ECE271 homework1

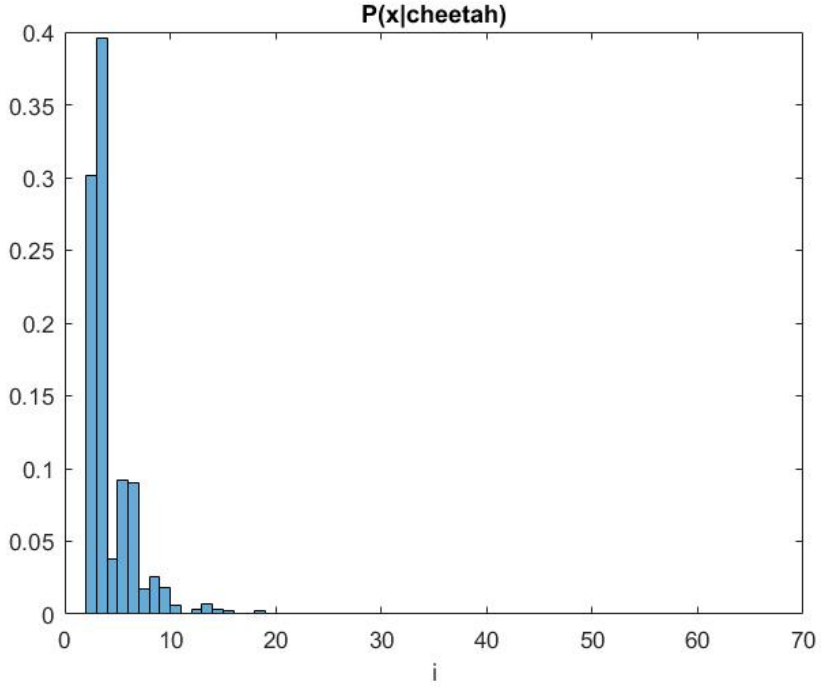
Name:Hangran Zhang
PID: A53256363

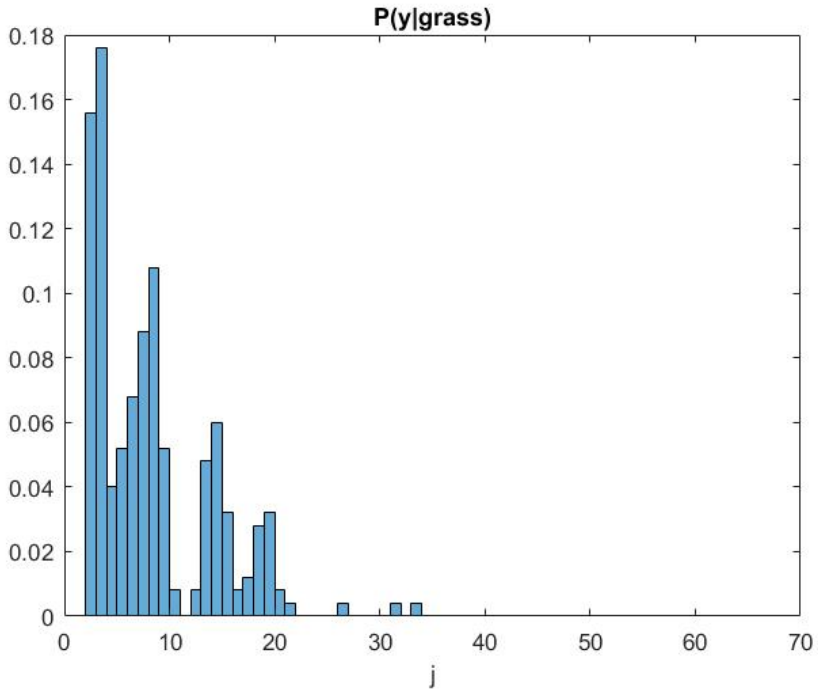
(a)
Using the training data in TrainingSamplesDCT 8.mat, we can obtain the reasonable estimates for the prior probabilities:

P_Cheetah=0.1919
P_Grass=0.8081

(b)
For each vector, we compute the index (position within the vector) of the coefficient that has the second largest energy value (absolute value)

By building an histogram of these indexes we obtain the class-conditionals for the two classes PX|Y (x|cheetah) and PX|Y (x|grass). This is our observation or feature X, X from 1 to 64 and the results are in the following figures.





(c)

Classifier
–break training images into 8x8 blocks
– for each block
• compute DCT,
• order coefficients with zig-zag scan
• pick position of 2nd largest magnitude as the feature value
– use BDR to find class Y for each block
– create a binary mask with 1's for foreground blocks and 0's for background blocks

```
% classifier
prob1=BG_hist.Values;
prob2=FG_hist.Values;
final = zeros(row,column);
for i3=1:row
   for j3=1:column
      l1=loc(i3,j3);
      PRBG=prob1(l1);
      PRFG=prob2(l1);
      if (PRBG*P_Cheetah > PRFG*P_Grass)

         final(i3,j3)=1;
```

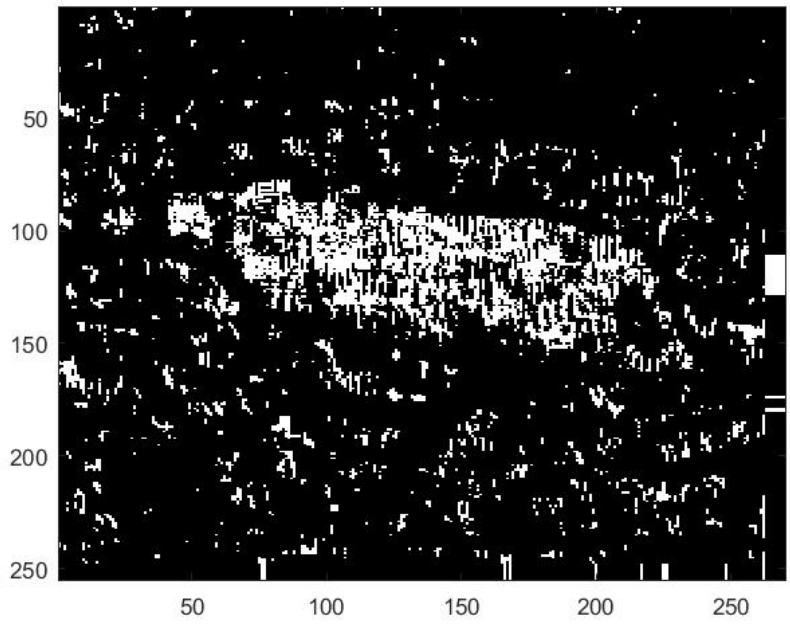
```
        else
            final(i3,j3)=0;
        end
    end
end
```

The zig-zag pattern goes from 0 to 63, not 1 to 64. Need to add 1 to the number in the file to get to the matlab coordinates. Zig = zig+1

Let 1 represents cheetah and 0 represents grass.
According to Bayesian Decision Rule
Threshold=$\frac{L[0,1]P_Y(1)}{L[1,0]P_Y(1)}$

The optimal decision function is
G(x)= 0 if $\frac{P_{X|Y}(x|0)}{P_{X|Y}(x|1)}$>Threshold
1 if $\frac{P_{X|Y}(x|0)}{P_{X|Y}(x|1)}$<Threshold

Threshold= 0.2374
Compute feature X and the state variable Y using the minimum probability of error rule and store in an array A. Using the commands imagesc and colormap(gray(255)) to create the picture:



(d)

When comparing mask with the ground truth in image cheetah.mask.bmp, there are some error makes the boundary not clear and some parts missing.

compute the error of the mask is 0.1693

compute the probability of error:

error(FG)=$\frac{number\ of\ FG\ pixels\ misclassified\ as\ BG}{number\ of\ FG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ FG$

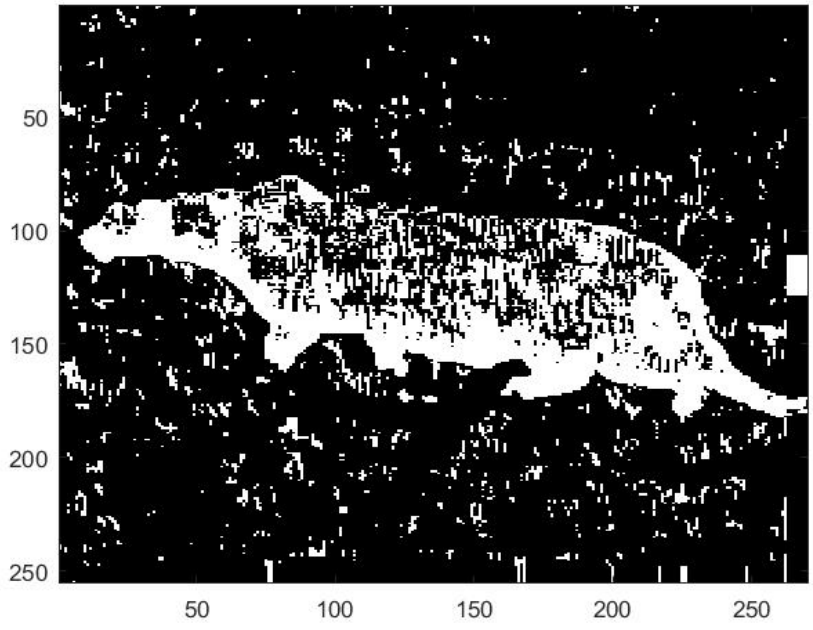
error(BG)= $\frac{number\ of\ BG\ pixels\ misclassified\ as\ FG}{number\ of\ BG\ pixels\ in\ ground\ truth\ of\ test\ set} \times prior\ probability\ of\ BG$

probability of error = $\sum_i Px|y(g(x)! = i|i)Py(i)$

where g(x) is the prediction, (index of the DCT coefficient with 2nd greatest energy), i is the ground truth class label

other method calculate the error = $\sum_{x=0}^{64} Py, x(y \neq g(x), x)$=$\sum_{x=0}^{64} P(Y = 1, X = x)ð0\big(g(x)\big) + P(Y = 0, X = x)ð1(g(x))$=0.15

The error distribution is generated as follows:



The error of the mask is 0.1693, which is close the probability of the error of the algorithm 0.15.

Matlab Code:

```
load('TrainingSamplesDCT_8.mat');

% (a)reasonable estimates for the prior probilities
feature_num=64;
cheetah_num=250;
grass_num=1053;
P_Cheetah= (cheetah_num/(cheetah_num+grass_num));
P_Grass=grass_num/(grass_num+cheetah_num);



% (b)for each vector, we compute the index (position within
the vector)
% of the coefficient that has the second largest energy
value (absolute value)
FG = abs(TrainsampleDCT_FG);
BG = abs(TrainsampleDCT_BG);


 for i=1:grass_num
     p= sort(BG,2,'descend');
     x_index(i)=find (BG (i,:)== p(i,2));
 end
 for j=1:cheetah_num
     q= sort(FG,2,'descend');
     y_index(j)=find (FG (j,:)== q(j,2));
 end


% histogram and conditional probability

figure(1)
FG_hist=histogram(x_index,1:64,'Normalization','probability
');
cheetah_con = FG_hist.Values;
title('P(x|cheetah)');
xlabel('i');


figure(2);
```

```
BG_hist=histogram(y_index,1:64,'Normalization','probability
');
grass_con = BG_hist.Values;
title('P(y|grass)');
xlabel('j')


% (c)
% read image/files
zig=load('Zig-Zag Pattern.txt');
zig=zig+1;
zig=zig(:);

cheetach=imread('cheetah.bmp');
cheetach=im2double(cheetach);
[row, column] = size(cheetach);

% computing the dct and features
for m=1:(row-7)
  for n=1:(column-7)
      temp_row = zeros(1,64);
        index_DCT= abs(dct2(cheetach(m:m+7, n:n+7)));
        index_DCT= index_DCT(:);

        for k=1:64
            temp_row(zig(k)) = index_DCT(k);
        end

       s = sort(temp_row, 'descend');
       s2= s(2);
       lo= find(temp_row==s2);
       loc(m:m+7,n:n+7)=lo;
 end
end


% classifier
prob1=BG_hist.Values;
prob2=FG_hist.Values;
final = zeros(row,column);
for i3=1:row
    for j3=1:column
        l1=loc(i3,j3);
        PRBG=prob1(l1);
```

```
        PRFG=prob2(l1);
        if (PRBG*P_Cheetah > PRFG*P_Grass)

            final(i3,j3)=1;
        else
            final(i3,j3)=0;
        end
    end
end


figure(3);
imagesc(final);
colormap(gray(255));


% (d)
% compute the error of the mask

mask=imread('cheetah_mask.bmp');
mask=im2double (mask);
error= mask ~= final;
figure(4);
imagesc(error);
colormap(gray(255));

sm= sum(error);
error= sum(sm)/(row*column)



% compute the probability of error
error_p = 0;
for i = 1: feature_num
    if final(i) == 0
        error_p = error_p + P_Cheetah * prob1(i);
    end
    if final(i) == 1
        error_p = error_p + P_Grass* prob2(i);
    end
end
error_p
```

Collaborate with Ziyan Zhu