Arda Cankat Bati

ECE 271A Homework #1, Computer Assignment

Std Id = A53284500

ANSWERS

a)

The training set has 1053 instances of length 64 vectors associated with background (No cheetah), and 250 instances with foreground (Cheetah). Therefore it is reasonable to assume:

P(X = Cheetah) = (#Of Cheetah Data Points / # Total Data Points)

= 250/1303 = 0.1918649 = 0.1919 (roughly)

And similarly for No Cheetah points:

P(X = No Cheetah) = (#Of No Cheetah Data Points / # Total Data Points)

= 1053/1303 = 0.8081351 = 0.8081 (roughly)

b)

In the train data sets we are given Zig-Zag patterned, 64 units long DCT transformations of 8x8 blocks from an original image. From the train sets, for each data point (each row of data set) we extract the index of the second largest absolute value. We name this X for each data point and do this separately for Foreground and Background data sets. In the end we get two distributions which are named SecLarFG and SecLarBG in the code. To turn these distributions actual conditional probability distributions, we use histograms.

For the computation of conditional probability histograms, MATLAB's histogram function is used with the 'Normalization' and 'PDF' inputs. The histogram is normalized considering that $P_{X|Y}$ (X | cheetah) across all x sums up to 1. For each value of x, the conditional probability will be:

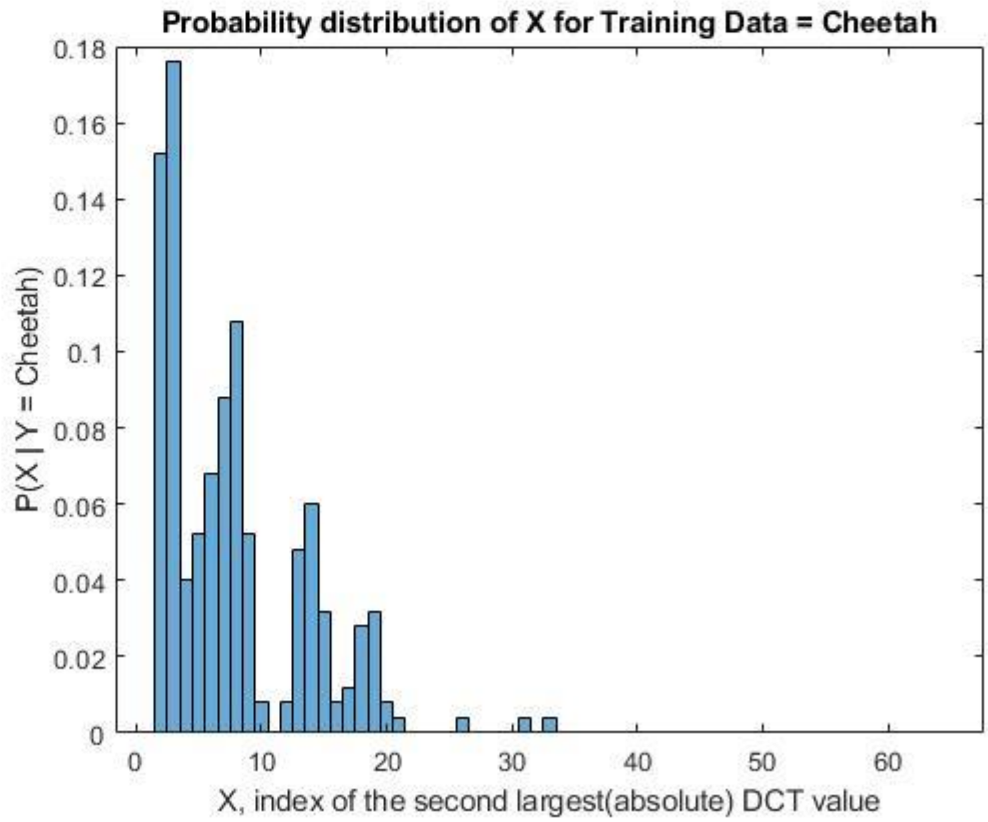
$P_{X|Y}$(X = x | cheetah) = (# data points that yielded X = x) / (total number of points)

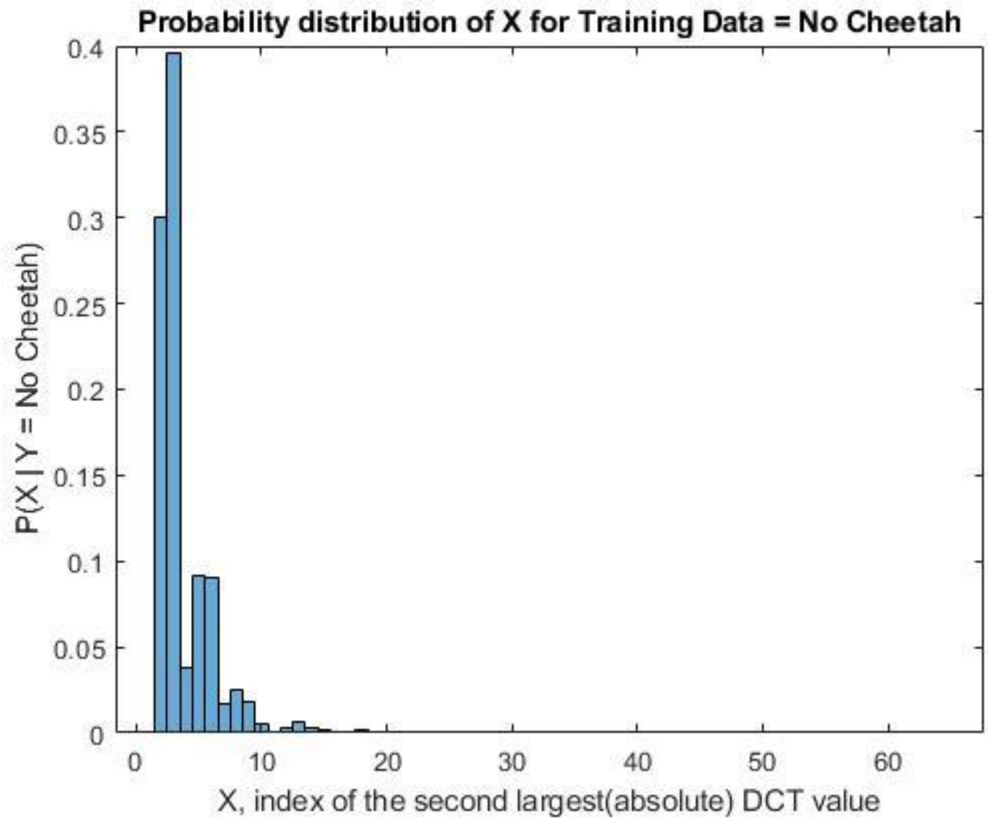
For example for X = 2,

$P_{X|Y}$(X = 2 | cheetah) = (# data points that yielded X = 2) / (total number of points).

MATLAB uses a 'PDF' function to determine the bins and edges. As our X's are made up of index values, the resulting edges are quite trivial. For each X = x, the limits are determined as [x-0.5 x+0.5] with this function, which is reasonable for our data.

Analysing the histograms, in general $P_{X|Y}$ (X | cheetah) histogram seems to have broader frequency range compared to $P_{X|Y}$ (X | No cheetah). This is reasonable, considering the cheetah’s camouflage pattern, that has various frequencies in it, compared to the plain grassland of the background.





c)

This part of the homework is mainly handled in the code. The important thing to note here is the decision function. Here we use a “0-1” loss model. In this case using the MAP rule would be optimal. For the “0-1” loss case, The Bayesian Risk is the probability of error of the MAP Rule. We can use the second form of the MAP Rule:

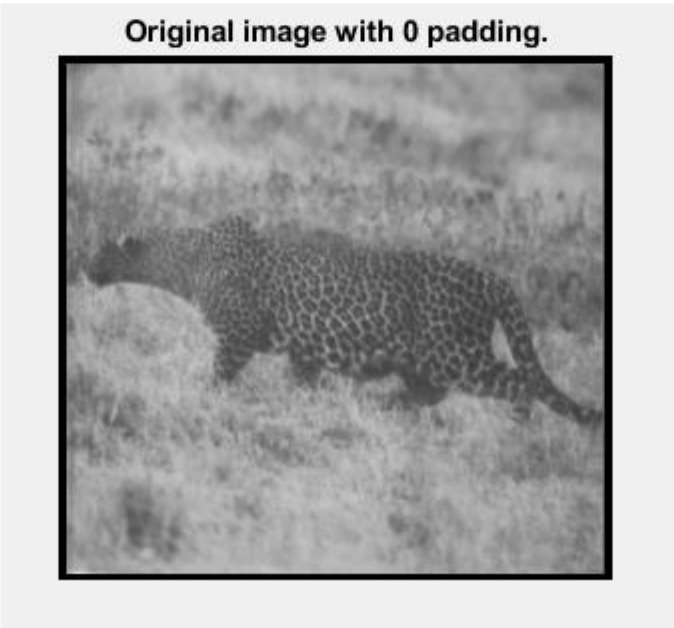
$$i^{*}(x) = \text{argmax}_{(i)} \, [\, P_{X|Y} (X \mid i) * P_{Y} (i) \,]$$

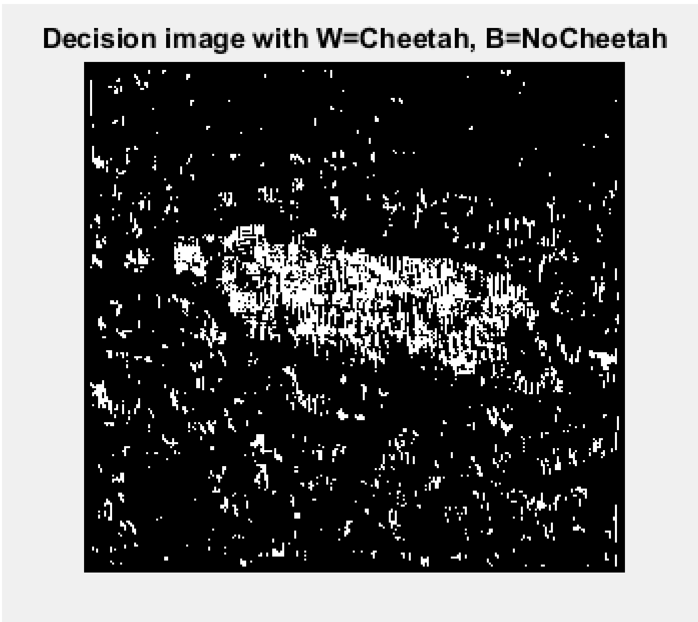
For class i, we have two cases, i_1 = cheetah, i_2 = no cheetah. For each pixel in the image, we have to extract a corresponding X value. We should then compare the output of $P_{X|Y} (X \mid i) * P_{Y} (i)$ for i_1 and i_2, and make a class decision for the one with the higher probability.

Code Explanation:

For extracting the value X, we use a sliding windows of 8x8 along the image. The feature X will be calculated for the pixel with position (4,4) in the 8x8 window matrices. To enable this, we pad the image by zeros: 4 zeros added to the top and left, 3 zeros to bottom and right. After the padding we slide the 8x8 windows over the padded image by a for loop.

At each iteration of the loop, DCT of the window is calculated and absolute of it is taken. The DCT matrix is turned into a 64 unit length vector. Then, the zig zag pattern given in the question is applied to this vector. Using this final vector (named zigzag in the code), the second largest value’s index is found. This value is the feature X for the corresponding pixel. Putting this value into the MAP rule’s argmax function, a corresponding decision is made. Decision = 0 corresponds to no cheetah, decision = 1 corresponds to cheetah. An 250x277 matrix of decision values are formed, corresponding to the size of the original test image. Below, the padded original image, and final prediction image is displayed:





d)

We extract the ground truth from ‘cheetah mask.bmp’ by converting it to a matrix of 0 and 1’s. Therefore we get the correct decisions for each point. Now, our decision points and the true data points are of the same form. We can create an error mask by subtracting the original values from our decision values:

Error_Mask = Decision_Points - Real_Image;

For each element x of the error mask we have 3 options:

- x = 0, this means that our prediction was true.
- x = 1, this means a false positive, Type I Error. We predicted Cheetah but the truth was No Cheetah.
- x = -1, this means a false negative, Type II Error. We predicted No Cheetah but the truth was Cheetah.

For the Probability of Error we have to consider the priors and alpha & beta values. As the case is 0 – 1 loss we do not have to take more steps. The Probability of Error will consist of two parts:

- P(Y = No Cheetah) * P(False Positive) → Prior(No Cheetah) * Alpha
- P(Y = Cheetah) * P(False Negative) → Prior(Cheetah) * Beta
-

Where the priors are taken directly from the test data as:

P(Y = Cheetah) = (#Cheetah pixels) / (# all pixels)

P(Y = No Cheetah) = (#No Cheetah pixels) / (# all pixels) = 1 - P(Y = Cheetah)

Then the Probability of Error P(E) will be:

P(E) = P(Y = No Cheetah) * P(False Positive) + P(Y = Cheetah) * P(False Negative)

From MATLAB results:

- P(Y = No Cheetah) = 8.081481e-01
- P(False Positive) = 5.273090e-02
- P(Y = Cheetah) = 1.918519e-01
- P(False Negative) = 6.734802e-01

And finally P(E) = 1.718228e-01 < P(Y = Cheetah) = 1.918519e-01 which would be the error of a naïve classifier that labels everything as No Cheetah. Therefore our classifier is currently better than a naïve classifier, as required in the assignment. Below, the code written by me for this assignment is given. I worked on all the steps alone.

```
%**********************
% Arda Cankat Bati
% ECE 271A - Homework#1
%***********************

clear
clc
load('TrainingSamplesDCT_8.mat');

%******* IMAGE EXTRACTION AND PRIOR ESIMATION *******
%******* EXTRACTING FEATURE X FROM TRAIN DATA *******

FG = abs(TrainsampleDCT_FG);
BG = abs(TrainsampleDCT_BG);
FG_size = size(FG,1);
BG_size = size(BG,1);

ArraySize = size(FG,2);
SecLarFG = zeros(1,FG_size);
SecLarBG = zeros(1,BG_size);
FG(1,:) = 0; BG(1,:) = 0;
CheetahPrior = FG_size/(FG_size + BG_size);
fprintf('"Cheetah prior" estimated from the training data is: %d\n',CheetahPrior);
NoCheetahPrior = BG_size/(FG_size + BG_size);
fprintf('"No Cheetah prior" estimated from the training data is: %d\n',NoCheetahPrior);
for n = 1:size(FG,1)
    [M,I] = max(FG(n,:));
    FG(n,I) = 0;
    [M,SecLarFG(n)] = max(FG(n,:));
end
for n = 1:size(BG,1)
    [M,I] = max(BG(n,:));
    BG(n,I) = 0;
    [M,SecLarBG(n)] = max(BG(n,:));
end

%****************************************************

%******* CREATING HISTOGRAMS FOR CONDITIONAL PROBS OF X *******

XGivenCheetah = histogram(SecLarFG,'Normalization','PDF');
X1 = XGivenCheetah.Values;
XGivenCheetah.BinLimits = [1.5 64.5];
title('Probability distribution of X for Training Data = Cheetah');
xlabel('X, index of the second largest(absolute) DCT value');
ylabel('P(X | Y = Cheetah)'); figure();
XGivenNoCheetah = histogram(SecLarBG,'Normalization','PDF');
X2 = XGivenNoCheetah.Values;
XGivenNoCheetah.BinLimits = [1.5 64.5];
title('Probability distribution of X for Training Data = No Cheetah');
xlabel('X, index of the second largest(absolute) DCT value');
ylabel('P(X | Y = No Cheetah)'); figure();
X1 = padarray(X1,[0 (ArraySize - size(X1,2))], 'post');
X2 = padarray(X2,[0 (ArraySize - size(X2,2))], 'post');


%****************************************************
```

```
%******* GETTING THE TEST IMAGE AND PADDING *******

[cImageOld colormap] = imread('cheetah.bmp');
RGB = ind2rgb(cImageOld,colormap);
cImage = RGB(:,:,1);
paddingType = 0;
cImage = padarray(cImage,[4 4],paddingType,'pre');
cImage = padarray(cImage,[3 3],paddingType,'post');
imshow(cImage); title('Original image with 0 padding.'); figure();

%*****************************************************

%******* CLASSIFYING EACH PIXEL IN THE TEST IMAGE ********
%******* PRINTING THE FINAL BLACK & WHITE IMAGE    *******

cImageOldX = size(cImageOld,1); cImageOldY = size(cImageOld,2);
cImageX = size(cImage,1); cImageY = size(cImage,2);
A = [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];
A = A + 1;
decisionImage = zeros(size(cImageOld,1),size(cImageOld,2));

for i = 1:cImageOldX
    for j = 1:cImageOldY
        temp = (abs(dct2(cImage(i:i+7, j:j+7))))';
        vectorDct= temp(:);
        zigzag(A) = vectorDct;
        [M,I] = max(zigzag);
        zigzag(I) = 0;
        [M,I] = max(zigzag);
        [M,decision] = max([NoCheetahPrior*X2(I) CheetahPrior*X1(I)]);
        decision = decision - 1;
        decisionImage(i,j) = decision;
    end
end

I = mat2gray(decisionImage,[0 1]);
imshow(I); title('Decision image with W=Cheetah, B=NoCheetah');

%*****************************************************

%******* CALCULATING THE TOTAL ERROR *******

[cImageReal colormap] = imread('cheetah_mask.bmp');
cImageReal = double(cImageReal)/255;
errorMask = decisionImage - cImageReal;
Image_Size = size(cImageReal,2)*size(cImageReal,1);
FG_Sum = sum(sum(cImageReal));
BG_Sum = Image_Size - FG_Sum;
truePriorCheetah = FG_Sum/Image_Size;
truePriorNoCheetah = 1 - truePriorCheetah;

%FG misclassified as BG / total true FG
beta = sum(sum(errorMask == -1)) / FG_Sum; %False Negative --> beta
%BG misclassified as FG / total true BG
alpha = sum(sum(errorMask == 1)) / BG_Sum;  %False Positive --> alpha
```

```
ProbOfError = truePriorCheetah * beta + truePriorNoCheetah * alpha;
fprintf('True Prior for Cheetah is: %d\n',truePriorCheetah);
fprintf('True Prior for No Cheetah is: %d\n',truePriorNoCheetah);
fprintf('False Positive alpha is: %d\n',alpha);
fprintf('False Negative beta: %d\n',beta);
fprintf('Total Probability of Error is: %d\n',ProbOfError);

%*****************************************************
```

```
"Cheetah prior" estimated from the training data is: 1.918649e-01
"No Cheetah prior" estimated from the training data is: 8.081351e-01
True Prior for Cheetah is: 1.918519e-01
True Prior for No Cheetah is: 8.081481e-01
False Positive alpha is: 5.273090e-02
False Negative beta: 6.734802e-01
Total Probability of Error is: 1.718228e-01
```