[ECE 271] Homework-1

Yang Yue

A53301503

Problem 1.5

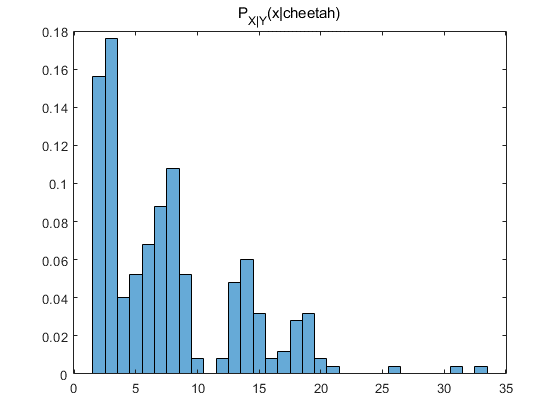
1. 1) By simply looking at the Cheetah.bmp and estimate the ratio of the area that cheetah covers in the whole picture.

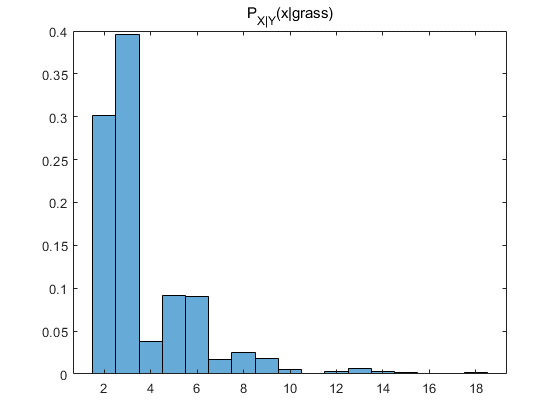
2) By loading the TrainingSampleDCT\_8.mat, from which we can possess the size of training samples’ matrices of front ground and background. And then the prior probability can be estimated as:

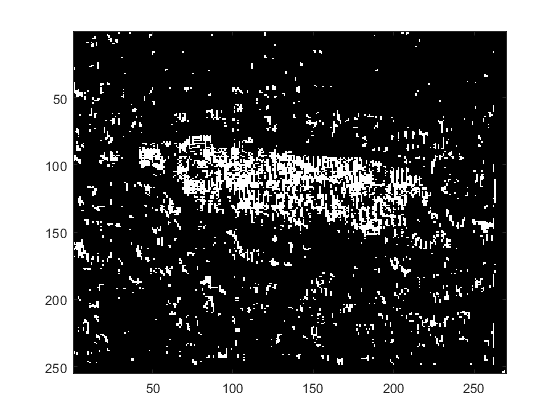
3) By loading the Cheetah\_mask.bmp, then transform it from black and white pixels into a matrix consist of 0s and 1s. The prior probability can be written as:

All methods mentioned to estimate the prior probability of front ground, the corresponding prior probability of background is simply defined as:

1. Index histograms are shown below:





1. The image file generated from array A is shown below:
2. By loading the Cheetah\_mask.bmp, we can get an array, each of whose elements represent whether the pixel belongs to the front ground or the background by its 0-1 status.

An error determined by:

Then the error rate is calculated as follows:

Appendix

Find2ndx.mat

function x = find2ndX(oriMatrix)

%To extract the second largest number's position --x

%in TrainingSample\_BG/FG

%First, set the largest element in BG/FG matrices to 0

oriMatrix(oriMatrix == max(oriMatrix,[],2))=0;

%second, find the largest element in TSDCT\_FG/BG, send their positions to

%xFG/BG.

[~, x] = max(oriMatrix,[],2);

end

hw1\_5.mat

%load training samples and zig-zag order

load('TrainingSamplesDCT\_8.mat');

TSDCT\_FG = TrainsampleDCT\_FG;

TSDCT\_BG = TrainsampleDCT\_BG;

zigzag = load('Zig-Zag Pattern.txt');

zigzag = reshape(zigzag, 1, []) + 1;

%set prior prob P(X), P(Y)

lenFG = size(TrainsampleDCT\_FG, 1);

lenBG = size(TrainsampleDCT\_BG, 1);

xFG = find2ndX(TSDCT\_FG);

xBG = find2ndX(TSDCT\_BG);

%print out histographs of BG/FG

figure(1);

h\_fg = histogram(xFG,'Normalization','probability');

title('P\_{X|Y}(x|cheetah)');

figure(2);

h\_bg = histogram(xBG,'Normalization','probability');

title('P\_{X|Y}(x|grass)');

%Pxy\_BG <= P(X=xi|Y=BG)

%Pxy\_FG <= P(X=xi|Y=FG)

Pxy\_BG = zeros([64,1]);

Pxy\_FG = zeros([64,1]);

tlb\_BG = tabulate(xBG);

tlb\_FG = tabulate(xFG);

Pxy\_BG(1:size(tlb\_BG,1)) = tlb\_BG(:,3);

Pxy\_FG(1:size(tlb\_FG,1)) = tlb\_FG(:,3);

Pxy\_BG(Pxy\_BG == 0) = 0.0001;

py\_BG = lenBG/(lenBG+lenFG);

py\_FG = 1-py\_BG;

PdP = (Pxy\_FG.\*(py\_FG))./(Pxy\_BG.\*(py\_BG)); %decision boundary

%read the test img cheeta

cheetah\_img = imread('cheetah.bmp');

cheetah\_dw = im2double(cheetah\_img);

%set a blank padding

cheetah\_pad = [cheetah\_dw, zeros([size(cheetah\_dw,1),7]); zeros([7,size(cheetah\_dw,2)+7])];

%initialize test set

test\_set = zeros([255\*270,64]);

cnt = 1;

%test set implementation

%slice the image into 8\*8 blocks and run the dct2()

for col = (1:size(cheetah\_pad,2))

if (col+7) > size(cheetah\_pad,2)

break;

end

for row = (1:size(cheetah\_pad,1))

if (row+7) > size(cheetah\_pad,1)

break;

end

test\_set(cnt,:) = reshape(abs(dct2(cheetah\_pad(row:row+7,col:col+7))),1,[]);

cnt = cnt + 1;

end

end

%find the 2nd max value's position and map it with zigzag

testMx2nd = find2ndX(test\_set);

test2ndZig = zeros([size(testMx2nd,1),1]);

for i = 1:size(testMx2nd,1)

test2ndZig(i) = zigzag(testMx2nd(i));

end

%Cheetah or not?

A = reshape(isCheetah(test2ndZig,PdP),size(cheetah\_dw,1),size(cheetah\_dw,2));

figure(3);

imagesc(A);

colormap(gray(255));

%error computation

cheetah\_mask = imread("cheetah\_mask.bmp");

cheetah\_mask\_dw = im2double(cheetah\_mask);

errorRate = sum(abs(A - cheetah\_mask\_dw),'all')/(size(cheetah\_mask,1)\*size(cheetah\_mask,2))