Face detection using pi camera

工作原理

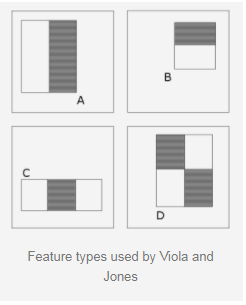
臉部偵測(Face Detection)基本作法是先做特徵擷取(Feature Extraction)，接著偵測檢查特徵(Cascade Detection)，進行特徵部位的辨識，如眼睛、耳朵、鼻子、眉毛、臉型。

這邊OpenCV人臉偵測的演算法，是採用Paul Viola與Michael Jones發表的Robust Real-Time Face Detection實現，該演算法分為四個階段 :

1.使用Haar-like特徵檢測人臉特徵 : 所有的人臉都有一些相似的特徵，透過

Haar-like定義的四種檢測視窗，對待識別的圖片進行掃描，並根據檢測框

架中的黑色區域與白色區域之比例，計算出該掃描區域的特徵值。



2.使用積分圖（Integral Image）加快對Haar-like特徵求值的速度:由於每一塊

檢測視窗中掃描後的Haar-like特徵值，擁有數以萬計的排列組合，如以Viola

牛提出的最基本四個特徵為例，在一個24×24大小的視窗中的任意排列至

少可以產生10萬種的特徵，因此，面對這麼龐大的計算量，積分圖可以求

出圖像中所有區域像素和的算法，進而提高了圖像特徵值計算的效率。

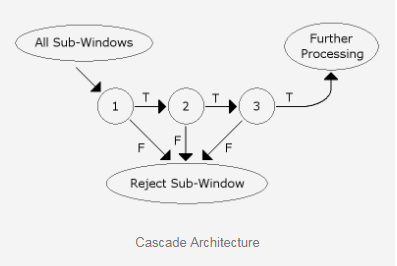
3.用AdaBoost算法區分人臉和非人臉的強分類器 : 已檢測的人臉特徵

值作為一個樣本，每一份樣本視為一個弱分類，AdaBoost便是把弱分類集

合， 轉換成強分類，幫助我們建立對人臉偵測更加嚴謹的辨識基準。

4.用Cascade Classifier把強分類器級聯一起，提高人臉辨識的準確率

一開始將特徵(feature)分成好幾個classifier。最前面的classier辨識率最低，但是可以先篩選掉不是人臉的圖片；接下來的Classier處理比較難篩選掉的圖片依此類推，直到最後一個classier為止，留下來的就會是確定有人臉的照片。



[OpenCV] 人臉偵測 (Face Detection)

<https://cg2010studio.com/2011/04/26/opencv-人臉偵測face-detection/>

[OpenCV] 人臉偵測2 (Face Detection)

<https://cg2010studio.com/2012/04/28/opencv-人臉偵測2-face-detection/>

人臉偵測Face Detection 使用OpenCV 2.4.2

<http://mark-jo-prog.blogspot.tw/2012/08/face-detection-opencv-242.html>

OpenCV 2.4 的人臉偵測（Face Detection）cvHaarDetectObjects 函數

<http://bugworkshop.blogspot.tw/2015/10/opencv-24-face-detection.html>

OpenCV,人臉辨識 (OpenCV, Python, Face Recognition)

<http://gogoprivateryan.blogspot.tw/2015/09/opencv-3-opencv-python-face-recognition.html>

淺析人臉檢測之Haar分類器方法

<http://alex-phd.blogspot.tw/2014/03/haarhaar-adaboost.html>

Face Detection using Haar Cascades

<https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html>

Robust Real-Time Face Detection

<http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf>

參考程式 :

1. 使用說明 : 先確認OpenCV 放置Cascade Classifier的路徑，若無更改路徑會發生編譯錯誤的訊息，這邊使用的OpenCV預設路徑為:

/home/pi/opencv-3.0.0/data/haarcascades/haarcascade\_frontalface\_alt.xml'

2.靜態辨識 : 透過Picamera鏡頭拍攝圖像後，人臉辨識結果以圖片方式呈現。

-程式碼修改

-程式碼參考

<https://pythonprogramming.net/raspberry-pi-camera-opencv-face-detection-tutorial/>

1. 動態辨識 : 透過Picamera鏡頭以錄影方式，進行人臉辨識。

-程式碼修改

-程式碼參考

<https://github.com/AsankaD7/Raspberry-Pi-Camera-Face-Detection-Python/blob/master/faceDetectPiVideo.py>

Haa-like特徵分類器訓練

1. 實作教學

Creating your own Haar Cascade OpenCV Python Tutorial

<https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/>

Tutorial: OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features)

<http://note.sonots.com/SciSoftware/haartraining.html#v6f077ba>

TRAIN YOUR OWN OPENCV HAAR CLASSIFIER

<http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>

1. 實作步驟

分類器訓練分為下列三個步驟

1. 資料準備 : 正面圖像樣本與負面圖像樣本
2. 樣本創建 : OpenCV推荐训练样本的最佳尺寸是20x20，但是在下一步生成样本描述文件时可以轻松地将其它尺寸缩放到20x20。

2.1資料準備

1. 正面圖像樣本-

指的是需要識別的圖片

opencv\_createsamples -img watch5050.jpg -bg bg.txt -info info/info.lst -pngoutput info -maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 -num 1950

1. 負面圖像樣本

所谓负样本，是指不包含待识别物体的任何图片，因此你可以将天空、海滩、大山等所有东西都拿来当负样本。但是，很多时候你这样做是事倍功半的。大多数模式识别问题都是用在视频监控领域，摄像机的角度跟高度都相对固定。如果你知道你的项目中摄像机一般都在拍什么，那负样本可以非常有针对性地选取，而且可以事半功倍。举个例子，你现在想做火车站广场的异常行为检测，在这个课题中行人检测是必须要做的。而视频帧的背景基本都是广场的地板、建筑物等。那你可以在人空旷的时候选择取一张图，不同光照不同时段下各取一张图，然后在这些图上随机取图像块，每个块20x20，每个块就是一个负样本。这几张图就能缠上数以千计数以万计的负样本！而且针对性强。因为海洋、大山等东西对你的识别一点帮助也没有，还会增加训练的时间，吃力不讨好的事还是少做为好。我写了一段小程序，功能是根据背景图片自动随机生成指定数量指定尺寸的负样本：

訓練正樣本

opencv\_createsamples -info info/info.lst -num 1950 -w 20 -h 20 -vec positives.vec

opencv\_traincascade -data data -vec positives.vec -bg bg.txt -numPos 1800 -numNeg 900 -numStages 10 -w 20 -h 20

列出目錄下清單轉成文字檔

<http://honoyang.pixnet.net/blog/post/25702657-opencv-haartraining-%E7%9A%84xml%E6%AA%94%E8%A3%BD%E4%BD%9C>

ls > positive.txt

dir /b /s> postivite.txt