What do we mean by “object recognition”?  

Verification: is that a lamp?  

Next 15 slides adapted from Li, Fergus, & Torralba’s excellent short course on category and object recognition.

**Readings**  
- Szeliski, Chapter 14.2.1 (eigenfaces)
Detection: are there people?

Identification: is that Potala Palace?

Object categorization:
- mountain
- tree
- building
- banner
- street lamp
- people
- vendor

Scene and context categorization:
- outdoor
- city
- ...
Object recognition
Is it really so hard?

Find the chair in this image

Output of normalized correlation

This is a chair

Find the chair in this image

Pretty much garbage
Simple template matching is not going to make it

Object recognition
Is it really so hard?

A "popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts." Nisbet & Binford, 1977.

Why not use SIFT matching for everything?

• Works well for object instances

• Not great for generic object categories
Applications: Computational photography

Applications: Assisted driving

Pedestrian and car detection

Lane detection

• Collision warning systems with adaptive cruise control,
• Lane departure warning systems,
• Rear object detection systems.

Applications: image search

Mobileye (c) 2005

Pedestrian and car detection

Lane detection

Applications: Computational photography

Applications: Assisted driving

Pedestrian and car detection

Lane detection

• Collision warning systems with adaptive cruise control,
• Lane departure warning systems,
• Rear object detection systems.

Applications: image search

Mobileye (c) 2005

Pedestrian and car detection

Lane detection

Applications: Computational photography

Applications: Assisted driving

Pedestrian and car detection

Lane detection

• Collision warning systems with adaptive cruise control,
• Lane departure warning systems,
• Rear object detection systems.

Applications: image search

Mobileye (c) 2005

Pedestrian and car detection

Lane detection

Applications: Computational photography

Applications: Assisted driving

Pedestrian and car detection

Lane detection

• Collision warning systems with adaptive cruise control,
• Lane departure warning systems,
• Rear object detection systems.

Applications: image search

Mobileye (c) 2005

Pedestrian and car detection

Lane detection

Applications: Computational photography

Applications: Assisted driving

Pedestrian and car detection

Lane detection

• Collision warning systems with adaptive cruise control,
• Lane departure warning systems,
• Rear object detection systems.

Applications: image search

Mobileye (c) 2005

Pedestrian and car detection

Lane detection

Applications: Computational photography

Applications: Assisted driving

Pedestrian and car detection

Lane detection

• Collision warning systems with adaptive cruise control,
• Lane departure warning systems,
• Rear object detection systems.

Applications: image search

Mobileye (c) 2005

Pedestrian and car detection

Lane detection

Applications: Computational photography

Applications: Assisted driving

Pedestrian and car detection

Lane detection

• Collision warning systems with adaptive cruise control,
• Lane departure warning systems,
• Rear object detection systems.
Challenges: viewpoint variation

Michelangelo 1475-1564

Challenges: illumination variation

Challenges: occlusion

Magritte, 1957

Challenges: scale
Challenges: deformation

Challenges: background clutter

Challenges: intra-class variation

How do human do recognition?

- We don’t completely know yet
- But we have some experimental observations.
Observation 1

• We can recognize familiar faces even in low-resolution images

Observation 2:

Jim Carrey
Kevin Costner

• High frequency information is not enough

What is the single most important facial features for recognition?

Observation 4:

• Image Warping is OK
The list goes on

Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About


Let’s start simple

- Today
  - skin detection
  - eigenfaces

Face detection

- Do these images contain faces? Where?

One simple method: skin detection

Skin pixels have a distinctive range of colors
- Corresponds to region(s) in RGB color space
- For visualization, only R and G components are shown above

Skin classifier
- A pixel $X = (R,G,B)$ is skin if it is in the skin region
- But how to find this region?
Skin detection

Learn the skin region from examples
- Manually label pixels in one or more “training images” as skin or not skin
- Plot the training data in RGB space
  - skin pixels shown in orange, non-skin pixels shown in blue
  - some skin pixels may be outside the region, non-skin pixels inside. Why?

Skin classifier
- Given \( X = (R,G,B) \): how to determine if it is skin or not?

Skin classification techniques

Skin classifier
- Given \( X = (R,G,B) \): how to determine if it is skin or not?
- Nearest neighbor
  - find labeled pixel closest to \( X \)
  - choose the label for that pixel
- Data modeling
  - fit a model (curve, surface, or volume) to each class
- Probabilistic data modeling
  - fit a probability model to each class

Probability

Basic probability
- \( X \) is a random variable
- \( P(X) \) is the probability that \( X \) achieves a certain value
- called a PDF
- probability distribution/density function
- a 2D PDF is a surface, 3D PDF is a volume

- \( 0 \leq P(X) \leq 1 \)
- \( \int_{-\infty}^{\infty} P(X) dX = 1 \) or \( \sum P(X) = 1 \)
- continuous \( X \)
- discrete \( X \)

- Conditional probability: \( P(X | Y) \)
  - probability of \( X \) given that we already know \( Y \)

Probabilistic skin classification

Now we can model uncertainty
- Each pixel has a probability of being skin or not skin
  - \( P(\sim \text{skin}|R) = 1 - P(\text{skin}|R) \)

Skin classifier
- Given \( X = (R,G,B) \): how to determine if it is skin or not?
- Choose interpretation of highest probability
  - set \( X \) to be a skin pixel if and only if \( R_1 < X < R_2 \)

Where do we get \( P(\text{skin}|R) \) and \( P(\sim \text{skin}|R) \) ?
Learning conditional PDF’s

We can calculate \( P(R | \text{skin}) \) from a set of training images
• It is simply a histogram over the pixels in the training images
  – each bin \( R_i \) contains the proportion of skin pixels with color \( R \).

This doesn’t work as well in higher-dimensional spaces. Why not?

Approach: fit parametric PDF functions
• common choice is rotated Gaussian
  – center \( c \)
  – covariance \( \sum(X - \bar{X})(X - \bar{X})^T \)
  - orientation, size defined by eigenvecs, eigenvals

Bayes rule

\[
P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}
\]

In terms of our problem:
what we measure (likelihood)  \( P(Y) \)
domain knowledge (prior)
\( P(R|\text{skin}) \)
what we want (posterior)

The prior: \( P(\text{skin}) \)
• Could use domain knowledge
  – \( P(\text{skin}) \) may be larger if we know the image contains a person
  – for a portrait, \( P(\text{skin}) \) may be higher for pixels in the center
• Could learn the prior from the training set. How?
  – \( P(\text{skin}) \) could be the proportion of skin pixels in training set

Bayesian estimation

\[
P(R|\text{skin})
\]

\[
P(R|\text{~skin})
\]

Bayesian estimation
• Goal is to choose the label (skin or ~skin) that maximizes the posterior
  – this is called Maximum A Posteriori (MAP) estimation
• Suppose the prior is uniform: \( P(\text{skin}) = P(\text{~skin}) = 0.5 \)
  – In this case, \( P(R|\text{skin}) = P(R|\text{~skin}) \)
  – maximizing the posterior is equivalent to maximizing the likelihood
  – if and only if \( P(R|\text{skin}) > P(R|\text{~skin}) \) if and only if \( P(R|\text{skin}) > P(R|\text{~skin}) \)
  – this is called Maximum Likelihood (ML) estimation
Skin detection results

General classification

This same procedure applies in more general circumstances

- More than two classes
- More than one dimension

Example: face detection

- Here, X is an image region
- dimension = # pixels
- each face can be thought of as a point in a high dimensional space

Linear subspaces

Classification can be expensive

- Must either search (e.g., nearest neighbors) or store large PDF’s

Suppose the data points are arranged as above

- Idea—fit a line, classifier measures distance to line

Dimensionality reduction

How to find $v_1$ and $v_2$?

Dimensionality reduction

- We can represent the orange points with only their $v_1$ coordinates
  - since $v_2$ coordinates are all essentially 0
- This makes it much cheaper to store and compare points
- A bigger deal for higher dimensional problems
Linear subspaces

Consider the variation along direction $v$ among all of the orange points:

$$\text{var}(v) = \sum_{x} [(x - \bar{x})^T v]^2$$

What unit vector $v$ minimizes $\text{var}(v)$?

$$v_1 = \min_v \{ \text{var}(v) \}$$

What unit vector $v$ maximizes $\text{var}(v)$?

$$v_2 = \max_v \{ \text{var}(v) \}$$

Solution: $v_1$ is eigenvector of $A$ with largest eigenvalue

$v_2$ is eigenvector of $A$ with smallest eigenvalue

Principal component analysis

Suppose each data point is N-dimensional

- Same procedure applies:
  $$\text{var}(v) = \sum [(x - \bar{x})^T v]^2$$
  $$= \sum [x^T (x - \bar{x}) (x - \bar{x})^T] v$$
  $$= \sqrt{\sum (x - \bar{x})(x - \bar{x})^T} v$$
  $$= \sqrt{A} v$$

- Eigenvectors of $A$ define a new coordinate system
  - Eigenvector with largest eigenvalue captures the most variation among training vectors $x$
  - Eigenvector with smallest eigenvalue has least variation

- We can compress the data by only using the top few eigenvectors
  - Corresponds to choosing a “linear subspace”
  - Represents points on a line, plane, or “hyper-plane”
  - These eigenvectors are known as the principal components

The space of faces

An image is a point in a high dimensional space

- An $N \times M$ intensity image is a point in $\mathbb{R}^{NM}$
- We can define vectors in this space as we did in the 2D case

Dimensionality reduction

The set of faces is a “subspace” of the set of images

- Suppose it is K dimensional
- We can find the best subspace using PCA
- This is like fitting a “hyper-plane” to the set of faces
  - Spanned by vectors $v_1, v_2, ..., v_k$
  - Any face $x \approx x + \alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_k v_k$
Eigenfaces

PCA extracts the eigenvectors of $A$
- Gives a set of vectors $v_1, v_2, v_3, ...$
- Each one of these vectors is a direction in face space
  - what do these look like?

Projecting onto the eigenfaces

The eigenfaces $v_1, ..., v_K$ span the space of faces
- A face is converted to eigenface coordinates by
  $$x ightarrow (x - \bar{x}) \cdot v_1, (x - \bar{x}) \cdot v_2, ..., (x - \bar{x}) \cdot v_K$$
  $$a_1 \quad a_2 \quad \ldots \quad a_K$$
  $$x \approx \bar{x} + a_1 v_1 + a_2 v_2 + \ldots + a_K v_K$$

Detection and recognition with eigenfaces

Algorithm
1. Process the image database (set of images with labels)
   • Run PCA—compute eigenfaces
   • Calculate the $K$ coefficients for each image
2. Given a new image (to be recognized) $x$, calculate $K$ coefficients
   $$x \rightarrow (a_1, a_2, \ldots, a_K)$$
3. Detect if $x$ is a face
   $$\|x - (\bar{x} + a_1 v_1 + a_2 v_2 + \ldots + a_K v_K)\| < \text{threshold}$$
4. If it is a face, who is it?
   • Find closest labeled face in database
     • nearest-neighbor in $K$-dimensional space

Choosing the dimension $K$

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
Object recognition

This is just the tip of the iceberg

• Better features:
  – edges (e.g., SIFT)
  – motion
  – depth/3D info
  – ...

• Better classifiers:
  – e.g., support vector machines (SVM)

• Speed (e.g., real-time face detection)

• Scale
  – e.g., Internet image search

Recognition is a very active research area right now