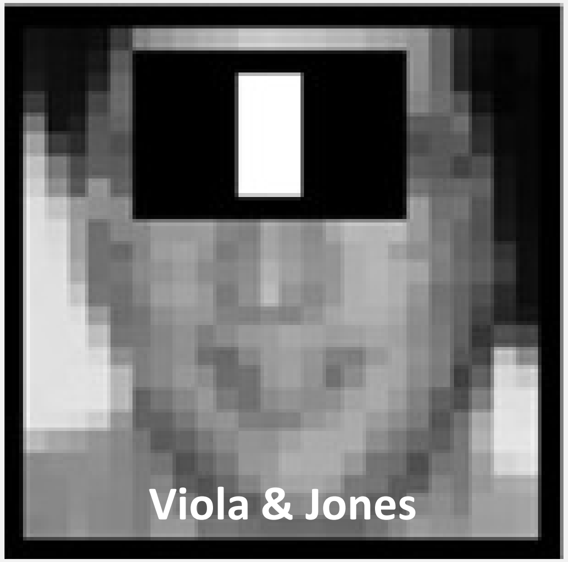
Department of Computer Science
University of Bristol

COMS30121/COMSM0020
Image Processing and Computer Vision

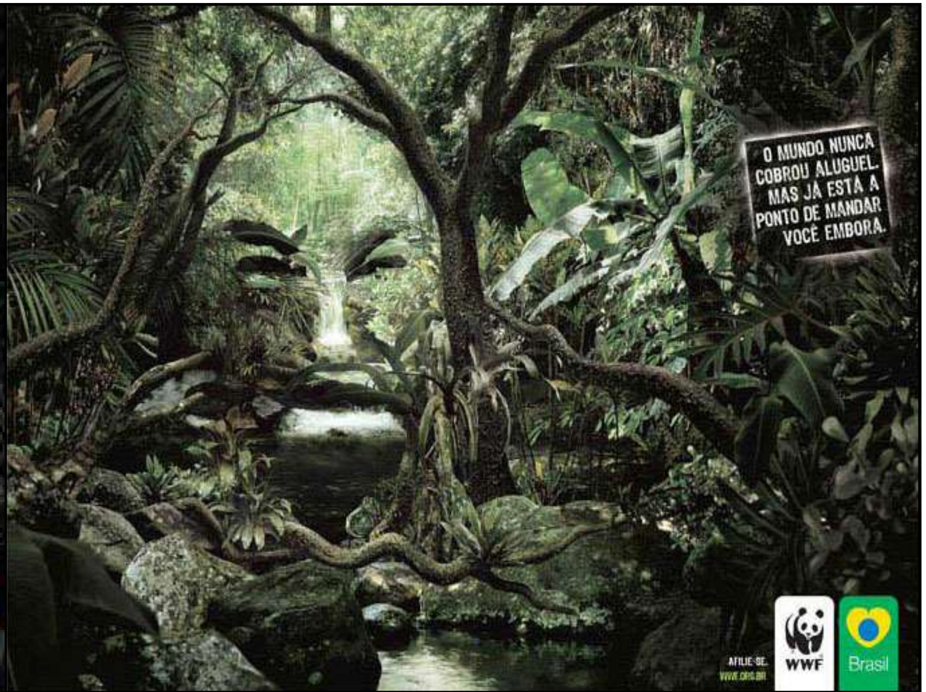


Lecture 06

Basics of Classical Object Detection

16 Slides


A GENTE SÓ QUER
QUE O CÉU
CONTINUE AZUL,
A MATA CONTINUE
VERDE E SUA
CONSCIÊNCIA LIMPA.
WWF
Brasil


O MUNDO NUNCA
COBROU ALUGUEL.
MAS JÁ ESTÁ A
PONTO DE MANDAR
VOCÊ EMBORA.
AFILIE-SE.
WWW.ORG.BR
WWF
Brasil


A SOLUÇÃO PARA
O AQUECIMENTO
GLOBAL NÃO
É COMPRAR UM
AR-CONDICIONADO.
WWF
Brasil


REFLORESTE-SE.
AFILIE-SE.
WWW.ORG.BR
WWF
Brasil

What is ‘Object Detection’?

- Object Detection aims at bridging the ‘semantic gap‘ between...
  - given pixel values, *and*
  - meaningful objects (grouping of pixels + classification of groups)

→ **image regions need to be found and assigned with semantic labels from a space of object classes**

- Why do classical shape detection and segmentation on their own rarely work for real-world object detection?
  - high intra-class, low inter-class variance
  - classes are rarely well defined
  - change of illumination, scale, pose + deformation, occlusion...

→ **object recognition is a difficult task**

Variable visual appearance

First Real-time Detection Method: Viola & Jones' (2001)
(base line standard for off-the-shelf method for almost a decade)

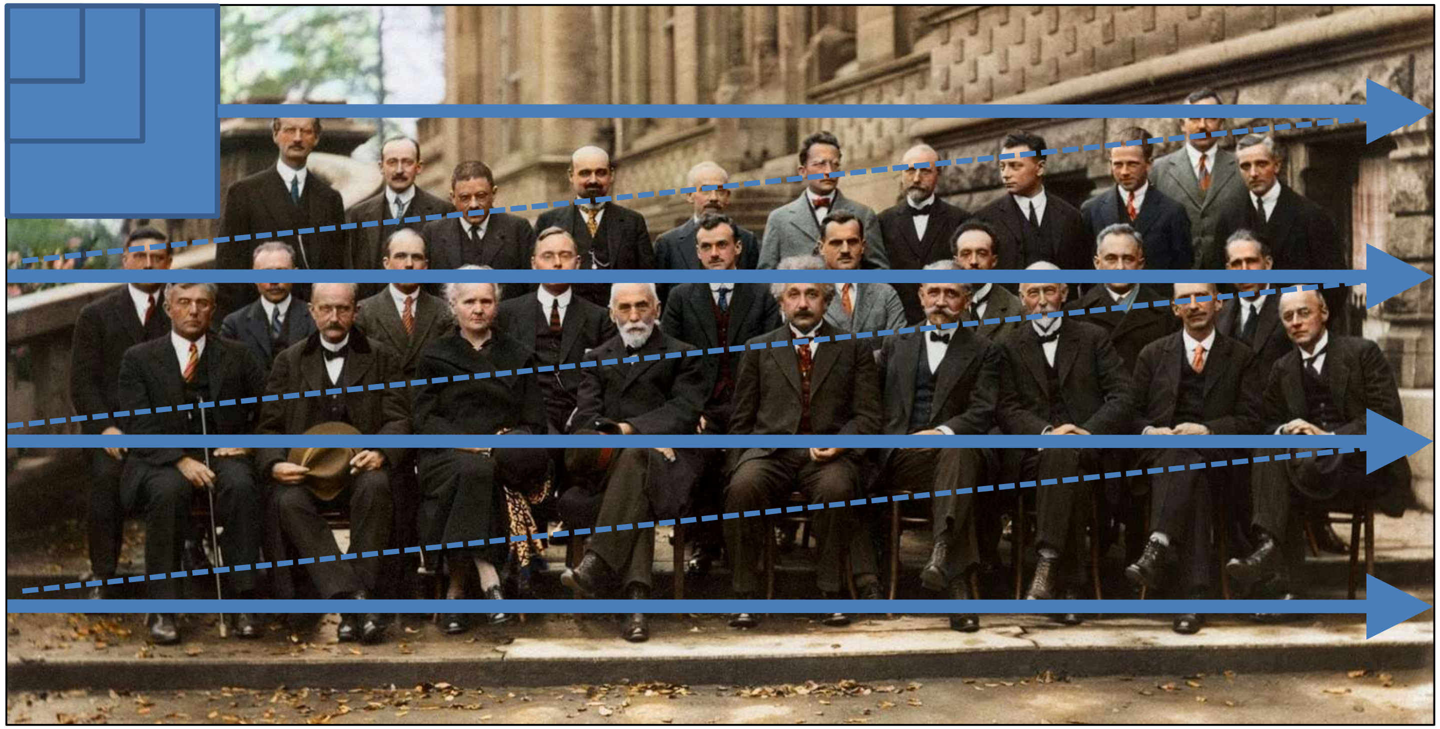
Our Agenda:

– Viola Jones technique overview
– Sliding Window Detectors
– Haar-like Features
– Feature Extraction and Integral Images
– Weak Classifiers
– Boosting and Classifier Evaluation
– Cascades of Boosted Classifiers

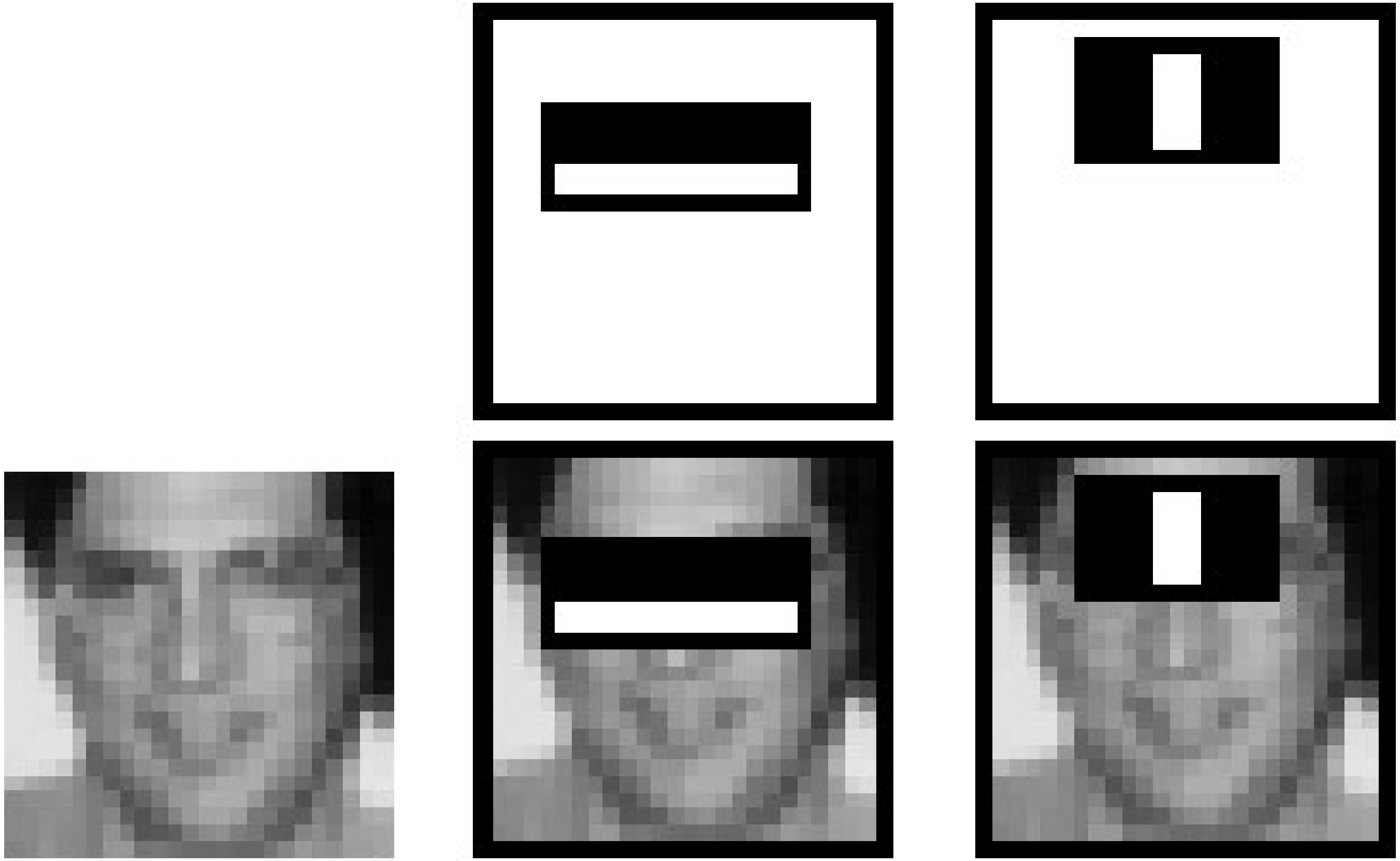
Best description of full details available in consolidated paper by Viola and Jones, International Journal of Computer Vision, 2004

Shift and Scale Invariance: Sliding Window Detectors

- image is tested for object presence window-by-window
- the window is `slided’ and `scaled’ throughout the image
- each resulting window is judged w.r.t. an object model giving a response indicating object presence or absence

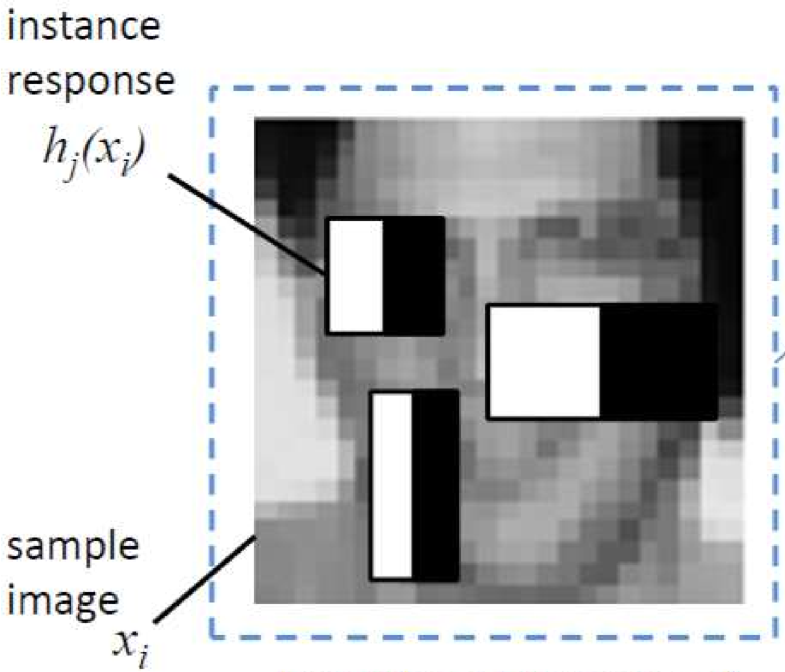


Basic Object Model Idea:
Characteristic Set of Block Features

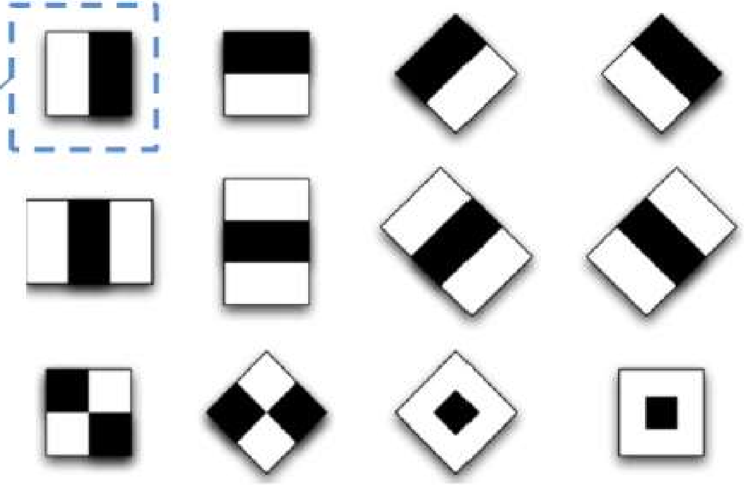


Viola & Jones' (2001)

Haar-like Features as Weak Classifiers



Examples of Instances of
1 Feature Type



Haar-like Feature Types

Integral Images & Integration Rule

(INTEGRATION RULE OF CONVOLUTION)

$$(\mathbf{S_k} * \mathbf{I})^{[n]} = \mathbf{S_k}^{[q]} * \mathbf{I}^{[p]} \quad \text{given} \quad n = p + q$$

I

	0	1	2	3	4	5	6	7
0	1	1	1	2	3	1	2	1
1	1	2	0	0	0	3	1	1
2	1	1	1	1	1	2	3	1
3	1	1	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	1

$\sum$

II

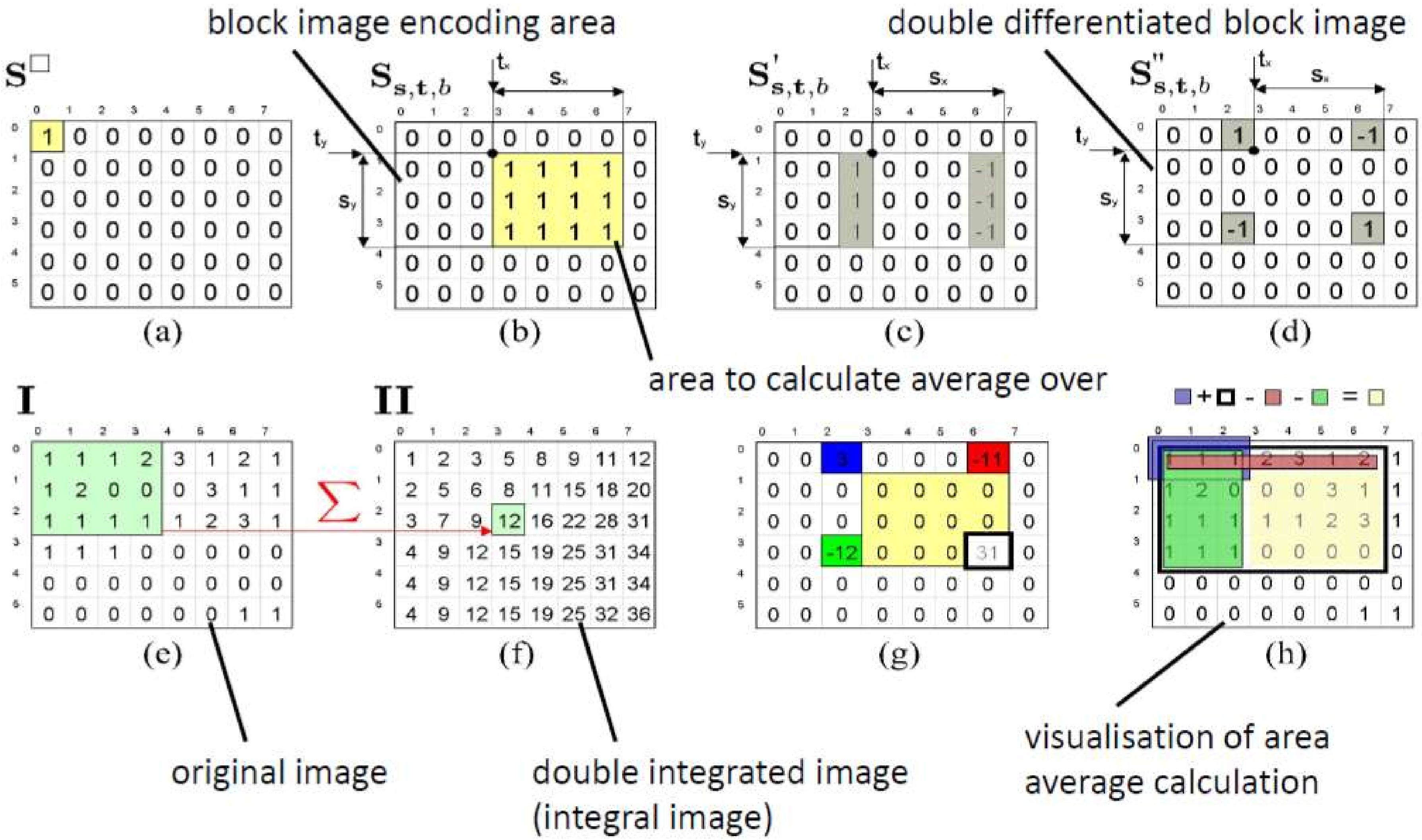
	0	1	2	3	4	5	6	7
0	1	2	3	5	8	9	11	12
1	2	5	6	8	11	15	18	20
2	3	7	9	12	16	22	28	31
3	4	9	12	15	19	25	31	34
4	4	9	12	15	19	25	31	34
5	4	9	12	15	19	25	32	36

(IMAGE INTEGRATION)

$$\mathbf{II}(-1, y) = 0; \qquad \mathbf{II}(x, y) = \mathbf{II}(x-1, y) + A(x, y);$$

$$A(x, -1) = 0; \qquad A(x, y) = A(x, y-1) + \mathbf{I}(x, y).$$

Calculating the Avg Pixel Value of Large Block Fast



Training Data

sample image x_i

Source: MIT CBCL

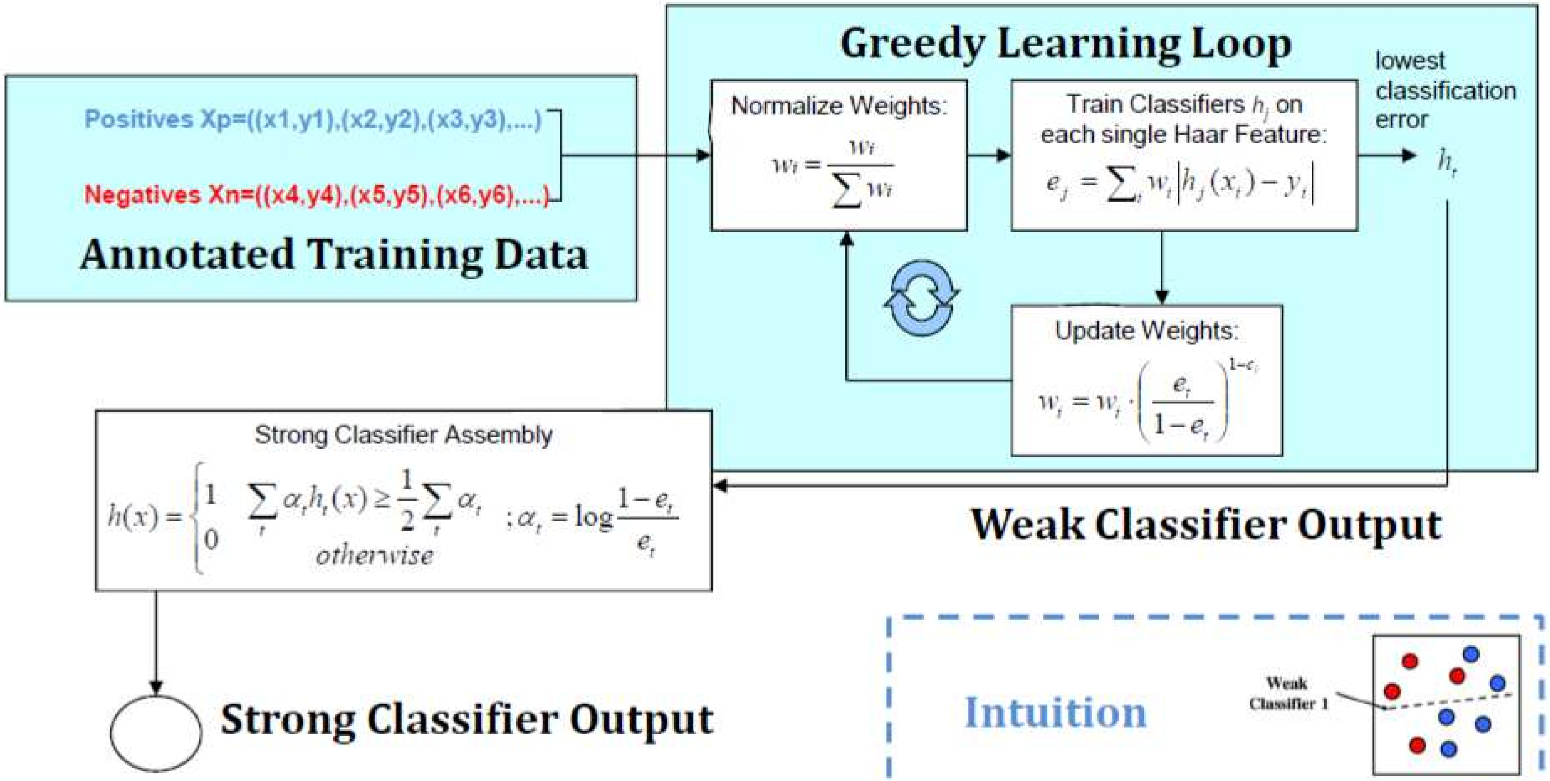
Positive Samples (e.g. FACE) ... $(x_i, y_i = 1), w_i = 1$

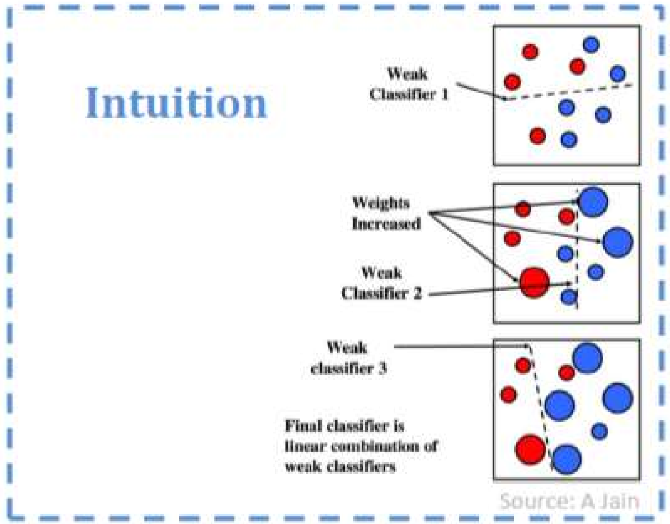
sample image x_i

Source: MDPI

Negative Samples (e.g. NO-FACE) ... $(x_i, y_i = 0)$, $w_i = 1$

Overview of Adaboost





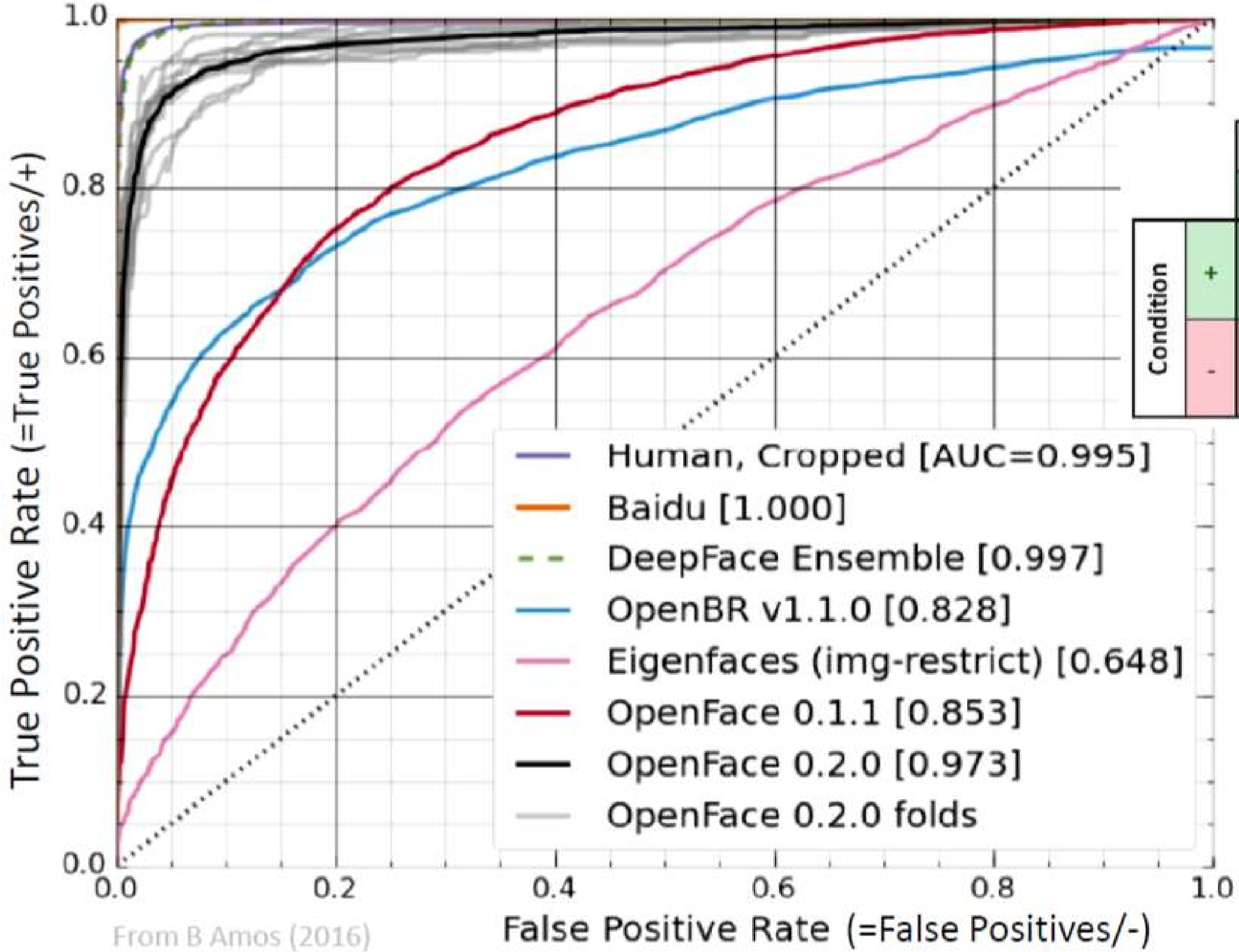
(also see paper by Viola and Jones 2004)

Adaboost Algorithm (from Viola & Jones 2001)

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  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 classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

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

Performance Considerations (Training or Testing)



		Classification	
		Positive	Negative
Condition	+	True Positive	False Negative
	-	False Positive	True Negative

From R Wingate (2016)

On Window Resolution