

Recognition of Human Queuing Behaviors Using Accelerometers

高新惟	黃俐君	吳政瑋	曾煜棋
國立交通大學資訊科 學與工程研究所 scott02308@gmail.com	國立交通大學資訊科 學與工程研究所 lijyunhuang@gmail.com	國立交通大學資訊科學 與工程研究所 cww0403@cs.nctu.edu.tw	國立交通大學資訊科 學與工程研究所 yctseng@cs.nctu.edu.tw

摘要

以自動化方式辨識顧客排隊的行為能裨益顧客及商家。然而，先前的研究較少針對多行近似平行隊伍的情境而設計。本研究以智慧型手機內的加速度計為基礎，實作一套適用於多行近似平行隊伍情境的排隊辨識架構，其主要包括以下三個模組。

(1) 走路動作辨識模組：其利用智慧型手機內的加速度資料辨識用戶是否正在走路。
(2) 排隊者辨識模組：其利用用戶一段時間內循序移動的行為推論該用戶是否處於排隊狀態。
(3) 隊伍人數辨識模組：利用排隊者的移動特徵及分群法辨識哪些排隊者排在相同的隊伍。本研究實作上述模組並收集真實資料對各模組進行實驗評估及探討。

關鍵詞：群體行為辨識、排隊行為辨識、機器學習、感測器、行動運算。

1. 緒論

隨著科技的蓬勃發展，智慧型手機已成為許多人生活不可或缺的工具。除了通訊功能外，智慧型手機還提供許多有趣的功能，如：網路存取、多媒體影音播放、行動計算以及無線通訊等功能。此外，智慧型手機內部經常內嵌著許多感測器，如：光線感測器(Light Sensor)、加速度計(Accelerometer)、羅盤(Compass)以及磁力計(Gyroscope)等。

由於智慧型手機具備行動計算、低建置成本、易於攜帶及使用方便的特性，目前已有許多研究[1][3][4]以智慧型手機為基礎發展相當有趣的新興應用。其中一門新興研究領域為人類行為辨識(Human Activity Recognition; HAR)[1-4][5][8][10]，其主要目的乃利用感測器以自動化方式辨識個體或群體的行為。

人類行為辨識是近年來一個相當活躍的研究議題，這門技術已在不同領域被廣泛地研究，特別是體適能健康管理、軍事訓練、醫療與銀髮族長照等等領域。雖然目前已有許多研究致力於簡單的個體行為辨識(Individual Activity Recognition)，卻較少探討群體行為辨識(Group Activity Recognition)。

排隊行為(Queuing Behavior)是人類生活中常見的一種群體行為，在許多公共場所中，例如：大賣場、遊樂園、電影院及餐廳，人們經常花費大量的時間在排隊等待上。在排隊時，人們很自然地想知道還需要等待多久才能被櫃檯人員服務。當有多行隊伍時，知道每行隊伍的等待時間及服務時間能夠同時裨益顧客及業者。對於顧客而言，該資訊能幫助顧客選擇等待時間較短的隊伍，進而節省顧客寶貴的時間。對業者而言，該資訊能幫助業者管理顧客排隊的行為以及人力資源，進而提升顧客滿意度及獲利。

目前已有一些相關研究致力於以攝影機、Wi-Fi、智慧型手機內的感測裝置辨識人類排隊的行為。LineKing [1]利用 Wi-Fi 訊號強度及 GPS 定位來計算出排隊者的等待時間。也有其他學者透過 Wi-Fi 訊號強度去區分出排隊者的排隊狀態，其中包含等待、服務、離開。雖然上述方法在某些應用中可能具備不錯的執行效率，他們僅針對單一隊伍的情境(Single-line Queuing Scenarios)而設計，並未考慮多行隊伍的情境(Multiple-line Queuing Scenarios)。QueueSense [12]和 QueueVadis [11]皆考慮多行隊伍的情境。然而，QueueSense 採協同式架構[12]，因而可能導致傳輸成本過高及延遲過久等問題。QueueVadis 並非針對多行近似平行隊伍的情境而設計，導致 QueueVadis 在該情境下較容易判斷錯誤。

有鑑於此，本研究率先提出一套適用於多行近似平行隊伍情境的排隊辨識架構，其主要包括以下三個核心模組。(1) 走路動作辨識模組：其利用手機內的加速度資料低階特徵 (Low-level Feature) 及決策樹 (Decision Tree) 辨識用戶是否正在走路。(2) 排隊者辨識模組：其利用用戶循序移動的高階特徵 (High-level Feature) 及支援向量機 (Support Vector Machine) [6] 推論該用戶是否處於排隊狀態。(3) 隊伍人數辨識模組：其利用 K-Means [6] 及凝聚式階層分群法 (Agglomerative Hierarchical Clustering) [6] 辨識哪些排隊者排在相同的隊伍。

我們實作上述模組並以真實資料評估上述各模組之正確率。我們在交通大學內的餐廳收集使用者排隊的真實資料。在我們實驗結果中，走路動作辨識模組的正確率高達 97%；排隊者辨識模組具備相當不錯的正確率，其正確率約介於 75% 至 80% 之間；在隊伍人數辨識模組方面，該模組在兩行及三行隊伍時最佳的正確率約介於 65% 至 79% 之間。

本論文之後續章節安排如下。第二節回顧與排隊辨識相關的研究。第三節介紹本研究提出的方法架構。第四節以真實資料評估我們所提出的方法。最後一節為結論及未來工作。

2. 相關研究

於此章節，我們探討與排隊辨識相關的研究。我們將相關的研究歸納為三類，分別為：(1) 以攝影機為基礎的方法、(2) 以 Wi-Fi 為基礎的方法、(3) 以手機為基礎的方法。以下分別敘述各類方法之主要特色及其優缺點。

2.1 以攝影機為基礎的方法

現今有一些方法(如：[2][7][9])利用監控攝影機以及影像處理技術偵測出不同人群移動的方向。然而，這些方法無法從擷取的影片中自動地判斷出哪些人正在排隊。此外，這些方法經常需要架設多台監控攝影機於數個公共區域。這種方式不僅需要昂貴的建置成本，還可能侵犯顧客的隱私權。此外，影像辨識的結果也可能受到環境因素的影響(如：光線、攝影機位置和環境背景色等)而導致辨識效率不佳。

2.2 以 Wi-Fi 為基礎的方法

LineKing [1] 是由 Bulut 等學者所提出的排隊等待時間預估系統，其利用 GSP (Global Positioning System) 定位技術以及 Wi-Fi AP 預估排隊者的等候時間。Wang 等學者 [13] 利用手機內的 Wi-Fi 訊號變化推論排隊者處於哪種排隊狀態，包括：排隊等待中、於櫃台接受服務、離開櫃台。雖然這兩個方法皆利用 Wi-Fi 訊號來推論排隊者的相關資訊，但他們僅適用於單行隊伍的情境，並未考慮多行隊伍的情境，也無法用來辨識排隊者屬於哪一行隊伍。

2.3 以智慧型手機為基礎的方法

QueueSense [12] 和 QueueVadis [11] 皆考慮多行隊伍的情境。QueueSense 可分為兩個階段。在第一階段中，QueueSense 透過智慧型手機以協同式方式交換鄰居手機的感測資料，並利用這些資料辨識使用者是否正在排隊。在第二階段中，QueueSense 利用分群法將排於相同隊伍的排隊者分群至相同的群中。由於 QueueSense 採協同式架構，相鄰的手機必須互相交換資料，因而可能導致傳輸成本過高及延遲過久等問題。QueueVadis 首先利用一個兩階層式的分類器去辨識使用者是否正在排隊，再利用另一個分類器去判斷兩兩排隊者的移動模式是否屬於相同隊伍。由於 QueueVadis 利用羅盤的角度資訊來區分兩兩排隊者是否同屬於相同隊伍，當不同隊伍朝向櫃檯的角度差異較大時，QueueVadis 會具備相當不錯的正確率。然而，QueueVadis 並未針對多行近似平行隊伍的情境而設計。在該情境下，不同隊伍朝向櫃檯的角度差異較小，QueueVadis 的正確率可能會大幅受到影響。此外，在 QueueVadis 中，以分類器判斷排隊者屬於哪個隊伍需事先收集大量訓練資料。在實務上，建置模型的成本較高。

3. 方法架構

本章節介紹我們所設計的方法架構。於本章節中，我們首先介紹本研究所考慮的排隊情境，再概述方法的主要架構及各模組的目的，最後詳細描述各模組的核心技術。

3.1 所考慮的排隊情境

圖 1 顯示本研究所考慮的排隊情境。我們假設每位使用者都攜帶著智慧型手機，並將智慧型手機放置於口袋中或平穩的拿在手上。而使用者可分為非排隊者(Non-queueer)以及排隊者(Queueer)兩類。在一個區域內，共有 N ($N \geq 1$) 行面向櫃檯的隊伍，每一行隊伍至少有兩位排隊者且近似於直線形狀。櫃檯之間互相平行且面向同一方向，如：麥當勞的櫃台。

3.2 架構概述

本研究以智慧型手機內的加速度計為基礎，設計一套適用於多行近似平行隊伍情境的排隊辨識架構，其主要包括以下三個模組：

(1) 走路動作辨識模組：在智慧型手機上建立分類器模型，並以此來作為我們排隊行為判斷的依據。我們收集智慧型手機上的加速度值，從智慧型手機內的加速度資料所萃取的低階特徵辨識用戶是否正在走路。

(2) 排隊者辨識模組：透過走路動作辨識模組的結果，我們可以得到使用者循序移動的行為，從中擷取高階特徵後，我們使用分類模型判斷該使用者是否為排隊者。

(3) 隊伍人數辨識模組：應用 K-Means 及凝聚式階層分群法將不同的人區分到不同的隊列中。利用排隊者的循序移動特徵以及不同隊伍會有不同的服務時間跟隊伍前進時間，來將使用者進行分群，將相似度高的一群排隊者歸類為同一隊伍。

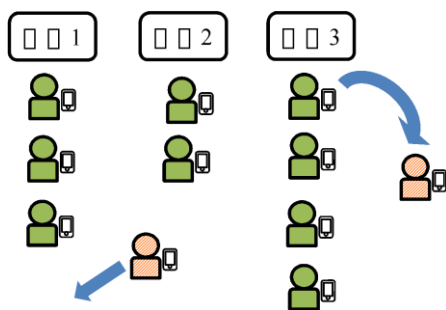


圖 1 本研究所考慮的排隊情境示意圖

3.3 走路動作辨識模組

排隊是一種先來先服務(First-Come, First-Served; FCFS)的行為。在一行隊伍中，若有一位排隊者往前移動，則在其後面的排隊者也會在一小段延遲後往前移動。相較於非排隊者，排隊者會較有規律且斷斷續續地交錯著往前移動(走路動作)及停下來等待(站立動作)的動作。基於上述排隊者的行為特性，我們可利用[14]所提的方法，藉由智慧型手機內的加速度資料推論每 θ 秒內使用者正在走路或站立，藉此得知使用者於 ω ($\omega > \theta$) 秒內循序移動的行為。

在系統的實作中，我們利用安卓(Android)智慧型手機內的加速度計(Accelerometer)感測裝置，擷取三軸加速度串流資料(3-axis Accelerometer Data)。該資料由 x 、 y 及 z 軸的資料所組成，其分別代表前-後、上-下、左-右維度的加速度資料。在資料收集的過程中，本研究採用 50Hz 的取樣頻率(Sampling Rate)，也就是每秒對於每一個軸向取樣 50 筆感測資料。

我們採用不重疊的滑動視窗(Non-overlapped Sliding Window)模式切割加速度計資料。我們將 θ 設定為 1 秒，並以 θ 為單位切割所收集到的串流感測資料。在本研究中，經過切割處理後所得的資料稱為資料片段(Data Segment)。每個滑動視窗由 ω 個連續的資料片段所組成。基於我們在實作過程中的實驗測試， ω 設為 24。

對於每一個資料片段 S ，我們對其擷取低階特徵(Low-level Features)，如表 1 所示。令 $A = \langle a_1, a_2, \dots, a_n \rangle$ 為資料片段 S 中任意一軸的時序資料，其中 a_i 為 A 中第 i ($1 \leq i \leq n$) 次取樣的結果。我們對 A 擷取平均值(Mean)、變異數(Variance)及過零率(Zero Crossing Rate; ZCR)等低階特徵，其定義如下：

- **平均值(Mean)：**用來量測時序資料 A 的中央趨勢，其定義如下：

$$\mu(A) = \frac{\sum_{i=1}^n a_i}{n}。$$

- **變異數(Variance)：**來測量時序資料 A 的變異量，也就是資料相對於 $\mu(A)$ 的離散程度，其定義如下：

$$\text{Var}(A) = \frac{\sum_{i=1}^n (a_i - \mu(A))^2}{n}。$$

表 1 走路動作辨識模組所擷取的低階特徵

Mean	$\mu(x), \mu(y), \mu(z)$
Variance	$var(x), var(y), var(z)$
ZCR	$ZCR(x), ZCR(y), ZCR(z)$
Covariance	$cov(x, y), cov(y, z), cov(x, z)$
Correlation	$corr(x, y), corr(y, z), corr(x, z)$

表 2 循序移動特徵

特徵編號	高階特徵
F_1	Chg
F_2	$Num(R, W)$
F_3	$Min(R, W)$
F_4	$Avg(R, W)$
F_5	$Max(R, W)$
F_6	$Num(R, S)$
F_7	$Min(R, S)$
F_8	$Avg(R, S)$
F_9	$Max(R, S)$

- **過零率(Zero Crossing Rate)**：用來測量時序資料 A 中連續兩個資料點跨過零的總次數，其定義如下：

$$ZCR(A) = \sum_{i=2}^n \text{diff}(a_i, a_{i-1}),$$

其中若 $\text{diff}(\cdot)$ 的輸入值一個為正數且一個為負數，則 $\text{diff}(\cdot)$ 輸出 1；否則， $\text{diff}(\cdot)$ 輸出 0。

令 $A = \{a_1, a_2, \dots, a_n\}$ 和 $B = \{b_1, b_2, \dots, b_n\}$ 為資料片段 S 中加速度計的任兩軸時序資料，以下是共變異數及相關係數的定義：

- **共變異數(Covariance)**：為 A 與 B 共同變化的趨勢測量值，其定義如下：

$$\text{Cov}(A, B) = \frac{\sum_{i=1}^n |(a_i - \mu(A))(b_i - \mu(B))|}{n}.$$

- **相關係數(Correlation)**：為 A 與 B 的共變異數除以 A 和 B 的標準差乘積，其定義如下：

$$\text{Corr}(A, B) = \frac{\text{Cov}(A, B)}{\sigma_A \sigma_B},$$

其中 σ_A 及 σ_B 分別為 A 與 B 的標準差。

擷取低階特徵後，我們套用[6]的方法，利用一個預先建立好的分類器來分類使用者在 θ 秒內為走路狀態或站立狀態。此分類器的輸入為擷取低階特徵後的資料片段，其輸出為走路或站立。我們利用 Weka 內的 J48 決策樹分類方法[15]建立此分類模型，並將模型建置於使用者的手機端。

(SSSSSSSSSSWSSSSSSSSSSWSSSS)

圖 2 使用者的循序移動序列示意圖

3.4 排隊者辨識模組

由於排隊者會有規律且斷斷續續地交錯著往前移動(走路動作)及停下來等待(站立動作)的動作。而非排隊者的移動模式較不會具有上述特性。因此，我們以 ω 為單位，收集走路動作辨識模組的 ω 個連續判斷結果。這 ω 個判斷結果由走路及站立所組成，又稱為使用者的循序移動序列 (Movement Sequence)，其代表使用者在 ω 秒內循序移動的方式。

舉例說明，圖 2 為一個使用者的循序移動序列，其中 W 代表走路動作(Walking)， S 代表站立(Standing)動作。令 T 為 W (走路)或 S (站立)，對於每一個循序移動序列 R ，我們以[11]中所提的方法對其擷取下列高階特徵，也就是循序移動特徵：

- **Chg** ：在 R 中動作改變的總次數。在圖 2 中， S 轉變為 W 或 W 轉變為 S 的總次數為 4 次。
- **$Num(R, T)$** ：在 R 中，動作 T 發生的總次數。在圖 2 中，動作 S 發生的總次數為 21；動作 W 發生的總次數為 3。
- **$Min(R, T)$** ：在 R 中，動作 T 的最小連續發生次數。在圖 2 中，動作 S 的最小連續發生次數為 $\min\{9, 8, 4\} = 4$ ；動作 W 的最小連續發生次數為 $\min\{2, 1\} = 1$ 。
- **$Avg(R, T)$** ：在 R 中，動作 T 的平均連續發生次數。在圖 2 中，動作 S 的平均連續發生次數為 $(9+8+4)/3 = 7$ ；動作 W 的平均連續發生次數為 $(2+1)/2 = 1.5$ 。
- **$Max(R, T)$** ：在 R 中，動作 T 的最大連續發生次數。在圖 2 中，動作 S 的最大連續發生次數為 $\max\{9, 8, 4\} = 9$ ；動作 W 的最大連續發生次數為 $\max\{2, 1\} = 2$ 。

我們利用一個預先建立好的分類器來分類使用者在 ω 秒內的行為較屬於排隊狀態或是非排隊狀態。此分類器的輸入為上述的 9 個高階特徵，其輸出為排隊者或非排隊者。在實作上，我們利用 Weka 內的 SVM 分類演算法[15]建置分類模型，並將排隊者辨識模組實作於手機端。

輸入： (1) k ：群集的數目； (2) D ：包含 n 筆資料的資料集；
輸出： (1) k 個群集；
演算法步驟： 01 隨機選取 D 中 k 筆資料作為初始的群中心； 02 計算每一筆資料與各群中心之距離； 03 對每一筆資料 r ，找出離其最近的群中心 c ， 將 r 視為與 c 同群； 04 依據各群集內的資料找出新的群中心； 05 重複步驟 02 至 05 直到群中心不變；

圖 3 K-Means 分群法之虛擬碼

輸入： (1) k ：群集的數目； (2) D ：包含 n 筆資料的資料集；
輸出： (1) k 個群集；
演算法步驟： 01 將 D 中的每一筆資料視為一群； 02 如果兩個群 C_a 及 C_b 有最小的距離 $dis(C_a, C_b)$ ， 則合併 C_a 及 C_b 形成一群； 03 重複步驟 02 及 03，直到群集數量等於 k ；

圖 4 凝聚式階層分群法之虛擬碼

3.5 隊伍人數辨識模組

由於排於相同隊伍的排隊者會有比較相似的循序移動特徵，而不同隊伍的排隊者其循序移動特徵較不相似，我們可以利用這項特性以分群方法將具有相似循序移動特徵的使用者歸納在同一群。

在實作上，若排隊者辨識模組的辨識結果為排隊狀態，則系統會將該使用者的辨識碼 (Identifier) 及循序移動特徵傳送到 Server 端的隊伍人數辨識模組集中處理。隊伍人數辨識模組會將所收集到的資料輸入至分群演算法中，以辨識哪些使用者排在相同的隊伍。在本研究中，我們所嘗試的分群方法為 K-Means[6] 及凝聚式階層分群法[6]。圖 4 及圖 5 分別顯示 K-Means 及凝聚式階層分群法的虛擬碼。

在 K-Means 中，資料與群中心的距離計算方式是採用歐幾里德距離 (Euclidean Distance)[6]。假設資料 x 及群中心 y (或資料 y) 的特徵值分別為 $(x_1, x_2, \dots, x_m)$ 及 $(y_1, y_2, \dots, y_m)$ ，則 x 與 y 的歐幾里德距離為：

$$dis(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

其中 m 為循序移動特徵的總數量 (如表 2)，也就是 9。

在凝聚式階層分群法中，我們嘗試以 Average Linkage、Single Linkage 及 Complete Linkage 等三種不同的方式計算群與群的距離。令 C_a 及 C_b 為兩個不同的群， $|C_a|$ 及 $|C_b|$ 分別代表 C_a 及 C_b 群內的資料筆數，則 Average Linkage、Single Linkage 及 Complete Linkage 的距離計算方式如下。

- **Average Linkage**： C_a 及 C_b 的距離定義為 C_a 中各點至 C_b 中各點的平均距離：

$$dis(C_a, C_b) = \frac{1}{|C_a| |C_b|} \sum_{i=1}^{|C_a|} \sum_{j=1}^{|C_b|} dis(u_i, u_j)$$

- **Single Linkage**： C_a 及 C_b 的距離定義為 C_a 及 C_b 中最接近的兩點的距離：

$$dis(C_a, C_b) = \min_{u_i \in C_a, u_j \in C_b} dis(u_i, u_j)$$

- **Complete Linkage**： C_a 及 C_b 的距離定義為 C_a 及 C_b 中最遠的兩點的距離：

$$dis(C_a, C_b) = \max_{u_i \in C_a, u_j \in C_b} dis(u_i, u_j)$$

4. 實驗評估

於此章節，我們以實驗評估走路動作辨識模組、排隊者辨識模組及隊伍人數辨識模組之執行效率。我們在交通大學內的不同餐廳收集真實排隊者的資料。資料收集的時間為非假日的中午十二點至下午一點之間，也就是餐廳內有最多消費者的時段。每位受測者攜帶一隻 Google Nexus4 的智慧型手機，並在手機內預先安裝我們實作的程式。Google Nexus4 智慧型手機的硬體規格為 1.5GHz Qualcomm S4APQ8064 Processor、2GB Memory，Android 的版本為 4.2。Server 端的硬體規格為 3.40 GHz Intel Core Processor、4GB Memory 及 Windows 7 作業系統。我們以 MYSQL 管理 Server 端的資料。所有的程式皆以 JAVA 程式語言撰寫。

表 3 走路動作辨識模組之訓練資料及測試資料特性

	走路狀態	站立狀態	總筆數
訓練資料	75	75	150
測試資料	50	50	100

表 4 混亂矩陣

	被模型分類為 C_i	被模型分類為 非 C_i
在測試資料中其 目標屬性為 C_i	TP	FN
在測試資料中其 目標屬性不為 C_i	FP	TN

表 5 走路動作辨識模組的混亂矩陣

	被模型分類為 走路狀態	被模型分類為 站立狀態
在測試資料中其 目標屬性為走路狀態	48	2
在測試資料中其 目標屬性為站立狀態	1	49

表 6 走路動作辨識模組的效率

	準確率	召回率	F-度量
走路狀態	96%	98%	97%
站立狀態	98%	94%	94%

4.1 走路動作辨識模組之效率評估

於此實驗中，我們評估走路動作辨識模組內的分類器的執行效率。我們利用智慧型手機收集訓練資料及測試資料，其資料特性如表 3 所示。令 $C = \langle C_1, C_2, \dots, C_{|C|} \rangle$ 為目標屬性的集合，對於每一類目標屬性 C_i ($1 \leq i \leq |C|$)，我們可以依據測試資料被模型分類的結果建構一個如表 4 的混亂矩陣 (Confusion Matrix)。在表 4 中，TP、FP、FN 和 TN 分別表示真陽性(True Positives)、真陰性(True Negatives)、偽陽性(False Positives)和偽陰性(False Negatives)[6]。

基於表 4 的混亂矩陣，我們計算模型的正確率(Accuracy)、模型對於目標屬性 $C_i \in C$ ($1 \leq i \leq |C|$)的準確率(Precision)、召回率(Recall)以及 F-度量(F-Measure)，其計算方式如下所示：

- **Accuracy** : $\frac{TP+TN}{TP+FP+FN+TN}$ 。
- **Precision(C_i)** : $\frac{TP}{TP+FP}$ 。
- **Recall(C_i)** : $\frac{TP}{TP+FN}$ 。
- **F-Measure(C_i)** : $\frac{2 \times \text{Precision}(C_i) \times \text{Recall}(C_i)}{\text{Precision}(C_i) + \text{Recall}(C_i)}$ 。

表 5 為走路動作辨識模組的混亂矩陣。

從表 5 我們可看出走路動作辨識模組只誤判 3 筆測試資料，其正確率高達 97%。表 6 顯示走路動作辨識模組對於不同目標屬性的準確率、召回率及 F-度量。如表 6 所示，走路動作辨識模組具備相當優良的辨識結果，為後續的處理奠定良好的基礎。

4.2 排隊者辨識模組之效率評估

接著，我們評估排隊者辨識模組的效率。於此實驗中，我們共設計了三個實驗情境。第一個實驗情境為交通大學校園內的第一餐廳。於此實驗情境中，我們考慮單行隊伍的情況，在每次的實驗中有二到四位受測者參與排隊。第二個實驗情境為交通大學校園內的第二餐廳。於此實驗情境中，我們考慮兩行隊伍的情況，在每次的實驗中有八位受測者參與。我們將受測者分配在兩行不同的隊伍，每行隊伍中包含四位受測者。第三個實驗情境為交通大學校園內的麥當勞。於此實驗情境中，我們考慮三行隊伍的情況，在每次的實驗中有十二位受測者參與。我們將受測者分配在三行不同的隊伍，每行隊伍中包含四位受測者。在上述的實驗中，我們不僅收集排隊者的資料，也收集非排隊者的資料。表 7 及表 8 分別顯示此實驗所用的訓練資料及測試資料特性。表 7 及表 8 中的 #QR 及 #NQ 分別代表目標屬性為排隊狀態及非牌對狀態的資料筆數。

我們測試排隊者辨識模組使用不同分類器時的效果。參與比較的分類方法包括支援向量機(Support Vector Machine; SVM)、J48 決策樹(Decision Tree) 以及深度神經網路(Deep Neural Network; DNN)。SVM 及 J48 使用 Weka 中預設的參數。我們使用 Weka 中的多層感知器(Multilayer Perceptron) 套件實作 DNN，其網路結構為一層輸入層(Input Layer)、兩層隱藏層(Hidden Layer)以及一層輸出層(Output Layer)。學習率(Learning Rate)為 0.025，動量(Momentum)為 0.6，迭代數為 13000。

表 9 顯示排隊者辨識模組使用不同分類器時的正確率。如表 9 所示，排隊者辨識模組搭配 SVM 時會有較佳的正確率，其正確率約介於 72.4% 至 78.4%。此外，在三種情境下，SVM 的正確率皆優於 J48 及 DNN。

表 7 排隊者辨識模組之訓練資料特性

情境	實驗次數	類型	#QR	#NQ
第一餐廳	25	一行隊伍	50	50
第二餐廳	20	兩行隊伍	50	50
麥當勞	20	三行隊伍	50	50

表 8 排隊者辨識模組之測試資料特性

情境	實驗次數	類型	#QR	#NQ
第一餐廳	25	一行隊伍	50	50
第二餐廳	20	兩行隊伍	50	50
麥當勞	20	三行隊伍	50	50

表 9 排隊者辨識模組之正確率

	SVM	J48	DNN
第一餐廳	79.7%	75%	78.1%
第二餐廳	78.3%	71.7%	76.7%
麥當勞	71.4%	69%	70.2%

表 10 排隊者辨識模組之宏平均準確率

	SVM	J48	DNN
第一餐廳	79.8%	75.6%	78.1%
第二餐廳	78.4%	71.9%	76.8%
麥當勞	72.4%	69.5%	70.9%

表 11 排隊者辨識模組之宏平均召回率

	SVM	J48	DNN
第一餐廳	79.7%	75%	78.1%
第二餐廳	78.3%	71.7%	76.7%
麥當勞	71.4%	69%	70.2%

表 12 排隊者辨識模組之宏平均 F-指標

	SVM	J48	DNN
第一餐廳	79.7%	75%	78.1%
第二餐廳	78.3%	71.6%	76.6%
麥當勞	70.8%	68.6%	69.7%

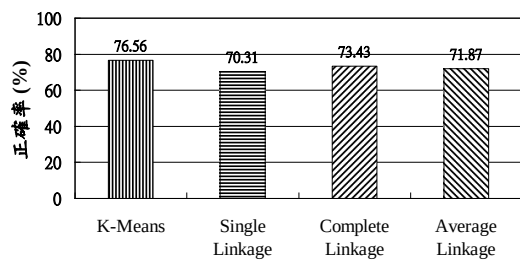


圖 5 兩行隊伍情境時的分群結果

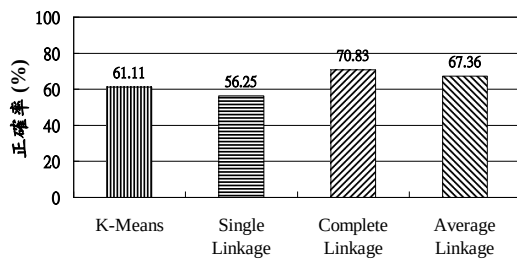


圖 6 三行隊伍情境時的分群結果

接著，我們以宏平均準確率(Macro-average Precision；簡稱 MP)、宏平均召回率(Macro-average Recall；簡稱 MR) 以及宏平均 F-度量(Macro-average F-Measure；簡稱 MF)評估排隊者辨識模組對於不同目標屬性的綜合辨識效率，其計算方式如下所示：

$$\bullet \text{ MP : } \frac{1}{|C|} \sum_{i=1}^{|C|} \text{Precision}(c_i)。$$

$$\bullet \text{ MR : } \frac{1}{|C|} \sum_{i=1}^{|C|} \text{Recall}(c_i)。$$

$$\bullet \text{ MF : } \frac{1}{|C|} \sum_{i=1}^{|C|} \text{F-Measure}(c_i)。$$

從表 10 至表 12 中，我們可看出排隊者辨識模組搭配 SVM 時對於不同目標屬性的綜合辨識效率優於其搭配 J48 及 DNN 時的情況。在資料探勘及機器學習領域，學者們普遍認為 SVM 對於數值型資料的辨識能力較強。由於排隊者辨識模組所使用的特徵皆為數值型資料，這可能是 SVM 的辨識結果會優於其它分類器的原因之一。

此外，排隊者辨識模組的辨識率隨著隊伍行數增加而略有降低。這是因為當隊伍行數增加時，排隊者的行為更添多樣化。若所收集到的訓練資料多樣性不足，或是未擷取到具鑑別力的特徵，都可能造成模型效率不佳。

4.3 隊伍人數辨識模組之效率評估

最後，我們評估隊伍人數辨識模組的效率。由於在第一餐廳所收集到的資料僅考慮單行隊伍，並未有多行隊伍的情況，我們僅使用在第二餐廳及麥當勞所收集到的資料對此辨識模組進行評估。基於此模組在測試資料上的結果，我們可以建構一個如表 13 的混亂矩陣。此模組的正確率計算方式為：

$$\frac{TP + TN}{TP + FP + FN + TN}。$$

表 13 排隊人數辨識模組的混亂矩陣

	被模組分在相同群中	被模組分在不同群中
實際情況為同隊伍	TP	FN
實際情況為不同隊伍	FP	TN

圖 5 及圖 6 分別顯示隊伍人數辨識模組在兩行隊伍及三行隊伍情境下的正確率。如圖 5 所示，在兩行隊伍的實驗情境中，隊伍人數辨識模組的正確率大約介於 70% 至 80% 左右，且 K-Means 分群法有最高的正確率。如圖 6 所示，在三行隊伍的實驗情境中，隊伍人數辨識模組的正確率約介於 50% 至 70% 左右，且當凝聚式階層分群法搭配 Complete Linkage 時的效果最好。

從這個真實的實驗結果中，我們可得知當隊伍行數較少時，K-Means 可能是一個不錯的選擇；然而，當隊伍行數增加時，凝聚式階層分群法搭配 Complete Linkage 的效果會比 K-Means 好。此外，從此實驗我們可看出，隊伍人數辨識模組的準確率隨著隊伍行數的增加而減少。原因可能是因為隨著隊伍數增加，有越多循序移動特徵相似的使用者。另外，若不同隊伍的服務時間相似，也可能造成隊伍人數辨識模組誤判。

5. 結論與未來工作

以智慧型手機內建的感測器辨識人類排隊的行為是近年來一門新興且重要的研究。在論文中，我們以智慧型手機內的加速度計為基礎，設計一套適用於多行近似平行隊伍的排隊辨識架構，其包括走路動作辨識模組、排隊者辨識模組以及隊伍人數辨識模組。我們實作上述各模組，並以真實資料評估所提出的架構。我們在交通大學內的不同餐廳收集真實的使用者排隊資料。從實驗結果來看，走路動作辨識模組的正確率高達 97%；排隊者辨識模組也有不錯的正確率，正確率大約介於 75% 至 80% 之間；隊伍人數辨識模組在兩行及三行隊伍的最加正確率大約介於 65% 至 79% 之間。

從我們真實的實驗結果中，可得知以智慧型手機辨識人類的排隊行為仍然是相當具有挑戰性的議題，並且存在許多改善空間。於未來工作中，我們預計整合更多元的技術於所提出的架構中，例如 Wi-Fi 訊號或手機內建的羅盤等資訊，提高各個模組的辨識率，並且以更豐富的實驗評估所提出的方法。

參考文獻

- [1] F. Bulut, M. Demirbas, and H. Ferhatosmanoglu, "LineKing: coffee shop wait-time monitoring using smartphones," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2045-2058, 2015.
- [2] M. Chang, N. Krahnstoeber, S. Lim, and T. Yu., "Group level activity recognition in crowded environments across multiple cameras," in *Proc. Of Advanced Video and Signal Based Surveillance*, pp. 56-63, 2010.
- [3] B. Chun and P. Maniatis., "Augmented smartphone applications through clone cloud execution," in *Proc. Of the 15th Workshop on Hot Topics in Operating Systems*, vol. 9, pp. 8-11, 2009.
- [4] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin., "Diversity in smartphone usage," in *Proc. Of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 179-194, 2010.
- [5] T. Huynh, M. Fritz, and B. Schiele. "Discover of activity patterns using topic models," in *Proc. Of the 2008 Ubiquitous Computing*, pp. 10-19, 2008.
- [6] J. Han, M. Kamber, and J. Pei. "*Data mining: concepts and techniques, 3rd edition*," Morgan Kaufmann, 2011.
- [7] T. Joshua, D. Vin, and L. John. "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319-2323, 2000
- [8] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proc. Of the 7th International Conference on Mobile Systems*, pp. 165-178, 2009.
- [9] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. Of the 2005 IEEE Computer Society*

- Conference on Computer Vision and Pattern Recognition**, vol. 1, pp. 878-885, 2005.
- [10] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in **Proc. Of the 6th ACM Conference on Embedded Network Sensor System**, pp. 337-350, 2008.
 - [11] T. Okoshi, Y. Lu, C. Vig, Y. Lee, R. Balan, and A. Misra. "QueueVadis: Queuing analytics using smartphones," in **Proc. Of the 14th International Conference on Information Processing in Sensor Networks**, pp. 214-225, 2015.
 - [12] L. Qiang, Q. Han, X. Chengand, and L. Sun. "Collaborative recognition of queuing behavior on mobile phones," **IEEE Transactions on Mobile Computing**, vol. 15, no. 1, pp. 60-73, 2016.
 - [13] Y. Wang, J. Yang, Y. Chen, H. Liu, M. Gruteser, and P. Martin. "Tracking human queues using single-point signal monitoring," in **Proc. Of the 12th Annual International Conference on Mobile Systems, Applications and Services**, pp. 42-54 2014.
 - [14] Y. Wang, J. Wang, and X. Zhang. "Qtime: a queuing-time notification system based on participatory sensing data," in **Proc. Of the 37th Annual Computer Software and Applications Conference**, pp.770-777, 2013.
 - [15] **Weka 3: data mining software in Java**, Available at: <http://www.cs.waikato.ac.nz/ml/weka/>
 - [16] **LIBSVM: A Library for Support Vector Machines**, Available at: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>