**EX5 – PCA**

load ('ex7faces.mat')

displayData(X(1:100, :));

%% =========== Part 1: PCA on Face Data: Eigenfaces ===================

[X\_norm, mu, sigma] = featureNormalize(X);

[U, S] = pca(X\_norm);

displayData(U(:, 1:36)');

%% ============= Part 2: Dimension Reduction for Faces =================

K = 100;

Z = projectData(X\_norm, U, K);

fprintf('The projected data Z has a size of: ')

fprintf('%d ', size(Z));

%% ==== Part 3: Visualization of Faces after PCA Dimension Reduction ====

K = 100;

X\_rec = recoverData(Z, U, K);

subplot(1, 2, 1);

displayData(X\_norm(1:100,:));

title('Original faces');

axis square;

subplot(1, 2, 2);

displayData(X\_rec(1:100,:));

title('Recovered faces');

axis square;

function [X\_norm, mu, sigma] = **featureNormalize**(X)

%FEATURENORMALIZE Normalizes the features in X

mu = mean(X);

X\_norm = bsxfun(@minus, X, mu);

sigma = std(X\_norm);

X\_norm = bsxfun(@rdivide, X\_norm, sigma);

end

function [U, S] = **pca**(X)

%PCA Run principal component analysis on the dataset X

[m, n] = size(X);

Sigma = (1./m)\*(X'\*X);

[U,S,dummy] = svd(Sigma);

end

function Z = **projectData**(X, U, K)

%PROJECTDATA Computes the reduced data representation when projecting only

%on to the top k eigenvectors

U\_reduce = U(:,1:K);

Z = X\*U\_reduce;

end

function X\_rec = **recoverData**(Z, U, K)

%RECOVERDATA Recovers an approximation of the original data when using the

%projected data

U\_reduce = U(:,1:K);

X\_rec = Z\*U\_reduce';

end

function [h, display\_array] = **displayData**(X, example\_width)

%DISPLAYDATA Display 2D data in a nice grid

if ~exist('example\_width', 'var') || isempty(example\_width)

example\_width = round(sqrt(size(X, 2)));

end

colormap(gray);

[m n] = size(X);

example\_height = (n / example\_width);

display\_rows = floor(sqrt(m));

display\_cols = ceil(m / display\_rows);

pad = 1;

display\_array = - ones(pad + display\_rows \* (example\_height + pad), ...

pad + display\_cols \* (example\_width + pad));

curr\_ex = 1;

for j = 1:display\_rows

for i = 1:display\_cols

if curr\_ex > m,

break;

end

max\_val = max(abs(X(curr\_ex, :)));

display\_array(pad + (j - 1) \* (example\_height + pad) + (1:example\_height), ...

pad + (i - 1) \* (example\_width + pad) + (1:example\_width)) = ...

reshape(X(curr\_ex, :), example\_height, example\_width) / max\_val;

curr\_ex = curr\_ex + 1;

end

if curr\_ex > m,

break;

end

end

h = imagesc(display\_array, [-1 1]);

axis image off

drawnow;

end