AA 2022/2023

Machine Learning for Modelling: *Supervised Learning*

Simone Bianco

1

AA 2022/2023

Viola-Jones Object Detection Framework

2

Basic concepts

Developed by Paul Viola and Michael Jones back in 2001, the **Viola-Jones Object Detection Framework** [1] can quickly and accurately detect objects in images

Despite its age the framework is still a leading player in face detection along side many of its CNNs counterparts.

To perform an object detection that is fast and accurate, the VJ Object Detection Framework combines the concepts of:

- Haar-like features
- Integral image
- AdaBoost algorithm
- Cascade classifier

Thus, to understand the framework, we first need to understand each of these concepts

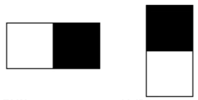
[1] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001.* Vol. 1. Ieee, 2001.

3

Haar-like features

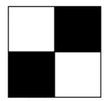
The detector classifies images based on the value of simple features, that are reminiscent of Haar basis functions. Three kinds of features are used:

- **Two-rectangle feature** is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and are horizontally or vertically adjacent



- **Three-rectangle feature** computes the sum withing two outside rectangles subtracted from the sum in a center rectangle

- **Four-rectangle feature** computes the difference between diagonal pairs of rectangles



The different types of features let us extract useful information from an image such as edges, straight lines, and diagonal lines that can be used to identify and object.

4

Haar-like features: example

We can compute the value of a feature in a given image location by subtracting the sum of the pixels falling in the white area from the sum of the pixel values falling in the black area

10	1	15	1	10
30	9	1	15	1
21	34	1	14	1
78	94	87	1	1
1	4	4	10	1

10	1	15	1	10
30	9	1	15	1
21	34	1	14	1
78	94	87	1	1
1	4	4	10	1

$$= B - W = (1 + 15 + 1 + 14) - (30 + 9 + 21 + 34) = 31 - 94 = -63$$

5

Integral image

Computing the value of the features can be very intensive since the number of pixels would be much larger within a large feature.

The integral image is an intermediate representation of an image where the value for location (x, y) on the integral image equals the sum of the pixel above and to the left (inclusive) of the (x, y) location on the original image:

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

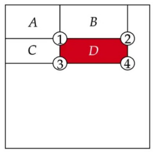
Which can be computed in one pass over the original image.

This intermediate representation is essential because it allows for fast calculation of rectangular regions.

6

Integral image

The sum of the red region can be calculated in constant time instead of having to loop through all the pixels in that region.



$$\begin{aligned} D &= 4 - 2 - 3 + 1 \\ &= 4 + 1 - (2 + 3) \end{aligned}$$

Original image

1	12	45	10
6	5	11	4
3	7	10	8
5	9	4	7

1	12	45	10
6	5	11	4
3	7	10	8
5	9	4	7

$$= 5 + 11 + 4 + 7 + 10 + 8 + 9 + 4 + 7 = 65$$

Integral image

1	13	58	68
7	24	80	94
10	34	100	122
15	48	118	147

1	13	58	68
7	24	80	94
10	34	100	122
15	48	118	147

$$= 147 + 1 - (68 + 15) = 65$$

7

Integral image

Because Haar-like features are rectangular, the use of the integral image cuts down their computation.

The computation of the sum of the pixels within **any** rectangle is constant and amounts to just 4 operations!

8

The AdaBoost algorithm

We have already seen it (remember?)

Assuming a base resolution of the detector equal to 24x24 there are about 180,000 rectangle features associated with each such image sub-window.

Even though each feature can be computer very efficiently, computing the complete set is too expensive.

The hypothesis is that a very small number of these features can be combined to form an effective classifier. The main challenge is to find these features!

The weak learning algorithm is designed to select the single rectangle feature that best separates the positive and negative examples.

For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified.

9

The AdaBoost algorithm

A weak classifier $h_j(x)$ thus consists of a feature f_j, a threshold θ_j and a polarity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 \quad if \quad p_j f_j(x) < p_j \theta_j \\ 0 \; otherwise \end{cases}$$

Where x is a 24x24 pixel sub-window of an image.

10

The AdaBoost algorithm

- Given example images $(x_1, y_1), \cdots, (x_n, y_n)$ where $y_i = 0{,}1$ for negative and positive examples, respectively.
- Initialize weights $w_{1,i} = {}^1/_{2m}, {}^1/_{2l}$ for $y_i = 0{,}1$ respectively, with m and l are the number of negative and positive example respectively
- For t=1,…,T:
  1. Normalize the weights $w_{t,i} = {}^{w_{t,i}}/_{\sum_{j=1}^{n} w_{t,j}}$ 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 ϵ_t
  4. Update the weights: $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$ where $e_i = 0$ if example x_i is correctly classified, $e_i = 1$ otherwise, and $\beta_t = {}^{\epsilon_t}/_{1-\epsilon_t}$
- The final strong classifier is:

$$\mathrm{H}(x) = \begin{cases} 1 \quad if \quad \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ \qquad 0 \quad otherwise \end{cases}$$

where $\alpha_t = \frac{1}{\beta_t}$

11

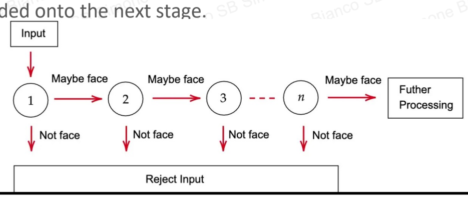
The cascade classifier

A cascade classifier is a multi-stage classifier that can perform detection quickly and accurately.

Each stage consists of a strong classifier produced by the AdaBoost classifier.

From one stage to another, the number of weak classifiers in a strong classifier increases.

An input is evaluated on a sequential (stage by stage) basis. If a classifier for a specific stage outputs a negative result, the input is discarded immediately. In case the output is positive, the input is forwarded onto the next stage.



12

The cascade classifier

According to Viola & Jones, this multi-stage approach allows for the construction of simpler classifiers which can then be used to reject most negative inputs quickly while spending more time on positive inputs.

The cascade training process involves two types of tradeoffs:

- In most cases classifiers with more features will achieve higher detection rates and lower false positive rates
- At the same time classifiers with more features require more time to compute

In practice a very simple framework is used to produce an effective classifier which is highly efficient: each stage in the cascade reduces the false positive rate and decreases the detection rate.

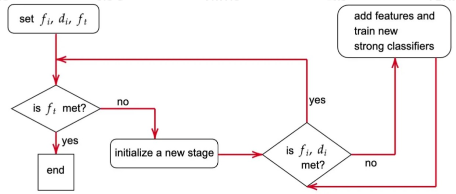
13

The cascade classifier

A target is selected for the minimum reduction in FP and the maximum decrease in detection.

Each stage is trained by adding features until the target detection and FP rate are met (they are determined by testing the detector on a validation set).

Stages are added until the overall target for FP and detection rate is met.



f_i = maximum acceptable false positive rate per stage
d_i = minimum acceptable true positive rate per stage
f_t = target overall false positive rate

14