# Computer Vision

MSc Informatics option GVR

James L. Crowley

Fall Semester 22 November 2012

Lesson 7

# Learning Visual Pattern Detectors with Adaboost

**Lesson Outline:**

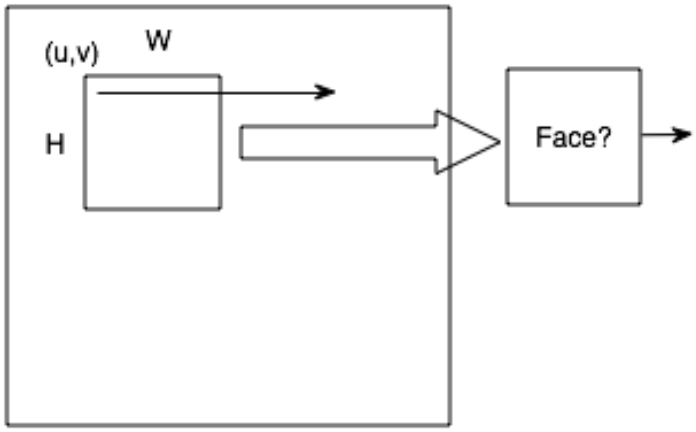

1. Processes Overview ..... 2
2. Image Description with Difference of Boxes ..... 3
   - 2.1. Box Features ..... 3
   - 2.2. Difference of Boxes ..... 3
   - 2.3. Haar Wavelets: ..... 4
3. Fast 2D Haar Wavelets using Integral Image ..... 6
   - 3.1. Integral Images ..... 6
   - 3.2. Fast Integral Image algorithm. ..... 7
4. Linear Classifiers for Face Detection ..... 8
   - 4.1. Training a committee of classifiers ..... 10
   - 4.2. Boosted Learning ..... 12
5. Learning a Committee of Classifiers with Boosting ..... 13
   - 5.1. ROC Curve ..... 13
6. Learning a Multi-Stage Cascade of Classifiers ..... 14

# 1. Processes Overview

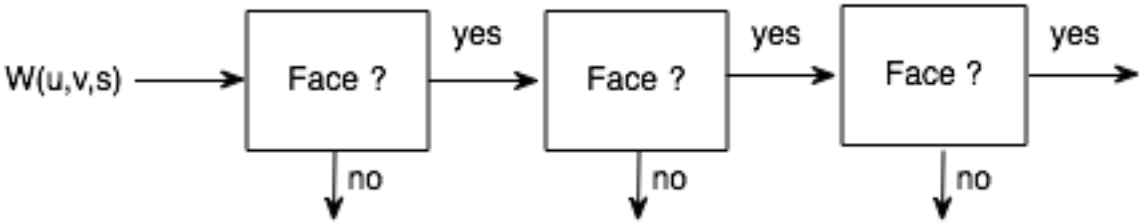
A Cascade of Classifiers detects faces with a window scanning approach.

An image window, *p(u, v, s)* is an (s x s) sized image window with its upper left corner at pixel (u, v). The image window is "texture mapped" using an affine transform into a starndard sized window, W(x,y) of size (WxH). This window is referred to as an "imagette".

For faces, the imagette is typically on the order of (24, 24) pixels.
(note - window does not have to be square).




The decision of whether the window *W(x,y)* contains a face is provided by a cascade of boosted linear classifiers.




The algorithm requires a large number of local "features" to classify the window. This can be provided by the Scaled Gaussian Derivatives seen in lectures 7 and 8.

They can also be provided by Haar wavelets computed using a Difference of Boxes.

## 2. Image Description with Difference of Boxes

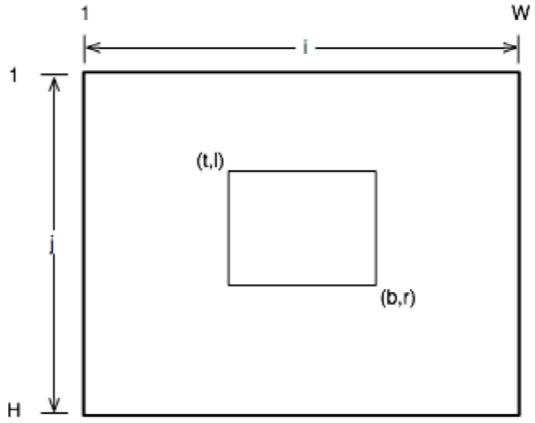
An image rectangle is defined by the top-left and bottom-right corner.
This may be represented by a vector (t, l, b, r). The sum of pixels in a rectangle defines a box feature.

### 2.1. Box Features

A box feature is the sum of pixels over a rectangle from top (t) to left (l) and bottom (b) to right (r)

With the constraints : top < bottom and right > left.

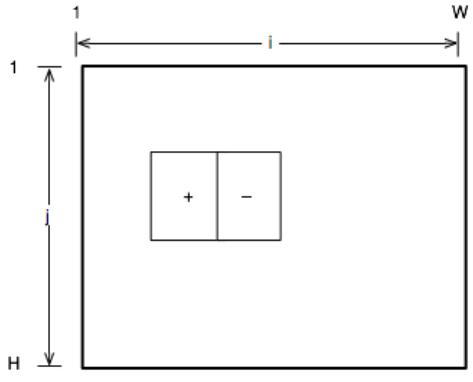
$$b(t,l,b,r) = \sum_{x=l}^{r} \sum_{y=t}^{b} W(x,y)$$




### 2.2. Difference of Boxes

A first order Difference of Boxes (DoB) feature is a difference of two boxes $box(t1,l1,b1,r1)$.

$$DoB(t1,l1,b1,r1,t2,l2,b2,r2) = box(t1,l1,b1,r1) - box(t2,l2,b2,r2)$$

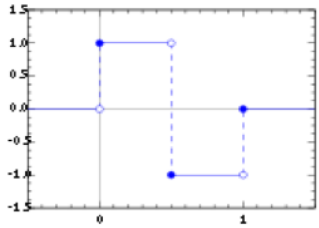


An interesting subclass are Difference of Adjacent Boxes where the sum of pixels is 0. These are Haar wavelets.

## 2.3. Haar Wavelets:

Haar A. Zur Theorie der orthogonalen Funktionensysteme, Mathematische Annalen, 69, pp 331–371, 1910.

The Haar wavelet is a difference of rectangular Windows.



The Digital (discrete sampled) form of Haar wavelet is

$$h(n;d,k) = \begin{cases} 1 & \text{for d} \le n < d + k/2 \\ -1 & \text{for d} + \text{k}/2 \le n < d + k \\ 0 & \text{for } n < d \text{ and } n \ge d + k \end{cases}$$

Haar wavelets can be used to define an orthogonal transform analogous to the Fourier basis. This can be used to define an orthogonal transform (the Walsh-Hadamard Transform). The basis is

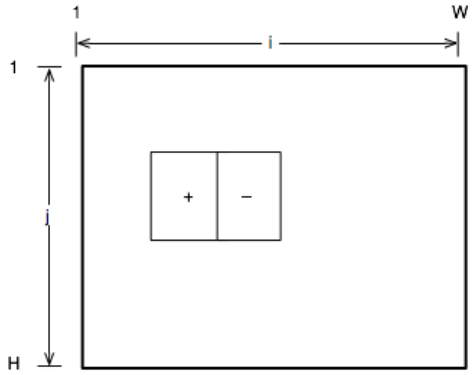
$$H_0 = +1 \qquad H_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad H_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad \ldots$$

$$H_m = \frac{1}{\sqrt{2}}\begin{bmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{bmatrix}$$

Haar Functions, and the Walsh-Hadamard transform have been used in Functional Analysis and signal processing for nearly a century.

In the 1980s the Wavelet community re-baptized the Haar functions as "wavelets" and demonstrated that the Walsh-Hadamard transform is the simplest form of wavelet transform.

A 2-D form of Walsh-Hadamard transform may be defined using DoB features using adjacent boxes. These can be calculated VERY fast using an algorithm known as Integral Images. They give a VERY large number of possible image features.




Assume a window is extracted from an image and mapped to the WxH imagette. Label the window coordinates (x, y) from [1,W] and [1,H]

Parameters:
1) The "polarity" of the difference ( [1 -1] or [-1 1])
1) order (number of adjacent boxes): 2nd or 3rd
2) orientation: vertical or horizontal
3) center position - ($c_x$, $c_y$) WxH possible positions
4) box size ($d_x$, $d_y$) (W/2)x(H/2) possible sizes

These can provide N image features. Label these with an integer index, *n*, $H_n(x,y)$
Note that each Haar wavelet corresponds to a specific position, size, and orientation in the imagette.

The product of each Haar wavelet $H_n(x,y)$ with the imagette $W(x,y)$ gives a number: $X_n$. This number is an image "feature" that describes the imagette.

$$X_n = \sum_{x=1}^{W} \sum_{y=1}^{H} W(x,y) H_n(x,y)$$

Given a WxH imagette of a face we can obtain N Feature numbers, $X_n$. Not all features are useful. We will use "machine learning to determine the subset of useful features for detecting faces.

Do not be confused by the reuse of W and H. W and H are the size of the imagette, W(x,y) is the imagette and $H_n(x,y)$ are the

# 3. Fast 2D Haar Wavelets using Integral Image

In 2001, Paul Viola and Mike Jones at MERL (Misubishi Research Labs) showed that Haar wavelets could be used for real time face detection using a cascade of linear classifiers.

They computed the Haar Wavelets (difference of adjacent boxes) for a window from integral images.

## 3.1. Integral Images

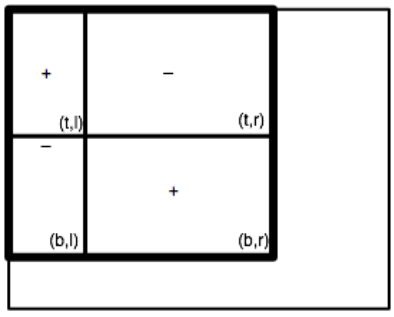
An integral image is an image where each pixel contains the sum from the upper left corner:

$$ii(u,v) = \sum_{i=1}^{u}\sum_{j=1}^{v} W(i,j)$$

An integral image provides a structure for very fast computation of 2D Haar wavelets.

Any box feature can be computed with 4 operations (additions/subtractions).

box(t,l,b,r)=ii(b,r)–ii(t,r)–ii(b,l)+ii(t,l)

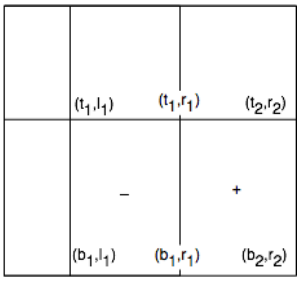



An arbitrary 1st order difference of boxes costs 8 ops.

$DoB(t_1,l_1,b_1,r_1,t_2,l_2,b_2,r_2) = box(t_1,l_1,b_1,r_1)–box(t_2,l_2,b_2,r_2)$
$=ii(b_1,r_1)–ii(t_1,r_1)–ii(b_1,l_1)+ii(t_1,l_1) – (ii(b_2,r_2)–ii(t_2,r_2)–ii(b_2,l_2)+ii(t_2,l_2))$

However, a 1st order Haar wavelet costs only 6 ops because $r_1=l_2$ and thus

$ii(t_1,r_1) = ii(t_2,l_2)$ and $ii(b_1,r_1)= ii(b_2,l_2)$



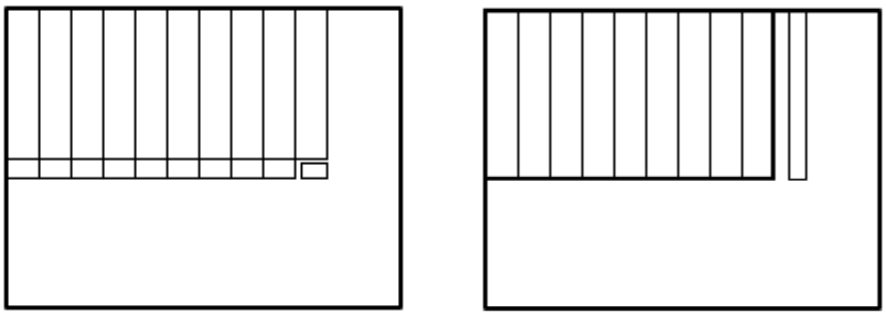

$$Haar(t_1,l_1,b_1,r_1,b_2,r_2) = ii(b_2,r_2)–2ii(b_1,r_1)+ii(b_1,l_1)–ii(t_2,r_2)+2ii(t_1,r_1)–ii(t_1,l_1)$$

### 3.2. Fast Integral Image algorithm.

Integral images have been used for decades to compute local energy for normalization of images. A fast recursive algorithm for computing the integral image makes use of a buffer, c(i). The buffer keeps a running sum of each column.

```
For j = 1 to H
For i = 1 to W
{       c(i) = c(i) +  p(i,j)
        ii(i,j) = ii(i-1,j) + c(i) }
```

Cost = 2WH ops.



## 4. Linear Classifiers for Face Detection

The innovation in the Viola-Jones face detector resulted from

1) A very large number of very simple features (Haar wavelets).
2) The use of the Adaboost learning algorithm to learn an arbitrariy good detector.

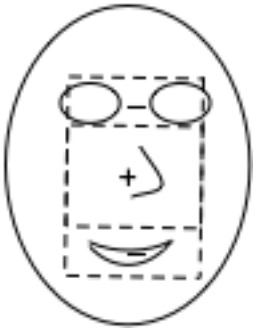
HAAR wavelets are computed using difference of Boxes, with Integral Images.

A WxH imagette contains $W^2H^2/4$ possible 1st order Haar wavelets $H_n$
(difference of adjacent boxes of same size ).


Similarly, any 2nd order Haar wavelet can be computed with 8 ops.

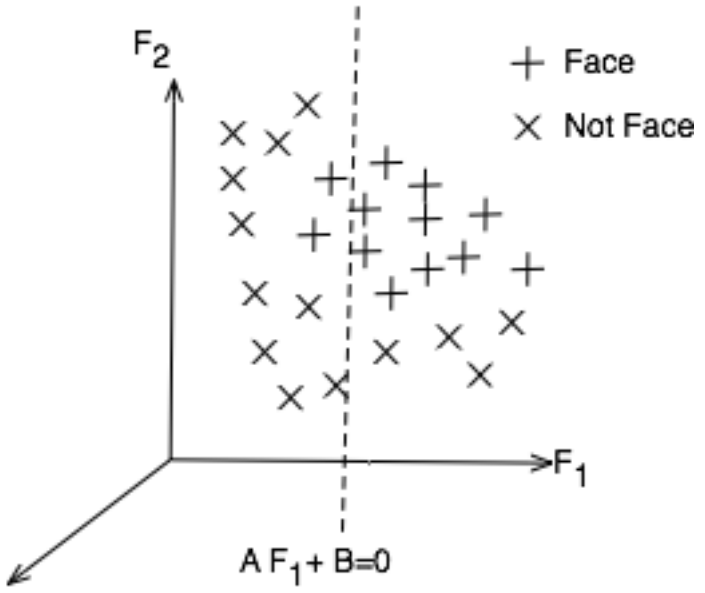

Each feature, $X_n$ is defined as the product of a Haar wavelet with the image window.

$$X_n = \sum_{x=1}^{W} \sum_{y=1}^{H} W(x,y) H_n(x,y)$$




Some features respond to the appearance of a face. These can be used to determine if the imagette contains a face or not.

Given an image of a Face (F), and a set of Haar wavelets $H_n$




Each feature can be used to define a hyper-plane $<W, H_n> + B = 0$.

where $\langle W, H_n \rangle = \sum_{x=1}^{W} \sum_{y=1}^{H} W(x,y), H_n(x,y)$

B is a "bias" that shifts the plane along the $H_n$ axis.

this can be noted as $<WH_n> + B > 0$ or simply $WH_n + B > 0$

B is a global Bias that determines the tradeoff between False Positives and False Negatives. The problem is to choose the best $H_n$ so that most non-face windows are on one side of the hyperplane and most face windows are on the other.

To do this we will use a "training" set of M imagette, $\{W_m\}$. some of which contain faces. We will note whether the imagette contains a face with an "indicator variable" $y_m$.

For imagettes that contain faces, $y_m = 1$. Imagettes that do not contain faces, $y_m = -1$.

### 4.1. Training a committee of classifiers

Assume a very large set of M face windows $\{W_m\}$ that have been labeled by a set of labels $\{y_m\}$ such that y=+1 if face and y=–1 if not face,

Then for an imagette, $W_m$, each feature "votes" for a face (P for positive) or not a face (N for negative).

if $W_m H_n + B > 0$ then P else N.

Whether this vote is true (T) of false (F) can be determined by the indicator variable.

if $(W_m H_n + B) \cdot y_m > 0$ then T else F.

For the training set $\{W_m\}$, the error rate for the feature $H_n$ is

$$E_n = \text{Card}\{(W_m H_n + B) \cdot y_m < 0\}$$

(Card is the cardinality operator - it counts the number of times something happens)

The error rate is composed of two parts : False Positives and False Negative.

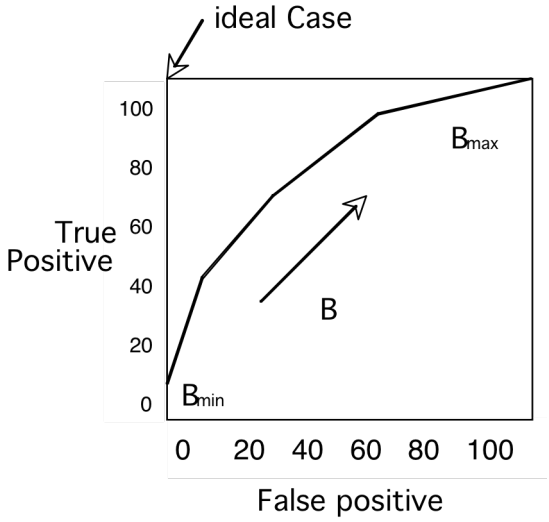
$$FP_n = \text{Card}\{(W_m H_n + B) > 0) \text{ and } (y = -1)\}$$
$$FN_n = \text{Card}\{(W_m H_n + B) < 0) \text{ and } (y = +1)\}$$
$$E_n = FP_n + FN_n$$

note that the number of true positives (TP) is TP = 1 – FP

We can trade FPs for FNs by adding to the global Bias B,

For a feature $H_n$

$$FP = \text{Card}\{ (W_m H_n + B) > 0 \text{ and } y_m = -1\}$$
$$FN = \text{Card}\{ (W_m H_n + B) < 0 \text{ and } y_m = +1\}$$




These are plotted in a graph called an "ROC" or Receiver Operating Characteristics Graph.

## 4.2. Boosted Learning

To boost the learning, after selection of each "best" classifier, $(F_n, B_n)$
we re-weight the incorrectly classified training samples with a weight, $a_m$ to increase the weight for incorrectly classed imagettes:

For all m = 1 to M if $(W_m H_n + B) \cdot y_m^{(i-1)} < 0$ then $a_m^{(i)} = a_m^{(i-1)} + 1$

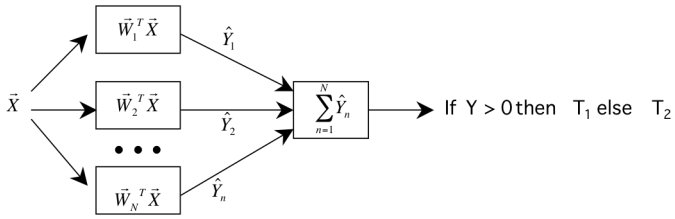
We then learn the $i^{th}$ classifier from the re-weighted set

$E_{min} = M$
For n=1 to N do
$E_n = \text{Card}\{a_m^{(i)}(W_m, H_n) \cdot y_m < 0\}$
if $E_n < E_{min}$ then $E_{min} := E_n$

Haar features are removed from the set after being used.

# 5. Learning a Committee of Classifiers with Boosting

We can improve classification by learning a committee of the best I classifiers.




The decision is made by voting. An imagettes W is determined to be a Face if the majority of classifiers (features) vote > 0.

$$\text{If } \sum_{i=1}^{I} W_m H_n + B > 0 \text{ then Face else Not-Face.}$$

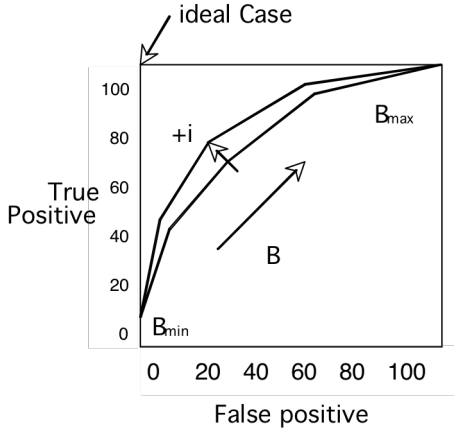
## 5.1. ROC Curve

We can describe a committee of classifiers with an ROC curve, but defining a global bias, B. The ROC describes the number of False Positives (FP) and False Negatives (FN) for a set of classifier as a function of the global bias B.

$$FP = \text{Card}\{(W_m H_n + B) > 0 \text{ and } y_m = -1\}$$

$$FN = \text{Card}\{(W_m H_n + B) < 0 \text{ and } y_m = +1\}$$

The Boosting theorem states that adding a new boosted classifier to a committee always improves the committee ROC curve. We can continue adding classifiers until we obtain a desired rate of false positives and false negatives.

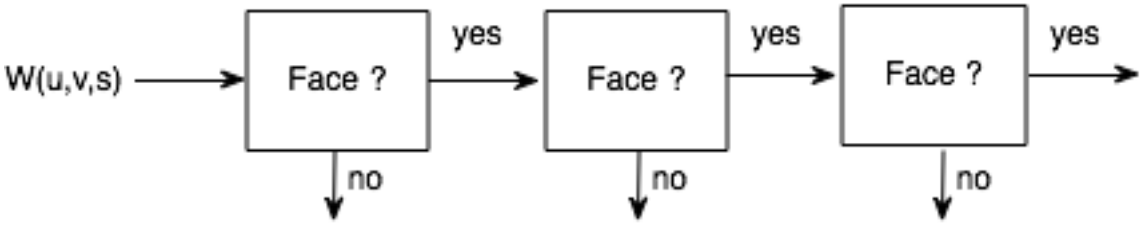


## 6. Learning a Multi-Stage Cascade of Classifiers

We can optimize the computation time by using a multistage cascade.

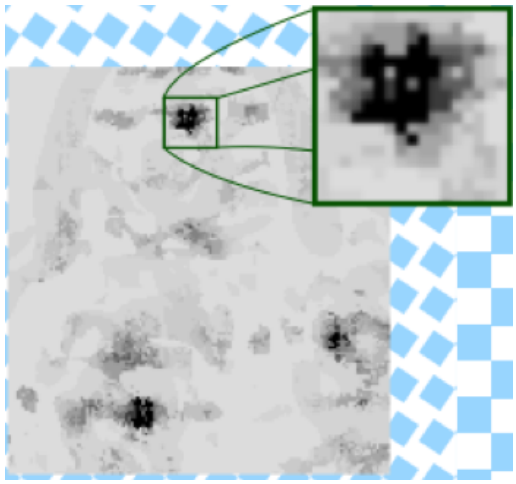
Algorithm:

1) Set a desired error rate for each stage j : ($FP_j$, $FN_j$).
2) For j = 1 to J
For all windows labeled as Face by j-1 stage, learn a boosted committee of classifiers that meets ($FP_j$, $FN_j$).




Each stage acts as a filter, rejecting a grand number of easy cases, and passing the hard cases to the next stage.

This is called a "cascade classifier"
Note that applying this to every position gives an "image" of cascade depths.



Faces can be detected as the center of gravity of "deep" detections.
Faces can be tracked using the Bayesian tracking described in the previous session.

This algorithm is part of the OpenCV tool box. It is widely used in digital cameras and cell phones for face detection and tracking.