

Appendix I

Source Code of HW3 & 4

dozigzag.m

```
function v = dozigzag(u,zig_order)
    v = zeros(size(u));
    for i = 1:64
        v(:,zig_order(i)) = u(:,i);
    end
end
```

ndGauss.m

```
function g = ndGauss(x,mu,cov)
    if istall(x)==0
        x = x.';
    end
    if size(x)~=size(mu)
        disp('Size unequal');
    end
    g1 = (2*pi)^(-size(x,1)/2)*(det(cov)^(-1/2))*exp(-((x-mu).')/(cov)*(x-
mu)*1/2);
    g=g1;
end
```

ysum.m

```
function a = ysum(matrix, dim)
    if dim == 2
        a = matrix*ones([size(matrix,2),1]);
    elseif dim == 1
        a = ones([1,size(matrix,1)])*matrix;
    end
end
```

cheetahCalc.m

```
function [cheetah_pred] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
%Compute to distinguish the cheetah
% Author:Yang Yue
% PID:A53301503
% email:y5yue@ucsd.edu
% -----
nBG = size(D_BG,1);
nFG = size(D_FG,1);
```

```

%mu of different datasets -- mu_ML
muD_BG = ysum(D_BG,1)/nBG;
muD_FG = ysum(D_FG,1)/nFG;

%covariances of different datasets -- cov
meanMatD_BG = zeros(64,64);
meanMatD_FG = zeros(64,64);

for i = 1:nBG
    meanMatD_BG(i,:) = D_BG(i,:)-muD_BG;
end
for i = 1:nFG
    meanMatD_FG(i,:) = D_FG(i,:)-muD_FG;
end

covD_BG = meanMatD_BG.'*meanMatD_BG/nBG;
covD_FG = meanMatD_FG.'*meanMatD_FG/nFG;

%  $P_{\mu|T(\mu|D)=G(\mu, \mu_{BG\_n}, cov_{BG\_n})}$ 
cov0 = diag(alpha*w0);
muBG0 = mu0_BG;
muBG_n = nBG*cov0/(covD_BG+nBG*cov0)*muD_BG' +
covD_BG/(covD_BG+nBG*cov0)*muBG0';
covBG_n = (covD_BG*cov0)/(covD_BG+nBG*cov0);

%  $P_{\mu|T(\mu|D)=G(\mu, \mu_{FG\_n}, cov_{FG\_n})}$ 
muFG0 = mu0_FG; % 1*64
muFG_n = nFG*cov0/(covD_FG+nFG*cov0)*muD_FG' +
covD_FG/(covD_FG+nFG*cov0)*muFG0';
covFG_n = (covD_FG*cov0)/(covD_FG+nFG*cov0);

covBG = covD_BG+covBG_n;
covFG = covD_FG+covFG_n;

if str == 0 %predictive
    BG_mu = muBG_n;
    BG_cov = covBG;

    FG_mu = muFG_n;
    FG_cov = covFG;
end
if str == 1 %MAP
    BG_mu = muBG_n;
    BG_cov = covD_BG;

    FG_mu = muFG_n;
    FG_cov = covD_FG;
end
if str == 2 %MLE
    BG_mu = muD_BG';
    BG_cov = covD_BG;

    FG_mu = muD_FG';
    FG_cov = covD_FG;
end

cheetah_pred = zeros(size(cheetah_blocks,1),1);
for xi=1:size(cheetah_blocks,1)

```

```

pFG = ndGauss(cheetah_blocks(xi,:),FG_mu,FG_cov);
pBG = ndGauss(cheetah_blocks(xi,:),BG_mu,BG_cov);
if (pFG/pBG)>1
    cheetah_pred(xi)=1;
else
    continue
end
end

end

```

hw_4.m

```

%% Step1:initialization, calculate DCT and do the zigzag transformation
clc;
% Once again we use the decomposition
% into 8 x 8 image blocks, compute the DCT of each block, and zig-zag scan.
zigzag = load('Zig-Zag Pattern.txt');
zigzag = reshape(zigzag, 1, []) + 1;

cheetah_img = imread('cheetah.bmp');
cheetah_dw = im2double(cheetah_img);

cheetah_mask = imread('cheetah_mask.bmp');
cheetah_maskdw = im2double(cheetah_mask);

%set a blank padding
cheetah_pad = [cheetah_dw, zeros([size(cheetah_dw,1),7]);
zeros([7,size(cheetah_dw,2)+7])];
[img_row, img_col] = size(cheetah_dw);
cheetah_blocks = zeros(img_row*img_col,64);
cnt = 1;

for col = 1:img_col
    for row = 1:img_row
        window = cheetah_pad(row:row+7,col:col+7);
        cheetah_blocks(cnt,:) = reshape(dct2(window),[],64);
        cnt = cnt+1;
    end
end
cheetah_blocks = dozigzag(cheetah_blocks,zigzag);

load('hw3Data/TrainingSamplesDCT_subsets_8.mat');
%% Strategy 1
load('Alpha.mat'); %
load('Prior_1.mat');
% % P_mu | T(mu | D)=G(mu,muBG_n,covBG_n)
muBG0 = mu0_BG;
% % P_mu | T(mu | D)=G(mu,muFG_n,covFG_n)
muFG0 = mu0_FG; % 1*64
%% Strategy1-Dataset1
%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s1d1_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D1_BDR =
cheetahCalc(cheetah_blocks,D1_BG,D1_FG,muBG0,muFG0,alpha(i),w0,0);

```

```

img64 = reshape(imgS1D1_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s1d1_BDR(i) = PE_BDR;
end

imgS1D1_MLE = cheetahCalc(cheetah_blocks,D1_BG,D1_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS1D1_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s1d1_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s1d1_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D1_MAP =
cheetahCalc(cheetah_blocks,D1_BG,D1_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS1D1_MAP,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s1d1_MAP(i) = PE_MAP;
end

% plot
figure('Name','PoE-log(\alpha) (Strategy1-Dataset1)')
semilogx(alpha,PE_s1d1_BDR)
grid on
hold on
semilogx(alpha,PE_s1d1_MLE)
hold on
semilogx(alpha,PE_s1d1_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy1-Dataset1)')

%% S1-Dataset2
%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s1d2_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D2_BDR =
cheetahCalc(cheetah_blocks,D2_BG,D2_FG,muBG0,muFG0,alpha(i),w0,0);
    img64 = reshape(imgS1D2_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s1d2_BDR(i) = PE_BDR;
end

imgS1D2_MLE = cheetahCalc(cheetah_blocks,D2_BG,D2_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS1D2_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s1d2_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s1d2_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D2_MAP =
cheetahCalc(cheetah_blocks,D2_BG,D2_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS1D2_MAP,size(cheetah_dw,1),size(cheetah_dw,2));

```

```

        PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
        PE_s1d2_MAP(i) = PE_MAP;
    end

% plot
figure('Name','PoE-log(\alpha) (Strategy1-Dataset2)')
semilogx(alpha,PE_s1d2_BDR)
grid on
hold on
semilogx(alpha,PE_s1d2_MLE)
hold on
semilogx(alpha,PE_s1d2_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy1-Dataset2)')
%% S1-Dataset3
%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s1d3_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D3_BDR =
cheetahCalc(cheetah_blocks,D3_BG,D3_FG,muBG0,muFG0,alpha(i),w0,0);
    img64 = reshape(imgS1D3_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s1d3_BDR(i) = PE_BDR;
end

imgS1D3_MLE = cheetahCalc(cheetah_blocks,D3_BG,D3_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS1D3_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s1d3_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s1d3_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D3_MAP =
cheetahCalc(cheetah_blocks,D3_BG,D3_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS1D3_MAP,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s1d3_MAP(i) = PE_MAP;
end

% plot
figure('Name','PoE-log(\alpha) (Strategy1-Dataset3)')
semilogx(alpha,PE_s1d3_BDR)
grid on
hold on
semilogx(alpha,PE_s1d3_MLE)
hold on
semilogx(alpha,PE_s1d3_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy1-Dataset3)')
%% S1-Dataset4

```

```

%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s1d4_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D4_BDR =
cheetahCalc(cheetah_blocks,D4_BG,D4_FG,muBG0,muFG0,alpha(i),w0,0);
    img64 = reshape(imgS1D4_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s1d4_BDR(i) = PE_BDR;
end

imgS1D4_MLE = cheetahCalc(cheetah_blocks,D4_BG,D4_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS1D4_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s1d4_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s1d4_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS1D4_MAP =
cheetahCalc(cheetah_blocks,D4_BG,D4_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS1D4_MAP,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s1d4_MAP(i) = PE_MAP;
end

% plot
figure('Name','PoE-log(\alpha) (Strategy1-Dataset4)')
semilogx(alpha,PE_s1d4_BDR)
grid on
hold on
semilogx(alpha,PE_s1d4_MLE)
hold on
semilogx(alpha,PE_s1d4_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy1-Dataset4)')

%% Strategy 2
% load('Alpha.mat');
load('Prior_2.mat');
% % P_mu|T(mu|D)=G(mu,muBG_n,covBG_n)
muBG0 = mu0_BG;
% % P_mu|T(mu|D)=G(mu,muFG_n,covFG_n)
muFG0 = mu0_FG; % 1*64
%% Strategy2-Dataset1
%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s2d1_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D1_BDR =
cheetahCalc(cheetah_blocks,D1_BG,D1_FG,muBG0,muFG0,alpha(i),w0,0);
    img64 = reshape(imgS2D1_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s2d1_BDR(i) = PE_BDR;

```

```

end

imgS2D1_MLE = cheetahCalc(cheetah_blocks,D1_BG,D1_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS2D1_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s2d1_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s2d1_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D1_MAP =
cheetahCalc(cheetah_blocks,D1_BG,D1_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS2D1_MAP,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s2d1_MAP(i) = PE_MAP;
end

% plot
figure('Name','PoE-log(\alpha) (Strategy2-Dataset1)')
semilogx(alpha,PE_s2d1_BDR)
grid on
hold on
semilogx(alpha,PE_s2d1_MLE)
hold on
semilogx(alpha,PE_s2d1_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy2-Dataset1)')

%% S2-Dataset2
%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s2d2_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D2_BDR =
cheetahCalc(cheetah_blocks,D2_BG,D2_FG,muBG0,muFG0,alpha(i),w0,0);
    img64 = reshape(imgS2D2_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s2d2_BDR(i) = PE_BDR;
end

imgS2D2_MLE = cheetahCalc(cheetah_blocks,D2_BG,D2_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS2D2_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s2d2_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s2d2_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D2_MAP =
cheetahCalc(cheetah_blocks,D2_BG,D2_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS2D2_MAP,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s2d2_MAP(i) = PE_MAP;
end

```

```

% plot
figure('Name','PoE-log(\alpha) (Strategy2-Dataset2)')
semilogx(alpha, PE_s2d2_BDR)
grid on
hold on
semilogx(alpha, PE_s2d2_MLE)
hold on
semilogx(alpha, PE_s2d2_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy2-Dataset2)')
%% S1-Dataset3
%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s2d3_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D3_BDR =
cheetahCalc(cheetah_blocks,D3_BG,D3_FG,muBG0,muFG0,alpha(i),w0,0);
    img64 = reshape(imgS2D3_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s2d3_BDR(i) = PE_BDR;
end

imgS2D3_MLE = cheetahCalc(cheetah_blocks,D3_BG,D3_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS2D3_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s2d3_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s2d3_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D3_MAP =
cheetahCalc(cheetah_blocks,D3_BG,D3_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS2D3_MAP,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s2d3_MAP(i) = PE_MAP;
end

% plot
figure('Name','PoE-log(\alpha) (Strategy2-Dataset3)')
semilogx(alpha, PE_s2d3_BDR)
grid on
hold on
semilogx(alpha, PE_s2d3_MLE)
hold on
semilogx(alpha, PE_s2d3_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy2-Dataset3)')
%% S1-Dataset4
%function [img64] =
cheetahCalc(cheetah_blocks,D_BG,D_FG,mu0_BG,mu0_FG,alpha,w0,str)
PE_s2d4_BDR = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D4_BDR =
cheetahCalc(cheetah_blocks,D4_BG,D4_FG,muBG0,muFG0,alpha(i),w0,0);

```



```

img64 = reshape(imgS2D4_BDR,size(cheetah_dw,1),size(cheetah_dw,2));
PE_BDR = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s2d4_BDR(i) = PE_BDR;
end

imgS2D4_MLE = cheetahCalc(cheetah_blocks,D4_BG,D4_FG,muBG0,muFG0,alpha(1),w0,2);
img64 = reshape(imgS2D4_MLE,size(cheetah_dw,1),size(cheetah_dw,2));
PE_MLE = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
PE_s2d4_MLE = ones(1,size(alpha,2))*PE_MLE;

PE_s2d4_MAP = zeros(size(alpha));
for i = 1:size(alpha,2)
    imgS2D4_MAP =
cheetahCalc(cheetah_blocks,D4_BG,D4_FG,muBG0,muFG0,alpha(i),w0,1);
    img64 = reshape(imgS2D4_MAP,size(cheetah_dw,1),size(cheetah_dw,2));
    PE_MAP = ysum(ysum(abs(cheetah_maskdw-
img64),1),2)/(size(cheetah_maskdw,1)*size(cheetah_maskdw,2));
    PE_s2d4_MAP(i) = PE_MAP;
end

% plot
figure('Name','PoE-log(\alpha) (Strategy2-Dataset4)')
semilogx(alpha,PE_s2d4_BDR)
grid on
hold on
semilogx(alpha,PE_s2d4_MLE)
hold on
semilogx(alpha,PE_s2d4_MAP)
hold on
legend('Predictive','ML','MAP')
title('PoE vs log(\alpha) (Strategy2-Dataset4)')

```