Robust Real-time Object Detection

by

Paul Viola and Michael Jones

ICCV 2001 Workshop on Statistical and Computation Theories of Vision

Presentation by Gyozo Gidofalvi

Computer Science and Engineering Department

University of California, San Diego

gyozo@cs.ucsd.edu

October 25, 2001

Outline

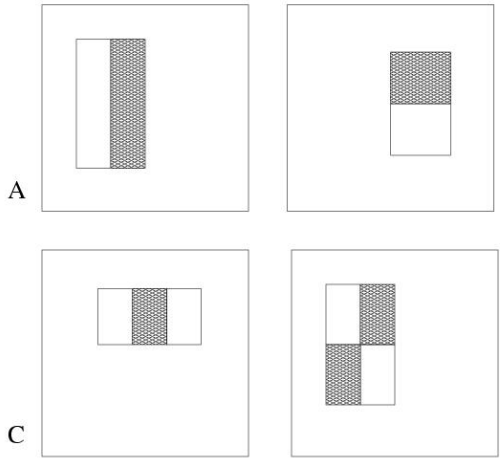
- Object detection task
- Definition and rapid evaluation of simple features for object detection
- Method for classification and feature selection, a variant of AdaBoost
- Speed-up through the Attentional Cascade
- Experiments and Results
- Conclusions

Object detection task

- Object detection framework: Given a set of images find regions in these images which contain instances of a certain kind of object.
- Task: Develop an algorithm to learn an fast and accurate method for object detection.

To capture ad-hoc domain knowledge classifiers for images do not operate on raw grayscale pixel values but rather on values obtained from applying simple filters to the pixels.

Definition of simple features for object detection



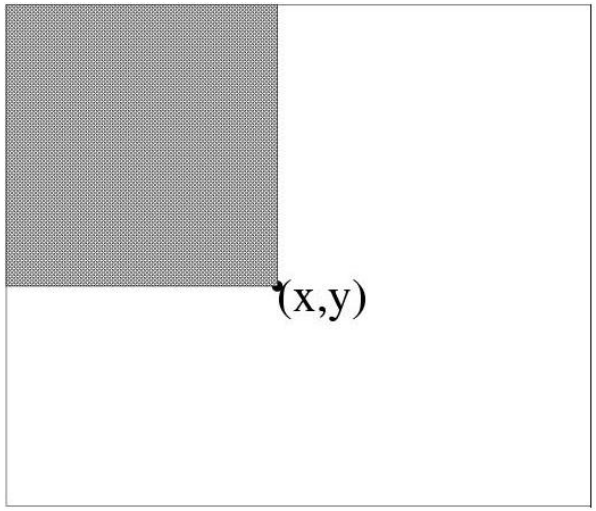
3 rectangular features types:

- *two-rectangle feature* type (horizontal/vertical)
- *three-rectangle feature* type
- *four-rectangle feature* type

Using a 24x24 pixel base detection window, with all the possible combination of horizontal and vertical location and scale of these feature types the full set of features has 49,396 features.

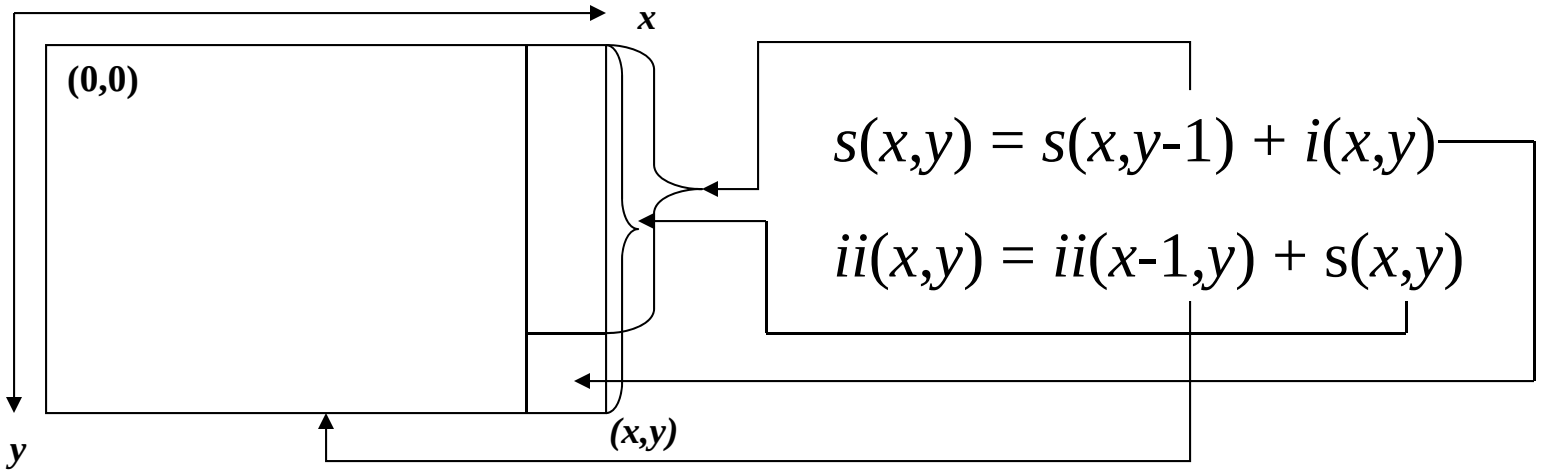
The motivation behind using rectangular features, as opposed to more expressive steerable filters is due to their extreme computational efficiency.

Integral image

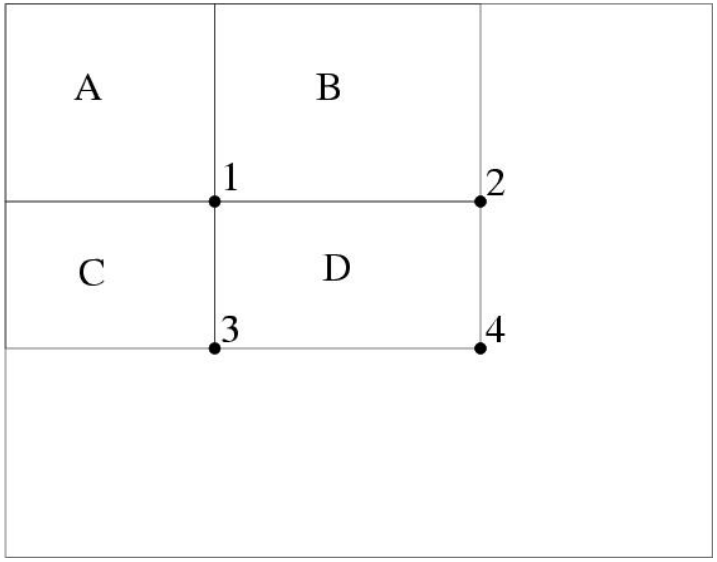


Def: The *integral image* at location (*x*,*y*), is the sum of the pixel values above and to the left of (*x*,*y*), inclusive.

Using the following two recurrences, where *i*(*x*,*y*) is the pixel value of original image at the given location and *s*(*x*,*y*) is the cumulative column sum, we can calculate the integral image representation of the image in a single pass.



Rapid evaluation of rectangular features



Using the integral image representation one can compute the value of any rectangular sum in constant time.

For example the integral sum inside rectangle D we can compute as:

$$ii(4) + ii(1) - ii(2) - ii(3)$$

As a result two-, three-, and four-rectangular features can be computed with 6, 8 and 9 array references respectively.

Challenges for learning a classification function

- Given a feature set and labeled training set of images one can apply number of machine learning techniques.
- Recall however, that there is 45,396 features associated with each image sub-window, hence the computation of all features is computationally prohibitive.
- Hypothesis: A combination of only a small number of these features can yield an effective classifier.
- Challenge: Find these discriminant features.

A variant of AdaBoost for aggressive feature selection

- Given example images (x_1,y_1) , … , (x_n,y_n) where y_i = 0, 1 for negative and positive examples respectively.
- Initialize weights $w_{1,i}$ = 1/(2m), 1/(2l) for training example i, where m and l are the number of negatives and positives respectively.

For t = 1 … T

1) Normalize weights so that w_t is a distribution
2) For each feature j train a classifier h_j and evaluate its error ε_j with respect to w_t.
3) Chose the classifier h_j with lowest error.
4) Update weights according to:

$$w_{t+1,i} = w_{t,i}\beta_t^{1-\varepsilon_i}$$

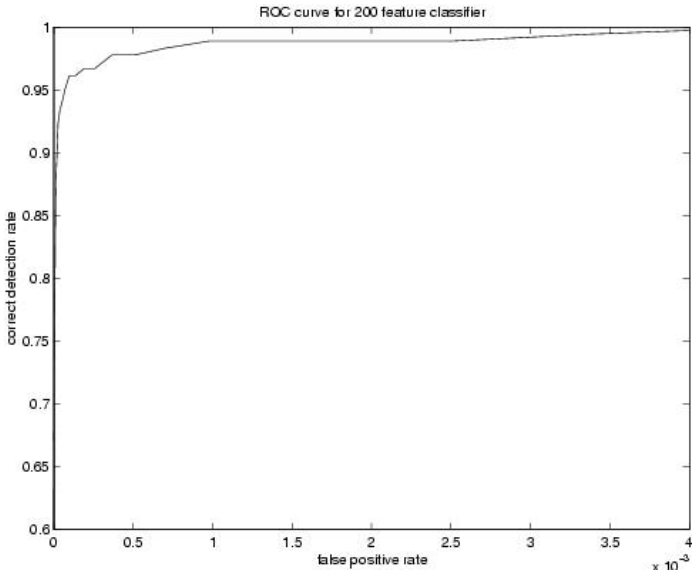
where e_i = 0 is x_i is classified correctly, 1 otherwise, and

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

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T}\alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T}\alpha_t, \\ 0 & \textit{otherwise} \end{cases} \qquad \text{where} \qquad \alpha_t = \log(\frac{1}{\beta_t})$$

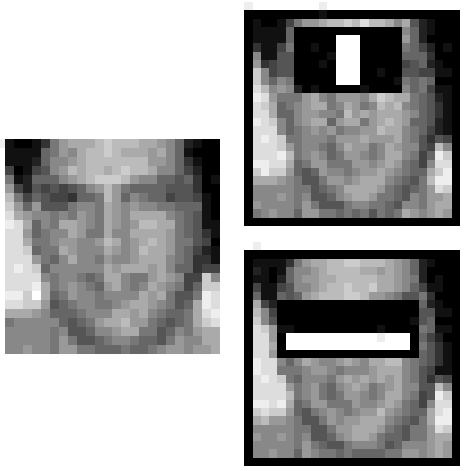
Performance of 200 feature face detector



The ROC curve of the constructed classifies indicates that a reasonable detection rate of 0.95 can be achieved while maintaining an extremely low false positive rate of approximately 10^{-4}.

- First features selected by AdaBoost are meaningful and have high discriminative power

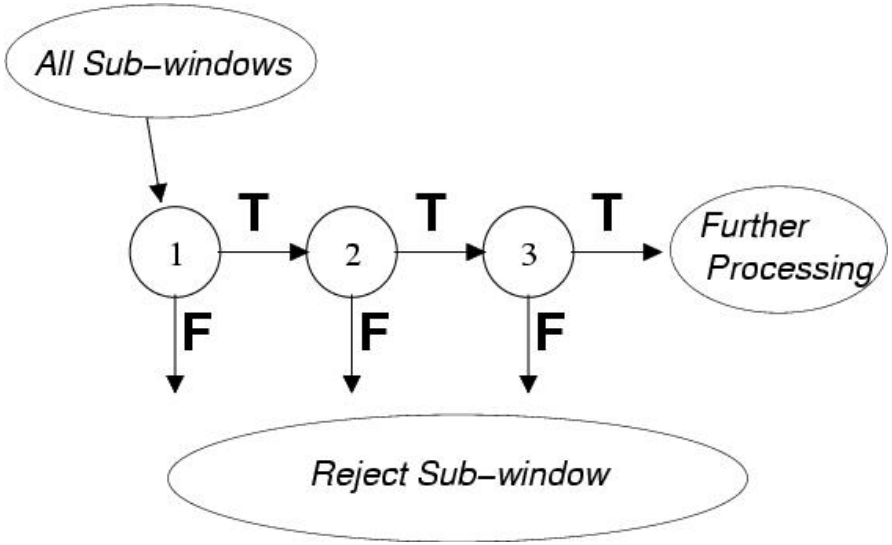
- By varying the threshold of the final classifier one can construct a two-feature classifier which has a detection rate of 1 and a false positive rate of 0.4.



Speed-up through the Attentional Cascade

• Simple, boosted classifiers can reject many of negative sub-windows while detecting all positive instances.

• Series of such simple classifiers can achieve good detection performance while eliminating the need for further processing of negative sub-windows.



Processing in / training of the Attentional Cascade

Processing: is essentially identical to the processing performed by a degenerate decision tree, namely only a positive result from a previous classifier triggers the evaluation of the subsequent classifier.

Training: is also much like the training of a decision tree, namely subsequent classifiers are trained only on examples which pass through all the previous classifiers. Hence the task faced by classifiers further down the cascade is more difficult.

To achieve efficient cascade for a given false positive rate F and detection rate D we would like to minimize the expected number of features evaluated N:

$$N = n_0 + \sum_{i=1}^{K} \left(n_i \prod_{j<i} p_j \right)$$

Since this optimization is extremely difficult the usual framework is to choose a minimal acceptable false positive and detection rate per layer.

Algorithm for training a cascade of classifiers

- User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
- User selects target overall false positive rate F_{target}.
- P = set of positive examples
- N = set of negative examples
- $F_0 = 1.0$; $D_0 = 1.0$; $i = 0$

While $F_i > F_{target}$

i++

$n_i = 0$; $F_i = F_{i-1}$

while $F_i > f$ x F_{i-1}

- n_i ++
- Use P and N to train a classifier with n_i features using AdaBoost
- Evaluate current cascaded classifier on validation set to determine F_i and D_i
- Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least d x D_{i-1} (this also affects F_i)

$N = \varnothing$

If $F_i > F_{target}$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N.

Experiments (dataset for training)

- 4916 positive training example were hand picked aligned, normalized, and scaled to a base resolution of 24x24
- 10,000 negative examples were selected by randomly picking sub-windows from 9500 images which did not contain faces

Experiments cont.
(structure of the detector cascade)

• The final detector had 32 layers and 4297 features total

Layer number	1	2	3 to 5	6 and 7	8 to 12	13 to 32
Number of feautures	2	5	20	50	100	200
Detection rate	100%	100%	-	-	-	-
Rejection rate	60%	80%	-	-	-	-

• Speed of the detector ~ total number of features evaluated
• On the MIT-CMU test set the average number of features evaluated is 8 (out of 4297).
• The processing time of a 384 by 288 pixel image on a conventional personal computer about .067 seconds.
• Processing time should linearly scale with image size, hence processing of a 3.1 mega pixel images taken from a digital camera should approximately take 2 seconds.

Operation of the face detector

- Since training examples were normalized, image sub-windows needed to be normalized also. This **normalization** of images can be efficiently done using two integral images (regular / squared).
- **Detection at multiple scales** is achieved by scaling the detector itself.
- The amount of **shift** between subsequent sub-windows is determined by some constant number of pixels and the current scale.
- **Multiple detections** of a face, due to the insensitivity to small changes in the image of the final detector were, were combined based on overlapping bounding region.

Results

Testing of the final face detector was performed using the MIT+CMU frontal face test which consists of:

- 130 images
- 505 labeled frontal faces

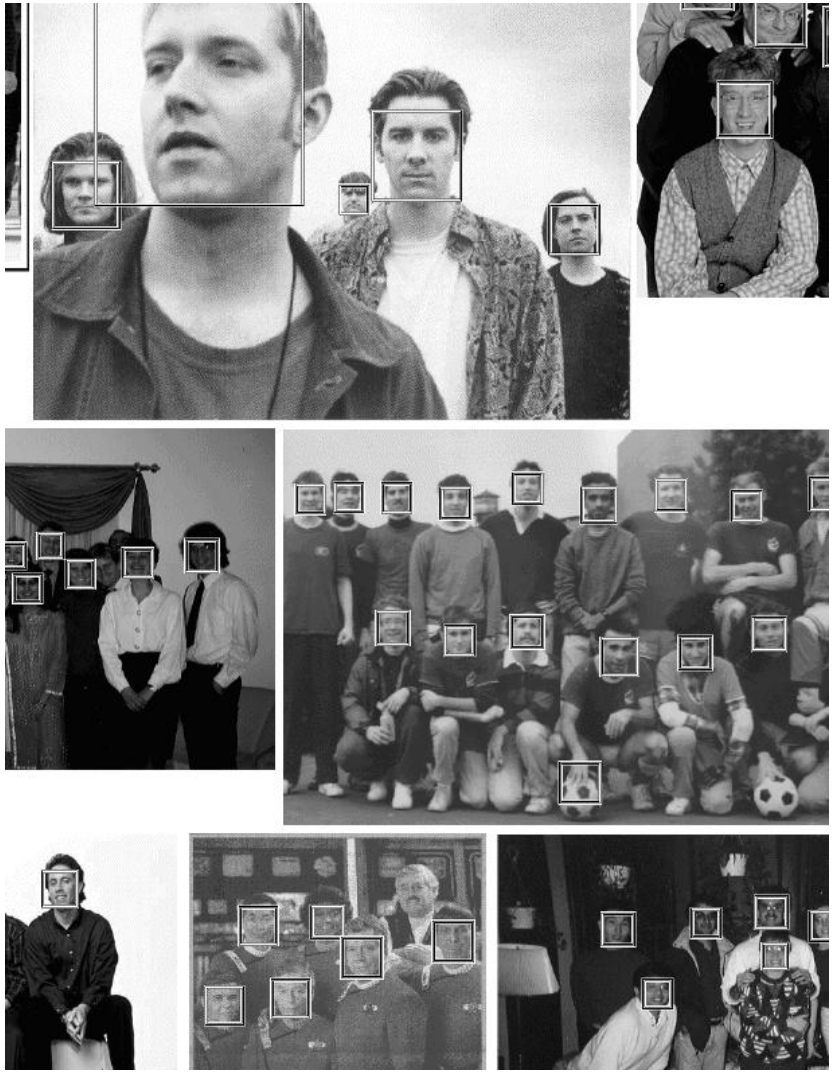
Results in the table compare the performance of the detector to best face detectors known.

False detections	10	31	50	65	78	95	110	167	422
Viola-Jones	78.3%	85.2%	88.8%	89.8%	90.1%	90.8%	91.1%	91.8%	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	-	90.1%	89.9%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-	-	-
Roth-Yang-Ajuha	-	-	-	-	94.8%	-	-	-	-

Rowley at al.: use a combination of 1wo neural networks (simple network for prescreening larger regions, complex network for detection of faces).

Schneiderman at al.: use a set of models to capture the variation in facial appearance; each model describes the statistical behavior of a group of wavelet coefficients.

Results cont.



Conclusion

- The paper presents general object detection method which is illustrated on the face detection task.
- Using the integral image representation and simple rectangular features eliminate the need of expensive calculation of multi-scale image pyramid.
- Simple modification to AdaBoost gives a general technique for efficient feature selection.
- A general technique for constructing a cascade of homogeneous classifiers is presented, which can reject most of the negative examples at early stages of processing thereby significantly reducing computation time.
- A face detector using these techniques is presented which is comparable in classification performance to, and orders of magnitude faster than the best detectors know today.