Object detection as supervised classification

Thurs April 13
Kristen Grauman
UT Austin

Last time

• Discovering visual patterns
  • Randomized hashing algorithms
  • Mining large-scale image collections

Review questions: on your own

• What kind of input data is searchable with min-hash hashing?
• What kind of input data is searchable with LSH using random projections?
• For Visual “PageRank” what do weights between nodes (images) signify?
What does recognition involve?

Detection: are there people?

Activity: What are they doing?
Attribute recognition

crowded  flat

made of  fabric

gray

Object Categorization

- Task Description
  - “Given a small number of training images of a category, recognize a-priori unknown instances of that category and assign the correct category label.”

- Which categories are feasible visually?

<table>
<thead>
<tr>
<th>“Fido”</th>
<th>German Shepherd</th>
<th>dog</th>
<th>animal</th>
<th>living being</th>
</tr>
</thead>
</table>

Visual Object Categories

- Basic Level Categories in human categorization [Rosch 76, Lakoff 87]
  - The highest level at which category members have similar perceived shape
  - The highest level at which a single mental image reflects the entire category
  - The level at which human subjects are usually fastest at identifying category members
  - The first level named and understood by children
  - The highest level at which a person uses similar motor actions for interaction with category members
Visual Object Categories

- Basic-level categories in humans seem to be defined predominantly visually.
- There is evidence that humans (usually) start with basic-level categorization before doing identification.

- Basic-level categorization is easier and faster for humans than object identification!
- How does this transfer to automatic classification algorithms?

How many object categories are there?

~10,000 to 30,000

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

Bladerman 1987
Other Types of Categories

- Functional Categories
  - e.g., chairs = “something you can sit on”

Why recognition?

- Recognition a fundamental part of perception
  - e.g., robots, autonomous agents
- Organize and give access to visual content
  - Connect to information
  - Detect trends and themes

Autonomous agents able to detect objects
Posing visual queries

Yeh et al., MIT

Belhumeur et al.

Finding visually similar objects

Finding visually similar objects

Exploring community photo collections

Simon & Szeliski
Discovering visual patterns

Objects

Categories

Auto-annotation

Challenges: robustness

Illumination

Object pose

Clutter

Occlusions

Intra-class appearance

Viewpoint

Slide: Kristen Grauman
Challenges: context and human experience

Context cues

Function

Dynamics

Challenges: complexity

• Millions of pixels in an image
• 30,000 human recognizable object categories
• 30+ degrees of freedom in the pose of articulated objects (humans)
• Billions of images online
• 82 years to watch all videos uploaded to YouTube per day!
• About half of the cerebral cortex in primates is devoted to processing visual information [Felleman and van Essen 1991]
Challenges: learning with minimal supervision

Less

More

Unlabeled, multiple objects

Class labeled, same objects

Grasp and objects, unlabeled, classes

Slide: Kristen Grauman
Recognizing flat, textured objects (like books, CD covers, posters)

Reading license plates, zip codes, checks

Frontal face detection

Fingerprint recognition

What kinds of things work best today?

What kinds of things work best today?

Progress charted by datasets

ROBERTS 1963

COIL 1996
Evolution of methods

- Hand-crafted models
- 3D geometry
- Hypothesize and align
- Hand-crafted features
- Learned models
- Data-driven
- "End-to-end" learning of features and models*, **

* Labeled data availability
** Architecture design decisions, parameters.

Next

- Supervised classification
- Window-based generic object detection
  - basic pipeline
  - boosting classifiers
  - face detection as case study

Supervised classification

- Given a collection of labeled examples, come up with a function that will predict the labels of new examples.

  "four"  
  "nine"  

  Training examples  Novel input

- How good is some function we come up with to do the classification?
- Depends on
  - Mistakes made
  - Cost associated with the mistakes
Supervised classification

• Given a collection of labeled examples, come up with a function that will predict the labels of new examples.

• Consider the two-class (binary) decision problem
  – \( L(4 \rightarrow 9) \): Loss of classifying a 4 as a 9
  – \( L(9 \rightarrow 4) \): Loss of classifying a 9 as a 4

• **Risk** of a classifier \( s \) is expected loss:
  \[
  R(s) = \Pr(4 \rightarrow 9 \mid s) L(4 \rightarrow 9) + \Pr(9 \rightarrow 4 \mid s) L(9 \rightarrow 4)
  \]

• We want to choose a classifier so as to minimize this total risk.

---

Supervised classification

Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

If we choose class “four” at boundary, expected loss is:
\[
= P(\text{class is 9} \mid x) L(9 \rightarrow 4) + P(\text{class is 4} \mid x) L(4 \rightarrow 4)
\]

If we choose class “nine” at boundary, expected loss is:
\[
= P(\text{class is 4} \mid x) L(4 \rightarrow 9)
\]

So, best decision boundary is at point \( x \) where
\[
P(\text{class is 9} \mid x) L(9 \rightarrow 4) = P(\text{class is 4} \mid x) L(4 \rightarrow 4)
\]

To classify a new point, choose class with lowest expected loss; i.e., choose “four” if
\[
P(4 \mid x) L(4 \rightarrow 9) > P(9 \mid x) L(9 \rightarrow 4)
\]
Supervised classification

Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

So, best decision boundary is at point \( x \) where

\[
P(\text{class is } 9 \mid x) L(9 \rightarrow 4) = P(\text{class is } 4 \mid x) L(4 \rightarrow 9)
\]

To classify a new point, choose class with lowest expected loss; i.e., choose “four” if

\[
P(4 \mid x) L(4 \rightarrow 9) > P(9 \mid x) L(9 \rightarrow 4)
\]

Example: learning skin colors

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.

What’s the probability we care about to do skin detection?
Bayes rule

\[
P(skin \mid x) = \frac{P(x \mid skin)P(skin)}{P(x)}
\]

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

Where does the prior come from?
Why use a prior?

Example: classifying skin pixels

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 \( p(skin|x) > 0 \), classify as skin
- if \( p(skin|x) < 0 \), classify as not skin

Supervised classification

- Want to minimize the expected misclassification
- Two general strategies
  - Use the training data to build representative probability model: separately model class-conditional densities and priors (generative)
  - Directly construct a good decision boundary, model the posterior (discriminative)
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

http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/hws/www/CVPR00.pdf

Today

• Supervised classification
• 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](image)

Yes, car.

No, not a car.

---

**Window-based models**

**Generating and scoring candidates**

![Car/non-car Classifier](image)

---

**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

![Feature extraction](image)

![Car/non-car Classifier](image)

---
Discriminative classifier construction

- Nearest neighbor
- Neural networks
- Support Vector Machines
- Boosting
- Conditional Random Fields

Boosting intuition

Weights Increased

Boosting illustration
Boosting illustration

Weak Classifier 2

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)

Viola-Jones face detector

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
Mitsubishi Electric Research Labs
201 Broadway, 4th FL
Cambridge, MA 02139

Michael Jones
mjjones@csail.mit.edu
Cambridge, MA

Abstract

This paper describes a machine learning approach for...
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 the integral image
Computing the integral image

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

Integral image: \( ii(x, y) = ii(x, y-1) + s(x, y) \)

Calculating 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
- Avoid scaling images \( \rightarrow \) scale features directly for same cost
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:

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

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.

Freund & Schapire 1995

Viola-Jones detector: features
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
Training the cascade

- Set target detection and false positive rates for each stage
- Keep adding features to the current stage until its target rates have been met
  - Need to lower AdaBoost threshold to maximize detection (as opposed to minimizing total classification error)
- If the overall false positive rate is not low enough, then add another stage
- Use false positives from current stage as the negative training examples for the next stage

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]

Viola-Jones detector: summary

- A seminal approach to real-time object detection
  - 15,000 citations and counting
- 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.
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.

Google street view blurs face of cow to protect its identity

Consumer application: iPhoto

http://www.apple.com/ilife/iphoto/
Consumer application: iPhoto

Things iPhoto thinks are faces


Privacy Gift Shop – CV Dazzle

http://www.wired.com/2015/06/facebook-can-recognize-even-dont-show-face/

Wired, June 15, 2015

Slide credit: Kristen Grauman
Privacy Visor


Slide: Kristen Grauman

Boosting: pros and cons

• Advantages of boosting
  • Integartes classification with feature selection
  • Complexity of training is linear in the number of training examples
  • Flexibility in the choice of weak learners, boosting scheme
  • Testing is fast
  • Easy to implement

• Disadvantages
  • Needs many training examples
  • Other discriminative models may outperform in practice (SVMs, CNNs,…)
    – especially for many-class problems

Slide credit: Lana Lazebnik

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
- Objects with less-regular textures not captured well with holistic appearance-based descriptions
Limitations (continued)

• If considering windows in isolation, context is lost

Figure credit: Derek Hoiem
Slide: Kristen Grauman

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

Image credit: Galen, Hoiem, & Shridhar
Slide: Kristen Grauman

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