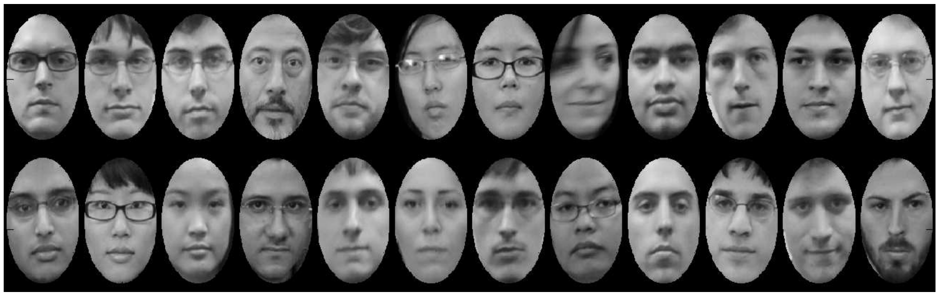
CS143: Introduction to Computer Vision
Homework Assignment 2
Eigenfaces and Recognition



Due: October 22 at 11:00am - Problem 1 and 2.

The assignment is worth 15% of your total grade.
It is graded out of a total of 100 (plus 10 possible extra points).

Goals.

In this assignment you will use what you have learned about bases, projection, templates, and eigen-representations. You will also use Gaussian densities and maximum-likelihood estimation.

This is a simple version of a real problem and consequently uses many of tools we will learn (though in a simplified form). This assignment might provide a good base for more advanced projects.

Summary.

You will be working with a training database of face images taken of the class. You will build a low-dimensional model of the face appearance using Principal Component Analysis. You will build a simple probabilistic model of faces and then use this to detect faces in images taken of the whole class.

Files you need.

The face training images that you should work with are in the directory

```
/course/cs143/asgn/asgn2/individual
```

Additionally, we provided three Matlab scripts `problem[1-2].m` in the directory `/course/cs143/asgn/asgn2/`, which you should start from. Their main purpose is to make it easier for us to grade your homework. You are therefore asked to copy those files and modify them, so that they perform the appropriate task and display your results. You are of course free to write your own scripts and/or functions and only add calls to these from the scripts we provided.

What to hand in.

You will hand in your Matlab code, which is supposed to show all your results, and any text or explanations that are required. Please hand in the modified `problem[1-2].m` scripts as well as any other Matlab scripts or functions that your solution requires. Make sure that all your results are displayed properly!

You will also hand in any text or explanations electronically. They should either be in plain text format, in Postscript, in PDF or in DOC.

To handin the assignment run the commands below from the folder that contains your solutions. The entire contents of that folder will be zipped and sent to us.

- Parts 1 and 2:

```
/course/cs143/bin/cs143_handin asgn2_p1p2
```

- Full assignment:

```
/course/cs143/bin/cs143_handin asgn2_all
```

Do all the following problems:

1. (20pts) Principal Component Analysis.

   - Perform PCA on the face images (you can use Matlab's `svd` routine to do this). Each face image has over 3000 pixels and there are many fewer training images than this. Consequently, you want to use

     ```
     [U,S,V] = svd(X,0)
     ```

     We provide a masking matrix in

`/course/cs143/data/Faces08/mugs/mask.png`

which is a 191×131 matrix with ones in the region of interest and zeros around the border. Each face image should be masked with the masking matrix to disregard regions outside of the face. Examples of the mask applied to the face images is shown on the first page of this assignment. We provide the code for loading the mask and applying it to the training images (see problem1.m).

- Display the mean face and the first six principal components (eigenfaces).
- Display the mean face varied along different principal component directions. Do this for each of the first 3 principal components. For each principal component, show 5 images that are -2, -1, 0, +1 and +2 standard deviations from the mean along that direction.
- Show a plot of the cumulative variance.
- How many bases account for 95% of the variance? What about 80% of the variance?

  Note: When computing [U, S, V]=svd(A), where A is your data matrix of images, the diagonal elements of S will be the singular values σ_i. In order to compute variance along each principal direction U_i you need to square the diagonal elements and divide by $N-1$, ie. $\lambda_i = \sigma_i^2/(N-1)$
- (5pts, Extra Credit) Find the direction in PCA space that corresponds to gender. Take a male face and make them more male and more female. Do the same for a female face. Show the results.

2. (30pts) Likelihood model.

Using the first $M = 6$ principal components from above, build an approximate probabilistic likelihood model $p(\mathbf{x}|\Omega)$ where $\mathbf{x}$ is a vectorized image patch ($\mathbf{x} \in \mathbf{R}^N$), where N is the size of the image vector, and Ω is the object class (faces in this case).

Assume that face images can be represented by their mean and covariance. In this case we would have

$$p(\mathbf{x}|\Omega) = \frac{1}{(2\pi)^{N/2}} \det(\Sigma)^{-1/2} \exp(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x}-\bar{\mathbf{x}}))$$

where $\bar{x}$ is the mean face image and Σ is the covariance matrix.

The term $(\mathbf{x} - \bar{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x} - \bar{\mathbf{x}})$ is called the Mahalanobis distance.

We can use the eigendecomposition to rewrite Σ as $US^2U^T/(N-1)$, where S and U are the matrices returned by svd(A).

Plugging that into the Mahalanobis distance we get

$$\begin{aligned} d(\mathbf{x}) &= (\mathbf{x} - \bar{\mathbf{x}})^T \left(\frac{US^2U^T}{N-1} \right)^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \\ &= (\mathbf{x} - \bar{\mathbf{x}})^T U \left(\frac{S^2}{N-1} \right)^{-1} U^T (\mathbf{x} - \bar{\mathbf{x}}) \\ &= \mathbf{y}^T \Lambda^{-1} \mathbf{y} \end{aligned}$$

where $\mathbf{y} = U^T(\mathbf{x} - \bar{\mathbf{x}})$ represents the coefficients of the projection of the data onto the subspace spanned by U^T and Λ is the diagonal matrix of variances (λ_i) as computed in the previous problem.

The Mahalanobis distance can now be written as a simple sum (remember that we are building only an approximate model so we are summing from 1 to $M = 6$):

$$d(\mathbf{x}) = \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i}$$

Define the approximate image likelihood using this formula. Also, feel free to ignore the normalization constant.

- Choose three different looking face training images, project each onto your six dimensional subspace to get vectors of coefficients representing the images. Compute the approximate version of each image using the subspace. Show the original images and the reconstructions. Compute the log likelihood of the original image using the approximate model.
- Draw six random samples from your approximate probabilistic model. Since the covariance matrix is diagonal, you can sample from each of the six directions independently. You can use `randn` in Matlab to do this. Make sure that you multiply these numbers by the corresponding standard deviations $\sqrt{\lambda_i}$ along each principal direction. Show each random face and compute its log likelihood using the approximate model.

- Comment on the random samples. Do they look like faces? If so, why? You might also want to try using more basis vectors.

3. (50pts) Coming soon...