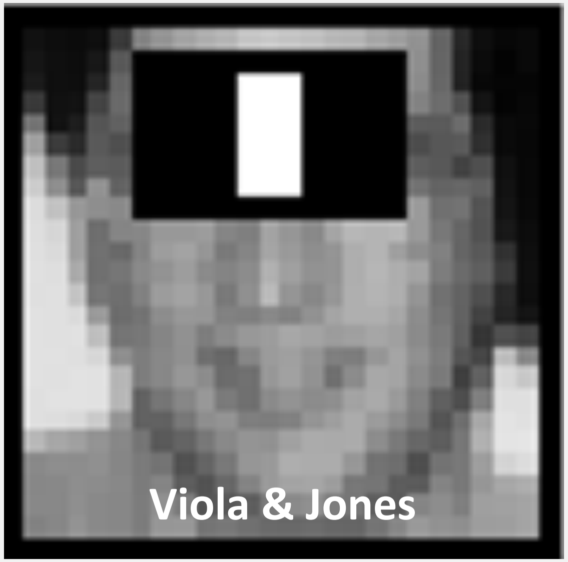
Department of Computer Science
University of Bristol

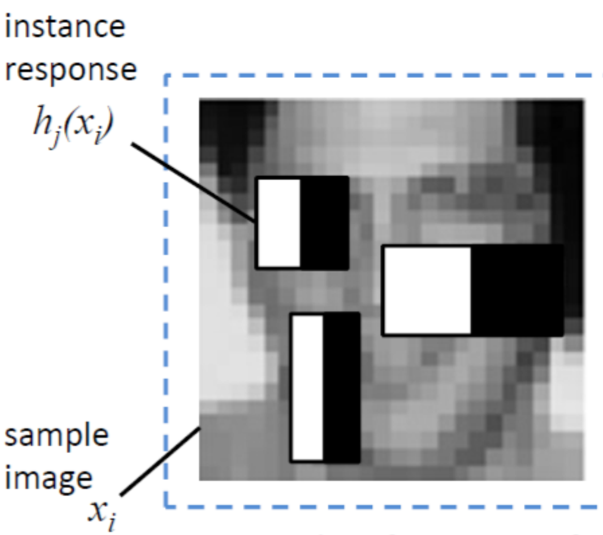
COMS30030 Image Processing and Computer Vision



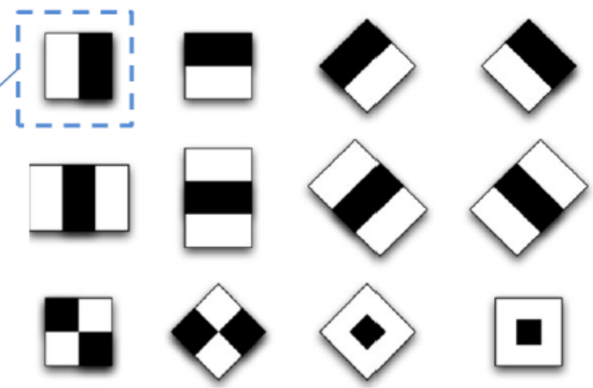
Lecture 11

Basics of Classical Object Detection

Example Algorithm:
Viola & Jones' Real-time Method (2001)



Examples of Instances of
1 Feature Type



Haar-like Feature Types

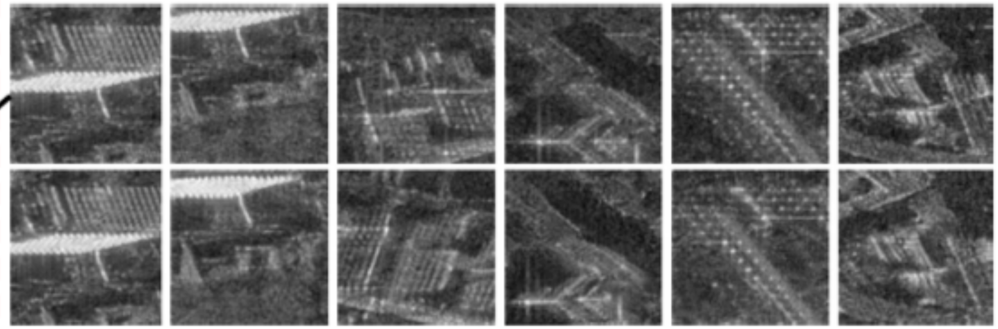
Training Data

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) … $(x_i, y_i = 0)$, $w_i = 1$

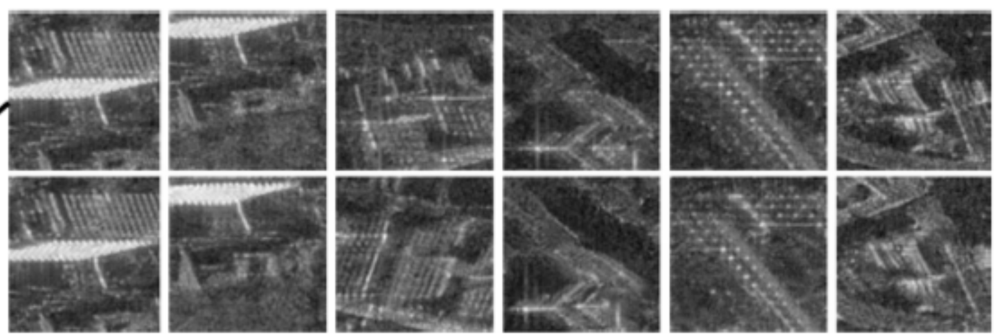
Training Data

sample image x_i

Source: MIT CBCL

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

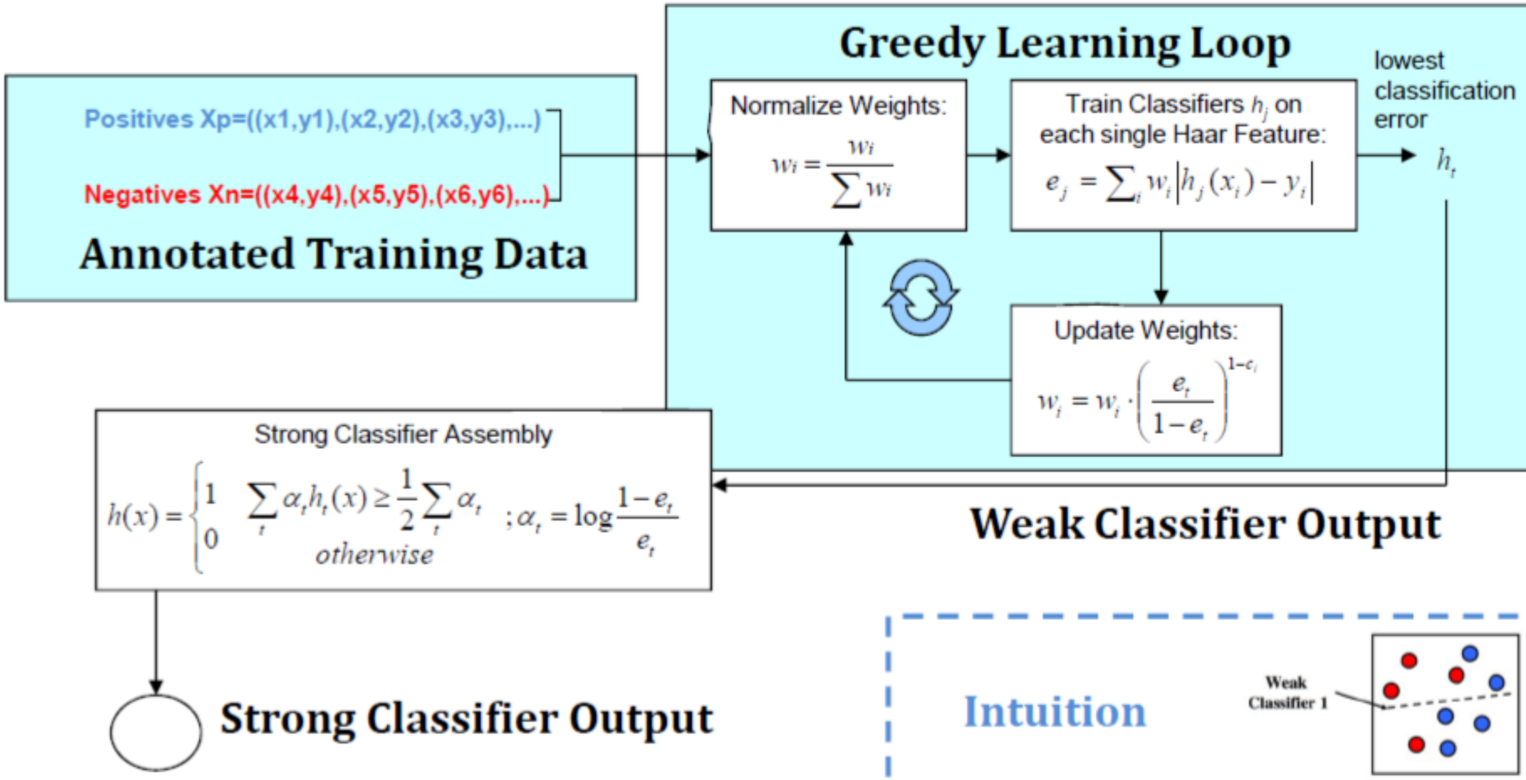
sample image x_i



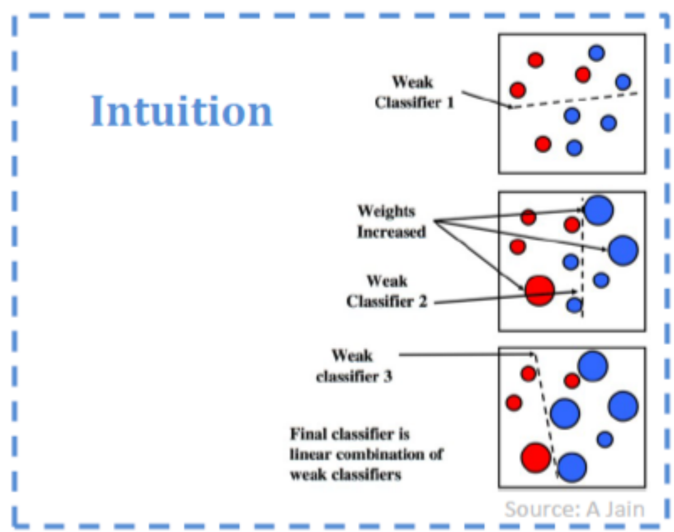
Source: MDPI

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, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  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 $\alpha_t = \log \frac{1}{\beta_t}$

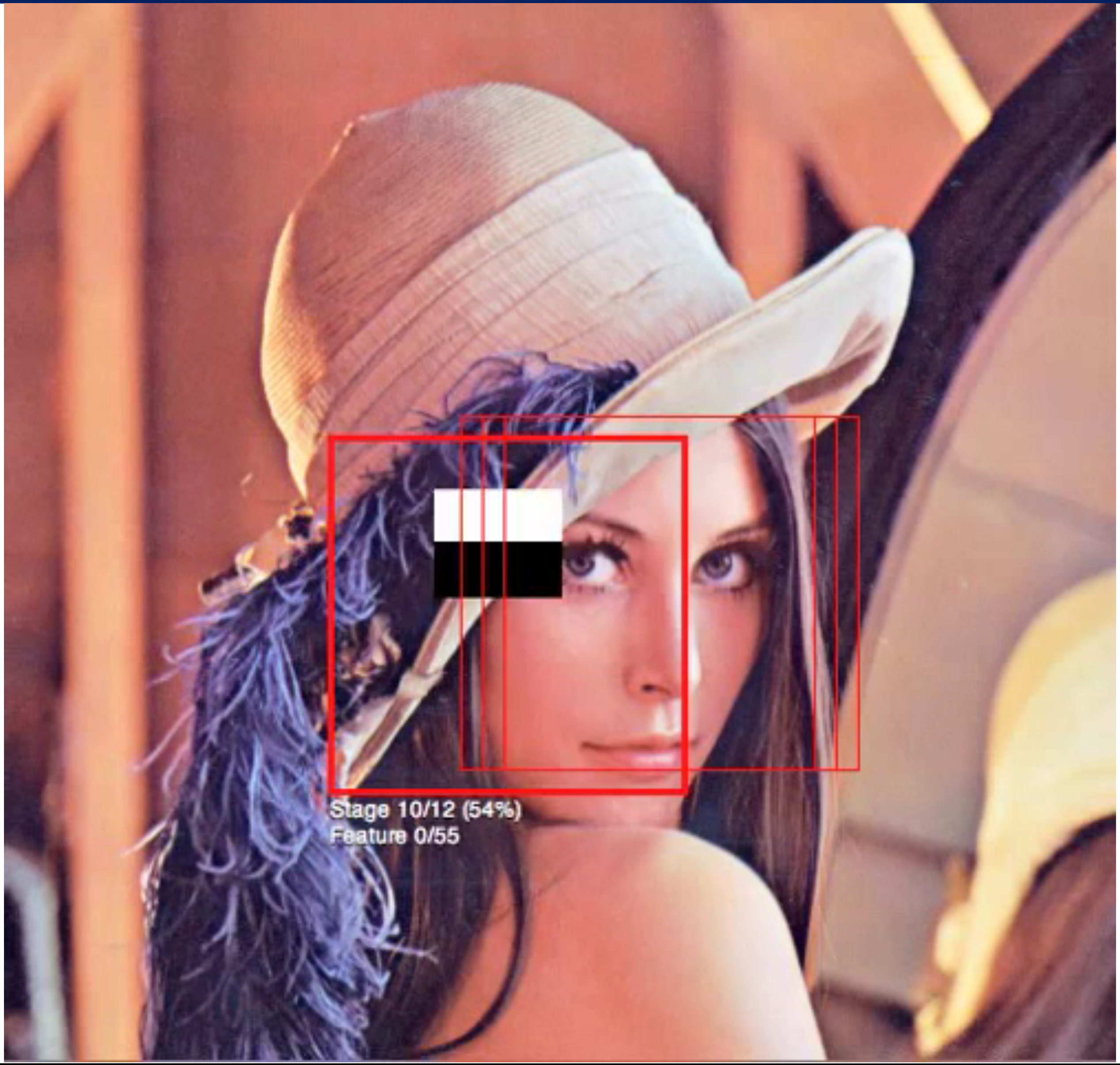
Adaboost Algorithm (from Viola & Jones 2001)

- 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:

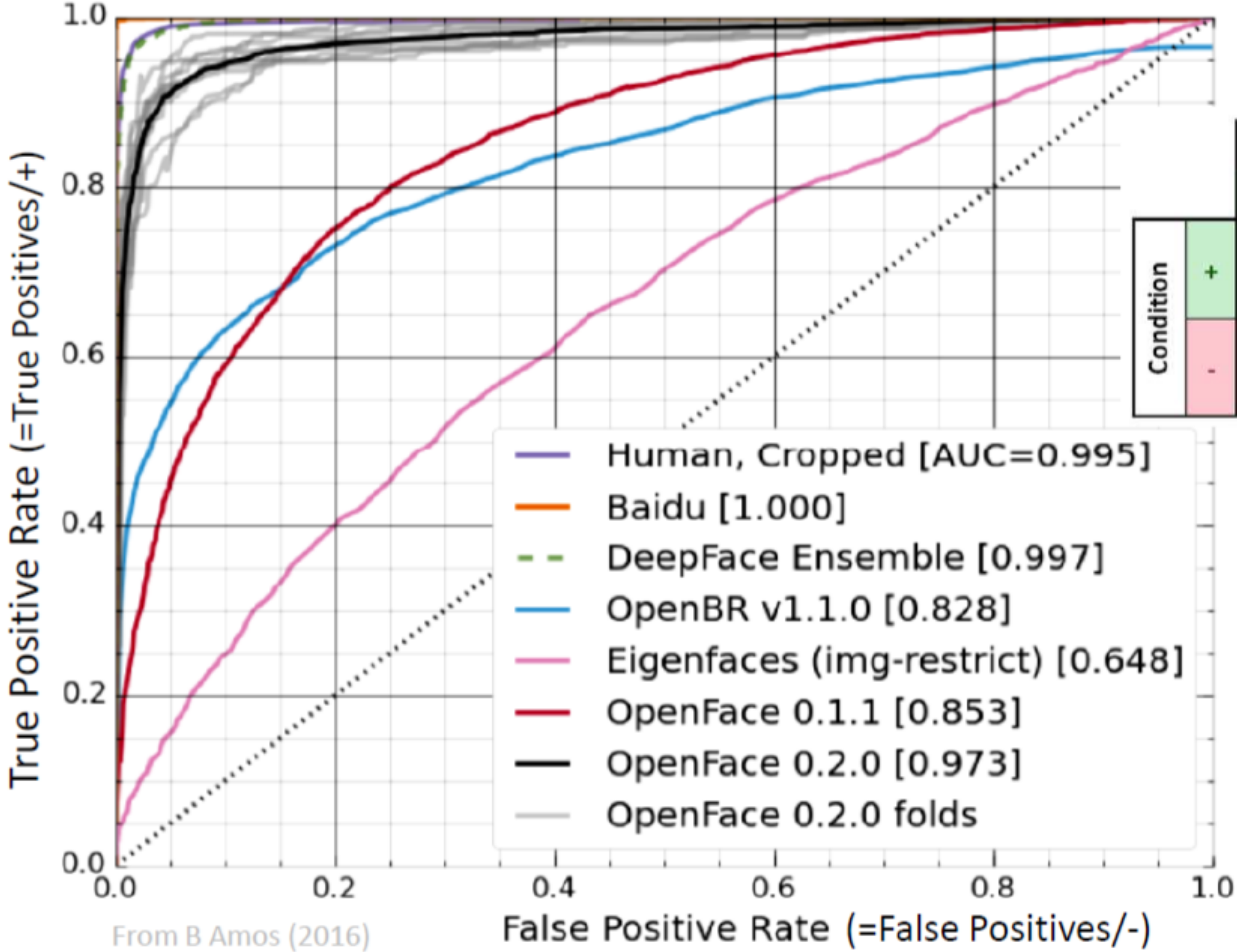
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

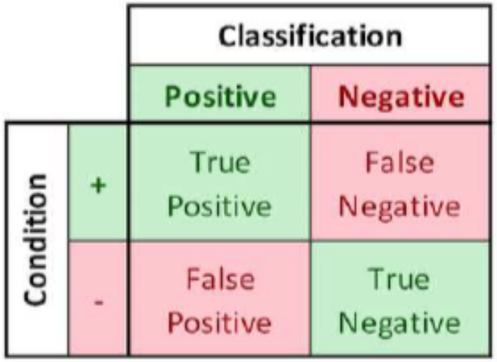
where $\alpha_t = \log \frac{1}{\beta_t}$

Visualisation



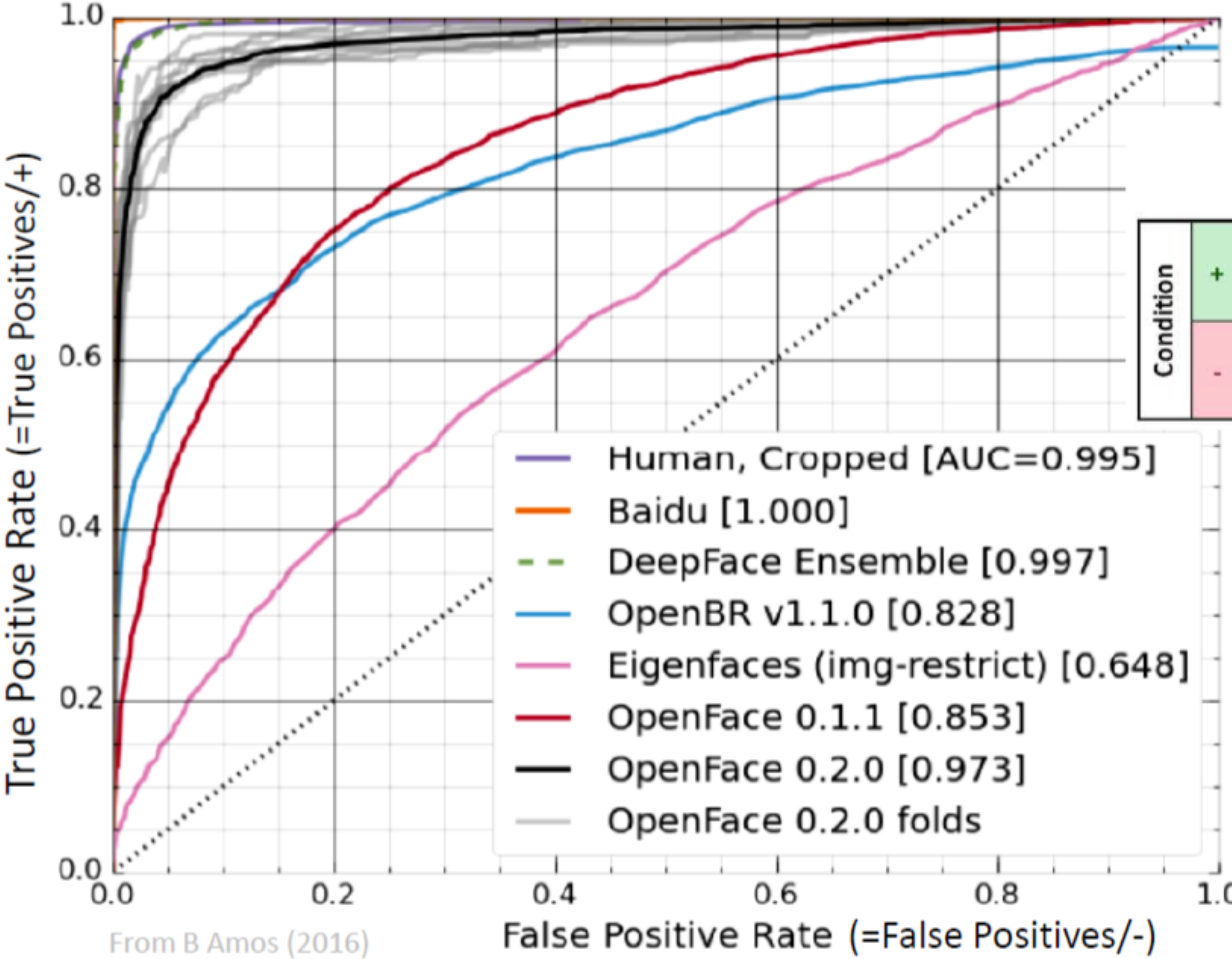
Performance Considerations (Training or Testing)





From R Wingate (2016)

Performance Considerations (Training or Testing)



From B Amos (2016)

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

From R Wingate (2016)

Conclusion

– Object detection is hard

– Object detectors learn from lots of data

– The Viola-Jones method:

1. Propose regions
2. Calculate features
3. Train a classifier