Face Detection and Recognition

Reading: Chapter 18.10 and, optionally, “Face Recognition using Eigenfaces” by M. Turk and A. Pentland

Face Recognition Problem

Face Verification Problem

- Face Verification (1:1 matching)
- Face Recognition (1:N matching)

Application: Access Control

www.viisage.com

www.visionics.com
Biometric Authentication

Amazon, Mastercard, Alibaba developing methods

Pay by Selfie

Application: Video Surveillance

Face Scan at Airports

www.facesnap.de

Application: Autotagging Photos in Facebook, Flickr, Picasa, iPhoto, …
iPhoto

- Can be trained to recognize pets!


iPhoto

- Things iPhoto thinks are faces

Why is Face Recognition Hard?

The many faces of Madonna
Recognition should be **Invariant** to

- Lighting variation
- Head pose variation
- Different expressions
- Beards, disguises
- Glasses, occlusion
- Aging, weight gain
- …

**Intra-class Variability**

- Faces with intra-subject variations in pose, illumination, expression, accessories, color, occlusions, and brightness

---

**Inter-class Similarity**

- Different people may have very similar **appearance**

  Twins  
  Father and son

---

**Face Detection in Humans**

There are processes for face detection and recognition in the brain
Blurred Faces *are* Recognizable

Michael Jordan, Woody Allen, Goldie Hawn, Bill Clinton, Tom Hanks, Saddam Hussein, Elvis Presley, Jay Leno, Dustin Hoffman, Prince Charles, Cher, and Richard Nixon. The average recognition rate at this resolution is one-half.

Upside-Down Faces are Recognizable

The "Margaret Thatcher Illusion", by Peter Thompson

Illumination and Shading Affect Interpretation


Context is Important


Face Recognition Architecture
Image as a Feature Vector

• Consider an $n$-pixel image to be a point in an $n$-dimensional "image space," $\mathbf{x} \in \mathbb{R}^n$
• Each pixel value is a coordinate of $\mathbf{x}$
• Preprocess images so faces are cropped and (roughly) aligned (position, orientation, and scale)

Nearest Neighbor Classifier

$\{ \mathcal{R}_j \}$ is a set of training images of frontal faces

$$ID(I) = \arg \min_{j} dist(R_j, I)$$

Key Idea

• Expensive to compute nearest neighbor when each image is big ($n$ dimensional space)
• Not all images are very likely – especially when we know that every image contains a face. That is, images of faces are highly correlated, so compress them into a low-dimensional, linear subspace that retains the key appearance characteristics of people’s faces

Eigenface Representation

• Each face image is represented by a weighted (linear) combination of a small number of “component” or “basis” faces
• Each of these basis faces is called an “eigenface”
Eigenface Representation

- These basis faces can be weighted differently to represent any face
- So, a vector of weights represents each face

Learning Eigenfaces

- How do we pick the set of basis faces?
- We take a set of real training faces
- Find the set of basis faces that best represent the differences between the training faces
- Use a statistical criterion for quantifying the “best representation of the differences”

Eigenfaces (Turk and Pentland, 1991)

- The set of training face images is clustered in a “subspace” of the set of all images
- **Training Phase**
  - Find best subspace to reduce the dimensionality
  - Transform all training images into the subspace
- **Testing Phase**
  - Transform test image into the subspace and use a nearest-neighbor classifier to label it

Linear Subspaces

- Suppose we have points in 2D and we have a line through that space
- We can project each point onto that 1D line
- Represent a point by its position on the line
Linear Subspaces

Some lines will represent the data well and others not, depending on how well the projection separates the data points.

Dimensionality Reduction

We can represent the orange points well with only their $v_1$ coordinates (since $v_2$ coordinates are all $\approx 0$).

This makes it much cheaper to store and compare points.

A bigger deal for higher dimensional problems.

Principal Component Analysis (PCA)

- Problems arise when performing recognition in a high-dimensional space ("curse of dimensionality").
- Significant improvements can be achieved by first mapping the data into a lower-dimensional subspace.

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \rightarrow \text{reduce dimensionality} \rightarrow y = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix} (K \ll N)$$

- The goal of PCA is to reduce the dimensionality of the data while retaining the largest variations present in the original data.
Principal Component Analysis (PCA)

- Dimensionality reduction implies information will be lost
- How to determine the best lower dimensional subspace?
  - Maximize information content in the compressed data by finding a set of $k$ orthogonal vectors that account for as much of the data’s variance as possible
  - Best dimension = direction in $n$-D with max variance
  - 2nd best dimension = direction orthogonal to first with max variance

• The covariance between two pixels expresses how correlated they are (i.e., to what degree they “co-vary”)
• The $n \times n$ covariance matrix for an image with $n$ pixels expresses how correlated each pair of pixels is
• The eigenvectors of this matrix with the largest eigenvalues correspond to the pixels in which the data varies the most

Principal Component Analysis (PCA)

• Geometric interpretation
  - PCA projects the data along the directions where the data varies the most
  - The vectors define a new coordinate system in which to represent each image

Eigenvectors

• An eigenvector is a vector, $u$, that satisfies $\lambda_i u_i = Cu_i$

  where $C$ is a square matrix, and $\lambda$ is a scalar called the eigenvalue

• Example (for 2D data):

  $C = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$
  $u = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$
  $Cu = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$

• So eigenvalue $\lambda = 4$ for this eigenvector, $u$
Principal Component Analysis (PCA)

- The best low-dimensional space can be determined by the "best" eigenvectors of the covariance matrix of the data, i.e., the eigenvectors corresponding to the largest eigenvalues – also called "principal components".

- Can be efficiently computed using the Singular Value Decomposition (SVD) algorithm.

Algorithm

- Each input image, $X_i$, is an $n$-D column vector of all pixel values (in raster order).
- Compute "average face" image $A$ from all $M$ training images of all people:
  \[
  A = \frac{1}{M} \sum_{i=1}^{M} X_i
  \]
- Normalize each training image, $X_p$, by subtracting the average face:
  \[
  Y_i = X_i - A
  \]

Algorithm

- Stack all training images together

  $Y = [Y_1 Y_2 ... Y_M]$  
  \[  \text{matrix} \]

- Compute the Covariance Matrix $C$:

  \[
  C = YY^T = \frac{1}{M} \sum_i Y_i Y_i^T
  \]

  where the eigenvalues are

  \[
  \lambda_1 > \lambda_2 > ... > \lambda_n
  \]

  and the corresponding eigenvectors are $u_1, u_2, ..., u_n$. 

Algorithm

- Each $u_i$ is an $n \times 1$ eigenvector called an “eigenface” (to be cute!)
- Each $u_i$ is a vector/direction in “face space”
  \[ Y_i = w_1 u_1 + w_2 u_2 + \ldots + w_n u_n \]
  \[ X_i = \sum_{i=1}^{n} w_i u_i + A \]
- Image can be exactly reconstructed by a linear combination of all eigenvectors

Eigenface Representation

Each face image is represented by a weighted combination of a small number of “component” or “basis” faces

Using Eigenfaces

- (Approximate) Reconstruction of an image of a face from a set of weights
- Recognition of a person from a new face image
Face Image Reconstruction

- Face $X$ in “face space” coordinates:
\[
x \rightarrow \begin{bmatrix} u_1^T (x - \mu), \ldots, u_k^T (x - \mu) \end{bmatrix}
\]

- Reconstruction:
\[
\hat{X} = A + w_1 u_1 + w_2 u_2 + w_3 u_3 + w_4 u_4 + \ldots
\]

Eigenfaces Recognition Algorithm

**Modeling (Training Phase)**

1. Given a collection of labeled training images
2. Compute mean image, $A$
3. Compute $k$ eigenvectors, $u_1, \ldots, u_k$, of covariance matrix corresponding to $k$ largest eigenvalues
4. Project each training image, $X_i$, to a point in $k$-dimensional “face space.”
   
   for $j = 1, \ldots, k$ compute $w_j = u_j^T (X_i - A)$
   
   $X_i$ projects to $W = [w_1, w_2, \ldots, w_k]$

Eigenfaces Algorithm

**Recognition (Testing Phase)**

1. Given a test image, $G$, project it into face space
   
   for $j = 1, \ldots, k$ compute $w_j = u_j^T (G - A)$

2. Classify it as the class (person) that is closest to it
   (as long as its distance to the closest person is “close enough”)
Choosing $k$

- How many eigenfaces to use?
- Look at the decay of the eigenvalues
  - the eigenvalue tells you the amount of variance “in the direction” of that eigenface
  - ignore eigenfaces with low variance

Example: Training Images

Note: Faces must be approximately registered (translation, rotation, size, pose)!

[ Turk & Pentland, 2001]
Example

Top eigenvectors: \( u_1, \ldots, u_k \)

Average: \( A \)

Experimental Results

- Training set: 7,562 images of approximately 3,000 people
- \( k = 20 \) eigenfaces computed from a sample of 128 images
- Test set accuracy on 200 faces was 95%

Difficulties with PCA

- Projection may suppress important details
  - smallest variance directions may not be unimportant
- Method does not take discriminative task into account
  - we want to compute features that allow good discrimination
  - not necessarily the same as largest variance

Limitations

- PCA assumes that the data has a Gaussian distribution

The shape of this dataset is not well described by its principal components
Limitations

- Background (de-emphasize the outside of the face – e.g., by multiplying the input image by a 2D Gaussian window centered on the face)
- Lighting conditions (performance degrades with light changes)
- Scale (performance decreases quickly with changes to head size); possible solutions:
  - multi-scale eigenspaces
  - scale input image to multiple sizes
- Orientation (performance decreases but not as fast as with scale changes)
  - plane rotations can be handled
  - out-of-plane rotations are more difficult to handle

Extension: Eigenfeatures

- Describe and encode a set of facial features: eigeneyes, eigennoses, eigenmouths
- Use for detecting facial features
Recognition using Eigenfeatures