ECE 271A: Statistical Learning I

Homework 1

Shang-Wei Hung
A53267981

Problem 5

(a) Prior probabilities can be estimated from the given training data in *TrainingSampleDCT_8.mat.* The numbers of rows for *TrainsampleDCT_FG* and *TrainsampleDCT_BG* are 250 and 1053, respectively. Thus, the prior probabilites can be estimated as follow:

$$P_Y(cheetach) = \frac{250}{250 + 1053} = 0.1919$$

$$P_Y(grass) = \frac{1053}{250 + 1053} = 0.8081$$

(b) To plot index histogram, we should compute the index of the second largest coefficient value in each row in both of *TrainsampleDCT_FG* and *TrainsampleDCT_BG.* The following two figures (Figure 1. and Figure 2.) indicate the index histogram for $P_{X|Y}(x|cheetach)$ and $P_{X|Y}(x|grass)$.

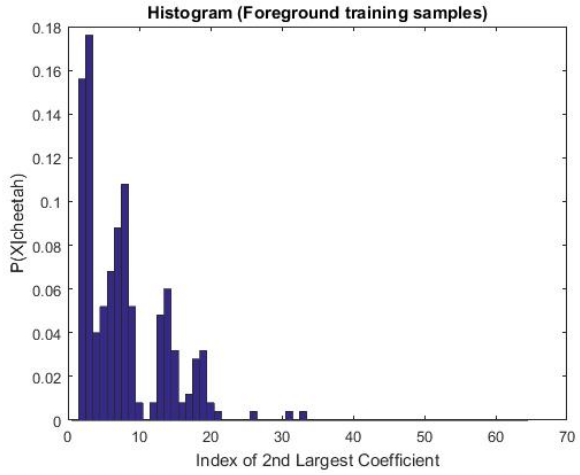

Figure 1.

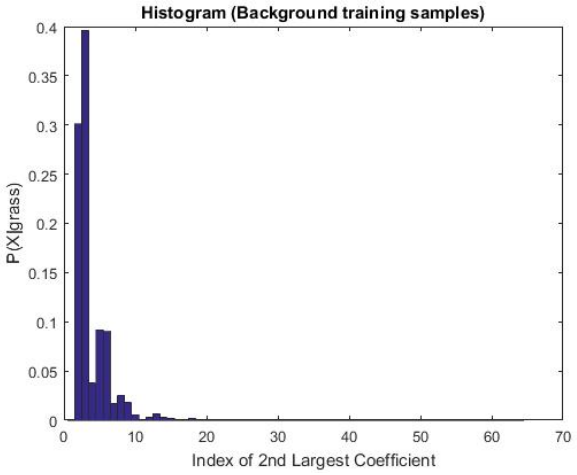

Figure 2.

(c) Firstly, to solve the corner and edge pixel issue. I adopt zero padding. Then, compute DCT for each 8x8 size block and set the top-left pixel as my target pixel. In each block, I compute the conditional probabilities for the index that has the second largest energy from the results of problem b. Then, after multiplying prior probabilities and apply Bayesian Decision Rule, we can predict the class of the top-left pixel in the block. The result is shown in the Figure 3.

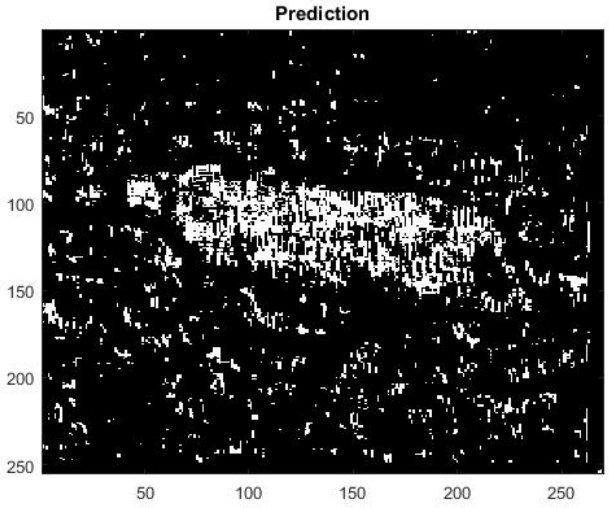

Figure 3.

(d) Compare my prediction and the ground truth, then computer the probability of error of my algorithm.

$$P(error) = P_{X|Y}(grass|cheetah) \times P_Y(cheetah) + P_{X|Y}(cheetah|grass) \times P_Y(grass)$$
$$= 0.6760 \times 0.1919 + 0.0531 \times 0.8081$$
$$= 0.1726$$

MATLAB CODE

```
close all
clear all
clc

load('TrainingSamplesDCT_8.mat')

%% Problem a
[row_f,col_f] = size(TrainsampleDCT_FG);
FG_index=zeros(1,row_f);
[row_b,col_b] = size(TrainsampleDCT_BG);
BG_index=zeros(1,row_b);

Prior_FG=row_f/(row_f+row_b);         % Calculate Prior
Prior_BG=row_b/(row_f+row_b);
disp('Prior Probability (cheetah):')
disp(Prior_FG)
disp('Prior Probability (grass):')
disp(Prior_BG)
%% Problem b
```

```
for i=1:row_f
    [maximum,index]=sort(abs(TrainsampleDCT_FG(i,:)),'descend');
    FG_index(i)=index(2);            %choose 2nd largest coefficient
end

for i=1:row_b
    [maximum,index]=sort(abs(TrainsampleDCT_BG(i,:)),'descend');
    BG_index(i)=index(2);
end

binranges=0.5:1:63.5;
[bincounts_FG] = histc(FG_index,binranges);
[bincounts_BG] = histc(BG_index,binranges);
bincounts_FG_p=bincounts_FG/sum(bincounts_FG);
bincounts_BG_p=bincounts_BG/sum(bincounts_BG);

figure;
bar(binranges,bincounts_FG_p,'histc')
xlabel('Index of 2nd Largest Coefficient')
ylabel('P(X|cheetah)')
title('Histogram (Foreground training samples)')
figure;
bar(binranges,bincounts_BG_p,'histc')
xlabel('Index of 2nd Largest Coefficient')
ylabel('P(X|grass)')
title('Histogram (Background training samples)')

%% Problem c
% Load Test sample
Img_ori=imread('cheetah.bmp');          %Use zero padding for edge & corner
Img = padarray(Img_ori,[7 7],'post');    %classified pixel:left top
Img=im2double(Img);
[row,col]=size(Img);

% Load Zig-Zag pattern
Zigzag=load('Zig-Zag Pattern.txt');
Zigzag=Zigzag+1;

Map_index=zeros(row-7,col-7);
```

```
for i=1:row-7
    for j=1:col-7
        Field=Img(i:i+7,j:j+7);
        DCT=dct2(Field);
        DCT_64(Zigzag)=DCT;           %assign value following zigzag pattern
        [value,index]=sort(abs(DCT_64),'descend'); %ignore the 1st coefficient
        Map_index(i,j)=index(2);
    end
end

A=zeros(row-7,col-7);
for i=1:row-7                         %Do noy predict padding area
    for j=1:col-7                     %Bayesian Decision Rule
        if bincounts_FG_p(1,Map_index(i,j))*Prior_FG >
bincounts_BG_p(1,Map_index(i,j))*Prior_BG
            A(i,j)=1;
        else
            A(i,j)=0;
        end

    end
end
figure;
imagesc(A);
title('Prediction')
colormap(gray(255));

%% Problem d
% Load Ground Truth
Gt=imread('cheetah_mask.bmp');
Gt=im2double(Gt);

Gt_FG=0;
Gt_BG=0;
for i=1:row-7
    for j=1:col-7
        if Gt(i,j)==1
            Gt_FG=Gt_FG+1;
        end
        if Gt(i,j)==0
```

```
            Gt_BG=Gt_BG+1;
        end
    end
end

Errors_FG=0;
Errors_BG=0;
for i=1:row-7
    for j=1:col-7
        if Gt(i,j)==1 && A(i,j)==0 % FG pixels, misclassifcied as BG
            Errors_FG=Errors_FG+1;
        end
        if Gt(i,j)==0 && A(i,j)==1
            Errors_BG=Errors_BG+1;
        end
    end
end

Errors_FG_p=Errors_FG/(Gt_FG);    %Type II (False Negative)
Errors_BG_p=Errors_BG/(Gt_BG);    %Type I (False Positive)

Errors=Errors_FG_p*Prior_FG + Errors_BG_p*Prior_BG;

disp('Error Rate (%):')
disp(Errors*100)
```