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)

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

\[ P(skin \mid x) \propto P(x \mid skin)P(skin) \]

Now for every pixel in a new image, we can estimate probability that it is generated by skin.

Classify pixels based on these probabilities
• if \( p(skin \mid x) > \theta \), classify as skin
• if \( p(skin \mid x) < \theta \), 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

Window-based models
Building an object model

Given the representation, train a binary classifier

Car/non-car Classifier

No/some car.

Window-based models
Generating and scoring candidates
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

Discriminative classifier construction

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

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

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

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

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

Weights Increased

Weak Classifier 1

Boosting illustration

Weak Classifier 2
Boosting illustration

Weights Increased

Boosting illustration

Weak Classifier 3

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)

Viola-Jones face detector

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola  
vio@research.com  
Mitsubishi Electric Research Labs  
201 Broadway, 8th FL  
Cambridge, MA 02190

Michael Jones  
mjones@dec.com  
Compaq  
One Cambridge Center  
Cambridge, MA 02142

Abstract
This paper describes a machine learning approach for detecting faces. It employs a cascade of simple features, which can be rapidly computed and combined.
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.

“Rectangular” filters
Value at (x,y) is sum of pixels above and to the left of (x,y)

Integral image

Kristen Grauman

Computing the integral image

Lana Lazebnik

Computing the integral image

Lana Lazebnik

• Cumulative row sum: \( s(x, y) = s(x-1, y) + i(x, y) \)
• Integral image: \( ii(x, y) = ii(x, y-1) + s(x, y) \)
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

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:

\[ h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases} \]

For next round, reweight the examples according to errors, choose another filter/threshold combo.

---

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.

---

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 (high recall) 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 cascade of classifiers with AdaBoost

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/]

Kristen Grauman
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


Viola-Jones Face Detector: Results

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.

Everingham, M., Sivic, J. and Zisserman, A.

"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006.  
http://www.robots.ox.ac.uk/~vgg/research/nface/index.html

---

**Google street view blurs face of cow to protect its identity**

---

---

---

---
Consumer application: iPhoto 2009

Things iPhoto thinks are faces

Can be trained to recognize pets!
Privacy Gift Shop – CV Dazzle

http://www.wired.com/2015/06/facebook-can-recognize-even-dont-show-face/
Wired, June 15, 2015
Slide: Kristen Grauman

Privacy Visor

Slide: Kristen Grauman

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
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, IJCV 2000]
  - Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]
  - SVM with HoGs [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)

- If considering windows in isolation, context is lost

Figure credit: Derek Hoiem
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 Thursday!