Recognition

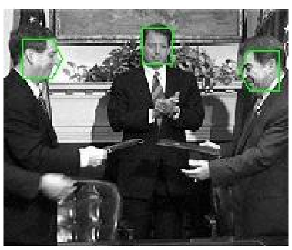
Introduction to Computer Vision
CSE 152
Lecture 16

Announcements

- Homework 3 is due May 29, 11:59 PM
- Homework 4 will be assigned today
- Final exam will be a take home exam

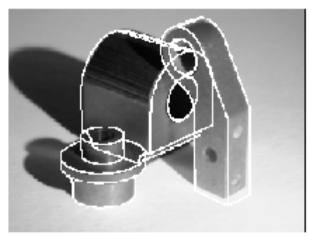
- Reading:
  – Section 14.2.1 Eigenfaces

Recognition



Given a database of objects and an image determine what, if any of the objects are present in the image.

Recognition

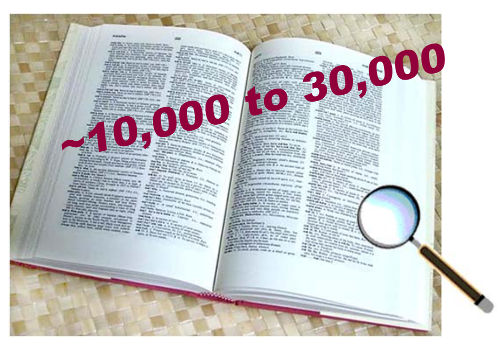


Given a database of objects and an image determine what, if any of the objects are present in the image.

Recognition

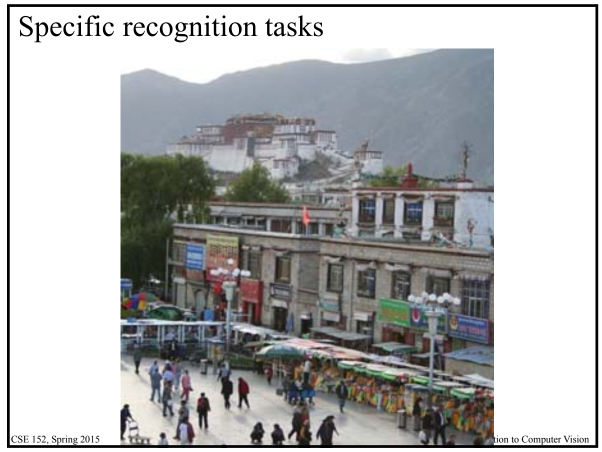
Given a database of objects and an image determine what, if any of the objects are present in the image.

How many visual object categories are there?



Biederman 198


~10,000 to 30,000


Specific recognition tasks
CSE 152, Spring 2015
tion to Computer Vision


Scene categorization
• outdoor/indoor
• city/forest/factory/etc.
CSE 152, Spring 2015
tion to Computer Vision

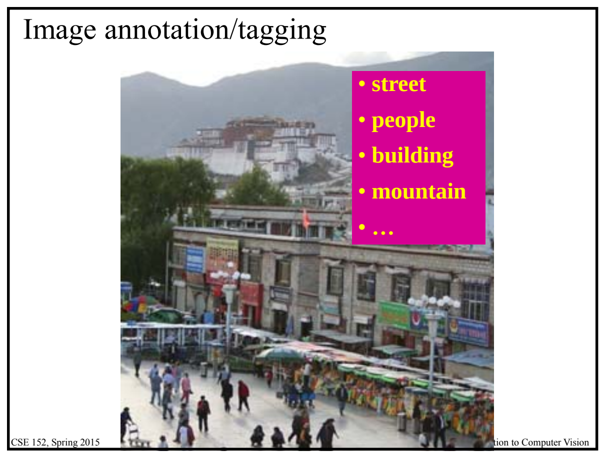


Image annotation/tagging
• street
• people
• building
• mountain
• …
CSE 152, Spring 2015
tion to Computer Vision


Object detection
• find pedestrians
CSE 152, Spring 2015
tion to Computer Vision

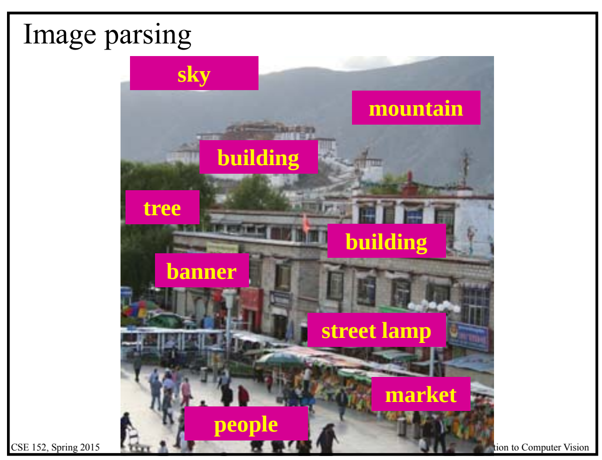
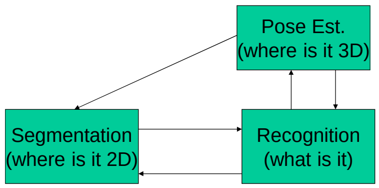


Image parsing
sky
mountain
building
tree
building
banner
street lamp
market
people
CSE 152, Spring 2015
tion to Computer Vision

Object Recognition: The Problem

Given: A database D of "known" objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object

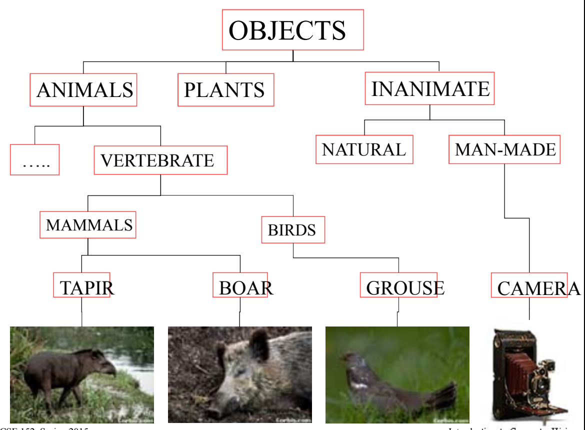


WHAT AND WHERE!!!

Within-class variations

Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional
- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability (perspective, orthographic projection)
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial
- Clutter in background -> false positives



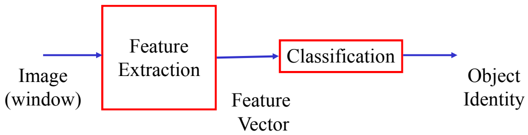
- Categories near top of tree (e.g., vehicles) – lots of within class variability
- Fine grain categories (e.g., species of birds) -- Moderate within class variation
- Instance recognition (e.g., person identification) – within class mostly shape articulation, bending, etc.

Appearance-Based Vision for Instances Level Recognition:

A Pattern Classification Viewpoint

1. Feature Space + Nearest Neighbor
2. Dimensionality Reduction
3. Bayesian Classification
4. Appearance Manifolds

Sketch of a Pattern Recognition Architecture



Sliding window approaches

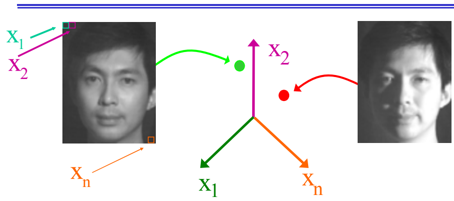
Example: Face Detection

- Scan window over image.
- Search over position & scale.
- Classify window as either:
  - Face
  - Non-face

Window → Classifier → Face / Non-face

- So, what are the features?
- So, what is the classifier?

The Space of Images



- We will treat an d-pixel image as a point in an d-dimensional space, $\mathbf{x} \in \mathbf{R}^d$.
- Each pixel value is a coordinate of $\mathbf{x}$.

More features

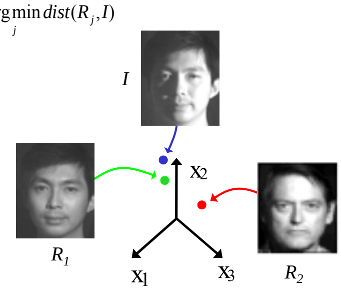
- Filtered image
- Filter with multiple filters (bank of filters)
- Histogram of colors
- Histogram of Gradients (HOG)
- Haar wavelets
- Scale Invariant Feature Transform (SIFT)
- Speeded Up Robust Feature (SURF)

- So, what are the features?
- So, what is the classifier?

Nearest Neighbor Classifier

$\{ R_j \}$ are set of training images.

$$ID = \arg\min_j dist(R_j, I)$$



Comments on Nearest Neighbor

- Sometimes called "Template Matching"
- Variations on distance function (e.g. L_1, robust distances)
- Multiple templates per class- perhaps many training images per class.
- Expensive to compute k distances, especially when each image is big (d-dimensional).
- May not generalize well to unseen examples of class.
- No worse than twice the error rate of the optimal classifier -- if enough training samples
- Some solutions:
  - Bayesian classification
  - Dimensionality reduction

Do features vectors have structure in the image space?

- Faces of individuals cluster in the image space. (Not true).
- Faces of individuals are confined to a linear or affine subspace of $\mathbf{R}^d$
- Faces of an individual are approximated by a linear subspace
- Faces and objects lie on or near a manifold in the space of images.

An idea:

Represent the set of images as a linear subspace

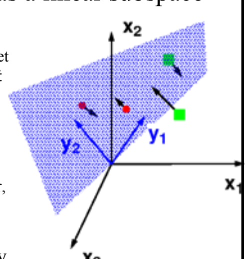
What is a linear subspace?

Let *V* be a vector space and let *W* be a subset of *V*. Then *W* is a subspace if and only if:

1. The zero vector, **0**, is in *W*.
2. If **u** and **v** are elements of *W*, then any linear combination of **u** and **v** is an element of *W*; a**u** + b**v** $\in$ *W*
3. If **u** is an element of *W* and *c* is a scalar, then the scalar product *c***u** $\in$ *W*

A *k*-dimensional subspace is spanned by *k* linearly independent vectors

It is spanned by a k-dimensional orthogonal basis.



Example: A 2-D linear subspace of $\mathbf{R}^3$ spanned by y_1 and y_2.

Linear Subspaces & Linear Projection

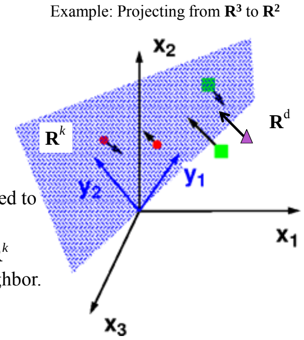
- A *d*-pixel image $x \in \mathbf{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbf{R}^k$ by

$$y = Wx$$

where *W* is an *k* by *d* matrix.

- Each training image is projected to the subspace
- Recognition is performed in $\mathbf{R}^k$ using, for example, nearest neighbor.
- How do we choose a good *W*?

Example: Projecting from $\mathbf{R}^3$ to $\mathbf{R}^2$



Linear Subspaces & Recognition

1. Eigenfaces: Approximate all training images as a single linear subspace
2. Distance to subspace: Represent lighting variation w/o shadowing for a single individual as a 3-D linear subspace. *n* individuals are modeled as as *n* 3-D linear subspaces.
3. Fisherfaces: Project all training images to a single subspace that enhances discriminabiliy

Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of n feature vectors $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ in $\mathbb{R}^d$. Write

$$\boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{x}_i$$

$$\Sigma = \frac{1}{n-1}\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

The unit eigenvectors of Σ — which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the k-dimensional set of linear features that preserves the most variance.

Algorithm 22.5: *Principal components analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.*

Eigen decomposition of covariance matrix.

Alternative: singular value decomposition of (mean-deviation form of) data matrix.

Singular value decomposition and its relationship to eigen decomposition

- Any *m* by *n* matrix ***A*** may be factored such that

$$A = U\Sigma V^T$$

[m x n] = [m x m][m x n][n x n]

- ***U***: *m* by *m*, orthogonal matrix
  - Columns of **U** are the eigenvectors of AA^T
- ***V***: *n* by *n*, orthogonal matrix,
  - columns are the eigenvectors of A^TA
- **Σ:** *m* by *n*, diagonal with non-negative entries ($\sigma_1, \sigma_2, \ldots, \sigma_s$) with s=min(m,n) are called the called the singular values
  - **Singular values are the square roots of eigenvalues of both AA^T and A^TA**
  - *Result of SVD algorithm:* $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$

SVD Properties

- In Matlab [u s v] = svd(A), and you can verify that: A=u*s*v'
- r=Rank(***A***) = # of non-zero singular values.
- ***U***, ***V*** give an orthonormal bases for the subspaces of ***A***:
  - 1st *r* columns of ***U***: Column space of ***A***
  - Last *m - r* columns of ***U***: Left nullspace of ***A***
  - 1st *r* columns of ***V***: Row space of ***A***
  - 1st *n - r* columns of ***V***: (Right) nullspace of ***A***
- *For some d where $d \leq r$, the first d column of U provide the best d-dimensional basis for columns of A in least squares sense.*

Performing PCA with SVD

- Singular values of *A* are the square roots of eigenvalues of both AA^T and A^TA
- Columns of U are corresponding Eigenvectors of AA^T
- And $\sum_{i=1}^{n} a_i a_i^T = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^T = AA^T$
- Covariance matrix is:

$$\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T$$

- So, ignoring 1/(n-1), subtract mean image μ from each input image, create a d x n data matrix, and perform thin SVD on the data matrix. D=[x_1-μ | x_2-μ | … x_n-μ]

Economy SVD

- Any *m* by *n* matrix ***A*** may be factored such that

$$A = U\Sigma V^T$$

[m x n] = [m x m][m x n][n x n]

- If m>n, then one can view **Σ** as: (i.e., more pixels than images)

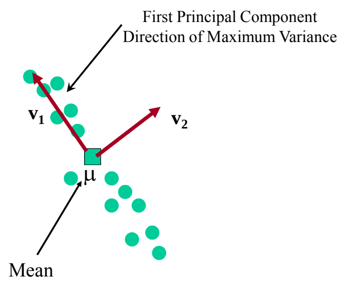
$$\begin{bmatrix} \Sigma' \\ 0 \end{bmatrix}$$

- Where **Σ**'=diag($\sigma_1, \sigma_2, \ldots, \sigma_s$) with s=min(m,n), and lower matrix is (n-m by m) of zeros.
- Alternatively, you can write:

$$A = U'\Sigma' V^T$$

- In Matlab, thin SVD is:[U S V] = svd(A,'econ')

PCA Example



Eigenfaces

Modeling

1. Given a collection of n training images x_i, represent each one as a d-dimensional column vector
2. Compute the mean image and covariance matrix.
3. Compute k Eigenvectors of the covariance matrix corresponding to the k largest Eigenvalues and form matrix $W^T=[u_1, u_2,\ldots,u_k]$ (Or perform using SVD)
   - Note that the Eigenvectors are images
4. Project the training images to the k-dimensional Eigenspace. $y_i=Wx_i$

Recognition

1. Given a test image x, project the vectorized image to the Eigenspace by $y=Wx$
2. Perform classification of y to the projected training images.

Why is W a good projection?

- The linear subspace spanned by W maximizes the variance (i.e., the spread) of the projected data.
- W spans a subspace that is the best approximation to the data in a least squares sense. E.g., W is the subspace that minimizes the the sum of the squared distances from each datapoint to the the subspace.

Eigenfaces: Training Images

[Turk, Pentland 91]

Eigenfaces

Mean Image

Basis Images

Difficulties with PCA

- Projection may suppress important detail
  - smallest variance directions may not be unimportant
- Method does not take discriminative task into account
  - typically, we wish to compute features that allow good discrimination
  - not the same as largest variance or minimizing reconstruction error.

Alternative projections

Fisherfaces: Class specific linear projection

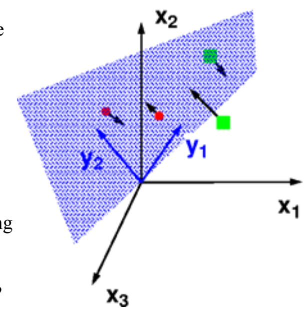
P. Belhumeur, J. Hespanha, D. Kriegman, *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*, PAMI, July 1997, pp. 711--720.

- An *n*-pixel image $x \in \mathbf{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbf{R}^k$ by

$$y = Wx$$

where W is an k by d matrix.

- Recognition is performed using nearest neighbor in $\mathbf{R}^k$.
- How do we choose a good W?



PCA & Fisher's Linear Discriminant

- Between-class scatter

$$S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$

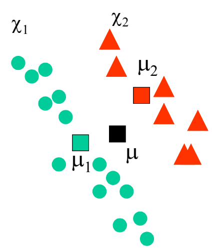
- Within-class scatter

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

- Total scatter

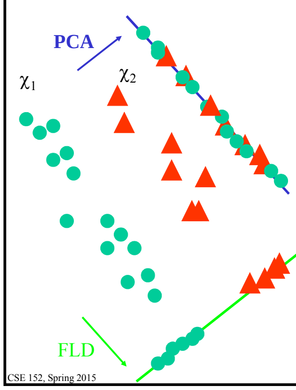
$$S_T = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu)(x_k - \mu)^T = S_B + S_W$$

- Where
  - c is the number of classes
  - μ_i is the mean of class χ_i
  - $|\chi_i|$ is number of samples of χ_i.



If the data points x_i are projected by $y_i = Wx_i$ and the scatter of x_i is S, then the scatter of the projected points y_i is $W^T S W$

PCA & Fisher's Linear Discriminant



- PCA (Eigenfaces)

$$W_{PCA} = \arg\max_W |W^T S_T W|$$

Maximizes projected total scatter

- Fisher's Linear Discriminant

$$W_{fld} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

Maximizes ratio of projected between-class to projected within-class scatter

Computing the Fisher Projection Matrix

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \dots \ \mathbf{w}_m] \quad (4)$$

where $\{\mathbf{w}_i \mid i = 1, 2, \dots, m\}$ is the set of generalized eigenvectors of S_B and S_W corresponding to the m largest generalized eigenvalues $\{\lambda_i \mid i = 1, 2, \dots, m\}$, i.e.,

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \quad i = 1, 2, \dots, m$$

- The w_i are orthonormal
- There are at most *c*-1 non-zero generalized Eigenvalues, so $m \leq c-1$
- Can be computed with *eig* in Matlab

Fisherfaces

$$W = W_{fld} W_{PCA}$$

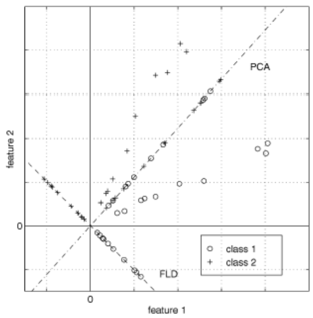
$$W_{PCA} = \arg\max_W |W^T S_T W|$$

$$W_{fld} = \arg\max_W \frac{|W^T W_{PCA}^T S_B W_{PCA} W|}{|W^T W_{PCA}^T S_W W_{PCA} W|}$$

- Since S_W is rank N-c, project training set to subspace spanned by first *N-c* principal components of the training set.
- Apply FLD to *N-c* dimensional subspace yielding *c-1* dimensional feature space.

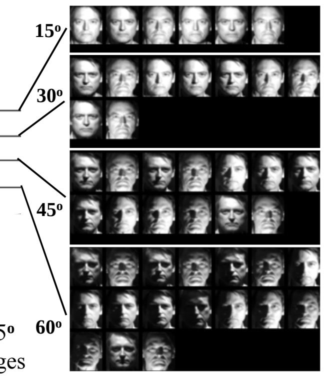
- Fisher's Linear Discriminant projects away the within-class variation (lighting, expressions) found in training set.
- Fisher's Linear Discriminant preserves the separability of the classes.

PCA vs. FLD



CSE 152, Spring 2015 | Introduction to Computer Vision

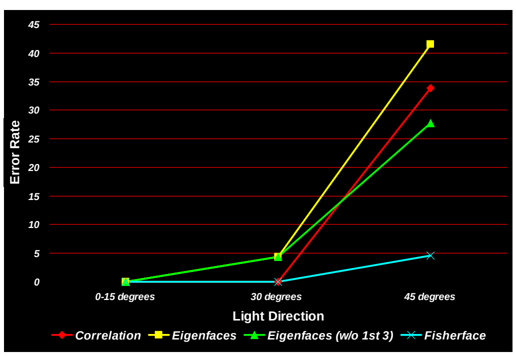
Harvard Face Database



- 10 individuals
- 66 images per person
- Train on 6 images at 15°
- Test on remaining images

CSE 152, Spring 2015 | Introduction to Computer Vision

Recognition Results: Lighting Extrapolation



CSE 152, Spring 2015 | Introduction to Computer Vision