Project 4: Eigenfaces

RIT

Anurag Lamsal, Bhawana Prasain

Principal Component Analysis (PCA): Brief Overview

- PCA involves finding new axes (called “principal components”) to represent our data in.

- These axes are defined by the orthogonal eigenvectors of (1 / N - 1) X @ X_T, where X is our mean-centered data, and N is the number of data points. The corresponding eigenvalues give the variance along the axes.

- Dimensionality Reduction is generally decided based on variance values along the axes, but it also depends on the downstream task.

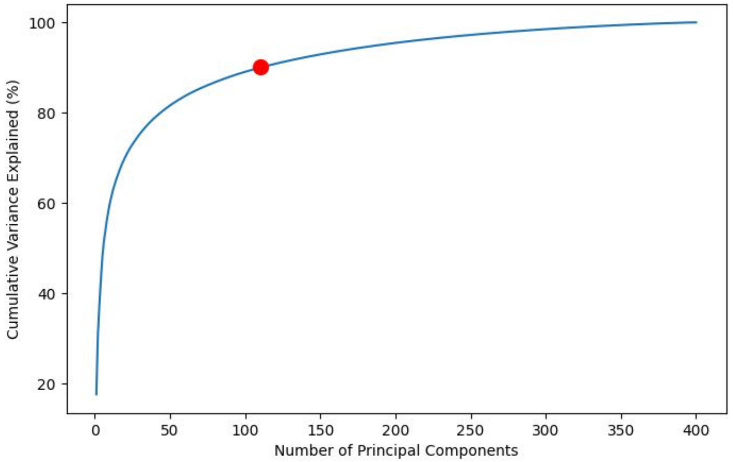
Talking about the project…

- **Dataset:** 400 images of human faces, 10 images each of 40 human subjects.
- We perform Singular Value Decomposition (SVD) on flattened and mean-centered images X to get the principal components (or *eigenfaces*) and their corresponding eigenvalues.
- Visualization of the first 10 eigenfaces after resizing to 112 x 92:

Importance of Eigenfaces

- **Variance Explained** = (Sum of Eigenvalues of first *i* principal components / Sum of all Eigenvalues) * 100

- We plot variance explained for all 400 principal components:

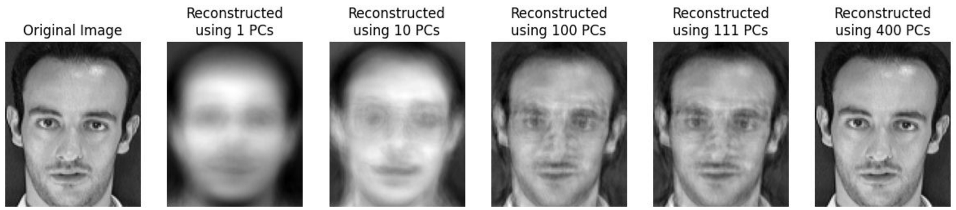


- We observe that the first 111 principal components are enough to explain 90% of variance.

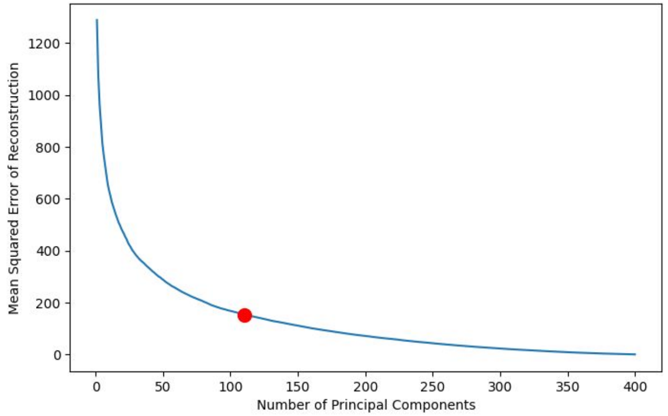
- 111 principal components may suffice for many tasks.

Reconstruction

- We calculate ((P_k @ Z) + mean) for reconstruction, P_k contains the first k principal components used for representation, and Z contains the representations.
- Example of reconstruction using 1, 10, 100, 111, and 400 principal components:



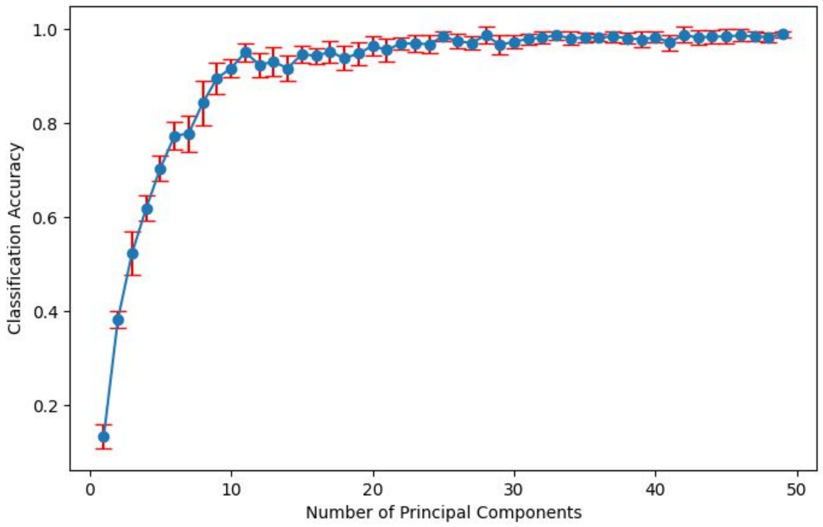
- We also plot MSE for reconstruction vs number of principal components for 80 randomly picked images:

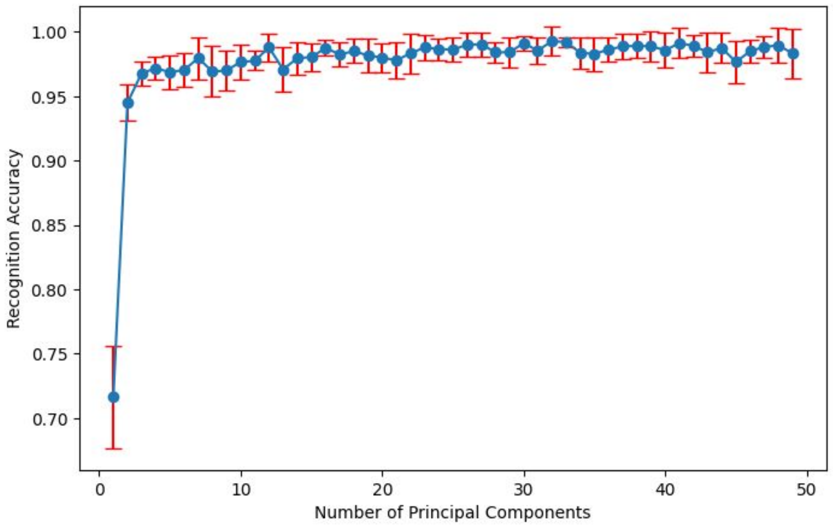


Recognition/Classification via LDA

- **Split for recognition**: Random 35 * 8 face images + 70% of randomly chosen 400 non-face images = Train Set. Rest is the test set.
- **Split for classification**: Random 40 * 8 face images for train set. Rest is the test set.
- We first do SVD on the train set, find PCA representations of train and test sets, and use LDA on these representations for recognition/classification.
- We plot the average accuracy (across 10 random-split trials) against the number of principal components used (from 1 to 50) for both recognition and classification.

Recognition/Classification Plots





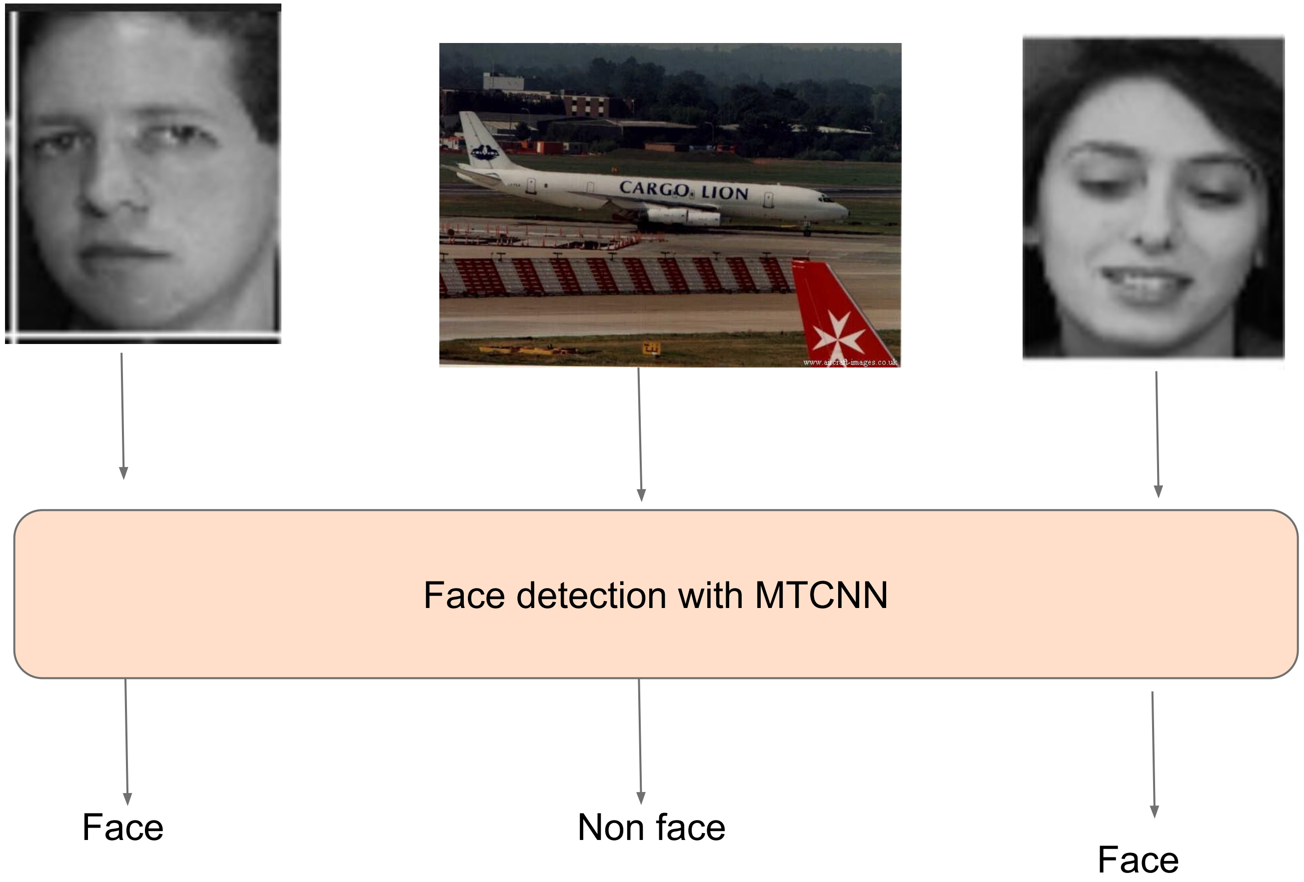
- **Accuracy using 4 principal components for recognition**: 0.972 with SD of 0.009.
- **Accuracy using 11 principal components for classification**: 0.950 with SD of 0.020.
- **Observation**: Classification requires higher number of principal components to get high accuracy as compared to recognition.

Classification with FaceNet

The pipeline consists of three main stages:

1. Face detection using MTCNN
2. Embedding extraction with FaceNet
3. Classification

Face Detection using MTCNN



MTCNN is a multi-stage convolutional neural network that detects faces.
Robust to pose and scale changes

FaceNet for Embedding

- We use a pretrained FaceNet model (Inception Resnet V1 trained on VGGFace2)

- FaceNet was trained to make the squared distance between two image embeddings of the same identities small, whereas the squared distance between two images of different identities is large

- Output is 512-dimensional vectors and L2 normalized
- Distance reflects identity similarity

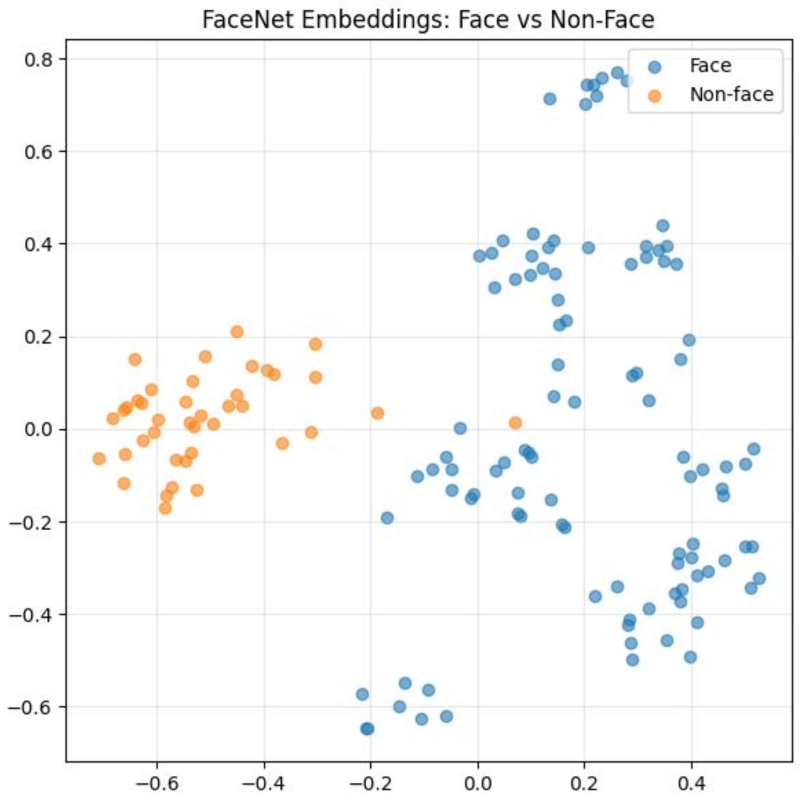


Fig: Cluster of Face vs non face embeddings from FaceNet

Classification with FaceNet

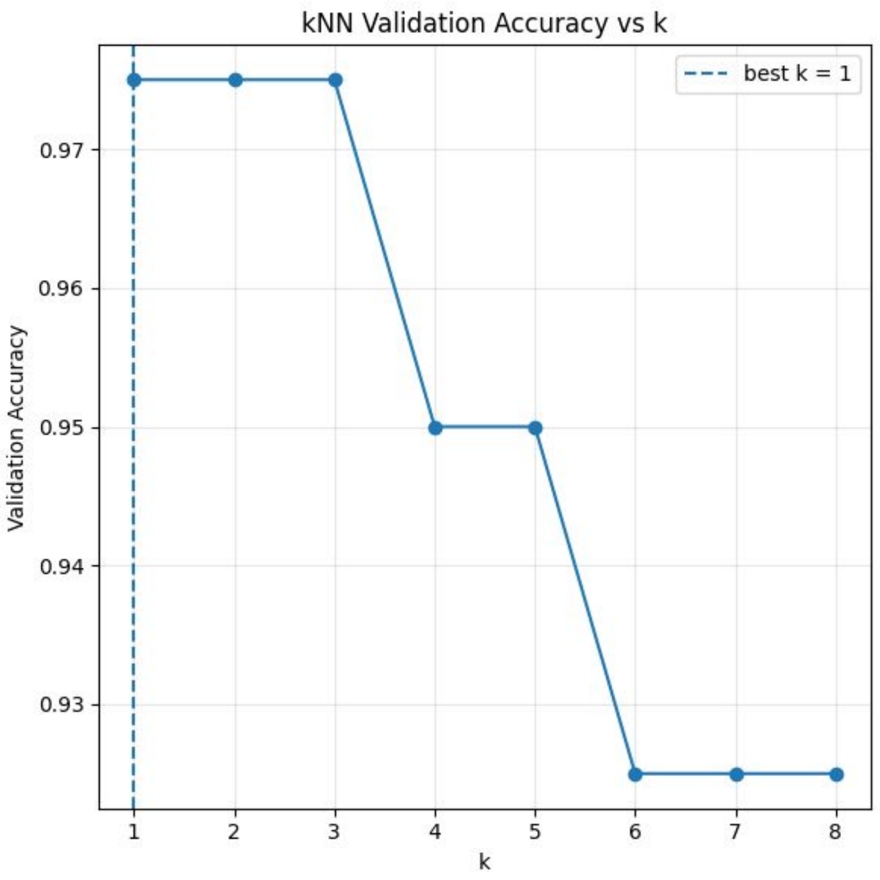


Fig: Validation accuracy on different values of k for KNN

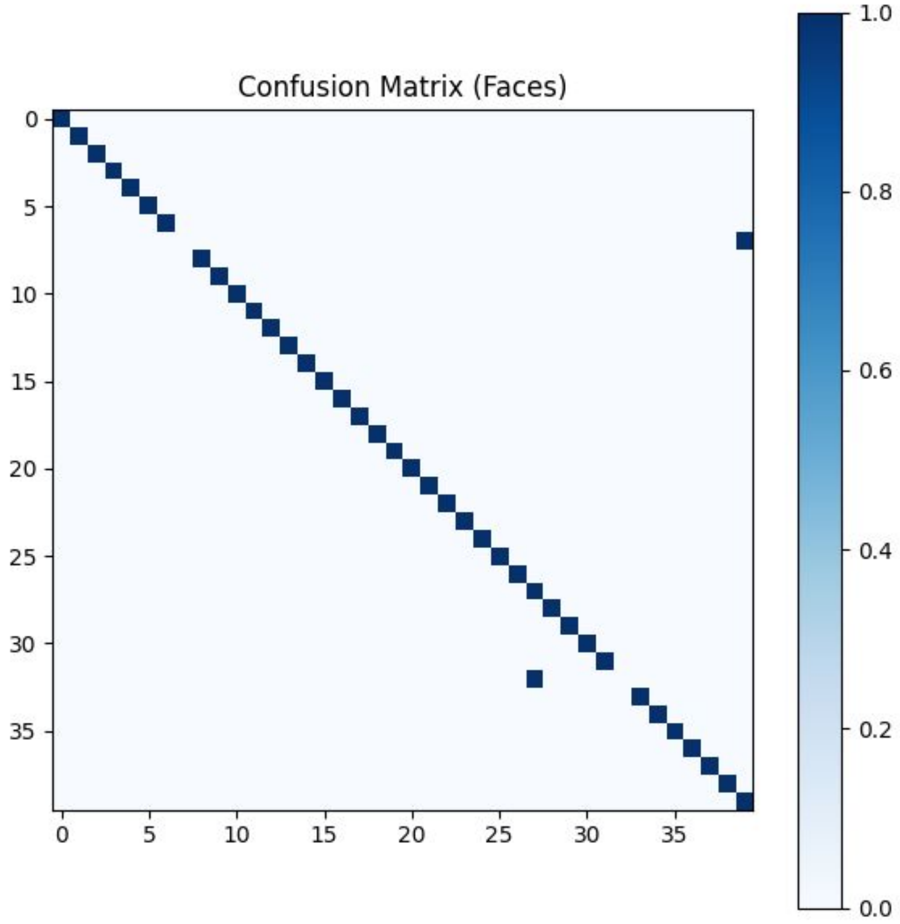


Fig: Confusion matrix on face images

Comparison and conclusion

Eigenfaces (PCA-based Face Representation)

- Captures global facial variations using a linear subspace
- Represents faces with few principal components
- Explains a large portion of data variance efficiently
- Limited identity discrimination
- Sensitive to:
  - Illumination changes
  - Pose variations
  - Facial expressions

Effective for dimensionality reduction, but not robust for identity recognition.

Deep Learning–Based Face Embeddings

- Learns nonlinear, discriminative embeddings
- Trained on large-scale face datasets
- Produces 512-dimensional embeddings
- Same-identity images:
  - Cluster tightly
- Different identities:
  - Well separated in embedding space

Embeddings are identity-focused, not just variance-focused.

Thank you for your attention!

Q&A