

國立臺南大學資訊工程學系

「行動普適計算」課程
期中報告

題目：

Activity Recognition-Realtime

班級 ： 資工碩一
姓名 ： 杜威慶
學號 ： M10459011

老師：林朝興

中華民國 104 年 12 月 2 日

目錄

一、簡介及功能介紹.....	1
二、理論介紹.....	3
三、程式架構.....	5
四、系統架構.....	6
五、實驗環境與結果.....	7
六、心得.....	13
七、參考文獻.....	14

一、簡介及功能介紹

I. 介紹

利用手機的三軸加速器收集資料，參考論文[1]裡的 43 個特徵值和 k nearest neighbors (KNN)的演算法與 Android 程式，去構成一個可以，辨認坐下、站立、走路、騎車、跑步、上樓梯、下樓梯的 APP。

為了避免手機晃動太大，導致於會增加誤判的結果，所以如(圖二)幫手機裝上手機套，褲子可穿如(圖一)，可以讓手機比較能固定不易搖晃。實驗時須將手機放置於褲子的口袋裡。



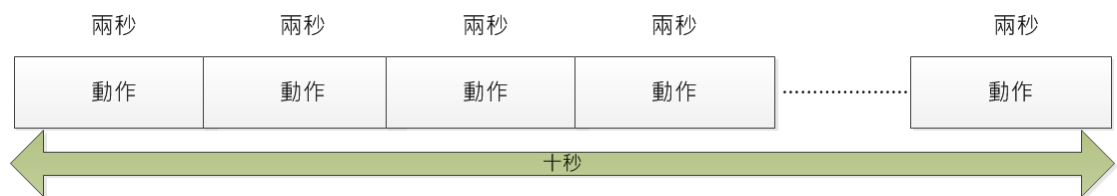
圖一、牛仔褲



圖二、手機套

II. 功能要求

- 1、取出三軸加速器的值(X、Y、Z 軸)，並分別取 43 個特徵值、正規化、標記類別，弄好的檔案放入手機裡，方便 APP 程式讀取。
- 2、手機讀取三軸加速器的值(X、Y、Z 軸)，做完(1)的描述後，與訓練資料比對，KNN 演算法 K 值為 3，投三票。
- 3、一筆為兩秒，十筆二十秒顯現出來(圖三)。



圖三、動作顯示

III. 功能介紹

- 1、一開始延遲十秒。
- 2、十秒後開始，播放提示音。
- 3、測完二十秒，結果顯示出來，並震動與提示音。

二、理論介紹

I. k nearest neighbors(KNN)演算法

是同一類別的物件，若以高度空間中的點來表示，則這些點的距離應該會比較接近。因此，那麼對於一個未知類別的一筆資料，我們只要找出來在訓練資料中和此筆資料最接近的點，就可以判定此筆資料的類別應該和最接近的點的類別是一樣的。最近鄰居分類法是一個最直覺的分類法，在測試各種分類器時，幫被當成是最基礎的分類器，以便和其他更複雜的分類器進行效能比較。

在機器學習中是屬於監督是學習(Supervised learning)，算法是所有算法中理論最簡單，最好理解的。KNN 是一種基於實例的學習，通過計算新數據與訓練數據特征值之間的距離，然後選取 K ($K \geq 1$) 個距離最近的鄰居進行分類判斷（投票法）或者回歸。如果 $K=1$ ，那麼新數據被簡單分配給其近鄰的類。需有一個訓練的資料(training)，輸入一個測試資料(test)，看測試資料跟訓練資料距離屬於哪個類別。

II. 特徵值

從手機的三軸加速器取出值後，分別對 X、Y、Z 軸做特徵值的計算，參考論文[1]的算法：

1. Average[3]: Average acceleration (for each axis)三個軸的平均[1]
2. Standard Deviation[3]: Standard deviation (for each axis)三個軸的標準差。[1]
3. Average Absolute Difference[3]: Average absolute difference between the value of each of the 200 readings within the ED and the mean value over those 200 values (for each axis)三個軸的平均絕對差值，這裡是 100 values[1]

4. Average Resultant Acceleration[1]: Average of the square roots of the sum of the values of each axis squared $\sqrt{x_i^2 + y_i^2 + z_i^2}$ ，三軸的平方總和開根號。[1]
5. Time Between Peaks[3]: Time in milliseconds between peaks in the sinusoidal waves associated with most activities (for each axis)三軸的波峰時間[1]
6. Binned Distribution[30]: We determine the range of values for each axis (maximum – minimum), divide this range into 10 equal sized bins, and then record what fraction of the 200 values fell within each of the bins.三軸的 10 各區間的落點個數，在這裡是 100 values [1]

III. 正規化

因為每個特徵值有大有小，若直接執行 knn 演算法的話，再算距離的時候，小的值會被大的值吃掉，導致動作的誤判，所以資料都須經過正規化，讓每筆特徵值資料的數據能夠被規範在 0~1 的區間內。

以下是正規化的公式：

1. 找出最大(max)與最小(min)值：

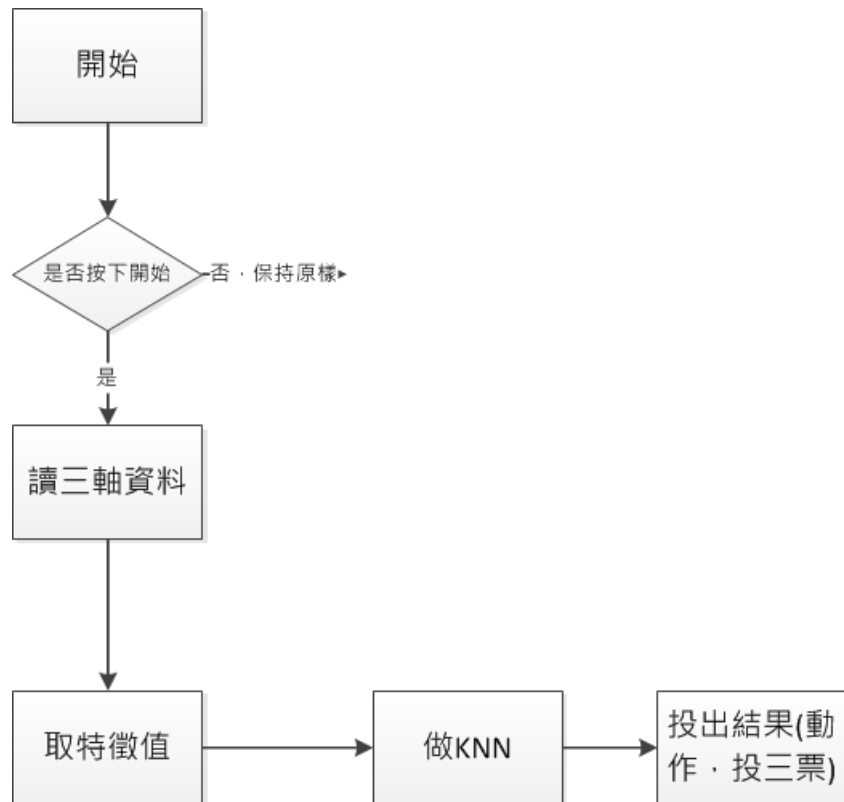
$$\max = \text{Max}(\text{data}), \min = \text{Min}(\text{data})$$

2. 每筆資料 data_i 帶入以下公式：

$$\text{data}_i = \frac{(\text{data}_i - \min)}{(\max - \min)}$$

三、程式架構

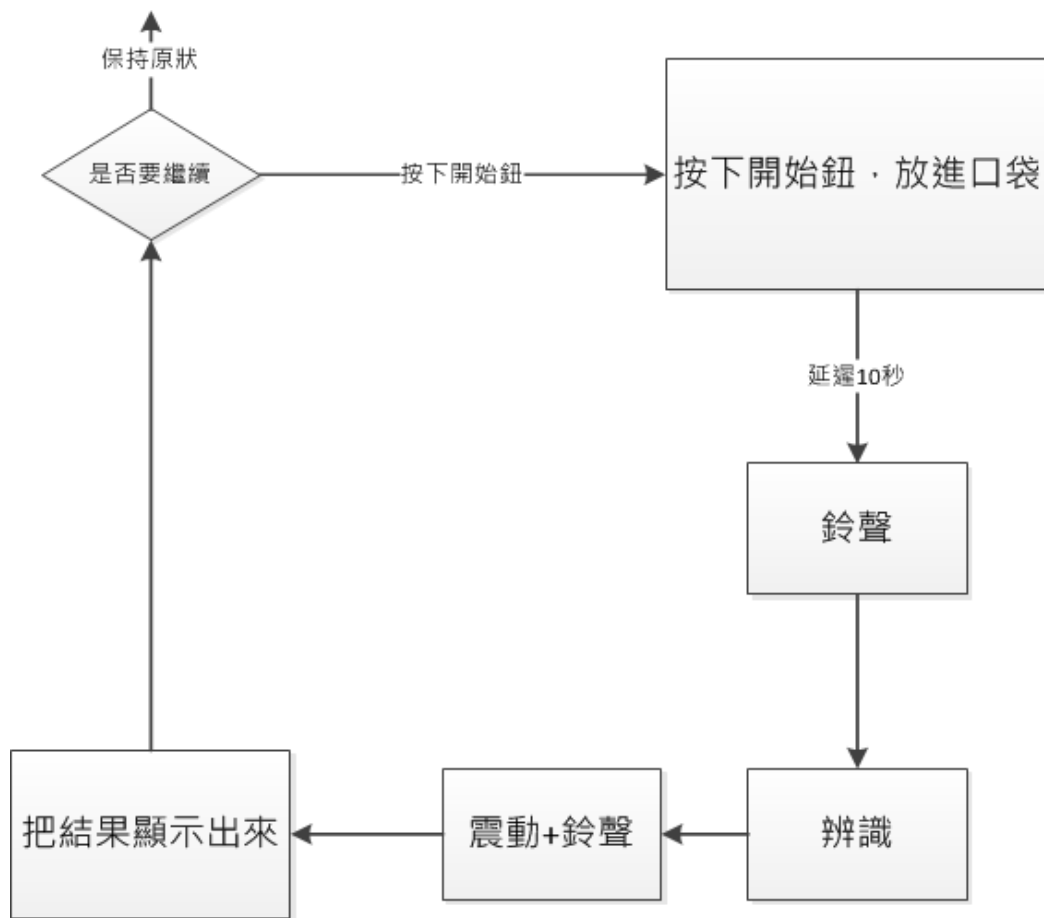
- 1、一開始如果按了開始鍵，手機開始讀取三軸加速器的資料。
- 2、再來取資料的特徵值。
- 3、與訓練資料做 KNN，取前 K 個短的距離。
- 4、印出投票後的結果(動作)。



圖四、程式架構

四、系統架構

- 1、按下開始鑑，放進口袋。
- 2、延遲 10 秒後，會有提示聲。
- 3、二十秒的時間，做辨識的動作。
- 4、二十秒後，會有提示音與震動，提醒測試者。
- 5、把結果(動作)顯示出來。



圖五、系統架構

五、實驗環境與結果

I. 測試環境與參數

樣本資料數:1 次兩秒，兩秒總共有 100 筆資料，做 10 次二十秒。

KNN 投票數:為三票。

手機規格:



圖六、手機

採用 Android 4.4 KitKat 作業系統。

◎ 內建 64 位元 Qualcomm Snapdragon 410, 1.2GHz 四核心處理器。

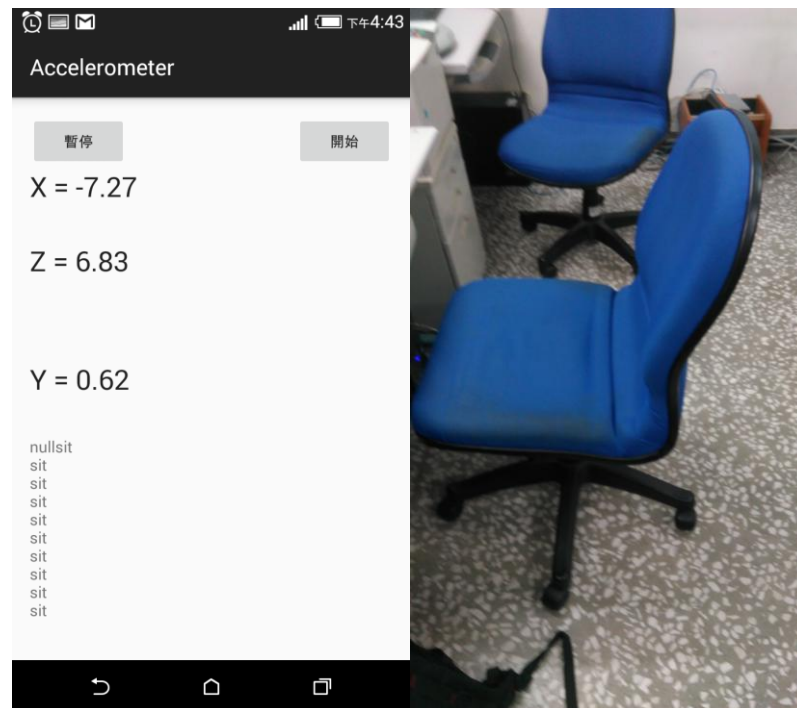
◎ 5 吋 720P HD 觸控螢幕、1,280 x 720pixels 螢幕解析度。

◎ 內建 1GB RAM / 16GB ROM 儲存空間。

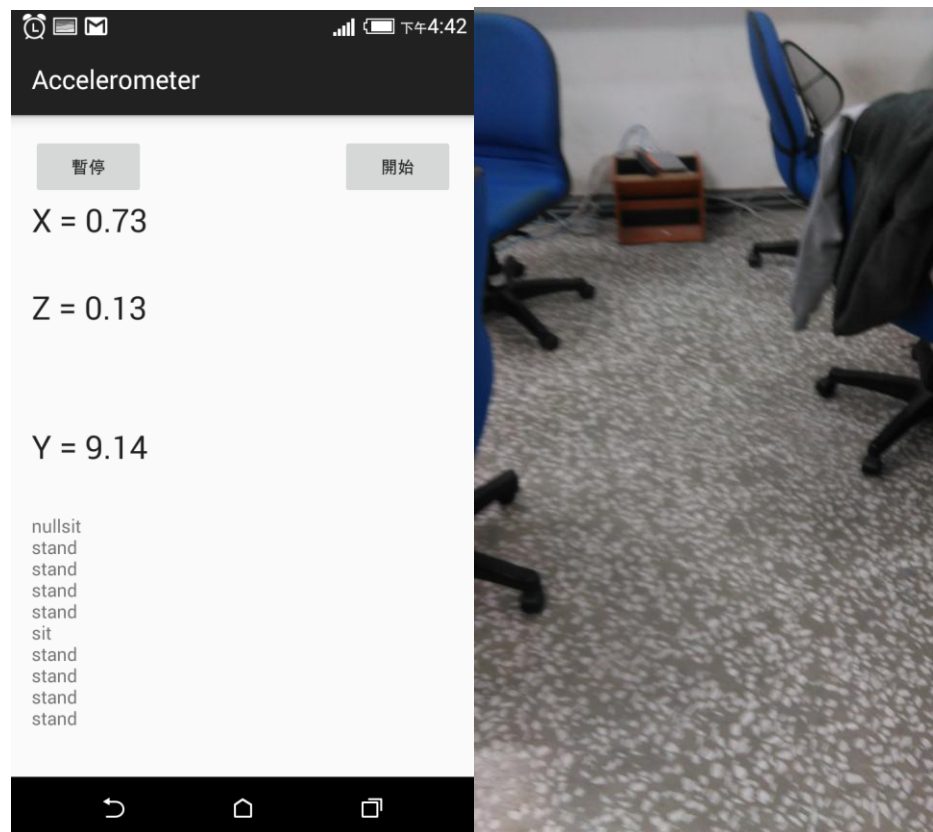
◎ 支援 microSD 記憶卡擴充，最高至 128GB 記憶體容量。

II. 測試地點與結果

1. 坐下地點在實驗室的椅子，站立地點在實驗室。

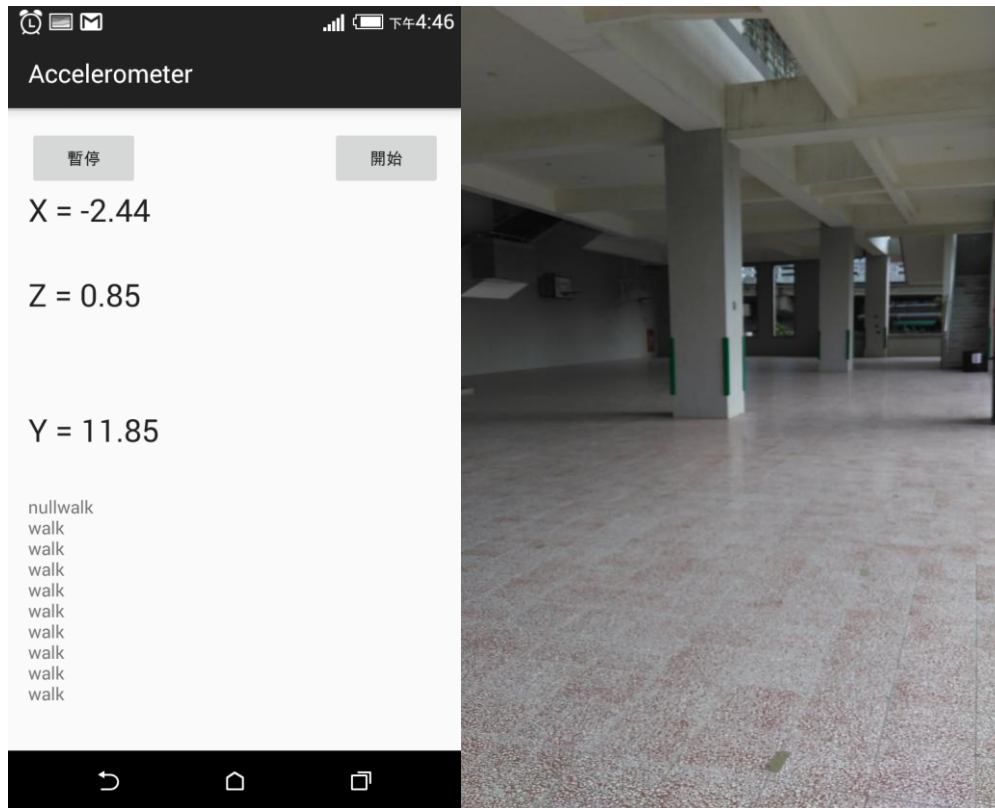


圖七、坐下

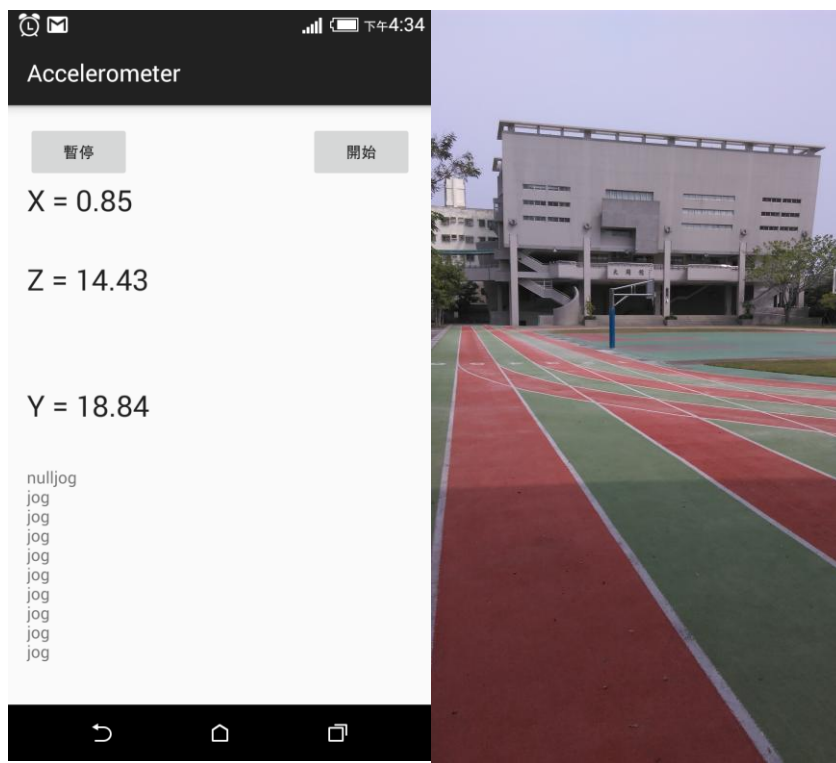


圖七、站立

2、走路是在大同館的一樓，跑步則在大同國小的操場。

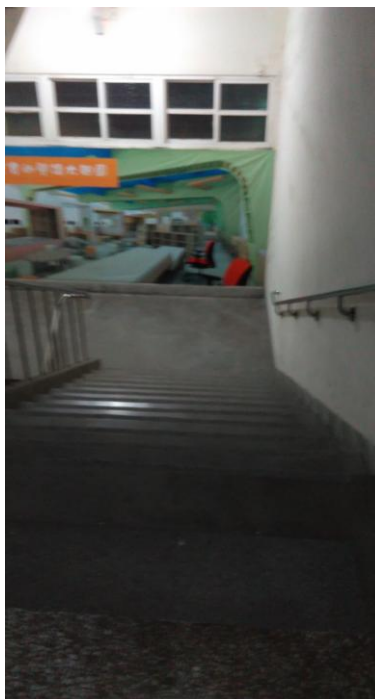
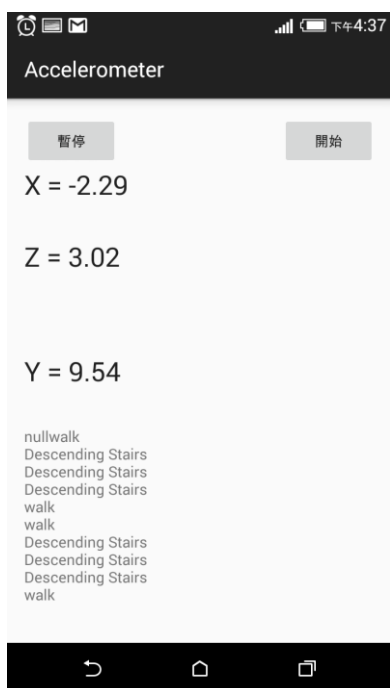


圖七、走路

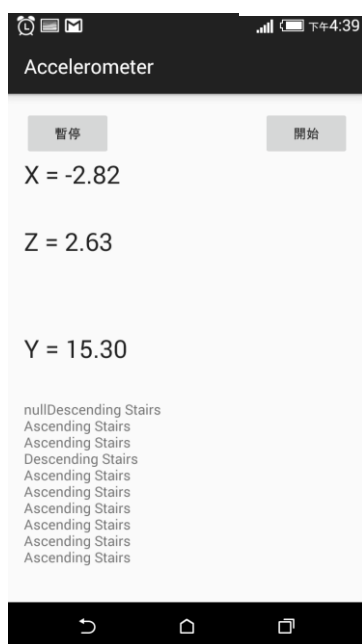


圖八、跑步

3、上、下樓都在實驗室的大樓樓梯。

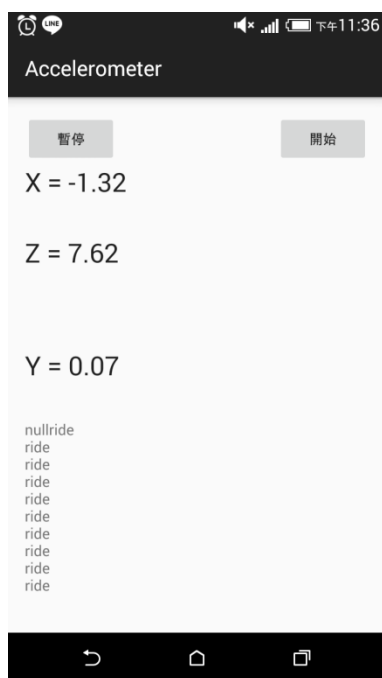


圖九、上樓



圖十、下樓

5、騎車地點是在運動場旁的道路。



圖十一、騎車

III. 執行結果:

	SIT	STAND	WALK	RIDE	JOG	ASCENDING STAIRS	DESCENDING STAIRS
SIT	9	1	0	1	0	0	0
STAND	0	9	0	0	0	0	0
WALK	0	0	9	0	0	0	2
RIDE	1	0	0	8	0	1	0
JOG	0	0	0	0	10	0	0
ASCENDING STAIRS	0	0	0	1	0	8	0
DESCENDING STAIRS	0	0	1	0	0	1	8
Precision	81%	100%	52%	88%	100%	88%	80%
Recall	90%	90%	90%	80%	100%	80%	80%
accuracy	87%						

在下樓梯容易被誤辦成走路，其他動作如:坐下、走路、站立...等等，都有相當高的準確率以及回照率。

六、心得

一開始修得時候，老師就說會寫到手機程式，想說自己已經有底子了，可以修這門課。

學期中，利用手機收集三軸加速器的值，把每種動作的值，利用 EXCEL 畫出來，至於收集的程式網路上已經有成熟的 OPEN SOURCE 了，再來是看論文[1]，要開始計算特徵值，前面幾個特徵值有給公式，所以在程式碼上的撰寫很容易，但在波峰與區間落點花比較久的時間。

當製作出訓練資料後，開始撰寫手機 APP 的程式，過程中有曾遇到只顯示坐下或站立的情況，後來檢查到正規化多一個迴圈，落點區間的陣列忘了歸零，改完後，準確率就有很明顯的提升，在測試的過程中，跑步最累，平常很少在運動，測試跑步的時候很喘。

最後在 DEMO 完的那一刻，覺得自己好棒，看著可以準確判斷自己目前的動作作品，很有成就感，從中累積了除錯、演算法實現的經驗與能力。

七、參考文獻

- [1] Kwapisz, Jennifer R., Gary M. Weiss, and Samuel A. Moore. "Activity recognition using cell phone accelerometers." *ACM SigKDD Explorations Newsletter* 12.2 (2011): 74-82.