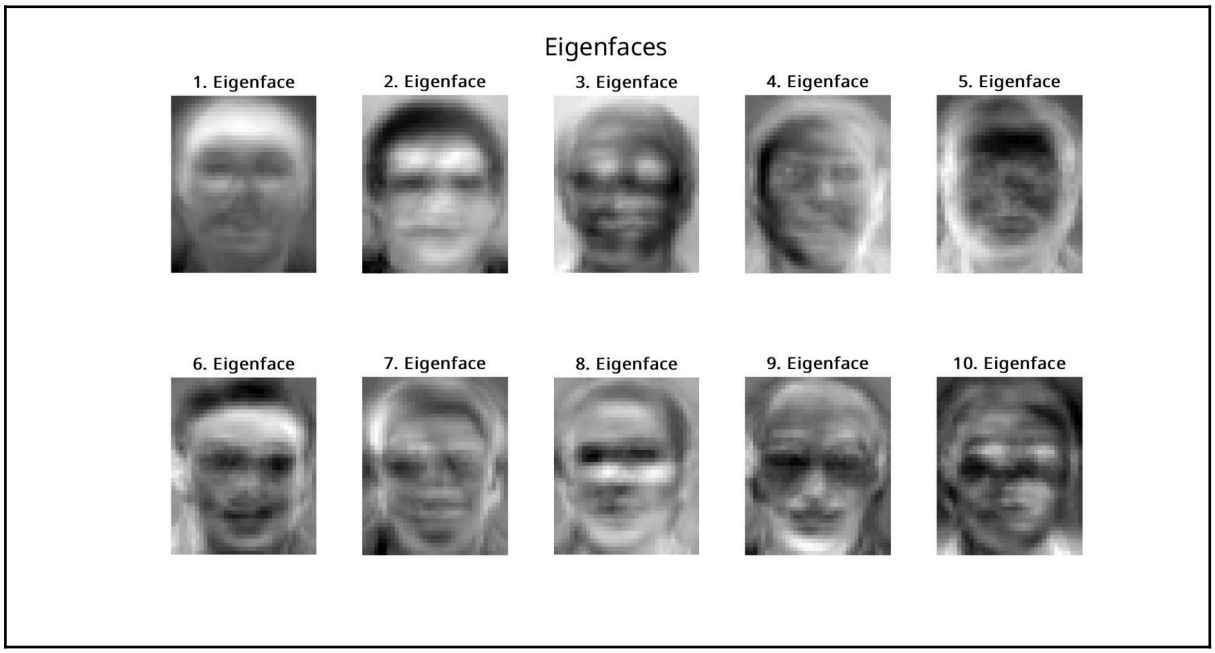
Face Recognition using Principal Component Analysis

PCA Transformation Matrix

After reading all images and converting each image into a vector using the code provided with the homework (read_images.m). I have calculated the mean of all training image vectors and subtracted this mean from each training image vector. Then I calculated eigenvectors and values of these new mean subtracted image vectors using matlab's cov() and eig() functions. By taking a previously set number(PCA_dim) of columns from the eigenvector matrix I calculated the PCA transformation matrix.

Eigenfaces

Each column in the eigenvector matrix corresponds to the image vector of an eigenface. By taking the first 10 columns and reshaping them into 36 x 44 matrices I have constructed 10 eigenfaces shown below.



These results can be reproduced by running Matlab code eigenfaces.m.

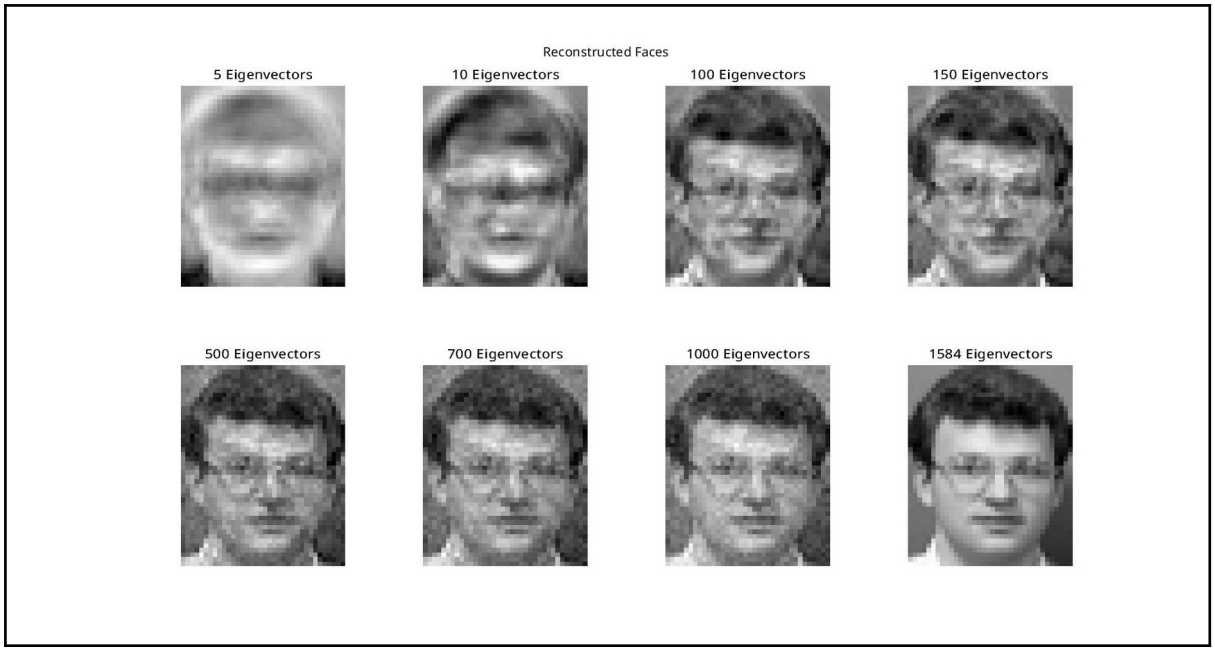
Reconstructed Faces

$\omega_i = \Psi^T \phi_i^T \quad \hat{\phi}_i = (\Psi \omega_i)^T$

By using two equations given above we can reconstruct images in the training set.

$\hat{\phi}_i = (\Psi \Psi^T \phi_i^T)^T$

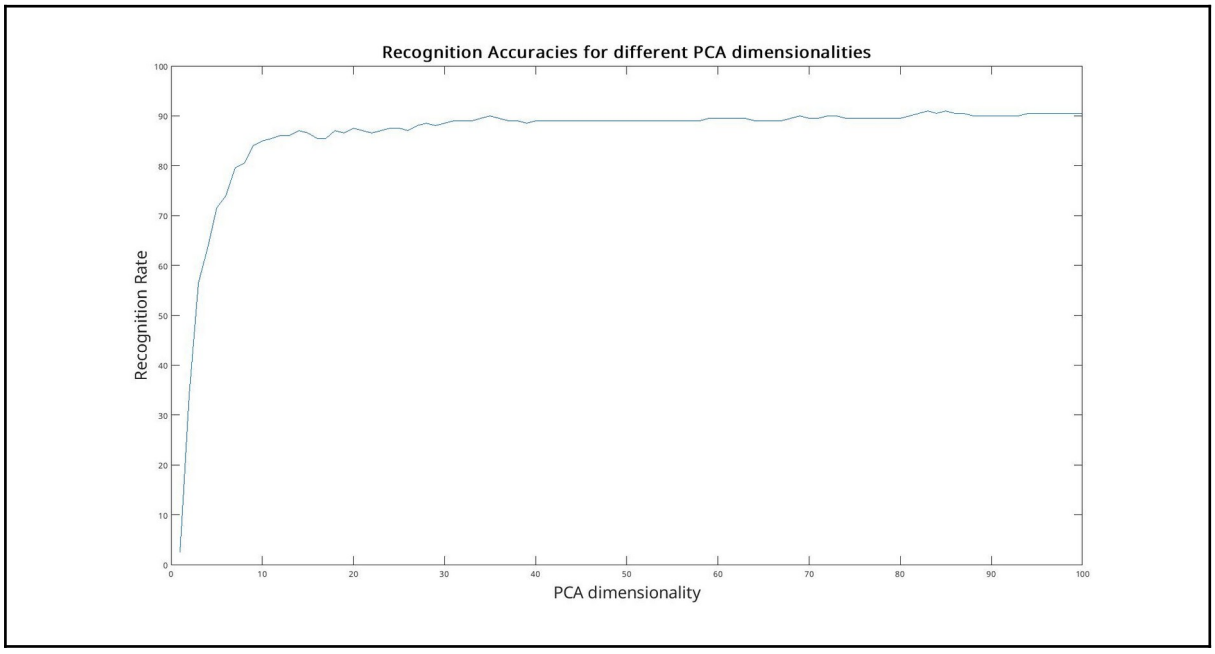
As shown below, by increasing number of eigenvectors used in PCA transformation matrix (Ψ), we can increase the qualiy of the reconstructed images.



These results can be reproduced by running Matlab code reconstruct_images.m.

Recognition Accuracy Plot

We can classify a given image by comuting the distance between feature vector of this image and feature vectors of all training images. We can simplify this computation by reducing the dimensionality of the PCA or increase accuracy of the classification by increasing the dimensionality of the PCA.



These results can be reproduced by running Matlab code recognition_accuracy.m.