ECE-271A

Haonan Peng

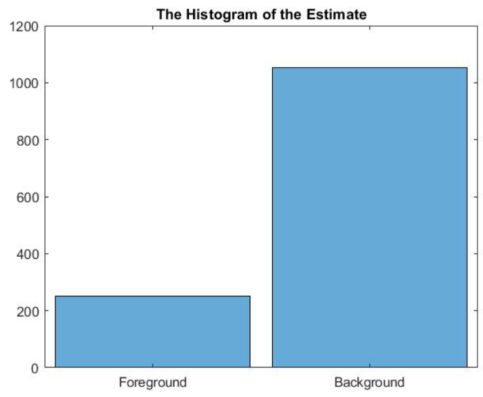
A14765890

HW2 Report

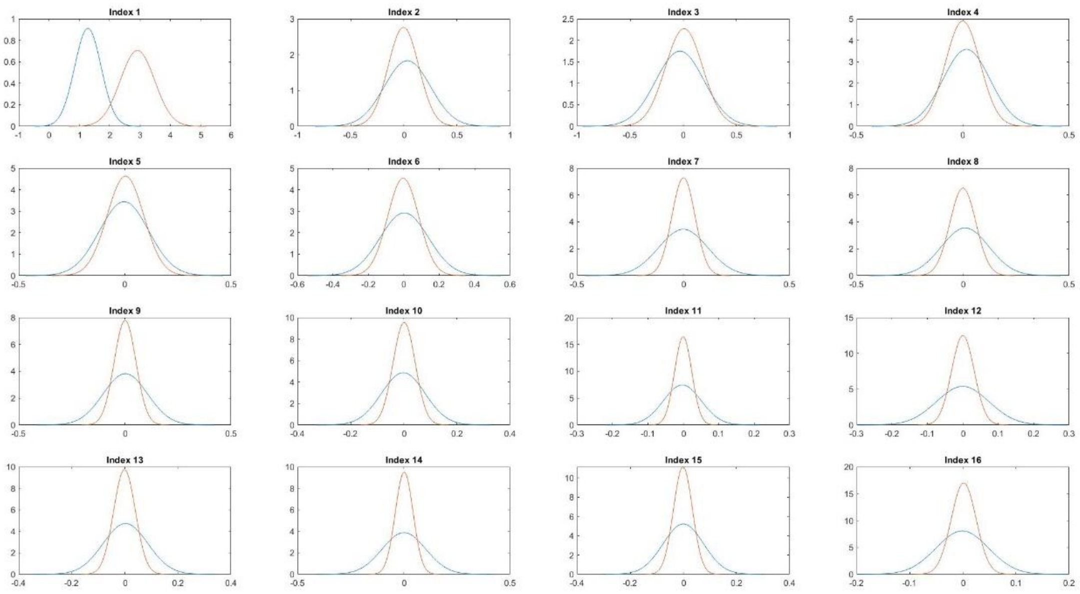
a. From problem 2, the prior probability can be calculated by $P_y(i) = \frac{C_i}{n}$, where C_i is the number of samples in the specific class, and n is the total number of all samples in this question. Then we have:

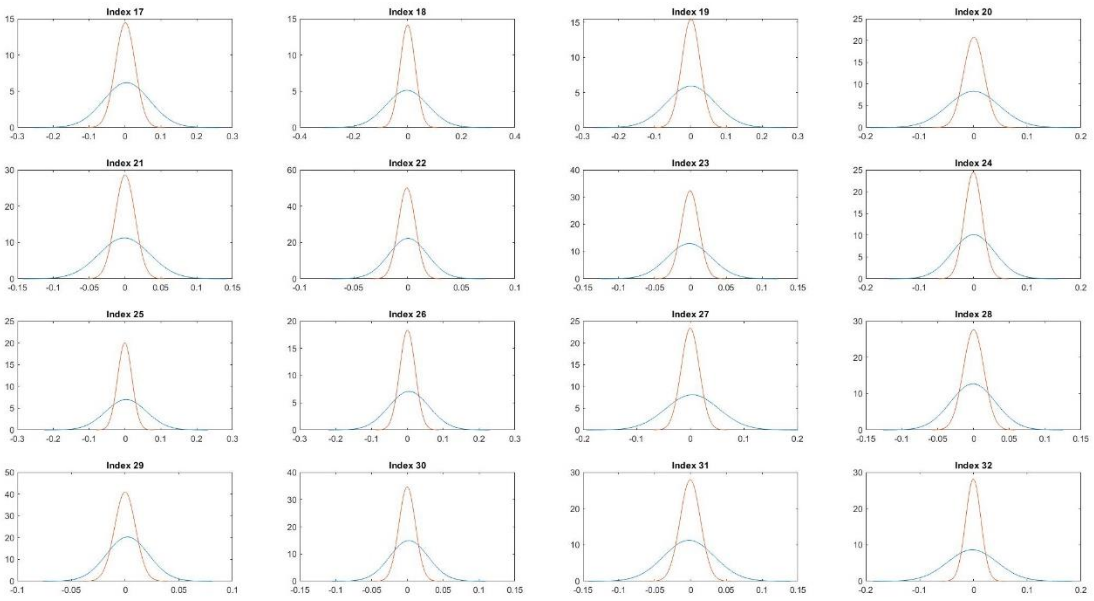
   1. $P_Y(Cheetah) = 0.1919$
   2. $P_Y(Grass) = 0.8081$

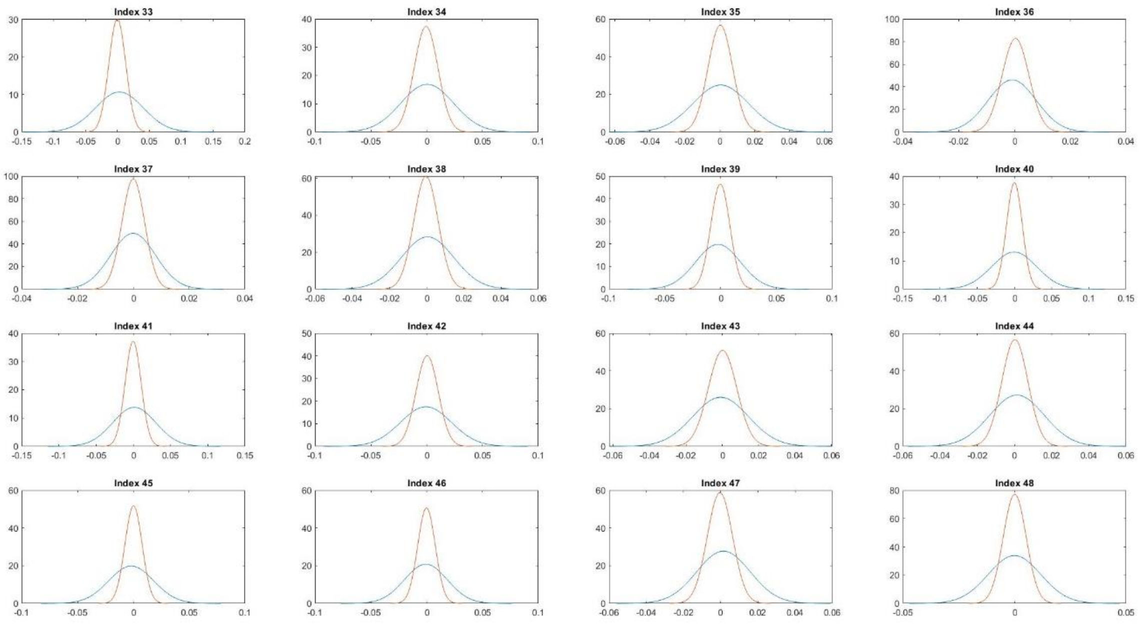
   The calculated results are the same as the results we obtained last week.

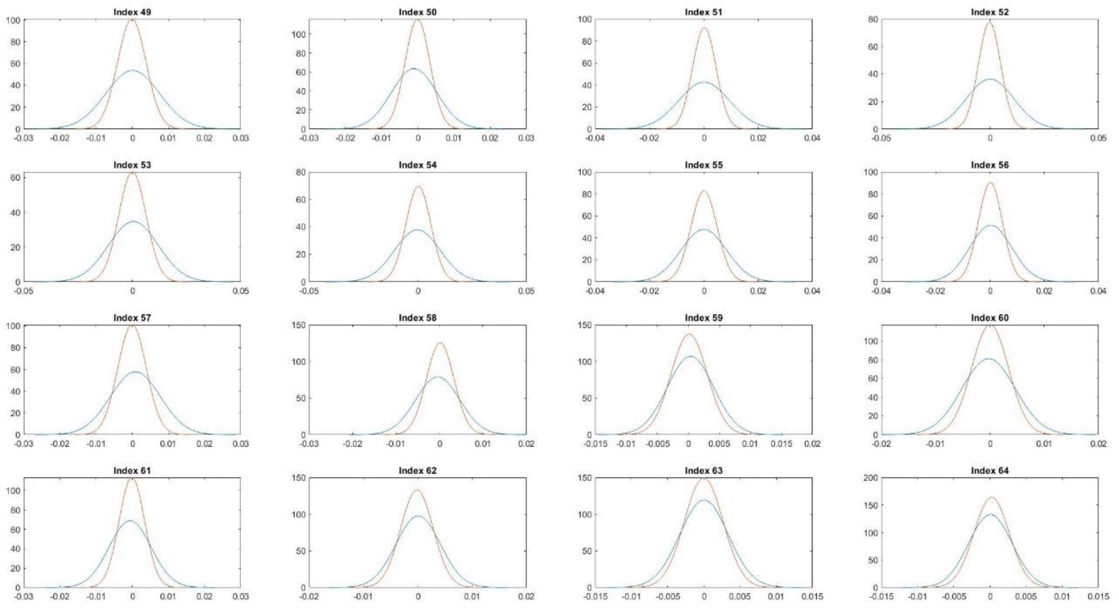


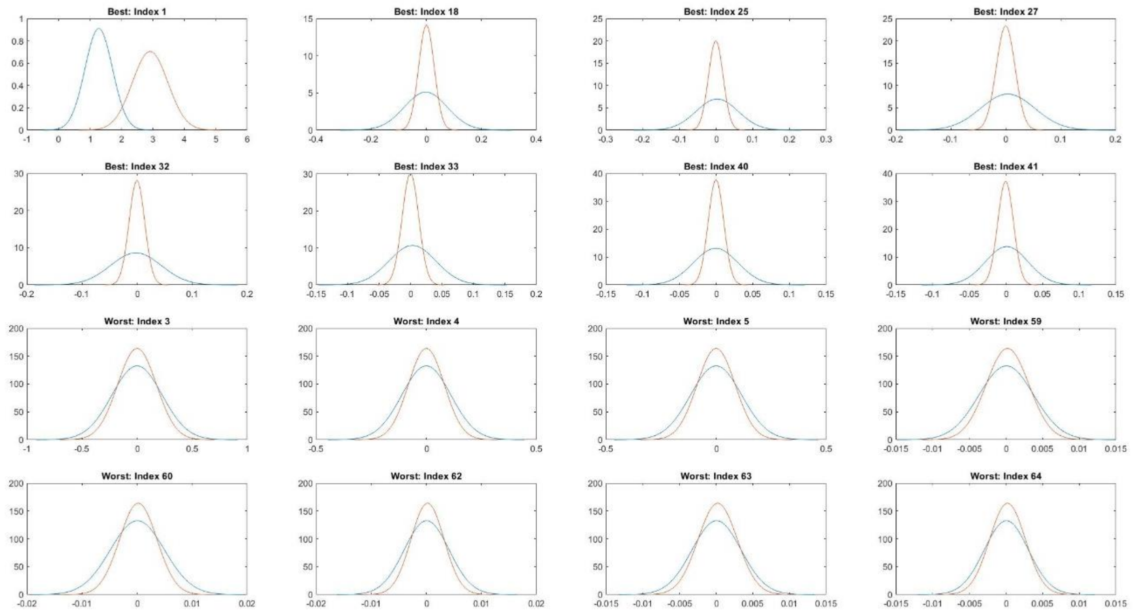
b. Here are the 64 marginal densities for the two classes $P_{X|Y}(Cheetah)\ and\ P_{X|Y}(Grass)$ under Gaussian assumption. (The blue plots are features of the class of foreground, and the red plots are the features of the class of background)


Index 1
Index 2
Index 3
Index 4
Index 5
Index 6
Index 7
Index 8
Index 9
Index 10
Index 11
Index 12
Index 13
Index 14
Index 15
Index 16


Index 17
Index 18
Index 19
Index 20
Index 21
Index 22
Index 23
Index 24
Index 25
Index 26
Index 27
Index 28
Index 29
Index 30
Index 31
Index 32

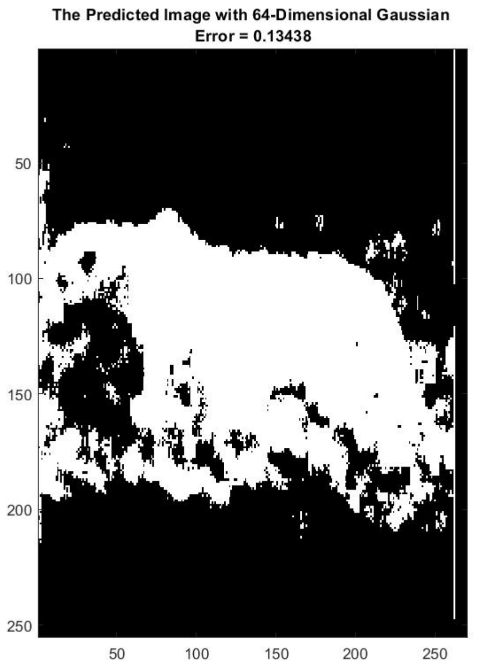

Index 33
Index 34
Index 35
Index 36
Index 37
Index 38
Index 39
Index 40
Index 41
Index 42
Index 43
Index 44
Index 45
Index 46
Index 47
Index 48

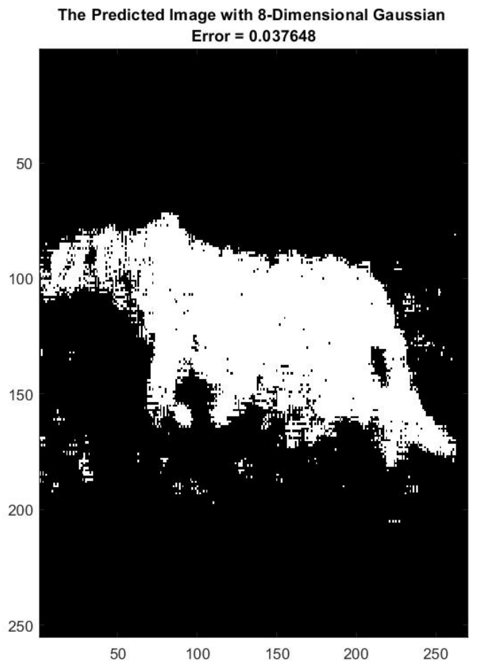

Index 49
Index 50
Index 51
Index 52
Index 53
Index 54
Index 55
Index 56
Index 57
Index 58
Index 59
Index 60
Index 61
Index 62
Index 63
Index 64



By visual inspection, the best 8 features for classification are: [1,18,25,27,32,33,40,41]. The worst 8 features are: [3,4,5,59,60,62,63,64].

c. The two figures shown below are the prediction image of the "Cheetah" by using 1) all 64-dimensional Gaussian, and 2) the best 8-dimensional Gaussian.





Comparing to both predictions of the cheetah image, the performance of the 8-dimensional Gaussian with $error = 0.037648$ is obviously better than the one of the 64-dimensional

Gaussian with $error = 0.13438$. When counting the probabilities of features other than the best 8 features, the performance becomes worse, because the probabilities of two classes of the features are so close that can misguide the Bayesian decision by using $P_{X|Y}(Cheetah) \times P_Y(Cheetah) > P_{X|Y}(Grass) \times P_Y(Grass)$

Code

```
%% a)
samples= load('TrainingSamplesDCT_8_new.mat');

FG = samples.TrainsampleDCT_FG;
BG = samples.TrainsampleDCT_BG;

PY_FG = size(FG,1) / (size(FG,1)+size(BG,1));
PY_BG = size(BG,1) / (size(FG,1)+size(BG,1));

h = [ones(size(FG,1),1);zeros(size(BG,1),1)];
hist = categorical(h,[1 0],{'Foreground','Background'});

figure(1)
histogram(hist);
title('The Histogram of the Estimate');
%% b)
mean_FG = mean(FG);
mean_BG = mean(BG);

cov_FG = cov(FG);
cov_BG = cov(BG);

MLE_FG = zeros(1,size(FG,1));
MLE_BG = zeros(1,size(BG,1));

% generate all 64 gaussian probability densities
for i = 1:4
    figure(i+1)
    for j = 1:16
        sig_FG = sqrt(cov_FG((i-1)*16+j,(i-1)*16+j));
        sig_BG = sqrt(cov_BG((i-1)*16+j,(i-1)*16+j));

        m_FG = mean_FG((i-1)*16+j);
        m_BG = mean_BG((i-1)*16+j);

        x_FG = (m_FG-4*sig_FG):8*sig_FG/199:(m_FG+4*sig_FG);
        x_BG = (m_BG-4*sig_BG):8*sig_BG/199:(m_BG+4*sig_BG);

        md_FG = exp(-(2*sig_FG^2)^(-1)*(x_FG-m_FG).^2)/(sig_FG*sqrt(2*pi));
        md_BG = exp(-(2*sig_BG^2)^(-1)*(x_BG-m_BG).^2)/(sig_BG*sqrt(2*pi));

        subplot(4,4,j);
        plot(x_FG,md_FG);
        hold on
        plot(x_BG,md_BG);
        hold off
        title(['Index ' num2str((i-1)*16+j)]);
    end
end

best = [1,18,25,27,32,33,40,41];
worst = [3,4,5,59,60,62,63,64];
```

```
% generate 8 best gaussian probability densities and 8 worst ones
figure(6)
for i = 1:8
    sig_FG_b = sqrt(cov_FG(best(i),best(i)));
    sig_BG_b = sqrt(cov_BG(best(i),best(i)));

    sig_FG_w = sqrt(cov_FG(worst(i),worst(i)));
    sig_BG_w = sqrt(cov_BG(worst(i),worst(i)));

    m_FG_b = mean_FG(best(i));
    m_BG_b = mean_BG(best(i));

    m_FG_w = mean_FG(worst(i));
    m_BG_w = mean_BG(worst(i));

    x_FG_b = (m_FG_b-4*sig_FG_b):8*sig_FG_b/199:(m_FG_b+4*sig_FG_b);
    x_BG_b = (m_BG_b-4*sig_BG_b):8*sig_BG_b/199:(m_BG_b+4*sig_BG_b);

    x_FG_w = (m_FG-4*sig_FG_w):8*sig_FG_w/199:(m_FG+4*sig_FG_w);
    x_BG_w = (m_BG-4*sig_BG_w):8*sig_BG_w/199:(m_BG+4*sig_BG_w);

    md_FG_b = exp(-(2*sig_FG_b^2)^(-1)*(x_FG_b-m_FG_b).^2)/(sig_FG_b*sqrt(2*pi));
    md_BG_b = exp(-(2*sig_BG_b^2)^(-1)*(x_BG_b-m_BG_b).^2)/(sig_BG_b*sqrt(2*pi));

    md_FG_w = exp(-(2*sig_FG^2)^(-1)*(x_FG-m_FG).^2)/(sig_FG*sqrt(2*pi));
    md_BG_w = exp(-(2*sig_BG^2)^(-1)*(x_BG-m_BG).^2)/(sig_BG*sqrt(2*pi));

    subplot(4,4,i);
    plot(x_FG_b,md_FG_b);
    hold on
    plot(x_BG_b,md_BG_b);
    hold off
    title(['Best: Index ' num2str(best(i))]);

    subplot(4,4,i+8);
    plot(x_FG_w,md_FG_w);
    hold on
    plot(x_BG_w,md_BG_w);
    hold off
    title(['Worst: Index ' num2str(worst(i))]);
end

%% c.
img= im2double(imread('cheetah.bmp'));
[row, colm] = size(img);

A_64 = zeros(row,colm);
A_8 = zeros(row,colm);

%read Zig-Zag Pattern.txt file
ZigZag = fopen('Zig-Zag Pattern.txt','r');
zzPat = fscanf(ZigZag,'%d',[8,8]);
fclose(ZigZag);

cov_FG8 = cov(FG(:,best));
```

```
cov_BG8 = cov(BG(:,best));

mean_FG8 = mean_FG(best);
mean_BG8 = mean_BG(best);

% using all 64 features
for i = 1:row-8
    for j = 1:colm-8
        dctImg = dct2(img(i:i+7,j:j+7));
        zzScan= zeros([1, 64]);
        for x = 1:8
            for y = 1:8
                zzScan(zzPat(x,y)+1) = dctImg(x,y);
            end
        end
        PX_FG = sqrt((2*pi)^64*det(cov_FG))^(-1)*exp(-(zzScan-
mean_FG)/cov_FG*(zzScan-mean_FG)'/2);
        PX_BG = sqrt((2*pi)^64*det(cov_BG))^(-1)*exp(-(zzScan-
mean_BG)/cov_BG*(zzScan-mean_BG)'/2);

        PX_FG8 = sqrt((2*pi)^8*det(cov_FG8))^(-1)*exp(-(zzScan(best)-
mean_FG8)/cov_FG8*(zzScan(best)-mean_FG8)'/2);
        PX_BG8 = sqrt((2*pi)^8*det(cov_BG8))^(-1)*exp(-(zzScan(best)-
mean_BG8)/cov_BG8*(zzScan(best)-mean_BG8)'/2);

        if PX_FG*PY_FG > PX_BG*PY_BG
            A_64(i,j) = 1;
        end

        if PX_FG8*PY_FG > PX_BG8*PY_BG
            A_8(i,j) = 1;
        end
    end
end

ground_truth = im2double(imread('cheetah_mask.bmp'));
pred = padarray(A_64, [4,4], 0);
pred8 = padarray(A_8, [4,4], 0);

missFG = 0;
missBG = 0;

missFG8 = 0;
missBG8 = 0;

gtFG = 0;
gtBG = 0;

for i = 1:size(ground_truth,1)
    for j = 1:size(ground_truth,2)
        if ground_truth(i,j) == 1
            gtFG = gtFG + 1;
            if pred(i,j) ~= ground_truth(i,j)
                missFG = missFG + 1;
            end
            if pred8(i,j) ~= ground_truth(i,j)
                missFG8 = missFG8 + 1;
            end
        else
            gtBG = gtBG + 1;
            if pred(i,j) ~= ground_truth(i,j)
                missBG = missBG + 1;
            end
            if pred8(i,j) ~= ground_truth(i,j)
                missBG8 = missBG8 + 1;
            end
        end
    end
end

% Calculate error
errFG = missFG / gtFG * PY_FG;
errBG = missBG / gtBG * PY_BG;

errFG8 = missFG8 / gtFG * PY_FG;
errBG8 = missBG8 / gtBG * PY_BG;

err = errFG + errBG;
err8 = errFG8 + errBG8;

figure
subplot(1,2,1)
imagesc(A_64);
colormap(gray(255));
title({['The Predicted Image with 64-Dimensional Gaussian '] ['Error = '
num2str(err)]});
subplot(1,2,2)
imagesc(A_8);
colormap(gray(255));
title({['The Predicted Image with 8-Dimensional Gaussian '] ['Error = '
num2str(err8)]});
```