健身家庭教練

Workout Home Coach

指導教授：韓欽銓

學生：李沛誼、陳于寧、程思嘉、張芝語、陳品萱

國立聯合大學 資訊工程學系

苗栗市南勢里聯大 2 號

{cchan, U0824007, U0824016, U0824018, U0824019, U0824022}@nuu.edu.tw

摘要

後疫情時代的來臨，健身浪潮崛起，許多人卻可能因懼怕疫情而選擇購買健身器材在家自行訓練，居家健身所產生的問題提高了運動傷害的發生。因此我們希望設計一套在家自主訓練的系統，讓人們不用出門就能享受到如健身教練般的專業指導。

在系統中，我們會利用感測器去收集教練標準的動作數據放入模型，透過機器模型訓練，作為辨識動作的標準，使用此系統時能夠改善錯誤動作。穿戴裝置後會利用藍牙進行數據傳輸，讓使用者的動作與人物模型同步，也會將該動作進行姿態比對，訓練結束後，系統會自動計算動作正確率和消耗熱量等並顯示於結算畫面，便於使用者隨時了解自己平時的訓練成果。

關鍵詞:健身、身體鍛鍊、重量訓練、動作追蹤、姿態辨識

Abstract

With the advent of the post-epidemic era and the rising tide of fitness, many people may choose to purchase fitness equipment to train on their own at home for fear of the disease.The problems associated with home fitness have increased the incidence of sports injuries.Therefore, we want to design a home training system so that people can enjoy the professional guidance like a fitness trainer without leaving home.

In the system, we will use the sensor to collect the trainer's standard movement data into the model, through the machine model training, as a standard to identify the movement, and the use of the system can improve the wrong movement.After wearing the device, data transmission will be performed by Bluetooth, so that the user's movements can be synchronized with the model, and the movements will be compared with the posture model.After training, the system will automatically calculate the correct rate of movements and the amount of heat consumed and display it on the checkout screen, so that the user can always know the results of his or her regular training.

Keywords: Fitness, Physical Exercise, Weight training, Motion Tracking, Posture Recognition

一、前言

對於小資族而言，使用我們的居家健身系統只須花費一次購買設備及訓練動作的費用，不蓄意弄壞就能使用一輩子，相較於上健身房健身每年都需要花上一筆年費來說，可謂是一個經濟實惠的選擇。而與健身房相同，我們的系統也能透過動作數據的辨識模擬專業的健身教練對動作進行指導，讓民眾自行在家中訓練時，也能避免因為動作的不標準，導致運動傷害的產生。

現在與姿態辨識較相關，且又廣為人知的技術是 OpenCV，而本論文未使用此技術是因為 OpenCV 辨識時的環境會受到許多限制，像是背景過暗或是過亮、附近有與手部相近的顏色、與其他部位重疊…等都會影響辨識出的準確率，所以我們選擇使用穿戴式硬體的方式來進行辨識，使系統不受到環境的影響。

在我們的系統中，我們會利用 IMU 感測器去收集健身教練標準的動作數據放入模型中，透過 Keras 輔助模型的建構去進行訓練，作為日後辨識動作的標準，可在使用者使用此系統時能夠即時校正動作，並糾正使用者的錯誤，使人們可以不需出門也不用花費上健身房高額的費用就能享受如健身教練一對一的專業指導。在穿戴裝置時會利用藍牙的方式將動作數據進行傳輸，也能讓使用者了解其動作與人體實時動態 3D 模型可同步及校正，且使用者在做動作時，資料會進行動作的比對，當使用者訓練結束後，系統會自動計算動作正確率和消耗熱量等資料並顯示相關的動作資料於結算畫面中，不需以人為的方式去記錄相關資料，也能便於使用者隨時了解自己在當日及各個月份的訓練成果。

二、系統內容

系統架構如圖 1 所示，使用 Android Studio 設計的 app 透過 Wi-Fi 與伺服器端的 MySQL 資料庫進行連結，並根據不同使用者從伺服器端的資料庫中取得相應的資料，再存入內建的 SQLite 資料庫中進行使用。當使用者確認要進行的訓練動作後，app 會將選擇的動作名稱透過 Wi-Fi 上傳至 MySQL 資料庫中，並跳轉至 Unity 的畫面。首先 Unity 會與硬體的 BLE 藍牙進行配對、訂閱，並取得四元數以及正確率的資料。接著將取得到的四元數資料放入 Unity 模型中進行運動追蹤，並顯示使用者目前的動作在模型中展示的畫面。接著 Unity 會透過 Wi-Fi 取得 MySQL 資料庫中的開始動作，並等待使用者輸入想要進行動作的次數，輸入完次數後會將動作名稱及次數透過 BLE 傳輸至硬體中，並等待辨識結果，最終將辨識結果(正確率)透過硬體的 BLE 傳輸至 Unity 中，並顯示訓練結果的畫面後，返回 Android Studio app 中進行後續處理。

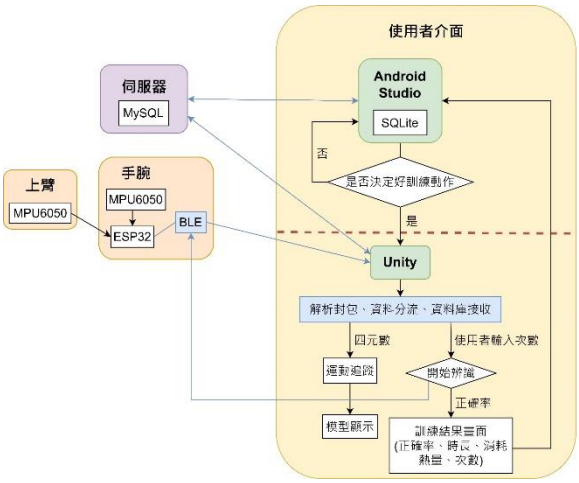

圖 1 系統架構圖

(一).資料庫

在資料庫的使用上，我們利用 MySQL 在伺服器端建立了一個資料庫，也利用了 Android Studio 內建的 SQLite 在本地端建立了一個的資料庫。其中伺服器端資料庫的結構如圖 2 的上半部分所顯示，擁有 Customer、Purchase、Movement、Start 以及 Result 這五個關聯表。Customer 關聯表用來儲存使用者註冊時所填寫的個人資料；Purchase 關聯表用來記錄使用者購買了哪些訓練項目；Movement 關聯表用來存放所有訓練動作的資訊；而 Start 和 Result 關聯表則用來作為 Android Studio 和 Unity 兩個使用者介面的資料傳遞媒介。而本地端的資料庫則如圖一的下半部分所顯示，擁有 Customer、Movement 以及 Record 這三個關聯表。雖然本地端的資料庫看似與伺服器端的資料庫所儲存的內容極為相似，但這兩個資料庫的差別在於伺服器端的資料庫是透過網際網路與我們的 app 做連接，儲存了所有在我們系統中註冊過的使用者的資料；而本地端的資料庫則是存在於使用者手機的內部，只儲存在該手機上登入過我們 app 的使用者的資料，所以後者主要的功用是用來提升使用者操作 app 時的流暢度。

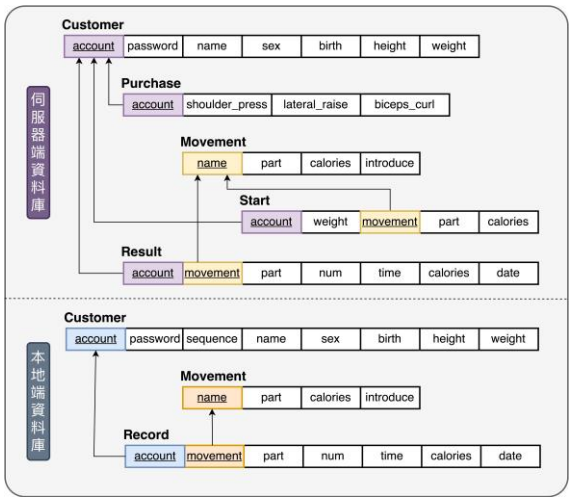

圖 2 資料庫結構圖

(二).使用者介面-Android Studio

在使用者介面的設計上，我們使用了 Android Studio 和 Unity 來進行開發，app 整體的流程如圖 3 上面所畫的，一開始打開 app，使用者會從起始畫面選擇進行註冊或登入，如果選擇註冊的話，完成後會再回到起始畫面，系統需要在進行登入後才能使用後續的操作，那接下來系統的主要功能就會像圖上所畫的紅線的走向，從購買動作開始，依序完成選擇動作、動作介紹、開始訓練、訓練結果及今日紀錄各步驟。而黃色框所標示的是由介面右上角的選單所提供的附加功能，像是歷史紀錄、修改個資及切換帳號的部分。至於藍色框標示的開始訓練和訓練成果則是以 Unity 來進行開發的，其更詳細的系統流程將在下一部分做說明。

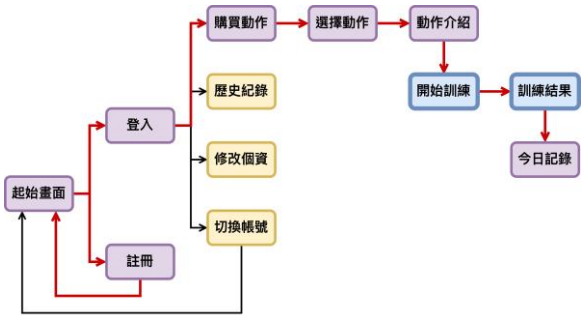

圖 3 使用者介面流程圖

(三).使用者介面-Unity

在 Unity 開發的使用者介面中，有 App 首頁、輸入動作次數、開始訓練、及結算四個畫面，其中 App 首頁包含連接硬體、動作示範影片和校正動作的功能；輸入動作次數界面提供使用者選擇次數；開始訓練界面提供計時、顯示剩餘次數的功能，最後結算界面會顯示使用者訓練資訊，也提供更改動作、再做一次、結束的功能。在 Unity App 中會連接伺服器，以接收及傳送資料，如圖 4 所示。

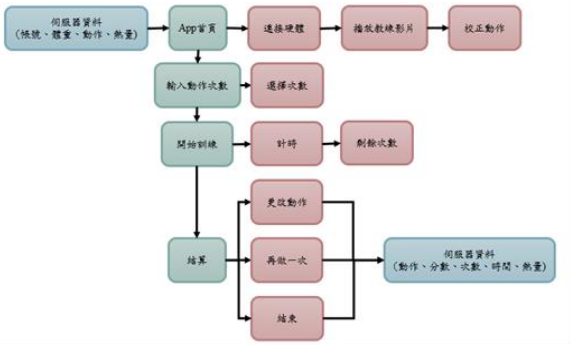

圖 4 Unity 介面流程圖

(四).硬體

本研究之硬體架構圖如圖 5 所示，一組硬體包含一顆 ESP32、兩顆 MPU6050 以及一顆鋰電池。因本系統之設備是利用藍牙功能去進行資料的傳輸，所以需要利用鋰電池來提供電源，而在感測器的方面，兩個 MPU6050 是不一樣的，上面這顆感測器之位址為 0x68，下面這顆感測器之位址為 0x69，兩者差在有在 AD0 腳位多焊了一條黑色的線，因為 MPU6050 的 i2c 位置會是一樣的，如果需要同時接收兩個感測器時，需要透過 AD0 腳位電位，可以改變 i2c 位置，這樣就可以接兩個感測器了。

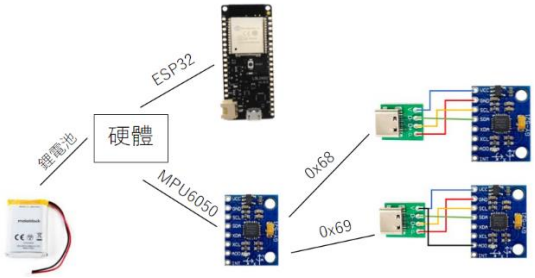

圖 5 硬體架構圖

(五).動作追蹤

本研究用於追蹤手腕和手臂在虛擬

三維座標(Unity)中的運動，會以三維虛擬人型(Unity)呈現手臂姿態。以程式語言 C# 於 Unity 平台上開發，其中使用了 Unity Asset Store 中的模型項目 Fast IK，重建手臂的運動軌跡[1]。以慣性測量單元 MPU6050 之加速度及角速度計算出能夠表示三維空間的旋轉之值(四元數)，放進以 Unity 開發環境 Mixamo 套件所建立之人物模型關節點處上，使其關節能夠旋轉以表示使用者之重量訓練時動作，並於 Unity 所顯示之介面左上角設立重新定位之按鈕，使用者穿戴好裝置後可以按此按鈕進行模型重新定位之矯正，進而開始重量訓練之動作，流程如圖 6 所示。

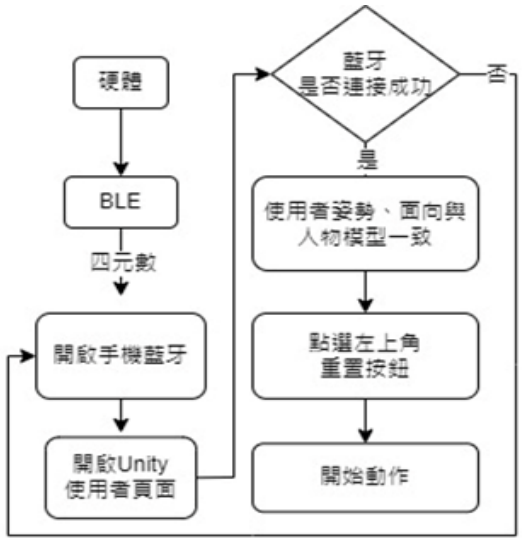

圖 6 動作追蹤流程

(六).姿態辨識

本論文中的辨識模型使用 TensorFlow Lite 框架程式開發，並引用其中的 Keras API 來輔助建設 Sequential 模型[2]，接著將訓練好的模型放置在硬體中等待使用者進行運動來開啟此辨識功能。啟用辨識功能的流程如圖 7 所示，硬體等待接收到 Unity 透過 BLE 傳輸過來的動作名稱、次數後，表示開啟辨識功能，接著判斷使用者是否處於休息狀態(總加速度小於 8)，一旦使用者處於非休息狀態時(總加速度大於等於 8)，開始接收使用者的該次動作資料(250 筆)，並將使用者每一次動作的資料放入模型中進行辨識，最後判斷是否完成使用者當初輸入的運動次數，完成即表示辨識結束。最終計算總正確率，並將辨識結果(總正確率)透過硬體的 BLE 傳至 Unity 來進行後續處理 。

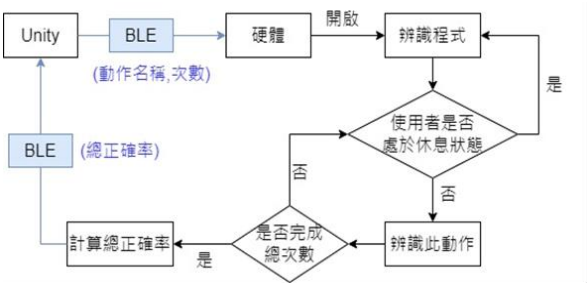

圖 7 姿態辨識流程

三、成果展示

(一).硬體

硬體穿戴示意圖如圖 8 所示，放置手臂上的感測器之位址為 0x68(編號為 1)，放置手腕處的感測器之位址為 0x69(編號為 2)，中間放置 ESP32 與提供電源的鋰電池，以雙頭 Type-C 接口連接，所有孔皆朝右邊放置，接著平視三者，檢查是否呈一直線，才算穿戴完成。

圖 8 硬體穿戴示意圖

(二).動作追蹤

首先使用者穿戴好穿戴式感測器裝置，於手腕及上手臂，之後開啟手機藍牙，連接感測器傳送出來之四元數數據，而後開啟 Unity 手機端之頁面，最後開啟 Unity 手機端之頁面。使用者與人體模型面朝一致，動作一致如圖 9 所示，點

選重制動作，即可開始動作，如圖 10 所示。

圖 9 使用者動作與人物模型一致

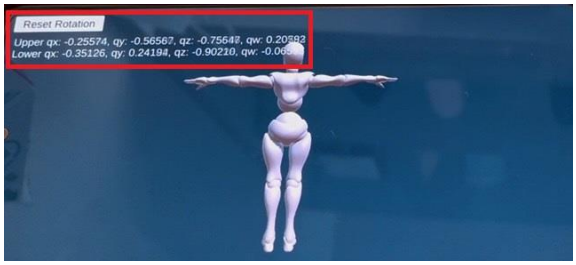

圖 10 Unity 手機端介面

(三).姿態辨識

透過 Arduino 環境以及 Unity 使用者介面中的畫面來展示實作成果，如圖 11、12 所示。

紅色框內容為此次實作中使用者在 app 中選擇的動作及次數，並透過 BLE 傳至硬體，動作為 flex(二頭彎舉)、次數為 3 次。黃色框以及綠色框中的內容為，手腕及手臂感測器分別在此次動作的正確率，黃色框的正確率高是因為測試時使用者做的動作為二頭彎舉，綠色框的正確率較低則是因為使用者在此次的動作不是二頭彎舉。直到使用者做完總次數後，系統會計算藍色框內容中的總正確率並將此數值回傳至 Unity 使用者介面中，如圖 11 所示。

最終 Unity 使用者介面會收到硬體透過BLE傳過去的總正確率，為紅色框的內容，如圖 12 所示，內容為 0.67。

圖 11 姿態辨識 Arduino 實作畫面

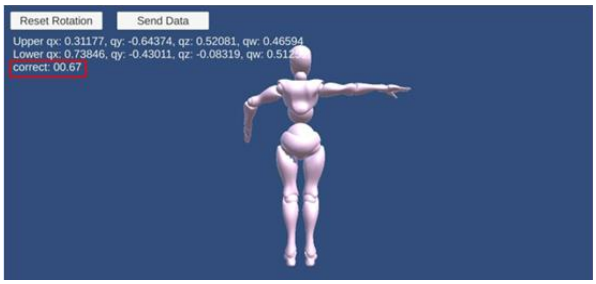

圖 12 姿態辨識 Unity 實作畫面

(四). 正式訓練和結算界面

正式訓練界面會顯示使用者目前訓練的經過時間和剩餘次數，也能透過觀看人體模型來確認動作狀況。結算介面會將使用者的動作名稱、準確度、次數、花費時間及熱量都計算好並顯示在畫面中，如圖 13 所示。

(五). 今日訓練紀錄和歷史紀錄

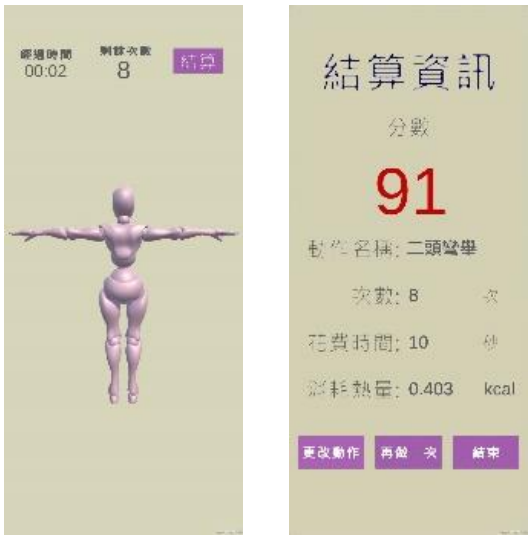

圖 13 手機畫面展示

完成訓練後，今日訓練紀錄畫面中會呈現使用者當天訓練總共花費的時間、消耗的熱量以及各動作分別的訓練狀況，如圖 14 所示。而在歷史紀錄畫面中，使用者能利用畫面中的下拉式選單選擇自己想要查看的月份，得知當月訓練總共花費的時間及總共消耗的熱量，

也能透過畫面下方的圖表了解到各部位的訓練情形，如圖 15 所示。

圖 14 今日訓練紀錄

圖 15 歷史紀錄

四、結論與未來發展

經過本團隊此次實作下來的結果，我們認為可以針對以下的一些問題再進行改進，像是在硬體的部分，我們發現製作出來的硬體重量會太重，導致在做動作時容易滑落，不易於固定在手臂上，因此我們希望之後能以 Nano33 取代 ESP32 和一個 MPU6050，因為 Nano33 同時可以取得六軸的數據還擁有 BLE 的功能，所以就能用以取代的原先上述的兩個硬體，大幅減輕了整體的重量，而在硬體之間的連接也能以電路片的方式取代線，便於使用者穿戴以及進行較大動作時，不會因為線的干擾及重量問題造成讀取資料的不準確。此外，可以將穿戴式裝置之慣性測量單元以擁有時間序列的感測器取代，使用者在動作過程中，若有太快或太慢之行為，可以實時偵查並做出提醒。

在使用者介面的部分，將 Unity 嵌入 Android Studio 可能會在版本和開發環境上遇到較大的問題，導致 Unity 的介面無法順利地被啟用，並且若使用不同的軟體進行開發，在兩個介面之間的資料傳遞方式也會較為複雜，因此，未來可以朝向將整個系統介面都以 Unity 來進行開發的方式，不但能讓使用者在操作上更為流暢，開發時也不需再另外考慮不同軟體之間的兼容問題。

在姿態辨識的部分，可以再試著用更多不同的模型去做訓練，從而找到正確率更高、匹配性更好的模型。

也期許未來可以新增不同的部位及動作，例如下肢的重量訓練。

五、 參考文獻

[1] 林士心，“慣性量測單元於重量訓練之人工智慧演算法開發”，學術論文，聯合大學，2021 年 7 月。

[2] 云水木石，理解 keras 中的 sequential 模型，騰訊云，民國一零八年。