**Haar Cascades**

Haar cascade is based on “Hare wavelets” which refers to a sequence of rescaled “square-shaped” functions which together form a wavelet family or basis.

Cascading training is done to extract the important features in the image for example in case of face detection eyes, nose etc. are the important features of a face.

Haar Cascade uses positive and negative images to train the classifier.

* **Positive images** – These images contain the images which we want our classifier to identify.
* **Negative Images** – Images of everything else, which do not contain the object we want to detect.

The algorithm has four stages:

1. Haar Feature Selection
2. Creating  Integral Images
3. Adaboost Training
4. Cascading Classifiers

**Haar Feature Selection:**

In this step, we divide the image into adjacent rectangular subsections and select important rectangle by considering the sum of intensities of all the pixels under the area.

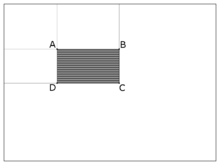
The difference between these region are then used to specify the important features in the image.  For example, let us say we have an image database with human faces. It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region.

**Creating Integral Images:**

Integral image is a matrix used to store the summarized data of a region in one cell. It makes the calculation of sum of intensity required in the previous step efficient and easy.

Each element of the integral image contains the sum of all pixels located on the up-left region of the original image (in relation to the element's position). This allows to compute sum of rectangular areas in the image, at any position or scale, using only four lookups:

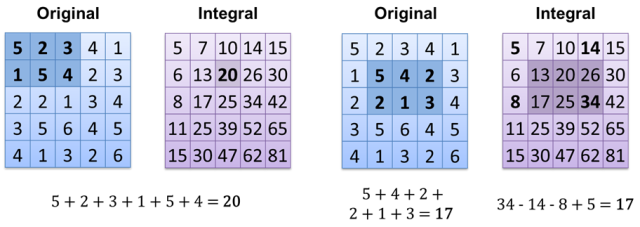
Sum = I(C) + I(A) – I(B) –I(D)

[](https://en.wikipedia.org/wiki/File:Prm_VJ_fig3_computeRectangleWithAlpha.png)

Finding the sum of the shaded rectangular area

{\displaystyle {\text{sum}}=I(C)+I(A)-I(B)-I(D).\,}

For example:



**Adaboost training:**

This algorithm is used to select small number of important features from a large set of features generated from the previous steps. By using this technique the predictive power of the model is increased, also it reduces dimensionality and improve execution time of machine learning classification model.

**Cascading Classifiers:**

It uses machine learning classifier models to predict the probability of input to be in a class and take the output of previous classifier to serve as input to the next model( cascading ).

For example: Let’s say we want to classify our image into two classes class1 and class2 based on three features: feature1, feature2 and feature3.

The classification rule says if atleast two classes are negative than the predicted class is class1 otherwise it is class2.

We can construct if-else rules as:

If(feature1==positive){

If(feature2==positive){

Predict class2

}

Else{

If(feature3==positive){

Predict class2

}

Else{

Predict class1

}

}

}

Else{

If(feature2==negative){

Predict class1

}

Else{

If(feature3==positive){

Predict class2

}

Else{

Predict class1

}

}

}

