SVD Exercise 2:

Part 1: Smiling face detector

In the following example, say we have a matrix called faces, which is a set of images of faces (of size 32x32 in the example). You can get the Yale dataset, or the ORL dataset. If you want to do classification, you'll have to label some of them as neutral or smiling.

http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

Then to get the principal components, we could do the following. Note that code is in Matlab syntax.

```
% Define the faces which are neutral or smiling for later classification
neutral = []; smile = [];

% Subtract the mean 'face' before performing PCA
h = 32; w = 32;
meanFace = mean(faces, 2);
faces = faces - repmat(meanFace, 1, numFaces);

% Perform Singular Value Decomposition
[u,d,v] = svd(faces, 0);

% Pull out eigen values and vectors
eigVals = diag(d);
eigVecs = u;

% Plot the mean sample and the first three principal components
figure; imshow(reshape(meanFace, h, w)); title('Mean Face');
figure;
title('First Eigenface');subplot(1, 3, 1); imagesc(reshape(u(:, 1), h, w)); colormap(gray);
title('Second Eigenface'); subplot(1, 3, 2); imagesc(reshape(u(:, 2), h, w)); colormap(gray);
title('Third Eigenface'); subplot(1, 3, 3); imagesc(reshape(u(:, 3), h, w)); colormap(gray);
```

```
% The cumulative energy content for the m'th eigenvector is the sum of the energy content across eigenvalues 1:m
for i = 1:numFaces
energy(i) = sum(eigVals(1:i));
end
propEnergy = energy./energy(end);

% Determine the number of principal components required to model 90% of data variance
percentMark = min(find(propEnergy > 0.9));

% Pick those principal components
eigenVecs = u(:, 1:percentMark);

% Do something with them; for example, project each of the neutral and smiling faces onto the corresponding eigenfaces
neutralFaces = faces(:, neutral); smileFaces = faces(:, smile);
neutralWeights = eigenVecs' * neutralFaces;
smileWeights = eigenVecs' * smileFaces;

% Use the coefficients of these projections to classify each smiling face
for i = 1:numFaces
weightDiff = repmat(smileWeights(:, i), 1, numFaces) - neutralWeights;
[val, ind] = min(sum(abs(weightDiff), 1));
bestMatch(i) = ind;
end
```

Part 2: Gender Classifier

In the directory you can find 2 folders: training and testing. In each of these you can find two folders: women and men. In the folder women there are pictures of women faces, while in the folder men there are men faces. Each of these images is a 36x36 gray scale image. Implement a gender classifier for face images by using SVD using the ideas similar to above.

Prepare a report complete with code, images and discussion of how successful you were with the two problems: a) classifying smiling versus neutral faces b) gender classification.