PHÉP BIẾN ĐỔI KL VÀ PHƯƠNG PHÁP PCA TRONG NHẬN DIỆN KHUÔN MẶT

Nhóm 3 : Bùi Văn Tài

Vũ Ngọc Thịnh

Phạm Bình Thiên Triều

MỤC LỤC

MỤC TIÊU

- Tìm hiểu về phép biến đổi karhunen – loeve trong xử lý ảnh
- Phương pháp PCA để giảm chiều dữ liệu
- Ứng dụng của PCA trong nhận dạng khuôn mặt

PHÉP BIẾN ĐỔI KARHUNEN - LOEVE

- **+Trị riêng và vector riêng**
  - Cho một ma trận vuông A ∈ R^{nxn} . Nếu **Ax = λx**, thì ta nói λ và x là một cặp trị riêng, vector riêng (eigenvalue, eigenvector) của ma trận A.
- **+Kỳ vọng và ma trận hiệp phương sai**
  - Cho N điểm dữ liệu được biểu diễn bởi các vector cột x_1 , x_2 , …. , x_N . Khi đó, vector kỳ vọng và ma trận hiệp phương sai của toàn bộ dữ liệu được định nghĩa là:
  - $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$
  - **S**=$\frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T = \frac{1}{N}\hat{X}\hat{X}^T$

PHÉP BIẾN ĐỔI KARHUNEN - LOEVE

- Ta xét một biểu diễn vector x bất kỳ :

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- Các giá trị x_i có thể là giá trị độ xám của ảnh

PHÉP BIẾN ĐỔI KARHUNEN - LOEVE

- Ta có kỳ vọng của biểu diễn trên là

$$\underline{m}_x = E\{\underline{x}\} = [m_1 \quad m_2 \quad \ldots \quad m_n]^T = [E\{x_1\} \quad E\{x_2\} \quad \ldots \quad E\{x_n\}]^T$$

- Ma trân hiêp phương sai sẽ được xác đinh :

$$C = E\left\{(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^T\right\}$$

PHÉP BIẾN ĐỔI KARHUNEN - LOEVE

- Ma trận A có các hàng là các vector riêng (eigenvector) của ma trận hiệp phương sai trên
- Chúng ta sẽ sắp xếp sao cho hàng đầu tiên là vector riêng ứng với trị riêng lớn nhất . Tiếp theo đó giảm dần theo giá trị của eigenvalue
- Ta sẽ có một biến đổi :

$$\underline{y} = A\left(\underline{x} - \underline{m}_x\right)$$

- Gọi là **biến đổi karhunen – loeve (KLT)**

PHÉP BIẾN ĐỔI KARHUNEN - LOEVE

- Ta sẽ dễ dàng chứng minh được :

$$E\{\underline{y}\} = 0$$

$$C_y = AC_xA^T$$

$$C_y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

PHÉP BIẾN ĐỔI KARHUNEN - LOEVE

- Ta sẽ biến đổi ngươc lại tín hiệu ban đầu $\underline{x}$ từ tín hiệu $\underline{y}$:

$$A^{-1} = A^T$$
$$\underline{x} = A^T \underline{y} + \underline{m}_x$$

- Ta xác định một ma trận A_K từ K eigenvectors ứng với K eigenvalue giá trị lớn nhất . ma trận A_K sẽ có chiều là Kxn.
- $\underline{y}$ cũng là vector có số chiều là K
- Vậy vector $\hat{x}$ khôi phục lại có dạng :

$$\underline{\hat{x}} = A_K^T \underline{y} + \underline{m}_x$$

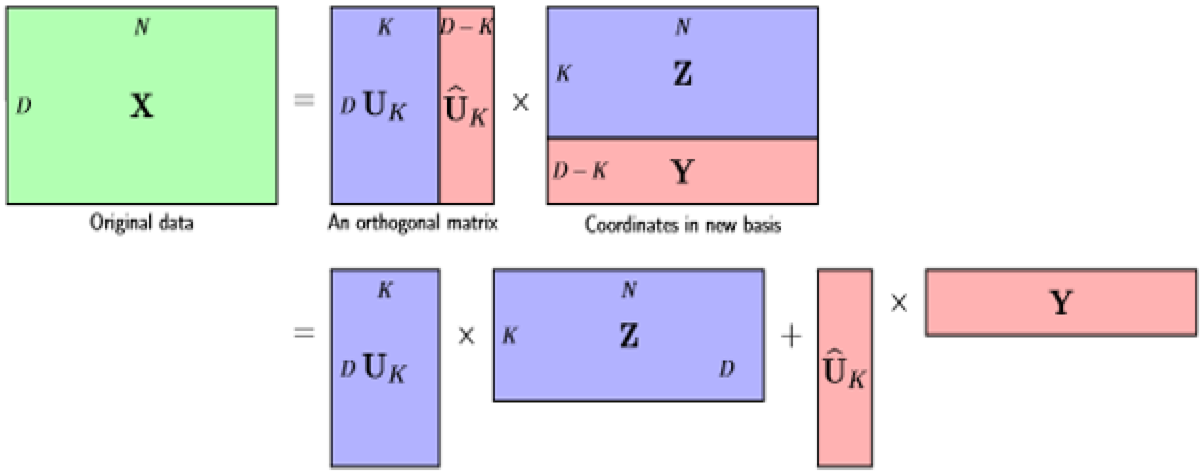
PHÉP BIẾN ĐỔI KARHUNEN - LOEVE

- Mean squared error của phép biến đổi KL :

$$e_{\mathrm{ms}} = \|\underline{x} - \hat{\underline{x}}\|^2 = \sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{K} \lambda_j = \sum_{j=K+1}^{n} \lambda_j$$

- Vậy ta thấy khi sử dụng ma trận A_K thay ma trận A phép biến đổi KL đã giúp ta giảm chiều dữ liệu.

PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT



Nguồn : http://machinelearningcoban.com

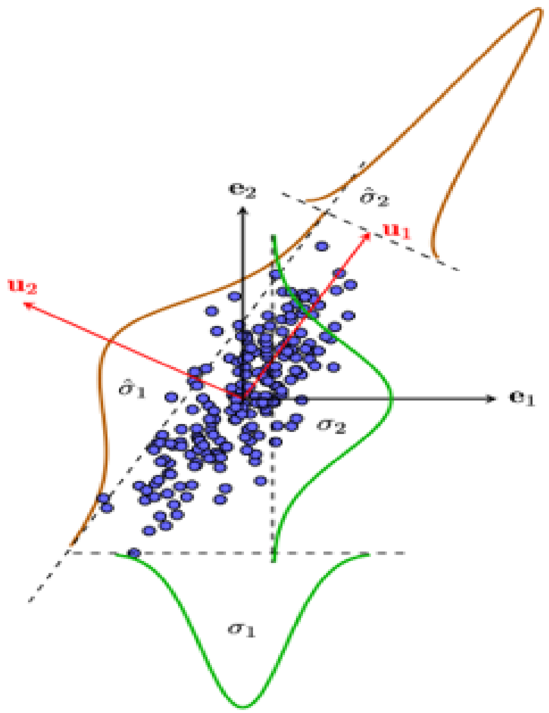
PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT

- Phương pháp PCA (phân tích thành phần chính) cũng dựa trên phép biến đổi KL để giữ lại K thành phần chính của dữ liệu , loại bỏ các thành phần không quan trong.
- Vấn đề đặt ra là xác đinh tầm quan trọng của một chiều dữ liệu như thế nào ?
- Nếu các chiều có cùng tầm quan trong thì phải làm thế nào ?

PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHÂN DIỆN KHUÔN MẶT

- **X =** $\boldsymbol{U_K Z} + \boldsymbol{\widehat{U}_K . Y}$
- $\begin{bmatrix} \boldsymbol{Z} \\ \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U_K}^{\boldsymbol{T}} \\ \boldsymbol{\widehat{U}_K}^{\boldsymbol{T}} \end{bmatrix} . \boldsymbol{X}$
- Ta muốn giữ lại phần lớn thông tin ở phần $\boldsymbol{U_K Z}$ và loại bỏ phần thông tin ở trong $\boldsymbol{\widehat{U}_K . Y}$

PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT



PCA thực chất là đi tìm một phép xoay tương ứng với một ma trận trực giao sao cho trong hệ toạ độ mới, tồn tại các chiều có phương sai nhỏ mà ta có thể bỏ qua; ta chỉ cần giữ lại các chiều/thành phần khác quan trọng hơn. Vì vậy, PCA còn được coi là phương pháp giảm số chiều dữ liệu sao tổng phương sai còn lại là lớn nhất

Nguồn : http://machinelearningcoban.com

PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT

- Ta sẽ có các bước thực hiện PCA như sau:
- 1.Tính vector kỳ vọng của toàn bộ dữ liệu : $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$
- 2. Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu để được dữ liệu chuẩn hoá: $\hat{x}_n = x_n - \bar{x}$
- 3. Đặt $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_D]$ là ma trận dữ liệu chuẩn hóa , tính ma trân hiệp phương sai : $\mathbf{S}=\frac{1}{N}\hat{X}\hat{X}^T$
- 4. Tính các trị riêng và vector riêng tương ứng của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng

PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHÂN DIỆN KHUÔN MẶT

- 5. Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trận U_K có các cột tạo thành một hệ trực giao. K vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hoá.
- 6. Chiếu dữ liệu ban đầu đã được chuẩn hóa $\hat{X}$ xuống không gian con đó.
- 7.Dữ liệu mới chính là tọa độ các điểm dữ liệu trong không gian mới :

$$\mathbf{Z}=\boldsymbol{U_K}^{\boldsymbol{T}}.\boldsymbol{\hat{X}}$$

- Dữ liệu ban đầu có thể tính xấp xỉ theo dữ liệu mới : x≈ $\boldsymbol{U_K}.\boldsymbol{Z} + \bar{x}$
- Một điểm dữ liệu mới **v** $\in R^D$ (có thể không nằm trong tập huấn luyện) sẽ được giảm chiều bằng PCA theo công thức w = $U_K{}^T$(v−x) $\in R^K$. Ngược lại, nếu biết w, ta có thể xấp xỉ **v** bởi **w.$\boldsymbol{U_K}$ + x**.

PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHÂN DIỆN KHUÔN MẶT

- Sử dung SVD (Singular value Decomposition)
- Một ma trận bất kỳ đều có thể phân tích thành dạng :

$$\boldsymbol{A_{mxn}}=\boldsymbol{U_{mxm}\Sigma_{mxn}V^{T}{}_{nxn}}$$

- Trong đó, U,V là các ma trận trực giao, Σ là một ma trận đường chéo cùng kích thước với A.
- Ta có ma trận $\boldsymbol{AA^T} = (U.\Sigma.V^T).).(\mathrm{U}.\Sigma.V^T)^T$

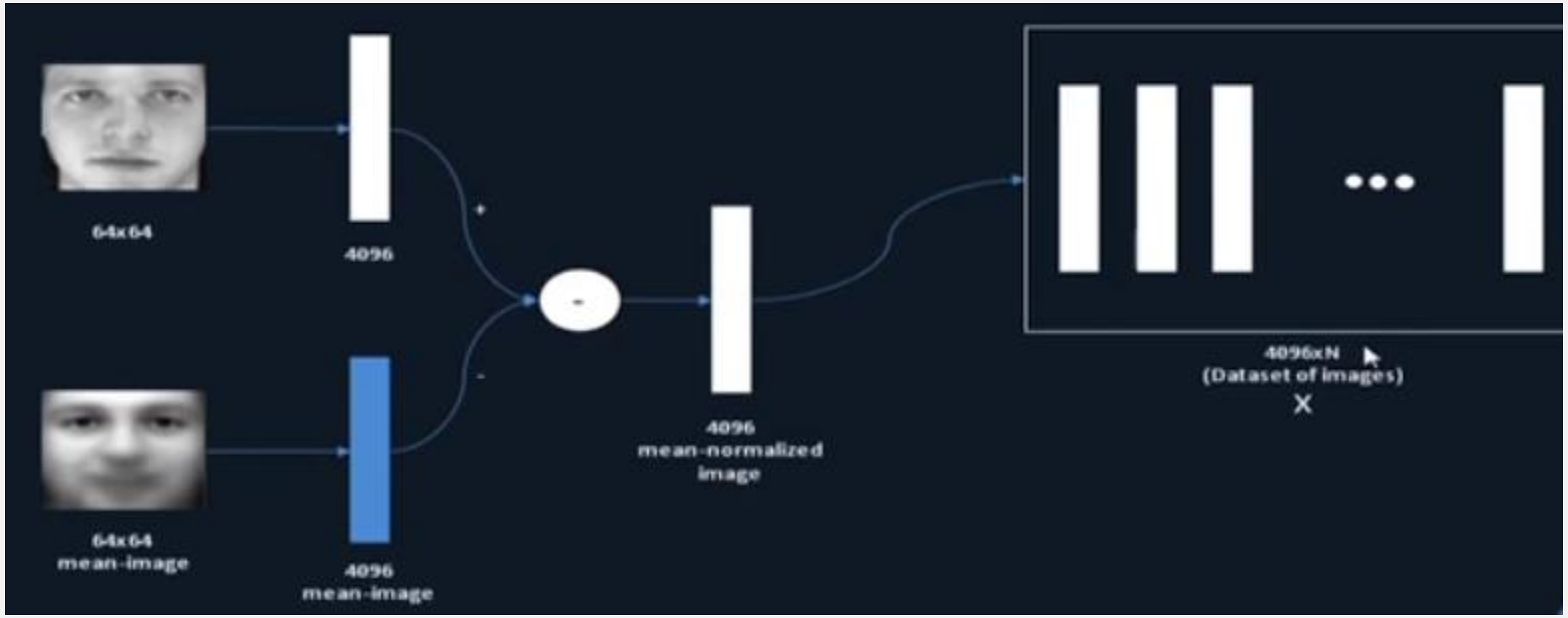
$$= \mathrm{U}.\Sigma.V^T.V.\Sigma^T.U^T$$

$$=U.\Sigma.\Sigma^T.U^T$$

- Ta nhận thấy $\Sigma.\Sigma^T$ chính là ma trận đường chéo với các giá trị là trị riêng của $\boldsymbol{A.A^T}$
- U là ma trận với các cột là các vector riêng của ma trận $\boldsymbol{A.A^T}$

PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT

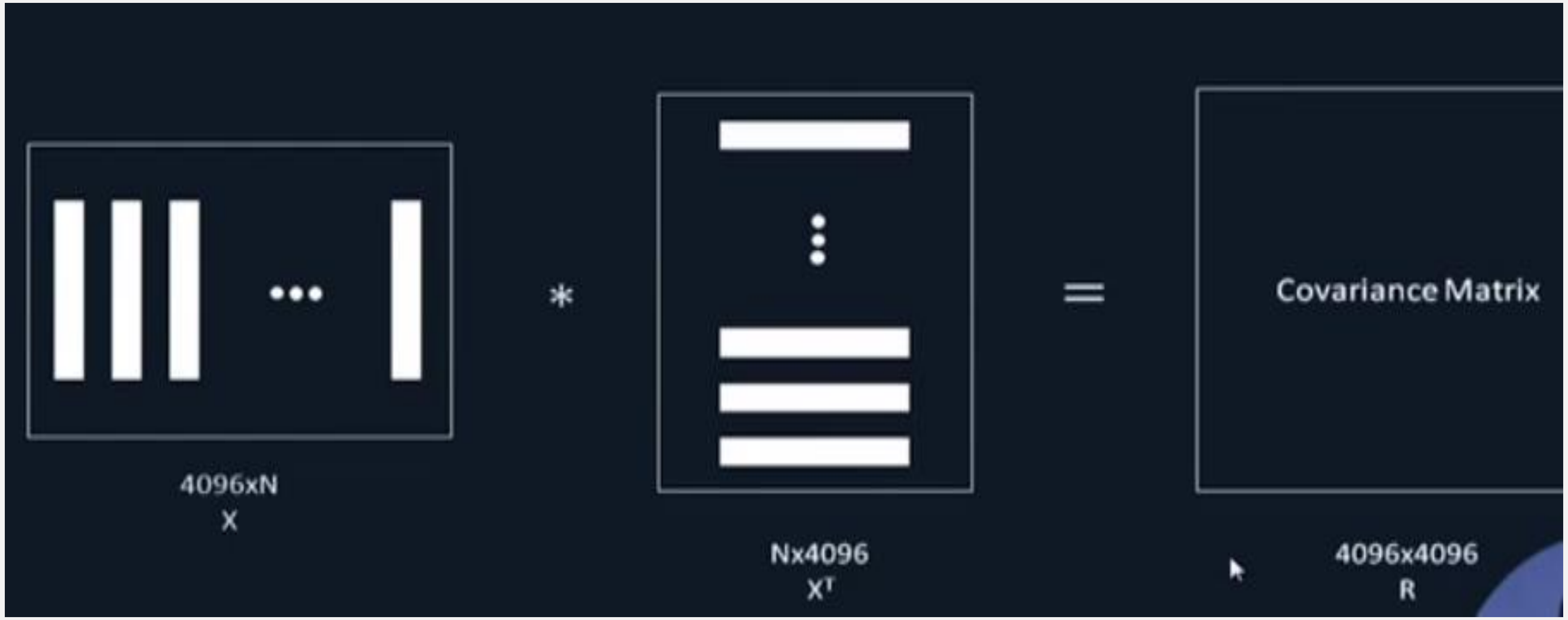
- Trong nhận diện khuôn mặt



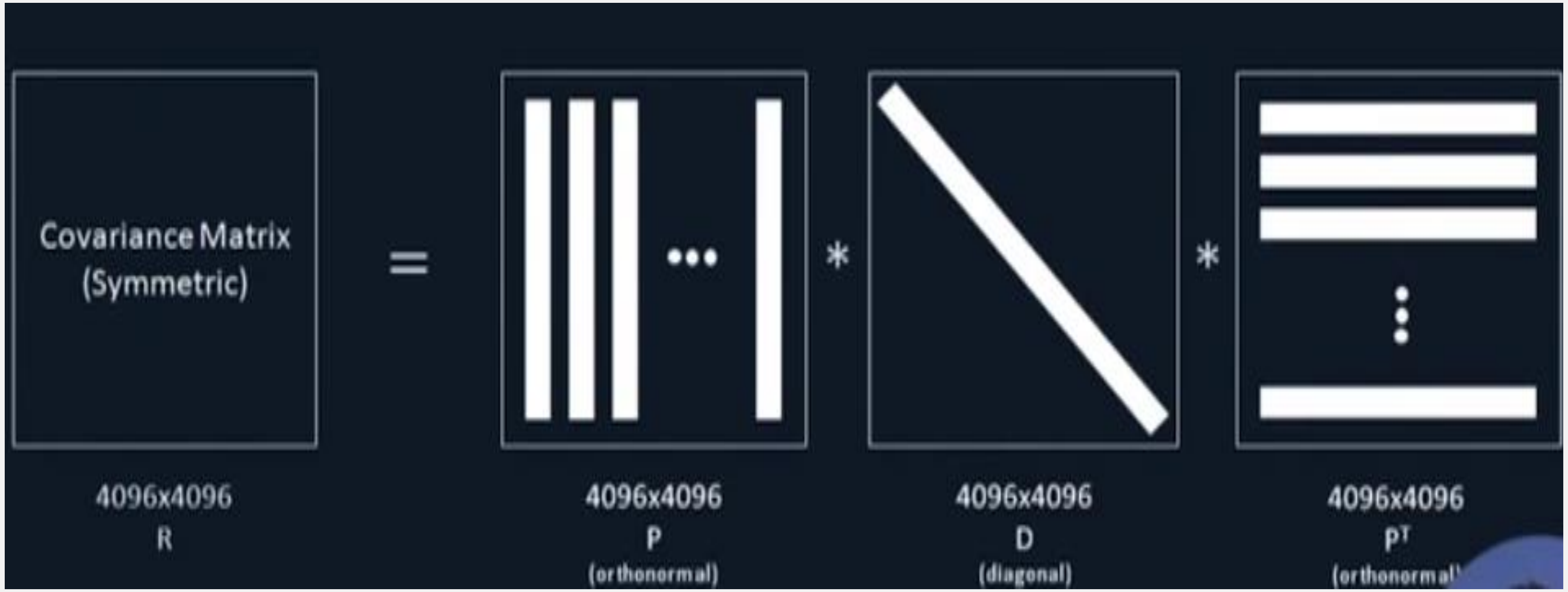
Nguồn https://github.com/Santara/ML-MOOC-NPTEL

-

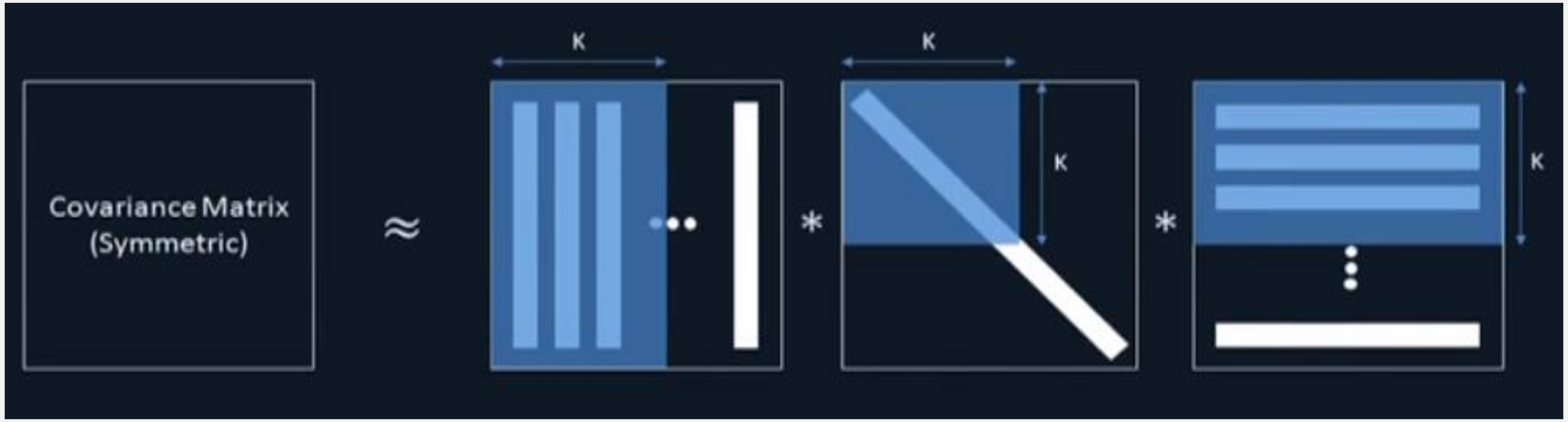
PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT

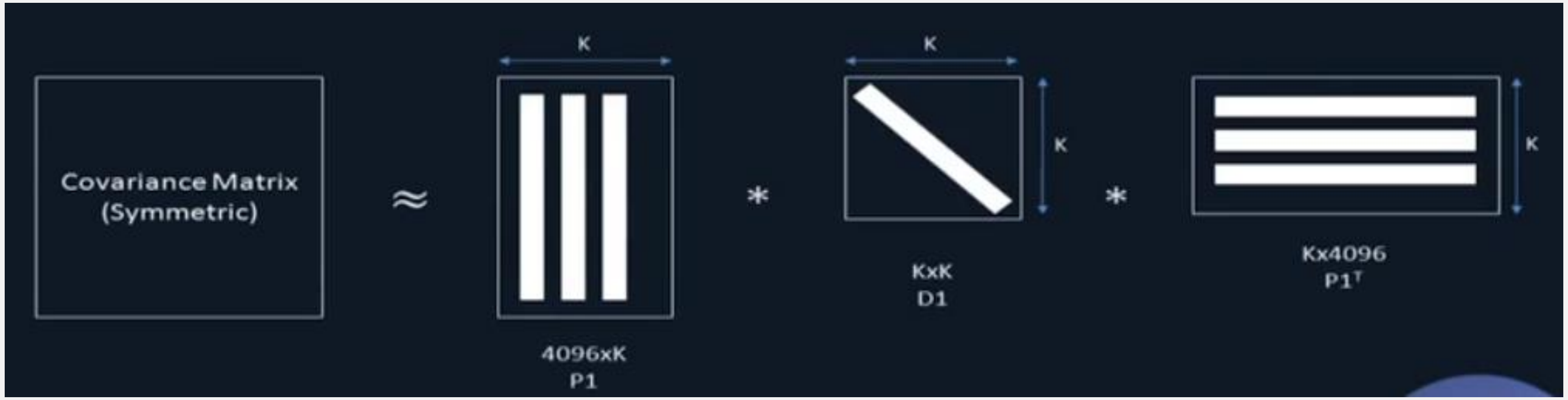


PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT

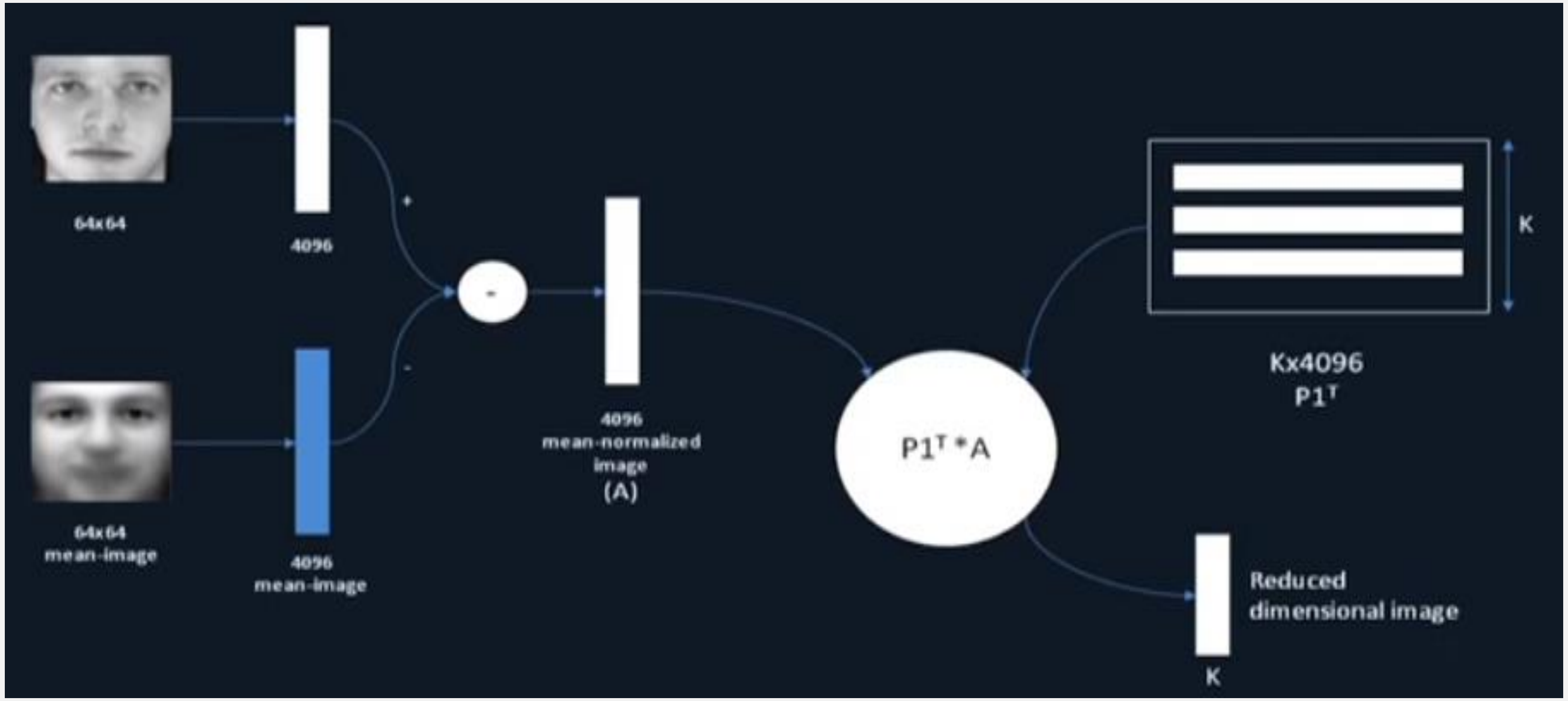


PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHÂN DIỆN KHUÔN MẶT



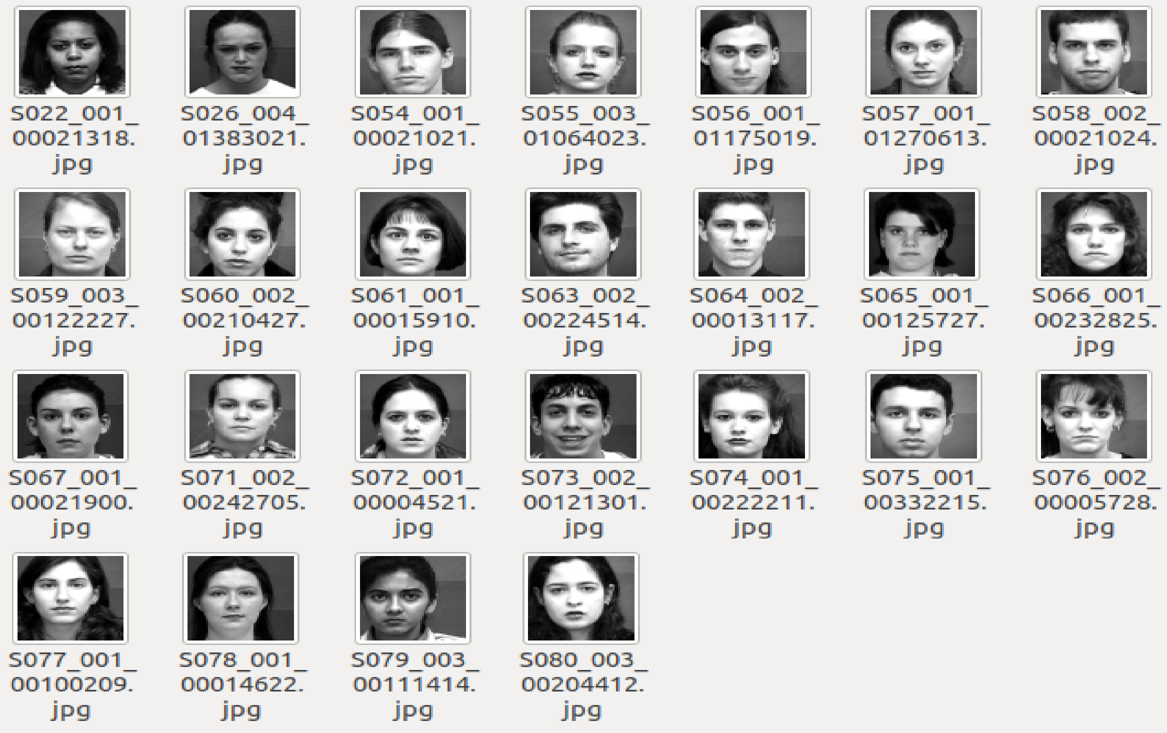


PHƯƠNG PHÁP PCA VÀ ỨNG DỤNG TRONG NHẬN DIỆN KHUÔN MẶT



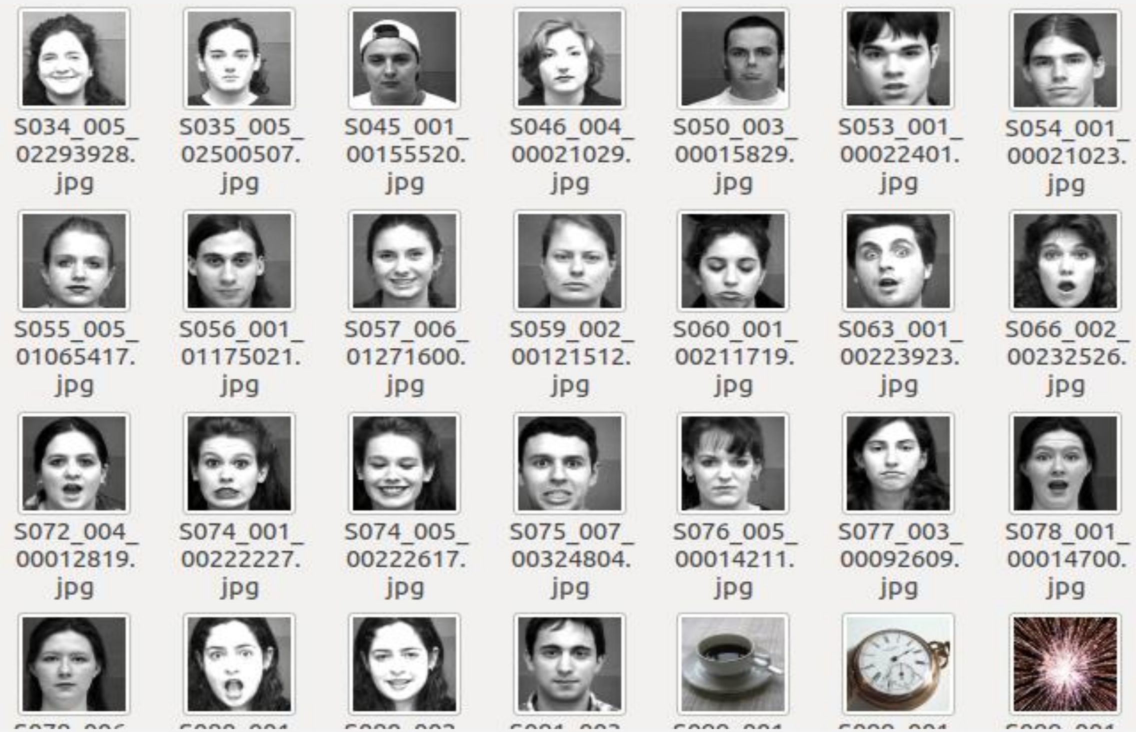
CHƯƠNG TRÌNH DEMO

- Tập dữ liệu train



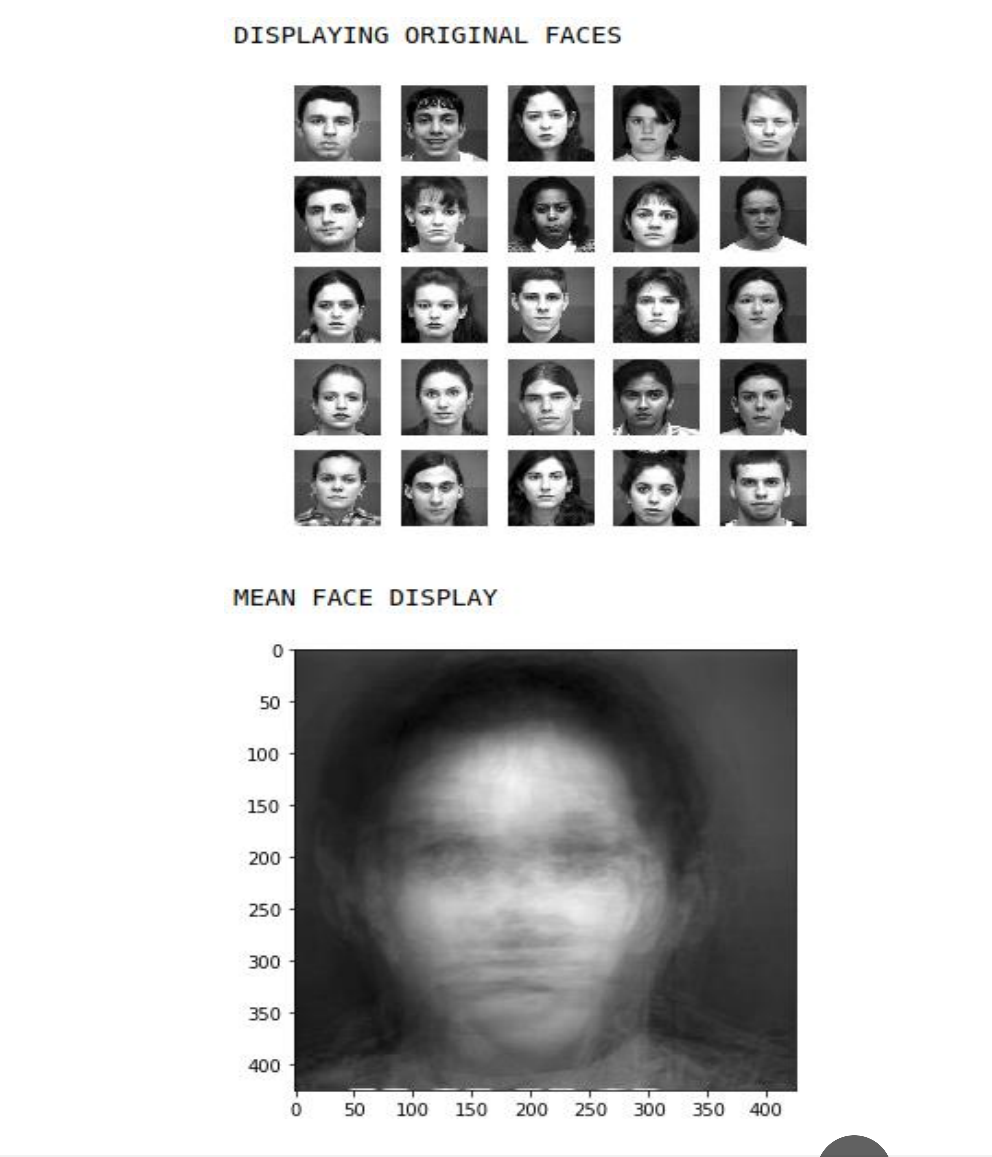
CHƯƠNG TRÌNH DEMO

- Tập dữ liệu test



CHƯƠNG TRÌNH DEMO

Ta tính Mean Face của toàn bộ tập train



CHƯƠNG TRÌNH DEMO

Ta tính ra các EIGEN FACES

RECONSTRUCTING FACES FOR K=2

RECONSTRUCTING FACES FOR K=5

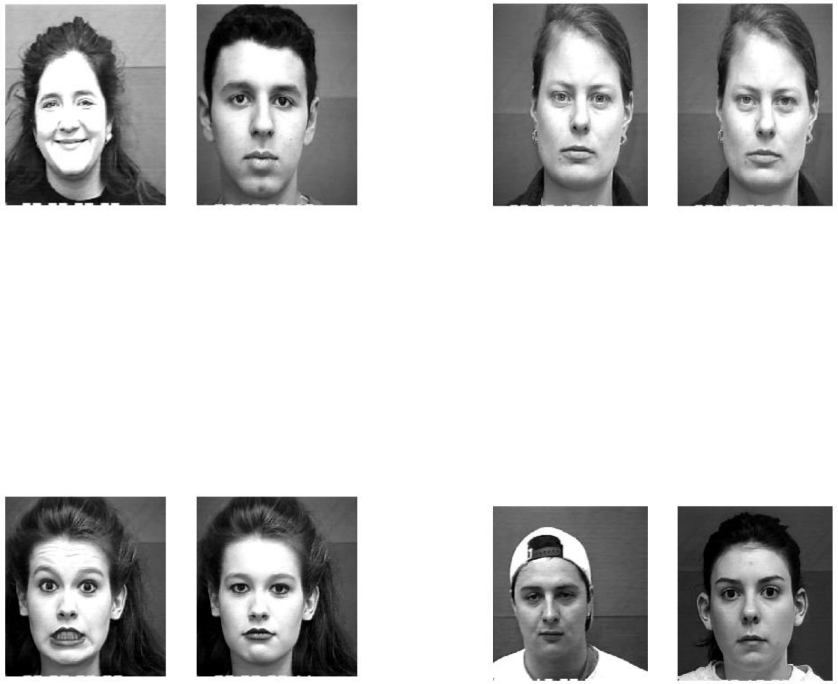
RECONSTRUCTING FACES FOR K=15

RECONSTRUCTING FACES FOR K=25

CHƯƠNG TRÌNH DEMO

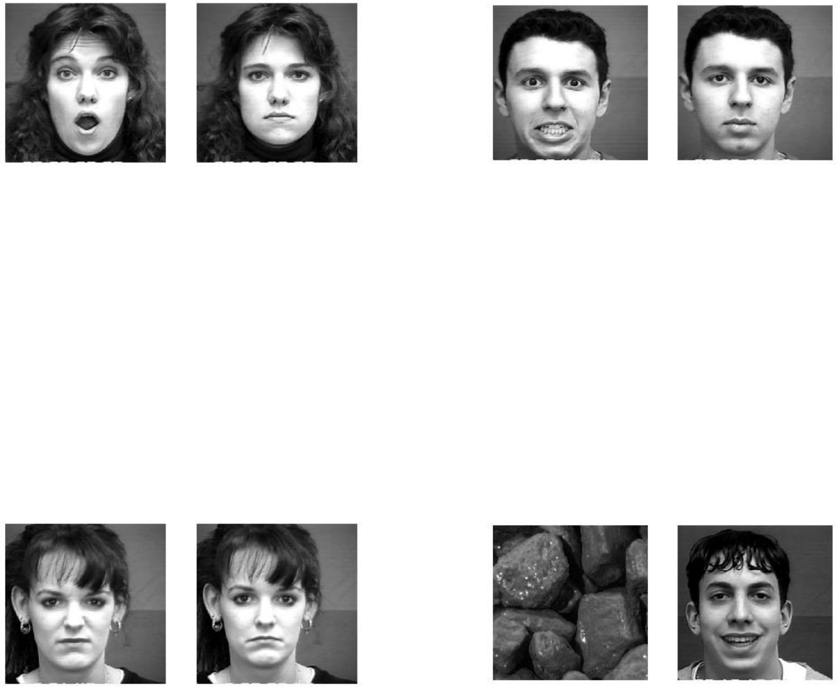
CHƯƠNG TRÌNH DEMO

- Test với K=2
- Kết quả ta thấy sai khá nhiều



CHƯƠNG TRÌNH DEMO

- Với K=5
- Kết quả nhận dạng được cải thiện đáng kể



TÀI LIỆU THAM KHẢO

- http://machinelearningcoban.com
- https://github.com/Santara/ML-MOOC-NPTEL