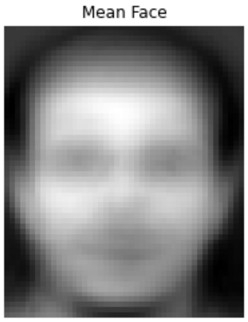
Homework #0

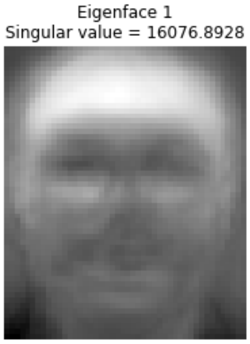
Deep Learning for Computer Vision

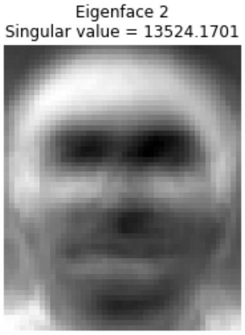
1. (20%) Perform PCA on the training set. Plot the mean face and the first four eigenfaces.

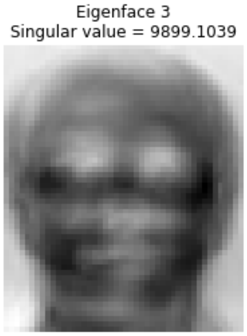
Mean face

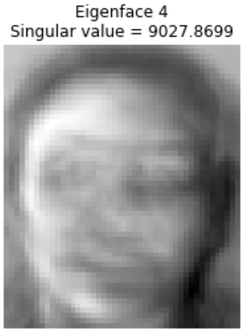


Eigenfaces (sorted by singular values)

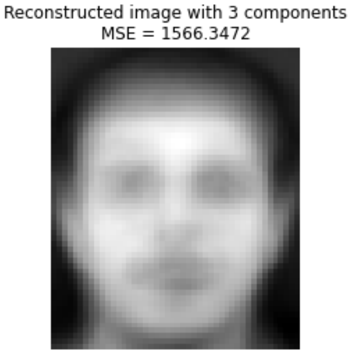


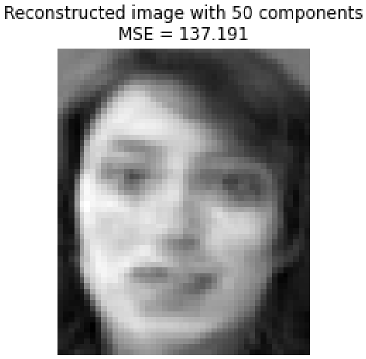


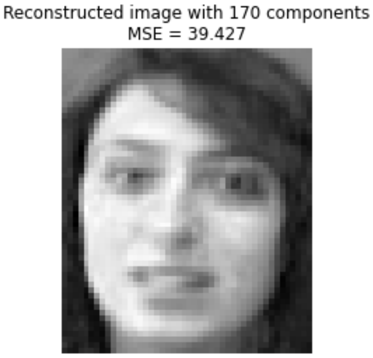


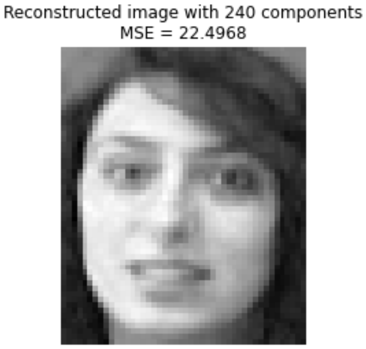


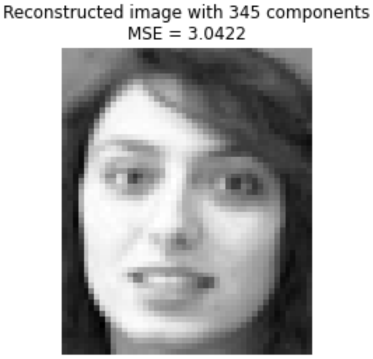
2. (20%) If the last digit of your student ID number is odd, take person 2 image 1 . If the last digit of your student ID number is even, take person 8 image 1 . Project it onto the PCA eigenspace you obtained above. Reconstruct this image using the first n = 3, 50, 170, 240, 345 eigenfaces. Plot the five reconstructed images.


Reconstructed image with 3 components
MSE = 1566.3472


Reconstructed image with 50 components
MSE = 137.191


Reconstructed image with 170 components
MSE = 39.427


Reconstructed image with 240 components
MSE = 22.4968


Reconstructed image with 345 components
MSE = 3.0422

3. (20%) For each of the five images you obtained in 2., compute the mean squared error (MSE) between the reconstructed image and the original image. Record the corresponding MSE values in your report.

The MSE values are shown in the figure titles in 2.

4. (20%) Now, apply the k-nearest neighbor algorithm to classify the testing set images. First, you will need to determine the best k and n values by 3-fold cross-validation. For simplicity, the choices for such hyperparameters are k = {1, 3, 5} and n = {3, 50, 170}. Show the cross-validation results and explain your choice for (k, n).

	n = 3	n = 50	n = 170
k = 1	Train_acc: 1.0 Valid_acc: 0.6417	Train_acc: 1.0 Valid_acc: 0.95	Train_acc: 1.0 Valid_acc: 0.9556
k = 3	Train_acc: 0.9569 Valid_acc: 0.6389	Train_acc: 0.9931 Valid_acc: 0.9306	Train_acc: 0.9931 Valid_acc: 0.9417
k = 5	Train_acc: 0.8778 Valid_acc: 0.5917	Train_acc: 0.9583 Valid_acc: 0.8778	Train_acc: 0.9556 Valid_acc: 0.8861

* Train_acc: mean of the training accuracies in 3-fold cross-validation
* Valid_acc: mean of the validation accuracies in 3-fold cross-validation
* Accuracy here means the face recognition rate

According to the highest validation accuracy 0.9556 in the above results, we choose k = 1 and n = 170 as the parameters in testing.

We can observe from the above results that the more components n we use, the recognition is more accurate. In addition, when we choose one nearest image to recognize, the accuracy is high enough.

5. (20%) Use your hyperparameter choice in 4. and report the recognition rate of the testing set.

Under k = 1 and n = 170, the recognition rate on the testing set is 0.95.