**淡江大學資工系**

**111學年度專題成果**

**居家健身姿勢偵測**

**組員**

資工三A\_周騏軍409416541

資工三A\_劉柏辰409410197

資工三A\_鐘金文409416160

資工三A\_丁麗璠409415071

資工三A\_鄧佳朋409411039

**目錄**

[**研究背景** - 1 -](#_Toc131887557)

[**研究動機** - 3 -](#_Toc131887558)

[**研究目的** - 3 -](#_Toc131887559)

[**預期成果** - 3 -](#_Toc131887560)

[**理論探討** - 4 -](#_Toc131887561)

[**設計過程** - 10 -](#_Toc131887562)

[**實際成效** - 12 -](#_Toc131887563)

[**優缺點** - 20 -](#_Toc131887564)

[**未來展望與期許** - 20 -](#_Toc131887565)

[**參考資料** - 21 -](#_Toc131887566)

# **研究背景**

任何將心率提高到靜息水平以上的體育活動都可以歸類為健身。健身是一種有組織的、重複的、有計劃的身體活動，以增強或維持身體健康為最終或中間目標 (Caspersen et al., 1985, 128) 。這對於保持身心健康至關重要。自 Covid-19 爆發以來，人們被限制不能長時間外出。對許多人的心理健康產生了影響 (Kaur et al., 2020)。以家庭為基礎的運動或居家健身可以成為人們保持健康 (包括心理健康和身體健康) 的眾多解決方案之一。 在家健身是另一種很好的調整和應對情況突然變化的活動。受疫情影響，在家健身成為很多人的習慣。 即使是現在，隨著疫情大流行逐漸消退，許多人更喜歡在家健身而不是去健身房。

一般健身可以分為使用設備的和不使用設備的。有多種類型的無設備健身，例如伏地挺身、仰臥起坐、平板撐、開合跳等。保持正確的姿勢運動很重要。健身不當會產生不良結果，甚至可能導致受傷。良好的姿勢可最大程度地減少肌肉拉傷和損傷，並促進身體的高效運作 (Rellinger，n.d.) 。為了安全有效地健身，良好的姿勢是必不可少的，它是健身訓練計劃的重要組成部分。這使得有必要讓一名教練在場監督健身並改善人的姿勢，特別是如果他們是初學者或業餘愛好者。然而，只有一些人有空餘時間、機會和特權與教練一起健身。使用專注於電腦視覺和圖像處理技術程式的人工智慧對於那些想要在沒有教練或其他專家指導的情況下自行健身的人來說可能是有益的。

人工智慧 (AI) 是一種可以利用自身智慧解決複雜問題的人造實體 (Singh & Haju, 2022, 1210)。電腦視覺和圖像處理是人工智慧的一個子類別，結合模式識別方法從圖片、影片和其他視覺輸入中提取信息(Computer Vision and Image Processing Specialization, n.d.)。作為機器學習的一個子集，該項目更注重 19人體姿態估計(HTE)的使用，也可以歸類為電腦視覺的一部分。此項目的目標是幫助人們以正確的姿勢進行深蹲、抬腿、伏地挺身等運動，並在不受傷的情況下取得最大的效果。因此，姿勢偵測對於解決人類和活動識別問題的挑戰至關重要 (Kanase et al., 2021, 1)。將利用 MediaPipe BlazePose標點模型 (Grishchenko，2020) 中提供的 33 個人體關節點。

# **研究動機**

居家健身已被證明對健康有相當大的積極影響，但如果做得不正確，它們也可能存在風險。這可能是訓練不足或不良習慣的結果。在進行健身運動時可能難以監控和調整姿勢，尤其是對於沒有私人教練的業餘愛好者。透過普遍地觀察發現，許多人仍然難以在家中透過健身來有效地鍛煉身體，也無法保持正確的姿勢。這可能會導致效果不佳、身體不適，甚至造成嚴重的運動傷害，例如背部和頸部疼痛(Minimize Injury During Upper Body Exercise With Proper Posture, n.d.)、肌肉和關節損傷、肌肉拉傷(The Relationship Between Posture and Exercise, 2020)等等。此外，目前仍然很難找到任何可使用的基於人工智慧的程式幫助人們以適當的形式居家健身。

# **研究目的**

本研究論文的目的是開發一種基於人工智慧的程式，該程式專注於圖像處理並結合人體姿勢估計技術，以幫助人們以正確的姿勢進行健身。這項研究宗旨在預防和評估運動過程中的不良姿勢，同時為人們提供正確的指導，以僅使用一台設備就能提高運動質量。

# **預期成果**

此次項目旨在實現一系列目標。以下項目為我們希望達到的預期成果：

* 提高訓練質量，在不受傷的情況下獲得最佳效果
* 創建一個軟體，利用圖像處理來幫助使用者在鍛煉時調整姿勢
* 能夠顯示使用者動作的準確度
* 可以幫助使用者計算運動次數
* 介面乾淨、方便使用

# **理論探討**

1. **Model選擇**

首先我們的目標是想要正確偵測姿勢，因此會需要一個能夠影像處理的工具，於是我們在眾多能辨識人體關節點的Model中作選擇：

* MediaPipe：由 Google 開發的開源跨平台機器學習Model，用於進行電腦視覺和音訊處理等任務。提供了一個包含各種預訓練模型和工具的函式庫，讓開發人員能夠輕鬆地構建影片和音訊處理軟體，包括人臉識別、手勢識別、關鍵點檢測、物體檢測、姿勢估計、視訊穿戴式裝置等。
* OpenCV：為一個開源的電腦視覺函式庫，由 Intel 公司開發，提供了各種圖像處理和電腦視覺相關的函數和工具，可以用於實現各種電腦視覺軟體。它支援多種平台和編程語言，如 C++、Python 和 Java 等，並且提供了豐富的文檔和示例，方便開發人員進行學習和開發。
* TensorFlow：一個由 Google 開發的開源機器學習Model，用於構建和訓練機器學習模型的函式庫，它支持深度學習和機器學習，並且能夠在不同的平台上運行。也支援多種機器學習任務，如圖像分類、目標檢測、自然語言處理等等。
* PyTorch：開源的機器學習Model，支援動態圖和靜態圖兩種模式。它提供了一個 Python 界面，可以方便地進行各種機器學習和深度學習任務的開發和實現。並以其簡潔的設計、靈活性和易於使用而受到了廣泛的歡迎。
* Caffe：開源的深度學習Model，以 C++ 為主要編程語言，具有很高的運行速度。在 CPU 和 GPU 上都可以實現高效的計算，也具有良好的模組化結構，易於擴展和定制。支援多種電腦視覺任務，如圖像分類、目標檢測等等。

經過多次的開會，以及和指導教授的討論，我們最終選擇了Mediapipe，並使用Python語法來編寫，因為Python是相較其他程式語言比較方便且簡單的語法，且Mediapipe可支援這個語法。另外為什麼決定要使用Mediapipe，原因主要是針對以下幾點：

* **開源、免費**：Mediapipe 是一個開源的函式庫，可以免費下載和使用。可以方便我們自由地修改、擴展和共享 Mediapipe 的代碼。
* **跨平台**：支持多種平台，包括桌機、移動設備和嵌入式系統等。能方便我們在不同的硬件平台上運行和測試自己的應用。
* **高效率**：提供了高度優化的算法和計算引擎，可以實現高效的影片處理和機器學習。此外，Mediapipe 還支持多線程和GPU加速，可以進一步提高處理效率。
* **容易使用**：Python API簡潔、易於使用，使得我們可以方便地進行模型的設計、訓練和測試。同時，Mediapipe 還提供了豐富的範例和解說文件，方便學習和使用。
* **多功能**：Mediapipe 提供了多種影片處理和機器學習功能，包括人臉檢測、姿態估計、手勢識別、物體追踪等。而剛好我們這次同時需要姿態和手勢的偵測。

除了上述優點之外，Mediapipe運用了深度學習技術，包括深度神經網路(DNN)、卷積神經網路(CNN)、循環神經網路(RNN)等來實現各種聲音和視覺的處理任務。且它的特徵檢測可以快速判斷出人體、手、眼睛、耳朵等，再透過運動估計(圖像或影片的變化)來推斷出物體的運動軌跡，非常適合成為我們這次專題使用的model。

1. **動作標準**

接著是要用什麼來判斷姿勢標不標準，總結開會結果，最終得出兩種想法。第一種是我們收集不同人做同一個運動的影片，並經由AI的「深度學習」訓練出該運動的模型，這種做法雖然可以減少錯誤率，但並不一定達到該運動的正確姿勢，畢竟做的人不一定是完全是專業的，另外這還需要大量的資料大概會需要數千個影像，要有相當好的設備才能完成。因此我們選擇第二種方式，在網路上尋找健身教練做某個動作的標準姿勢之錄影，然後透過健身教練影片中的姿勢，人工計算關節點之間的角度，藉此得到標準姿勢的關節位置。這種方式不僅能在找資料的過程中省去很多時間成本，也能有效地找出較正確的動作，但缺點是要比較花心力去關注身體各個關節點位置的細節。

Mediapipe 能偵測到人體的多個關節點，而我們的方式是根據影片中教練所做的正確姿勢去設定我們成品想達到的動作標準。可以在教練做該動作的時候，到達最高點以及最低點時暫停影片，找出主要關節點的標準角度，例如「深蹲」的動作，就是在站立以及蹲下兩個時刻去計算主要關節點的角度。另外我們把要偵測的動作切割成兩種狀態，以方便之後要去作計數以及姿勢辨識，像是蹲下時dir=0，站立時dir=1。透過dir的0和1數字切換來達到計數的要求。

如下圖2-1，各關節點都有相應的號碼，至於要如何判斷姿勢是否在我們規定的標準內，我們只需要找到重要關節點的位置，三個點可以形成一個對應角度。舉例來說，蹲下時是看髖關節是否到位，所以我們要計算左髖關節和右髖關節的角度，也就是找左肩-左髖-左膝(11-23-25)和右肩-右髖-右膝(12-24-26)形成的角度。這樣的方式可以方便我們去確認該動作正不正確。

Chart, radar chart

Description automatically generated

**圖2-1**人體各個關節點對應的編號

1. **進度條**

在動作的過程中，重要關節點會因為不同狀態而發生遞增或遞減，因此我們設定了重要關節點在兩種狀態會經過的角度範圍，並讓其可視化，形成進度條，方便使用者知道自己動作目前做得如何(詳情結果請參照實際成效3.計數器)。

1. **準確度算法**

經過開會討論，我們希望做出來的成品能夠偵測使用者有沒有做出正確的動作以外，同時計算準確度。而準確度的算法，最後決定的方式是利用使用者在做運動的時候，重要關節點做到定點時的角度和我們設定的標準定點角度的差距去計算準確度。

首先我們要先尋找使用者做到定點時的角度。拿深蹲舉例，在站立時宣告一個假想的髖關節定點角度(angle\_top = 180)，當我們越蹲越低時，髖關節角度(angle1)逐漸變小，因此加入一個判斷式：if angle1 < angle\_top : angle\_top = angle1，藉此可以從angle\_top得知使用者蹲到最低時的髖關節角度。當站立時(dir=1)，重設angle\_top = 180，這樣我們就可以在每一次的蹲下找到每一次定點的髖關節角度。

當使用者確實有進入到正確姿勢的範圍內，我們才會開始計算準確度，因此我們希望準確度的保底可以是60%左右。拿深蹲舉例來說，假設我們設定蹲下狀態的髖關節角度範圍是80~110，而教練做的標準髖關節角度為90，則當使用者蹲下後，準確度 : accuracy = 100 – abs(angle\_top - 90)，其中abs為取絕對值的意思。因為標準角度距離範圍角度的最大值為110-90 = 20，可以得知100最多只會減去20，間接說明出來的準確率不低於80%。我們要讓100減去的變數越接近40才能確保出來的準確度在60%~100%，因此讓計算出來的變數多乘以2，得到新的準確度 : accuracy = 100 –2\*abs(angle\_top - 90)，計算的結果正式60%~100%，並以此類推每個運動的準確度。由於每一下的angle\_top會重設，所以我們也可以得到每一下的準確度。

1. **手部辨識**

同理於人體關節點的方式，我們想要透過使用者在遠處比出手勢來達到遠端操控程式的目的。如下圖2-2，根據手指每個關節點所代表的號碼，來判斷每個手指頭是否有彎曲，並進一步推斷使用者是比出1~6的數字或是其他的手型，再透過該手型來執行對應的程式。

Text

Description automatically generated

**圖2-2**手部各個關節點對應的編號

1. **中途退出手勢**

使用者在運動的過程時，為了在中途可以方便地退出，我們討論出用手臂比出叉叉的手勢來達到遠端退出運動介面的方式。利用圖2-1的人體關節編號可以發現，當人在用手臂比叉叉時，編號13~16四點會形成一個梯形，再根據梯形上下底邊平行的關係，得知梯形的上下鄰角為互補角，相加為180度，利用這個原理來判斷使用者是否比出叉叉。



**圖2-3**人體比叉叉手勢示意圖，紅色框線正好為梯形

# **設計過程**

**討論前期 :**

在一起設計程式的過程中，礙於大家無法每次都聚在一起寫程式，組長周騏軍建立了可以讓我們五個人可以共同編輯的Github專案，實現了可以遠端一起討論及修改程式碼的問題。同時Github除了用來上傳程式碼外，也可以將收集到的資料以及範例影片、圖片藉由這個平台分享給其他人，是一個相當實用的討論工具。

另外我們在設計程式以及討論階段的前期，正是新冠疫情肆虐的時候，無法透過實體見面討論確實讓我們的專題進度慢於預期的進度，光是想著我們要做甚麼主題時就已經決定很久，前期我們利用MsTeams開會很多次，也與指導教授約了很多次的線上會議，本來一開始是想作關於手語辨識的主題，但礙於設備需求，以及需要龐大的函式庫難以蒐集和其他多種原因，經過多次輾轉才決定要設計居家健身姿勢偵測。

**實作階段 :**

而後來的開會的主題我們圍繞在哪一些室內運動適合做為我們這次專題主題的素材，我們預期是希望可以做出判斷姿勢是否正確以及計數的程式，首先我們在網路上收集一些健身教練的範例影片來做為我們判斷姿勢是否在正確範圍內的基礎。一開始組長周騏軍建立好了基本手型辨識以及攝像頭的程式碼後，先利用了深蹲和伏地挺身當作實驗性的測試，發現對於運動的判斷都是分為兩個部分，假設運動員在伏地挺身時，剛好分為伏地和挺身，因此我們利用這個特點來導入一個新的變數dir，透過狀態的切換讓dir在兩種狀態下分別等於0或1，藉此就能達到計數的目的了。

當然利用上面的方法可以完成大部分的居家健身動作，我們分別根據影片中健身教練的姿勢為標準，設計了深蹲、伏地挺身、開合跳、仰臥、抬腿。本來我們想要加入的動作其實還有平板撐，但發現這個動作比較特別，並沒有分為兩種狀態，而是持續在同一個狀態下完成的運動，跟我們想要方便計數的目的有落差，因此我們決定暫時移除這個項目，未來有機會會再加入這項運動。

在我們設計好上述提到的每個運動的姿勢偵測程式碼之後，我們開始進行實體的實驗，發現跟我們預期的結果差蠻多的，使用者在亂做的情況下，程式一樣會判斷使用者完成那一下運動，經過多次反覆觀察，找出問題是我們一直以來用切換狀態來計數是不合理的，因為這種偵測方式只會紀錄該動作的頭跟尾，如果中間的過程使用者是亂做的狀態，程式也不會去偵錯。為了解決這個問題，我們開會討論出的方法是，在該運動的過程中，安排至少3個過程的檢查點，讓使用者在做該運動的過程中，必須要經過那些檢查點才會計入一下，後來經過我們的再次實驗，發現這個方法其實不太容易，首先是這樣的標準太嚴格了，二來是很吃設備的需求，以我們的設備來說，很容易就當機了。

經過後來的討論結果，我們想到一個更有效抓使用者運動過程，而且不太會負荷到設備的方法，那就是在判斷dir狀態切換的外面，加入運動過程的範圍判斷式。舉例來說，伏地挺身這項運動，手臂和手肘的擺幅範圍，就可以當作過程判斷式的範圍；而過程中膝蓋不能彎，要維持接近在180度，一樣也可以寫入過程判斷式裡面，有效地約束使用者必須在規定角度範圍內完成動作，過程也不能再亂做了。

我們跟指導教授在討論時，教授提出了一個建議，那就是一開始我們的介面是有鏡頭拍到背景的，而介面上的選項文字容易跟背景混雜在一起。因此為了解決這個問題，我們將一開始的介面反白，雖然是反白但其實鏡頭還是開著的，右下角新增小文字讓使用者可以看到自己比的手型有沒有被偵測到，詳情可以看圖3-6，解決了介面畫面雜亂的問題。另外我們額外發現一個不方便的地方，那就是我們在使用的時候，很難邊做運動邊看螢幕，會不知道自己做了幾下。所以我們決定加入聲音系統，當使用者在成功完成一下的同時，會有音效做為提示。在前面選項介面選擇選項時也加入音效，達到通知使用者遠端操作成功的目的。

最後我們利用了過程判斷式和判斷狀態的延伸來加入準確度以及錯誤偵測，讓使用者在做運動時，每完成一下會告知使用者準確度多少，又或是做不完整時會告知使用者哪邊的動作細節沒有做好，達到類教練的目的，讓使用者在居家健身時，可以有效改善動作不良的地方，並提高健身的效率。

# **實際成效**

在本章節中，將討論程式的功能和程式的實際結果。結果以及特徵將以多種方式呈現，使我們的成果更易於被理解，包括:

1. **使用者介面**

當使用者開始運行我們的程式時，他們會看到一個黑色背景的螢幕和請選擇您要執行的操作指令。正如我們在下圖3-1中所看到的，第一頁包含6個不同的選項，我們可以選擇包括 : 仰臥起坐、伏地挺身、深蹲、仰臥抬腿、開合跳、離開。對於想要學習動作的正確姿勢或只是單純練習動作的使用者，他們可以對著鏡頭簡單用手比出1~5之間的對應數字開始程式。使用者也可以選擇數字6離開程式。

此外，當使用者選擇其中一個練習時(選項1~5)，將出現一個包含新選項列表的新頁面。如圖3-2中，我們可以看到第一個選項是範例影片，它會示範如何做到正確姿勢；第二個選項是開始運動，引導使用者在程式的幫助下進行練習；第三個是回到上一頁，將使用者帶回到上一頁。

如果使用者選擇選項2，將出現一個帶有“請選擇您要做的次數”指令的新頁面。正如我們在下圖3-3中看到的，現在使用者可以選擇他們想要做多少次練習，也可以返回到上一頁。選擇次數後，設備的攝像頭將打開，程式可以跟蹤和統計此次的運動，如圖3-4所示。**在練習結束時，程式會告訴使用者做了多少次“您已完成?次”和總準確度“準確度為%”。**另一方面，使用者也可以使用手檢測器功能來選擇選項，這將在下一部分與圖3-4中的其他功能一起解釋。

Text

Description automatically generated**圖 3-1**選擇1~5做練習，選擇6退出程式

Text

Description automatically generated

**圖3-2**選擇圖3-1中的練習後出現新的選項列表

**圖3-3**選擇練習的次數Graphical user interface, text, application

Description automatically generated with medium confidence



**圖3-4**打開相機並跟蹤使用者的移動

1. **手部檢測器**

 通過向相機展示使用者的手，手部檢測器功能將跟蹤他們的手或手指運動。用於選擇螢幕上列出的選項。例如，如果使用者要選擇選項1、2、3，他們可以做出如圖3-5所示的手勢，與選項4、5、6的原理相同。

**圖3-5**選項編號1、2、3的手指手勢

換句話說，當使用者執行程式時，使用者的相機從一開始就已經打開了。然而由於螢幕背景是黑色的，並且沒有在螢幕上顯示用戶的手部動作，用戶不知道程式是否可以正確讀取手部。所以為了解決這個問題，在螢幕右下角的小文字會顯示程式實時識別的數字。如果顯示的文字是“NAN”（非數字），則手勢無法識別。

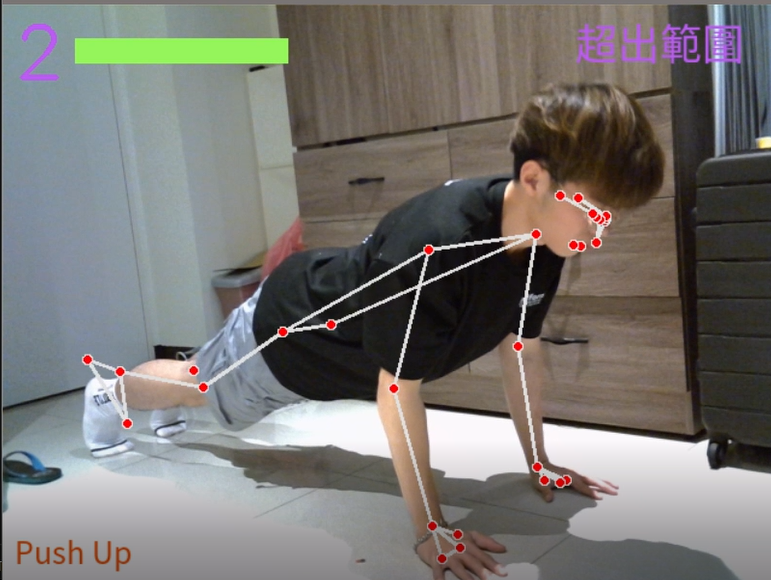


**圖3-6**右下角的小文字

1. **計數器**

 當使用者選擇進行鍛煉並打開相機時。如上圖3-4我們所看到的，在左上角有一個計數器可以幫助使用者計算他們已經做了多少次動作。在計數器旁邊，有一個綠色的進度條，會根據每一下的進度擺動。

**圖3-7**計數器和綠色的進度條

進度條作為一個參數，讓使用者知道他們在每次的運動練習中是接近指定角度還是偏離指定角度。例如在伏地挺身中，我們可以看到使用者抬起身體時，進度條增加；當他的身體向下時，進度條倒退。形成一次循環。

**圖3-8**挺身時，綠色進度條正在增加

**圖3-9**伏地時，綠色進度條正在減少

1. **準確度和錯誤檢測**

準確度用於讓使用者知道他們的動作距離標準有多接近，錯誤檢測則讓使用者知道他們需要修復哪些動作的細節以獲得高精度並讓程式對其進行計數。當相機打開時會位於螢幕的右上角。



**圖3-10**在顯示準確度前會先顯示 : 開始動作

一開始，它會如圖3-10所示，之後對於每一次的成功計數，程式都會更新一次準確度，出現的數字取決於使用者最近的動作有多正確。舉例開合跳來說，當使用者跳了正確的一下之後，左上角計數1次，右上角也顯示動作的正確百分比如圖3-11所示。



**圖3-11**正確完成一下後，準確度顯示在右上角

 然而，如果使用者的動作有一點點錯誤但仍然可以識別，它會告訴你的動作的哪一部分是錯誤的。例如在圖3-12中，當計數仍然是 1 時，使用者張開了手臂。但當他再把手臂合上時，計數並沒有增加，仍然顯示為1。正如我們在右上角看到的，這是因為他最近的動作中，他把手舉得太高了，在右上角顯示為“手太高”。最後，如果使用者的動作完全錯誤且無法識別，除了不會計數之外，右上角也會顯示“超出範圍”。

**圖3-12**用戶舉起手臂但超出正確範圍，右上角顯示超出範圍

**圖3-13**完成一下但不夠標準時，右上角顯示手太高

1. **中途退出功能**

如果在練習過程中，使用者通常必須做到他們選取的指定的計數次數才能結束。若想在做到一半中途退出時，使用者可以簡單地用雙手對著相機做一個“X ”手勢，如下圖3-14所示。

**圖3-14**兩手比X將中途退出，螢幕返回到第一個主頁（圖3-1）

1. **聲音系統**

有時使用者在運動時很難看到螢幕，導致使用者容易不知道自己做了多少次次數，也不知道自己的動作是否被程式識別。因此，聲音系統有助於通知使用者是否檢測到他們的動作或計數器是否正在增加。有時雖然計數器沒有增加，但仍然可以有通知聲音，這樣代表程式可以識別運動，但使用者的運動有一點錯誤，需要改進的地方會顯示在右上角的文字（錯誤檢測）。

# **優缺點**

在可用性方面，這個程式可以幫助使用者在進行運動時進行計數，並確保每個手勢和動作對於計數器都是正確的。這次成品非常適合剛開始鍛煉的人。新手可以通過播放範例影片來學習如何做這個運動。通過準確度和錯誤檢測，使用者可以輕鬆地識別他們的動作有多精確，以及如果程式檢測到他們的動作有錯誤，應該修復哪些動作細節。

在我們加入手部檢測器之前，很多時候我們在使用程式中的運動練習部分時需要返回到設備來更改選項，相當麻煩。自從有了手勢辨識，使用者就可以不需要在他們的設備前來回選擇選項，只需簡單地在遠處做出選項編號的手勢，程式就會自動作識別。聲音系統還可以幫助使用者通知他們的動作是否可以被程式識別。

但是，此程式僅與 PC（桌機）和筆記本電腦兼容，不與智慧手機兼容。對於沒有電腦或筆記本電腦的用戶來說，要使用這個程式不太方便。此外，使用者開啟攝像頭開始做運動時，有時畫面會發生卡頓現象，卡頓的嚴重程度取決於使用者的設備的性能。這種卡頓往往會影響程式跟蹤使用者移動的性能。

# **未來展望與期許**

我們希望在未來它能得到更多的發展，讓這個程式適合更多人使用。讓更多的人喜歡運動，養成健康的生活習慣。我們也希望這個程式可以幫助更多喜歡運動或想要學習如何正確健身的人。

期許未來可以在程式中加入更多的運動項目，並且可以盡可能兼容任何設備，讓人們隨時隨地使用起來更加方便。

# **參考資料**