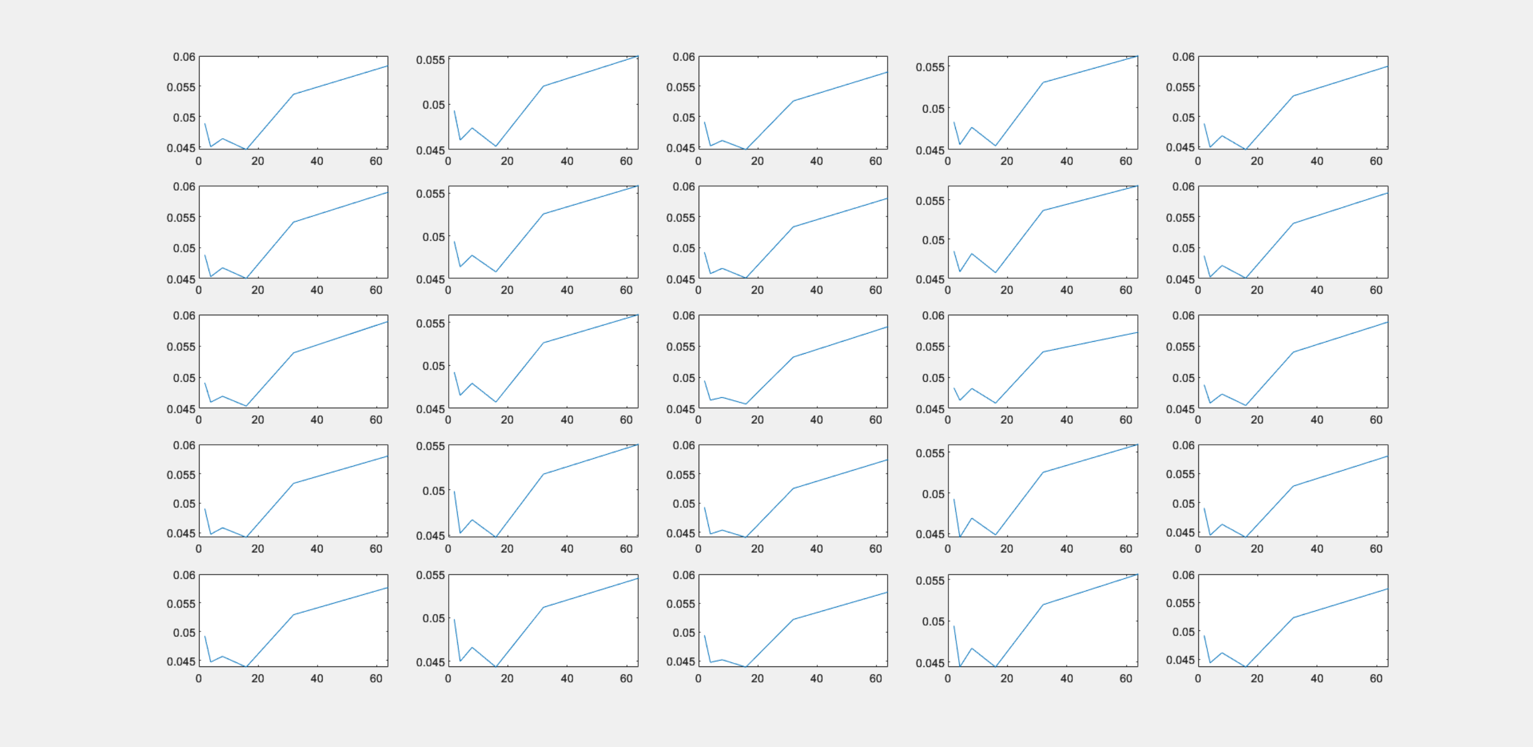
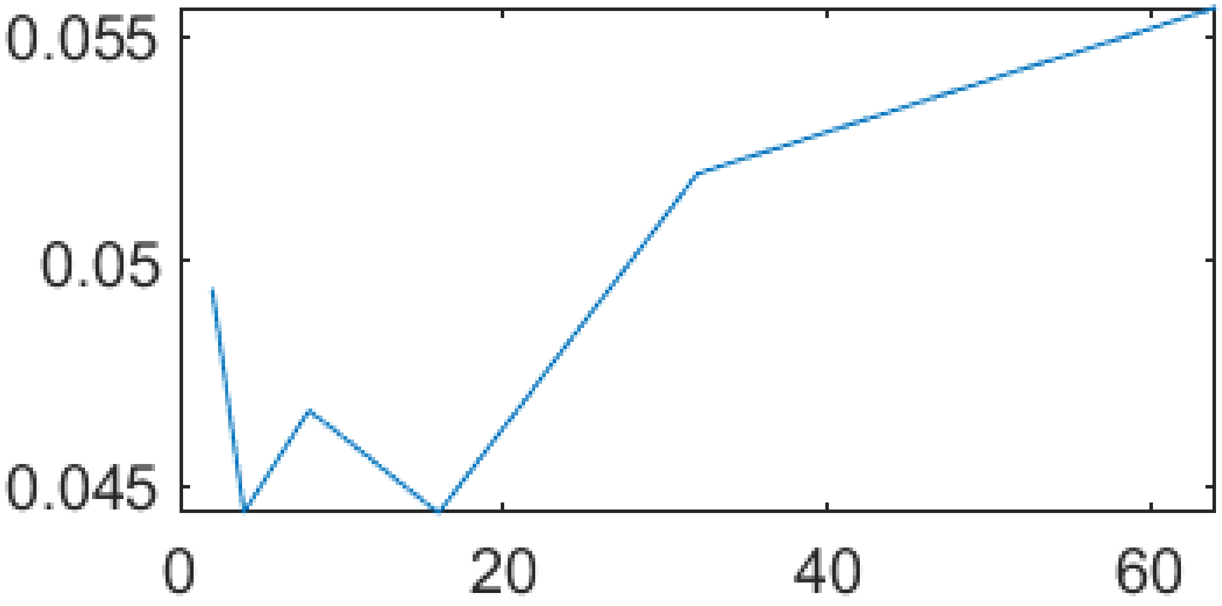
Hamad Alajeel
A16369878

ECE 271A
Quiz 4

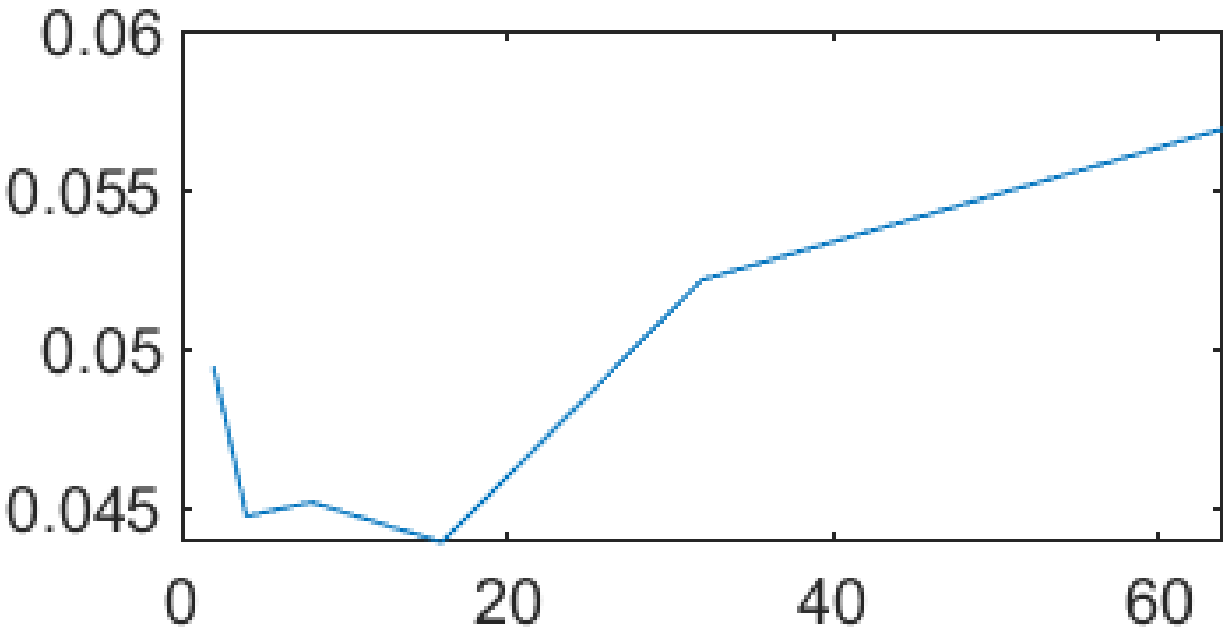
a) Figure of subplots for all 25 pairs of initializations:



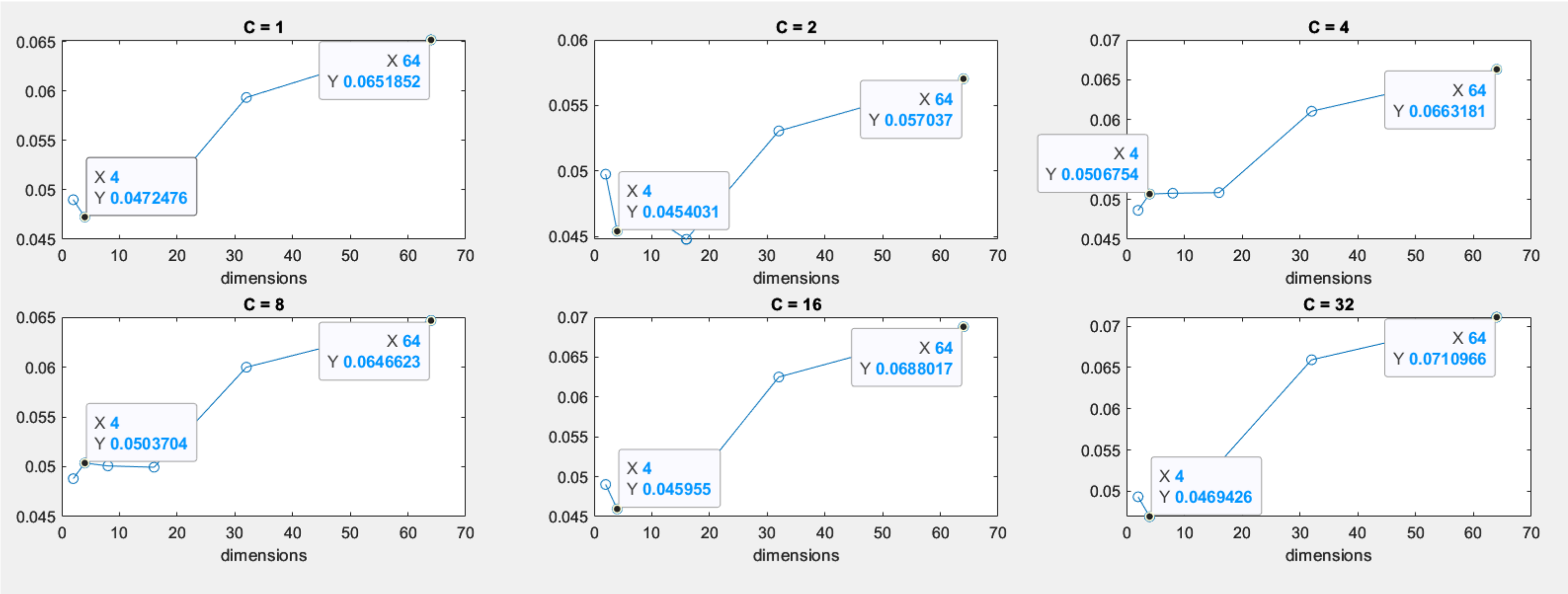
From this figure, it is obvious that different initializations of our mixture parameters will result in small deviations in errors when classifying our cheetah image. Below are two more plots to make that distinction clearer:



Hamad Alajeel
A16369878



b)



This figure shows the plots of errors vs. dimensions used for classification using different amounts of components, C. By inspecting this figure, we can see that using 2 components may be the optimal amount when finding the distribution of the complete data for each of the classes. Changing the number of mixture components has a marked effect on the probability of error for this classification problem because we approach a distribution closer to the true distributions of the complete data of the cheetah and grass classes when we do that. We see that using more than or less than 2 components for the mixtures only increases the probability of error overall.

Hamad Alajeel
A16369878

Matlab Code:

```
%% Load Training Data
load("TrainingSamplesDCT_8_new.mat");
BG_train = TrainsampleDCT_BG;
FG_train = TrainsampleDCT_FG;
size_BG = size(BG_train,1);
size_FG = size(FG_train,1);
zig_zag = readtable("Zig-Zag Pattern.txt");
zig_zag = table2array(zig_zag);
zig_zag = zig_zag + 1;
cheetah = imread("cheetah.bmp");
cheetah = double(cheetah)/255;
cheetah_mask = imread("cheetah_mask.bmp");
cheetah_mask = double(cheetah_mask)/255;
prior_FG = size_FG/(size_BG+size_FG);
prior_BG = size_BG/(size_BG+size_FG);
%% Setting Initial parameters for EM Algorithm:
C = 8;
mu_FG = zeros(5,C,64);
mu_BG = zeros(5,C,64);
cov_FG = zeros(5,C,64,64);
cov_BG = zeros(5,C,64,64);
pi_BG = zeros(5,C);
pi_FG = zeros(5,C);
for i = 1:5
   for j = 1:C
       mu_FG(i,j,:) = normrnd(0,1,[1,1,64]);
       mu_BG(i,j,:) = normrnd(0,1,[1,1,64]);
       cov_FG(i,j,:,:) = diag(abs(normrnd(0,1,[1,64])));
       cov_BG(i,j,:,:) = diag(abs(normrnd(0,1,[1,64])));
   end
pi_BG(i,:) = rand(1,C);
pi_BG(i,:) = pi_BG(i,:)/sum(pi_BG(i,:));
pi_FG(i,:) = rand(1,C);
pi_FG(i,:) = pi_FG(i,:)/sum(pi_FG(i,:));
end
%% Implementing EM Algorithm for FG
for mix = 1:5
   h =
estim(FG_train,C,pi_FG(mix,:),squeeze(mu_FG(mix,:,:)),squeeze(cov_FG(mix,:,:,:)));
   mu_new = maxim_mu(FG_train,h,C);
   cov_new = maxim_cov(FG_train,h,C,mu_new);
   pi_new = maxim_pi(FG_train,h,C);
   counter = 0;
   while any(abs(mu_new - squeeze(mu_FG(mix,:,:))) > 0.01, "all") || any(abs(cov_new
- squeeze(cov_FG(mix,:,:,:))) > 0.01, "all") || any(abs(pi_new - pi_FG(mix,:)) >
0.01, "all")
```

```
       mu_FG(mix,:,:) = reshape(mu_new,1,C,64);
       cov_FG(mix,:,:,:) = reshape(cov_new,1,C,64,64);
       pi_FG(mix,:) = pi_new;

       h =
estim(FG_train,C,pi_FG(mix,:),squeeze(mu_FG(mix,:,:)),squeeze(cov_FG(mix,:,:,:)));
       mu_new = maxim_mu(FG_train,h,C);
       cov_new = maxim_cov(FG_train,h,C,mu_new);
       pi_new = maxim_pi(FG_train,h,C);
       counter = counter + 1;
   end
end
%% Implementing EM Algorithm for BG
for mix = 1:5
   h =
estim(BG_train,C,pi_BG(mix,:),squeeze(mu_BG(mix,:,:)),squeeze(cov_BG(mix,:,:,:)));
   mu_new = maxim_mu(BG_train,h,C);
   cov_new = maxim_cov(BG_train,h,C,mu_new);
   pi_new = maxim_pi(BG_train,h,C);
   counter = 0;
   while any(abs(mu_new - squeeze(mu_BG(mix,:,:))) > 0.02, "all") || any(abs(cov_new
- squeeze(cov_BG(mix,:,:,:))) > 0.02, "all") || any(abs(pi_new - pi_BG(mix,:)) >
0.02, "all")
       mu_BG(mix,:,:) = reshape(mu_new,1,C,64);
       cov_BG(mix,:,:,:) = reshape(cov_new,1,C,64,64);
       pi_BG(mix,:) = pi_new;

       h =
estim(BG_train,C,pi_BG(mix,:),squeeze(mu_BG(mix,:,:)),squeeze(cov_BG(mix,:,:,:)));
       mu_new = maxim_mu(BG_train,h,C);
       cov_new = maxim_cov(BG_train,h,C,mu_new);
       pi_new = maxim_pi(BG_train,h,C);
       counter = counter + 1;
   end
end
%% Applying BDR on all 25 pairs of mixtures:
dims = [2, 4, 8, 16, 32, 64];
size_dims = size(dims);
state = zeros(5,5,size_dims(2),255,270);
for k = 1:5
   for l = 1:5
       for dim = dims
           dim_no = find(dims == dim);
           for i = 4:251%32
               for j = 4:266%34
                   block = dct2(cheetah(i-3:i+4,j-3:j+4));
                   x = zeros(1, 8);
                   linear_indices = sub2ind([1, 64], ones(size(zig_zag)), zig_zag);
                   x(linear_indices) = block(:);
                   x = x(1:dim);
                   state(k,l,dim_no,i,j) = 
BDR_mix(x,squeeze(mu_BG(k,:,1:dim)),squeeze(mu_FG(l,:,1:dim)),squeeze(cov_BG(k,:,1:di
m,1:dim)),squeeze(cov_FG(l,:,1:dim,1:dim)),pi_BG(k,:),pi_FG(l,:),prior_BG,prior_FG,C)
;
               end
           end
       end
   end
end
%% Calculating errors for all 25 pairs:
Error = zeros(5,5,size_dims(2));
for i = 1:5
   for j = 1:5
       for k = 1:size_dims(2)
       Error(i,j,k) = 1-sum(squeeze(squeeze(squeeze(state(i,j,k,:,:))))*(1/255) == 
cheetah_mask,"all")/(255*270);
       end
   end
end
%% Plotting error for all 25 pairs
figure('Name','Plot of Errors for all initializations vs. dimensions');
counter = 1;
for i = 1:5
   for j = 1:5
       subplot(5,5,counter)
       plot(dims,squeeze(Error(i,j,:)))
       hold on
       counter = counter + 1;
   end
end
hold off
%% learning mixtures with different # of components for each class
dims = [2, 4, 8, 16, 32, 64];
size_dims = size(dims);
C = [1,2,4,8,16,32];
size_C = size(C);
size_C = size_C(2);
state_b = zeros(size_C,size_dims(2),255,270);
for c = C
C_ind = find(C == c);
mu_FG = zeros(c,64);
mu_BG = zeros(c,64);
cov_FG = zeros(c,64,64);
cov_BG = zeros(c,64,64);
pi_BG = zeros(1,c);
pi_FG = zeros(1,c);
```

```
for j = 1:c
       mu_FG(j,:) = normrnd(0,1,[1,1,64]);
       mu_BG(j,:) = normrnd(0,1,[1,1,64]);
       cov_FG(j,:,:) = diag(abs(normrnd(0,1,[1,64])));
       cov_BG(j,:,:) = diag(abs(normrnd(0,1,[1,64])));
end
pi_BG(1,:) = rand(1,c);
pi_BG(1,:) = pi_BG(1,:)/sum(pi_BG(1,:));
pi_FG(1,:) = rand(1,c);
pi_FG(1,:) = pi_FG(1,:)/sum(pi_FG(1,:));
   h = estim(BG_train,c,pi_BG,mu_BG,cov_BG);
   mu_new = maxim_mu(BG_train,h,c);
   cov_new = maxim_cov(BG_train,h,c,mu_new);
   pi_new = maxim_pi(BG_train,h,c);
   while any(abs(mu_new - mu_BG) > 0.02, "all") || any(abs(cov_new - cov_BG) > 0.02,
"all") || any(abs(pi_new - pi_BG) > 0.02, "all")
       mu_BG = mu_new;
       cov_BG = cov_new;
       pi_BG = pi_new;

       h = estim(BG_train,c,pi_BG,mu_BG,cov_BG);
       mu_new = maxim_mu(BG_train,h,c);
       cov_new = maxim_cov(BG_train,h,c,mu_new);
       pi_new = maxim_pi(BG_train,h,c);
   end
   h = estim(FG_train,c,pi_FG,mu_FG,cov_FG);
   mu_new = maxim_mu(FG_train,h,c);
   cov_new = maxim_cov(FG_train,h,c,mu_new);
   pi_new = maxim_pi(FG_train,h,c);
   while any(abs(mu_new - mu_FG) > 0.02, "all") || any(abs(cov_new - cov_FG) > 0.02,
"all") || any(abs(pi_new - pi_FG) > 0.02, "all")
       mu_FG = mu_new;
       cov_FG = cov_new;
       pi_FG = pi_new;

       h = estim(FG_train,c,pi_FG,mu_FG,cov_FG);
       mu_new = maxim_mu(FG_train,h,c);
       cov_new = maxim_cov(FG_train,h,c,mu_new);
       pi_new = maxim_pi(FG_train,h,c);
   end
       for dim = dims
           dim_no = find(dims == dim);
           for i = 4:251%32
               for j = 4:266%34
                   block = dct2(cheetah(i-3:i+4,j-3:j+4));
                   x = zeros(1, 8);
                   linear_indices = sub2ind([1, 64], ones(size(zig_zag)), zig_zag);
                   x(linear_indices) = block(:);
                   x = x(1:dim);
                   state_b(C_ind,dim_no,i,j) = 
BDR_mix(x,mu_BG(:,1:dim),mu_FG(:,1:dim),cov_BG(:,1:dim,1:dim),cov_FG(:,1:dim,1:dim),p
i_BG,pi_FG,prior_BG,prior_FG,c);
               end
           end
       end
end
%% Calculating errors for part b:
Error_b = zeros(size_C,size_dims(2));
for i = 1:size_C
       for j = 1:size_dims(2)
           Error_b(i,j) = 1-sum(squeeze(squeeze(state_b(i,j,:,:)))*(1/255) == 
cheetah_mask,"all")/(255*270);
       end
end
%% Plotting errors for part b:
figure;
for c = C
C_ind = find(C==c);
subplot(size_C/2,size_C/2,C_ind)
plot(dims,Error_b(C_ind,:),'-o')
title("C = " + c)
xlabel("dimensions")
end
%% Visualization:
figure;
colormap(gray(255))
imagesc(squeeze(squeeze(state_b(1,2,:,:))));
title("Image Segmentation of Cheetah Image:")
%% Test Site:
any(isnan(state),"all")
A = [1,2,3;2,3,4;4,5,5]
any(isnan(A),"all")
%% Functions:
function val = G(x,mu,cov)
   dim = size(x);
   dim = dim(2);
   val = (1/((det(squeeze(cov))*(2*pi)^(dim))^(1/2)))*exp(-1/2 * 
(x-mu)*inv(squeeze(cov))*(x-mu)');
end
function pi_new= maxim_pi(data,h,C)
size_data = size(data);
size_data = size_data(1);
pi_new = zeros(1,C);
h_sum = sum(h,1);
for i = 1:C
temp = h_sum(i);
```

```
pi_new(1,i) = temp/size_data;
end
end
function mu_new = maxim_mu(data,h,C)
mu_new = zeros(C,64);
size_data = size(data);
size_data = size_data(1);
denom_vec = sum(h,1);
for j = 1:C
   temp = zeros(1,64);
   denom = denom_vec(j);
   for i = 1:size_data
       temp = temp + h(i,j).*data(i,:);
   end
   if (denom ~= 0)
       mu_new(j,:) = temp(1,:)/denom;
   else
       mu_new(j,:) = ones(1,64)*0.01;
   end
end
end
function cov_new = maxim_cov(data,h,C,mu_new)
cov_new = zeros(C,64,64);
size_data = size(data);
size_data = size_data(1);
denom_vec = sum(h,1);
exp_bool = 0;
for j = 1:C
   temp = zeros(64,64);
   denom = denom_vec(j);
   for i = 1:size_data
       temp = temp + h(i,j).*(data(i,:)-mu_new(j,:))'*(data(i,:)-mu_new(j,:));
   end
   cov_new(j,:,:) = reshape(diag(diag(temp/denom)),1,64,64);

   cov_new(j,isnan(squeeze(cov_new(j,:,:)))) = 0;
   if any(diag(squeeze(cov_new(j,:,:))==0))
       exp_bool = 1;
   end

end
if exp_bool == 1 || exp_bool == 0
   for j = 1:C
       cov_new(j,:,:) = cov_new(j,:,:) + reshape(eye(64)*0.01,1,64,64);
   end
end
end
function h = estim(data,C,pi,mu,cov)
```

Hamad Alajeel
A16369878

```
size_data = size(data);
size_data = size_data(1);
h = zeros(size_data,C);
   for i = 1:size_data
       x_i = data(i,:);
       denom = 0;
       for k = 1:C
           denom = denom + pi(1,k)*G(x_i,mu(k,:),cov(k,:,:));
       end
       for j = 1:C
           if denom == 0
               h(i,j) = 1/C;
           else
           h(i,j) = (G(x_i,mu(j,:),cov(j,:,:))*pi(1,j))/(denom);
           end
       end
   end
end
function value = BDR_mix(x,mu_BG,mu_FG,cov_BG,cov_FG,pi_BG,pi_FG,prior_BG,prior_FG,C)
   prob_BG = 0;
   prob_FG = 0;
   for i = 1:C
   prob_BG = prob_BG + G(x,mu_BG(i,:),cov_BG(i,:,:))*pi_BG(1,i);
   prob_FG = prob_FG + G(x,mu_FG(i,:),cov_FG(i,:,:))*pi_FG(1,i);
   end
   prob_BG = prob_BG * prior_BG;
   prob_FG = prob_FG * prior_FG;

   if prob_BG > prob_FG
       value = 0;
   else
       value = 255;
   end
end
```