Lecture 14: Introduction to Recognition
Announcements

• Final project page up, at
  – One person from each team should submit a proposal (to CMS) by next Wednesday at 11:59pm

• Project 3: eigenfaces
  – will be posted on the web soon
  – Adarsh will capture photos at the end of class
  – project will include a challenge competition
What do we mean by “object recognition”?

Next 15 slides adapted from Li, Fergus, & Torralba’s excellent short course on category and object recognition
Verification: is that a lamp?
Detection: are there people?
Identification: is that Potala Palace?
Object categorization

- mountain
- tree
- banner
- building
- street lamp
- vendor
- people
Scene and context categorization

- outdoor
- city
- ...

[Image of a cityscape with the Potala Palace in the background]
Object recognition
Is it really so hard?

Find the chair in this image

Output of normalized correlation

This is a chair
Object recognition

Is it really so hard?

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?

Find the chair in this image

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.” Nivatia & Binford, 1977.
And it can get a lot harder
Applications: Computational photography

[Face priority AE] When a bright part of the face is too bright
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

Refine your image search with visual similarity

Similar Images allows you to search for images using pictures rather than words. Click the "Similar images" link under an image to find other images that look like it. Try a search of your own or click on an example below.

- **paris**
  - ![Similar images](image)
  - ![Similar images](image)
  - ![Similar images](image)
  - ![Similar images](image)

- **temple**
  - ![Similar images](image)
  - ![Similar images](image)
  - ![Similar images](image)
  - ![Similar images](image)

Places
- London
- New York
- Egypt
- Forbidden City

Celebrities
- Michael Jordan
- Angelina Jolie
- Halle Berry
- Seth Rogen
- Rihanna

Art
- impressionism
- Keith Haring
- cubism
- Salvador Dali
- pointillism

Shopping
- evening gown
- necklace
- shoes
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
Observation 3:
Observation 3:

- Negative contrast is difficult
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

\[
P(X)
\]

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: \( P(X \mid Y) \)
  - 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 \( R_1 < X \leq R_2 \)

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

We can calculate \( P(R \mid \text{skin}) \) 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 \( R_i \)

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

Approach: fit parametric PDF functions

- common choice is rotated Gaussian
  - center \( c = \bar{X} \)
  - covariance \( \sum_X (X - \bar{X})(X - \bar{X})^T \)
  - orientation, size defined by eigenvectors, eigenvalues
Learning conditional PDF’s

We can calculate \( P(R \mid \text{skin}) \) 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 \( R_i \)

But this isn’t quite what we want

- Why not? How to determine if a pixel is skin?
- We want \( P(\text{skin} \mid R) \), not \( P(R \mid \text{skin}) \)
- How can we get it?
Bayes rule

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

In terms of our problem:

- what we measure (likelihood)
- domain knowledge (prior)

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

normalization term

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

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

- 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: \( P(\text{skin}) = P(\sim \text{skin}) = 0.5 \)
  - in this case \( P(\text{skin}|R) = cP(R|\text{skin}), \quad P(\sim \text{skin}|R) = cP(R|\sim \text{skin}) \)
  - maximizing the posterior is equivalent to maximizing the likelihood
    - \( P(\text{skin}|R) > P(\sim \text{skin}|R) \) if and only if \( P(R|\text{skin}) > P(R|\sim \text{skin}) \)
  - this is called Maximum Likelihood (ML) estimation
Skin detection results

Figure 25.3. The figure shows a variety of images together with the output of the skin detector of Jones and Rehg applied to the image. Pixels marked black are skin pixels, and white are background. Notice that this process is relatively effective, and could certainly be used to focus attention on, say, faces and hands. Figure from “Statistical color models with application to skin detection,” M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 © 1999, IEEE
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

convert \( x \) into \( v_1, v_2 \) coordinates

\[ x \rightarrow ((x - \bar{x}) \cdot v_1, (x - \bar{x}) \cdot v_2) \]

What does the \( v_2 \) coordinate measure?
- distance to line
- use it for classification—near 0 for orange pts

What does the \( v_1 \) coordinate measure?
- position along line
- use it to specify which orange point it is