Face detection using a boosted cascade of simple features
Face detection using a boosted cascade of simple features

Lecturer: Domenico Daniele Bloisi
Schedule: 1/7/2014 9:30 - Via Ariosto 25, room A6
Abstract: When we take a picture with our smartphone is normal to see a bounding box around the captured faces. The same when we tag a friend on Facebook. Actually, to detect a face in an image involves the use of multiple methods and algorithms. If the problem of face detection can be considered solved, it is thanks to the work of Paul Viola and Michael Jones. In this lecture we will go through the solutions used in the most famous face detection method. In particular, how to use the integral image representation for fast feature extraction, how to select the most discriminative features by using an Adaptive Boosting approach for face detection, and how to speed up the recognition process thanks to a cascade of weak classifiers for fast rejection of non-face sub-windows.
Welcome to my homepage!

I am a fixed-term assistant professor with the Department of Computer, Control, and Management Engineering at Sapienza University of Rome. I am a member of RoCoCo laboratory, held by Prof. Daniele Nardi.

My academic interests include intelligent surveillance, multi-sensor data fusion, image processing, and steganography.

I received my PhD in Computer Science Engineering on February the 3rd, 2010. The title of my PhD thesis is "Visual Tracking and Data Fusion for Automatic Video Surveillance" and my advisor was Prof. Luca Iocchi.

From February the 1st, 2008 to September the 15th, 2008 I was a visiting scholar at Digital Imaging Research Centre (DIRC), Faculty of Computing, Information Systems and Mathematics (CISM), Kingston University, London. My supervisor was Prof. Paolo Remagnino.

From April the 1st, 2009 to March the 31st, 2010 I was an associate researcher with the VIPS lab, University of Verona.
Cooperating Cognitive Robots
RoCoCo

Disaster Response Robots

Boat tracking in Venice

Vessel traffic monitoring

Face detection using a boosted cascade of simple features

July, the 1st 2014

Page 4
Outline

• Classification-based detection
• Learning a Classifier
• Combining Multiple Learners
• Cascading
• Examples
• OpenCV code
• Build your own classifier!
Some slides of this presentation adapted from

- KH Wong. “Ch. 6: Face detection”
- P. Viola and T.-W. Yue. “Adaboost for Face Detection”
- S. Lazebnik. “Face detection”
- C. Schmid. “Category-level localization”
- C. Huang and F. Vahid. “Scalable Object Detection Accelerators on FPGAs Using Custom Design Space Exploration”
- P. Smyth. “Face Detection using the Viola-Jones Method”
- K. Palla and A. Kalaitzis. “Robust Real-time Face Detection”
- P. Viola and M. Jones. “Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade”
- D. Hoiem. “Adaboost”
- H. Wen-Chung. “Introduction to AdaBoost”
Face Detection Problem

Given an image, find regions in the image which contain instances of faces.
Detection Issues

TP
True Positive

FN
False Negative

FP
False Positive
Additional Issues

- Rotation
- Blurring
- Illumination
- Occlusion
- Glasses
- ...

Face detection using a boosted cascade of simple features
General Problem

• Given an image, find a specific object in it.
Detection is not Identification

Detection

identification

PETS 2009 dataset
http://www.cvg.rdg.ac.uk/PETS2009
Detection is not Recognition

**Detection**

- Face 1
- Face 2
- Face 3

**Recognition**

- F. Engels
- K. Marx
- Me
Detection vs Recognition
Classification-based Detection

The output is a bounding box around the instance(s) belonging to the class of objects for which the classifier has been trained.
Basic Idea

Slide a window (e.g., 30x30) across the image and evaluate the current portion of the image w.r.t. an object model at every location.

The number of locations where the object is present is very very small.
Multiple Scales

Face detection using a boosted cascade of simple features

July, the 1st 2014
Scan window over image pyramid

Face detection using a boosted cascade of simple features
Object Detection

Machine Learning techniques can be used to deal with the object detection problem.

1. **Feature Computation**
   - What features?
   - How can they be computed as quickly as possible?

2. **Feature Selection**
   - What are the most discriminating features?

3. **Detection in real time**
   - Must focus on potentially positive areas.

Face detection using a boosted cascade of simple features

July, the 1st 2014
The Viola and Jones Method

- The most famous method
- Recognition is very fast
  (e.g., real-time for digital cameras)
- Key contributions
  1. Integral image for fast feature extraction
  2. Boosting (Ada-Boost) for face detection
  3. Attentional cascade for fast rejection of non-face sub-windows

Key Contributions

1. **Quick Feature Computation**
   - Rectangle features
   - Integral image representation

2. **Simple and Efficient Classification**
   - The Ada-Boost training algorithm

3. **Real-timeliness**
   - A cascade of classifiers
Features

Four basic types

– Easy to calculate.
– White areas are subtracted from the black ones.
– Integral image representation makes feature extraction faster.
Definition of Simple Features

Face detection using a boosted cascade of simple features
Rectangle Features

\[ \text{Value} = \sum (\text{pixels in white area}) - \sum (\text{pixels in black area}) \]
Rectangle Features

Value = \( \sum \) (pixels in white area) – \( \sum \) (pixels in black area)
The motivation behind using rectangular features, as opposed to more expressive steerable filters, is due to their extreme computational efficiency.
Integral image

- an intermediate representation of the image for rapid calculation of rectangle features

The *integral image* computes a value at each pixel \((x,y)\) that is the sum of the pixel values above and to the left of \((x,y)\), inclusive.
Integral image for feature extraction
Computing the integral image

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

### IMAGE

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### INTEGRAL IMAGE

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>16</td>
<td>21</td>
</tr>
</tbody>
</table>
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!
Feature Selection

For a 24x24 detection region, the number of possible rectangle features is ~160,000!
Feature selection

At test time, it is impractical to evaluate the entire feature set.

We want a subset of relevant features, which are informative to model a face.

Can we create a good classifier using just a small subset of all possible features?

How to select such a subset?
Combining Multiple Learners

- No Free Lunch thm: There is no algorithm that is always the most accurate
- Generate a group of base-learners which when combined has higher accuracy
- Different learners use different
  - Algorithms
  - Hyperparameters
  - Representations (Modalities)
  - Training sets
  - Subproblems
**Boosting (Schapire 1989)**

- Randomly select $n_1 < n$ samples from $D$ without replacement to obtain $D_1$
  - Train weak learner $C_1$

- Select $n_2 < n$ samples from $D$ with half of the samples misclassified by $C_1$ to obtain $D_2$
  - Train weak learner $C_2$

- Select all samples from $D$ that $C_1$ and $C_2$ disagree on
  - Train weak learner $C_3$

- Final classifier is vote of weak learners
Adaboost - Adaptive Boosting

• Instead of sampling, re-weight
  – Previous weak learner has only 50% accuracy over new distribution

• Can be used to learn weak classifiers

• Final classification based on weighted vote of weak classifiers
Boosting constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.

\[ F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots \]
Voting

- Linear combination
  \[ y = \sum_{j=1}^{L} w_j d_j \]
  \[ w_j \geq 0 \quad \text{and} \quad \sum_{j=1}^{L} w_j = 1 \]

- Classification
  \[ y_i = \sum_{j=1}^{L} w_j d_{ji} \]
AdaBoost is an algorithm for constructing a “strong” classifier as linear combination

\[ f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \]

of “simple” “weak” classifiers \( h_t(x) \).

**Terminology**

- \( h_t(x) \) ... “weak” or basis classifier, hypothesis, ”feature"
- \( H(x) = \text{sign}(f(x)) \) ... “strong” or final classifier/hypothesis

**Comments**

- The \( h_t(x) \)’s can be thought of as features.
- Often (typically) the set \( \mathcal{H} = \{ h(x) \} \) is infinite.
Boosting Example

Training data

$D_1$
First round

1st weak classifier

\[ \varepsilon_1 = 0.30 \]
\[ \alpha_1 = 0.42 \]

Error
Face detection using a boosted cascade of simple features

Second round

$\epsilon_2 = 0.21$
$\alpha_2 = 0.65$

1st weak classifier

2nd weak classifier
Third round

1st weak classifier

2nd weak classifier

3rd weak classifier

$\varepsilon_3 = 0.14$

$\alpha_3 = 0.92$
Weighted combination

\[ f = \left( \begin{array}{c} 0.42 \\ + 0.65 \\ + 0.92 \end{array} \right) \]

\[ f / (0.42 + 0.65 + 0.92) \]

\[ = \]
Final Classifier learned by Boosting

\[ H_{\text{final}} = \text{sign} \left( \begin{array}{c} 0.42 \\ + 0.65 \\ + 0.92 \end{array} \right) \]
Adaptive Boosting

**AdaBoost**

Generate a sequence of base-learners each focusing on previous one’s errors (Freund and Schapire, 1996)

**Training:**
- For all \( \{x^t, r^t\}_{t=1}^N \in \mathcal{X} \), initialize \( p_1^t = 1/N \)
- For all base-learners \( j = 1, \ldots, L \)
  - Randomly draw \( \mathcal{X}_j \) from \( \mathcal{X} \) with probabilities \( p_j^t \)
  - Train \( d_j \) using \( \mathcal{X}_j \)
  - For each \( (x^t, r^t) \), calculate \( y_j^t \leftarrow d_j(x^t) \)
  - Calculate error rate: \( \epsilon_j \leftarrow \sum_t p_j^t \cdot 1(y_j^t \neq r^t) \)
  - If \( \epsilon_j > 1/2 \), then \( L \leftarrow j - 1; \) stop
  - \( \beta_j \leftarrow \epsilon_j / (1 - \epsilon_j) \)
  - For each \( (x^t, r^t) \), decrease probabilities if correct:
    - If \( y_j^t = r^t \) \( p_j^{t+1} \leftarrow \beta_j p_j^t \) Else \( p_j^{t+1} \leftarrow p_j^t \)
- Normalize probabilities:
  - \( Z_j \leftarrow \sum_t p_j^{t+1} \); \( p_j^{t+1} \leftarrow p_j^{t+1} / Z_j \)

**Testing:**
- Given \( x \), calculate \( d_j(x), j = 1, \ldots, L \)
- Calculate class outputs, \( i = 1, \ldots, K \):
  - \( y_i = \sum_{j=1}^L \left( \log \frac{1}{\beta_j} \right) d_{ji}(x) \)
Cascading

Use $d_j$ only if preceding ones are not confident

Cascade learners in order of complexity
**Computational complexity**

- Define weak learners based on rectangle features
- For each round of boosting:
  - Evaluate each rectangle filter on each example
  - Select best threshold for each filter
  - Select best filter/threshold combination
  - Reweight examples
- Computational complexity of learning: $O(MNK)$
  - $M$ rounds, $N$ examples, $K$ features
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 Ada-Boost threshold to maximize detection (as opposed to minimizing total classification error)
  - Test on a validation set
- 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
Training Data

- Training Data
  - 5000 faces
    - All frontal, rescaled to 24x24 pixels
  - 300 million non-faces
    - 9500 non-face images
  - Faces are **normalized**
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose
AdaBoost Cascade Face Detector

A chain of classifiers that each reject some fraction of the negative training samples while keeping almost all positive ones.

Each classifier is an AdaBoost ensemble of rectangular Haar-like features sampled from a large pool.
Cascaded Classifier

- A 1 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative) – using data from previous stage.
- A 20 feature classifier achieves 100% detection rate with 10% false positive rate (2% cumulative).
Limitation of Adaboost

- AdaBoost minimizes a quantity related to classification error; it does not minimize the number of false negatives.

- Unfortunately feature selection proceeds as if classification error were the only goal, and the features selected are not optimal for the task of rejecting negative examples.
OpenCV

Open-source Computer Vision Library

- 2,500+ algorithms and functions
- Cross-platform, portable API
- Real-time performance
- Liberal BSD license
- Professionally developed
- C/C++/Python API
- Windows/Linux/Android/iPhone platforms

http://opencv.org/
Face Detection in OpenCV

OpenCV comes with a trainer as well as a detector

OpenCV already contains many pre-trained classifiers for face, eyes, smile etc.

![Image of face detection using a boosted cascade of simple features]
Face detection using a boosted cascade of simple features

July, the 1st 2014

OpenCV Example

Result

1. Here is the result of running the code above and using as input the video stream of a build-in webcam:

Remember to copy the files `haarcascade_frontalface_alt.xml` and `haarcascade_eye_tree_eyeglasses.xml` in your current directory. They are located in `opencv/data/haarcascades`

http://docs.opencv.org/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html
Haar-like features based boat detection

**POSITIVES**

HAAR-LIKE FEATURES BASED CLASSIFICATION

1. Edge features
   - (a)
   - (b)
   - (c)
   - (d)

2. Line features
   - (a)
   - (b)
   - (c)
   - (d)
   - (e)
   - (f)
   - (g)
   - (h)

3. Center-surround features
   - (a)
   - (b)

**NEGATIVES**

XML File

Face detection using a boosted cascade of simple features

July, the 1st 2014
Data set

Face detection using a boosted cascade of simple features
Cascade of weak-classifiers

At each stage a weak classifier is trained to achieve a hit rate of $h$ and a false alarm rate of $f$ ($f < 0.5$)
Detection Scheme

Input: Camera Frame, Haar Classifier
Output: List of Observations
Examples

VTS control center in Civitavecchia
Examples

VTS control center in Civitavecchia
False positive detections

Reflections and wakes on the water surface, can increase false positive detections
Solution: Adding levels to the cascade
weak-classifier for boat-weakes

Negative set

Weaks rejected

hitrate = h^N

falsealarms = f^N
Evaluation Metrics

The accuracy of a classifier can be computed in terms of detection rate (DR) and false alarm rate (FAR)

$$DR = \frac{TP}{TP + FN} \quad FAR = \frac{FP}{TP + FP}$$

where TP is the number of correctly detected objects, FN of not detected objects, and FP of the incorrect detections.
Detection Results on Real Data

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Coastline Detection</th>
<th>Detection Rate</th>
<th>False Alarm Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without wake examples</td>
<td>NO</td>
<td>0.892</td>
<td>0.475</td>
</tr>
<tr>
<td>Without wake examples</td>
<td>YES</td>
<td>0.892</td>
<td>0.265</td>
</tr>
<tr>
<td>With wake examples</td>
<td>YES</td>
<td>0.928</td>
<td>0.251</td>
</tr>
</tbody>
</table>

Domenico Bloisi, Luca Iocchi, Michele Fiorini, Giovanni Graziano

AUTOMATIC MARITIME SURVEILLANCE WITH VISUAL TARGET DETECTION
Detection Results on VOC dataset

Visual Object Classes (VOC)

20 classes. The train/val data has 11,530 images containing 27,450 ROI annotated objects and 5,034 segmentations.

The PASCAL Visual Object Classes (VOC) Challenge
### Detection Results on VOC dataset

<table>
<thead>
<tr>
<th>Coastline Detection</th>
<th>Detection Rate</th>
<th>False Alarm Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>0.872</td>
<td>0.332</td>
</tr>
<tr>
<td>YES</td>
<td>0.872</td>
<td>0.198</td>
</tr>
</tbody>
</table>

Domenico Bloisi, Luca Iocchi, Michele Fiorini, Giovanni Graziano

**Camera based target recognition for maritime awareness**

15th International Conference on Information Fusion
Fusion 2012
MAR
Maritime Activity Recognition dataset

Welcome to MAR home page

Maritime Activity Recognition (MAR) is a dataset containing data coming from different video sources (fixed and Pan-Tilt-Zoom cameras) and from different scenarios.

The aim of this project is to provide a set of videos that can be used to help in developing intelligent surveillance system for the maritime environment.

Feel free to contribute with your own videos!

If you make use of the MAR data, please cite the following reference in any publications:

http://www.dis.uniroma1.it/~labrococo/MAR
### MAR Data Set

<table>
<thead>
<tr>
<th>Filename</th>
<th>Sensor</th>
<th>Location</th>
<th>Time</th>
<th>Camera</th>
<th>Reflections</th>
<th>Occlusions</th>
<th>Fg-mask</th>
<th>Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR-validation.zip</td>
<td>IR</td>
<td>Northern Europe</td>
<td>Night</td>
<td>snapshots</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>reflections-1.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>reflections-2.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>reflections-3.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>occlusions-1.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>occlusions-2.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>occlusions-3.avi</td>
<td>EO</td>
<td>Northern Europe</td>
<td>Day</td>
<td>PTZ</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>wakes-1.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>wakes-2.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>wakes-3.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>two-boats-1.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>two-boats-2.avi</td>
<td>EO</td>
<td>Italy</td>
<td>Day</td>
<td>PTZ</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>venice-1.m4v</td>
<td>EO</td>
<td>Venice, Italy</td>
<td>Day</td>
<td>Static</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>venice-2.m4v</td>
<td>EO</td>
<td>Venice, Italy</td>
<td>Day</td>
<td>Static</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>venice-3.m4v</td>
<td>EO</td>
<td>Venice, Italy</td>
<td>Day</td>
<td>Static</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Boat Classification in Venice

Face detection using a boosted cascade of simple features

July, the 1st 2014
Additional References


F. Adolf, How-to build a cascade of boosted classifiers based on Haar-like features

V. Pisarevsky, OpenCV Object Detection: Theory and Practice

Face detection using a boosted cascade of simple features

Dr. Domenico Daniele Bloisi
bloisi@dis.uniroma1.it
www.dis.uniroma1.it/~bloisi

www.diag.uniroma1.it/~querzoni/great-ideas-in-ICT-2014