**工程數學專題書面報告**

**人臉偵測(Scale and orientation of faces)**

組別：A4   
指導老師：楊士萱 教授

組員： 102590013 楊雅婷、102590031 簡屏軒

口頭報告時間：2014/11/03

1. 摘要：

1. 研究目的：

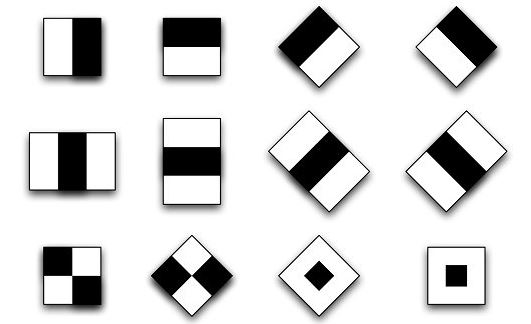
希望藉由實作與工程數學相關的應用，而精進學科的能力、增進學習興趣，並練習使用程式達到想完成的功能，也藉由報告訓練口頭表達能力。

1. 專題內容：
2. 演算法介紹：

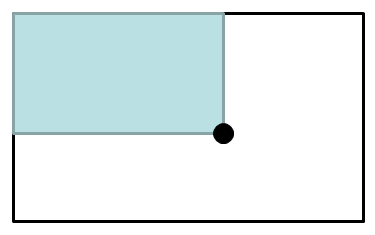
１. Integral Image積分影像

人臉偵測是使用Haar features辨別框架中的影像是否為人臉，但是用暴力方法來偵測人臉會導致速度太慢，因為要掃描整張影像，而影響的因素有位置、大小、方向等要考量，所以我們使用積分影像來快速偵測人臉特徵。

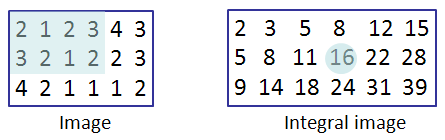
下圖為位置大小型態不同之示意圖：



(1)計算影像之前我們先將影像灰階化，也就是利用算式將RGB值改為灰階值，我們只計算框架區域的灰階值。



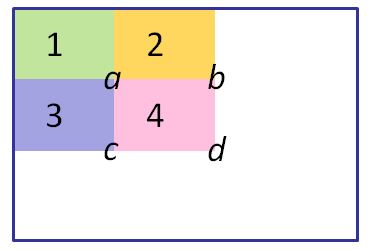
(2)從左上點往右下點拉出一個長方形藍色面積，我們把藍色面積裡的灰階值總和紀錄在右下點。



(3)只要在積分影像上對應的積分點進行幾次簡單的加減法運算，便可以求出**特徵差異值。**

如下圖：

a點的值是1區塊內所有灰階值的總和，b點的值是1和2區塊內所有灰階值的總和，c點的值是1、3區塊所有灰階值的總和，d區塊是1、2、3、4區塊所有灰階值的總和。



(4)**特徵差異值**計算方式為深色區與淺色區的灰階差值，對於人臉上不同區塊的灰階差值會有不同程度的結果，例如一般而言眼球區域的灰階總和會大於眼皮的灰階總和。如下圖所示：



2. Ada Boost自適應增強(原稱Adaptive Boosting)

是一種[機器學習](http://zh.wikipedia.org/wiki/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0" \o "機器學習)方法，用在分類上，由Yoav Freund和Robert Schapire提出。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。

算法如下：(參照維基百科的Adaboost解說)

先定義樣本的表示方式為 93592bfb9381cd7d9ff564bbc5010620.jpg

全部不論是正樣本還是負樣本一共有m個

xi屬於**X**

yi是已知的分類結果 [cab1576b5fe4064b5a797f918146da67.jpg](http://lsyueh.pixnet.net/album/photo/151034090)

樣本設定好後,接下來就是要開始boosting,首先要初始每個樣本的**權重**D

[0db56cfd1b2b53f805d3614c1a6dc515.jpg](http://lsyueh.pixnet.net/album/photo/151034077)

接下來就是開始重複boosting

For  [b7e19e8872b77eec52558e059f2ed8eb.jpg](http://lsyueh.pixnet.net/album/photo/151034089)

(1) 先計算每(j)個弱分類器h的分類後的權重合ε(或是叫分類誤差或是分類錯誤和比較貼切),計算的方式要遵守下面

[44d8d3c98bbcb6f1c0d015334906fb77.jpg](http://lsyueh.pixnet.net/album/photo/151034081)

上面的公式就是在第j個弱分類器下,把分錯的樣本的權重D累加起來就是εj的數值

然後在j個弱分類器下得到的權重合εj中找出最小的值

就是這次(**第t次boosting後**)的最佳分類器ht了(就是下面一團寫的)

[758198472dcc37370f45923aec863e7e.jpg](http://lsyueh.pixnet.net/album/photo/151034087)

(2) if εt > 0.5 then stop.

這是wiki上設定的停止條件,這邊可以依照自己分類需求去改

當εt (**第**t**次**boosting**後求得的分類錯誤和**)大於0.5時表示樣本已經很難在被分下去了

(3) 接者是計算這個弱分類器在未來強分類器內的權重"Alpha"(P.S:與樣本權重是**不同**的)

[f60f6fc4a062d2701457183963421d9b.jpg](http://lsyueh.pixnet.net/album/photo/151034091)

[a4ba1877721cf4b501e24b13b935e51a.jpg](http://lsyueh.pixnet.net/album/photo/151034088)

這樣的運算特性可以使數值越小的εt得到越大的值。

(4) 接下來就是要更新樣本權重D了,如下

[12fa4ca5f0e6da6c8445954b877d2f3b.jpg](http://lsyueh.pixnet.net/album/photo/151034080)

下一輪要用的權重Dt+1會因為exp內的關係而改變

[98a7fe654acb294d4f3551d1de4ca87d.jpg](http://lsyueh.pixnet.net/album/photo/151034085)

這次被ht分對的樣本權重會降低,而被判錯的樣本權重會變大

因為我們要找的是最小的樣本權重合,所以這樣的跟新方式一定可以確保找到的都是最佳弱分類器

(所以當連最小的εt超過0.5,就很明顯可以察覺出樣本都分錯了)

而Zt是將新的樣本權重,重新正規化到0~1之間

接著再回到第一步開始一輪新的boosting

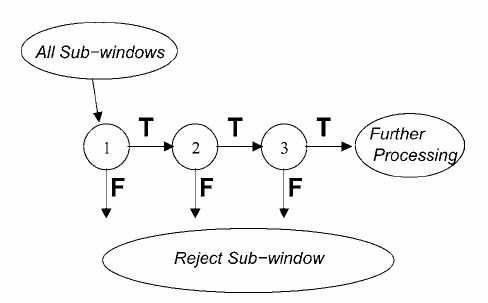
當整個循環達到自己設定的停止條件後,強分類器就是下面這樣

[53fcaac2df8212761c1c60b6aa968653.jpg](http://lsyueh.pixnet.net/album/photo/151034083)

３. Cascade Classifier瀑布分類

一開始將feature分成好幾個classifier。最前面的classier辨識率最低，但是可以先篩選掉很大一部份不是人臉的圖片；接下來的Classier處理比較難處理一點的case篩選掉的圖片也不如第一個classier多了；依此下去，直到最後一個classier為止。最後留下來的就會是我們想要的人臉的照片。

如下圖：



1. 程式整體與解析：

我們用OpenCV所使用的方法，是由「Viola & Jones」所發表的AdaBoost Learning with Haar-Like Features來實現人臉偵測。步驟如下：

1.首先定義一些Haar-Like Features。

2.再來，我們必須給定一些sample，例如:假如我們要偵測的是人臉，那麼 我們就要輸入一些人臉的sample。有了這些sample，我們將會利用AdaBoost learning algorithm來挑出某幾個代表性的Haar-Like Features，以這個例子來講，我們可以想成這些被挑選出來的feature就是代表人臉的feature。

3.每個被挑選出來的feature都代表一種classifier，許多種被挑選出來的feature因此構成了一連串的classifier，我們稱為strong classifier。而每個classifier皆用來判斷所輸入的圖片是否為人臉，並且回傳「是」或「否」，最後，通過所有classifier的圖片將被判定是一張人臉。

4.程式碼:

#include <cv.h>

#include <cxcore.h>

#include <highgui.h>

#include <iostream>

using namespace std;

// the minimum object size

int min\_face\_height = 50;

int min\_face\_width = 50;

int main( int argc , char \*\* argv ){

string image\_name="1.jpg";

// Load image

IplImage\* image\_detect=cvLoadImage(image\_name.c\_str(), 1);

string cascade\_name="C:/OpenCV2.0/data/haarcascades/haarcascade\_frontalface\_alt.xml";

// Load cascade

CvHaarClassifierCascade\* classifier=(CvHaarClassifierCascade\*)cvLoad(cascade\_name.c\_str(), 0, 0, 0);

if(!classifier){

cerr<<"ERROR: Could not load classifier cascade."<<endl;

system("pause");

return -1;

}

CvMemStorage\* facesMemStorage=cvCreateMemStorage(0);

IplImage\* tempFrame=cvCreateImage(cvSize(image\_detect->width, image\_detect->height), IPL\_DEPTH\_8U, image\_detect->nChannels);

if(image\_detect->origin==IPL\_ORIGIN\_TL){

cvCopy(image\_detect, tempFrame, 0); }

else{

cvFlip(image\_detect, tempFrame, 0); }

cvClearMemStorage(facesMemStorage);

CvSeq\* faces=cvHaarDetectObjects(tempFrame, classifier, facesMemStorage, 1.1, 3

, CV\_HAAR\_DO\_CANNY\_PRUNING, cvSize(min\_face\_width, min\_face\_height));

if(faces){

for(int i=0; i<faces->total; ++i){

// Setup two points that define the extremes of the rectangle,

// then draw it to the image

CvPoint point1, point2;

CvRect\* rectangle = (CvRect\*)cvGetSeqElem(faces, i);

point1.x = rectangle->x;

point2.x = rectangle->x + rectangle->width;

point1.y = rectangle->y;

point2.y = rectangle->y + rectangle->height;

cvRectangle(tempFrame, point1, point2, CV\_RGB(255,0,0), 3, 8, 0);

}

}

// Save the image to a file

cvSaveImage("finish.jpg", tempFrame);

// Show the result in the window

cvNamedWindow("Face Detection Result", 1);

cvShowImage("Face Detection Result", tempFrame);

cvWaitKey(0);

cvDestroyWindow("Face Detection Result");

// Clean up allocated OpenCV objects

cvReleaseMemStorage(&facesMemStorage);

cvReleaseImage(&tempFrame);

cvReleaseHaarClassifierCascade(&classifier);

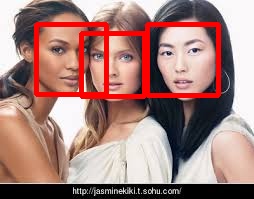
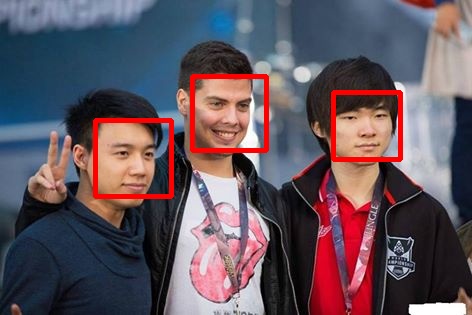
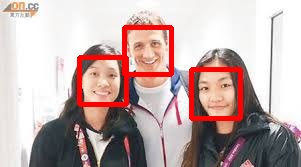
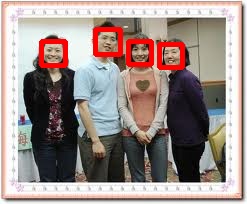
cvReleaseImage(&image\_detect);

system("pause");

return EXIT\_SUCCESS;

}

5.測試完成圖:



1. 參考資料：

[1] **Trek** [[OpenCV] Face Detection (人臉偵測)](http://seeyababy.blogspot.tw/2010/04/opencv-face-detection.html):

<http://seeyababy.blogspot.tw/2010/04/opencv-face-detection.html>

### [2] [finalevil blog](http://blog.finalevil.com/) [[程式]OpenCV學習筆記心得04：簡單的人臉偵測(face detection)，使用HaarDetectObjects](http://blog.finalevil.com/2008/03/opencv04face-detectionhaardetectobjects.html)

<http://blog.finalevil.com/2008/03/opencv04face-detectionhaardetectobjects.html>

### [3] [神來了](http://lsyueh.pixnet.net/blog) [Adaptive Boosting a.k.a. Adaboost](http://lsyueh.pixnet.net/blog/post/7525275)

<http://lsyueh.pixnet.net/blog/post/7525275-adaptive-boosting-a.k.a.-adaboost>

[4] **維基百科** AdaBoost

<http://zh.wikipedia.org/wiki/AdaBoost>

[5] [**逍遙文工作室**](http://cg2010studio.wordpress.com/)[[OpenCV] Dev-C++ 4.9.9.2 安裝 OpenCV 2.0](http://cg2010studio.wordpress.com/2011/03/31/opencv-dev-c-4-9-9-2-%e5%ae%89%e8%a3%9d-opencv-2-0/)

<http://cg2010studio.wordpress.com/2011/03/31/opencv-dev-c-4-9-9-2-%E5%AE%89%E8%A3%9D-opencv-2-0/>