

籃球犯規手勢識別

**Gesture recognition of basketball
referee signals from EMG,
accelerometer
and G-sensor using deep learning**

目錄

一、 摘要.....	1
1. 目的.....	1
2. 進行方法.....	1
3. 手勢選擇.....	2
二、 簡介.....	2
1. 特點.....	2
2. 硬體規格.....	3
(1) 材料	3
(2) 取樣頻率	3
(3) 流程圖.....	3
(a) 左手	3
(b) 右手	4
3. 軟體規格.....	4
(1) Linkit7697 SDK.....	4
(a) MPU6050 驅動.....	4
(b) Linkit7697 IIC接口	4
(c) MQTT	4
(2) Matlab R2018b	4
(3) Python	4
(a) 六軸加速器	4
(b) AI Model	4
(c) k-NN 演算法	4
(d) MQTT 訂閱端.....	5
(e) GUI介面.....	5
三、 理論.....	5
1. EMG訊號處理方法	5
(1) 電壓準位	5
(2) 濾波器.....	5
(3) 短時距傅立葉變換(short-time Fourier transform, STFT)	7
(4) 卷積神經網路(Convolution Neural Network, CNN)	8
(a) Dataset.....	8
(b) Model Structure	8
2. 陀螺儀與加速度計處理方法:.....	8
(1) 加速度計轉換角度:.....	8
(a) x軸至水平面夾角:.....	9
(b) y軸至水平面夾角:.....	9
(c) z軸至水平面夾角:.....	10
(2) 特徵值.....	11
(3) Overlapping 取值	11
(4) k-近鄰演算法(k-Nearest Neighbors, k-NN)	11
(a) k-NN algorithm.....	11
(b) k值的影響	12

四、 實驗.....	13
1. 硬體製作	13
(1) 硬體線路圖.....	13
(2) 實際裝置圖.....	13
2. 軟體製作	14
(1) 收集資料	14
(a) GUI介面.....	14
(b) 流程圖.....	14
(2) 辨識手勢動作	15
(a) GUI介面.....	15
(b) 流程圖.....	15
3. 實測結果	16
(1) 用CNN辨識的準確率	16
(2) 用k-NN辨識的準確率	16
4. 整體流程圖	16
五、 介面使用.....	16
1. 收集資料之GUI介面使用	16
2. 辨識手勢動作之GUI介面	17
六、 結論與未來發展.....	17
1. 總結.....	17
2. 未來展望	17
七、 參考資料.....	17
1. 論文.....	17

一、摘要

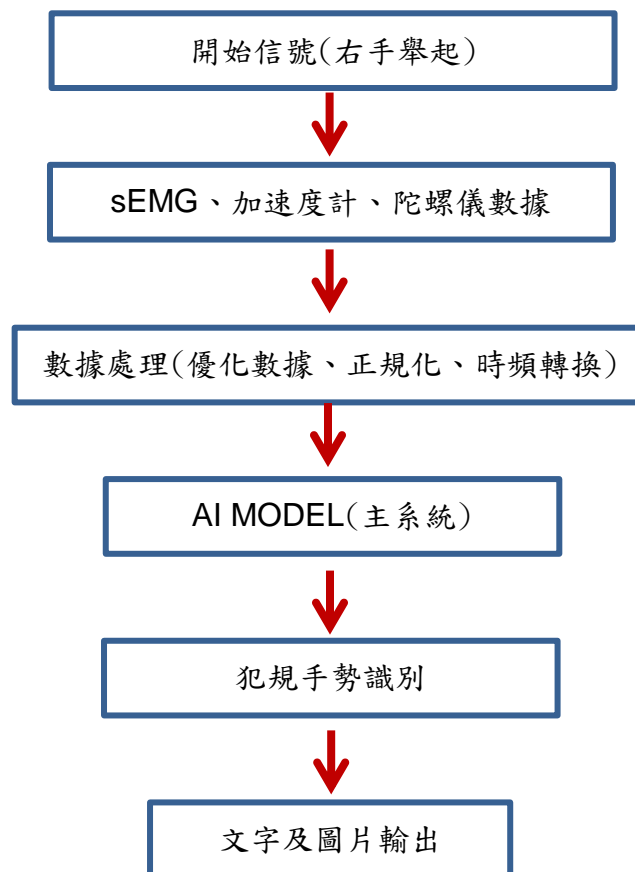
1. 目的

我們希望做出一個護肘形式的物品，上面會設置EMG感測器，加速度計與陀螺儀，不過在實測後，發現只利用加速度計與陀螺儀就能得到較好的精確度並且更為輕便，所以會更推薦以這兩個為主。

接著去收集使用者的運動數據，再將此數據經過優化處理後，送進訓練好的AI Model中，利用AI幫助我們判斷出裁判現在正在比的犯規手勢為何。


2. 進行方法

- (1) 將 sEMG Sensor、加速度計與陀螺儀設置在護肘上，使用者穿上護肘，收集運動訊號。
- (2) 利用無線方式，將訊號傳遞至電腦上。
- (3) 將收集到的數據經過優化處理，藉由 NN 來訓練 AI MODEL。
- (4) AI MODEL 當作後端，感測器所收集到的數據經由優化處理後，送入 MODEL 進行手勢判別，將使用者所做的犯規手勢顯示到螢幕上，讓觀眾得知。



3. 手勢選擇

我們從網路上選擇了一些籃球比賽中發生頻率比較高的違例及犯規手勢，從比較常見的下手，總共選擇了10種出來(表一.(a)):

				
技術犯規	非法用手	帶球撞人	對手非法接觸	拉人
				
非法掩護	推人	用手推擋	帶球走步	兩次運球

表一.(a)

二、 簡介

1. 特點

大多數在做手勢辨識時，都會利用影像辨識的方法去做，可是我們則是選擇利用肌電訊號、陀螺儀與加速度計。因為影像資料量大，會需要龐大計算量，可能無法即時分辨手勢；另外辨識能力可能受環境光源影響，且實際比賽場合，裁判會跑動，若採用影像處理，還需攝影機跟著移動，故影像處理較不實用。

考量到計算量與移動性，選擇這種方法可以使我們在做動作時，不會被鏡頭所能照到的範圍侷限，這樣更適合應用在籃球賽場上這些常需要跟著球員跑的裁判，若是利用影像辨識，那還要跑到定點會造成許多不便。

最後在實測過程中，雖然肌電訊號較不受個體差異性影響，但因為 sensor 較多，移動起來較為不便；而陀螺儀與加速度計，各軸只須達到 50Hz，資料量比肌電訊號還要更少，且準確率也在一定標準上，使用起來也輕便更多，因此我們推薦只用陀螺儀與加速度計的資料來辨識。表二.(a)為影像辨識、肌電訊號、陀螺儀與加速度計，三者資料量比較。

影像處理	肌電訊號	陀螺儀與加速度計
影格為600*600影像 影格速率30 FPS(Frame per Second)	左右手各四個感測器 總共8個channels 取樣頻率1000Hz	各有x,y,z三軸 左右手各一個 取樣頻率50Hz
每秒產生資料量 $600*600*30 = 10,800,000$ 筆	每秒產生資料量: $1000*8 = 8000$ 筆	每秒產生資料量 $50*3*2 = 300$ 筆

表二.(a)

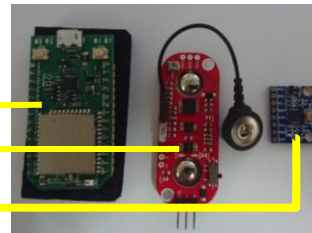
2. 硬體規格

(1) 材料

Linkit 7697 微控制器

MYO Ware 肌電感測器

MPU6050 六軸加速器



(2) 取樣頻率

MYO Ware 肌電感測器 – 1000 Hz

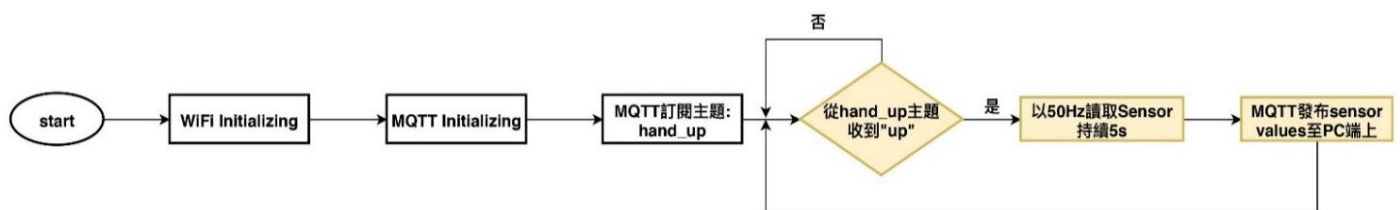
MPU6050 六軸加速器 – 每軸各 50 HZ

(3) 流程圖

開機後，由於我們把右手舉起當作起始信號，因此右手的 Linkit7697 會去判別右手是否舉起，當右手舉起後則會發布一 hand_up 訊息到 MQTT，接著 MQTT 會再回傳給左、右手，這時左、右手就會向 Sensors 收資料，共收 5 秒的時間。

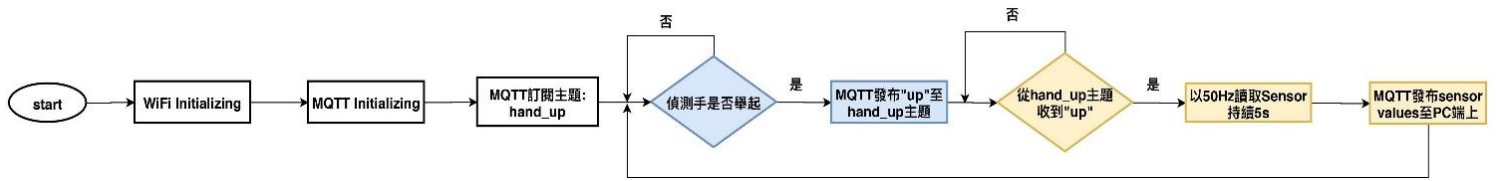
資料收集好後，左、右手各有 3000 Bytes(6 軸*250 筆*每筆 2 Bytes)的資料要傳，一次傳會導致通道阻塞，因此我們每次傳 50 Bytes，共傳 60 次。下面分為左、右手兩個流程圖(圖二.(a)、圖二.(b))。

(a) 左手



圖二.(a)

(b) 右手



圖二.(b)

3. 軟體規格

(1) Linkit7697 SDK

透過 Linkit7697 SDK 並使用 KeilC 來開發 Linkit 7697，其包含：

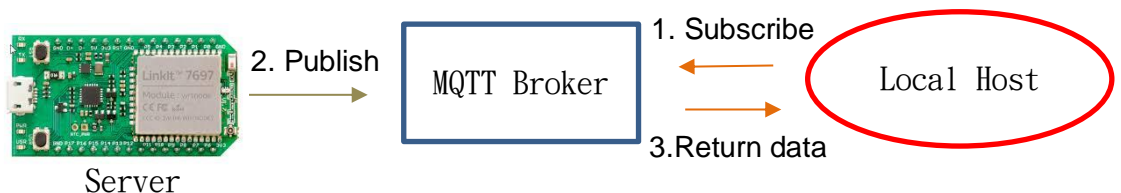
(a) MPU6050 驅動

(b) Linkit7697 IIC 接口

IIC 接口與 MPU6050 連接，並進行 MPU6050 的初始化，設定電源模式、資料提取模式、以及測量精度，最後讀取加速度計及陀螺儀之信號並換算。

(c) MQTT

基於發布 (Publish)/訂閱 (Subscribe) 式的消息協議，上述兩者間有 Broker 當作中繼站，所以不需要知道彼此 IP 位置；我們利用它進行 WIFI 的傳輸，它工作在 TCP/IP 協議上，是為硬體性能低下的遠程設備以及網絡狀況糟糕的情況下而設計的發布/訂閱型消息協議。圖二.(c)為示意圖。



圖二.(c)

(2) Matlab R2018b

主要對 EMG 訊號進行處理，利用 Matlab 對 EMG 訊號做正規化，且設計一橢圓帶通濾波器，將正規化後的訊號進行濾波，並做 Short-time Fourier transform。

(3) Python

(a) 六軸加速器

將六軸所傳輸資料做正規化，並取特徵值。

(b) AI Model

利用 Python 內部套件，建立 CNN Model 作為 EMG 訊號判別動作所用。以及六軸之 KNN 分類演算法。

(c) k-NN 演算法

六軸所收集到的資料，利用 KNN 演算法去做動作分類識別。

(d) MQTT 訂閱端

用 Python 做 MQTT 的訂閱端，接收 WIFI 所傳遞資料。

(e) GUI 介面

設計二個簡易 GUI 介面，分別是收集資料的介面，和實際要辨識動作的介面。

三、理論

本專題收取了 EMG 以及六軸兩種資料，分別對兩種資料做不同的處理，以下對這兩種資料所做的處理方法與理論作說明：

1. EMG 訊號處理方法

(1) 電壓準位

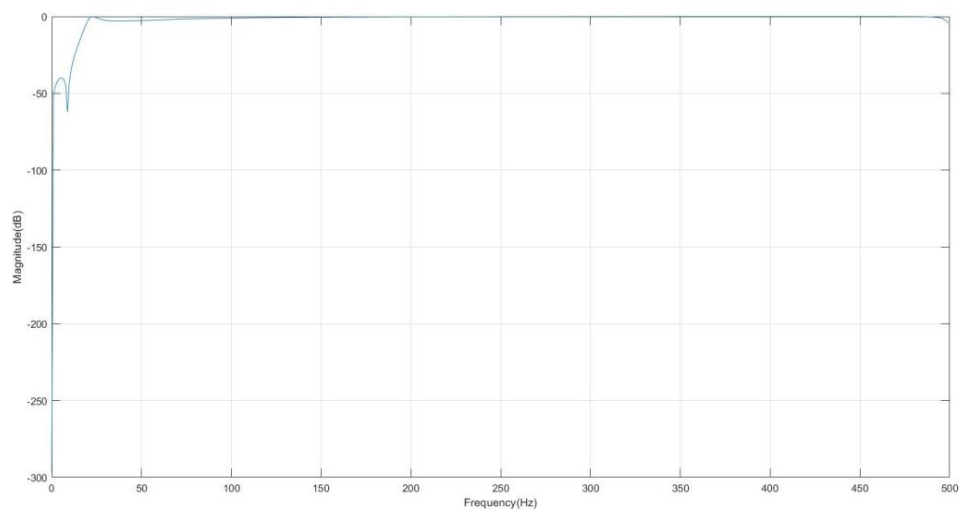
由於 MYOWARE 所量測的肌電訊號準位在 1.25V，先將它準位移動到 0V，這時電壓範圍在 $\pm 1.25V$ 間，再把它們正規化到 $\pm 1V$ 間(方程式三.(a))。

$$Data_{Normalize} = \frac{(Data_{Raw} - 1.25)}{1.25} \quad \text{-方程式三.(a)}$$

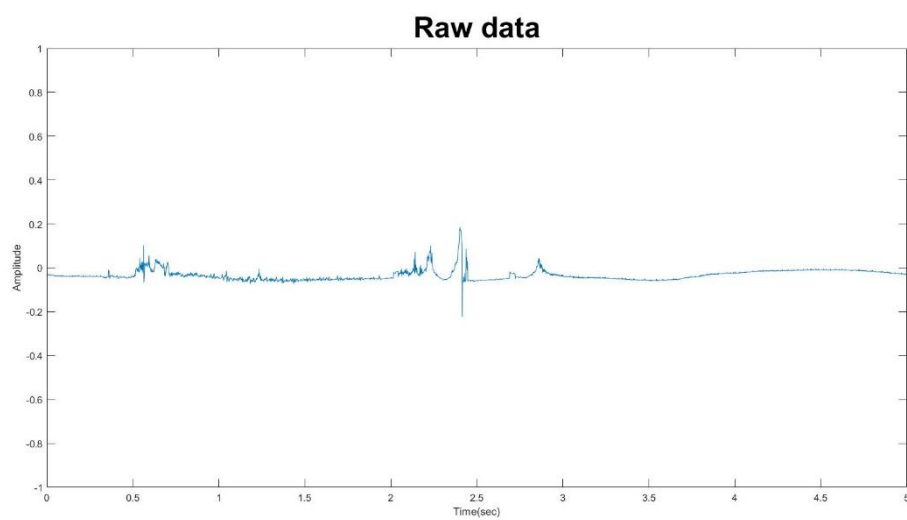
(2) 濾波器

肌電訊號容易因為感測器線路的擺動，而產生許多的雜訊，所以需要濾波器將不必要雜訊濾掉。而濾波器又可分為有限脈衝響應濾波器(FIR)以及無限脈衝響應濾波器(IIR)，兩者在相同性能下，無限脈衝響應濾波器(IIR)的階數又比有限脈衝響應濾波器(FIR)低很多。因此本專題中我們選用無限脈衝響應濾波器濾波器(IIR)。

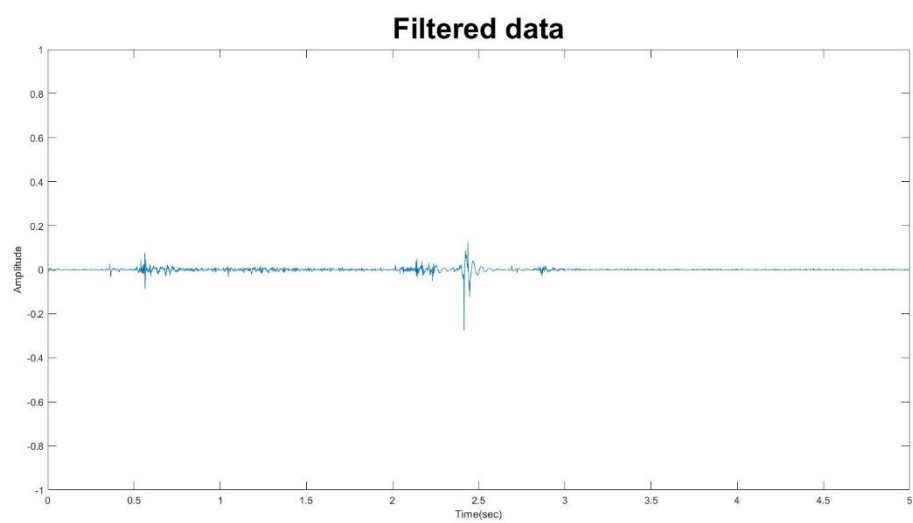
又無限脈衝響應濾波器濾波器(IIR)中我們選用橢圓濾波器，因為和其它型濾波器相比，它能有較窄的過度帶寬，雖然會有些許波動，但在可接受的範圍內。而從論文中我們可以得知肌電訊號大都分布在 20~500HZ，因此利用 matlab 模擬設計一個濾波器通帶為 20~500HZ，其波特圖如圖三.(a)。圖三.(b)為還未濾波的 Raw data;圖三.(c)為經過濾波後的 Filtered data。



圖三.(a)



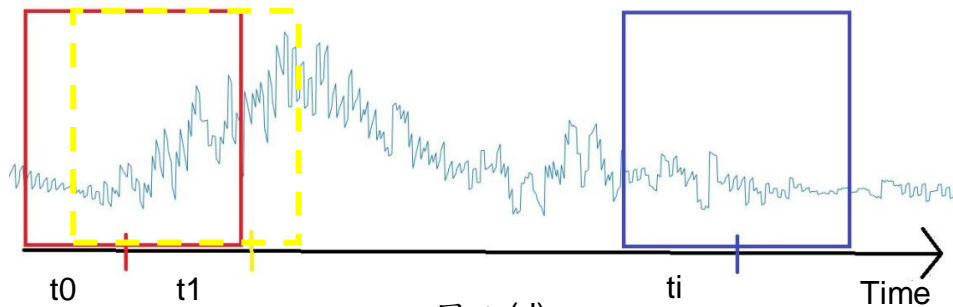
圖三.(b)



圖三.(c)

(3) 短時距傅立葉變換(short-time Fourier transform, STFT)

肌電訊號是一個會隨時間而改變頻率的訊號，若我們用一般的Fast Fourier Transform(FFT)，那就只能分析一維的訊號分布，無法看到時域和頻域的關係。因此本專題採用STFT，可以同時分析頻域及時域的信號分布，更有助於找到其特徵性。以下簡易地介紹STFT做法：



圖三.(d)

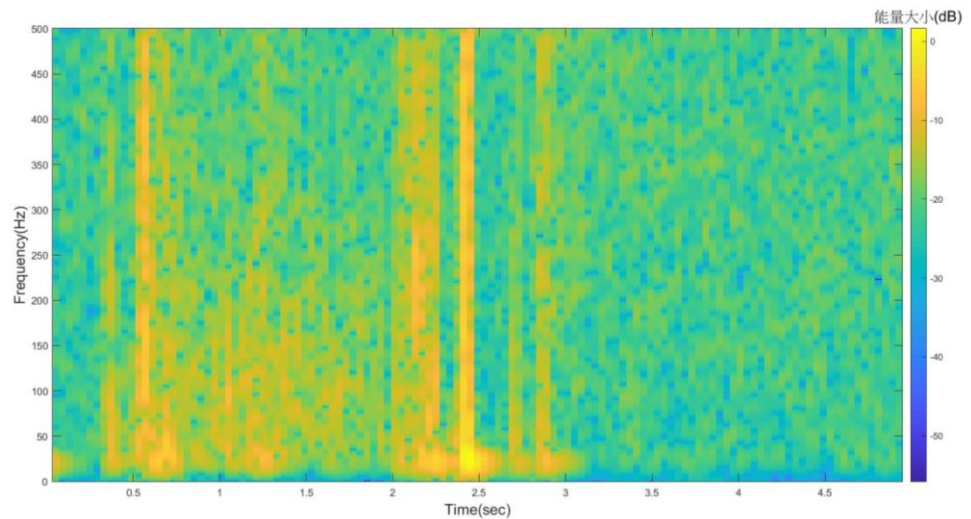
圖三.(d)中方框為選定好的窗函數， t_0 為窗的中央值，將窗內所擷取的信號與窗函數相乘，再將這一長度的資料去做FFT的轉換，得出結果大約為 t_0 這時間點的頻域特性。而窗函數長短並不是隨便亂設定，長窗提高頻域解析度，短窗則會提升時域解析度。

接著將窗函數向右移，計算下一個時間點的頻域特性，但在為了確保頻域特性的基礎上也要提高時域的特性，前後窗函數會重疊一部分，兩個時刻的間距也會比較短，不過也不能重疊過多，否則將會拉低電腦運算效率。最後窗函數一直右移到整個信號計算完結束。

本專題在STFT轉換上，窗函數選用hanning window、window size=100、重疊部分(overlapping)=60、取樣頻率=1000Hz。

輸出結果為一 501×123 (Frequency*Time)之矩陣(501表示從0~500Hz分成501等分；123是將時間5秒分成123等分)。而由於左右手共有8個肌電感測器，因此最後完整一筆資料為 $501 \times 123 \times 8$ (Frequency*Time*Channel)的格式。

圖三.(e)為將濾波完的肌電訊號，進行STFT的轉換後結果，不過在訓練Model時並不是直接將圖片送進去訓練，這是因為讀取圖片的時間比較慢，所以我們直接將上述所轉換出來的矩陣形式的數值，送進去訓練。



圖三.(e)

(4) 卷積神經網路(Convolution Neural Network, CNN)

將肌電訊號做 STFT 轉換出來的二維矩陣，利用 CNN Model 進行訓練。

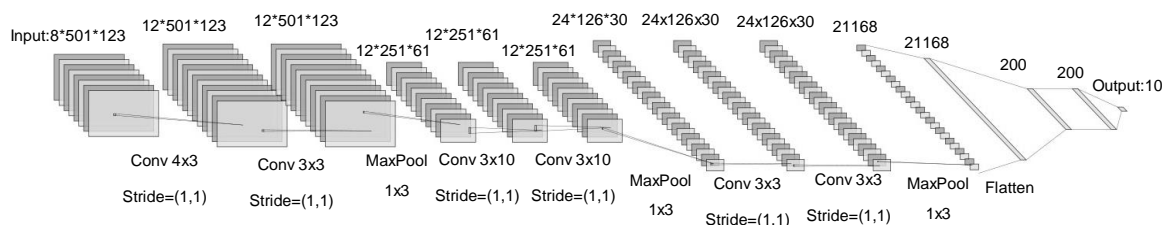
(a) Dataset

Training: 215 pieces of data

Testing: 93 pieces of data

Categories: 10

(b) Model Structure

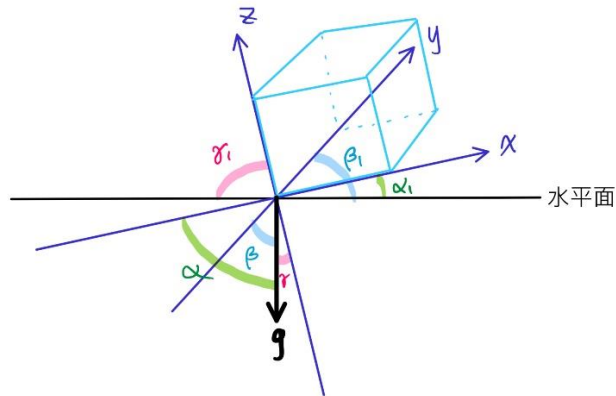


圖三.(f)

2. 陀螺儀與加速度計處理方法:

(1) 加速度計轉換角度:

在三軸加速度上，可測得重力 g 對 x 、 y 、 z 軸造成的力，分別以 A_x 、 A_y 、 A_z 表示，而我們需要的是三軸至水平面的傾斜角 α_1 、 β_1 、 γ_1 ，計算過程如下:



圖三.(g)

已知 A_x 、 A_y 、 A_z ，求 α_1 、 β_1 、 γ_1 ，(i.e. 用 A_x 、 A_y 、 A_z 來表示角度) A_x 、 A_y 、 A_z 分別為重力 g 在 x 、 y 、 z 上的分量，換言之， g 可由 A_x 、 A_y 、 A_z 三個向量組成，其大小可表示為：

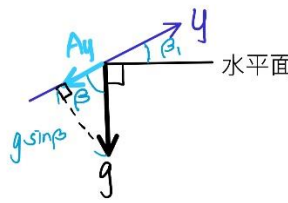
$$\Rightarrow g^2 = A_x^2 + A_y^2 + A_z^2 \dots\dots\dots (1)$$

(a) x 軸至水平面夾角：

	$\alpha + \alpha_1 + \frac{\pi}{2} = \pi$ $\Rightarrow \alpha = \frac{\pi}{2} - \alpha_1 \dots\dots\dots$ <p>(2)</p>
$A_x = g \cos \alpha = g \sin \alpha_1 \dots\dots\dots \text{由(2)得}$ $\Rightarrow \sin \alpha_1 = \frac{A_x}{g}$	
$g \sin \alpha = g \cos \alpha_1 \dots\dots\dots \text{由(2)得}$ $g^2 = A_x^2 + (g \cos \alpha_1)^2 \dots\dots\dots \text{畢氏定理}$ $\Rightarrow \cos \alpha_1 = \frac{\sqrt{g^2 - A_x^2}}{g} = \frac{\sqrt{A_y^2 + A_z^2}}{g} \dots\dots\dots \text{由(1)得}$	
$\Rightarrow \tan \alpha_1 = \frac{\sin \alpha_1}{\cos \alpha_1} = \frac{A_x}{\sqrt{A_y^2 + A_z^2}}$ $\Rightarrow \alpha_1 = \tan^{-1} \left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}} \right)$	

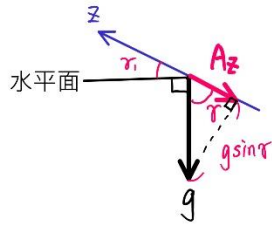
表三.(a)

(b) y 軸至水平面夾角：

	$\beta + \beta_1 + \frac{\pi}{2} = \pi$ $\Rightarrow \beta = \frac{\pi}{2} -$ $\beta_1 \dots\dots\dots(3)$
$Ay = g \cos \beta = g \sin \beta_1 \dots\dots\dots\text{由(3)得}$ $\Rightarrow \sin \beta_1 = \frac{Ay}{g}$	
$g \sin \beta = g \cos \beta_1 \dots\dots\dots\text{由(3)得}$ $g^2 = Ay^2 + (g \cos \beta_1)^2 \dots\dots\dots\text{畢氏定理}$ $\Rightarrow \cos \beta_1 = \frac{\sqrt{g^2 - Ay^2}}{g} = \frac{\sqrt{Ax^2 + Az^2}}{g} \dots\dots\dots\text{由(1)得}$	
$\Rightarrow \tan \beta_1 = \frac{\sin \beta_1}{\cos \beta_1} = \frac{Ay}{\sqrt{Ax^2 + Az^2}}$ $\Rightarrow \beta_1 = \tan^{-1}\left(\frac{Ay}{\sqrt{Ax^2 + Az^2}}\right)$	

表三.(b)

(c) z 軸至水平面夾角:

	$\gamma + \gamma_1 + \frac{\pi}{2} = \pi$ $\Rightarrow \gamma = \frac{\pi}{2} -$ $\gamma_1 \dots\dots\dots(4)$
$Az = g \cos \gamma = g \sin \gamma_1 \dots\dots\dots\text{由(4)得}$ $\Rightarrow \sin \gamma_1 = \frac{Az}{g}$	
$g \sin \gamma = g \cos \gamma_1 \dots\dots\dots\text{由(4)得}$ $g^2 = Az^2 + (g \cos \gamma_1)^2 \dots\dots\dots\text{畢氏定理}$ $\Rightarrow \cos \gamma_1 = \frac{\sqrt{g^2 - Az^2}}{g} = \frac{\sqrt{Ax^2 + Ay^2}}{g} \dots\dots\dots\text{由(1)得}$	
$\Rightarrow \tan \gamma_1 = \frac{\sin \gamma_1}{\cos \gamma_1} = \frac{Az}{\sqrt{Ax^2 + Ay^2}}$	

$$\Rightarrow \gamma_1 = \tan^{-1}\left(\frac{Az}{\sqrt{Ax^2 + Ay^2}}\right)$$

表三.(c)

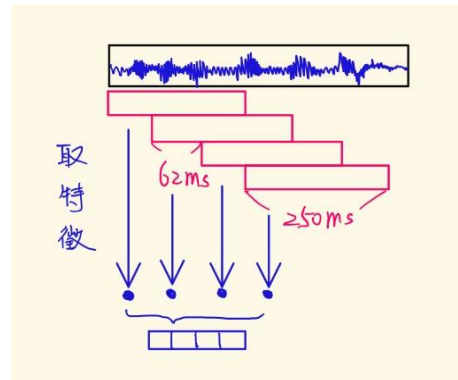
(2) 特徵值

在陀螺儀與加速度計的前處理上，我們會另外取 Mean, ZC, SSC, WL, RMS 這五個特徵值，來增加準確率；又經過列舉的排列組合後，發現由 Mean, WL, RMS 這三個特徵值組合，其準確度最高。

1. Mean = $\frac{1}{N} \sum_{n=1}^N x_n$
2. Zero Crossing(ZC) = $\sum_{n=1}^N$ 訊號由正變負的次數
3. Slope sign changes(SSC) = $\sum_{n=1}^N$ 斜率正變負的次數(即極大值發生次數)
4. Waveform length(WL) = $\sum_{i=1}^{N-1} |x_{i+1} - x_i|$
5. Root Mean Square(RMS) = $\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$

(3) Overlapping 取值

可使資料量減少，並大幅減少 KNN 的計算量，加快辨識速度，且能提升準確度，本專題參考論文，每移動 62ms，抓取 250ms window length，如圖三.(h)。

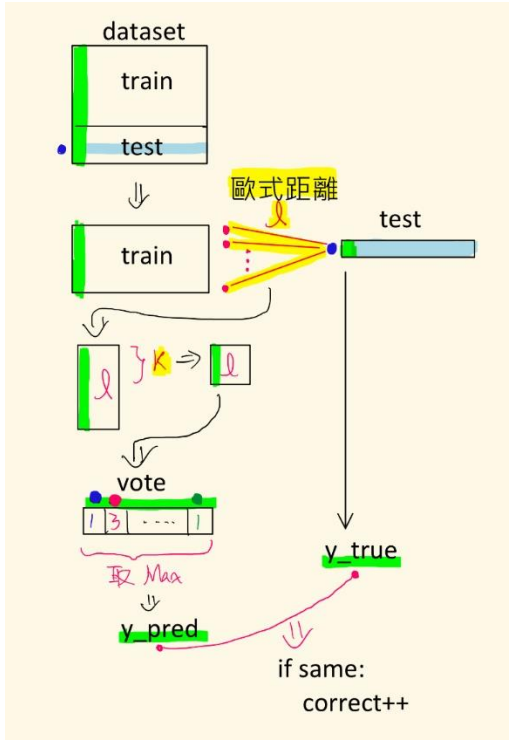


圖三.(h)

(4) k-近鄰演算法(k-Nearest Neighbors, k-NN)

(a) k-NN algorithm

計算該筆 test data 與全部 train data 的歐式距離(表三.(d))，之後取出距離最近的前 k 個，再用投票的方式決定其類別。

k-NN 示意圖	歐式距離
 <p>註: 綠色螢光筆代表該動作的類別</p>	<p>定義: n維空間中，兩點之間的真實距離。</p> <ul style="list-style-type: none"> ● 二維空間: $\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ $\Rightarrow \rho$ 為點 (x_2, y_2) 到點 (x_1, y_1) 的距離 ● 三維空間: $\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$ $\Rightarrow \rho$ 為點 (x_2, y_2, z_2) 到點 (x_1, y_1, z_1) 的距離 ● n 維空間: 令 $U=(u_1, u_2, \dots, u_n)$, $V=(v_1, v_2, \dots, v_n)$ $\Rightarrow \rho = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}$, 為向量 U 到向量 V 的距離

表三.(d)

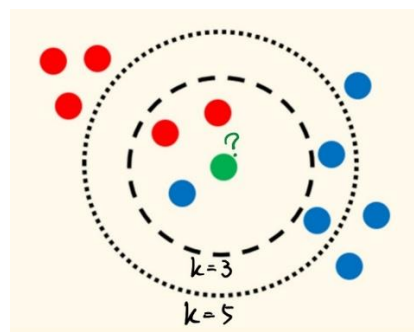
(b) k 值的影響

如圖三.(h)，綠點代表 test data

當 $k=3$ (內圈)，發現紅點 2 個，藍點 1 個，故預測類別為紅點。

當 $k=5$ (外圈)，發現紅點 2 個，藍點 3 個，故預測類別為藍點。

由上述可知，k 值的選擇，將會影響預測的結果。



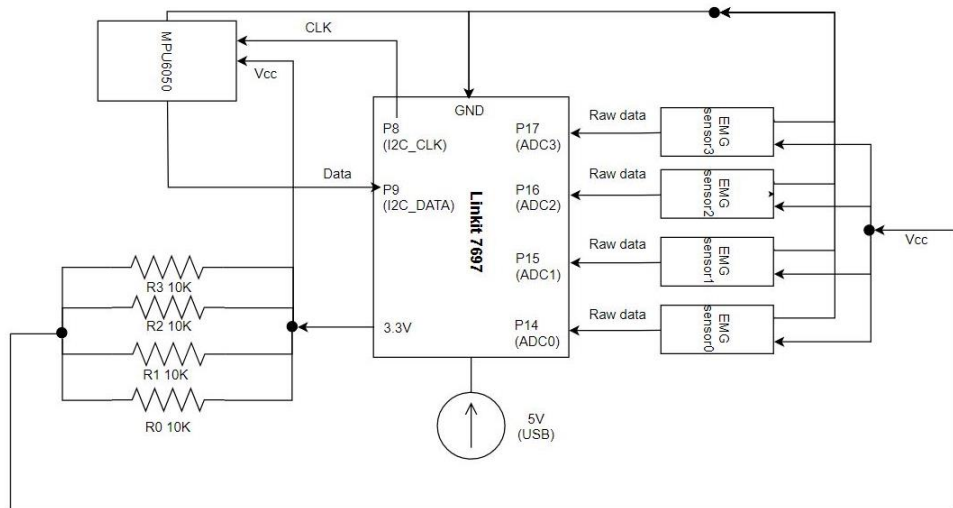
圖三.(h) 12

四、實驗

1. 硬體製作

(1) 硬體線路圖

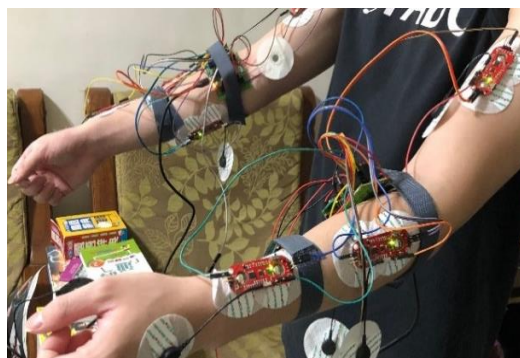
Linkit7697 中，4 個 ADC 腳位分別接 4 個 EMG Sensor(MYO Ware)，由於 EMG 輸入電壓設定 2.5V，因此接電阻分壓；另外 I2C 兩個腳位，分別為 CLK 和 Data 和 Linkit7697 相接，而 MPU6050 供電電壓需在 3~5V 間，所以不需分壓(圖四.(a))。



圖四.(a)

(2) 實際裝置圖

原先裝置有 EMG、陀螺儀與加速度計(圖四.(b))，後來發現只採用陀螺儀與加速度計準確率不降，且更輕便(圖四.(c))。



圖四.(b)含EMG



圖四.(c)不含 EMG

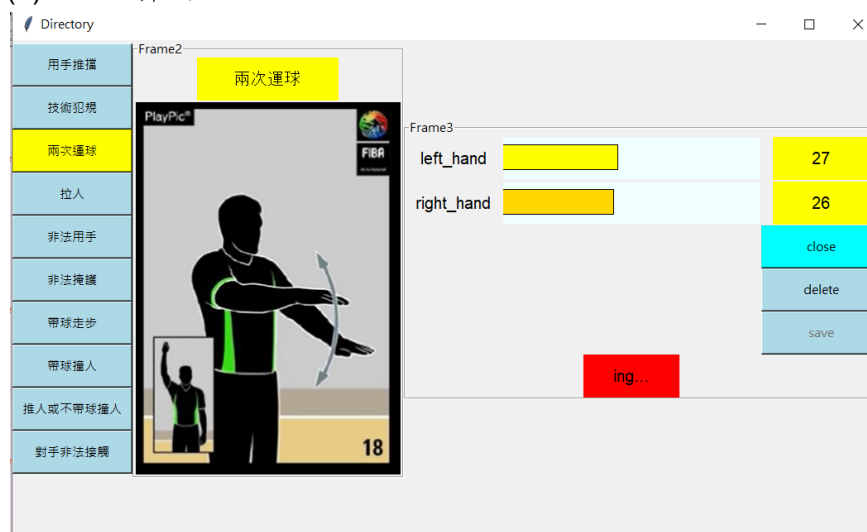
2. 軟體製作

在軟體整合上，我們分別設計了兩個 GUI 介面，一個用來收集資料，另一個則是用來實際辨識手勢動作。

而在最後的整合中，我們只有針對陀螺儀和加速度計的資料做收集，這是因為我們前面部分有提到，這兩項感測器的資料量，相對於肌電訊號少很多，裝置上也更輕便，且準確率跟加了肌電相比並未降低。

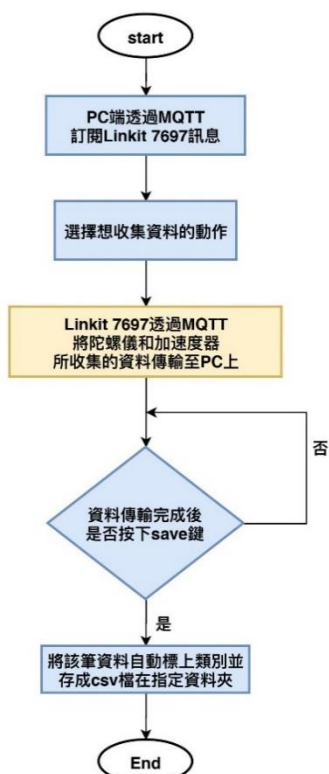
(1) 收集資料

(a) GUI 介面



圖四.(d)

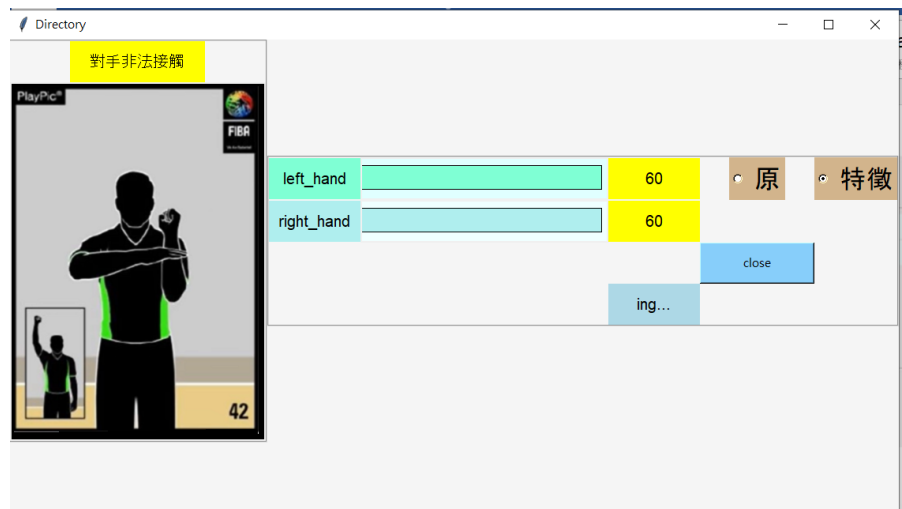
(b) 流程圖



圖四.(e) 14

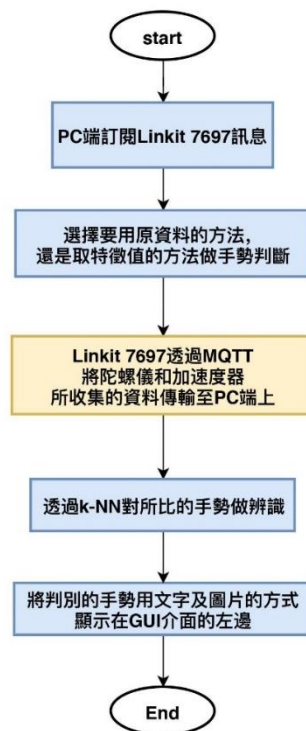
(2) 辨識手勢動作

(a) GUI 介面



圖四.(f)

(b) 流程圖



圖四.(g)

3. 實測結果

(1) 用 CNN 辨識的準確率

EMG 8channels	EMG 6channels	EMG 4channels	EMG 2channels
92%	86%	80%	62%

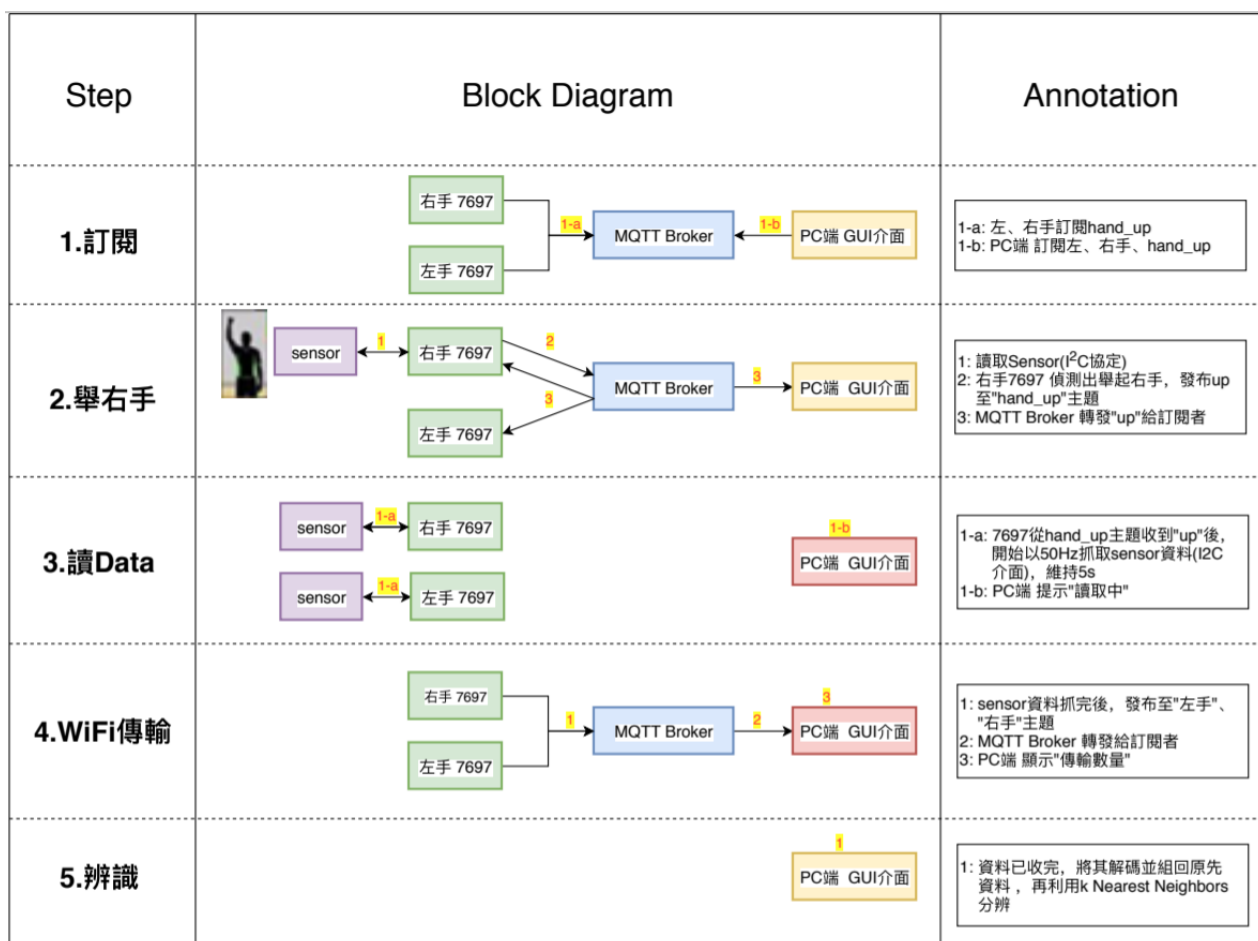
表四.(a)

(2) 用 k-NN 辨識的準確率

ACC	GYO	EMG(未加特徵)	EMG(加特徵)
93%~95%	80%~85%	33%~41%	53%~69%

表四.(b)

4. 整體流程圖



圖四.(h)

五、介面使用

1. 收集資料之 GUI 介面使用

(1) PC 端向 Linkit7697 訂閱訊息

- (2) 選擇想收集的動作(參考「四-2 軟體製作」)
 - (3) 開始做動作，資料會自動透過 WiFi 傳送，GUI 有進度條顯示
 - (4) 當進度條跑完後，按下 Save 鍵便可將這次所收集資料存成 csv 檔儲存在指定路徑
 - (5) 若想再繼續收集新的資料，便再重新執行一次(2)步驟
2. 辨識手勢動作之 GUI 介面
 - (1) PC 端向 Linkit7697 訂閱訊息
 - (2) 選擇要用原資料還是取特徵值的方式來辨識動作(建議用特徵值，準確率較高)。(參考「四-2 軟體製作」)
 - (3) 開始做動作，資料會自動透過 WiFi 傳送，GUI 有進度條顯示
 - (4) 進度條跑完後，會將辨識之動作顯示在左邊方框中
 - (5) 重複步驟(3)，即可再辨識新的動作。

六、結論與未來發展

1. 總結

一開始我們是先用有線傳輸來收集資料，將收集完的資料進行前處理後，盡量的取出特徵值，接著將這些經過處理的數據，送進 Model 進行訓練。

EMG 訊號，若是用 8 個 Channels 所收集到的訊號進行訓練，可達到最高 92%，且 EMG 訊號較能適應個體的差異，就算是不同人比同樣的動作，準確率也不會差很多，但是在便利性上，因為 sensor 較多，比動作時會較不方便。

而陀螺儀及加速度計的訊號，若是用取特徵值的方式，也能達到 90%左右的準確率，且資料量少，不過在用這類資料做 inference 時，就必須要先針對做動作的人建立專屬他的 Model，這樣就能達到較高準確率。

2. 未來展望

雖然本專案是判別籃球犯規手勢，但只要收集其他運動的手勢資料，訓練出新的 Model，就能廣泛地運用在其他運動賽事上，如此觀眾更能即時了解比賽的進行狀況。

甚至也未必只侷限在運動賽事，其餘手勢，像是盲胞們比的手語，目前大多是用影像的方式識別；而我們未來也可以收集這些手語的肌電訊號、六軸出來分析，讓盲胞們不一定只能侷限在鏡頭前，在各個場所都能夠適用。

七、參考資料

1. 論文

- [1] 葉明杰(2004)。《手臂運動之肌電訊號特性分析與應用》。國立交通大學研究所論文，未出版，新竹市。
- [2] 蔡政龍 楊谷洋(2005)《肌電圖強度與速度分析於機器手臂控制之應用》。國立交通大學研究所論文，未出版，新竹市。
- [3] 吳苑娟 楊谷洋(2010)《肌電訊號的處理、判讀與回授應用》。國立清華大學研究所論文，未出版，新竹市。
- [4] G. A. Garcia,AR. Nishitani,AR. Okuno,andAK. Akazawa,
“Independent Component Analysis as Preprocessing Tool for
Decomposition of Surface Electrode-Array Electromyogram,” 4th
International Symposium on Independent Component Analysis
and Blind Signal Separation,pp.191-196(2003)
- [5] Geng, W.; Du, Y.; Jin, W.; Wei, W.; Hu, Y.; Li, J. “Gesture
recognition by instantaneous surface EMG images,”Sci. Rep.
(2016)
- [6] Du, Y.; Jin, W.; We, W.; Hu, Y.; Geng, W. “Surface EMG-Based
Inter-Session Gesture Recognition Enhanced by Deep Domain
Adaptation,” Sensors 17(3), 458(2017)
- [7] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre
Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre
Glette, François Laviolette, Benoit Gosselin,“Deep Learning for
Electromyographic Hand Gesture Signal Classification Using
Transfer Learning,”