**SYDE 675**

**Programming Assignment 5**

**Wenhan Liu**

**Zhiqi Bei**

**Question 1**

Firstly, the data was loaded using the hint given.

A picture containing chart

Description automatically generated

Chart

Description automatically generated with medium confidence

With the train\_labels and test\_labels containing the training and testing data, and train\_labels and test\_labels containing training and testing labels.

**Average Face**

Next, by making an (2500,) np array and make each row containing the mean of each column in train\_data, we have created the avg\_face.

A picture containing graphical user interface

Description automatically generated

**Mean Subtraction**

Just using train\_data to subtract the avg\_face we just got, mean subtraction is done. And it’s being done for both the training and testing data.

A picture containing text, indoor, kitchen appliance, tiled

Description automatically generated

A screenshot of a video game

Description automatically generated with medium confidence

**Eigenface**

By using the svd that numpy has, it could be done simply by performing svd on Mean\_train, which is the result we just got from Mean subtraction. And the top 10 faces in V is plotted easily.

A picture containing graphical user interface

Description automatically generatedA picture containing chart

Description automatically generatedA picture containing graphical user interface

Description automatically generatedA picture containing graphical user interface

Description automatically generatedA picture containing graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidenceA picture containing graphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generatedApplication

Description automatically generated with medium confidence

**Low-Rank Approximation**

Firstly, s needs to be a diagonal matrix, and np.diag can do the job. Next, by using a for loop that ranges from 1 to 200, in each loop, Xr was calculated using the equation, and the rank-r approximation error ||X – Xr||F was calculated by using np.linalg.norm, and got appended to the list. Finally, the error list was plotted and look like follows.

Chart

Description automatically generated

**Eigenface Feature**

So two functions were created for this part. They could generate r-dimensional feature matrix F for training images, and Ftest for testing images. And the way to do is given in the hint, which is to multiply X(train\_data) to the transpose of first r rows of VT.

Text

Description automatically generated

**Face Recognition**

Logistic Regression was first imported from sklearn.



Since we have to use “One-vs-Rest” logistic regression, multi\_class = ‘ovr’ needs to be stated.



With a for loop ranges from 1 to 200, the model was trained on the features F and corresponding labels train\_labels, and it was tested and calculated the accuracy with Ftest and test\_labels. Finally, the list containing the accuracy was plotted, with the highest accuracy being 94%.

Graphical user interface, chart

Description automatically generated with medium confidence