

1. PCA based data distribution visualizations

Let the data matrix \mathbf{X} be of $n \times p$ size.

First, perform SVD:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (1)$$

The columns of $\mathbf{U}\mathbf{S}$ are the principal components and the columns of \mathbf{V} are the eigenfaces.

For the principal components, only calculation for the highest dimension is necessary. For reduction to dimension k , the first k principal components can be obtained from the first k columns of \mathbf{U} and the diagonal matrix formed by the first k values along the diagonal of \mathbf{S} .

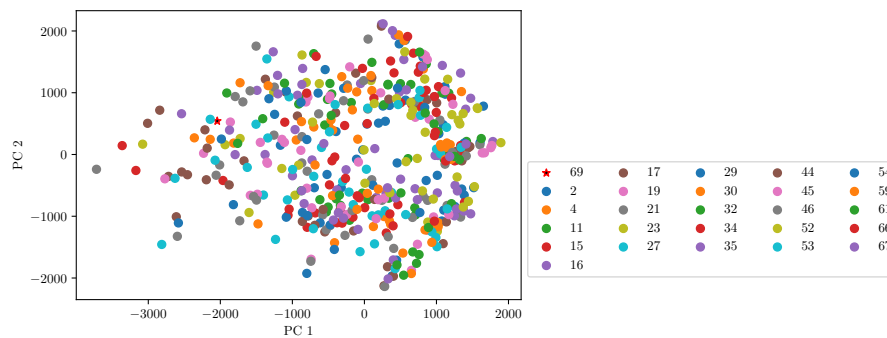


Figure 1: PCA projected data vector in 2D.

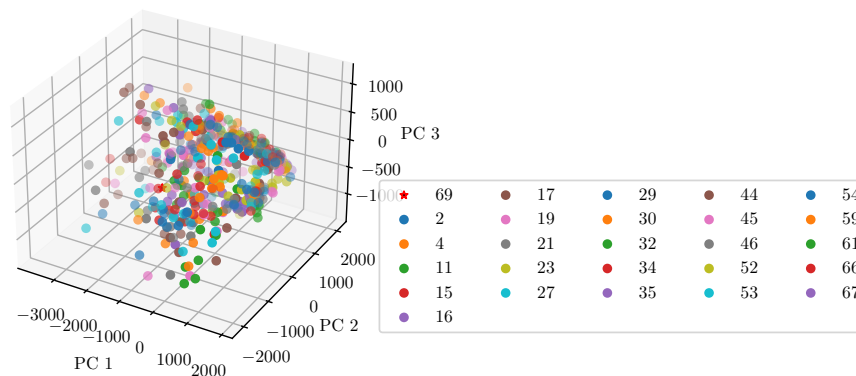


Figure 2: PCA projected data vector in 3D.

2. PCA plus nearest neighbor classification results

For classification with PCA, obtain \mathbf{XV} , projection of the faces in the test data matrix \mathbf{X} on the eigenface space. Each column of \mathbf{V} and therefore each column of \mathbf{XV} corresponds to a principal component. Again, only calculation for the highest dimension is necessary. The first k projections can be obtained with the test data matrix and the first k columns of \mathbf{V} .

It is expected that classification of selfies is unsuccessful. There are only three selfies in the test set and random sampling of 500 images for training included only one selfie which is an outlier in 3D as can be seen from the scatter plots. Moreover, my setup and lighting condition differs from those of the images in the PIE data set.

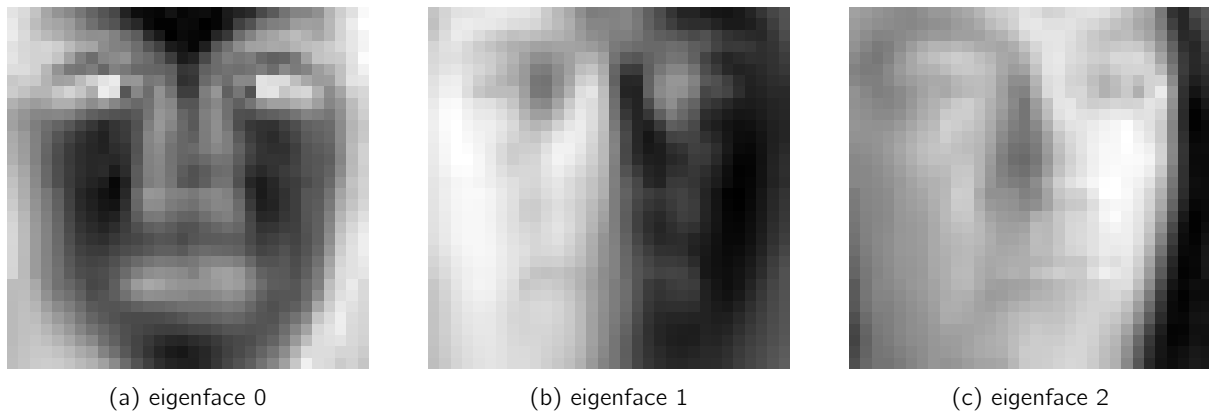


Figure 3: Corresponding 3 eigenfaces used for PCA dimensionality reduction.

Test Images	Dimensionality	Accuracy
CMU PIE	40	49.02 %
	80	52.47 %
	200	55.22 %
Selfies	40	0 %
	80	0 %
	200	0 %

3. LDA based data distribution visualization
4. LDA plus nearest neighbor classification results
5. SVM classification results with different parameter values
6. CNN classification results with different network architectures

For CNN, the accuracy ranges from 95.00% to 97.00 % with shuffling of the order of images.

Following the updated instructions received via email, my model includes two convolutional layers and two fully connected layers. The updated number of nodes is "20-50-500-26". There is an additional scaling layer at the start to scale the range of each pixel from the range 0–255 to 0–1. Although often recommended, scaling does not seem to improve the classification accuracy of this model.

For previous parts of this assignment, labels were taken directly from the folder names in the provided PIE dataset. However, for CNN, one hot encoding is necessary because there is no natural ordinal relationship for the order of PIE subjects. For classification of two or more labels using one-hot encoded representation, categorical crossentropy is the option in Keras. The final layer uses a softmax function to convert logits to probabilities. Otherwise, ReLU is used to introduce nonlinearity.

The latest optimizer offered by Keras, the AMSGrad variant of ADAM was used for best performance. As ADAM is an adaptive algorithm, no tuning of the learning rate was necessary or offered any improvement to classification accuracy.

Model: "sequential"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 32, 32, 1)	0
conv2d (Conv2D)	(None, 28, 28, 20)	520
max_pooling2d (MaxPooling2D)	(None, 14, 14, 20)	0
conv2d_1 (Conv2D)	(None, 10, 10, 50)	25050

max_pooling2d_1 (MaxPooling 2D)	(None, 5, 5, 50)	0
flatten (Flatten)	(None, 1250)	0
dense (Dense)	(None, 500)	625500
dense_1 (Dense)	(None, 26)	13026

```

=====
Total params: 664,096
Trainable params: 664,096
Non-trainable params: 0
-----

```

For simplicity, I arbitrarily set the number epochs to 100 with batch size of 128.

It takes about 55 epochs to reach 100 % accuracy on the training set and for the training loss to fall from an initial value of about 3.25 to 0.

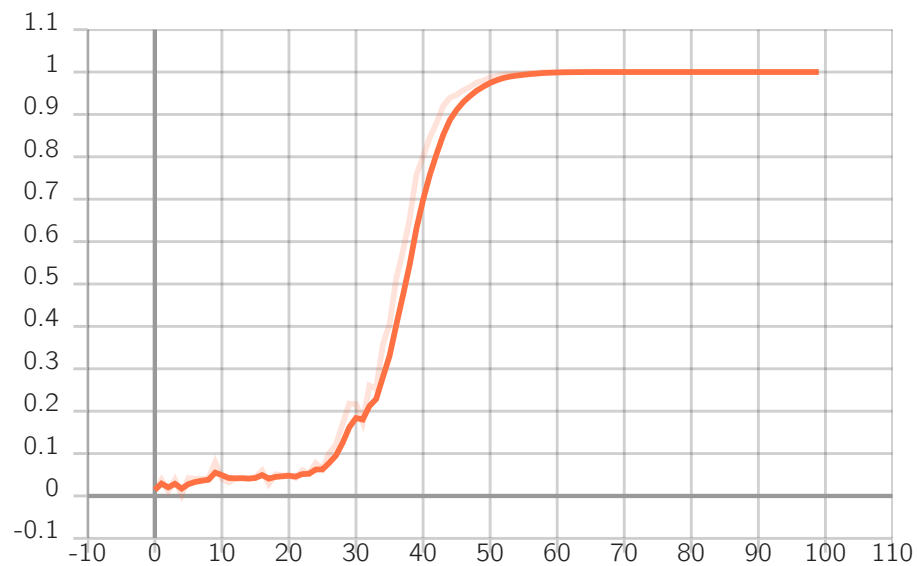


Figure 4: Epoch accuracy.

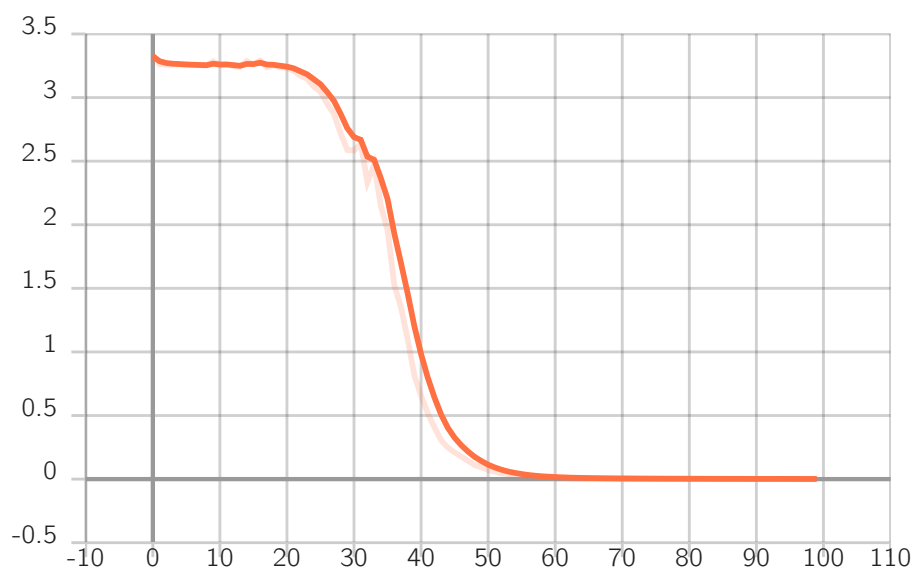


Figure 5: Epoch loss.