Window based detectors

CS 554 – Computer Vision
Pinar Duygulu
Bilkent University

(Source: James Hays, Brown)
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
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
Window-based models
Building an object model

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

- Locally orderless: offers invariance to small shifts and rotations
- Contrast-normalization: try to correct for variable illumination

• Summarize local distribution of gradients with histogram

Kristen Grauman
Window-based models
Building an object model

Given the representation, train a binary classifier

Kristen Grauman
Discriminative classifier construction

**Nearest neighbor**

10^6 examples

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

**Support Vector Machines**

Guyon, Vapnik
Heisele, Serre, Poggio, 2001,...

**Neural networks**

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
...

**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
Influential Works in Detection

- Sung-Poggio (1994, 1998) : ~1450 citations
  - Basic idea of statistical template detection (I think), bootstrapping to get “face-like” negative examples, multiple whole-face prototypes (in 1994)
  - “Parts” at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
  - Careful feature engineering, excellent results, cascade
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
  - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-Huttenlocher (2000): ~800
  - Efficient way to solve part-based detectors
  - Excellent template/parts-based blend
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

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
Discriminative classifier construction

Nearest neighbor

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

10^6 examples

Support Vector Machines

Guyon, Vapnik
Heisele, Serre, Poggio, 2001,...

Neural networks

LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
...

Boosting

Viola, Jones 2001, Torralba
McCallum, Freitag, Pereira
et al. 2004, Opelt et al.
2000; Kumar, Hebert 2003
...

Conditional Random Fields

Slide adapted from Antonio Torralba
Boosting intuition

Weak Classifier 1
Boosting illustration

Weights Increased
Boosting illustration

Weak Classifier 2
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
  – 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
Viola-Jones face detector

Accepted Conference on Computer Vision and Pattern Recognition 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

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

Abstract

This paper describes a machine learning approach for vi-

tected at 15 frames per second on a conventional 700 MHz
Intel Pentium III. In other face detection systems, auxiliary
information, such as image differences in video sequences,
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

Avoid scaling images → 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.

Outputs of a possible rectangle feature on faces and non-faces.

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

1. Start with uniform weights on training examples.
2. For $t = 1, \ldots, T$:
   1. Normalize the weights,
      $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_j w_{t,j}}$
      so that $w_t$ is a probability distribution.
   2. For each feature, $j$, train a classifier $h_j$ which is restricted to using a single feature. The error is evaluated with respect to $w_t$, $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
   3. Choose the classifier, $h_t$, with the lowest error $\epsilon_t$.
   4. Update the weights:
      $w_{t+1,i} = w_{t,i} \beta_t^{1-\epsilon_i}$
      where $\epsilon_i = 0$ if example $x_i$ is classified correctly, $\epsilon_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_i}{1-\epsilon_t}$.
3. The final strong classifier is:
   $$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
   where $\alpha_t = \log \frac{1}{\beta_t}$.

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
  - Features which can be evaluated very quickly with *Integral Images*
  - Cascade model which rejects unlikely faces quickly
  - Mining hard negatives


Viola-Jones Face Detector: Results
Viola-Jones Face Detector: Results
Viola-Jones Face Detector: Results
Detecting profile faces?

*Can we use the same detector?*
Viola-Jones Face Detector: Results
Viola Jones Results

<table>
<thead>
<tr>
<th>Detector</th>
<th>10</th>
<th>31</th>
<th>50</th>
<th>65</th>
<th>78</th>
<th>95</th>
<th>167</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viola-Jones</td>
<td>76.1%</td>
<td>88.4%</td>
<td>91.4%</td>
<td>92.0%</td>
<td>92.1%</td>
<td>92.9%</td>
<td>93.9%</td>
</tr>
<tr>
<td>Viola-Jones (voting)</td>
<td>81.1%</td>
<td>89.7%</td>
<td>92.1%</td>
<td>93.1%</td>
<td>93.1%</td>
<td>93.2%</td>
<td>93.7%</td>
</tr>
<tr>
<td>Rowley-Baluja-Kanade</td>
<td>83.2%</td>
<td>86.0%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>89.2%</td>
<td>90.1%</td>
</tr>
<tr>
<td>Schneiderman-Kanade</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>94.4%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Roth-Yang-Ahuja</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>(94.8%)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

MIT + CMU face dataset

Slide: Derek Hoiem
Schneiderman later results

<table>
<thead>
<tr>
<th></th>
<th>89.7%</th>
<th>93.1%</th>
<th>94.4%</th>
<th>94.8%</th>
<th>95.7%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian Network</td>
<td>1</td>
<td>8</td>
<td>19</td>
<td>36</td>
<td>56</td>
</tr>
<tr>
<td>Semi-Naive Bayes*</td>
<td>6</td>
<td>19</td>
<td>29</td>
<td>35</td>
<td>46</td>
</tr>
<tr>
<td>[6]</td>
<td>31</td>
<td>65</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>[7]*</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>78</td>
<td>--</td>
</tr>
<tr>
<td>[16]*</td>
<td>--</td>
<td>--</td>
<td>65</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 2. False alarms as a function of recognition rate on the MIT-CMU Test Set for Frontal Face Detection. * indicates exclusion of the 5 images of hand-drawn faces.
Speed: frontal face detector

- Schneiderman-Kanade (2000): 5 seconds
- Viola-Jones (2001): 15 fps
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 now erases faces, license plates on Map Street View

By Elinor Mills, CNET News.com
Friday, August 24, 2007 01:37 PM

Google has gotten a lot of flack from privacy advocates for photographing faces and license plate numbers and displaying them on the Street View in Google Maps. Originally, the company said only people who identified themselves could ask the company to remove their image.

But Google has quietly changed that policy, partly in response to criticism, and now anyone can alert the company and have an image of a license plate or a recognizable face removed, not just the owner of the face or car, says Marissa Mayer, vice president of search products and user experience at Google.

"It's a good policy for users and also clarifies the intent of the product," she said in an interview following her keynote at the Search Engine Strategies conference in San Jose, Calif., Wednesday.

The policy change was made about 10 days after the launch of the product in late May, but was not publicly announced, according to Mayer. The company is removing images only when someone notifies them and not proactively, she said. "It was definitely a big policy change inside."
Consumer application: iPhoto 2009

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

Slide credit: Lana Lazebnik
Consumer application: iPhoto 2009

- Things iPhoto thinks are faces
Consumer application: iPhoto 2009

- Can be trained to recognize pets!