Statistical Learning Assignment-1

Jaw Yuan, Chang (A59002885)

October, 2020

Problem set (a) : Prior probability

Given the TrainingSamplesDCT_8.mat, there are 1053 package samples of the background and 250 package samples of the foreground. Those samples can be mapped to grass and cheetah respectively by assuming that all foreground samples in the training set are generated from cheetah blocks and vice versa. Since there is only two class in this case, it is intuitive to figure out the prior probability using common sense:

$$P_y(Cheetah) = \frac{numbers\, of\, foreground\, samples}{numbers\, of\, foreground\, samples + numbers\, of\, background\, samples}$$

and

$$P_y(Grass) = 1 - P_y(Cheetah)$$

With those equations, the probability can be calculated:

$P_y(grass) = 80.81\%$ and $P_y(Cheetah) = 19.19\%$

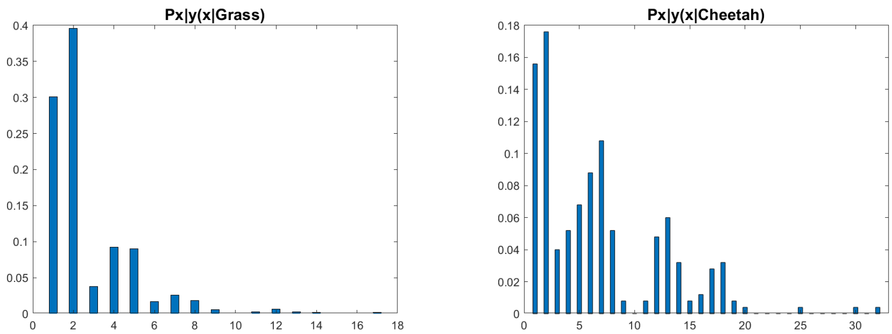
Problem set (b): Conditional Probability

Following the guide and the slide from the assignment, every image is separated as a collection of 8*8 image blocks. For each image block, the discrete cosine transform is applied for the frequency coefficients and then compute the 2^{nd} largest frequency coefficients index. After taking the summation over the 2^{nd} large index from the training set, the class conditional distribution can be calculated with histogram:

$$P_{X|Y}(x|Cheetah) = \frac{summation\, of\, index\, position\, in\, foreground}{numbers\, of\, foreground\, package\, samples}$$

$$P_{X|Y}(x|grass) = \frac{summation\, of\, index\, position\, in\, background}{numbers\, of\, background\, package\, samples}$$

Result



(a) $P_{X|Y}(x|Grass)$ (b) $P_{X|Y}(x|Cheetah)$

Figure 1: Class conditional distribution

Problem set (c): Cheetah mask

Predicting whether the image block is cheetah or not is the binary classification problem. The formula can be derived as:

$$P_{Y|X}(Cheetah|x) > P_{Y|X}(Grass|x)$$

However, it is too complicated to calculate the probability from the above equation. By the Bayes rule, it substitutes the relation between cause and result and makes life easier.

$$P_{Y|X}(Cheetah|x) = \frac{P_{X|Y}(x|Cheetah) * P_y(Cheetah)}{P_X(x)}$$

and also

$$P_{Y|X}(Grass|x) = \frac{P_{X|Y}(x|Cheetah) * P_y(Grass)}{P_X(x)}$$

Finally, identify it as cheetah if:

$$P_{X|Y}(x|Cheetah) * P_y(Cheetah) > P_{X|Y}(x|Cheetah) * P_y(Grass)$$

The mask is shown in Figure 2 and the overlay is shown in Figure 3. Principle of this algorithm is base on textures, that is, the high frequency components on the cheetah's body and the low frequency components from the grass. With the observation from Figure 3, the algorithm is hard to make a right decision while the texture is quite blur or monochrome. For instance, take a look at the tail and legs part, the model completely be overwhelmed. Also, the model is not robust enough to neglect the grass part with cheetah-liked components. For example, on the bottom left of the diagram, the model just assumes it as a cheetah while it is grass.

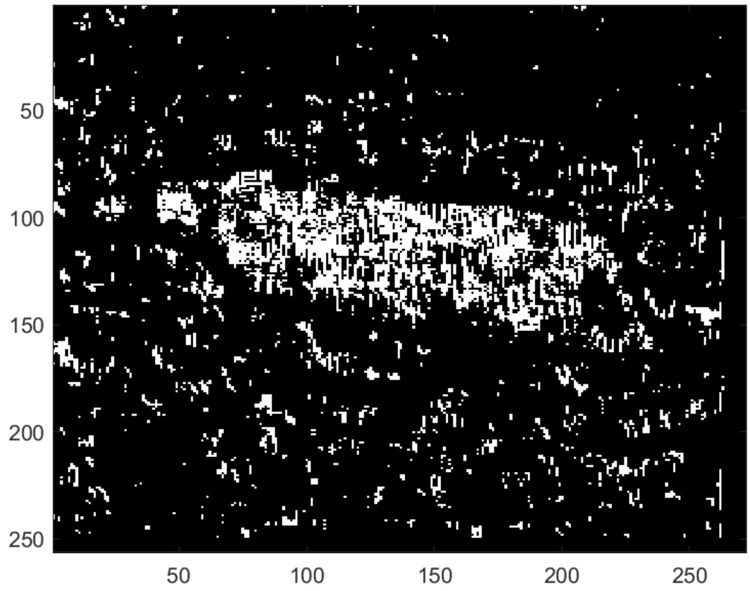


Figure 2: Cheetah mask

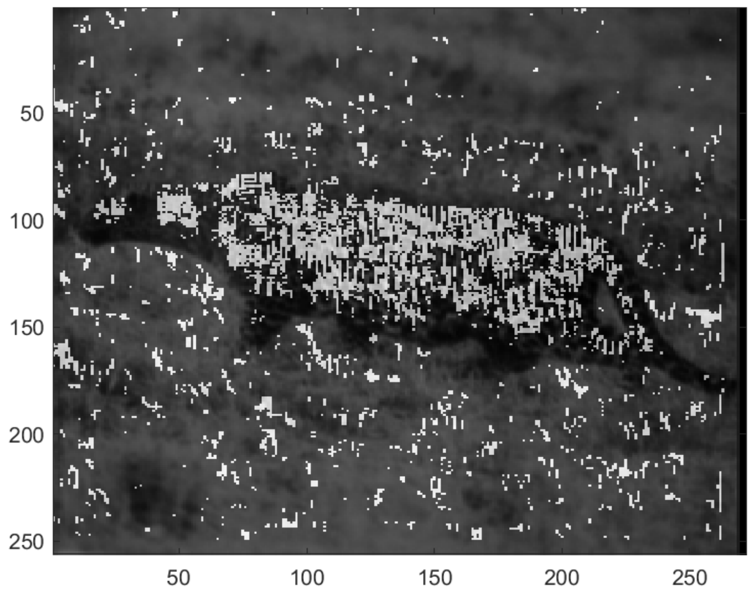


Figure 3: Cheetah mask overlay

Problem set (d): Performance

My total error rate is 17.08%

```
% This code is for ECE271A HW1. Create date: Oct/6/2020.
close all;
%% Setup
%Load the files;
Train_sample = load('TrainingSamplesDCT_8.mat');
Trainbg = Train_sample.TrainsampleDCT_BG;
Trainfg = Train_sample.TrainsampleDCT_FG;
%-----------------------------------------------------------------%
%% Problem (a)
% Calculate the probability of Py(Cheetah) and Py(grass)
Py_cheetah = size(Trainfg,1)/( size(Trainfg,1)+size(Trainbg,1) );
Py_grass = 1 - Py_cheetah;
%-----------------------------------------------------------------%
%% Problem (b)
% Setting the largest coefficient as a dummy variable.
Trainbg(:,1) = -1;
Trainfg(:,1) = -1;
% Finds out second largest coefficient's index
[¬,bg_index] = max(Trainbg,[],2);
[¬,fg_index] = max(Trainfg,[],2);
bincount_bg_index = histcounts(bg_index);
Pg = bincount_bg_index/size(Trainbg,1);
f1 = figure('name','Problem (b)_2');
bar(Pg,0.4);
title('\fontsize{14}Px|y(x|Grass)');
bincount_fg_index = histcounts(fg_index);
Pc = bincount_fg_index/size(Trainfg,1);
f2 = figure('name','Problem (b)_1');
bar(Pc,0.4);
title('\fontsize{14}Px|y(x|Cheetah)');
Pc = padarray(Pc,[0,32],0,'post');
Pg = padarray(Pg,[0,47],0,'post');
%-----------------------------------------------------------------%
%% Problem (c)
imaC = im2single(imread('cheetah.bmp'));
imaC_pad = padarray(imaC,[1,2],'replicate','post');
Path = load('Zig-Zag Pattern.txt');
Path = Path+1;
current_blocks = zeros(8,8);
CheetahMask = zeros(size(imaC_pad,1),size(imaC_pad,2));
for i = 1 : size(imaC_pad,1)-7
    for j = 1 : size(imaC_pad,2)-7
        current_dc2_blocks = dct2(imaC_pad(i:i+7,j:j+7));
        current_blocks(Path) = current_dc2_blocks;
        [¬,index] = max(abs(current_blocks(2:64)));
        if Pc(index) * Py_cheetah > Pg(index) * Py_grass
            CheetahMask(i,j) = 1;
        else
            CheetahMask(i,j) = 0;
        end
    end
end
f3 = figure(3);
imagesc(CheetahMask);
colormap(gray(255));
```

```
imaC_test = imaC_pad + CheetahMask;
imagesc(imaC_test);
%saveas(f1,'1.png')
%saveas(f2,'2.png')
saveas(f3,'3.png')
%-------------------------------------------------------------------%
%% Problem(d)
%-------------------------------------------------------------------%
imay = im2single(imread("cheetah_mask.bmp"));
imay_pad = padarray(imay,[1,2],'replicate','post');
mask_difference = sum(abs(CheetahMask - imay_pad),'all');
error_Mask = mask_difference/(256*272);
% coded by Jaw-Yuan, Chang
```