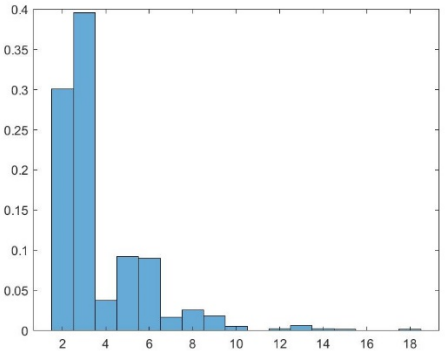
ECE271A HW1 Report

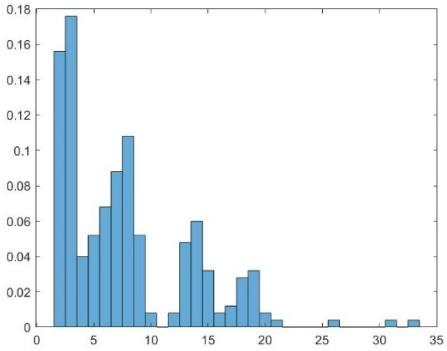
Problem5

a. Based on the number of samples from grass and cheetah in training sets, the prior probability would be determined as:

$$\mathbf{P_Y(cheetah) = 250/(1053+250) = 0.1919}$$
$$\mathbf{P_Y(grass) = 1053/(1053+250) = 0.8081}$$

b. The obtained histogram is:





(a): histogram of background data (b): histogram of foreground data

For background data, the specific value for $\mathbf{P_Y(x \mid grass)}$ is: (from x = 1 to x = 64)

0.301044634377968	0.396011396011396	0.0379867046533713	0.0921177587844255
0.0902184235517569	0.0170940170940171	0.0256410256410256	0.0180436847103514
0.00569800569800570	0	0.00284900284900285	0.00664767331433998
0.00284900284900285	0.00189933523266857	0	0
0.00189933523266857	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

For foreground data, the specific value for $\mathbf{P_Y(x \mid cheetah)}$ is: (from x = 1 to x = 64)

0.156000000000000	0.176000000000000	0.0400000000000000	0.0520000000000000
0.0680000000000000	0.0880000000000000	0.108000000000000	0.0520000000000000
0.00800000000000000	0	0.00800000000000000	0.0480000000000000
0.0600000000000000	0.0320000000000000	0.00800000000000000	0.0120000000000000
0.0280000000000000	0.0320000000000000	0.00800000000000000	0.00400000000000000
0	0	0	0
0.00400000000000000	0	0	0

0	0.00400000000000000	0	0.00400000000000000
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

c. The image predicted is shown below:

During the processing, padding layers are added to the picture. Specifically, 4 layers are added into the top and left edge and 3 layers are added into right and bottom edge. Then the 8*8 blocks could be utilized as sliding windows to process the pixel in original picture and make an output of the same size as input image.

d. The error is obtained by:

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

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

$$\text{error} = \text{error(FG)} + \text{error(BG)}$$

The specific result is:

$$\text{error(FG)} = 0.1516$$

$$\text{error(BG)} = 0.0119$$

$$\text{error} = 0.1635$$

Code:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% problem a
prior_cheetah = 250/(1053+250);
prior_grass = 1053/(1053+250);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% problem b
train_set = load('TrainingSamplesDCT_8.mat');
FGmat = train_set.TrainsampleDCT_FG;
BGmat = train_set.TrainsampleDCT_BG;

%%% foreground
cache1 = [];
prob_fore = zeros(1,64);
for row=1:size(FGmat,1)
    ind2 = second_large_idx(FGmat(row,:));
    cache1 = [cache1, ind2];
end
h1 = histogram(cache1);
h1.Normalization = 'probability';
for i=1:size(h1.Values,2)
    prob_fore(i) = h1.Values(i);
end
saveas(h1, 'foreground_hist', 'jpg');

%%% background
cache2 = [];
prob_back = zeros(1,64);
for row=1:size(BGmat,1)
    ind2 = second_large_idx(BGmat(row,:));
    cache2 = [cache2, ind2];
end
h2 = histogram(cache2);
h2.Normalization = 'probability';
for i=1:size(h2.Values,2)
    prob_back(i) = h2.Values(i);
end
saveas(h2, 'background_hist', 'jpg');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% problem c
img = imread('cheetah.bmp');
img = im2double(img);
%%% Add paddle
img = [zeros(size(img,1),4) img];
img = [zeros(4, size(img,2)); img];
img = [img zeros(size(img,1),3)];
img = [img; zeros(3, size(img,2))];

%%% DCT
[m,n] = size(img);
Blocks = zeros(m-7,n-7);
for i=1:m-7
    for j=1:n-7
        DCT = abs(dct2(img(i:i+7,j:j+7)));
        zigzag_order = zigzag(DCT);
        index = second_large_idx(zigzag_order);
        Blocks(i,j) = index;
    end
end

%%% Try to find something
yesornot = zeros(1,64);
for idx=1:64
    yes = prob_fore(1, idx) * prior_cheetah;
    no = prob_back(1, idx) * prior_grass;
    if yes >= no
```

```
        yesornot(1,idx) = 1;
    end
end
%%%%% Here we found when idx >= 12, the probability to be
%%%%% cheetah will be larger than the probability to be grass

%%% Predict
results = zeros(m-7, n-7);
p = size(results,1);
q = size(results,2);
for i=1:p
    for j=1:q
        yes = prob_fore(1, Blocks(i,j)) * prior_cheetah;
        no = prob_back(1, Blocks(i,j)) * prior_grass;
        if yes >= no && Blocks(i,j) >= 12
            results(i,j) = 1;
        end
    end
end

%%% save prediction
imwrite(results, 'prediction.jpg');
prediction = mat2gray(results);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% problem d
ground_truth = imread('cheetah_mask.bmp')/255;
x = size(ground_truth, 1);
y = size(ground_truth, 2);
count1 = 0;
count2 = 0;
count_cheetah_truth = 0;
count_grass_truth = 0;
for i=1:x
    for j=1:y
        if prediction(i,j) > ground_truth(i,j)
            count2 = count2 + 1;
            count_grass_truth = count_grass_truth + 1;
        elseif prediction(i,j) < ground_truth(i,j)
            count1 = count1 + 1;
            count_cheetah_truth = count_cheetah_truth + 1;
        elseif ground_truth(i,j) >0
            count_cheetah_truth = count_cheetah_truth + 1;
        else
            count_grass_truth = count_grass_truth + 1;
        end
    end
end
error1 = (count1/count_cheetah_truth) * prior_cheetah;
error2 = (count2/count_grass_truth) * prior_grass;
error = error1 + error2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function output = zigzag(in)
% initializing the variables
%-----------------------------------
h = 1;
v = 1;
vmin = 1;
hmin = 1;
vmax = size(in, 1);
hmax = size(in, 2);
i = 1;
output = zeros(1, vmax * hmax);
%-----------------------------------
while ((v <= vmax) && (h <= hmax))

    if (mod(h + v, 2) == 0)                 % going up
        if (v == vmin)
            output(i) = in(v, h);       % if we got to the first line
```

```
        if (h == hmax)
            v = v + 1;
        else
            h = h + 1;
        end
        i = i + 1;
    elseif ((h == hmax) && (v < vmax))   % if we got to the last column
        output(i) = in(v, h);
        v = v + 1;
        i = i + 1;
    elseif ((v > vmin) && (h < hmax))    % all other cases
        output(i) = in(v, h);
        v = v - 1;
        h = h + 1;
        i = i + 1;
  end

  else                              % going down
    if ((v == vmax) && (h <= hmax))       % if we got to the last line
        output(i) = in(v, h);
        h = h + 1;
        i = i + 1;

    elseif (h == hmin)                  % if we got to the first column
        output(i) = in(v, h);
        if (v == vmax)
      h = h + 1;
     else
         v = v + 1;
        end
        i = i + 1;
    elseif ((v < vmax) && (h > hmin))     % all other cases
        output(i) = in(v, h);
        v = v + 1;
        h = h - 1;
        i = i + 1;
    end
  end
  if ((v == vmax) && (h == hmax))         % bottom right element
     output(i) = in(v, h);
     break
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ind2 = second_large_idx(arr)
  [max1, ind1] = max(arr);
  arr(ind1)      = -Inf;
  [max2, ind2] = max(arr);
end
```