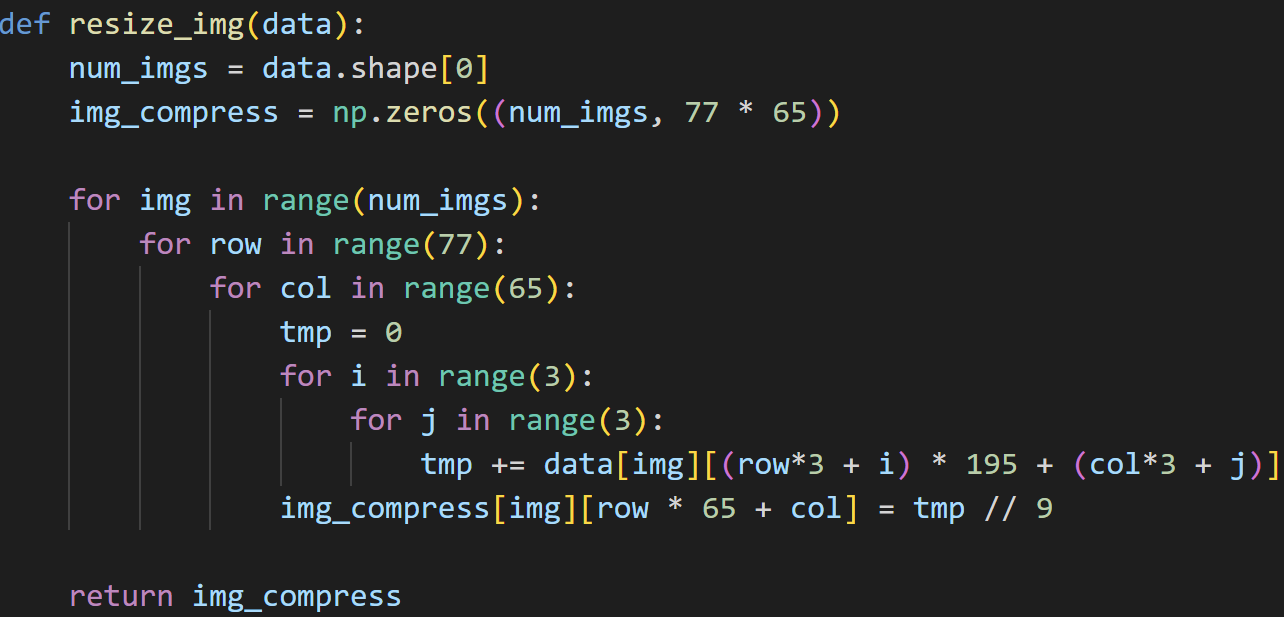
**Machine Learning HW7**

**311551094 資科工碩一 廖昱瑋**

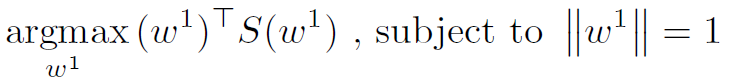
1. **Code**
2. Kernel Eigenfaces
3. Part1
4. Compress image

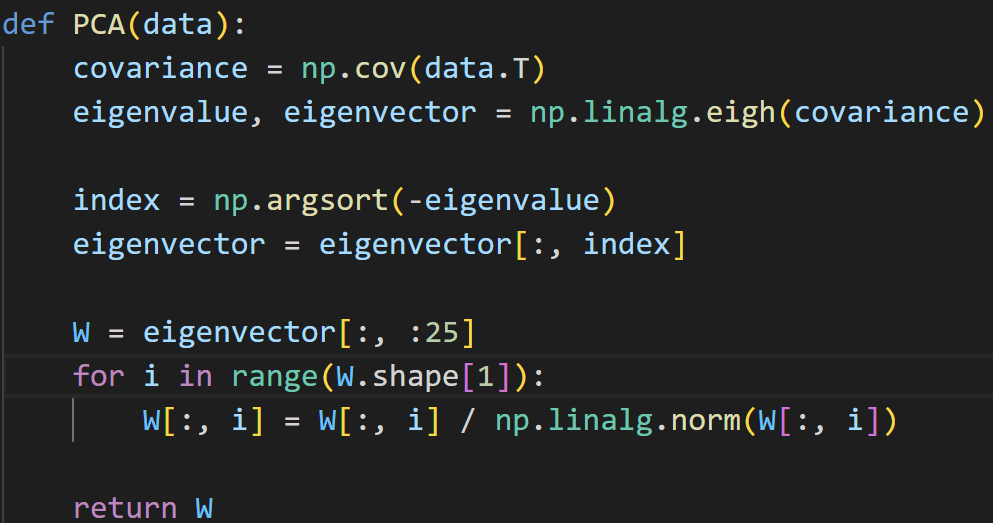
Because the dimensionality is too high in original data. Resize image to 77 \* 65 first.



1. PCA

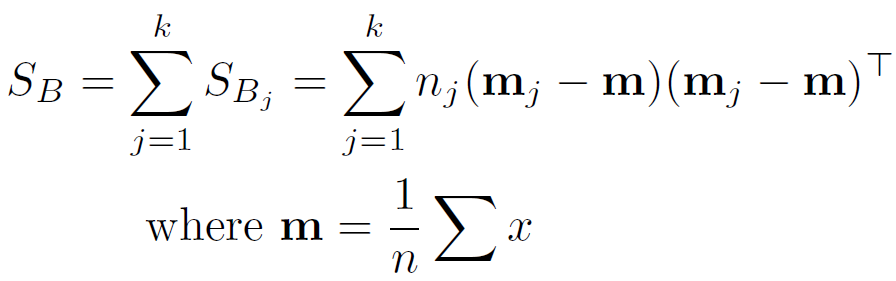
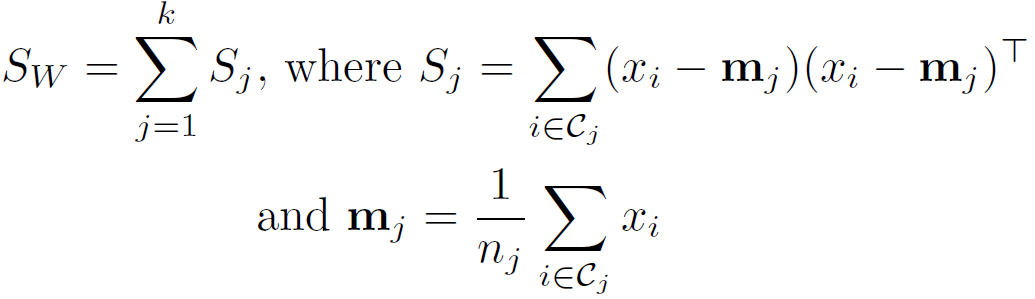
Compute the covariance matrix of data. Then, do eigen decomposition of the covariance matrix. Get 25 first largest eigenvectors and normalize them. They are stored as matrix W.



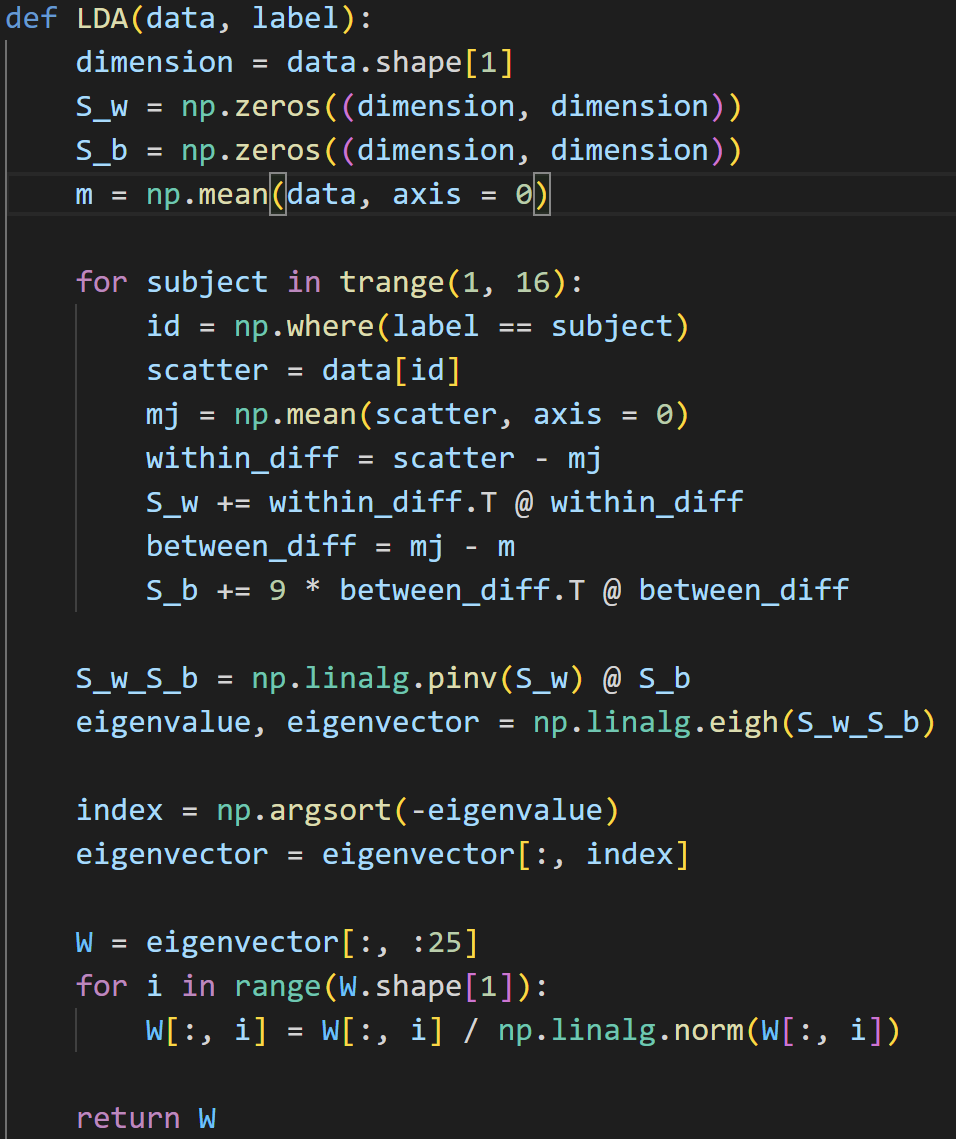


1. LDA

Compute SW, SB by the following formula.

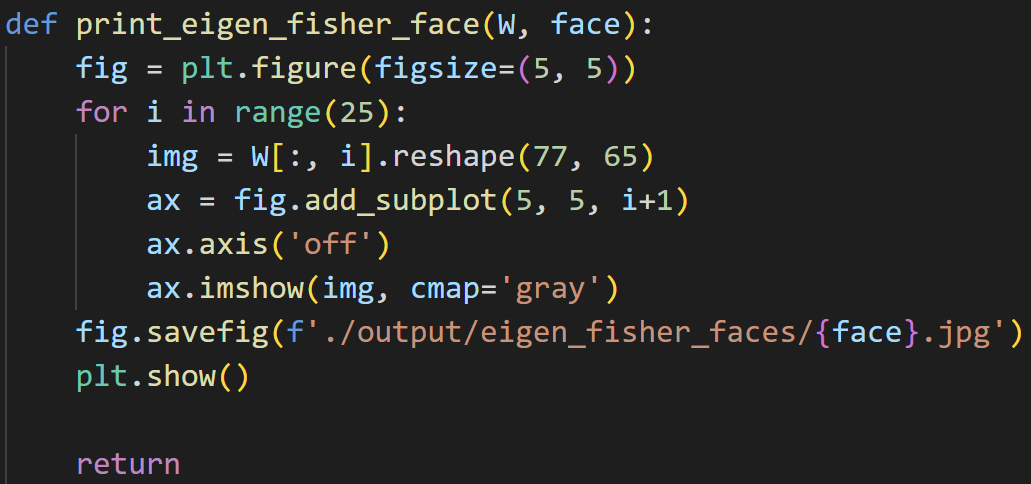


Solve the equation by doing eigen decomposition on and get first 25 largest eigenvectors. Note that SW in un-invertible when , so use pseudo inverse on SW. Normalize the eigenvectors and store them as matrix W.



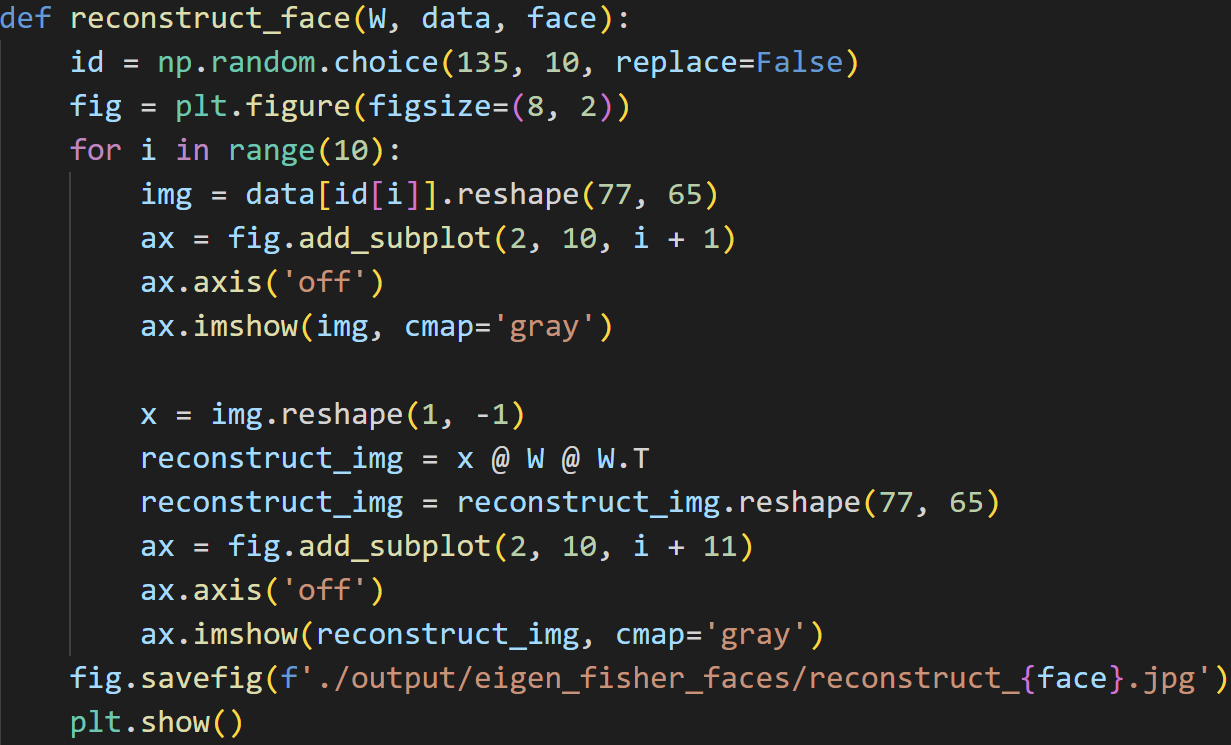
1. Eigenfaces & Fisherfaces

Print eigenfaces and fisherfaces by W.



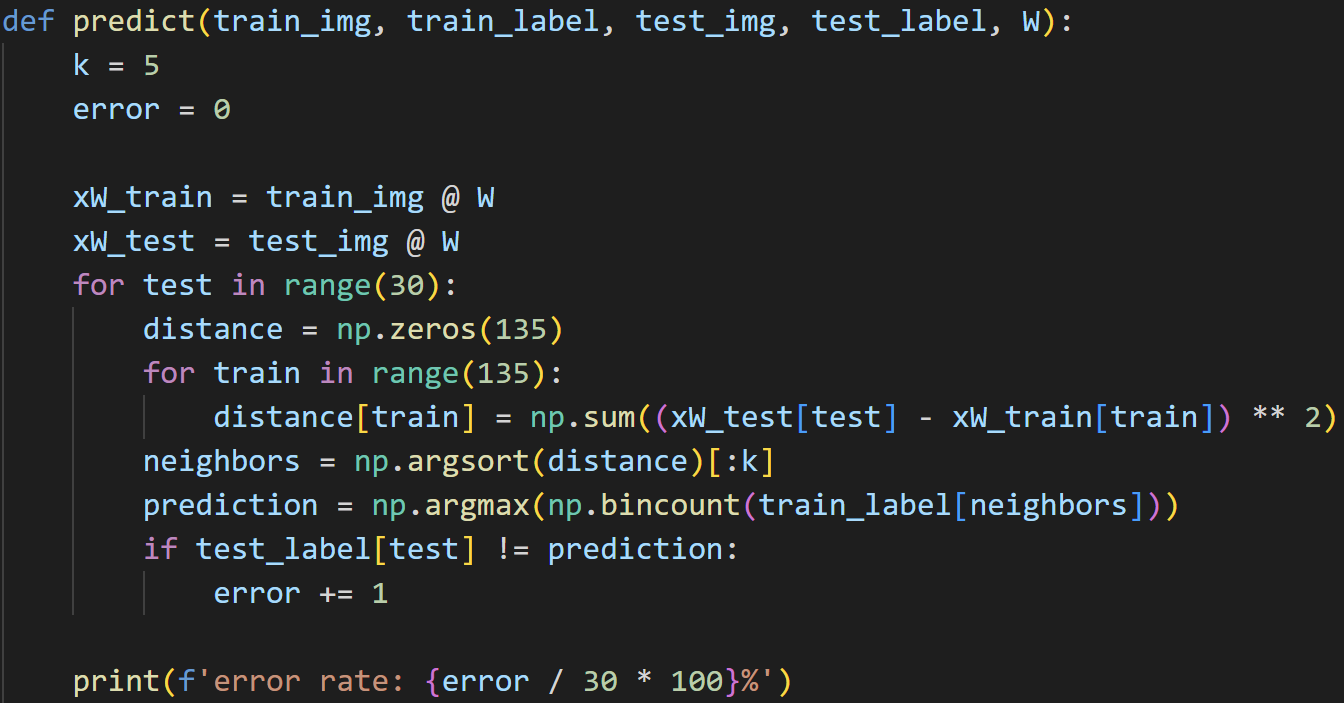
1. Reconstruct faces

Randomly choose 10 images and reconstruct images by .

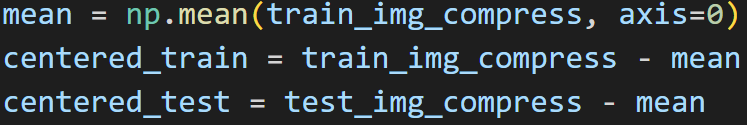


1. Part2

Compute coordinates in the space built upon principal components by . Then, compute distance between each test coordinates between every train coordinates. Get the prediction by 5 nearest neighbors.

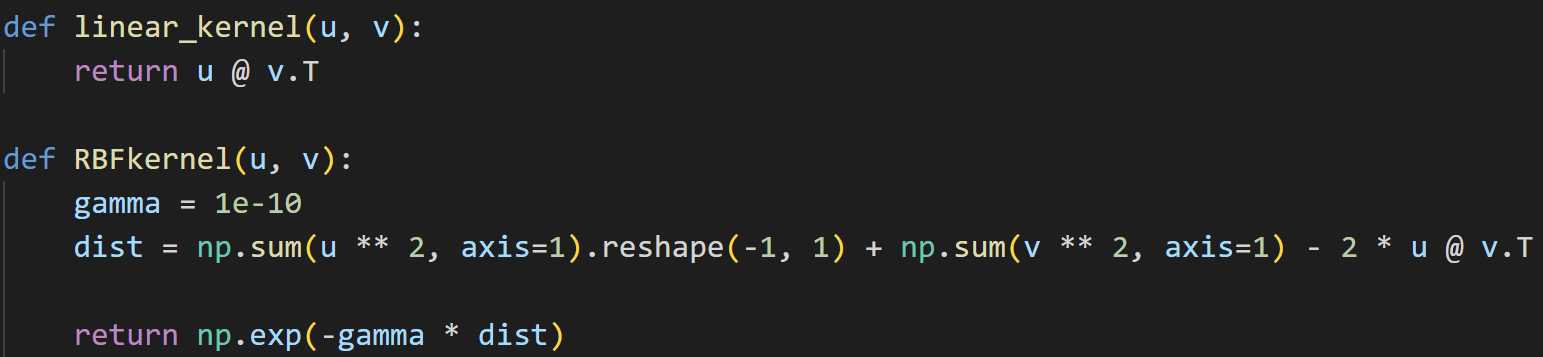


1. Part3
2. Center the data



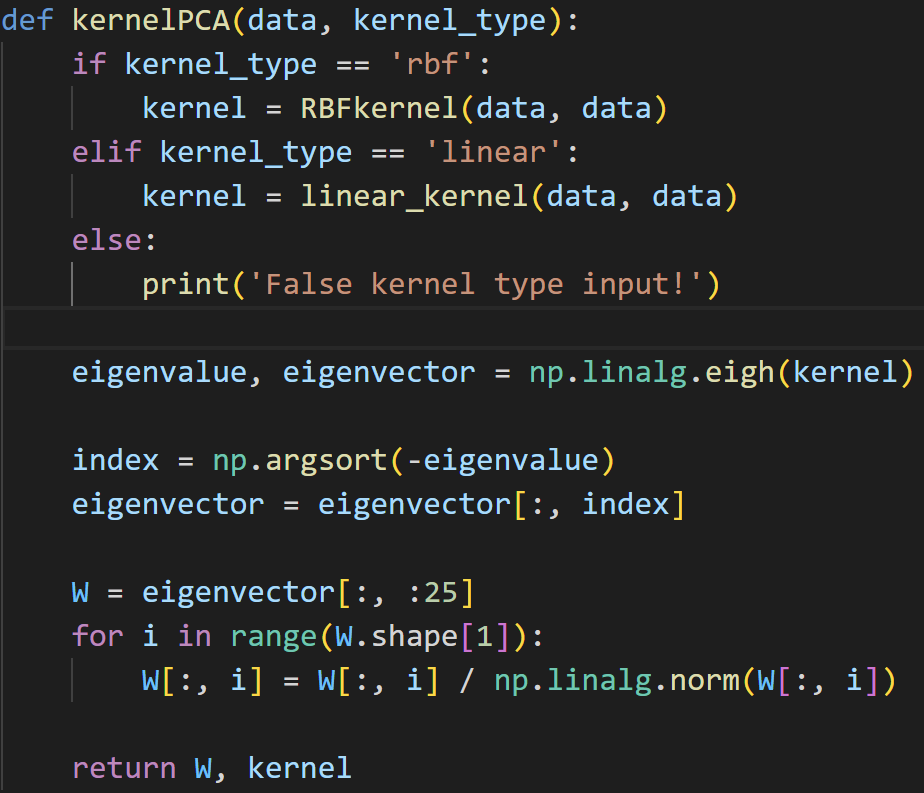
1. Kernel

Define 2 kernels, linear kernel and RBF kernel.



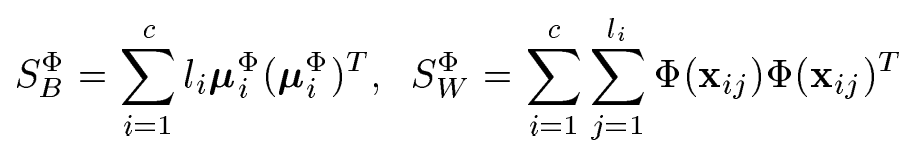
1. Kernel PCA

Compute the kernel matrix of data. Then, do eigen decomposition of the kernel matrix to solve . Get 25 first largest normalized eigenvectors.

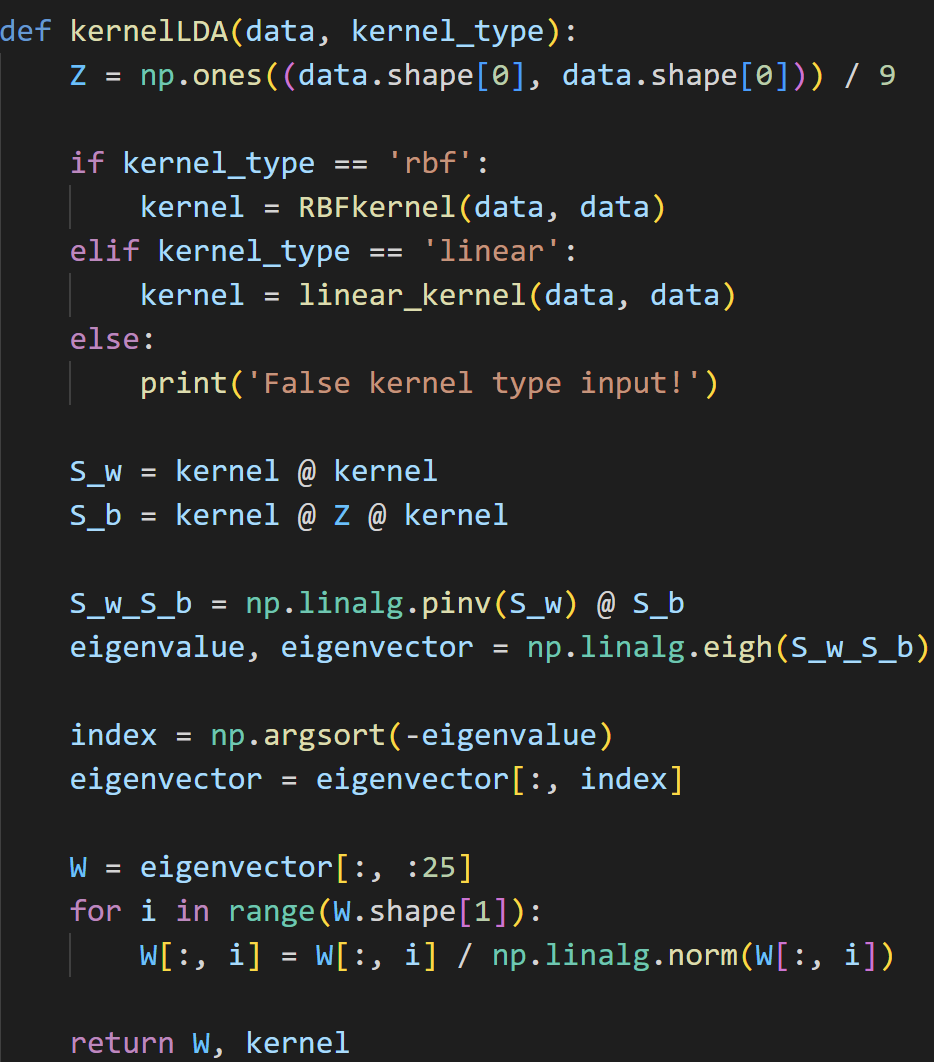


1. Kernel LDA

Compute the kernel matrix of data. Compute SW, SB by the following formula.

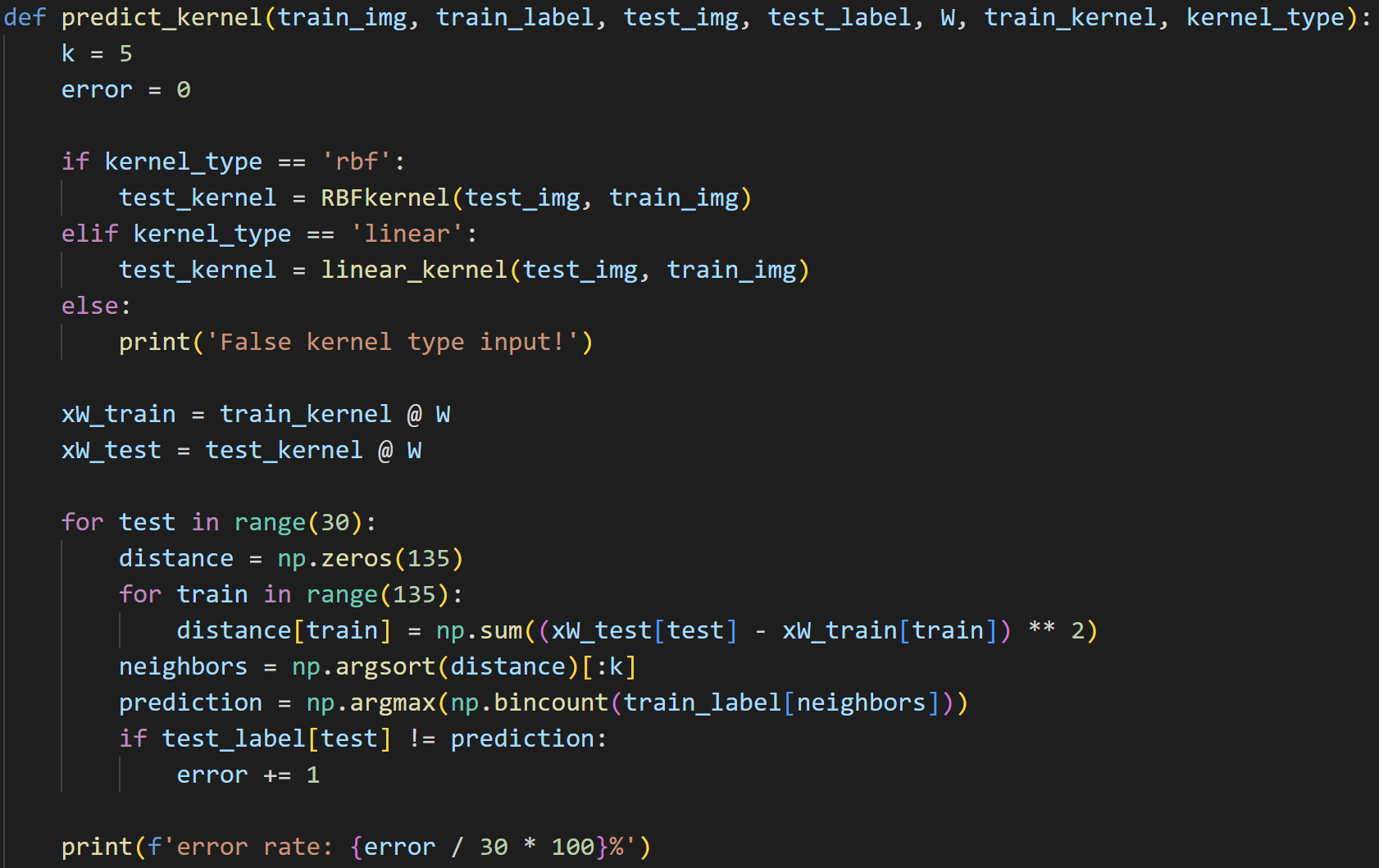


Then, do eigen decomposition of the kernel matrix to solve . Get 25 first largest normalized eigenvectors.



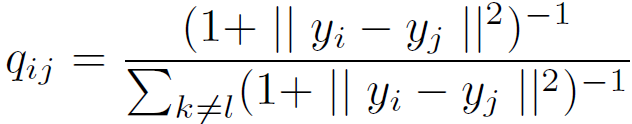
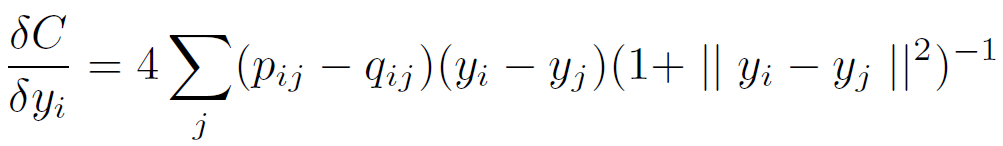
1. Compute performance

Compute kernel matrix of test data. Then, compute coordinates in the space built upon principal components by . Get the prediction by 5 nearest neighbors.

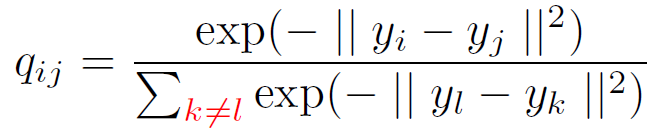
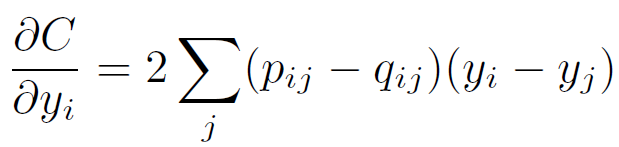


1. t-SNE
2. Part1

t-SNE:

Symmetric SNE:

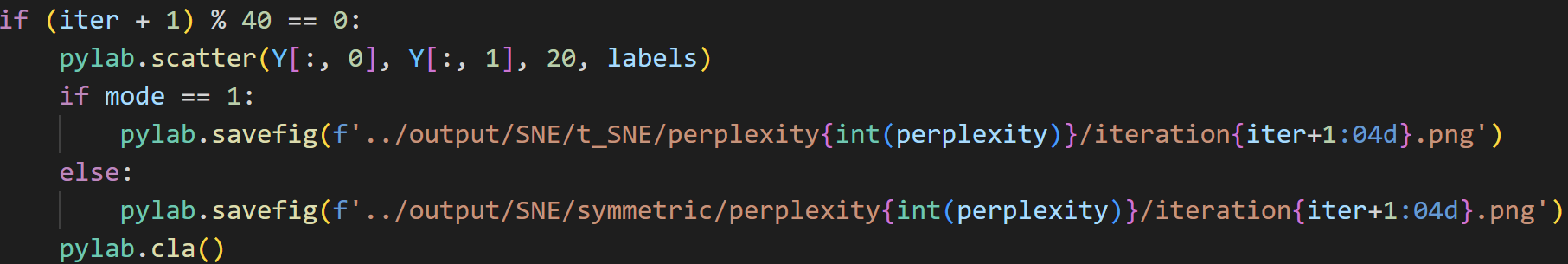
 





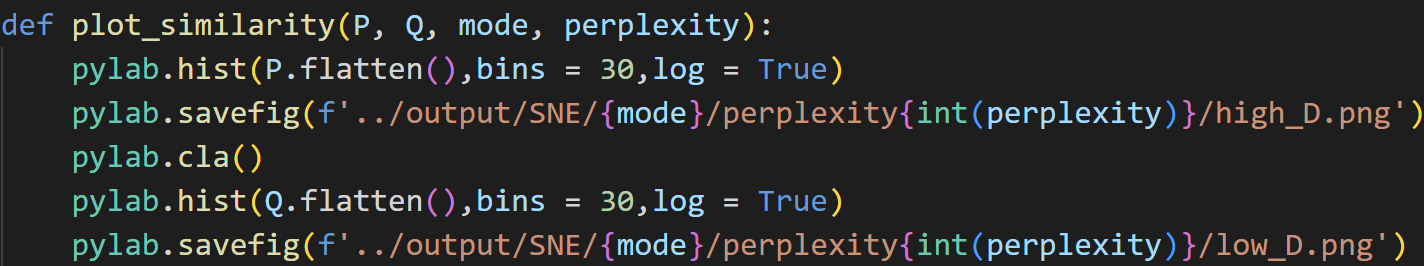
1. Part2

Save embedding images every 40 iterations.



1. Part3

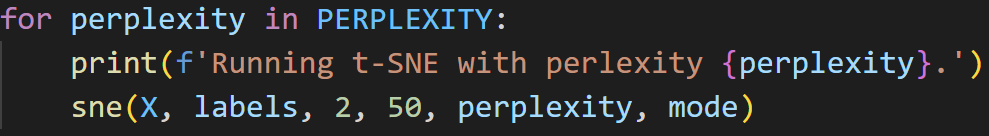
Use histogram to plot pairwise similarities in both high-dimensional space and low-dimensional space.



1. Part4

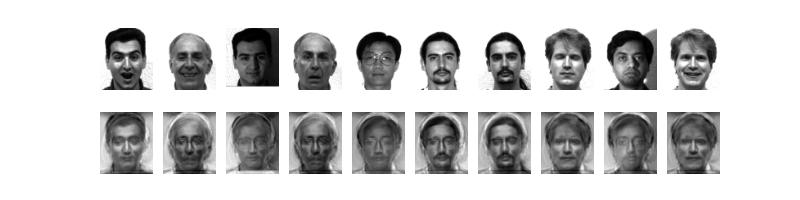
Try different perplexity values with 5, 20, 35, 50, 65.



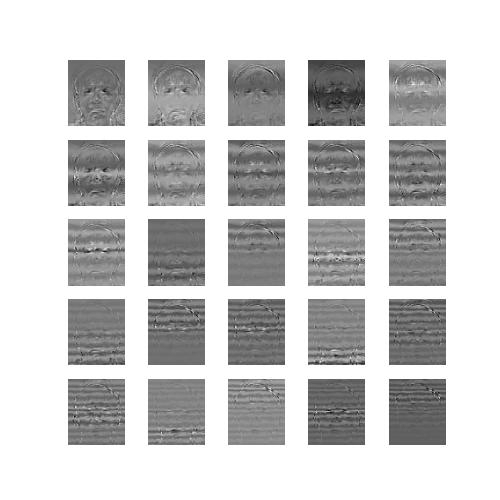
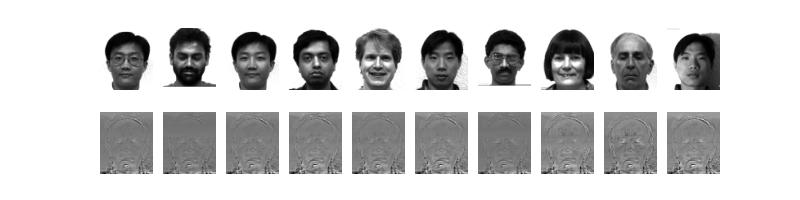


1. **Result & Discussion**
2. Kernel Eigenfaces
3. Part1

Eigenfaces and its reconstruct faces:



Fisherfaces and its reconstruct faces:

Eigenfaces and its reconstruct faces provide a good contour of original face. However, fisherfaces and its reconstruct faces are quite blurred. It’s because dimensionality of lower-dimension space should be smaller than number of classes. We have 15 classes in the dataset but set dimensionality of lower-dimension space as 25. Therefore, the remaining fisherfaces are literally grey.

1. Part2

|  |  |
| --- | --- |
| **Algorithm** | **Error rate** |
| PCA | 10.00% |
| LDA | 16.67% |

1. Part3

|  |  |  |
| --- | --- | --- |
| **Algorithm** | | **Error rate** |
| PCA | Linear kernel | 16.67% |
| RBF kernel | 20.00% |
| LDA | Linear kernel | 23.33% |
| RBF kernel | 26.67% |

Roughly, PCA performs better than LDA because dimensionality of lower-dimension space we choose in LDA is not adequate. Surprisingly, kernel method doesn’t perform better than original PCA and LDA. In my opinion, it’s because kernel trick is used to extract nonlinear features but the dimensionality in original dataset is high enough. Therefore, we don’t need to project data to higher-dimension space by kernel trick. In this experiment, linear kernel performs better than RBF kernel.

1. t-SNE
2. Project data onto 2D space

|  |  |  |
| --- | --- | --- |
| **Perplexity** | **t-SNE** | **Symmetric SNE** |
| 5 |  |  |
| 20 |  |  |
| 35 |  |  |
| 50 |  |  |
| 65 |  |  |

We can see that symmetric SNE has crowded problem which t-SNE doesn’t have. If perplexity is small, the embedding will tend to be fluffy because repulsive forces will dominate. If perplexity is large, the embedding will tend to be denser structure.

1. Pairwise similarities in high-D low-D space

Result of pairwise similarities in high-D low-D space with perplexity set as 35.

|  |  |  |
| --- | --- | --- |
| **Dimension** | **t-SNE** | **Symmetric SNE** |
| High-D |  |  |
| Low-D |  |  |

The pairwise similarity of t-SNE in low-D is range from 0 to 8e-5 and the pairwise similarity of symmetric SNE in low-D is range from 0 to 3.5e-6. Therefore, points do not need to be too far to achieve low probability in symmetric SNE which may lead to crowded problem.

1. **Discussion**
2. The meaning of eigenfaces

Eigenfaces are a specific case of doing PCA on human faces. In PCA, we want to preserve maximum the variance when doing projection. It allows us to drop unnecessary information by selecting only the principal components. The principal components are all uncorrelated and we get fewer vectors than the whole space that allow us to reconstruct any possible face.

1. The crowded problem of symmetric SNE

In symmetric SNE, we always use Gaussian to transform Euclidean distance into probability, no matter in high-dimensional or low-dimensional space. However, small probability can be achieved by using not-so-far distance in Gaussian. We may have two data that are far away in high-dimensional space but the corresponding data in low-dimensional space are not-so-far.

1. Anything you want to discuss
2. The value of gamma in RBF kernel is important. Inadequate value may lead to bad result.
3. The larger the perplexity is, the quicklier the pairwise similarities converge.
4. Symmetric SNE converges quicklier than t-SNE.
5. Perplexity has greater effect on t-SNE than symmetric SNE because symmetric SNE has crowded problem.