

CSE185

Introduction to Computer Vision

Lab 11: Eigenfaces

Instructor: Prof. Ming-Hsuan Yang
TA: Tiantian Wang & Tsai-Shien Chen

Eigenfaces

- Eigenfaces are a set of representative faces from a given dataset



Eigenfaces

- Eigenfaces are a set of representative faces from a given dataset



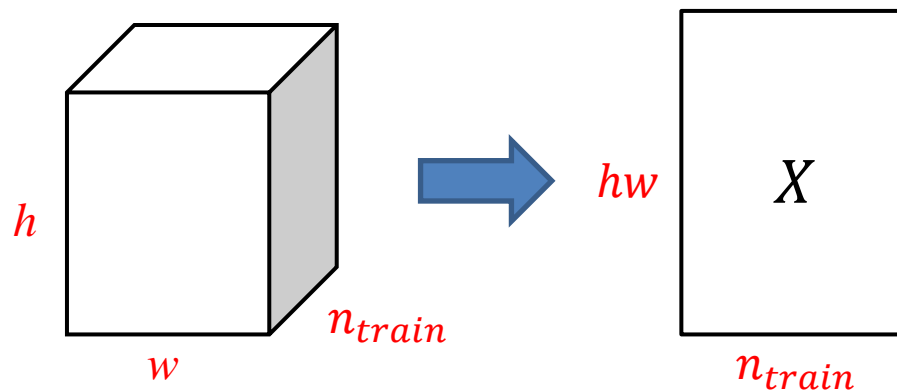
mean face



eigenfaces

Step 1: Reshape Training Data

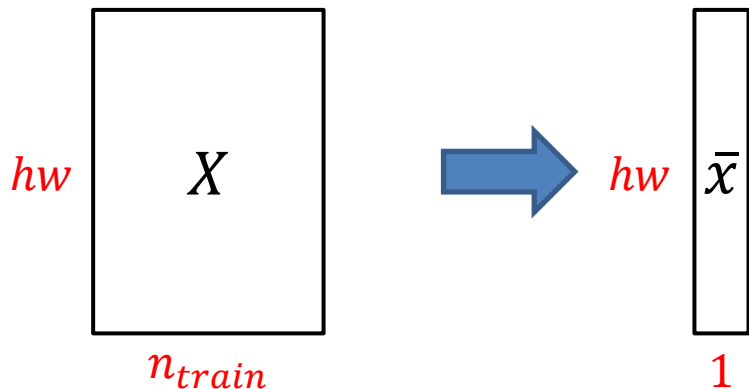
- Use `load('att_face.mat')` to load the mat file to your workspace:
 - `face_training` ($56 \times 46 \times 40$): training images
 - `face_testing` ($56 \times 46 \times 160$): testing images
 - `id_training` (40×1): the id/label of training images
 - `id_testing` (160×1): the id/label of testing images
- Reshape `face_training` from $h \times w \times n_{train}$ to $(hw) \times n_{train}$: use `X = reshape(...)`



each column is
a feature vector

Step 2: Mean Face

- Compute a mean face from X
 - \bar{x} is a $hw \times 1$ vector



- Plot mean face by reshaping it back to $h \times w$



mean face

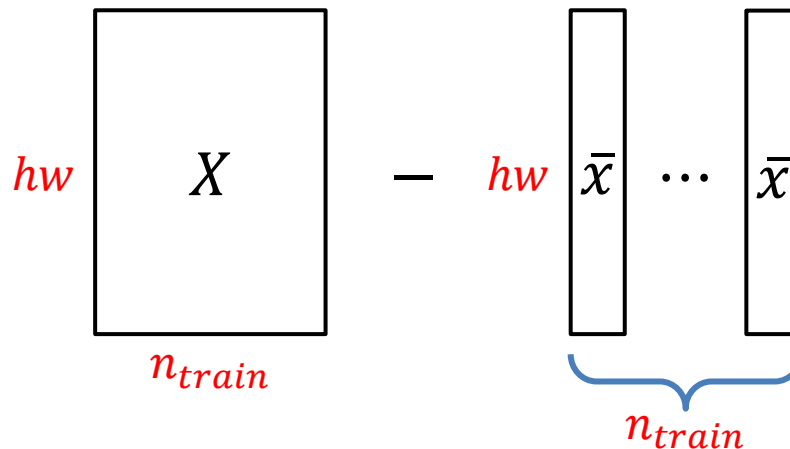
Step 3 : Covariance Matrix

- Covariance matrix:

$$C = \sum_i (x_i - \bar{x})(x_i - \bar{x})^T$$

x_i is a column of X

- 3 methods to subtract \bar{x} from each column of X
 - use for loop
 - `Y = X - repmat(x_bar, 1, n_train);`
 - `Y = bsxfun(@minus, X, x_bar);`



Step 3 : Covariance Matrix

- Covariance matrix, let $y_i = x_i - \bar{x}$:

$$C_i = y_i y_i^T, \quad C = \sum_i C_i$$

- Use for loop to compute and accumulate C_i :

The diagram illustrates the matrix multiplication $Y Y^T = C_1$. Matrix Y is a vertical rectangle with a red vertical bar on its left side, labeled hw on the left and n_{train} on the bottom. Matrix Y^T is a horizontal rectangle with a red horizontal bar on its top side, labeled n_{train} on the left and hw on the bottom. The result matrix C_1 is a square with a solid red fill, labeled hw on the left and hw on the bottom. The multiplication is indicated by a \times symbol between Y and Y^T , and an equals sign between Y^T and C_1 .

Step 3 : Covariance Matrix

- Covariance matrix, let $y_i = x_i - \bar{x}$:

$$C_i = y_i y_i^T, \quad C = \sum_i C_i$$

- Use for loop to compute and accumulate C_i :

The diagram illustrates the computation of the covariance matrix C_2 as the product of matrix Y and its transpose Y^T . Matrix Y is a tall, narrow rectangle with a red vertical stripe on its left side, labeled with height hw and width n_{train} . Matrix Y^T is a wide, short rectangle with a red horizontal stripe on its top side, labeled with height n_{train} and width hw . The product of these two matrices is shown as a red square labeled C_2 , with both height and width labeled hw .

$$\begin{matrix} hw \\ \boxed{Y} \\ n_{train} \end{matrix} \times \begin{matrix} n_{train} \\ \boxed{Y^T} \\ hw \end{matrix} = \begin{matrix} hw \\ \boxed{C_2} \\ hw \end{matrix}$$

Step 3 : Covariance Matrix

- Covariance matrix, let $y_i = x_i - \bar{x}$:

$$C_i = y_i y_i^T, \quad C = \sum_i C_i$$

- Use for loop to compute and accumulate C_i :

The diagram illustrates the computation of the covariance matrix C_3 as the sum of outer products of centered data vectors. It shows three matrices:

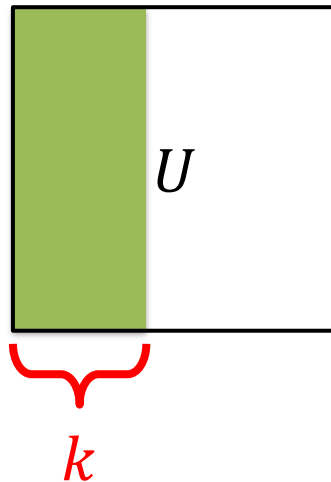
- A matrix Y of size $hw \times n_{train}$, represented as a white rectangle with a vertical red stripe. The label hw is to its left and n_{train} is below it.
- A matrix Y^T of size $n_{train} \times hw$, represented as a white rectangle with a horizontal red stripe. The label n_{train} is to its left and hw is below it.
- An equals sign followed by a red square matrix C_3 of size $hw \times hw$. The label hw is to its left and hw is below it.

The operation is shown as $Y \times Y^T = C_3$.

- There exists a one-line solution to compute the covariance matrix
- Do NOT use built-in function `cov(Y)`

Step 4: Singular Value Decomposition

- Apply SVD to the covariance matrix:
 - $[U, S, D] = \text{svd}(C)$;
 - columns in U are the eigen-vectors/eigenfaces
- Select the first k **columns** of U as our eigenfaces



Visualize Eigenfaces

- Reshape the column of U to $h \times w$, and add 0.5 or \bar{x} before `imshow`



eigenfaces

Represent Face in the Face Space

- Represent each face image as coefficients of the eigenfaces

$$coef_i = (x - \bar{x}) \cdot u_i$$

inner product

- Encode each face image as the coefficients

```
x = face_training(:, :, 1);  
x = x(:);  
% subtract mean  
x = ?  
% inner product with U  
coef = ?
```

x , \bar{x} , and u_i are
 $hw \times 1$ vectors

coef is a $k \times 1$ vector

Reconstruct Image from the Face Space

- Given a coefficient vector $coef$ and mean face \bar{x} , we can reconstruct a face image by:

$$x_{rec} = \bar{x} + coef_1 u_1 + coef_2 u_2 + \cdots + coef_k u_k$$



input image



reconstruct image

k = 10

Reconstruct Image from the Face Space

- Given a coefficient vector $coef$ and mean face \bar{x} , we can reconstruct a face image by:

$$x_{rec} = \bar{x} + coef_1 u_1 + coef_2 u_2 + \cdots + coef_k u_k$$



input image



reconstruct image

$k = 20$

Reconstruct Image from the Face Space

- Given a coefficient vector $coef$ and mean face \bar{x} , we can reconstruct a face image by:

$$x_{rec} = \bar{x} + coef_1 u_1 + coef_2 u_2 + \cdots + coef_k u_k$$



input image



reconstruct image

$k = 30$

Reconstruct Image from the Face Space

- Given a coefficient vector $coef$ and mean face \bar{x} , we can reconstruct a face image by:

$$x_{rec} = \bar{x} + coef_1 u_1 + coef_2 u_2 + \cdots + coef_k u_k$$



input image



reconstruct image

k = 40

Reconstruct Image from the Face Space

- Given a coefficient vector $coef$ and mean face \bar{x} , we can reconstruct a face image by:

$$x_{rec} = \bar{x} + coef_1 u_1 + coef_2 u_2 + \cdots + coef_k u_k$$



input image



reconstruct image

$k = 50$

Face Recognition with Eigenfaces

- In Lab 06, we use Sobel features as feature vectors
- In this lab, we will use the coefficients of eigenfaces as feature vectors

```
coef_train = zeros(k, n_train);  
% TODO: compute coef_train  
  
id_predict = zeros(size(id_testing));  
for i = 1:n_test  
    img_test = face_testing(:, :, i);  
    coef_test = ? % TODO: replace this line  
  
    error = zeros(n_train, 1);  
    for j = 1:n_train  
        diff = coef_train(:, j) - coef_test;  
        error(j) = sum( diff.^2 );  
    end
```

Face Recognition with Eigenfaces

- Fill in the table with different k

%	-----	%
%	----- Fill in this table -----	%
%	-----	%
%	k Accuracy	%
%	-----	%
%	10	%
%	-----	%
%	20	%
%	-----	%
%	30	%
%	-----	%
%	40	%
%	-----	%
%	50	%
%	-----	%

Lab Assignment 11

- Complete lab11.m
- Fill in the table at the bottom of lab11.m
- Upload **lab11.m** and 5 reconstructed images separately by using $k = 10, 20, 30, 40, 50$