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

一、摘要：

此次工程數學的報告我們是負責**人臉辨識(Scale and orientation of faces)**的部分，其中會介紹人臉辨識所需要的演算法(Integral Image、Ada Boost 及 Cascade Classifier)，透過演算法我們可以瞭解程式碼與如何分析人臉的位置。

二、研究目的：

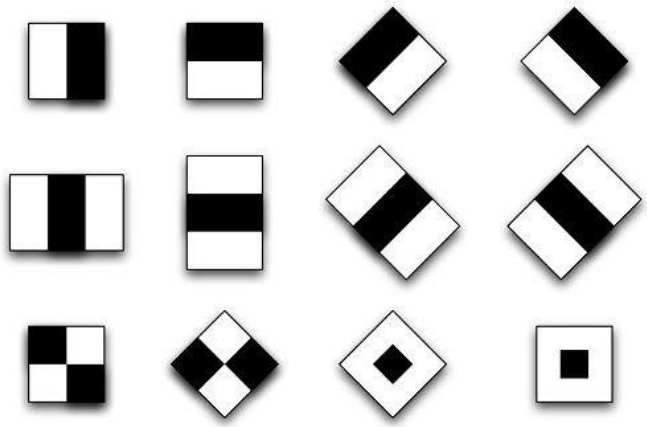
希望藉由實作與工程數學相關的應用，而精進學科的能力、增進學習興趣，並練習使用程式達到想完成的功能，也藉由報告訓練口頭表達能力。

三、專題內容：

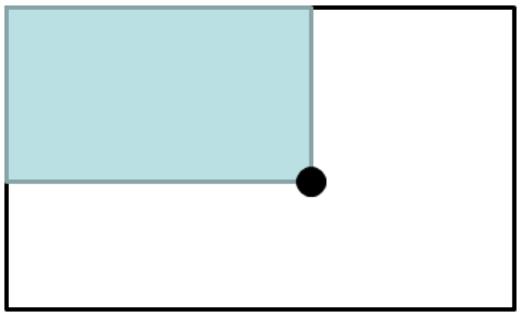
（一）演算法介紹：

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。

2	1	2	3	4	3
3	2	1	2	2	3
4	2	1	1	1	2

Image

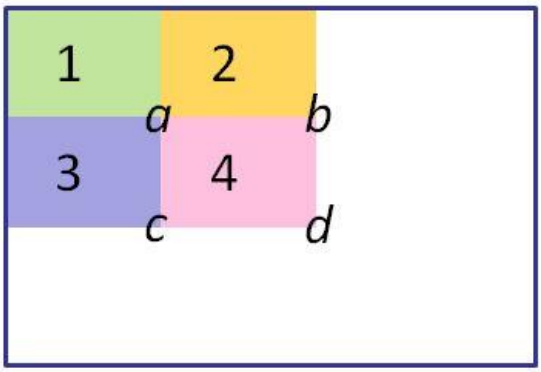
2	3	5	8	12	15
5	8	11	16	22	28
9	14	18	24	31	39

Integral image

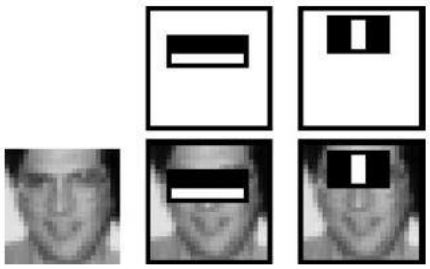
(3)只要在積分影像上對應的積分點進行幾次簡單的加減法運算，便可以求出**特徵差異值。**

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 $(x_1, y_1), \ldots, (x_m, y_m)$

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 $x_i \in X, y_i \in Y = \{-1, +1\}$

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

$$D_1(i) = \frac{1}{m}, i = 1, \ldots, m.$$

接下來就是開始重複 boosting

For $t = 1, \ldots, T$

(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

$$\epsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$$

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)

$$h_t = \underset{h_j \in \mathcal{H}}{\operatorname{argmin}} \epsilon_j$$

(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)

$$\alpha_t \in \mathbf{R}$$

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$$

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

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

下一輪要用的權重 Dt+1 會因為 exp 內的關係而改變

$$-\alpha_t y_i h_t(x_i) \begin{cases} < 0, & y(i) = h_t(x_i) \\ > 0, & y(i) \neq h_t(x_i) \end{cases}$$

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

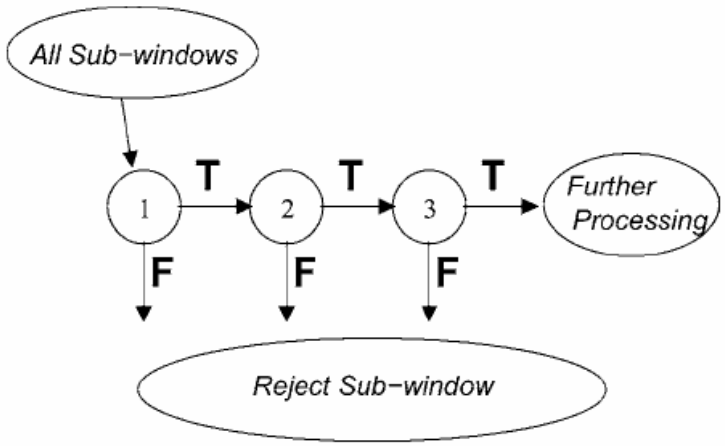
接著再回到第一步開始一輪新的 boosting

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

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

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：



（二）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

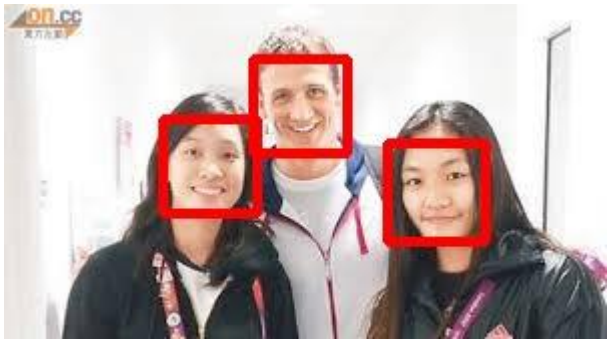
classifier 皆用來判斷所輸入的圖片是否為人臉，並且回傳「是」或「否」，最後，通過所有 classifier 的圖片將被判定是一張人臉。

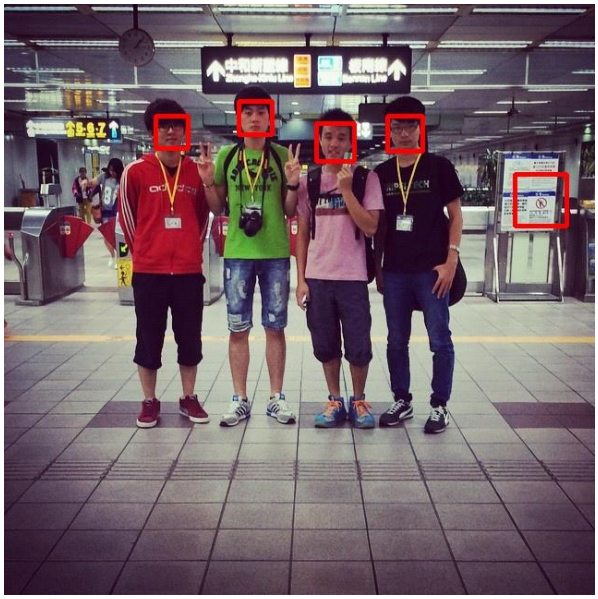
4. 程式碼:

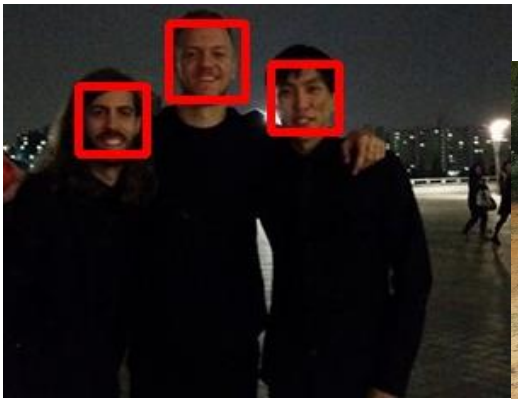
```
#include <cv.h>
#include <cxcore.h>
#include <highgui.h>
#include <iostream>
using namespace std;
// the minimum object size
int min_face_height = 50;
int min_face_width = 50;
int main( int argc , char ** argv ){
	string image_name="1.jpg";
	// Load image
	IplImage* image_detect=cvLoadImage(image_name.c_str(), 1);
	string
cascade_name="C:/OpenCV2.0/data/haarcascades/haarcascade_frontalface_alt.x
ml";

	// 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);
```

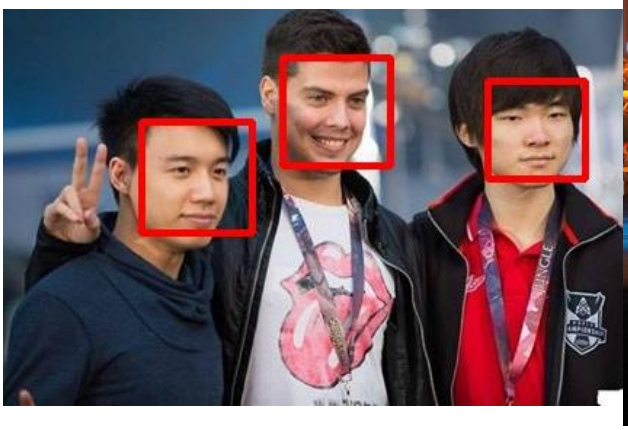
```
CvSeq* faces=cvHaarDetectObjects(tempFrame, classifier, facesMemStorage,
1.1, 3
, CV_HAAR_DO_CANNY_PRUNING, cvSize(min_face_width,
min_face_height));
        if(faces){
            for(int i=0; i<faces->total; ++i){
                // Setup two points that define the extremes of the rectangle,
                // 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;
}
```

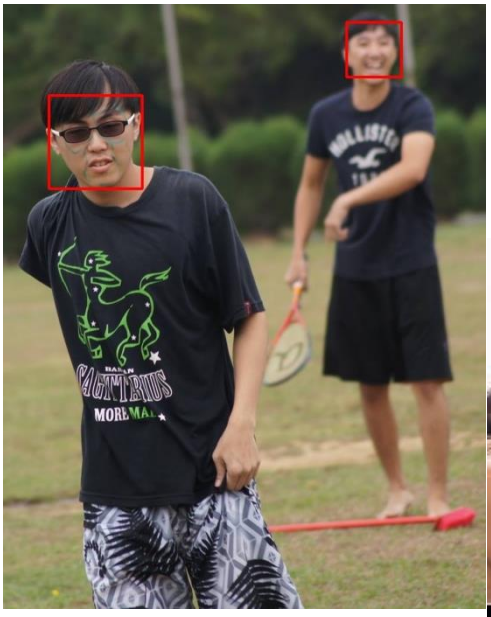
5. 測試完成圖:

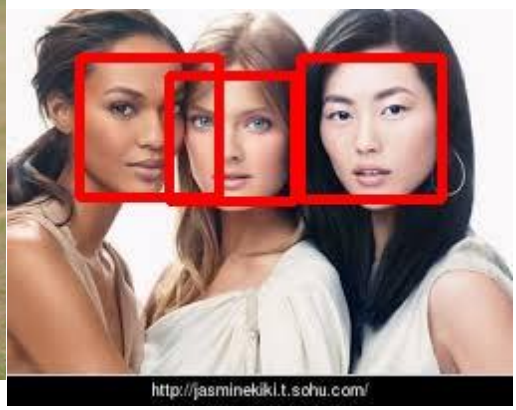











http://jasmineliki.t.sohu.com/

（三）參考資料：

[1] **Trek** [OpenCV] Face Detection (人臉偵測):
http://seeyababy.blogspot.tw/2010/04/opencv-face-detection.html
[2] **finalevil blog** [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
[3] **神來了** Adaptive Boosting a.k.a. Adaboost
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**[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/