Object Detection

Lu Peng
School of Computer Science,
Beijing University of Posts and Telecommunications

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

What we will learn today?

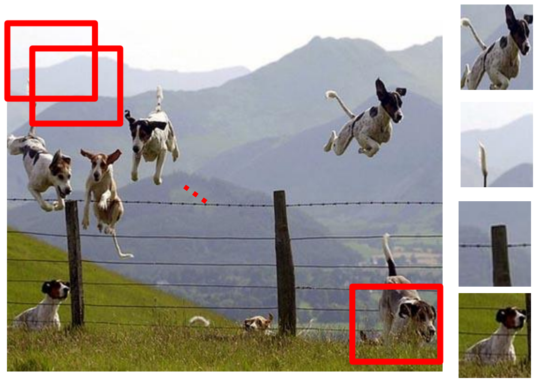
- Introduction of object detection
- Face Detection
- Pedestrian Detection

What we will learn today?

- Introduction of object detection
- Face Detection
- Pedestrian Detection

Object Category Detection

- Focus on object search: “Where is it?”
- Build templates that quickly differentiate object patch from background patch



Object or Non-Object?

Source: James Hays

Challenges in modeling the object class

Illumination

Object pose

‘Clutter’

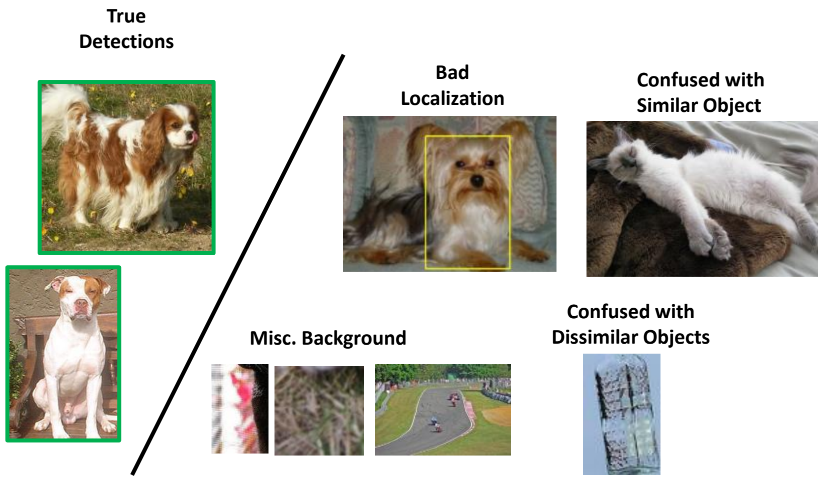
Occlusions

Intra-class appearance

Viewpoint

Source: K. Grauman

Challenges in modeling the non-object class



Source: James Hays

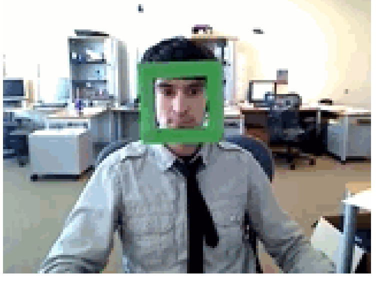
Object Detection Design challenges

- How to efficiently search for likely objects
  - Even simple models require searching hundreds of thousands of positions and scales.
- Feature design and scoring
  - How should appearance be modeled?
  - What features correspond to the object?
- How to deal with different viewpoints?
  - Often train different models for a few different viewpoints

What we will learn today?

- Introduction of object detection
- **Face Detection**
- Pedestrian Detection

Face Detection



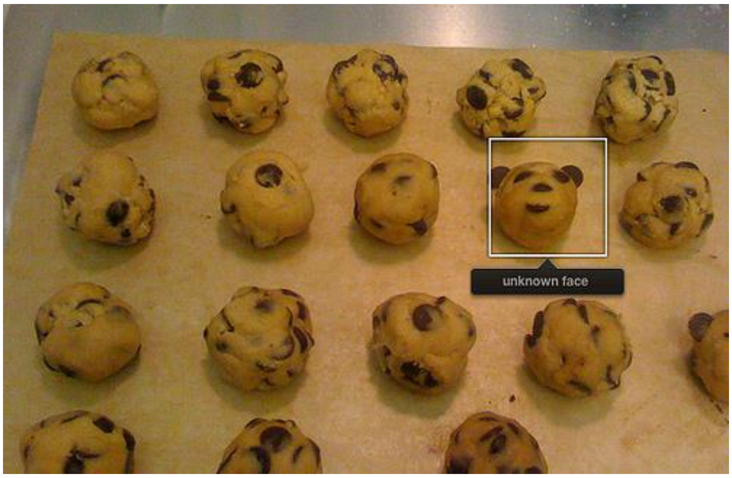
Behold a state-of-the-art face detector!

(Courtesy Boris Babenko)

Source： Svetlana Lazebnik

Consumer application: Apple iPhoto

[Things iPhoto thinks are faces]



Source： Svetlana Lazebnik

"The Nikon S60 detects up to 12 faces."

Source： Svetlana Lazebnik

Face detection and recognition

Source： Svetlana Lazebnik

Challenges of face detection

- **Sliding window = tens of thousands of location/scale evaluations**
  - One megapixel image has ~10^6 pixels, and a comparable number of candidate face locations
- **Faces are rare: 0–10 per image**
  - For computational efficiency, spend as little time as possible on the non-face windows.
  - For 1 Mpix, to avoid having a false positive in every image, our false positive rate has to be less than 10^{-6}

Source:James Hays

Sliding Window Face Detection with Viola-Jones

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* CVPR 2001.

P. Viola and M. Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Source： Svetlana Lazebnik

Rapid object detection using a boosted cascade of simple features
P Viola, M Jones - Proceedings of the 2001 IEEE computer …, 2001 - ieeexplore.ieee.org
This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the" integral image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely …
☆ 99 被引用 20640 次 相關文章 全部共 108 個版本 [文献分析]

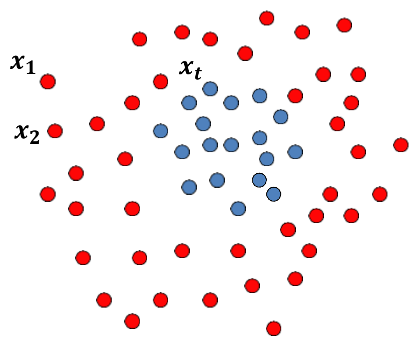
Histograms of oriented gradients for human detection
N Dalal, B Triggs - 2005 IEEE computer society conference on …, 2005 - ieeexplore.ieee.org
We study the question of feature sets for robust visual object recognition; adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of histograms of oriented gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality …
☆ 99 被引用 31161 次 相關文章 全部共 82 個版本 [文献分析]

Why boosting?

- A simple algorithm for learning robust classifiers
  - Freund & Shapire, 1995
  - Friedman, Hastie, Tibshhirani, 1998
- Provides efficient algorithm for sparse visual feature selection
  - *Tieu & Viola, 2000*
  - *Viola & Jones, 2003*
- Easy to implement, not requires external optimization tools.

Boosting - mathematics

- **It is a sequential procedure:**



Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\bullet) \end{cases}$$

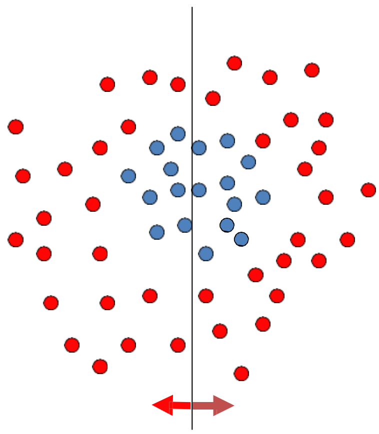
and a weight:

$w_t = 1$

Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

Toy example

Weak learners from the family of lines



Each data point has a class label:

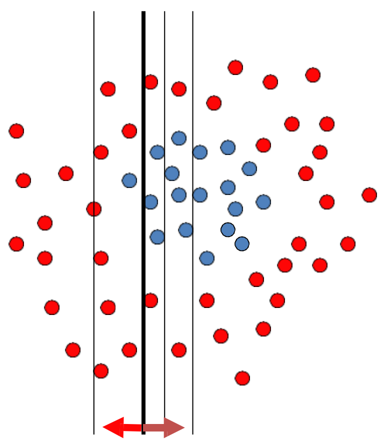
$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\bullet) \end{cases}$$

and a weight:

w_t**=1**

h => p(error) = 0.5 it is at chance

Toy example



Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\bullet) \end{cases}$$

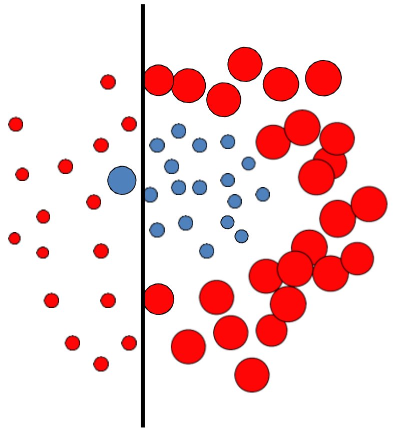
and a weight:

w_t**=1**

This one seems to be the best

This is a 'weak classifier': It performs slightly better than chance.

Toy example



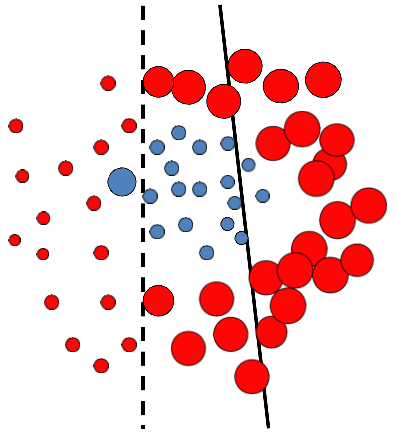
Each data point has a class label:

$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{blue}) \end{cases}$$

We update the weights:

$$\boxed{w_{t+1,i} \leftarrow w_{t,i}\beta_t^{1-|h_t(x_i)-y_i|}}$$

Toy example



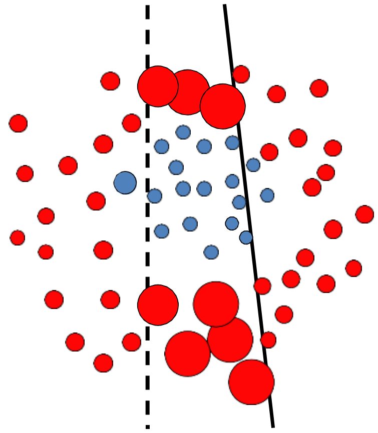
Each data point has a class label:

$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{blue}) \end{cases}$$

We update the weights:

$$\boxed{w_{t+1,i} \leftarrow w_{t,i}\beta_t^{1-|h_t(x_i)-y_i|}}$$

Toy example



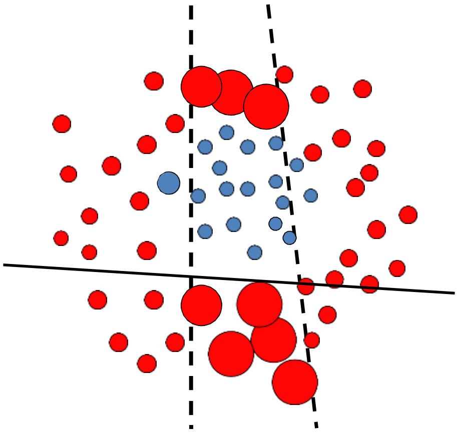
Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\bullet) \end{cases}$$

We update the weights:

$$w_{t+1,i} \leftarrow w_{t,i}\beta_t^{1-|h_t(x_i)-y_i|}$$

Toy example



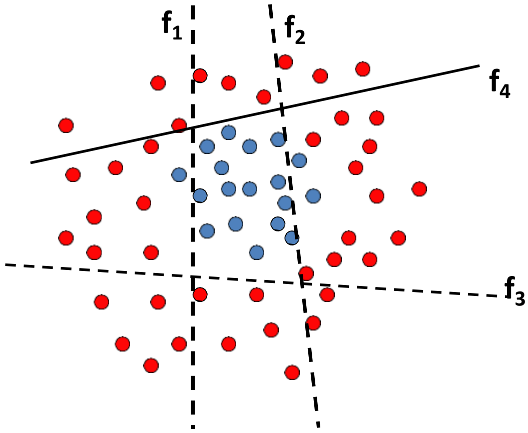
Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\bullet) \end{cases}$$

We update the weights:

$$w_{t+1,i} \leftarrow w_{t,i}\beta_t^{1-|h_t(x_i)-y_i|}$$

Toy example



The strong (non- linear) classifier is built as the combination of all the weak (linear) classifiers.

Boosting - mathematics

- Defines a classifier using an additive model:

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{\alpha}_1\boldsymbol{h}_1(\boldsymbol{x}) + \boldsymbol{\alpha}_1\boldsymbol{h}_1(\boldsymbol{x}) + \boldsymbol{\alpha}_1\boldsymbol{h}_1(\boldsymbol{x}) + \cdots$$

Strong classifier

Features vector

Weight

Weak classifier

- **We need to define a family of weak classifiers**

$\boldsymbol{h}_k(\boldsymbol{x})$ **form a family of weak classifiers**

Boosting - mathematics

- **Weak learners**

value of rectangle feature

$$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

threshold

- **Final strong classifier**

$$\boxed{h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}}$$

Viola & Jones algorithm

- A “paradigmatic” method for 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* for fast rejection of non-face windows

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

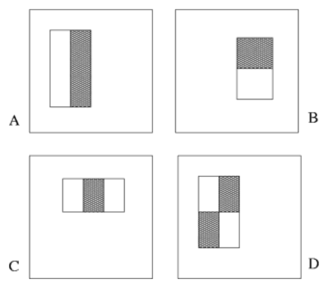
Viola & Jones algorithm

- A "paradigmatic" method for 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* for fast rejection of non-face windows

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

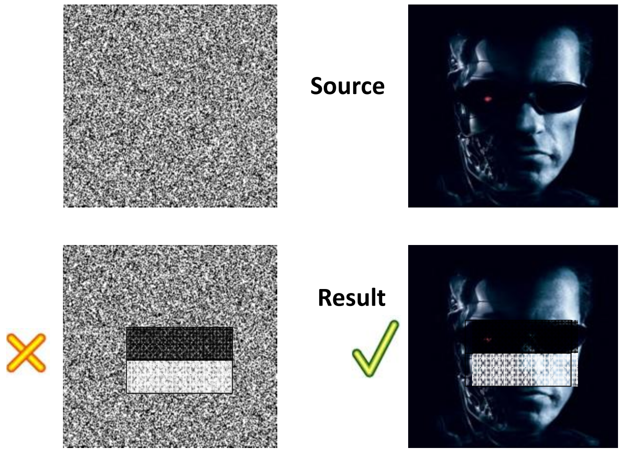
Weak classifier

- 4 kind of Rectangle filters
- Value = ∑ (pixels in white area) – ∑ (pixels in black area)



Source： Svetlana Lazebnik

Weak classifier

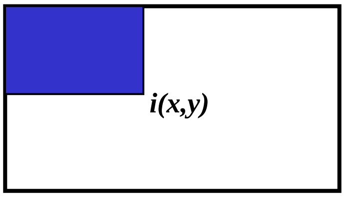


Source： Svetlana Lazebnik

Integral images for fast feature evaluation

- The integral image computes a value at each pixel (x,y) that is the sum of all pixel values above and to the left of (x,y), inclusive.
- This can quickly be computed in one pass through the image.
- ‘Summed area table’

$$I_{\Sigma}(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$

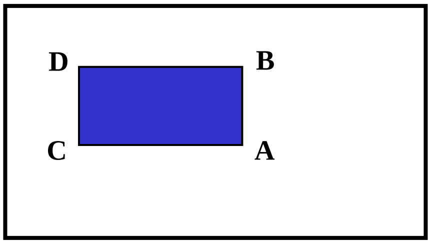


Source： Svetlana Lazebnik

Computing sum within a rectangle

- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- The sum of original image values within the rectangle can be computed as:

  sum = A – B – C + D

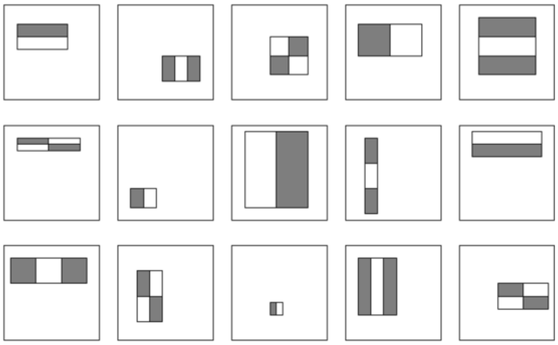


Only 3 additions are required for any size of rectangle!

Source: Svetlana Lazebnik

Viola & Jones algorithm

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



P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Source: Svetlana Lazebnik

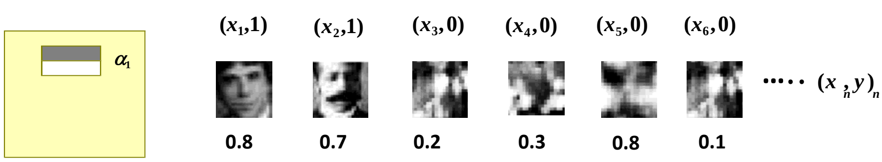
Viola & Jones algorithm

- A “paradigmatic” method for 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* for fast rejection of non-face windows

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Viola & Jones algorithm

1. Evaluate each rectangle filter on each example



value of rectangle feature

Weak classifier $$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

threshold

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Viola & Jones algorithm

2. Select best filter/threshold combination

a. Normalize the weights $\boldsymbol{w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_j^n w_{t,j}}}$

$$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

b. For each feature, j $\boldsymbol{\varepsilon_j = \sum_i w_i |h_j(x_i) - y_i|}$

c. Choose the classifier, h_t with the lowest error t

3. Reweight examples

$$\boldsymbol{w_{t+1,i} \leftarrow w_{t,i}\beta_t^{1-|h_t(x_i)-y_i|}} \qquad \boldsymbol{\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}}$$

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Viola & Jones algorithm

4. The final strong classifier is

$$\boldsymbol{\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}}$$

$$\boldsymbol{h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2}\sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}}$$

$$\boldsymbol{\alpha_t = \log \frac{1}{\beta_t}}$$

The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

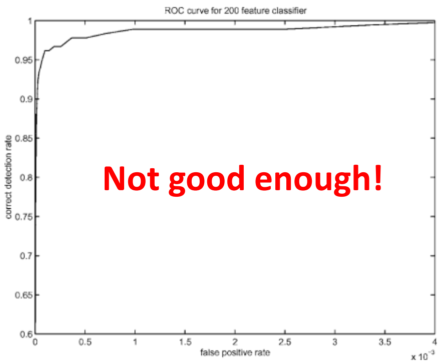
Viola & Jones algorithm

- A “paradigmatic” method for 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* for fast rejection of non-face windows

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Boosting for face detection

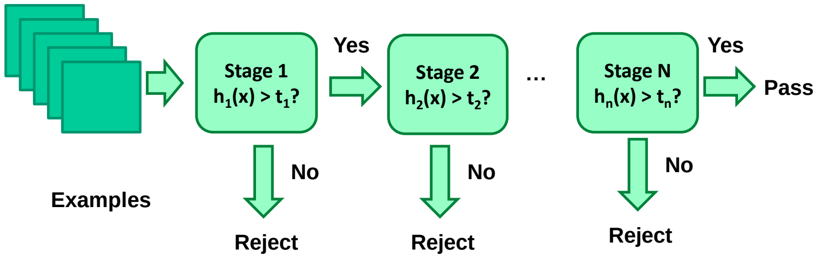
- A 200-feature classifier can yield 95% detection rate and a false positive rate of 1 in 14084



Receiver operating characteristic (ROC) curve

Source： Svetlana Lazebnik

Cascade for Fast Detection

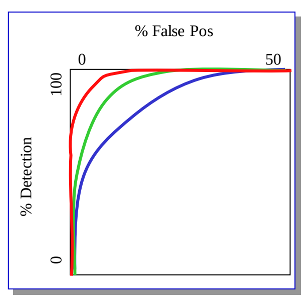


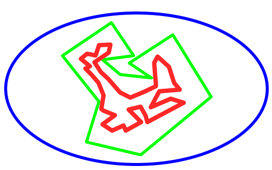
- Fast classifiers early in cascade which reject many negative examples but detect almost all positive examples.
- Slow classifiers later, but most examples don't get there.

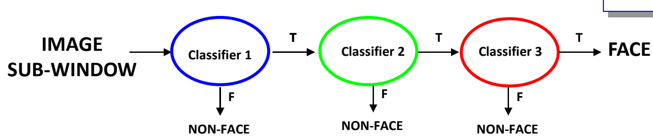
Attentional cascade

Chain classifiers that are progressively more complex and have lower false positive rates:

Receiver operating characteristic



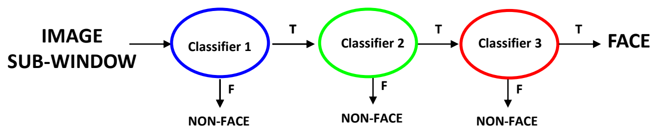




Source： Svetlana Lazebnik

Attentional cascade

- The detection rate and the false positive rate of the cascade are found by multiplying the respective rates of the individual stages
- A detection rate of 0.9 and a false positive rate on the order of 10^{-6} can be achieved by a 10-stage cascade if each stage has a detection rate of 0.99 ($0.99^{10} \approx 0.9$) and a false positive rate of about 0.30 ($0.3^{10} \approx 6\times10^{-6}$)



Source：Svetlana Lazebnik

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 boosting 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

The implemented system

- 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

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

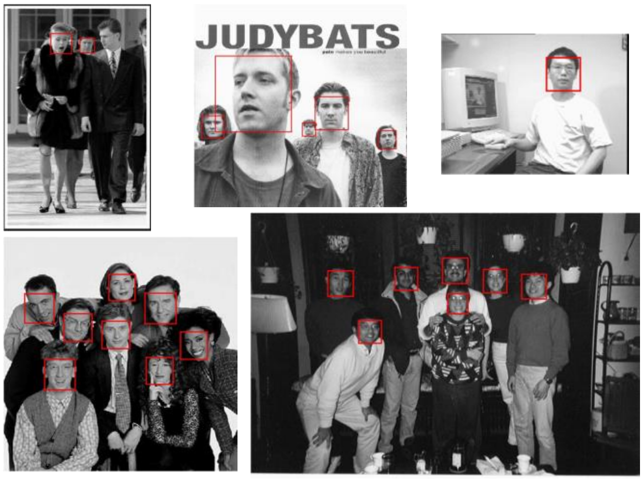
System performance

- Training time: “weeks” on 466 MHz Sun workstation
- 38 layers, total of 6061 features
- Average of 10 features evaluated per window on test set
- “On a 700 Mhz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds”
  - 15 Hz
  - 15 times faster than previous detector of comparable accuracy (Rowley et al., 1998)

- First two features selected by boosting:
- This feature combination can yield 100% detection rate and 50% false positive rate

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

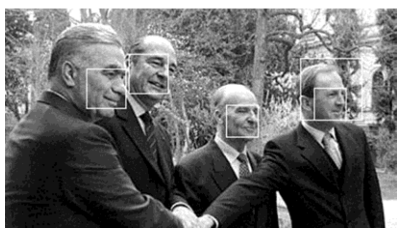
Output of Face Detector on Test Images



P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

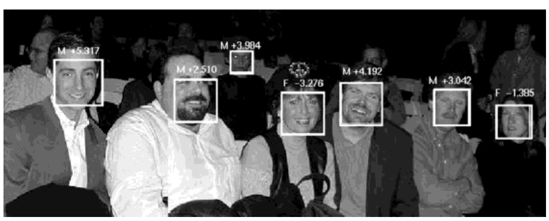
Other detection tasks

Facial Feature Localization



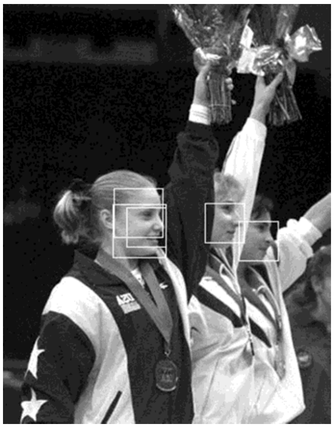
Profile Detection

Male vs. female



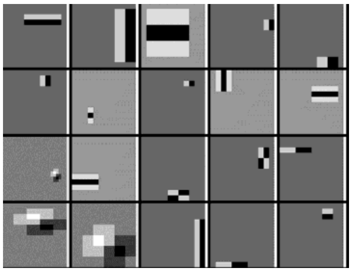
Source： Svetlana Lazebnik

Profile Detection



Source： Svetlana Lazebnik

Profile Features



Source： Svetlana Lazebnik

Face Image Databases

- Databases for face recognition can be best utilized as training sets
  - Each image consists of an individual on a uniform and uncluttered background
- Test Sets for face detection
  - MIT, CMU (frontal, profile), Kodak

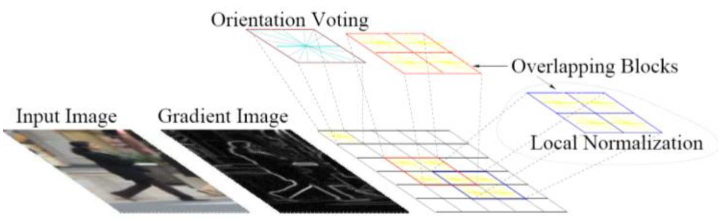
What we will learn today?

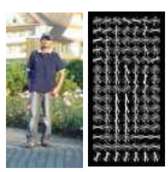
- Introduction of object detection
- Face Detection
- **Pedestrian Detection**

Rapid object detection using a boosted cascade of simple features
P Viola, M Jones - Proceedings of the 2001 IEEE computer …, 2001 - ieeexplore.ieee.org
This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the" integral image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely …
☆ 99 被引用 20640 次 相關文章 全部共 108 個版本 [文献分析]

Histograms of oriented gradients for human detection
N Dalal, B Triggs - 2005 IEEE computer society conference on …, 2005 - ieeexplore.ieee.org
We study the question of feature sets for robust visual object recognition; adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of histograms of oriented gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality …
☆ 99 被引用 31161 次 相關文章 全部共 82 個版本 [文献分析]

HoG Feature





detection window :64x128

Block size:16x16

Stride:8x8

Cell size:8x8

Bin number: 9

HOG dim： 3780

Code available: http://pascal.inrialpes.fr/soft/olt/

Source： Kristen Grauman

Dalal-Triggs pedestrian detector

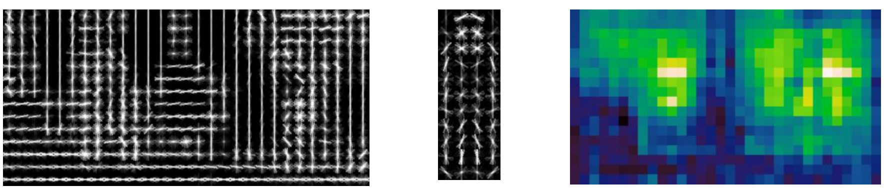
1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

N. Dalal and B. Triggs, [Histograms of Oriented Gradients for Human Detection](), CVPR 2005

Pedestrian detection with HOG

- Learn a pedestrian template using a support vector machine
- At test time, compare feature map with template over sliding windows.
- Find local maxima of response
- *Multi-scale*: repeat over multiple levels of a HOG pyramid

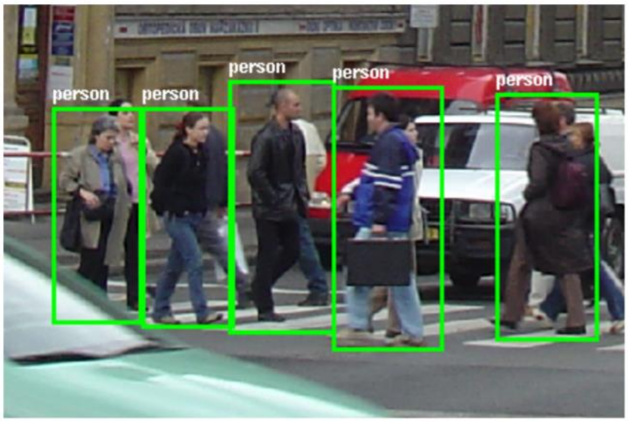
HOG feature map **Template** **Detector response map**



Can be continuous for more sophisticated maxima finding

N. Dalal and B. Triggs, [Histograms of Oriented Gradients for Human Detection](), CVPR 2005

INRIA pedestrian database



Source： Kristen Grauman

Something to think about…

- Sliding window detectors work
  - \- *very well* for faces
  - \- *fairly well* for cars and pedestrians
  - \- *badly* for cats and dogs
- Why are some classes easier than others?

Strengths/Weaknesses of Statistical Template Approach

- **Strengths**
  - Works very well for non-deformable objects with canonical orientations: faces, cars, pedestrians
  - Fast detection
- **Weaknesses**
  - Not so well for highly deformable objects or “stuff”
  - Not robust to occlusion
  - Requires lots of training data