Previously

• Intro to generic object recognition
• Supervised classification
  – Main idea
  – Skin color detection example

Last time:
Example: skin color classification

• We can represent a class-conditional density using a histogram (a “non-parametric” distribution)
Last time:
Example: skin color classification

- We can represent a class-conditional density using a histogram (a "non-parametric" distribution)

\[ P(x|\text{skin}) \propto P(x|\text{skin})P(\text{skin}) \]

Now we get a new image, and want to label each pixel as skin or non-skin.

Classify pixels based on these probabilities
- if \( p(\text{skin}|x) > \theta \), classify as skin
- if \( p(\text{skin}|x) < \theta \), classify as not skin

Today

- Window-based generic object detection
  - basic pipeline
  - boosting classifiers
  - face detection as case study
Generic category recognition: 
basic framework

- Build/train object model
  - Choose a representation
  - Learn or fit parameters of model / classifier
- Generate candidates in new image
- Score the candidates

---

Generic category recognition: 
representation choice

Window-based Part-based

---

Window-based models
Building an object model

Simple holistic descriptions of image content
- grayscale / color histogram
- vector of pixel intensities

Kristen Grauman
Window-based models
Building an object model

• Pixel-based representations sensitive to small shifts

• Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Window-based models
Building an object model

• Consider edges, contours, and (oriented) intensity gradients

• Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Contrast-normalization: try to correct for variable illumination
Window-based models
Building an object model

Given the representation, train a binary classifier

Car/non-car Classifier

Yes, car.

No, not a car.

Discriminative classifier construction

Nearest neighbor
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks
LaCun, Bottou, Bengio, Haftner 1998
Rowley, Baluja, Kanade 1998

Support Vector Machines
Guyon, Vapnik
Heisele, Same, Poggio, 2001,...

Boosting
Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,...

Conditional Random Fields
McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003

Generic category recognition:
basic framework

• Build/train object model
  – Choose a representation
  – Learn or fit parameters of model / classifier
• Generate candidates in new image
• Score the candidates
Window-based models
Generating and scoring candidates

![Image of car/non-car classifier]

Kristen Grauman

Window-based object detection: recap
Training:
1. Obtain training data
2. Define features
3. Define classifier

Given new image:
1. Slide window
2. Score by classifier

![Image of training examples and classifier]

Kristen Grauman

Discriminative classifier construction

- Nearest neighbor
  - Shaarhoovitch, Viola, Darrell 2003
  - Berg, Berg, Malik 2005

- Neural networks
  - LeCun, Bottou, Bengio, Haflner 1998
  - Rowley, Baluja, Kanade 1998

- Support Vector Machines
  - Guyon, Vapnik, Heisele, Serre, Poggio, 2001

- Boosting
  - Viola, Jones 2001
  - Torralba et al. 2004
  - Opelt et al. 2006

- Conditional Random Fields
  - McCallum, Freitag, Pereira 2000
  - Kumar, Hebert 2003

Slide adapted from Antonio Torralba
Boosting illustration

Weights Increased

Boosting illustration

Weak Classifier 3

Boosting illustration

Final classifier is a combination of weak classifiers
Boosting: training

- Initially, weight each training example equally
- In each boosting round:
  - Find the weak learner that achieves the lowest weighted training error
  - Raise weights of training examples misclassified by current weak learner
- Compute final classifier as linear combination of all weak learners (weight of each learner is directly proportional to its accuracy)
- Exact formulas for re-weighting and combining weak learners depend on the particular boosting scheme (e.g., AdaBoost)

Boosting: pros and cons

- Advantages of boosting
  - Integrates classification with feature selection
  - Flexibility in the choice of weak learners, boosting scheme
  - Testing is fast
  - Easy to implement

- Disadvantages
  - Needs many training examples
  - Often found not to work as well as an alternative discriminative classifier, support vector machine (SVM)
    - especially for many-class problems

Viola-Jones face detector
Viola-Jones face detector

Main idea:
- Represent local texture with efficiently computable “rectangular” features within window of interest
- Select discriminative features to be weak classifiers
- Use boosted combination of them as final classifier
- Form a cascade of such classifiers, rejecting clear negatives quickly

Viola-Jones detector: features

“Rectangular” filters
Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time.

Computing sum within a rectangle
- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:
  \[ \text{sum} = A - B - C + D \]
- Only 3 additions are required for any size of rectangle!
Viola-Jones detector: features

“Rectangular” filters
Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time

Value at \((x,y)\) is sum of pixels above and to the left of \((x,y)\)

Integral image

Kristen Grauman

Viola-Jones detector: features

Considering all possible filter parameters: position, scale, and type:
180,000+ possible features associated with each 24 x 24 window

Which subset of these features should we use to determine if a window has a face?
Use AdaBoost both to select the informative features and to form the classifier

Viola-Jones detector: AdaBoost

Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of weighted error.

Resulting weak classifier:

\[
h(x) = \begin{cases} +1 & \text{if } f(x) > 0 \\ -1 & \text{otherwise} \end{cases}
\]

For next round, reweight the examples according to errors, choose another filter/threshold combo.
Perceptual and Sensory Augmented Computing

Visual Object Recognition Tutorial

AdaBoost Algorithm

Start with uniform weights on training examples.

For T rounds:

- Evaluate weighted error for each feature, pick best.
- Re-weight the examples:
  - Incorrectly classified → more weight
  - Correctly classified → less weight

Final classifier is combination of the weak ones, weighted according to error they had.

Freund & Schapire 1995

Viola-Jones Face Detector: Results

First two features selected

• Even if the filters are fast to compute, each new image has a lot of possible windows to search.
• How to make the detection more efficient?
Cascading classifiers for detection

- Form a cascade with low false negative rates early on
- Apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative

Viola-Jones detector: summary

- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in all layers

[Implementation available in OpenCV:
http://www.intel.com/technology/computing/opencv/]

Viola-Jones detector: summary

- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - Integral images for fast feature evaluation
  - Boosting for feature selection
  - Attentional cascade of classifiers for fast rejection of non-face windows

P. Viola and M. Jones. Robust real-time face detection, IJCV 57(2), 2004.
Detecting profile faces?

Can we use the same detector?

Viola-Jones Face Detector: Results

Example using Viola-Jones detector

Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Consumer application: iPhoto 2009

http://www.apple.com/ilife/iphoto/

Things iPhoto thinks are faces
Consumer application: iPhoto 2009
Can be trained to recognize pets!


What other categories are amenable to window-based representation?

Pedestrian detection
- Detecting upright, walking humans also possible using sliding window’s appearance/texture; e.g.,

 SVM with Haar wavelets [Papageorgiou & Poggio, ICCV 2000]
 Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]
 SVM with HaGc [Dalal & Triggs, CVPR 2005]
Window-based detection: strengths

- Sliding window detection and global appearance descriptors:
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes

Window-based detection: Limitations

- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

Limitations (continued)

- Not all objects are “box” shaped
Limitations (continued)

• Non-rigid, deformable objects not captured well with representations assuming a fixed 2d structure; or must assume fixed viewpoint

Limitations (continued)

• If considering windows in isolation, context is lost

Limitations (continued)

• In practice, often entails large, cropped training set (expensive)
• Requiring good match to a global appearance description can lead to sensitivity to partial occlusions
Summary

• Basic pipeline for window-based detection
  – Model/representation/classifier choice
  – Sliding window and classifier scoring
• Boosting classifiers: general idea
• Viola-Jones face detector
  – Exemplar of basic paradigm
  – Plus key ideas: rectangular features, Adaboost for feature selection, cascade
• Pros and cons of window-based detection

Questions?

See you Tuesday!