

國立台北大學資訊工程學系專題報告

The Introduction of the Format of Project Report

專題組員:張瀚升、張瑋倫、溫景翔、李信緯

專題編號:PRJ-CSIE-114-008

執行期間: 2025 年 9 月 至 2026 年 5 月

1. 摘要

本專題實作一套結合電腦視覺與 AI 教練回饋的手部運動偵測系統，目的是協助使用者在日常使用 3C 產品之餘，透過簡單的手部訓練達到保健與互動練習效果。

系統以 MediaPipe 進行手部關鍵點偵測，並搭配 CVZone、距離與角度計算，以及狀態機流程，判斷使用者是否正確完成動作。本系統包含指尖互碰、雙手彈指、十指交扣與虎口夾緊四種手勢，並針對各動作設計校正機制與錯誤容忍機制，以降低手部晃動、遮擋或關鍵點不穩造成的誤判。系統亦整合大語言模型作為 AI 教練，定期根據訓練數據產生文字與語音回饋，提供使用者即時修正建議。

2. 簡介

(一) 研製背景

近年來，隨著科技的高速發展，人們對於 3C 產品的使用時間大幅增加。然而，長時間使用不僅容易造成手部疲勞、疼痛，甚至可能引發相關疾病。此外，短影音平台的盛行，

使人們獲取娛樂的方式變得快速且碎片化，減少深度思考的時間，進而增加腦部老化的風險[1]。

基於上述問題，我們決定設計一套系統，讓使用者在日常使用 3C 產品之餘，仍促進手部活動與腦部運作，達到保健與訓練的效果[2]。

(二) 研究目標

我們希望完成一個手部運動偵測系統，在執行過程中，使用者可以自行選擇要訓練的動作，接著跟著教學影片一起練習。練習過程中，透過 MediaPipe[3]以及 CVZone[4]進行手部關鍵點偵測，再計算這些關鍵點之間的距離與角度，判斷是否符合設定好的標準，使用者能夠即時在畫面上看到自己有哪些錯誤。同時也結合 AI 教練功能，透過語音提示及畫面字幕來提醒使用者該如何調整。

(三) 主要預期效益

當系統可以準確判斷使用者動作的正確性，且 AI 教練[5]也能夠針對主要問題提供具體且即時的錯誤回饋

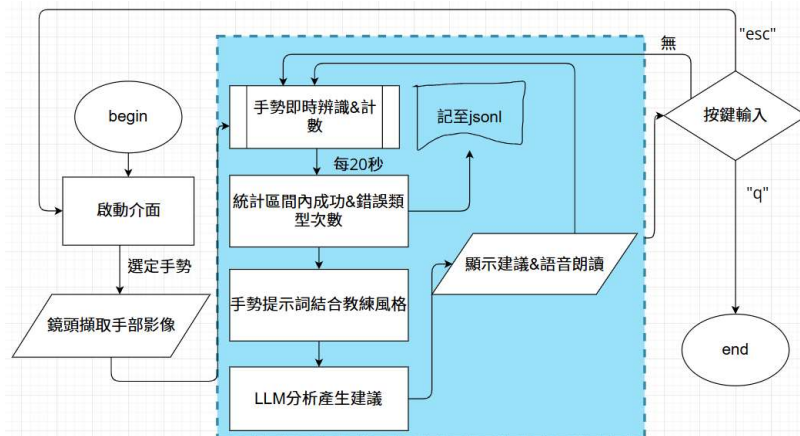


圖 1. 系統總流程

與修正建議，使用者就可以透過在家中進行簡單的訓練，降低因長時間使用 3C 產品所帶來的健康風險。

3. 專題進行方式

(一) 系統架構

我們將本專題分為以下三個子系統：(1)啟動介面、(2)手勢辨識、(3)教練回饋，整體流程如圖 1 所示。其中手勢辨識子系統涵蓋以下四種手勢動作：(a)指尖互碰、(b)雙手彈指、(c)十指交扣、(d)虎口夾緊。

1. 啟動介面

本專題的啟動介面具備 UI 互動與嚴謹的行程管理機制。當系統開啟時，會先進入主畫面(如圖 2 所示)。當點擊「開始訓練」按鈕後，系統會切換至動作選擇畫面(如圖 3 所示)，提供使用者從四種訓練項目中選擇動作。使用者按下「啟動」按鈕後，主介面會以獨立行程的方式開啟對應的訓練程式，並在背景持續保持運作。為了防範多個訓練程式同時啟動而導致硬體資源衝突，系統內建了互斥鎖定機制，確保同一時間只能執行一個訓練程式。此外，系統具備同步關閉機制，當啟動介面被關閉時，會自動偵測並一併關閉仍在執行的訓練程式，確保所有行程同步結束以防止資源殘留，詳細的流程如圖 4 所示。



圖 2. 系統主畫面



圖 3. 動作選擇畫面

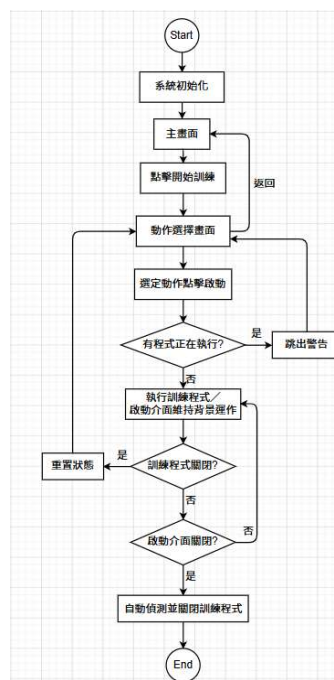


圖 4. 啟動介面架構圖

2. 手勢辨識

這部分是根據在啟動介面選擇的手勢，透過鏡頭即時辨識使用者手部動作，統計正確次數與各項錯誤情況。每 20 秒產生一個數據紀錄，供後續 AI 教練回饋進行分析並提供建議。

2.1. 指尖互碰系統

指尖互碰系統採用校正機制與多階段狀態機來進行辨識。系統啟動後，會先要求使用者將雙手置於鏡頭前進行校正，在校正期間收集雙手手掌大小的數據，作為後續判斷的基準值。進入訓練後，系統會持續檢查雙手與鏡頭的距離，若手部過近、過遠，或左右手大小差異過大，便會暫時觸發警示，避免因位置不佳造成誤判。

系統會先計算左右手對應指尖的距離，每一組指尖距離都會與接觸門檻值比較，此門檻值是依據校正階段取得的手掌大小換算而成，當指尖距離低於門檻值時，該組指尖才會被判定為完成接觸。

為了避免手部晃動或單一影格辨識不穩造成誤判，系統加入短時間的接觸保留機制，當某一指尖曾被判定

為接觸，即使後續出現短暫偏移，也不會立刻視為發生錯誤。當只有部分手指完成接觸時，系統會停留在預備接觸狀態，等待其餘手指完成接觸。直到所有指尖皆達到接觸條件後，才會進入下一階段。

系統接著會進一步檢查雙手是否維持正確的拱形姿勢。系統會計算兩手腕及兩手掌心間的距離，確認使用者在指尖互碰的同時有保留適當的手掌弧度。若五組指尖接觸且手腕、掌心距離皆達到設定門檻，系統才會判定本次動作成功，進入接觸狀態並累計成功次數；

完成一次動作後，系統會要求使用者要先將手指確實分開。當指尖距離皆超過釋放門檻值，且接觸狀態完全清除後，系統才會回到分離狀態，準備偵測下一次動作。整體流程如圖 5 所示。

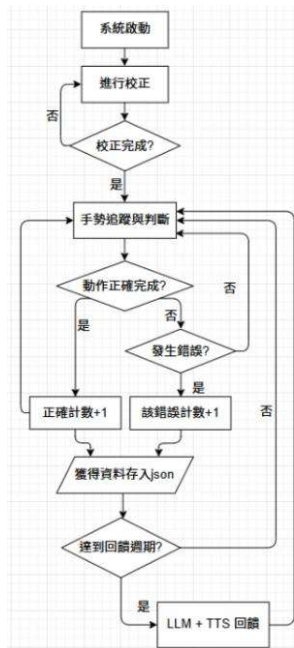


圖 5. 指尖互碰系統架構圖

2.2. 雙手彈指系統

雙手彈指系統採用狀態機架構來辨識彈指動作，並內建距離偵測及各種錯誤的處理機制。首先系統透過手掌比例分析手和鏡頭的距離，若過近或過遠都會發出警示，以確保辨識品質。在有效空間內，系統一次只會

鎖定並追蹤一根手指，以避免多指同時動作的干擾。

進入特徵判定後，系統依據手指結構進行分流，一般手指計算指間關節的彎曲角度，無名指則計算指尖至掌指關節的長度比例。當目標指尖與拇指指尖的距離低於門檻值，且彎曲程度達標時，判定手指進入捏合狀態，狀態轉移至第一階段，並同步記錄蓄力起始時間。進入蓄力狀態後，系統持續追蹤指尖距離、伸展變化與停留時間，並進行分流判斷。當指尖與拇指距離超過釋放門檻值，且瞬間達到完全伸直標準時，轉移至第二階段，累計至成功計數器並解除手指鎖定；而若超過釋放門檻但未達完全伸直標準便提前滑開，判定為爆發力不足之代償動作，累計至錯誤計數器；最後，若維持蓄力捏合狀態超過 1 秒鐘仍未釋放彈出，則判定為使用者發力遲緩或動作猶豫，系統主動中斷並累計至停滯計數器。完成上述任一分支判定後，系統會重置狀態變數並解除手指鎖定，以確保下一輪動作偵測的精確性。整體流程如圖 6 所示。

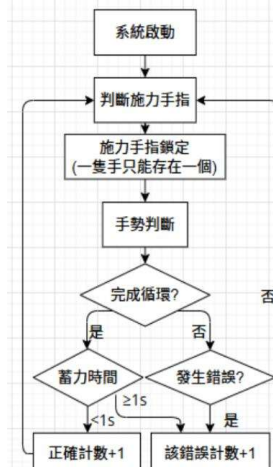


圖 6. 雙手彈指系統架構圖

2.3. 十指交扣系統

系統啟動後先透過校正階段計算手和鏡頭的相對距離，並在運行時持續確認距離是否在可接受範圍內。系統會從準備狀態開始，確認左右手皆在正確位置且大拇指朝上。

當雙手交會時，就進入扣入階段。一開始會先預設存在深度錯誤，透過其中一隻手指尖的平均 X 軸位置相對於雙手掌指關節的比較，判斷手指是否進入對方手掌範圍。當偵測到雙手手指在 X 軸方向交扣足夠深入時，清除深度錯誤並進入拔開階段。

在拔開階段，系統收集雙手各自內部的指尖開合間距，當此距離小於閾值時，代表手指有因用力對抗而夾緊，視為有用力拔開。當雙手再次張開時結算本次動作的所有錯誤，若本次動作從準備到拔開狀態沒有發生任何錯誤，就計入成功次數；否則計入錯誤次數。整體流程如圖 7 所示。

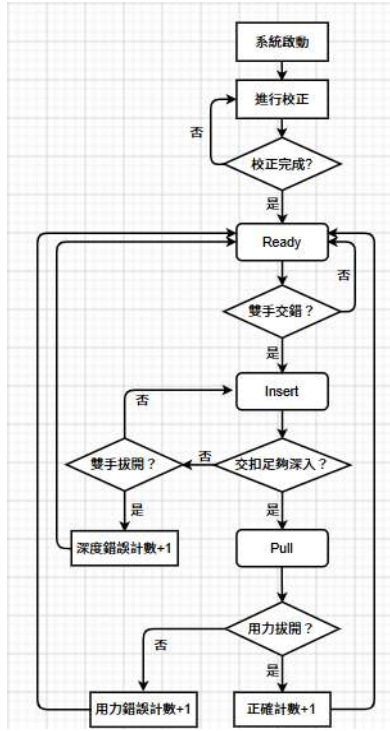


圖 7. 十指交扣系統架構圖

2.4. 虎口夾緊系統

虎口夾緊系統採用三階段狀態機進行辨識，系統先計算兩手拇指掌指關節關鍵點之間的距離，當距離低於門檻值並且有一隻手為施力手時，就將狀態轉移至第一階段(虎口插入)。

進入虎口插入狀態後，系統先計算食指指尖到另一隻手腕的關鍵點距離來判斷哪隻手為施力手，接著計算施力手的拇指指尖到食指近端指

間關節的距離當作捏合量。當捏合量低於門檻值時，確認虎口有效施壓，狀態轉移至第二階段(虎口夾緊)。

當雙手距離超過釋放門檻值時，進入到第三階段(腕部分開)。系統依據當前狀態進行分流判斷：若未經過第二階段便提前釋放，判定為動作不完整，累計至虎口無夾緊計數；若完整經歷全部階段，則判定為有效動作，累計至成功計數器。

在虎口插入和虎口夾緊狀態之間，系統會記錄施力手小指 y 值最低點和最高點的數值和時間，進入腕部分開時，如果最高點的時間比最低點早，就會累計至無向上傾斜計數，減少因為拇指被遮擋後以鄰近關鍵點的相對位置進行預估，導致捏合量很難低於門檻值，使錯誤計數失準。整體流程如圖 8 所示。

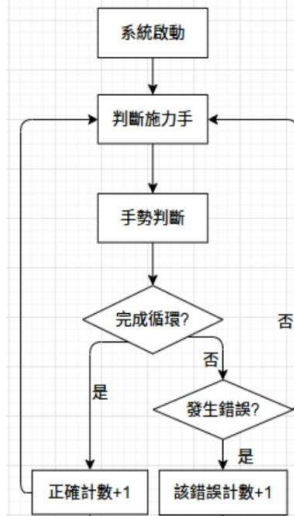


圖 8. 虎口夾緊系統架構圖

3. 教練回饋

根據手勢辨識子系統每 20 秒回傳的訓練數據，透過 AI 模型產生即時的動作建議，並以文字與語音同步呈現給使用者。為了確保四種手勢的回饋風格一致，我們設計了一套共用的教練風格 prompt。此外，在每種手勢各自的 prompt 會先介紹錯誤類型與判斷邏輯，讓 LLM 理解在做什麼，依照使用者目前最主要的問題，結合共同的教練風格模板給予建議。

當 LLM 生成回饋，建議文字會

即時顯示在畫面下方的「AI Coach」區塊，同時也會透過 Edge TTS[6]以中文語音朗讀這段文字。

(二) 主要困難與解決辦法

(1) API 同步執行造成系統阻塞

本系統需要在不影響手勢辨識的前提下，串接大語言模型。為了確保系統的高效和流暢度，我們必須學習如何實作多執行緒的架構。

由於能實作的時間不多，所以我們採用 AI 協作優化的實作模式。利用生成式 AI 作為輔助工具，針對執行緒管理及複雜資料結構進程式碼優化和即時偵錯。在 AI 的協助下，我們實現了非阻塞式的 API 串接結構。如圖 9 所示，透過建立獨立的背景執行緒與佇列進行資料交換，主執行緒負責即時手勢辨識，並將每 20 秒的分析結果透過佇列傳遞至背景執行緒。背景執行緒持續監聽佇列中的資料，如果偵測到有新資料產生，會即時呼叫 LLM 進行分析。

```
def llm_worker():
    """背景執行緒"""
    print("[LLM Worker] 啟動，等待 log 產生...")
    while True:
        last_line = _get_last_json_line(LOG_FILE)
        if last_line:
            try:
                feedback = build_gemini_prompt(last_line)
                _llm_feedback_buffer["text"] = feedback
                with open(FEEDBACK_FILE, "w", encoding="utf-8") as f:
                    f.write(feedback)
            except Exception as e:
                print(f"[LLM Worker 錯誤] {e}")
                time.sleep(LLM_INTERVAL)
        else:
            print(f"[LLM Worker] {LOG_FILE} 尚未產生，等待中...")
            time.sleep(8)
    print("[System] 啟動 LLM 背景線程...")
    llm_thread = threading.Thread(target=llm_worker, daemon=True)
    llm_thread.start()
```

圖 9. LLM 背景執行緒

(2) 關鍵點判斷不準確

在系統開發過程中，我們發現無論是使用 MediaPipe 還是 CVZone 進行手部偵測，當雙手交疊或接觸時，皆會出現短暫偵測不穩定的問題，在雙手發生交疊或接觸的瞬間，關鍵點會有錯位或是消失的情況發生，導致計算出的角度或是距離出現異常。

為了解決此問題，我們的方法是引入錯誤容忍機制(如圖 10 所示)，透過設定異常狀態的幀數閾值，規定

錯誤狀態的持續時間必須要高於此閾值，系統才視為有效狀態。

```
if candidate == distance_issue_candidate:
    distance_stable_count += 1
else:
    distance_issue_candidate = candidate
    distance_stable_count = 1
if distance_stable_count >= DISTANCE_STABLE_FRAMES:
    distance_issue_stable = candidate
```

圖 10. 錯誤容忍機制

(3) Prompt 設計

在本系統中，prompt 的設計是 AI 教練能否精準運作最關鍵的因素。我們觀察到 AI 教練的回饋存在三大問題：回饋語氣和敘述方式不夠自然、無法準確針對問題進行回答以及給出的建議與偵測到的錯誤類型不符。

經過反覆的調校，我們歸納出三個 prompt 優化策略：不要規定太多細節，保留適度空間讓 LLM 發揮、減少讓 LLM 執行邏輯判斷和條件取捨以及引入 XML 標籤來規範輸入與輸出的格式。尤其，XML 結構能夠在核心語意不變的情況下，大幅提升 LLM 的回饋品質(見圖 11)。

```
COACH_FEEDBACK_POLICY = """
<task>
根據程式已判斷出的教練資料，產生一句即時復健回饋。
這句話要讓使用者知道下一次動作應該調整什麼，並保持溫和、清楚、可立即執行。
</task>

<principles>
- 相信資料中的主要問題，不重新推測未提供的姿勢細節。
- 只提最重要的一件事。
- 若沒有明顯問題，給具體且穩定的鼓勵。
- 可以自然改寫，不要每次使用固定句型。
</principles>

<style_rules>
- 使用繁體中文。
- 回應長度盡量在 30 字以內，最多不超過 40 字。
- 不列點，不加標題，不提原始數字。
- 不輸出 XML/HTML 標籤。
</style_rules>

<output_format>
只輸出要顯示的教練回饋文字。
</output_format>
"""
```

圖 11. AI 教練回饋 prompt

4. 主要成果與評估

在手勢辨識方面，正常拍攝環境的情況下，程式幾乎能夠精準辨識使用者的動作正確性，也能正確歸類各種錯誤。然而，在一些像嚴重遮擋等特殊情況下，系統仍會出現不少誤判，也顯示出僅依賴關鍵點距離與角度進行運算的侷限性。而在 AI 教練回饋方面，LLM 提供的回饋都能對應到每次測試的狀況，代表系統輸出的特徵數據能夠和 prompt 有效結合，針對

使用者出現的問題進行引導與修正，達成AI教練互動的目標。圖12~15為各動作的實際執行畫面。



圖 12. 指尖互碰執行畫面



圖 13. 雙手彈指執行畫面

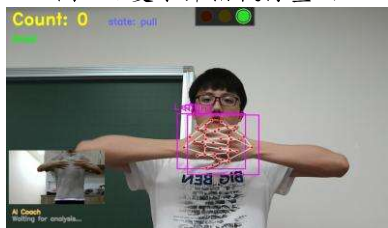


圖 14. 十指交扣執行畫面



圖 15. 虎口夾緊執行畫面

5. 結語與展望

本專題透過 MediaPipe、CVZone 與大語言模型 (LLM) 等多項工具，完成結合電腦視覺與智慧互動的手部運動偵測系統，目的是要緩解現代人過度使用 3C 產品導致的手部疲勞與潛在健康風險。

在未來的延伸發展上，我們希望能夠增加更多的手部保健招式，提升個人化訓練的實用性。此外，針對目前系統多依賴固定的幾何門檻值來判定動作狀態，未來預計導入機器學習模型進行動態調整，以提升系統在不同使用者個體差異下的辨識準確度。

6. 銘謝

感謝指導教授在這次專題製作中的悉心指導。在每週的定期會議中，針對研究內容提供寶貴建議與精準指正，協助我們奠定扎實的基礎與研究方向，最後得以順利完成。

7. 參考文獻

[1] 長谷川嘉哉, "動動手指，9大功效防失智," *康健雜誌*, Nov. 8, 2017.

[Online]. Available:

<https://www.commonhealth.com.tw/article/76130>

[2] 健康 2.0, "簡文仁 10 招健腦操訓練手腳協調預防失智症!【如果云知道鄭凱云】feat.簡文仁物理治療師," *YouTube*, Apr. 8, 2022. [Online]. Available:

<https://youtu.be/lzwuxstXycY>

[3] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device real-time hand tracking," *arXiv*, Jun. 2020. [Online]. Available:

<https://arxiv.org/abs/2006.10214>

[4] Computer Vision Zone, "cvzone," *PyPI*, ver. 1.6.1, Sep. 2023. [Online]. Available:

<https://pypi.org/project/cvzone>

[5] N. Tambe, "How AI is replacing the personal trainer: Inside the future of adaptive fitness coaching," *Polaris Market Research*, Mar. 2026.

[Online]. Available:

<https://www.polarismarketresearch.com/blog/how-ai-is-replacing-the-personal-trainer-inside-the-future-of-adaptive-fitness-coaching>

[6] rany, "edge-tts," *PyPI*, ver. 7.2.8, Mar. 2026. [Online]. Available:

<https://pypi.org/project/edge-tts>