Quantitative Foundations - Project 4

Anurag Lamsal, Bhawana Prasain

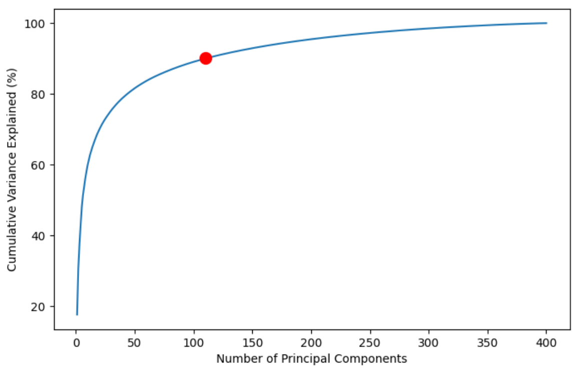
1 Eigenfaces: Visualization and Analysis

The given dataset consists of 400 images of human faces. Specifically, it has 10 images each of 40 human subjects. We calculate Principal Component Analysis (PCA) eigenvectors using these images (termed *eigenfaces*) and visualize them. We first flatten these images, mean-center them, and perform Singular Value Decomposition (SVD) on these mean-centered and flattened images (let's call this collection X) to get the eigenfaces and their corresponding eigenvalues. Note that we perform SVD directly on X instead of XX^T as the left singular vectors of X is the same as XX^T and its singular values are the square root of the eigenvalues of XX^T; the calculation is much faster this way.

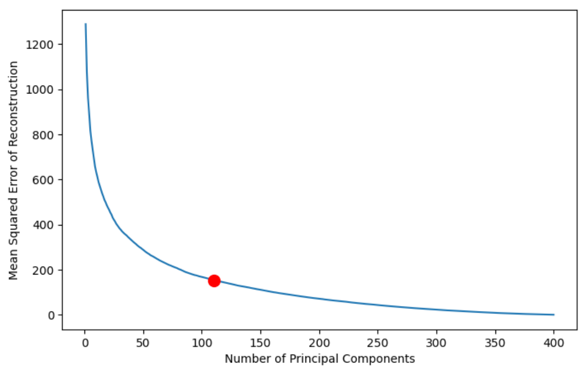
Figure 1: The first 10 eigenfaces.

1 visualizes the first 10 eigenfaces after resizing to 112×92 (which is the image shape for the given dataset). Not all of the eigenfaces are equally important, and we judge their importance by the value of variance in the axis defined by them, which is given by their corresponding eigenvalues.

For this dataset, we observe that using the first 111 eigenfaces are enough to explain 90% of variance in the given data, which is indicated by the red marker in 2a. This figure is a plot of *percentage of variance explained* against the *first i principal components used*. We see that there is a sharp increase in the variance explained until the curve starts plateauing. The start of the plateau is indication that the new eigenfaces being considered are not as important as the previous ones.



(a) Cumulative variance (in percentage) plotted against the number of principal components.



(b) Mean Squared Error (MSE) of reconstruction plotted against the number of principal components.

Figure 2: Plots for the analysis of variance explained (left) and reconstruction (right).

2 Face Reconstruction

To get faces back from representations, we calculate $P_k Z$, where P_k contains the first k principal components used for the representations, and Z is the matrix with the representations.

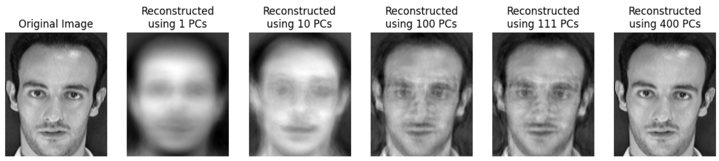


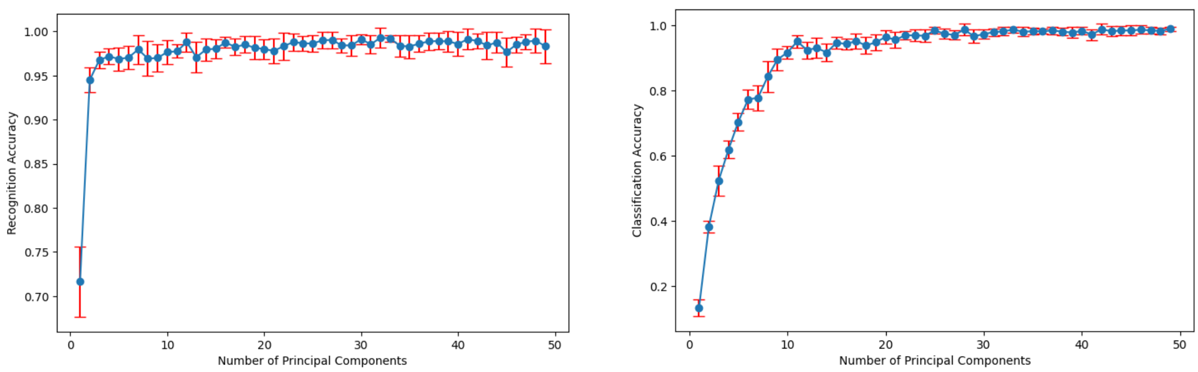
Figure 3: Reconstruction of faces from representation using the first 1, 10, 100, 111, and 400 principal components.

As we can see in 3, the reconstructions get more and more close to the original data as the number of principal components used increases. Reconstruction using the first 111 principal components (which explains 90% of variance) is already quite close to the original data barring some imperfections. We calculate Mean Squared Error (MSE) for reconstruction incorporating 80 randomly chosen images and plot it, in 2b, against the number of principal components used. We observe that this curve basically follows the same shape as 2a but flipped, which is intuitive.

3 Face Recognition/Classification via Linear Discriminant Analysis (LDA)

For recognition, along with randomly picking 8 images from randomly chosen 35 subjects, we also randomly pick 400 non-face images from an external image dataset. We combine the 35×8 face images with 70% of the non-face images to form the train set, and the remaining face and non-face images form the test set. We first calculate the PCA eigenvectors using the train set and find representations of both train and test sets using the first i principal components, where i goes from 1 to 50. We run 10 random-split trials for each i and calculate the average accuracy and standard deviation. As seen in 4a, the accuracy steadily increases until the first dip at 4 principal components, and the accuracy there is 0.972 with standard deviation of 0.008.

For classification, we randomly pick 8 images from each of the 40 subjects to form the train set, and the remaining form the test set. We follow the same pipeline as above for this (but with the number of classes being 40 instead of 2). As seen in 4b, the accuracy steadily increases until the first dip at 11 principal components, and the accuracy there is 0.95 with standard deviation of 0.020. One interesting observation here is that *classification requires more principal components for high accuracy as compared to detection/recognition.*



(a) Average recognition accuracy across 10 trials against the number of principal components.

(b) Average classification accuracy across 10 trials against the number of principal components.

Figure 4: Accuracy plots for recognition (left) and classification (right) while varying the number of principal components.

4 Classification with FaceNet

For potentially better results, we also try out a more modern pipeline for dimesionality reduction and classification.

4.1 Face Detection using MTCNN

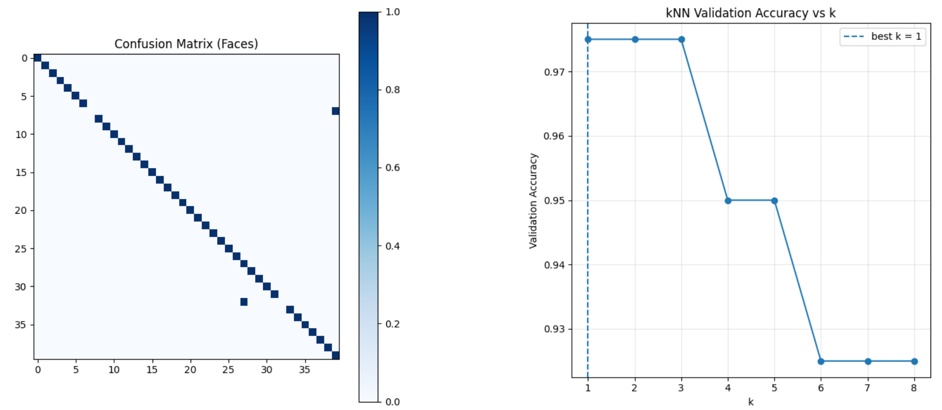
Before feature extraction, face detection is performed using the Multi-task Cascaded Convolutional Neural Network (MTCNN). This step ensures that only valid face regions are passed to the embedding network. Detection statistics indicate a detection rate of 100% on the ATT Faces dataset and a rejection rate of 100% on images that do not contain faces.

4.2 Face Embedding Extraction (FaceNet)

Detected face regions are resized to 160×160 pixels and passed through a pretrained FaceNet (Inception-ResnetV1) model. The network outputs a 512-dimensional embedding vector for each face image. Each embedding is ℓ_2-normalized to enable cosine similarity comparisons. Each embedding is an identity-preserving representation, where samples of the same subject are expected to cluster closely in feature space. No fine-tuning is performed; the network is used strictly as a fixed feature extractor.

4.3 k-Nearest Neighbor (kNN) Classification

A k-Nearest Neighbor (kNN) classifier is trained on the extracted FaceNet embeddings using cosine distance as the similarity metric. The value of k is selected using a validation set, with exactly one validation image per subject. Model selection is performed by evaluating validation accuracy for $k \in \{1, 2, \ldots, 8\}$.



(a) Confusion matrix for face classification on the test set.

(b) Validation accuracy vs. number of neighbors k (best at $k = 1$).

Figure 5: FaceNet + kNN evaluation: confusion matrix (left) and validation model selection (right).

5 Conclusion

The Eigenfaces method captures global variations in facial appearance and represents faces using a low-dimensional linear subspace. While it explains a large portion of the data variance with relatively few components, its ability to distinguish identities is limited due to its linear nature and sensitivity to factors such as lighting, pose, and expression.

In contrast, the FaceNet-based pipeline learns highly discriminative, nonlinear face embeddings using a deep convolutional network trained on large-scale data. Even without fine-tuning, the 512-dimensional embeddings cluster images of the same person closely while remaining well separated from other identities. This is reflected in the near-diagonal confusion matrix and high validation accuracy obtained with a simple k-Nearest Neighbor classifier, demonstrating better identity separability than the PCA-based approach.