Cascade Classifier

Object Detection using Haar feature-based（註 1） cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.
使用基於Haar特徵的級聯分類器進行物體檢測是Paul Viola和Michael Jones在2001年發表的論文 "Rapid Object Detection using a Boosted Cascade of Simple Features "中提出的一種有效的物體檢測法。它是一種基於機器學習的方法，从大量的正負圖像中訓練出一個級聯函數。然後用它來檢測其他圖像中的物體。

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.
在這裡，我們將致力於人臉檢測。最初，算法需要大量的正向圖像(人臉的圖像)和負向圖像(沒有人臉的圖像)來訓練分類器。然後我們使用了下圖所示的Harr特徵。他們就像我們的卷積核(convolutional kernal)一樣。每個這特徵都是由白色矩形下的像素之和減去黑色矩形下的像素之和得到一個值。

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.
現在，每个內核的所有可能的大小和位置都被用來計算很多特征。(試想一下，這需要多少計算量？即使是一个24x24的窗口也 產生超過160000个特徵)。) 對於每个特征計算，我們需要找到白色和黑色矩形下的像素之和。為了解決這个問題，他們引入了積分圖像。無論你的圖像有多大，它都能將給定像素的計算減少到只涉及四个像素的操作。它讓事情變得超級快

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by Adaboost（註 2）.

但在我們計算的這些特徵中，大部分都是無關緊要的。例如，考慮下面的圖像。最上面一行顯示了兩個好的特徵。選擇的第一个特征似乎集中在眼睛的區域通常比鼻子和臉頰的區域更深的属性上。選擇的第二个特征依賴于眼睛比鼻梁更深的属性。但同　的窗口應用于臉頰或其他任何地方是無關緊要的。那麼，我們如何在160000多个特征中選擇最佳特征呢？這是由Adaboost（註 2）實現的。

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found).

為此，我們在所有的訓練圖像上應用每一个特征。對於每一个特征，它都會找到最佳的閾值，从而將人臉分類為正面和負面。顯然，會有錯誤或誤分類。我們選擇錯誤率最小的特徵，也就是說它們是最能準確分類人臉和非人臉圖像的特徵。(這个過程并不是這么簡單。一開始，每張圖像都被賦予同等的權重。在每次分類後，增加被誤分類圖像的權重。然後再做同樣的過程。計算出新的錯誤率。也是新的權重。這个過程一直持續到達到所需的准确率或錯誤率，或者找到所需的特征數量）。)

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

最後的分 器是這些弱分類器的加權和。之所以稱為弱，是因為它單獨不能對圖像進行分類，但與其他分類器一起構成了一個強分類器。論文說，即使是200个特徵也能提供95%准确率的檢測。他們最終的設置有大約6000个特徵。想像一下，從160000多个特徵減少到6000个特徵。這是一个很大的收穫)。)

So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. Wow.. Isn't it a little inefficient and time consuming? Yes, it is. The authors have a good solution for that.
In an image, most of the image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot, and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.
（太廢話了，直接看）

For this they introduced the concept of Cascade of Classifiers. Instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. How is that plan!
為此，他們引入了Cascade of Classifiers的概念。而不是將6000个特征全部應用在一个窗口上，而是將特征分成不同階段的分類器，并逐一應用。通常前幾個階段包含的特徵會非常少)。如果一個窗口沒有通過第一階段，就丟棄它。我們不考慮它的剩餘特徵。如果它通過了，則應用第二階段的特徵並繼續這個過程。通過所有階段的窗口就是一個面區域。

The authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages. (The two features in the above image are actually obtained as the best two features from Adaboost). According to the authors, on average 10 features out of 6000+ are evaluated per sub-window.
So this is a simple intuitive explanation of how Viola-Jones face detection works.
作者的檢測器有6000多个特徵，38个階段，前五個階段有1、10、25、25和50个特徵。(上圖中的兩個特徵其實是Adaboost中獲得的最好的兩個特徵)。劇作者介紹，平均每个子窗口6000多个特徵中，有10个特徵被評估。
所以這是對Viola-Jones人臉檢測工作原理的一個簡單直觀的解釋。

(註 1)

Haar-like feature

A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, with a human face, 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. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

In the detection phase of the Viola – Jones object detection framework, a window of the target size is moved over the input image, and for each subsection of the image the Haar-like feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because such a Haar-like feature is only a weak learner or classifier (its detection quality is slightly better than random guessing) a large number of Haar-like features are necessary to describe an object with sufficient accuracy. In the Viola – Jones object detection framework, the Haar-like features are therefore organized in something called a *classifier cascade (級聯分類器)* to form a strong learner or classifier.

The key advantage of a Haar-like feature over most other features is its calculation speed. Due to the use of *integral images*, a Haar-like feature of any size can be calculated in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature).

（註2）

AdaBoost

AdaBoost方法的自適應在於：前一個分類器分錯的樣本會被用來訓練下一個分類器。

AdaBoost方法對於噪聲數據和異常數據很敏感。但在一些問題中，AdaBoost方法相對於大多數其它學習算法而言，不會很容易出現過擬合現象。AdaBoost方法中使用的分類器可能很弱（比如出現很大錯誤率），但只要它的分類效果比隨機好一點（比如兩類問題分類錯誤率略小於0.5），就能夠改善最終得到的模型。而錯誤率高於隨機分類器的弱分類器也是有用的，因為在最終得到的多個分類器的線性組合中，可以給它們賦予負係數，同樣也能提升分類效果。

AdaBoost方法是一種疊代算法，在每一輪中加入一個新的弱分類器，直到達到某個預定的足夠小的錯誤率。每一個訓練樣本都被賦予一個權重，表明它被某個分類器選入訓練集的概率。如果某個樣本點已經被準確地分類，那麼在構造下一個訓練集中，它被選中的概率就被降低；相反，如果某個樣本點沒有被準確地分類，那麼它的權重就得到提高。通過這樣的方式，AdaBoost方法能「聚焦於」那些較難分（更富信息）的樣本上。在具體實現上，最初令每個樣本的權重都相等，對於第k次疊代操作，我們就根據這些權重來選取樣本點，進而訓練分類器C_k。然後就根據這個分類器，來提高被它分錯的樣本的權重，並降低被正確分類的樣本權重。然後，權重更新過的樣本集被用於訓練下一個分類器C_k[2]。整個訓練過程如此疊代地進行下去。

Cascading classifiers (from Wikipedia)

Cascading is a particular case of ensemble learning based on the concatenation of several classifiers, using all information collected from the output from a given classifier as additional information for the next classifier in the cascade. Unlike voting or stacking ensembles, which are multiexpert systems, cascading is a multistage one.

Cascading classifiers are trained with several hundred "positive" sample views of a particular object and arbitrary "negative" images of the same size. After the classifier is trained it can be applied to a region of an image and detect the object in question. To search for the object in the entire frame, the search window can be moved across the image and check every location for the classifier. This process is most commonly used in image processing for object detection and tracking, primarily facial detection and recognition.

The first cascading classifier was the face detector of Viola and Jones (2001). The requirement for this classifier was to be fast in order to be implemented on low-power CPUs, such as cameras and phones.

級聯是一種特殊的基於多個分類器聯接的集合學習案例，將從給定分類器的輸出中收集到的所有信息作為級聯中下一個分類器的附加信息。與投票或堆疊式合奏不同的是，級聯是一个多專家系統，是一个多階段系統。

級聯分類器是用几百个特定對象的 "正 "樣本試圖和相同大小的任意 "負 "圖像來訓練的。分類器訓練完成後，就可以應用於圖像的某个區域，並檢測出相關的對象。為了在整个幀中搜索對象，可以在圖像上移動搜索窗口，並為分 器檢查每個位置。這個過程在圖像處理中最常用於物体檢測和跟蹤，主要是面部檢測和識別。

第一個級聯分類器是 Viola 和 Jones（2001）的人臉檢測器。這種分類器的要求是快速，以便在低功耗的 CPU 上實現，如相機和手機。