2 /
    var_cheetah(1));
90 vec_cheetah = exp(pow_cheetah) / deno_cheetah;
91 vec_grass = exp(pow_grass) / deno_grass;
92 subplot(4, 2, 1);
93 plot(x_cheetah, vec_cheetah);
94 hold on
95 plot(x_grass, vec_grass);
96 delta = mean_cheetah(1) - mean_grass(1);
97 text = "\delta = " + round(abs(delta), 4);
98 subtitle(text)
99 sgtitle('Best 8 Features')
100 figure
101
102 % extract the worst 8 features
103 cnt = 1;
104 desire_worst = [37 40 51 59 60 62 63 64];
105 for i = 1 : 8
106     indx = desire_worst(i);
107     x_cheetah = mean_cheetah(indx) - 0.05 : 0.001 :
        mean_cheetah(indx) + 0.05;
108     x_grass = x_cheetah;
109     co_cheetah = 1 / (sqrt(var_cheetah(indx) * 2 * pi));
110     co_grass = 1 / (sqrt(var_grass(indx) * 2 * pi));
111     pow_cheetah = -0.5 * ((x_cheetah - mean_cheetah(indx)).^2
        / var_cheetah(indx));
112     pow_grass = -0.5 * ((x_cheetah - mean_grass(indx)).^2 /
        var_cheetah(indx));
113     vec_cheetah = exp(pow_cheetah) * co_cheetah;
114     vec_grass = exp(pow_grass) * co_grass;
115     subplot(4, 2, cnt);
116     plot(x_cheetah, vec_cheetah);
117     hold on
118     plot(x_grass, vec_grass);
119     delta = mean_cheetah(indx) - mean_grass(indx);
120     text = "\delta = " + round(abs(delta), 4);

```

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

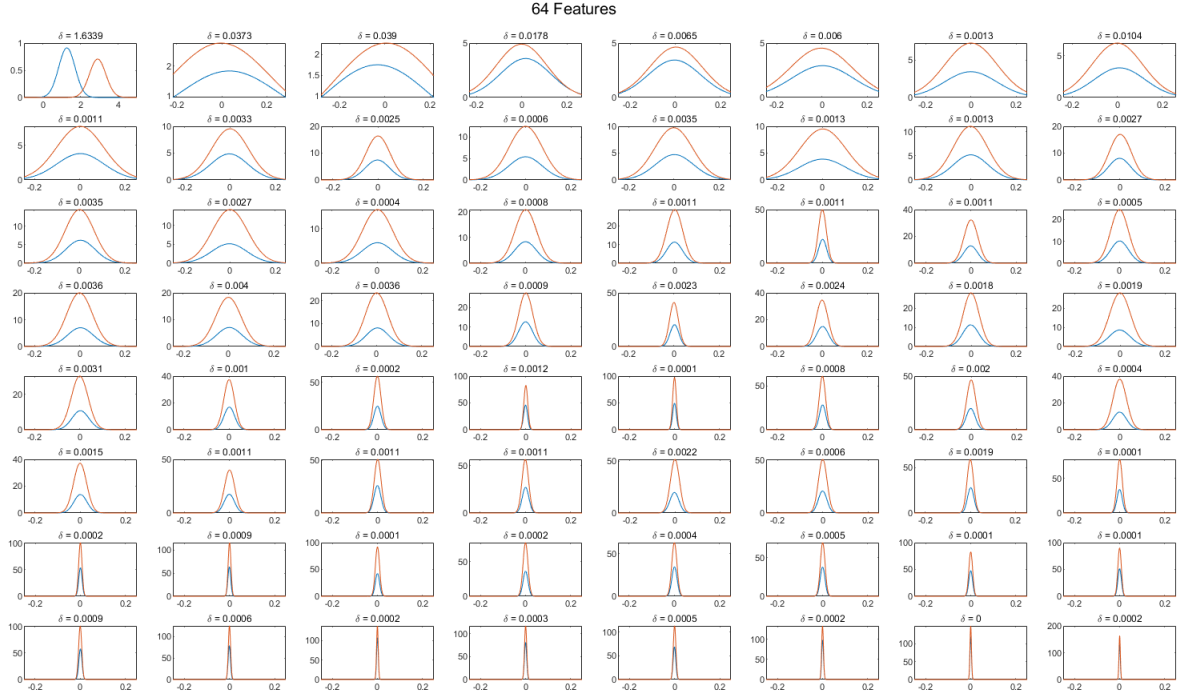


Figure 2: 64 Features

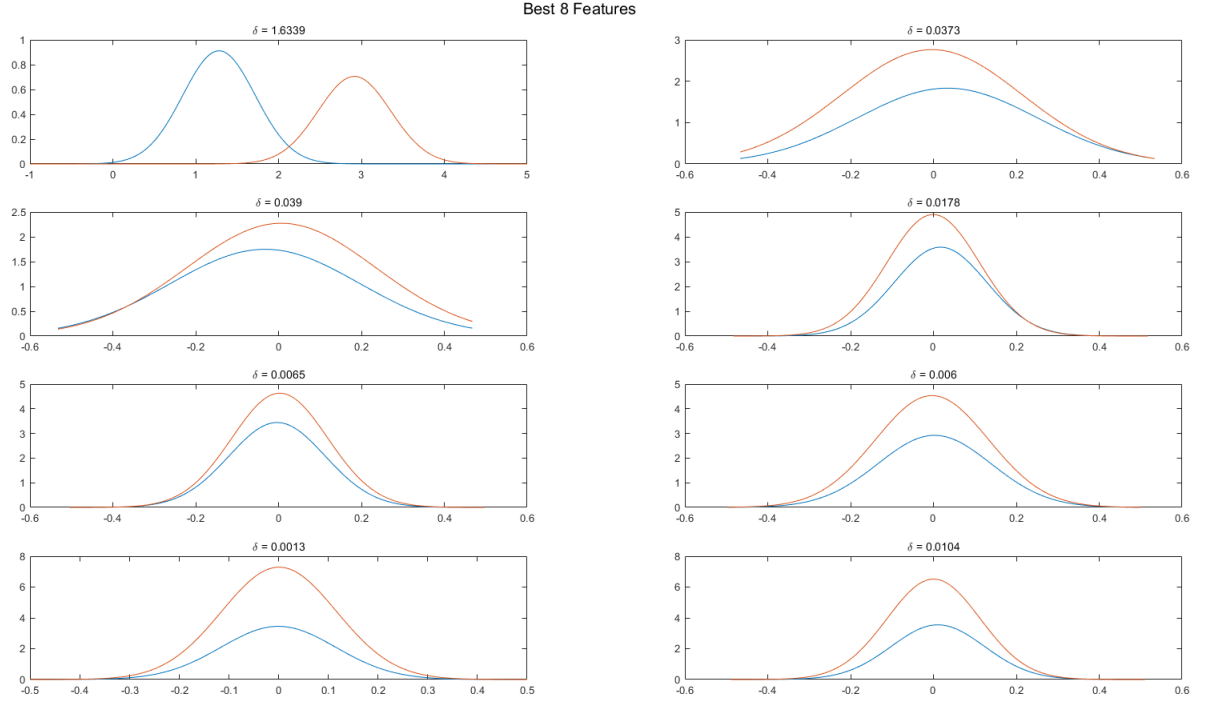


Figure 3: Best 8 Features

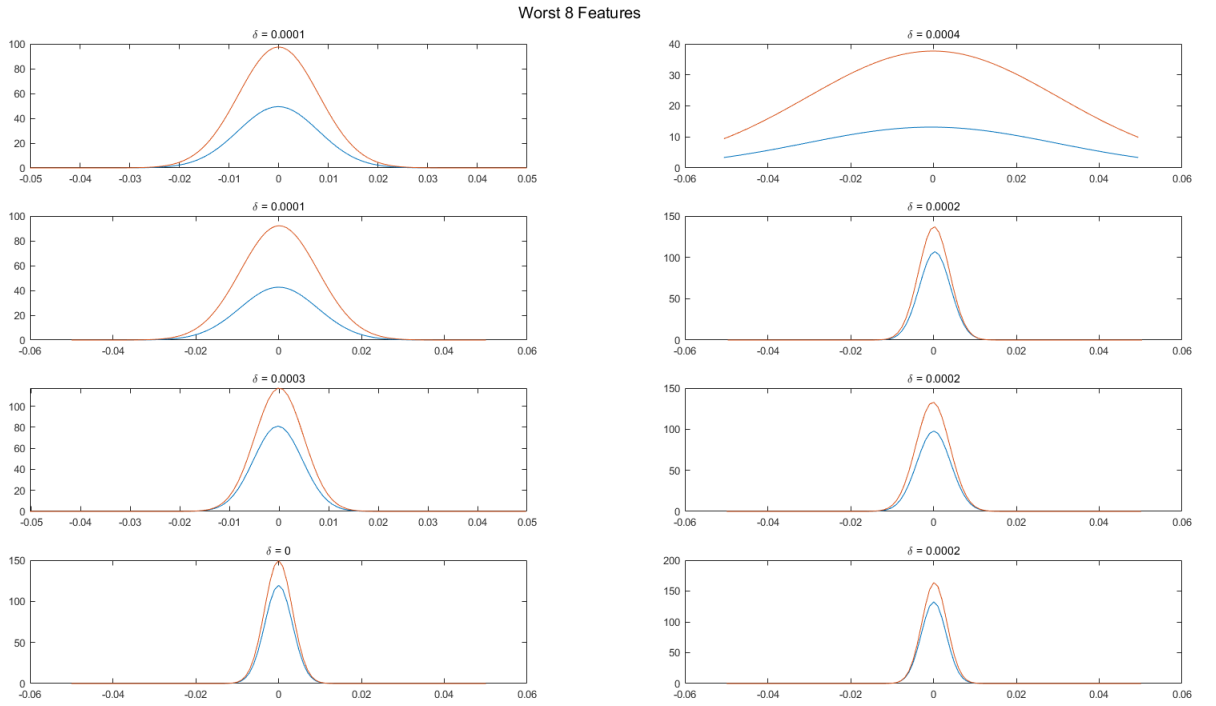


Figure 4: Worst 8 Features

- (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 from 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 easily distinguish **cheetah** from **grass**; however, if the value of δ is small, then we cannot tell them apart.

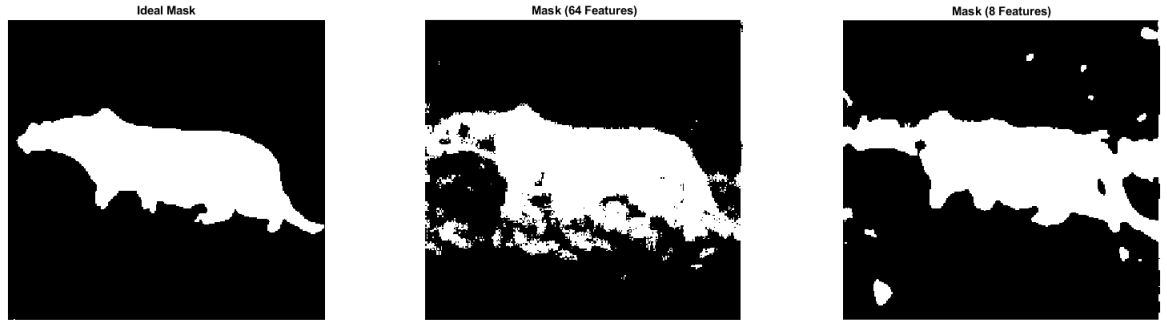


Figure 5: Side-by-side comparison between three masks

Matlab Code

```
1 clear
2 clc
3
4 load("HW2\TrainingSamplesDCT_8_new.mat");
5 img_cheetah = imread("cheetah.bmp");
6 mask_ans = imread("cheetah_mask.bmp");
7 img_cheetah = im2double(img_cheetah);
8 mask_ans = imbinarize(mask_ans);
9
10 subplot(1, 3, 1);
11 imshow(mask_ans);
12 hold on
13 title('Ideal Mask')
14
15 % creat vectors for means and variances
16 mean_cheetah_64D = zeros(1, 64);
17 var_cheetah_64D = zeros(1, 64);
18 mean_grass_64D = zeros(1, 64);
19 var_grass_64D = zeros(1, 64);
20
21 % calculate means and variances of cheetah and grass
22 for i = 1 : 64
23     col = TrainsampleDCT_FG(:, i);
24     mean_cheetah_64D(1, i) = mean(col);
25     var_cheetah_64D(1, i) = var(col);
26     col = TrainsampleDCT_BG(:, i);
27     mean_grass_64D(1, i) = mean(col);
28     var_grass_64D(1, i) = var(col);
29 end
30
31 % DCT (64D)
32 [m, n] = size(img_cheetah);
33 mask_64D = zeros(m - 7, n - 7);
34
35 % padding the image
36 I = zeros(m + 7, n + 7);
37 I(1:m, 1:n) = img_cheetah;
```

```

38
39 % the zigzag map
40 ZigZagM = [0    1    5    6   14   15   27   28;
41           2    4    7   13   16   26   29   42;
42           3    8   12   17   25   30   41   43;
43           9   11   18   24   31   40   44   53;
44          10   19   23   32   39   45   52   54;
45          20   22   33   38   46   51   55   60;
46          21   34   37   47   50   56   59   61;
47          35   36   48   49   57   58   62   63];
48 ZigZagM = ZigZagM + 1; % index in MATLAB starts with 1
49 ZigZagV = ZigZagM(:); % 64 x 1
50 ZigZagV = ZigZagV.'; % 1 x 64
51
52 % calculate the sigmas and sigma^-1s
53 sigma_cheetah_64D = cov(TrainsampleDCT_FG);
54 sigma_grass_64D = cov(TrainsampleDCT_BG);
55 sigmaInv_cheetah_64D = inv(sigma_cheetah_64D);
56 sigmaInv_grass_64D = inv(sigma_grass_64D);
57
58 % calculate the coefficients
59 deno_cheetah = (sqrt(((2 * pi)^64) * det(sigma_cheetah_64D)));
60 deno_grass = (sqrt(((2 * pi)^64) * det(sigma_grass_64D)));
61
62 % calculate DCT2 (64D)
63 m = m - 7;
64 n = n - 7;
65 for i = 1 : m
66     for j = 1 : n
67         Block = I(i : i + 7, j : j + 7);
68         Block_DCT = dct2(Block, 8, 8);
69         V = Block_DCT(:).';
70         X = zeros(1, 64);
71         % mapping
72         for k = 1 : 64
73             X(ZigZagV(k)) = V(k);
74         end
75     % calculate the probabilities
76     P_X_cheetah = exp(-0.5 * (X - mean_cheetah_64D) * (

```

```

        sigmaInv_cheetah_64D) * (X - mean_cheetah_64D).') /
        deno_cheetah;
77     P_X_grass    = exp(-0.5 * (X - mean_grass_64D)    * (
        sigmaInv_grass_64D)    * (X - mean_grass_64D).')    /
        deno_grass;
78     % based on the results , decide whether the pixel is
        grass or cheetah
79     if (P_X_grass > P_X_cheetah)
80         mask_64D(i, j) = 0;
81     else
82         mask_64D(i, j) = 1;
83     end
84 end
85 end
86
87 % compare my mask with ans
88 e_64D = 0;
89 for i = 1 : m
90     for j = 1 : n
91         if (mask_ans(i, j) ~= mask_64D(i, j))
92             e_64D = e_64D + 1;
93         end
94     end
95 end
96 error_rate_64D = e_64D / (m * n);
97
98 % plot the result
99 subplot(1, 3, 2);
100 imshow(mask_64D);
101 hold on
102 title('Mask (64 Features)')
103
104 % creat vectors for means and variances (8D)
105 mean_cheetah_8D = zeros(1, 8);
106 var_cheetah_8D  = zeros(1, 8);
107 mean_grass_8D   = zeros(1, 8);
108 var_grass_8D    = zeros(1, 8);
109
110 % calculate means and variances of cheetah and grass

```

```

111 for i = 1 : 8
112     col = TrainsampleDCT_FG(:, i);
113     mean_cheetah_8D(1, i) = mean(col);
114     var_cheetah_8D(1, i) = var(col);
115     col = TrainsampleDCT_BG(:, i);
116     mean_grass_8D(1, i) = mean(col);
117     var_grass_8D(1, i) = var(col);
118 end
119
120 % DCT (64D)
121 [m, n] = size(img_cheetah);
122 mask_8D = zeros(m - 7, n - 7);
123
124 % padding the image
125 I = zeros(m + 7, n + 7);
126 I(1:m, 1:n) = img_cheetah;
127
128 % calculate the sigmas and sigma^-1s
129 sigma_cheetah_8D = cov(TrainsampleDCT_FG(:, 8));
130 sigma_grass_8D = cov(TrainsampleDCT_BG(:, 8));
131 sigmaInv_cheetah_8D = inv(sigma_cheetah_8D);
132 sigmaInv_grass_8D = inv(sigma_grass_8D);
133
134 % calculate the coefficients
135 deno_cheetah_8D = (sqrt(((2 * pi)^8) * det(sigma_cheetah_8D)))
136     ;
137 deno_grass_8D = (sqrt(((2 * pi)^8) * det(sigma_grass_8D)));
138
139 % calculate DCT2 (64D)
140 m = m - 7;
141 n = n - 7;
142 for i = 1 : m
143     for j = 1 : n
144         Block = I(i : i + 7, j : j + 7);
145         Block_DCT = dct2(Block, 8, 8);
146         V = Block_DCT(:) .';
147         X = zeros(1, 64);
148         % mapping
149         for k = 1 : 64

```

```

149         X(ZigZagV(k)) = V(k);
150     end
151     % calculate the probabilities
152     P_X_cheetah = exp(-0.5 * (X(1:8) - mean_cheetah_8D) *
        (sigmaInv_cheetah_8D) * (X(1:8) - mean_cheetah_8D)
        .') / deno_cheetah_8D;
153     P_X_grass = exp(-0.5 * (X(1:8) - mean_grass_8D) *
        (sigmaInv_grass_8D) * (X(1:8) - mean_grass_8D).')
        / deno_grass_8D;
154     % based on the results, decide whether the pixel is
        grass or cheetah
155     if (P_X_grass > P_X_cheetah)
156         mask_8D(i, j) = 0;
157     else
158         mask_8D(i, j) = 1;
159     end
160 end
161 end
162
163 % compare my mask with ans
164 e_8D = 0;
165 for i = 1 : m
166     for j = 1 : n
167         if (mask_ans(i, j) ~= mask_8D(i, j))
168             e_8D = e_8D + 1;
169         end
170     end
171 end
172 error_rate_8D = e_8D / (m * n);
173
174 % plot the result
175 subplot(1, 3, 3);
176 imshow(mask_8D);
177 hold on
178 title('Mask (8 Features)')
179 figure
180
181 imshow(mask_64D);
182 title('Mask (64 Features)')

```

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