```
images_75 = cell(1,75); % Creates a 1-by-75 cell array of empty
matrices
images_30 = cell(1,30); % Creates a 1-by-30 cell array of empty
 matrices
files = cell(1,75);
image vec = zeros(900,1);
count 30=1;
sum_30 = zeros(30,30);
mean_ = zeros(30,30);
mean face = zeros(900,1);
X \text{ mat} = zeros(900,30);
C = zeros(900,900);
dir_files = dir('proj03_face_images/*.bmp'); % dir: lists files and
 folders in the current folder.
num_files = length(dir_files); % Number of files found
for i = 1:num files %1 to 75
    currentfilename = [dir_files(i).name]; %Index into the dir_files
 to access a particular item
    currentimage = double(imread(currentfilename)); % read current
 image file and save it as an image
    image_vec = reshape(currentimage,900,1);
    images 75\{i\} = image vec; % append current image to list of 75
 image matrices (array of vectors)
    % getting 30 images
    if (mod(i,5)>0) \&\& (mod(i,5)<4)
        sum 30 = sum 30 + currentimage; % does NOT work. gives matrix
 of
        %all elements 755
          image_vec = reshape(currentimage,900,1);
        images 30{count 30} = image vec; % append current image to
 list of 30 image vectors 900 x 1
        count_30 = count_30 + 1;
    end
```

```
% mean face calculation
mean_ = sum_30 / 30;
disp("Mean face:");
imshow(mean_,[]);
snapnow
mean face = reshape(mean ,900,1);
% Calculate igen_faces/basis faces
mat images 30 = cell2mat(images 30); % converts array of matices to a
single matrix 900 x 30
for col = 1:30
   X_mat(:,col) = mat_images_30(:,col) - mean_face; % 900 x 30 =
900x30 - 900x1
end
% Covariance matrix C
C = X_mat * transpose(X_mat); % C is 900 x 900
% eigen vector and eigen value calculation
[vec_mat,val_d_mat] = eigs(C,75); %returns a diagonal matrix of
eigenvalues and a matrix V whose columns are the corresponding
eigenvectors.
eigVal_col_vec = diag(val_d_mat); %returns a column vector of the main
diagonal elements of val d mat.
[d,ind] = sort(eigVal_col_vec, 'descend');
all_eig_vals = val_d_mat(ind,ind); %reorder the diagonal elements
all eig vec = vec mat(:,ind);
% top 75 eigen values
E 75 \text{ mat} = zeros(900,75);
for t=1:75
   E 75 mat(:,t)= all eig vec(:,t);
end
disp("eigen faces");
% print top 75 eigen faces
for k = 1:75
   eig_face = E_75_mat(:,k);
   eig_face_img = reshape(eig_face,30,30);
   응응응응응
   eig_face2 = vec_mat(:,k);
   eig face img2 = reshape(eig face2,30,30);
   imshow(eig_face_img2,[]);
   snapnow
```

end

```
응응응응응
end
% Coefficient
% subjectsCoeff = zeros(10,NUMBERTRAININGS(loops)*6);
mat_images_75 = cell2mat(images_75);
for i = 1:75
   coefs_mat(:,i) = transpose(vec_mat) * (double(mat_images_75(:,i)-
mean face(:,1))); % 75x1 = 900x75 * (900x1 - 900x1)
end
% Generate genuine scores and impostor scores by computing the
Euclidean distance
%between the feature vectors of every pair of face images.
imp\_score = zeros(1175,1);
gen_score = zeros(75,1);
count = 1;
g count = 1;
im_count = 1;
for i = 1:75
   for j = i+1:75
       d_= sum((coefs_mat(:,i) - coefs_mat(:,j)).^2);
       dist = sqrt(d );
       diff_ (count,1) = dist;
       count = count + 1;
       if idivide(j-1,int32(5)) ~= idivide(i-1,int32(5))
           imposter(im_count) = dist;
           im count = im count + 1;
       else
          genuine(g_count) = dist;
          g_count = g_count + 1;
       end
```

3

end

end

```
figure();
histogram(genuine);
hold on;
histogram(imposter);
xlabel("Distance")
ylabel("Count")
legend("Genuine", "Imposter")
hold off;
t_genuine = transpose(genuine);
t_imposter = transpose(imposter);
drawROC(t_genuine,t_imposter,'d');

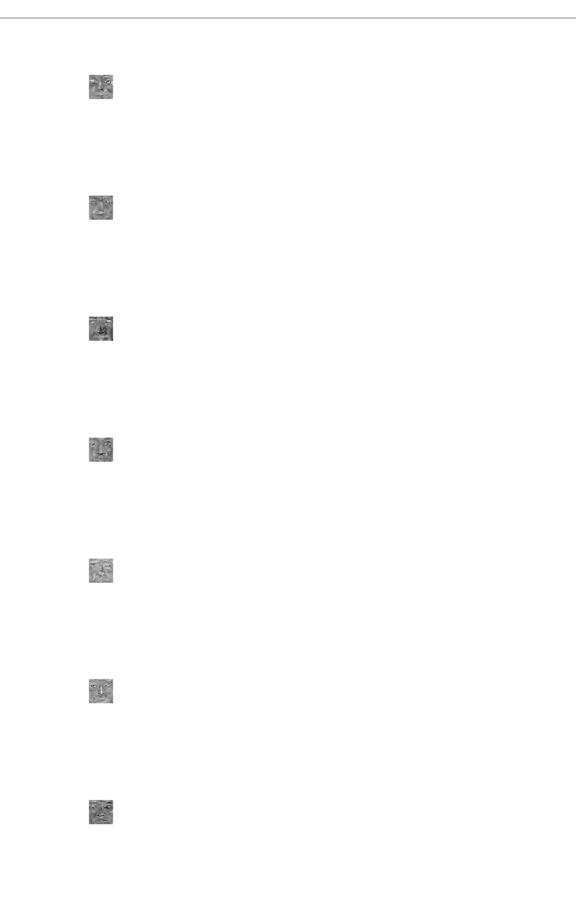
Mean face:
```

eigen faces

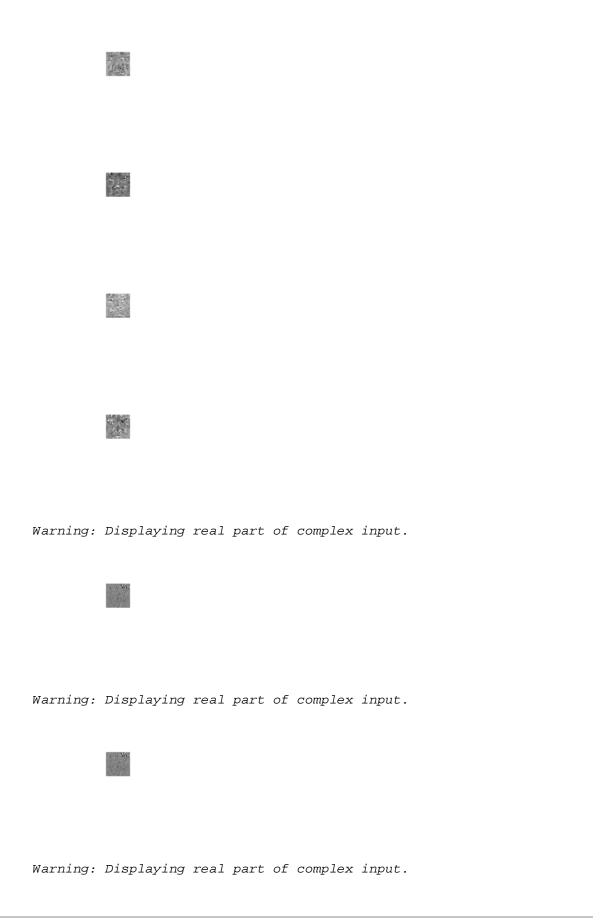
11









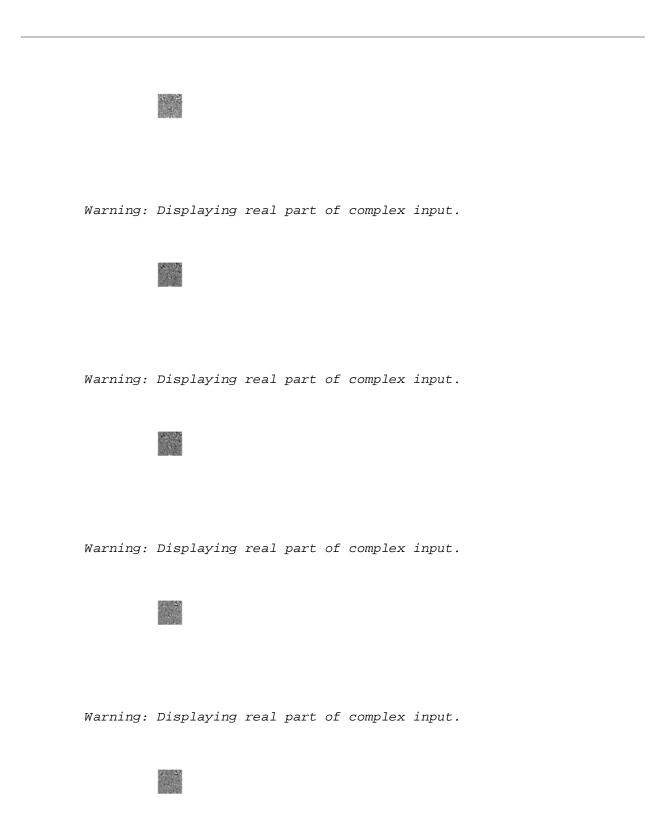


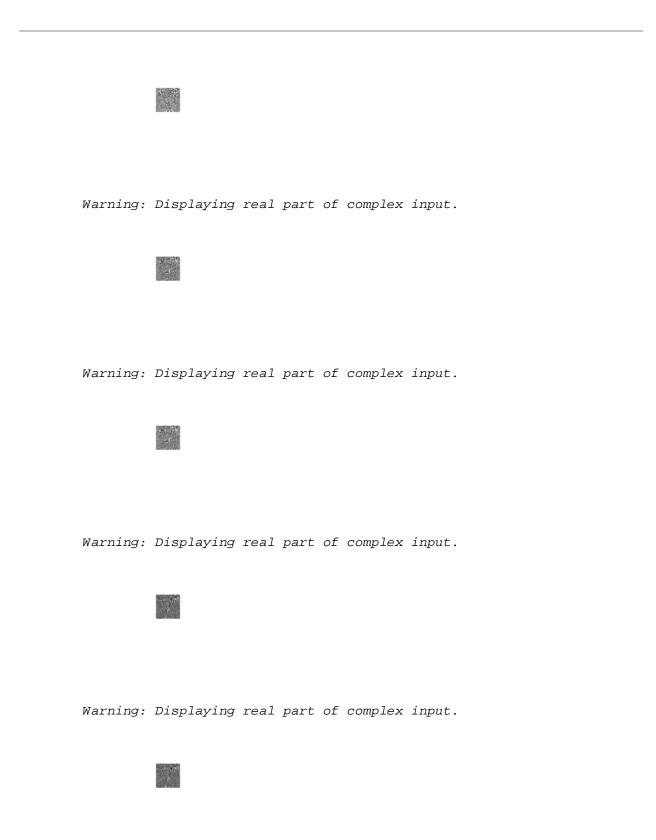
Warning: Displaying real part of complex input.

Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input.

Warning: Displaying real part of complex input.

Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input.





Warning: Displaying real part of complex input.

Warning: Displaying real part of complex input.



Warning: Displaying real part of complex input.

Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input. Warning: Displaying real part of complex input.

Index in position 2 exceeds array bounds (must not exceed 50).

Published with MATLAB® R2018b