Eigenfaces for Face Recognition

Sucheta Jhunjhunwala, MDS202151
Ujjwal Saxena, MDS202152
Ved Prakash, MDS202153

Instructor: Professor Kavita Sutar
Chennai Mathematical Institute

May 6, 2022

Introduction

Face recognition

- The face plays a major role in conveying the identity of a person.
- We can recognize familiar faces even after changes in the visual stimulus.
- Because of this, not only computer science researchers are interested in it, but neuroscientists and psychologists are also interested.

- We try to extract information about the face by looking at the variation in the collection of face images.
- An eigenface is the name given to a set of eigenvectors.
- Our target is to classify an image as a face image or a non-face image. Further, if it is a face image then we have to check whether it is a known image or not.
- Eigenfaces is probably one of the simplest face recognition method.

History of Face Recognition

- Initially, face recognition done by comparing the position and size of the individual features.
- Then there was a shift into the semi-automated system which used fiducial markers.
- Kaya & Kobayashi and others approached automated facial recognition by characterizing a face by a set of geometric parameters.
- To understand the essential global and local features of a face, the information theory approach of coding and decoding face images started being implemented.

Database

We are working with the "Yalefaces Database". This database has a collection of 165 grayscale images, 11 images of 15 individuals. The images of each individual vary in facial expression or have different lighting conditions.

Figure: Yaleface Database

Calculating the Eigenfaces

- We have a set of M face images $\Gamma_1, \Gamma_2, ...\Gamma_M$, each of which can be represented as an array of size N * N. Each of the array representation of a face image can be flattened to form a vector of dimension N^2.
- Calculate the deviation of each face image from it's mean. The mean of the set of face images is given by,

$$\psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

Let, $\phi_i = \Gamma_i - \psi$ for i in 1,2,...M.

- Define a matrix A $= [\phi_1\ \phi_2\ \phi_3\ .....\phi_M]$. Note that, A is a N^2*M matrix, where each column represents the mean deviation of each face image.

Figure: Deviation of the images from the mean

- We need to find the Covariance Matrix, i.e., C, of the set of face images. Note that, C is an $N^2 * N^2$ matrix where each element represents the covariance between two features.
  Let, $X_i = \{\Gamma_{1i}, \Gamma_{2i}, ..., \Gamma_{Mi}\}$ for i in 1,2,... N^2.

$$C = \begin{bmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) & . \quad . \quad . & Cov(X_1, X_{N^2}) \\ Cov(X_2, X_1) & Cov(X_2, X_2) & . \quad . \quad . & Cov(X_2, X_{N^2}) \\ . & & & \\ . & & & \\ . & & & \\ Cov(X_{N^2}, X_1) & Cov(X_{N^2}, X_2) & . \quad . \quad . & Cov(X_{N^2}, X_{N^2}) \end{bmatrix}$$

- Cov$(X_i, X_j) = \dfrac{\sum_{k=1}^{M}(\Gamma_{ki} - \psi_i)(\Gamma_{kj} - \psi_j)}{M}$
- On simplifying we get, C $= \dfrac{\sum_{k=1}^{M} \phi_k \phi_k^T}{M}$
  $= AA^T$

- If μ_i is an eigenvalue of $A^T A$ with eigenvector as v_i then we have, $A^T A v_i = \mu_i v_i$.
  $\iff \mathrm{A}(\mathrm{A}^T A v_i) = A(\mu_i v_i)$
  $\iff (\mathrm{AA}^T) A v_i = \mu_i (A v_i)$
  Hence, v_i is an eigenvector of $A^T A$ iff $A v_i$ is an eigenvector of AA^T
- A is an $N^2 * M$ matrix, so AA^T is $N^2 * N^2$. Instead we look at the M*M matrix, $A^T A$.
- $u_i = \mathrm{A} v_i$ form the eigenvectors of the covariance matrix of the set of images with μ_i as the eigenvalues.
  Hence, the eigenfaces are $u_i = \sum_{k=1}^{M} v_{ik} \phi_k$, for i in 1,2,...M.
- The eigenvalues obtained are ordered and we consider the top M' many eigenvalues and the corresponding eigenvectors. These M's many eigenvectors are our eigenfaces on which we shall project our images.

- In our dataset we have images of 15 individuals, we assume M'$=$ 6.

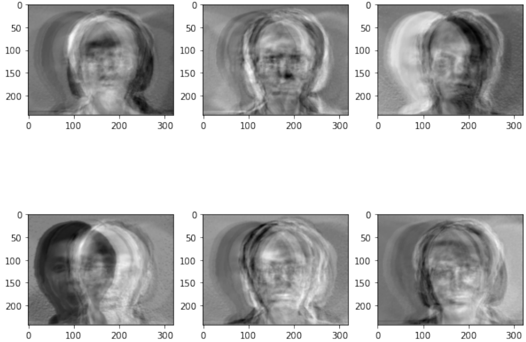

Figure: Top 6 Eigenfaces

- Now if M'$=$ 12.

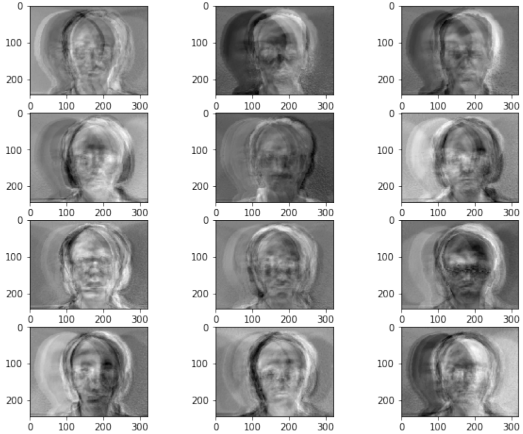

Figure: Top 12 Eigenfaces

Projecting an Image

Once we have calculated the eigenfaces that span our eigenspace, known as the face space, we shall project any image Γ to the face space.

- We have, $\omega_i = u_i^T(\Gamma - \psi)$ for i = 1,2,3.... M',
  as the transformation of our new image into the eigenface components.
- ω_i is basically the set of weights along each eigenface. We get, $\Omega^T = [\omega_1 \, \omega_2 \, ...\omega_M]$. Ω describes the contribution of each eigenface in representing the face image, treating the eigenfaces as a basis set for face images.

Figure: Projected Face Images

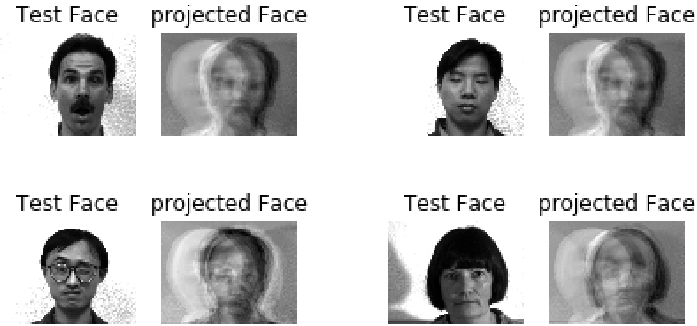

Figure: Test vs Projected Face Images

Classifying an Image

- Calculate the two distances, ϵ_k and ϵ.
- $\epsilon_k^2 = ||(\Omega - \Omega_k)||^2$ where Ω_i represents a k^{th} face class. The k^{th} face class is defined to be the average of the eigenface representation of the face images of the k^{th} individual.
- $\epsilon^2 = ||\phi - \phi_f||^2$,where $\phi = \Gamma - \psi$, i.e., it is the mean adjusted input image and $\phi_f = \sum_{i=1}^{M'} \omega_i u_i$, i.e., the projection of the face image.
- Set a threshold for both ϵ_k and ϵ, call it θ_ϵ.

Based on the values of ϵ_k, ϵ and θ_ϵ, there can be four possibilities:

- $\epsilon < \theta_\epsilon$ and $\epsilon_k < \theta_\epsilon$: The input image is the face image of a known individual.
- $\epsilon < \theta_\epsilon$ and $\epsilon_k > \theta_\epsilon$: The input image is the face image of an unknown individual.
- $\epsilon > \theta_\epsilon$ and $\epsilon_k < \theta_\epsilon$: The input image is not a face image, but is the image of a known individual, this is a case of False Positive.
- $\epsilon > \theta_\epsilon$ and $\epsilon_k > \theta_\epsilon$: The input image is not a face image.

Other Features of the Face Images

- While classifying an image we could have taken into consideration the following points as well:
- **Eliminating the Background** : We ignored the affect of the background. To deal with this we can multiply the input image by a 2-D gaussian window centred on the face.
- **Head size**: The motion analysis gives an estimate of head size, from which the face image is rescaled to the eigenface size.

- **Head Orientation**: The eigenfaces approach is not extremely sensitive to head orientation.
- **Multiple Views**: We can also extend the system to deal with full frontal views by defining a limited number of face classes for each known person corresponding to characteristic views.

Conclusion

- The early attempts at making computers recognize faces were limited by the use of impoverished face models and feature descriptions.
- The eigenface approach to face recognition was motivated by information theory.
- This approach does not provides an elegant solution to the general recognition problem.
- This approach does provide a practical solution.
- It can also be implemented using modules of connectionist or neural networks.

Experimental Results

We have implemented Principal Component Analysis on the space of the face images. We took different values of M' to check which reduces the error.

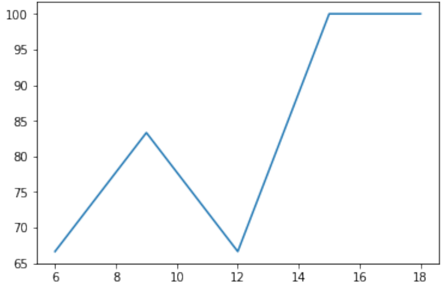

Figure: M' against the number of True Positives

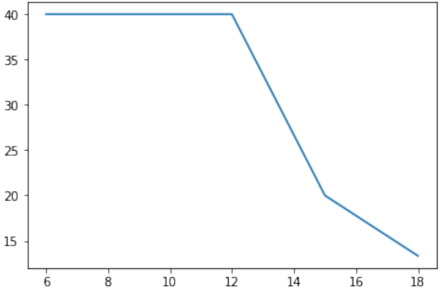

Figure: M' against the number of Face Images Classified

References

- https://en.wikipedia.org/wiki/Eigenface#: :text=An%20eigenface %20(%2F%CB%88a%C9%AA%C9%A1,Alex%20Pentland%20in%20 face%20classification.
- https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition
- http://vision.ucsd.edu/content/yale-face-database