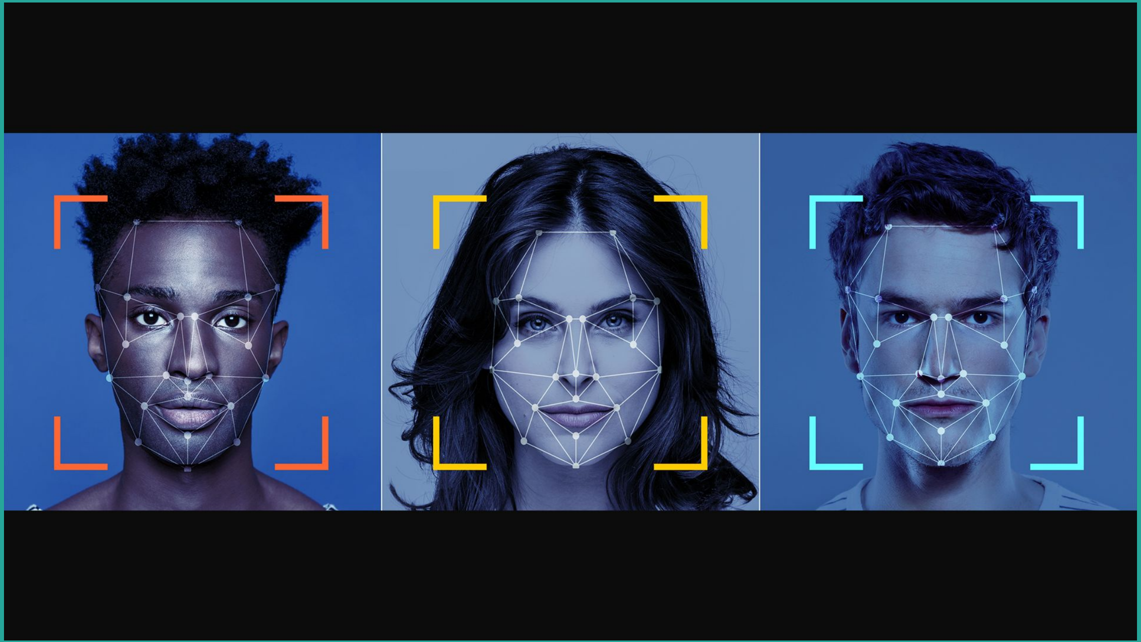
Face Recognition using PCA

By Dmytro Lutchyn, Nazarii Kuspys

Introduction



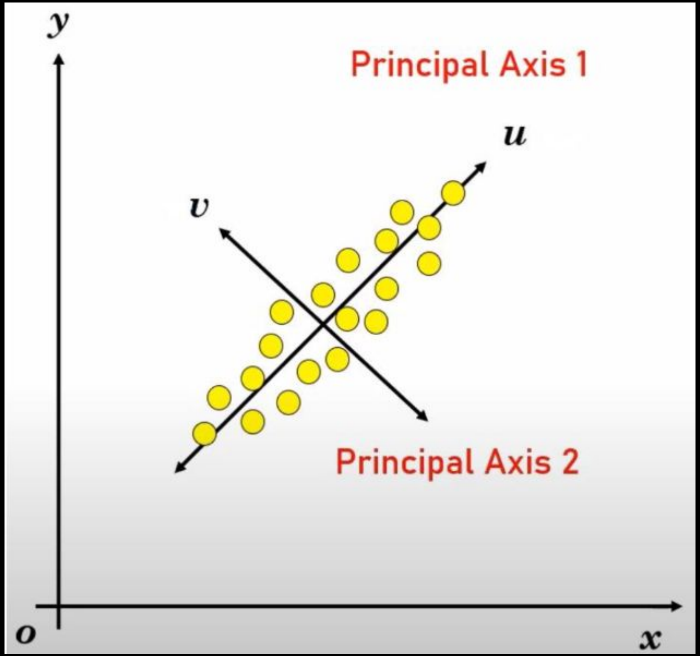
The idea behind

Principal component analysis

for face recognition

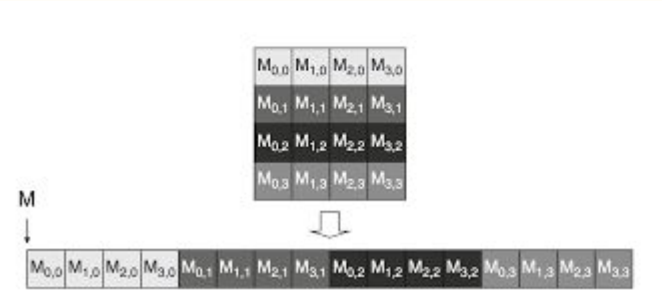
Main purpose

of principal component analysis



Implementation

- Grayscale image
- Represent images as 1D vectors
- Normalize images
- Stack to create a matrix



$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots\cdots & x_{1MN} \\ x_{21} & x_{22} & x_{23} & \cdots\cdots & x_{2MN} \\ x_{31} & x_{32} & x_{33} & \cdots\cdots & x_{3MN} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots\cdots & x_{nMN} \end{bmatrix}$$

Implementation

- Create covariance matrix
- Diagonalize covariance matrix
- Project set of images to new axis system

$$Q = \left(\frac{X_m^T X_m}{n-1}\right)$$

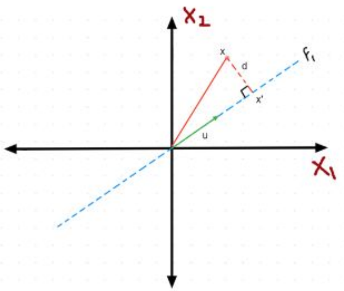
$$\boldsymbol{\sigma}^2(\mathbf{b}) = \begin{bmatrix} \sigma^2(b_0) & \sigma(b_0, b_1) \\ \sigma(b_1, b_0) & \sigma^2(b_1) \end{bmatrix}$$

$$P = [P_1\ P_2\ P_3\ \ldots\ldots\ldots P_{MN}] = \begin{bmatrix} P_{11} & P_{21} & P_{31} & & P_{MN1} \\ P_{12} & P_{22} & P_{32} & & P_{MN2} \\ \vdots & \vdots & \vdots & \ldots\ldots\ldots & \vdots \\ P_{1MN} & P_{2MN} & P_{3MN} & & P_{MNMN} \end{bmatrix}$$

Details

S - covariance Matrix from a previous slide,
u - our required principal component

- Minimize the projection error of each point
- Solve modified optimization problem with Lagrange's Multiplier



$$d^2 = \| x_i \|^2 - (u^T x_i)^2$$
$$= (x_i^T x_i) - (u^T x_i)^2$$

So our optimization function becomes:

$$\min_u \sum_{i=1}^{n} (x_i^T x_i) - (u^T x_i)^2$$

subject to $\| u \| = 1$

Our optimization problem was to find a direction u which

$$\max_u \frac{1}{n} \sum_{i=1}^{n} (u^T x_i)^2 , \ \textit{subject to} \ \| u \| = 1.$$

$$\frac{1}{n} \sum_{i=1}^{n} (u^T x_i)^2 = u^T \frac{(X^T X)}{n} u = u^T S u.$$

Let lambda be our lagrange's multiplier.

So, our Lagrange function becomes:

$$L(u, \lambda) = u^T S u - \lambda(u^t u - 1)$$

Partially differentiating wrt u, we get:

$$\frac{\partial L(u, \lambda)}{\partial u} = 2Su - 2\lambda u$$

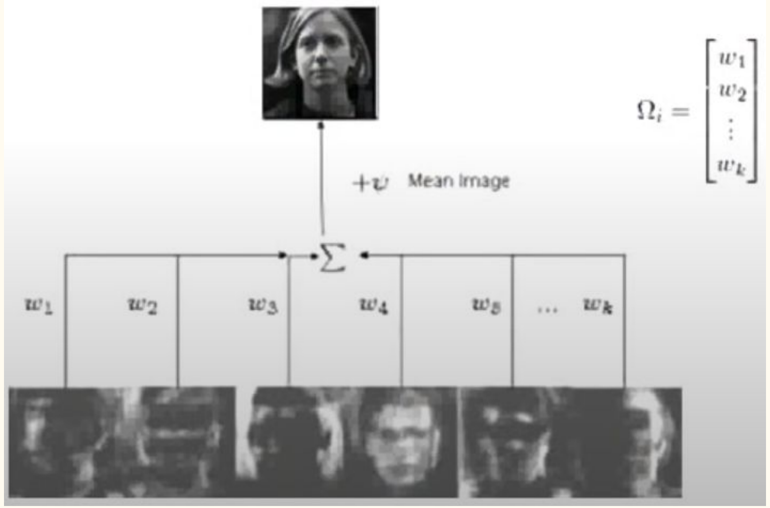
Equating the derivative to 0 and solving for u we get:

$$\boldsymbol{Su = \lambda u}$$

Implementation

Image representation as linear combination of eigenfaces

- Project input image onto new principal axis system
- Find the best fit (identify the person)

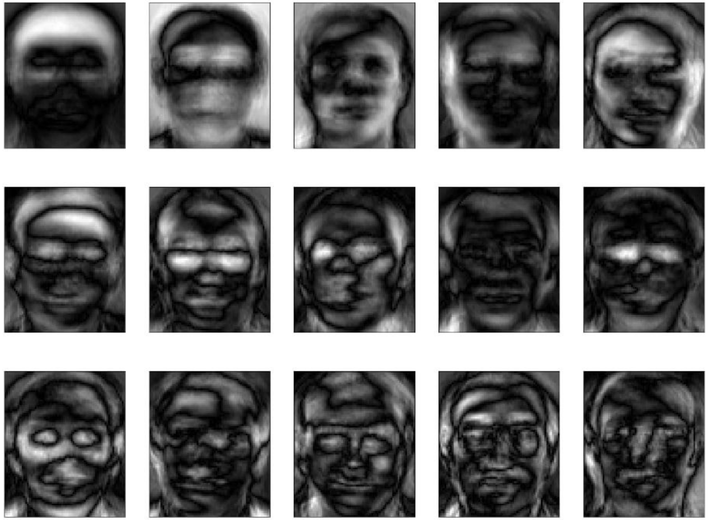


$$T_{n\times L} = [X - m]_{n\times MN} \cdot [P_{PCA}]_{MN\times L}$$

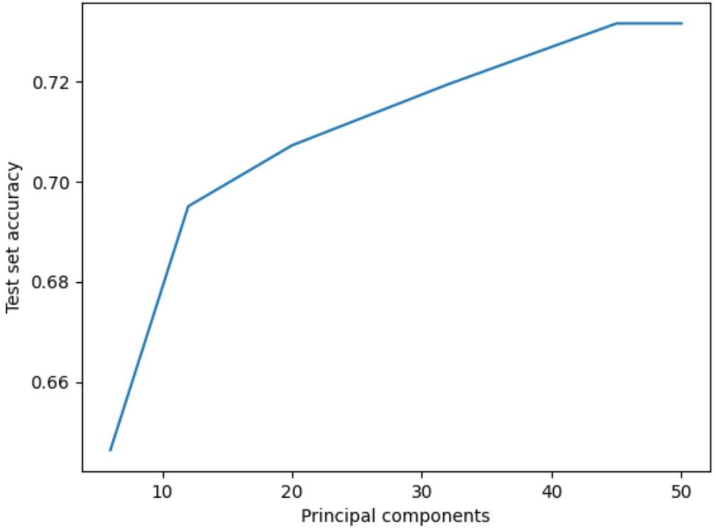
$$I_{PCA(1\times L)} = [I - m]_{(1\times MN)} \cdot [P_{PCA}]_{(MN\times L)}$$

$$L(Img, Iw) = \sum_{i=1}^{k} |Img_i - Iw_i|$$

Results

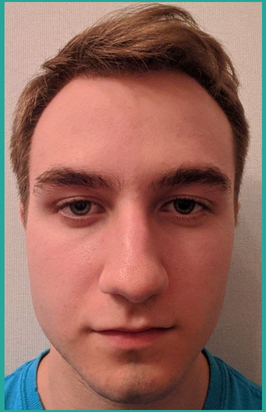


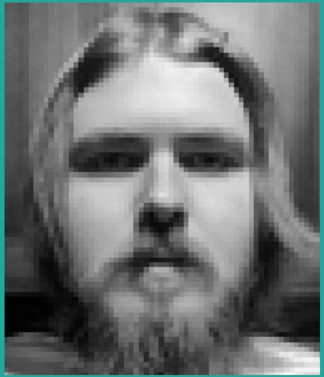
Eigenfaces



Increase of accuracy with number of pc





Thank you
for attention