

國立屏東大學

資訊科學系

碩士論文

基於 LBP 特徵與 AdaBoost 的行人追蹤
AdaBoost-Based Pedestrian Tracking with
LBP Features



研究生：潘俊龍
指導教授：林義凱

中華民國一百零七年七月

國立屏東大學資訊科學系

碩 士 論 文

基於 LBP 特徵與 AdaBoost 的行人追蹤

論文口試委員會主席

陳建良

委員

楊政興

委員

指導教授

林義凱

系主任

楊政興

研究生

潘俊龍 撰

中 華 民 國 107 年 7 月 4 日

誌 謝

在進入研究所後，有著短暫的迷茫，不知道未來該朝著什麼領域研究比較好，有幸找到林義凱老師作為我的指導教授，為我分析和指點各種不同的方向。在就學的期間，老師一直不斷關心我的學習狀況，也給予我很多協助，不論是論文的方向還是遇上困難時的協助，至今仍銘感五內。

也謝謝我的同學李柏融，協助我口試相關的準備和論文的要點，並且在我最後的準備期間，鼓勵我前進。

也謝謝其他同學的關心，雖然每個人都有不同的題目，但是大家一起努力地共同的朝畢業所邁進，互相提攜、一起奮鬥的感覺令人深刻。

謹誌

中華民國一零七年七月

摘 要

近年來，低成本嵌入式系統的計算能力大幅增加，但仍需要使用低計算量的演算法才能達到即時追蹤行人的工作。

我們使用 AdaBoost 搭配 LBP 特徵來偵測物體並使用匹配鄰近偵測物體來追蹤物體。實驗結果顯示在即時的應用中我們的方法 recall 約為 54%~90%。

關鍵詞：LBP 特徵、行人追蹤、AdaBoost

Abstract

In recent years, the computing power of low-cost embedded systems has greatly increased, but it still requires the use of efficient algorithms to achieve real-time tracking of pedestrians.

We use AdaBoost with LBP features to detect objects and use matching algorithm to track objects. The experimental results show that in our method, recall is about 54% to 90% in real-time applications.

keywords : LBP features 、 Pedestrian Tracking 、 AdaBoost

目 錄

誌 謝	i
摘 要	ii
Abstract	iii
第 1 章 緒論	1
1.1 研究背景與動機	1
1.2 相關研究	2
1.2.1 Local Binary Pattern (LBP) 方法介紹	2
1.2.2 AdaBoost 方法介紹	3
第 2 章 研究方法	5
2.1 正樣本去背	6
2.2 追蹤行人	6
第 3 章 實驗	11
第 4 章 結論	25
參考文獻	26

圖 錄

圖 1，LBP 遮罩	2
圖 2，AdaBoost cascade	3
圖 3，計算偵測到的物體與追蹤中物體的距離	8
圖 4，從影格中切出來的行人影像作為訓練用正樣本	11
圖 5，去除背景之後的正樣本圖	12
圖 6，連續(a)~(d)影格顯示 ground truth 追蹤標籤。紅色為追蹤 id.....	12
圖 7，(a)~(f) 採用 5000 張正樣本，5000 張負樣本的訓練參數追蹤物體的結果.....	13
圖 8，(a)~(f) 採用 9000 張正樣本，9000 張負樣本的訓練參數追蹤物體的結果.....	14
圖 9，9000 樣本實驗組結果趨勢.....	22
圖 10，5000 樣本實驗組結果趨勢.....	22
圖 11，兩組實驗結果偵測率趨勢比較	23
圖 12，兩組實驗結果比較	24

表 錄

表 1，以 9000 張正樣本與 9000 負樣本訓練參數偵測 vidf 影片前 100 影格的結果 ...	16
表 2，以 5000 張正樣本與 5000 負樣本訓練參數偵測 vidf 影片前 100 影格的結果 ...	19
表 3，兩組實驗結果比較	24

第1章 緒論

1.1 研究背景與動機

影像資訊在人類的感知扮演很重要的角色，例如利用監視器監控車流或人流。目前使用卷積神經網路 Convolutional Neural Network [1]（簡稱 CNN）來辨識物體已經有非常好的成果。但是其所需的計算資源比較高。

在沒有使用 GPU 來輔助運算之下，intel i7 CPU 機器上使用 CNN 來辨識一張大小為 800x600 的圖就需要 2~3 秒的時間。這樣就無法達成即時（real-time）應用。因此傳統的手刻（hand-crafted）的方法在計算能力不高的設備上仍有存在的價值。

在近年來小型的嵌入式系統擁有體積小、低耗電的特點。這類的嵌入式系統被使用於自動控制與影像處理上。本研究即是要探討在這樣的計算能力限制下如何達到行人的偵測。

1.2 相關研究

1.2.1 Local Binary Pattern (LBP) 方法介紹

LBP (Local Binary Pattern, 局部二值模式) 是一種用來描述圖像局部紋理特徵的方法，它在2002年由T.Ojala [2] 首先提出，具有灰度不變性等顯著優點。用於紋理特徵提取。

其中 I_d 為中心灰度值、 I_n 為鄰近灰度值，LBP特徵bn由下式來定義

$$b_n = \begin{cases} 1 & \text{if } I_n \geq I_d \\ 0 & \text{if } I_n < I_d \end{cases} \quad (1)$$

如圖 1，首先定義像素的 8 鄰域中，以中心像素的灰度值為門檻值，將周圍 8 個像素的值與其比較，如果周圍的像素值值小於中心像素的灰度值，該像素位置就被標記為 0，否則為 1。每個像素得到一個二進制組合，就像 11110001。每個像素有 8 個相鄰的像素點，即有 2^8 種可能性組合，這些組合就被叫做局部二值模式或 LBP 碼。

6	5	2
7	6	1
9	8	7

1	0	0
1		0
1	1	1

1	2	4
128		8
64	32	16

Pattern = 11110001

LBP = 1+16+32+64+128=241

圖 1，LBP 遮罩

1.2.2 AdaBoost 方法介紹

搜尋圖片上的 LBP 特徵可以辨識圖片上特定的物體，但利用單一特徵所能辨識的物體無法太複雜，因此 Viola 和 Jones [3] 利用數個簡單的 Haar-like 特徵來共同辨識物體，然後透過 AdaBoost 的方式聯合數個簡單的特徵來辨識人臉。

我們也採用類似的方法，使用數個 LBP 特徵以 AdaBoost 的方法聯合來辨識物體。

AdaBoost 的方法簡述如下。輸入的圖片被分成 n 個子圖 (sub-image)，每個子圖分別計算特徵值，以是否存在特定特徵值作為一個 classifier。在監督式學習的過程中會找出 positive 結果的 classifier。所有影像上的子圖 (sub-image) 都丟入第一個 stage 處理。請看圖 2。

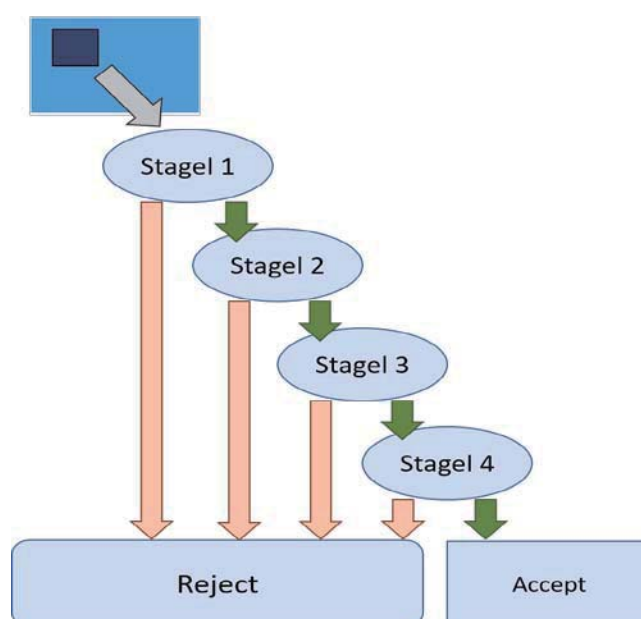


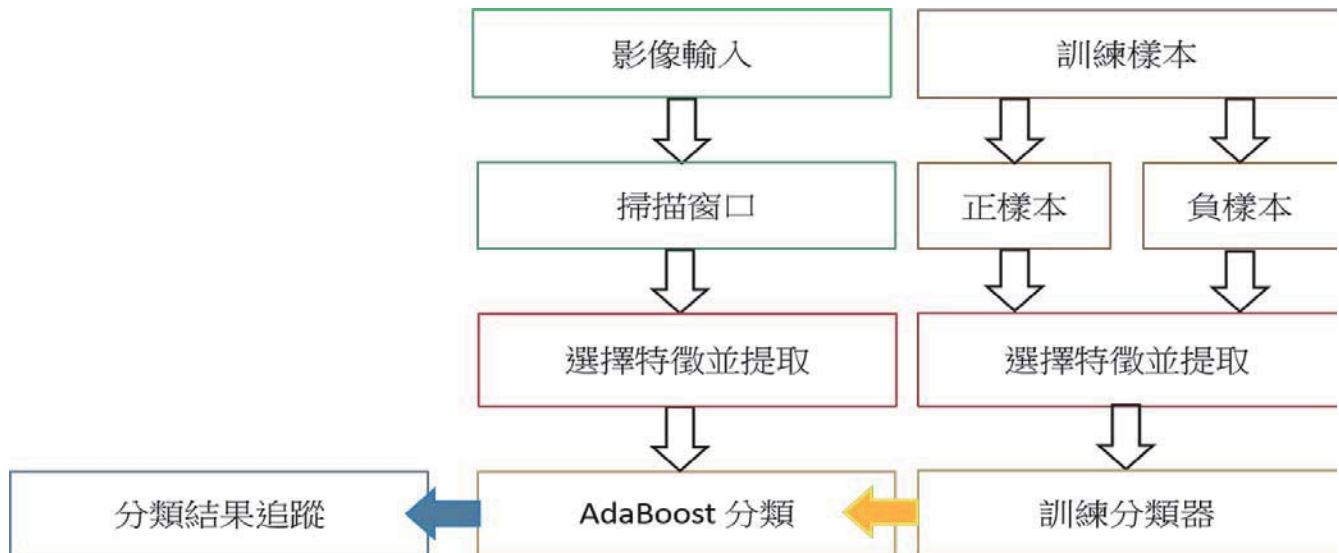
圖 2，AdaBoost cascade

每一個 stage 會使用數個 features（特徵）以不同的權重聯合起來形成一個強分類器（strong classifier）。在 stage 1 被判定為 negative 的子圖就直接判定為不包含所要找的物件不再進一步處理。

而在 stage 1 被判定為 positive 的子圖就進入 stage 2 再進一步由另一 strong classifier 來判定是否含有我們要找的特徵。

每一個 stage 所做的工作都是類似的，不同的是所用的 classifiers 不同。假設有人臉的 subimages 在 stage 1 會有 99% 的機會被判定為有人臉，因此會被送到 stage 2（有 1% 的機會把這些有人臉的 subimages 判別為沒有人臉而直接 reject）。這些通過 stage 1 有人臉的 subimages 在 stage 2 也會有 99% 的機會被判定為有人臉，也才會進入 stage 3，因此有人臉的 subimages 會有 0.99×0.99 的機會通過 stage 2。依此類推，有人臉的 subimages 通過 20 個 stages 的機率就是 $0.99^{20} = 81.7\%$ 。

第2章 研究方法



方法流程

如上流程，本論文會先從訓練樣本中找出正與負的樣本，然後提取特徵後，再交由 AdaBoost 分類器做為訓練用。

然後在實際應用時會將輸入影像的每個影格做小圖掃描，最後提取特徵後，交由訓練好的 AdaBoost 分類器做行人的分類辨識，確定為行人後輸出其行人的窗口座標位置。

最後將其輸出結果的座標資訊轉由行人追蹤系統中進行追蹤，本方法會有多個陣列儲存行人座標資訊，有追蹤中行人陣列、已偵測行人陣列，程式會先將每個行人標示編號 ID，然後在之後的每個影格做追蹤中行人和已偵測行人的距離分析，如果兩者有最短的，那麼表示為同一個行人，程序就會匹配，並在進行下一輪的追蹤分析。

2.1 正樣本去背

在準備正樣本影像時需注意影像如果含有複雜背景，容易混淆了人形的學習。因此行人正樣本的背景最好能去除。去背（keyer effects）的工作如果是施做在少量的影像上可以用人工操作影像處理軟體的方式完成。在 AdaBoost cascade training 的辨識器學習階段需要大約 5000 張的正樣本，以人工來處理是一個相當浩大的工程。

我們將收集來，有人形在內的影像經找出人形的位置座標。再加以截圖取出一張張有人形的影像。

2.2 追蹤行人

經由 AdaBoost 偵測 LBP 特徵找出每一張影格中的行人位置後，將行人位置資訊交由追蹤程式考慮前後影格的關連性以追蹤行人的移動。

追蹤程式的輸出是每一時刻 t 輸出 tuple $\langle id, x, y \rangle$ $id \in \{1, n\}$ 序列。 id 即為物體的編號， (x,y) 為物體的位置。

例如 t 時刻的輸出為

$\langle 1, 50, 20 \rangle, \langle 2, 30, 101 \rangle, \langle 3, 75, 99 \rangle$

$t+1$ 時刻的輸出為

$\langle 1, 52, 24 \rangle, \langle 2, 37, 100 \rangle, \langle 3, 70, 97 \rangle, \langle 4, 10, 132 \rangle$

表示 id 為 1, 2, 3 的物體持續追蹤中，且在 $t+1$ 時刻出現了一個 id 為 4 的新物體。

追蹤程式每一時刻 t 的輸入為 n 個偵測到的行人位置資訊 $\langle x_1, y_1, w_1, h_1 \rangle, \langle x_2, y_2, w_2, h_2 \rangle, \dots \langle x_n, y_n, w_n, h_n \rangle$ 。此處 (x_i, y_i) 為物體左上角所在位置座標， w_i, h_i 分別為物體的寬與高。

我們的追蹤演算法將追蹤中的物體資訊儲存於雙向鏈結串列 `tracked_objs` 中。每一個追蹤中物體的資訊以 tuple $\langle id, x, y, w, h, c \rangle$ 記錄於 `tracked_objs` 的節點 (node) 中， x, y, w, h, c 分別為追蹤中物體的 x 座標、 y 座標、寬、高、信心值 `confidence`。信心值用來控制將失去追蹤的物體從追蹤中串列移除的時機。

追蹤物體的流程如下。首先將輸入的 n 個行人位置資訊 $\langle x_1, y_1, w_1, h_1 \rangle, \langle x_2, y_2, w_2, h_2 \rangle, \dots, \langle x_n, y_n, w_n, h_n \rangle$ 存入雙向鏈結串列 `detected_objs` 中。接著對 `detected_objs` 的每一 node d 記載的偵測到的物體與 `tracked_objs` 內的追蹤中的物體計算歐基理德距離。將計算的結果記錄於 node d 上的陣列 `proposal` 中。請看圖 3。

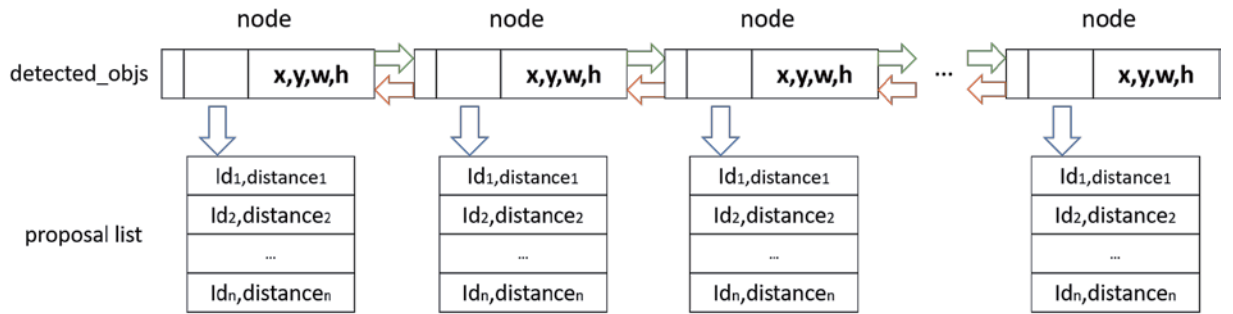


圖 3，計算偵測到的物體與追蹤中物體的距離

`detected-objs` 上的每一個 node d 所記錄的陣列 `proposal` 中記錄著 $\langle id_1, distance_1 \rangle, \dots, \langle id_m, distance_m \rangle$ 其中 $id_i, distance_i$ 為追蹤中物體 id_i 與 node d 所記錄的偵測到的物體的距離 $distance_i$ 計算的演算法片段為

```

for each node d in detected-objs
    i ← 1
    For each node e in tracked-objs
        distance ← Compute-dis(d,e)
        d.proposal[i] ← (e.id,distance)
    i ← i + 1

```


計算完所有 t 時刻偵測到的行人與追蹤中物體兩兩之間的距離之後，接著找出所有兩兩距離中短的距離。

以 `detected_objs` 中的 node d 為例，找出其所附的 proposal 陣列中 distance 最小的一個。接著再找 `detected_objs` 中的下一個點中 proposal 陣列內有最小 distance 的 id 。如此掃過 `detected_objs` 中所有的 node 找出 distance 最小的 $\text{tuple} \langle id_{\min}, distance_{\min} \rangle$ 。

例如最小 $distance=37$ 發生在 node d 所附的 proposal 陣列中，且 tuple 是 $\langle 5, 37 \rangle$ ，則表示追蹤中串列 `tracked_objs` 有一行人 $id=5$ 且與 node d 所記錄的偵測到的行人距離為 37。因此認定 $id=5$ 這個行人在 frame t 中即為 node d 所代表的行人，這樣完成了一個配對(matching)。

參與此配對的偵測到行人(以 node d 紀錄與追蹤中行人以 $d_{\min}, distance_{\min}$ 紀錄)不能再參與其他的配對因此要從 `detected_bojs` 中刪除，因此將 node d 刪除且將 `detected_objs` 中每一個 node 所附的 proposal 陣列中的有 id_{\min} 的紀錄也都刪除。

而對 id_{\min} 所代表的追蹤中行人應該利用所配對的偵測到行人位置資訊來更新，並將其信心值 confidence 更新為 1，其他未能配對成功的追蹤中行人其信心值會乘以 0.9，經過幾 frame 之後一直未成功配對的追蹤中物體其信心直就會大幅降低。

例如經過 6 個 frame 都未能配對到任何偵測物體的追蹤中物體其信心值就降為 $0.96^6=0.5314$ 。低於 0.6 信心值的追蹤中行人就從 tracked_objs 中刪除。

經過上述更新 detected_objs 步驟之後已配對過的行人就從 detected_objs 中刪除，接著再重複配對的工作，將此時 detected_objs 中擁有最小 distance 紀錄的 id 和 node 找出來。形成新的配對，再重複更新 detected_objs 和 tracked_objs。

這樣的步驟一直進行到無法產生新的配對為止。無法產生配對時，detected_objs 有可能不是空的，這時代表這些無法找到追蹤中行人可以配對的偵測物件是新出現在場景中或是重新被偵測到的行人，這時就將這些未能匹配的偵測物體加入追蹤中行人，開始追蹤。新加入的追蹤中行人，會被設定一個新的 id。

到此完成一個影格(Frame)的處理，接著將整個 detected_objs 刪除，再讀入新 frame，找出所有行人的位置，建立新的 detected_objs，重複前述的配對，追蹤的程序直到所有的 frame 都處理完畢。

第3章 實驗

我們使用行人影片資料集 UCSD [4]來測試所提方法的效能。此資料集有兩組不同攝影機視角所拍攝的影片。此兩組影片集分別命名為 vidf 和 vidd。並各自包含數個子影片。

我們先將所有影片的影格 (frame) 取出，分成訓練集 training set 和測試集 testing set。我們使用的 vidf 資料集共有 32597 張圖，我們將 2/3 的圖來做為訓練資料，並取剩下的圖中的 100 張當作測試集。

首先，我們把訓練集影格中的行人位置標示出來並儲存成為文字檔清單，並依據此清單將行人部分影像切出來，分別儲存成影像檔。請看圖 4 顯示從影格中切出來的行人影像。



圖 4，從影格中切出來的行人影像作為訓練用正樣本

可以從圖 4 中看出這些從影格中切出來的訓練正樣本，可能含有背景物件。這些背景物件可能會在訓練的過程中被誤判為前景的一部份。因此我們將測試正樣本圖的背景去除是否會影響訓練結果。請參考圖 5 去除背景之後的正樣本圖。



圖 5，去除背景之後的正樣本圖

為了評估各個系統參數對最後正確追蹤物體物體的影響，我們需要先標示出正確的追蹤結果（ground truth）。請看圖 6 正確追蹤的標記。正樣本與負樣本的數量

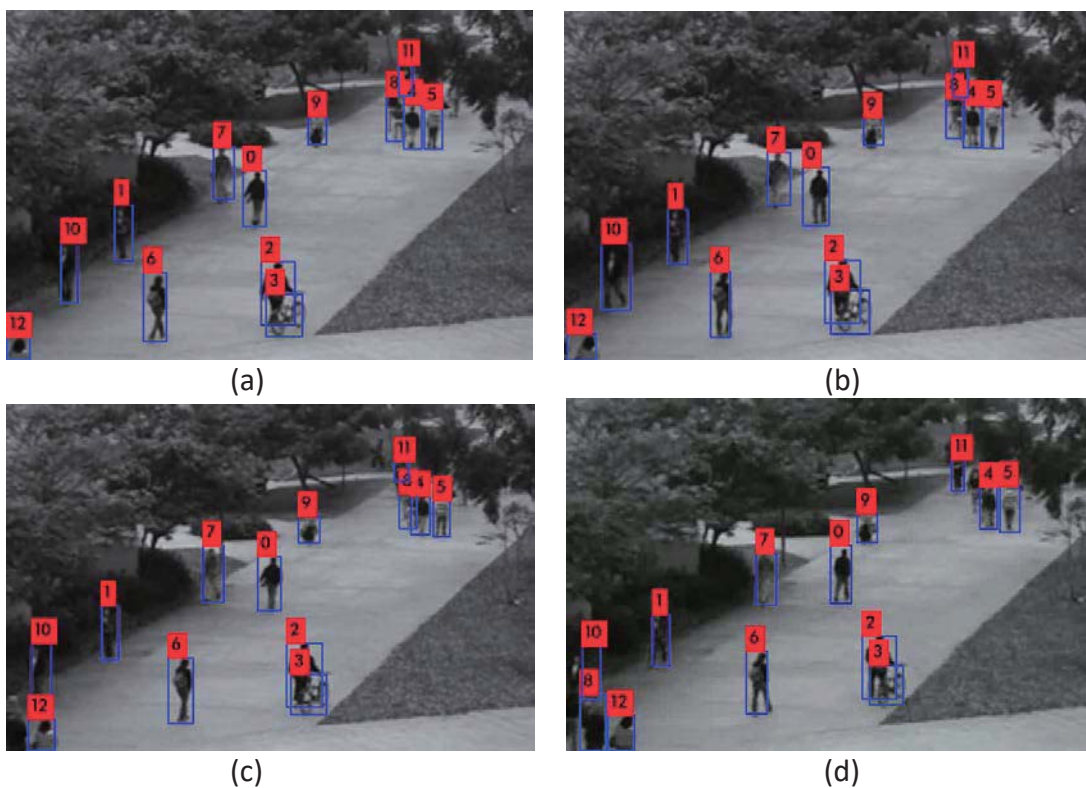


圖 6，連續(a)~(d)影格顯示 ground truth 追蹤標籤。紅色為追蹤 id

第一個測試的參數組合是使用 5000 張正樣本，5000 張負樣本的訓練資料。

參考圖 7 使用 5000 張正樣本，5000 張負樣本訓練參數的追蹤結果。

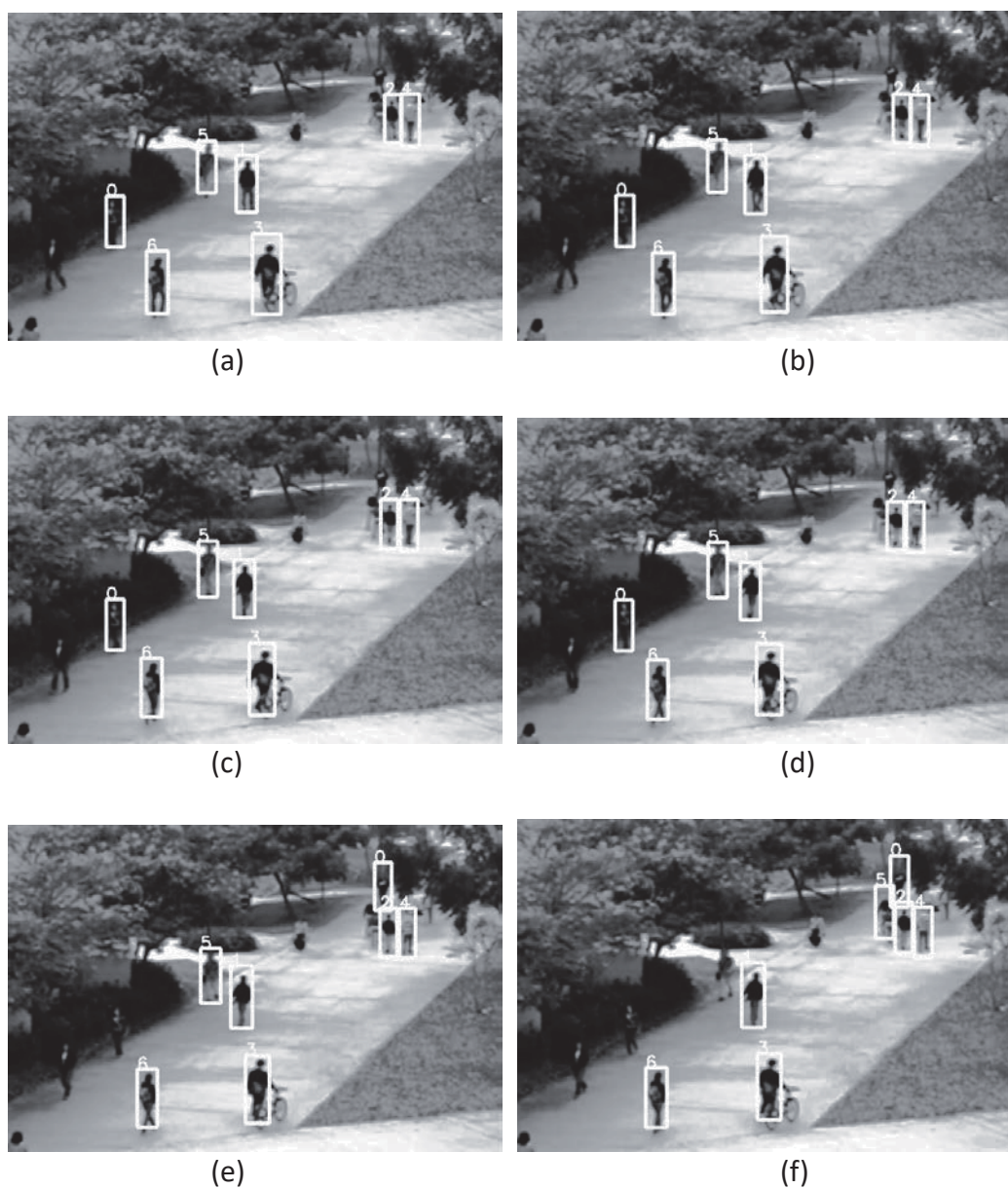


圖 7，(a)~(f) 採用 5000 張正樣本，5000 張負樣本的訓練參數追蹤物體的結果

第二個測試的參數組合是使用 9000 張正樣本，9000 張負樣本的訓練資料。

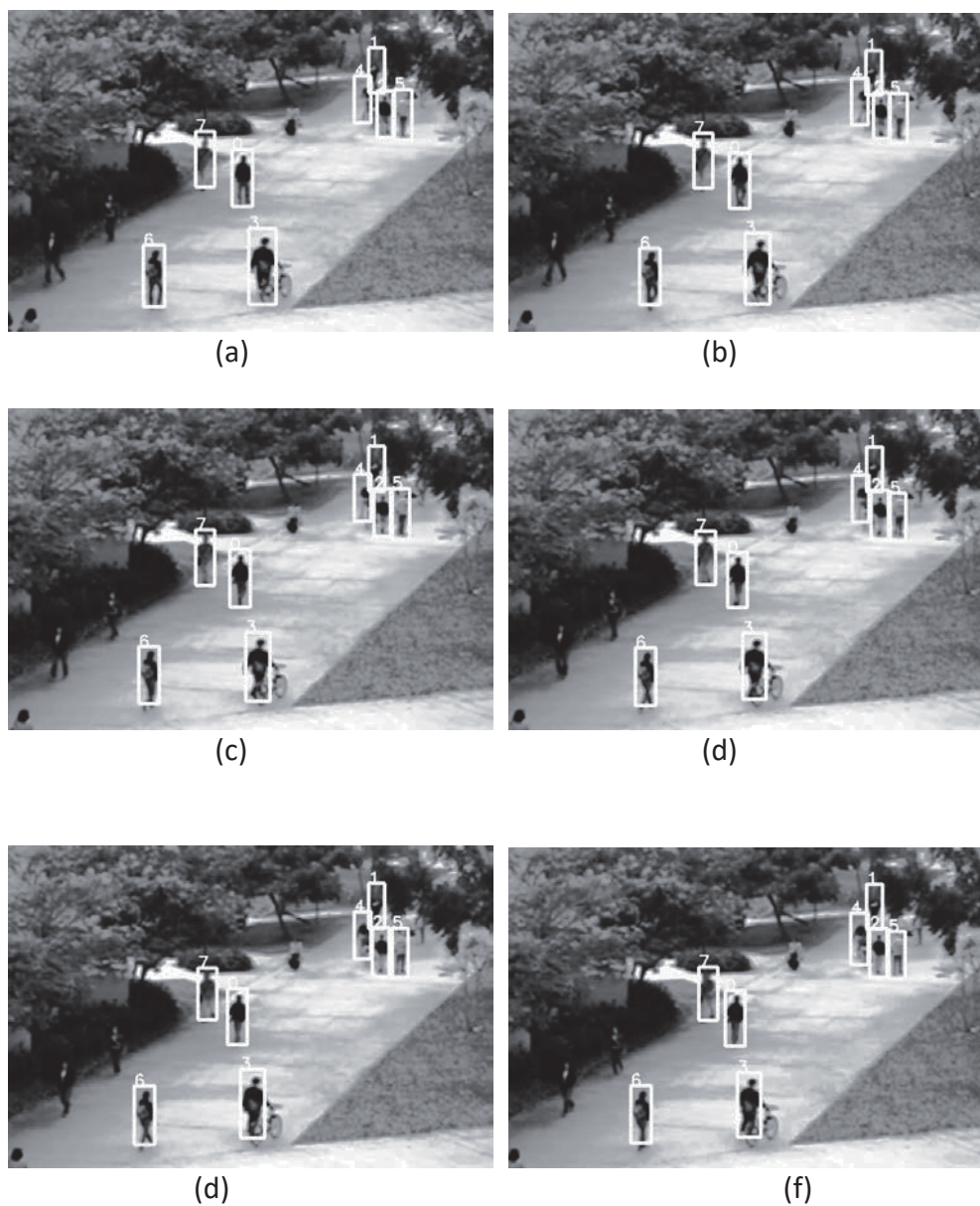


圖 8， (a)~(f) 採用 9000 張正樣本，9000 張負樣本的訓練參數追蹤物體的結果

我們計算我們追蹤成果的正确率 accuracy 。定義如下式

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

- 此處 TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

這個指標用來指出判讀為有行人和無行人的總和正确率。

另一個衡量效能的指標是

$$\text{Recall} = TP/(TP+FN)$$

此指標衡量所有應該被找出來的行人有多少比例被偵測出來。

另一個衡量效能的指標是精確度

$$\text{Precision} = TP/(TP+FP)$$

此指標衡量，被偵測為行人的位置有多少是正確的。

我們以前述 vidf 影片的前 100 個影格為例子，計算 precision, accuracy, recall 來評估我們演算法的效能。

表 1 是使用 9000 張正樣本與 9000 負樣本的參數，偵測與追蹤的結果。因為無法由影格中計算 TN，我們估計每張圖片中可能判讀為行人的位置數減掉判讀為 TP、FN 和 FP 的值作為 TN。此處每張影格的大小為 640x420，行人的大小為 20x84，因此 $640/20 = 32$ ， $420/84=5$ ，估計一張圖不考慮行人 overlap 情況下有 160 個可判讀為行人的位置。

表 1，以 9000 張正樣本與 9000 負樣本訓練參數偵測 vidf 影片前 100 影格的結果

	實際有人		實際無人		Precision	Accuracy	Recall
影格編號	判讀有人 TP	判讀無人 FN	判讀有人 FP	判讀無人 TN	TP/(TP+FP)	(TP+TN) / (TP+TN+FP+FN)	TP/(TP+FN)
1	8	3	0	149	1.000	0.981	0.727
2	8	3	0	149	1.000	0.981	0.727
3	8	3	0	149	1.000	0.981	0.727
4	8	3	0	149	1.000	0.981	0.727
5	8	3	0	149	1.000	0.981	0.727
6	8	3	0	149	1.000	0.981	0.727
7	7	4	0	149	1.000	0.975	0.636
8	6	5	0	149	1.000	0.969	0.545
9	6	5	0	149	1.000	0.969	0.545
10	6	5	0	149	1.000	0.969	0.545
11	6	5	0	149	1.000	0.969	0.545
12	6	5	0	149	1.000	0.969	0.545
13	6	5	0	149	1.000	0.969	0.545
14	6	5	0	149	1.000	0.969	0.545
15	6	5	0	149	1.000	0.969	0.545
16	6	5	0	149	1.000	0.969	0.545
17	6	5	0	149	1.000	0.969	0.545
18	6	4	0	150	1.000	0.975	0.600
19	6	4	0	150	1.000	0.975	0.600
20	6	4	0	150	1.000	0.975	0.600
21	6	3	0	151	1.000	0.981	0.667
22	6	3	0	151	1.000	0.981	0.667
23	6	3	0	151	1.000	0.981	0.667

24	6	5	0	149	1.000	0.969	0.545
25	6	5	0	149	1.000	0.969	0.545
26	6	5	0	149	1.000	0.969	0.545
27	7	4	0	149	1.000	0.975	0.636
28	7	4	0	149	1.000	0.975	0.636
29	7	4	0	149	1.000	0.975	0.636
30	7	4	0	149	1.000	0.975	0.636
31	7	4	0	149	1.000	0.975	0.636
32	7	4	0	149	1.000	0.975	0.636
33	7	4	0	149	1.000	0.975	0.636
34	7	4	0	149	1.000	0.975	0.636
35	7	4	0	149	1.000	0.975	0.636
36	7	2	0	151	1.000	0.988	0.778
37	8	2	0	150	1.000	0.988	0.800
38	8	2	0	150	1.000	0.988	0.800
39	8	2	0	150	1.000	0.988	0.800
40	8	2	0	150	1.000	0.988	0.800
41	8	2	0	150	1.000	0.988	0.800
42	8	2	0	150	1.000	0.988	0.800
43	8	2	0	150	1.000	0.988	0.800
44	8	2	0	150	1.000	0.988	0.800
45	8	2	0	150	1.000	0.988	0.800
46	8	2	0	150	1.000	0.988	0.800
47	8	2	0	150	1.000	0.988	0.800
48	8	2	0	150	1.000	0.988	0.800
49	8	2	0	150	1.000	0.988	0.800
50	8	2	0	150	1.000	0.988	0.800
51	8	2	0	150	1.000	0.988	0.800
52	8	2	0	150	1.000	0.988	0.800
53	8	2	0	150	1.000	0.988	0.800
54	8	2	0	150	1.000	0.988	0.800
55	8	2	0	150	1.000	0.988	0.800
56	8	2	0	150	1.000	0.988	0.800
57	8	2	0	150	1.000	0.988	0.800
58	7	2	0	151	1.000	0.988	0.778
59	8	2	0	150	1.000	0.988	0.800
60	8	2	0	150	1.000	0.988	0.800
61	8	2	0	150	1.000	0.988	0.800
62	9	2	0	149	1.000	0.988	0.818
63	9	2	0	149	1.000	0.988	0.818
64	9	2	0	149	1.000	0.988	0.818
65	9	2	0	149	1.000	0.988	0.818
66	9	2	0	149	1.000	0.988	0.818

67	9	2	0	149	1.000	0.988	0.818
68	9	2	0	149	1.000	0.988	0.818
69	9	2	0	149	1.000	0.988	0.818
70	7	2	0	151	1.000	0.988	0.778
71	7	2	0	151	1.000	0.988	0.778
72	9	1	0	150	1.000	0.994	0.900
73	9	1	0	150	1.000	0.994	0.900
74	9	1	0	150	1.000	0.994	0.900
75	9	1	0	150	1.000	0.994	0.900
76	9	1	0	150	1.000	0.994	0.900
77	9	1	0	150	1.000	0.994	0.900
78	9	2	0	149	1.000	0.988	0.818
79	9	1	0	150	1.000	0.994	0.900
80	9	1	0	150	1.000	0.994	0.900
81	9	1	0	150	1.000	0.994	0.900
82	9	1	0	150	1.000	0.994	0.900
83	9	1	0	150	1.000	0.994	0.900
84	9	1	0	150	1.000	0.994	0.900
85	9	1	0	150	1.000	0.994	0.900
86	9	1	0	150	1.000	0.994	0.900
87	9	1	0	150	1.000	0.994	0.900
88	8	3	0	149	1.000	0.981	0.727
89	8	3	0	149	1.000	0.981	0.727
90	8	3	0	149	1.000	0.981	0.727
91	8	3	0	149	1.000	0.981	0.727
92	8	3	0	149	1.000	0.981	0.727
93	8	3	0	149	1.000	0.981	0.727
94	7	4	0	149	1.000	0.975	0.636
95	7	4	0	149	1.000	0.975	0.636
96	7	4	0	149	1.000	0.975	0.636
97	7	4	0	149	1.000	0.975	0.636
98	8	3	0	149	1.000	0.981	0.727
99	8	3	0	149	1.000	0.981	0.727
100	8	3	0	149	1.000	0.981	0.727
平均					1.000	0.983	0.739

表 2 則 是使用 5000 張正樣本與 5000 負樣本的參數，
偵測與追蹤的結果

表 2，以 5000 張正樣本與 5000 負樣本訓練參數偵測 vidf 影片前 100 影格的結果

	實際有人		實際無人		Precision	Accuracy	Recall
影格 編號	判讀有人 TP	判讀無人 FN	判讀有人 FP	判讀無人 TN	TP/(TP+FP)	(TP+TN) / (TP+TN+FP+FN)	TP/(TP+FN)
1	7	4	0	149	1.000	0.975	0.636
2	7	4	0	149	1.000	0.975	0.636
3	7	4	0	149	1.000	0.975	0.636
4	7	4	0	149	1.000	0.975	0.636
5	7	4	0	149	1.000	0.975	0.636
6	7	4	0	149	1.000	0.975	0.636
7	7	4	0	149	1.000	0.975	0.636
8	7	4	0	149	1.000	0.975	0.636
9	5	6	1	148	0.833	0.956	0.455
10	5	6	1	148	0.833	0.956	0.455
11	5	6	1	148	0.833	0.956	0.455
12	5	4	1	150	0.833	0.969	0.556
13	5	4	0	151	1.000	0.975	0.556
14	6	5	0	149	1.000	0.969	0.545
15	6	5	0	149	1.000	0.969	0.545
16	6	5	0	149	1.000	0.969	0.545
17	6	5	0	149	1.000	0.969	0.545
18	6	5	0	149	1.000	0.969	0.545
19	6	5	0	149	1.000	0.969	0.545
20	5	7	0	148	1.000	0.956	0.417
21	6	6	0	148	1.000	0.963	0.500
22	7	5	0	148	1.000	0.969	0.583
23	7	5	0	148	1.000	0.969	0.583
24	7	5	0	148	1.000	0.969	0.583
25	7	5	0	148	1.000	0.969	0.583
26	7	5	0	148	1.000	0.969	0.583
27	7	5	0	148	1.000	0.969	0.583
28	7	6	0	147	1.000	0.963	0.538
29	7	6	0	147	1.000	0.963	0.538

30	7	6	0	147	1.000	0.963	0.538
31	7	6	0	147	1.000	0.963	0.538
32	7	6	0	147	1.000	0.963	0.538
33	7	6	0	147	1.000	0.963	0.538
34	7	5	0	148	1.000	0.969	0.583
35	7	5	0	148	1.000	0.969	0.583
36	7	3	0	150	1.000	0.981	0.700
37	9	2	0	149	1.000	0.988	0.818
38	9	2	0	149	1.000	0.988	0.818
39	9	2	0	149	1.000	0.988	0.818
40	9	2	0	149	1.000	0.988	0.818
41	9	3	0	148	1.000	0.981	0.750
42	9	3	0	148	1.000	0.981	0.750
43	9	3	0	148	1.000	0.981	0.750
44	9	3	0	148	1.000	0.981	0.750
45	9	3	0	148	1.000	0.981	0.750
46	8	4	0	148	1.000	0.975	0.667
47	9	3	0	148	1.000	0.981	0.750
48	8	4	0	148	1.000	0.975	0.667
49	8	4	0	148	1.000	0.975	0.667
50	8	4	0	148	1.000	0.975	0.667
51	8	4	0	148	1.000	0.975	0.667
52	8	4	0	148	1.000	0.975	0.667
53	8	4	0	148	1.000	0.975	0.667
54	8	4	0	148	1.000	0.975	0.667
55	8	4	0	148	1.000	0.975	0.667
56	7	3	0	150	1.000	0.981	0.700
57	7	3	0	150	1.000	0.981	0.700
58	7	3	0	150	1.000	0.981	0.700
59	8	2	0	150	1.000	0.988	0.800
60	8	2	0	150	1.000	0.988	0.800
61	9	1	0	150	1.000	0.994	0.900
62	9	1	0	150	1.000	0.994	0.900
63	9	2	1	148	0.900	0.981	0.818
64	9	2	1	148	0.900	0.981	0.818
65	9	2	1	148	0.900	0.981	0.818
66	8	2	1	149	0.889	0.981	0.800
67	8	2	1	149	0.889	0.981	0.800
68	9	2	0	149	1.000	0.988	0.818
69	9	2	0	149	1.000	0.988	0.818
70	9	2	0	149	1.000	0.988	0.818
71	9	1	0	150	1.000	0.994	0.900
72	9	1	0	150	1.000	0.994	0.900

73	9	1	0	150	1.000	0.994	0.900
74	9	1	0	150	1.000	0.994	0.900
75	9	1	0	150	1.000	0.994	0.900
76	9	1	0	150	1.000	0.994	0.900
77	9	1	0	150	1.000	0.994	0.900
78	9	2	0	149	1.000	0.988	0.818
79	9	2	0	149	1.000	0.988	0.818
80	9	2	0	149	1.000	0.988	0.818
81	9	2	0	149	1.000	0.988	0.818
82	9	2	0	149	1.000	0.988	0.818
83	9	2	0	149	1.000	0.988	0.818
84	9	2	0	149	1.000	0.988	0.818
85	9	2	0	149	1.000	0.988	0.818
86	9	2	0	149	1.000	0.988	0.818
87	8	3	0	149	1.000	0.981	0.727
88	8	3	0	149	1.000	0.981	0.727
89	8	3	0	149	1.000	0.981	0.727
90	7	4	1	148	0.875	0.969	0.636
91	7	4	1	148	0.875	0.969	0.636
92	7	4	1	148	0.875	0.969	0.636
93	7	4	1	148	0.875	0.969	0.636
94	7	4	1	148	0.875	0.969	0.636
95	7	4	1	148	0.875	0.969	0.636
96	8	3	0	149	1.000	0.981	0.727
97	8	3	0	149	1.000	0.981	0.727
98	8	3	0	149	1.000	0.981	0.727
99	8	3	0	149	1.000	0.981	0.727
100	8	3	0	149	1.000	0.981	0.727
平均					0.981	0.977	0.694

假設 9000 樣本實驗結果為 A 組，5000 樣本實驗結果為 B 組，則實驗分析如下：

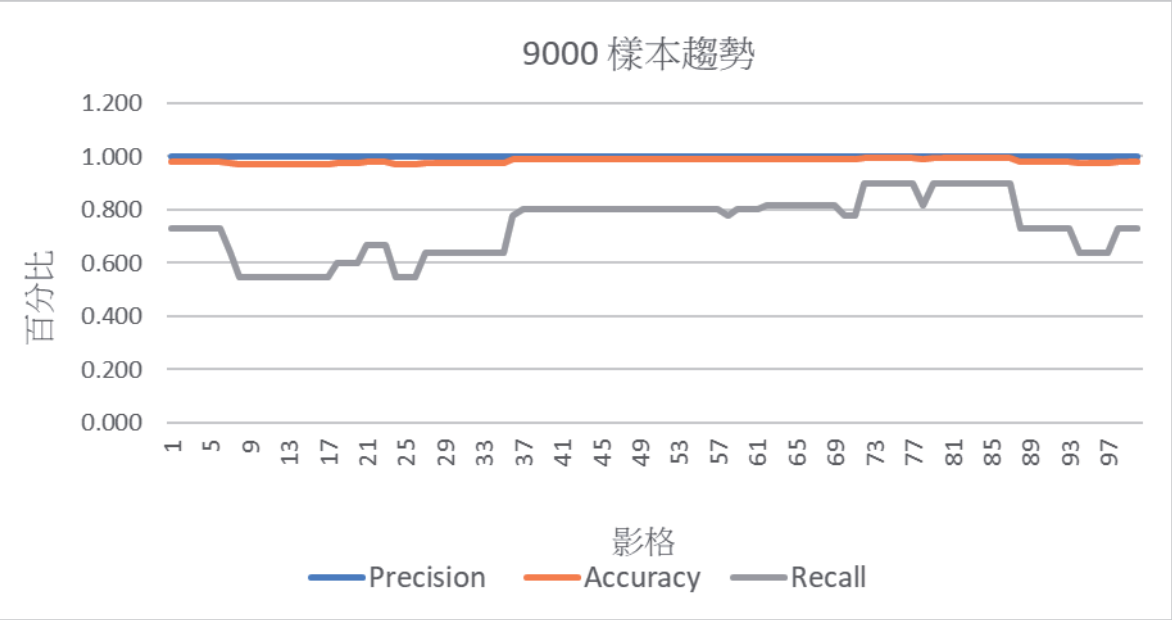


圖 9，9000 樣本實驗組結果趨勢

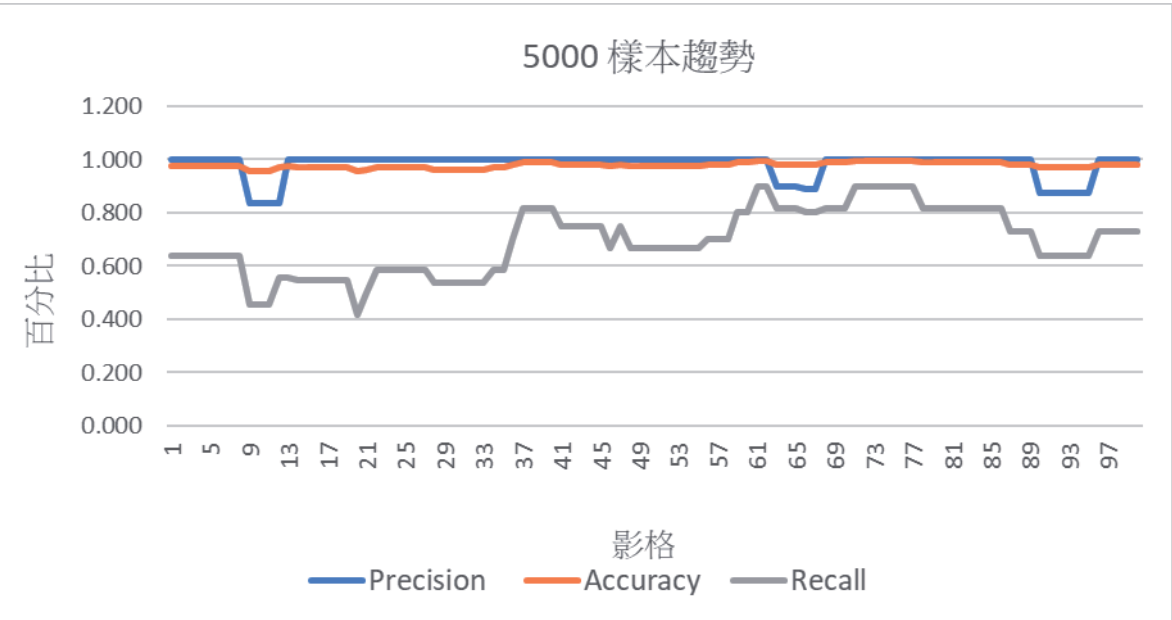


圖 10，5000 樣本實驗組結果趨勢

由圖 9 和圖 10 可見，第 33 格影格之前，其偵測率相較中段低，是因為在整體影像其鏡頭邊緣有著一些人沒有完全入境，導致分類器無法明顯區分是人或非人，但是進入中段影格後，行人陸陸續續移動後，分類器也逐漸能夠正確分類了。

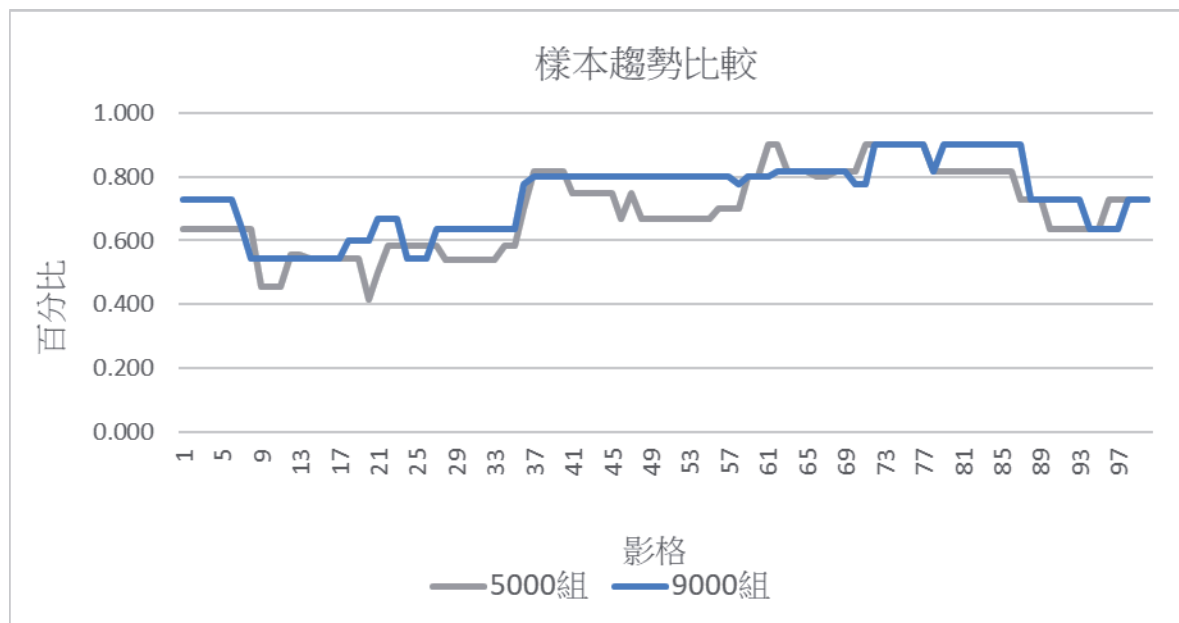


圖 11，兩組實驗結果偵測率趨勢比較

再由圖 9 和圖 10 比較可見圖 11 的偵測率趨勢差異，A/B 兩組樣本訓練後的結果用在分類時，其 B 組實驗在 37 到 61 影格間相較 A 組實驗有著很大的浮動，因為這段影格有著部分重疊的行人出現，導致樣本數少的 B 組這組分類器無法明確分類辨識出來，甚至分類錯誤，導致整體正確率下降。

然而 A 組分類器因為有很大量的各種訓練樣本，而助益於各種不同型態的影像分類，正可用來正確區分是否為行人，提升偵測率和準確率。

	Name	Precision	Accuracy	Recall
A 組	9000 張 正負樣本	1.000	0.983	0.739
B 組	5000 張 正負樣本	0.981	0.977	0.694
差異		0.019	0.005	0.044

表 3，兩組實驗結果比較

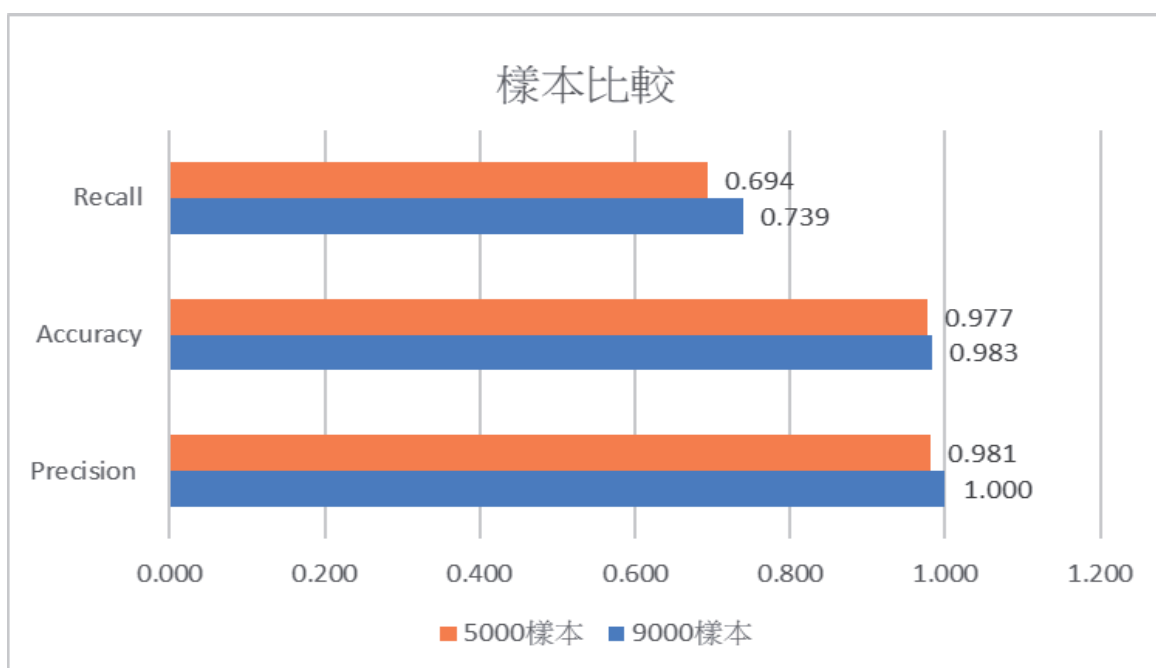


圖 12，兩組實驗結果比較

整體來看，由表 3 見到第 A/B 組實驗，比較後可以發現，兩組之間的整體準確率和整體檢測率會因為訓練樣本數量的關係而有所差異，尤其是 A 組檢測率對比 B 組多達 4% 的漲幅，B 組則因為訓練樣本數少，其偵測率和準確率都相對不高。

由此可見，如果訓練的樣本越多，準確率和偵測率將會越高，尤其在偵測率的成長更明顯，對於後續的追蹤也會有好的影響。

第4章 結論

在小型裝置中，因為計算能力比較低，無法使用需要高計算量的演算法。我們採用的方法，先以 LBP 特徵搭配 cascade classifier 找出行人所在的位置，再以追蹤演算法追蹤物體的移動軌跡。

在現在，基於類神經網路的深度學習方法已經是行人追蹤領域最佳的方法，基於 LBP 特徵的方法仍有其適用的時機。在小型的裝置與即時應用上，類神經網路和雲端運算再回傳結果的方式有其侷限性。我們的研究結果探索了小型裝置與即時要求。

我們以小型的嵌入式系統 Raspberry 3 B (第三代樹梅派) 測試我們方法的執行時間。其測試的結果 frame per second (FPS) 為 15，因此我們的方法，在人的移動速度不快的情形下可以及時處理追蹤行人的工作。

參考文獻

- [1]. Alex Krizhevsky, Sutskever, Ilya, Hinton, Geoffrey E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.
- [2]. T Ojala, M Pietikainen, Maenpaa. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971-987
- [3]. Paul Viola, Michael J.Jones, (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57, 137-154.
- [4]. Antoni B Chan, Nuno Vasconcelos, (2012). Counting People with Low-Level Features and Bayesian Regression. *IEEE Trans. on Image Processing (TIP)*, 21(4), 2170-2177.