

第二十三屆旺宏金砂獎

作品企劃書

AR 穴位解析—岐黃妙訣

AR analysis of acupoints

參賽組別：應用組

參賽編號：A23-162

隊 長：魏仲彥

隊 員：徐伯元、李偉德、莊汶娟

目錄

(一) 緒論.....	3
1.1 前言.....	3
1.2 研究動機與研究目的.....	3
1.2.1 研究動機.....	3
1.2.2 研究問題.....	3
(二) 背景知識與相關技術介紹.....	4
2.1 文獻探討與回顧.....	4
2.2 AR 穴位解析—岐黃妙訣內容簡介.....	5
2.3 Unity 使用結構.....	5
2.3 穴位映射原理.....	6
2.4 AI 問診工作原理.....	7
(三) 企劃研究方法及步驟.....	8
3.1 系統架構.....	8
3.2 企劃研究步驟.....	8
3.3 軟體端.....	9
3.3.1 AR 穴位映射.....	9
3.3.2 手部姿勢辨識.....	11
3.3.3 custom medapipe calculator.....	12
3.4 硬體端.....	14
3.5 伺服器端.....	15
(四) 系統實測.....	17
4.1 軟體端測試.....	17
4.2 伺服器端測試.....	20
4.3 系統整合測試.....	21
(五) 結論及未來展望.....	23
5.1 結論.....	23

5.2 未來展望.....	24
(六)參考文獻.....	24

(一) 緒論

1.1 前言

智慧鏡子系統：在家也能輕鬆保養身體。現今疫情肆虐全球，大多數人生了小病也不會出門，而是選擇待在家裡靜養，疾病影響人類的生活，許多人患了重大疾病往往因為缺乏知識而選擇忽視，導致延誤就醫，造成嚴重的後果，甚至死亡。根據世界衛生組織2020年提供的全因死亡報告[1]，可以得知因疾病死亡的人數之多，我們對於其重視程度也隨之提高。為此，我們開發了一款智慧鏡子系統，讓使用者能夠在家輕鬆保養身體。

本系統使用Jetson Nano J1010作為開發板，並搭載Mediapipe技術實現了AR手部穴位互動，讓使用者可以透過手勢與智慧鏡子互動，查看對應穴位點在身體的位置。我們還設有手機APP與智慧鏡子連結，使用者可以選擇想要了解的症狀，螢幕會反饋使用者舒緩症狀的穴位，本系統也提供AI問診，使用Rasa + BERT技術，針對使用者輸入進行分析，判斷可能的疾病或是症狀，並回傳建議穴位，進一步的進行症狀舒緩。這種基於穴位保健的系統可以提供一種安全、方便、高效的保健健康保健解決方案，尤其是對於缺乏經驗和醫學知識的人來說，更具有重要的價值。

本計劃的創新點在於結合智慧鏡子、手機APP和AI+AR技術，實現了基於穴位的健康保健解決方案，與傳統的健康保健方式相比，更具備了便利性和效率性

1.2 研究動機與研究目的

1.2.1 研究動機

按穴道是保養身體的方法之一，有很多能夠治癒疾病的特效穴位，不但可以達到治癒單一疾病的效果，還可以調整全身機能，強身健體，但是穴位遍布人體，穴道的實際位置往往要藉由書本或他人的協助，初學者才能得知，我們希望使用者能一眼看出手部穴道在自己身上的位置與作用，使得使用者能更容易運用手上以及頭部的穴道，做到即時緩解症狀的功效，進一步預防疾病的發生。

對於偏遠地區的診所資源相對而言較缺乏，本計畫希望可以建構簡單的醫療輔助系統，透過手機APP當作客戶端，讓使用者輸入相關症狀以及相關資訊，紀錄到雲端伺服器上，讓看診醫生能夠快速以圖表的方式了解使用者的狀況。以節省醫療資源。

1.2.2 研究問題

本計畫使用Unity當作開發平台，以jetson nano j1010當作開發版，實作智慧鏡

子，使用者只需要使用手勢，就可以操控智慧鏡子的運作。本計畫還有設立遠端伺服器，可以讓使用者進行問診以及資訊回饋，本計畫所要探討研究問題包含：

- (1) 如何讓穴位以 AR 的方式正確的顯示在螢幕上？
- (2) 如何運用神經網路進行手勢辨識？
- (3) 使用 Jetson Nano J1010 的效能如何進行改進？
- (4) 使用 BERT 模型進行疾病對話推斷的效果如何？
- (5) 對於疾病分析技術的改善與發展？

(二) 背景知識與相關技術介紹

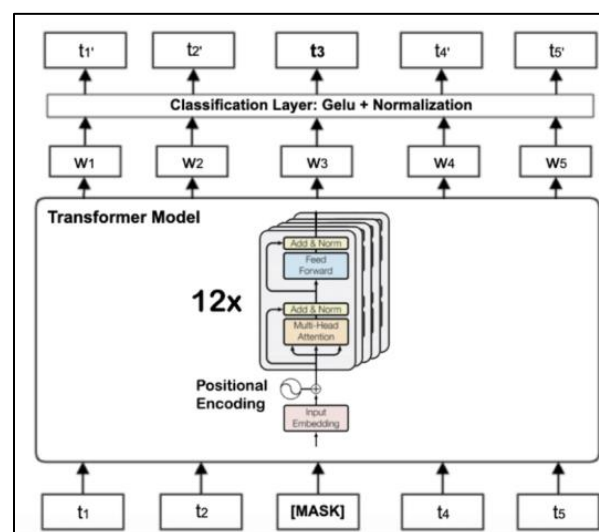
2.1 文獻探討與回顧

目前人工智慧在自然語言處理、影像辨識、視覺檢索、醫療影像分析等領域已有了很大的進步 [3, 4]，尤其是最近幾年來，隨著電腦規格的快速提升，並且各領域人才的成長，人工智慧的實力也在逐年增強。

2.1.1 Mediapipe

MediaPipe[5]是由Google開發的一款開源的跨平台多媒體處理框架。它提供了一組用於圖像(graphy)和視頻處理的模板，可以用來構建複雜的多媒體應用程序。MediaPipe支持多種平台，包括Android、IOS、Linux和Windows。它提供了許多內置的算法，如人臉檢測(Face Detection)、全身偵測(Holistic Detection)、手部追蹤(hand tracking)等，並且可以方便地擴展添加自己的算法。MediaPipe可以在多種環境中使用，如在手機、嵌入式設備上。本計畫運用 Mediapipe 當作 AR 顯示的工具，利用Mediapipe提供的Holistic模型[6]，可以透過模型得到對應的身體定位點資訊，使用並自定義DNN calculator，達到辨識手勢的功能。

2.1.2 BERT



圖一 BERT 架構圖

BERT (Bidirectional Encoder Representations from Transformers) [7]是一種用於自然語言處理(NLP)的預訓練深度學習模型，由 Google 研發出來的 Transforms[8] decoder 部分延伸出來的。BERT 能夠對句子進行雙向編碼，也就是同時考慮句子中所有詞的前後文脈信息。這使得 BERT 在許多 NLP 任務中都有出色的表現，包括情感分析、機器翻譯、關鍵字提取和問答系統等。本計畫使用的 BERT 的架構包括了 12 個 transformer 層(見圖 1)，這是一種能夠有效地學習長距離依賴關係的神經網絡架構。BERT 在訓練過程中使用了一種稱為自適應學習的技巧，使得模型能夠根據輸入的句子動態地適應不同的輸入長度。此外，BERT 也使用了殘差連接和自注意力機制(self-attention)[8]，這使得模型能夠更好地捕捉句子中的相關性。透過 BERT 模型，我們希望能夠藉由與用戶的對話來推斷用戶的疾病，給出相對應的特效穴位並顯示在智慧鏡子上。

2.2 AR穴位解析—岐黃妙訣內容簡介

本企劃主要是利用 Jetson Nano J1010 開發板和 Webcam 來擷取使用者的手勢動作，並顯示在智慧鏡子上。透過單面鏡和液晶螢幕，讓穴位能夠顯示在鏡子上，達到智慧鏡子與使用者互動的功能。主要包括以下幾點：

- (1)開發板：本企劃所採用的開發板為 Jetson Nano J1010，這是一個功能強大、高效能的單板電腦，擁有多核處理器和強大的 GPU，能夠提供足夠的運算效能來處理影像辨識的工作。
- (2)手勢識別：使用 Webcam 讀取使用者的手勢動作，通過對手勢的辨識，能夠實現呼叫選單、取消動作、選擇手勢等多種功能，使用者可以透過選擇手勢與穴位進行互動。
- (3)穴位顯示：利用單面鏡，將穴位顯示在智慧鏡子上，使用者可以直接在鏡子上查看頭部或手部穴位的詳細功能，增加了使用者與鏡子的互動性。
- (4)手機連線：透過手機與智慧鏡子的連線，能夠傳送特定療效的穴位到智慧鏡子上，也可以使用手機連線到伺服器進行 AI 問診，回傳特定療效的穴位，讓使用者能夠更快地找到適合自己的穴位療效。
- (5)醫師網頁：設立網頁，讓醫師可以進行登入，查看使用者的症狀，同時也能夠幫助醫生快速了解使用者的狀況，以便更有效地治療。

總體來說，本企劃採用了多種技術和工具，實現了智慧鏡子與使用者的互動和信息共享，能夠為使用者提供更好的療效服務。

2.3 Unity使用結構

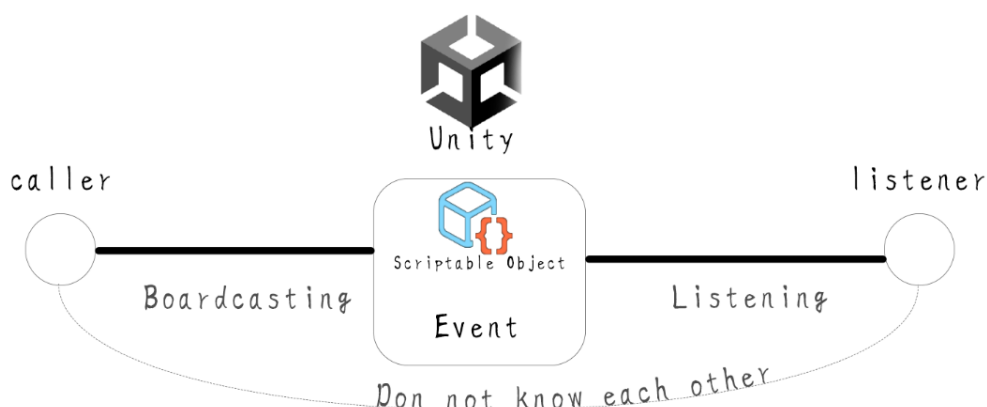
Unity 是一個非常流行的遊戲引擎，其系統架構可以使用多種方式實現。在本研究中，我們採用了 Scriptable Objects & Event System 架構[11]，相較於傳統的 Singleton 架構，這種架構具有更少的相依性。

使用 Scriptable Objects 的好處在於它的資料是全域上可存取的且與場景無

關，這意味著我們可以在多個場景中重複使用這些資料。此外，Scriptable Objects 還可以作為橋梁系統使用，讓多個物件之間進行通信和資料共享。在我們的系統架構中，我們使用 Scriptable Objects 來儲存和管理軟體中的各種資料，例如資料庫的穴位點座標、用戶訊息、觸發事件的自串等。

我們使用 Event System 來處理物件之間的通信，例如觸發事件或傳遞訊息等。使用 Event System 可以讓物件之間進行低耦合的通信，降低相依性。這樣可以使得程式更容易維護和擴展，同時也增加了遊戲的可靠性和穩定性。

總體來說，Scriptable Objects & Event System 架構具有許多優點，這種系統架構具有高度的彈性和可擴展性，可以方便地增加或刪除任何元素，而不會對系統的其他部分造成任何影響，提高軟體的可靠性和穩定性。



圖二 Scriptable Objects & Event System 架構

2.3 穴位映射原理

我們的系統主要使用 MediaPipe 的 Holistic 模型，這個模型可以幫助我們偵測人體的姿勢，包括手部和頭部的位位置。透過這個模型，我們可以得到手部的 Landmark 座標資訊，進而推算出手部的穴道位置。在推算手部穴道位置時，我們做了以下幾個動作：

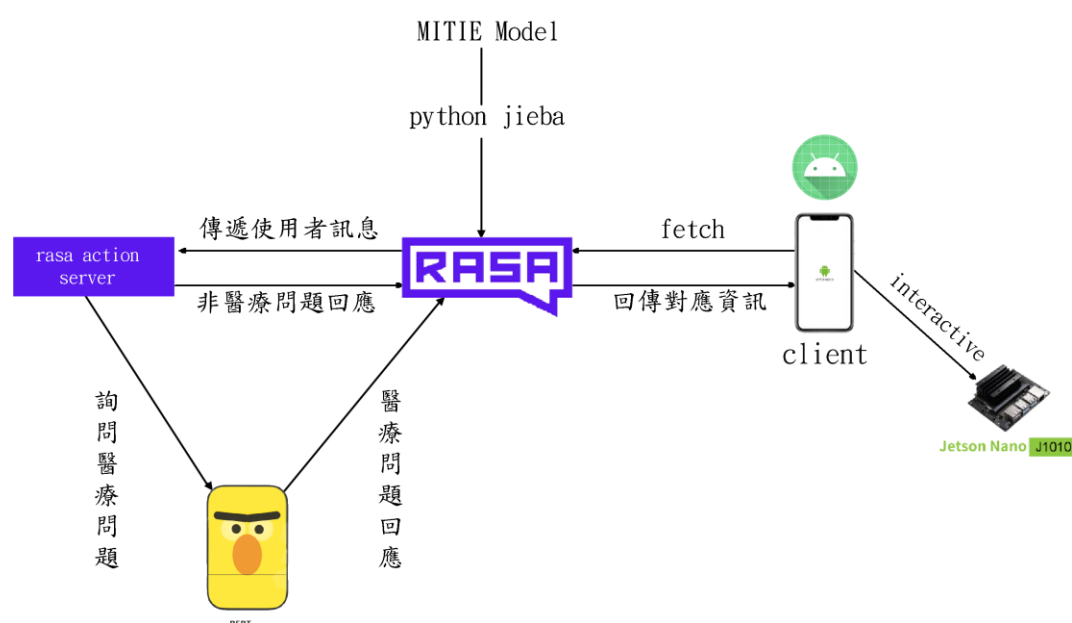
- (1)計算 Landmark 與穴道的相對位置：為了能夠動態顯示穴道位置，我們先將 Landmark 的位置與手部的穴道進行位置的計算，得到其的相對位置。透過這個方式，我們可以確定穴道的位置，並且在系統中顯示出來。
- (2)判斷手部正面背面：我們利用 Landmark 的位置與手正面背面的特性，將其中位置 0 與位置 17 和位置 0 與 5 的位置做垂直法向量。透過這個方式，我們可以確定手部的正面和背面，並且根據這個判斷來調整穴道位置的顯示。
- (3)判斷手部旋轉：我們利用位置 0 和位置 9 的當作初始向量，在計算之後的位置 0 和位置 9 之間與初始向量的角度，進而得出每個穴道點手部旋轉後的位置。透過這個方式，我們可以確定每個穴道點在手部旋轉後的位置，並且在系統中顯示出來。



圖三 手部穴位映射示意圖

而對於頭部穴位，我們則是使用相對位置坐運算。透過 Holistic 模型，我們可以得到頭部的 Landmark 座標資訊，並且透過相對位置的計算，推算出頭部穴位的位置。

2.4 AI問診工作原理



圖四 AI 問診示意圖

我們的 AI 問診主要使用 Rasa[9]+BERT，Rasa[9]是一個開源的對話人工智慧平台，主要用於開發和建立智能對話機器人。Rasa[9]的核心功能是理解使用者輸入的意圖和實體，然後進行相應的回應。Rasa 的訓練資料由 NLU（自然語言理解）模型來處理，而 NLU 模型則是由 MITIE（MIT Information Extraction）模型來實現的。MITIE 模型可以將輸入文本轉換為向量表示，然後利用這些向量進行意圖和實體識別。因為本系統使用中文，所以需要先準備中文的詞向量模型，這裡採用了網路上的 total word feature extractor zh[13]。透過 jieba 進行分詞，可以將中文文本轉換為詞的序列。接著將分好詞的文本資料提供給 Rasa 進行模型訓練。在訓練完成後，當使用者輸入文字時，Rasa AI Server 會根據使用者的輸

入，利用已經訓練好的模型來預測使用者的意圖，然後進行相應的回應。另外，當使用者輸入疾病相關的意圖時，本系統建立的 action server 會將使用者的輸入送至 BERT 模型進行處理，輸出內容再回傳給使用者。

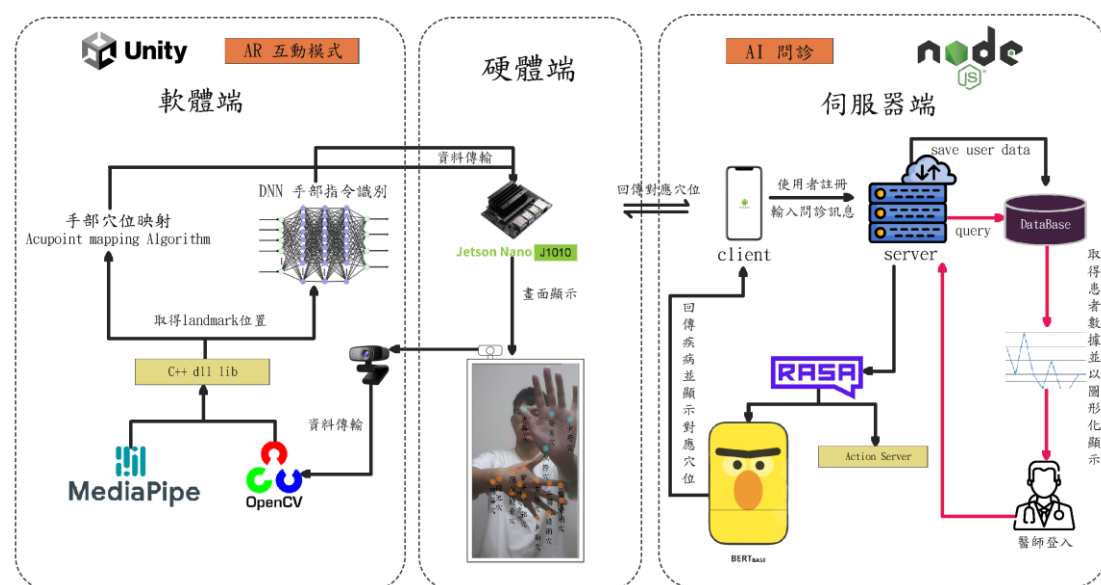
(三) 企劃研究方法及步驟

3.1 系統架構

在這個計畫中，研發了一個智慧鏡子產品，能夠辨識手部和頭部的穴位點，並幫助使用者學習和了解穴位的知識。這個產品主要包括三個部分：軟體端、硬體端和伺服器端。

硬體端使用 Jetson Nano J1010 作為開發版，並使用 Webcam 與使用者進行互動。軟體端使用 Unity 開發，並使用 MediaPipe 來獲取人體座標點，利用演算法可以從 MediaPipe 得到的座標點映射到頭部和手部的穴位，讓鏡子可以顯示出穴位，讓使用者能夠學習和了解穴位的知識。

此外，這個產品還設有一個 Android APP 讓使用者下載，可以透過這個 APP 進行 AI 問診，並記錄使用者的身體狀況和相關資訊。使用者在問診時，APP 能夠連到伺服器端進行 AI 問診，伺服器會進行自然語言理解 (NLU) 並判斷使用者的身體狀況，並將使用者的資訊存到伺服器上。醫生可以透過網頁登入伺服器，快速了解使用者的身體狀況和相關資訊。這樣的設計能夠讓醫生更快地了解使用者的身體狀況，提高診斷效率，也讓使用者可以更加方便地獲得醫療幫助。

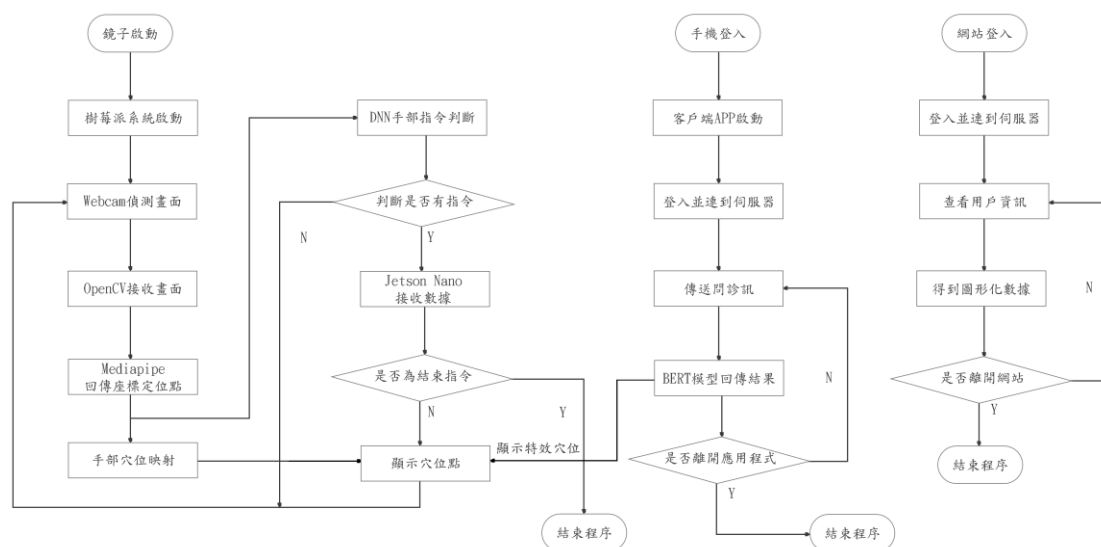


圖五 系統架構圖

3.2 企劃研究步驟

在研究步驟中，我們主要分成三大部分來討論，分別為軟體端、硬體端以及伺服器端，本計畫的智慧鏡子結合了硬體端和軟體端的技術，而伺服器端則是用

於醫療輔助系統，圖六是我们的系統流程圖。



圖六 系統流程圖

3.3 軟體端

軟體端負責處理運算邏輯、AR穴位顯示邏輯、資料庫連接處理、手部姿勢辨別，在數據處理完以後，與硬體端進行交互，進行畫面顯示。軟體端在這個計畫中是非常重要的部分，因為它負責整個系統的運算邏輯和使用者體驗，而且需要與硬體端緊密配合，以提供完整的功能。

3.3.1 AR穴位映射

Algorithm 1 Acupuncture Point Mapping Algorithm

1. **procedure** HandAcupuncturePointMapping(Landmarks, database, Handness, Scale)
2. $v1 = (\text{HandLandmarks}[5] \text{ position} - \text{HandLandmarks}[0] \text{ position}).\text{normalized}$
3. $v2 = (\text{HandLandmarks}[17] \text{ position} - \text{HandLandmarks}[0] \text{ position}).\text{normalized}$
4. $c = \text{crossproduct}(v1, v2)$
5. $\text{HandAcupointsData}[] \text{ currentAcupointsData}$
6. **if** Handness is left hand && c.z axis value is bigger than 0 **then**
7. $\text{currentAcupointsData} = \text{take acupoints on the back of the left hand data from database}$
8. **else if** Handness is left hand && c.z axis value is smaller than 0 **then**
9. $\text{currentAcupointsData} = \text{take acupoints on the front of the left hand data from database}$
10. **else if** Handness is right hand && c.z axis value is bigger than 0 **then**
11. $\text{currentAcupointsData} = \text{take acupoints on the back of the right hand data from database}$

```

12. else if Handness is left hand && c.z axis value is smaller than 0 then
13.     currentAcupointsData =take acupoints on the front of the right hand data
        from database
14. end if
15. referenceLength = Distance between Landmarks[0] position and
        Landmarks[1] position
16. fromVector = (0,1,0)
17. toVector = ( Landmarks[9] poistion - Landmarks[0] position ).normalized
18. Quaternion q;
19. vector a = crossproduct(fromVector, toVector)
20. q.xyz = a;
21. q.w = sqrt(((fromVector.Length ^ 2) * (toVector.Length ^ 2)) +
        dotproduct(fromVector, toVector)
22. MappedAcupoints[] AcupointList
23. repeat
24.     i <- i + 1
25.     currentAcupoint <-Landmarks[currentAcupointsData[i] coresspon
        landmark] position
26.     offset <- q * currentAcupointsData[i] coresspon offset
27.     currentLandmark += offset * referenceLength * Scale
28.     Add currentAcupoint to AcupointsList
29. until i == size of currentAcupointsData -1
30. end procedure

```

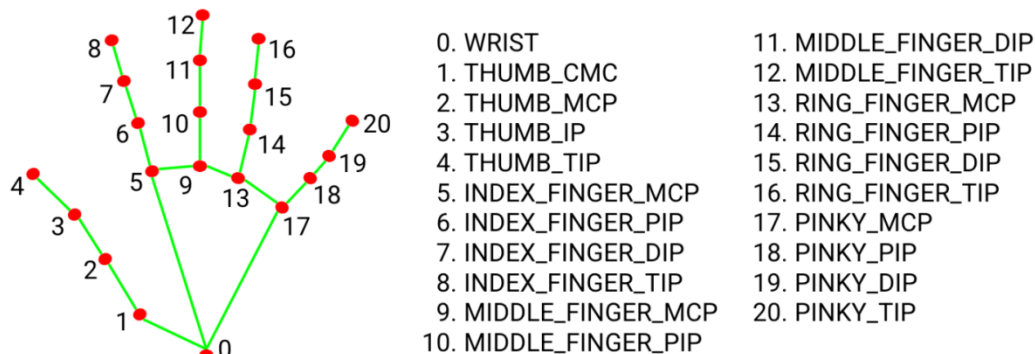
表一 穴位映射演算法



圖四 穴位映射展示

在判斷手部正反面的地方，我們使用自定義的手部穴位映射演算法（見

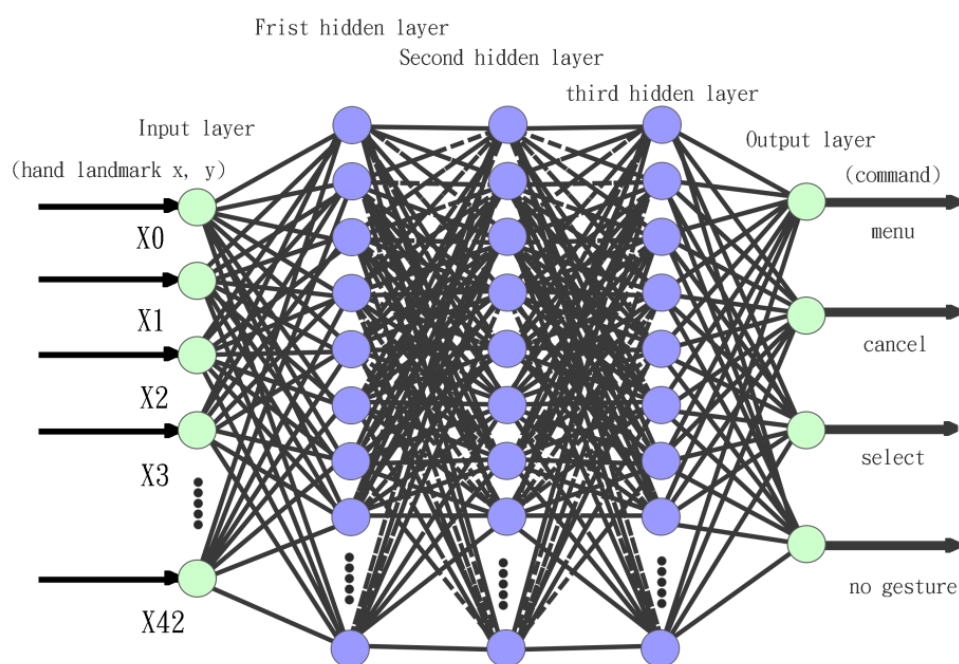
表一)，我們用外積、四位元數多種數學函、比對資料庫的相對定位點以及 Mediapipe 提供的手部定位點資訊(見圖五)，通過這些技術的結合，讓穴位可以正確顯示在螢幕上(見圖四)，並根據距離縮放。這個自定義演算法的開發使得我們的智慧鏡子產品更加準確和可靠。



圖五 Mediapipe hand_landmark

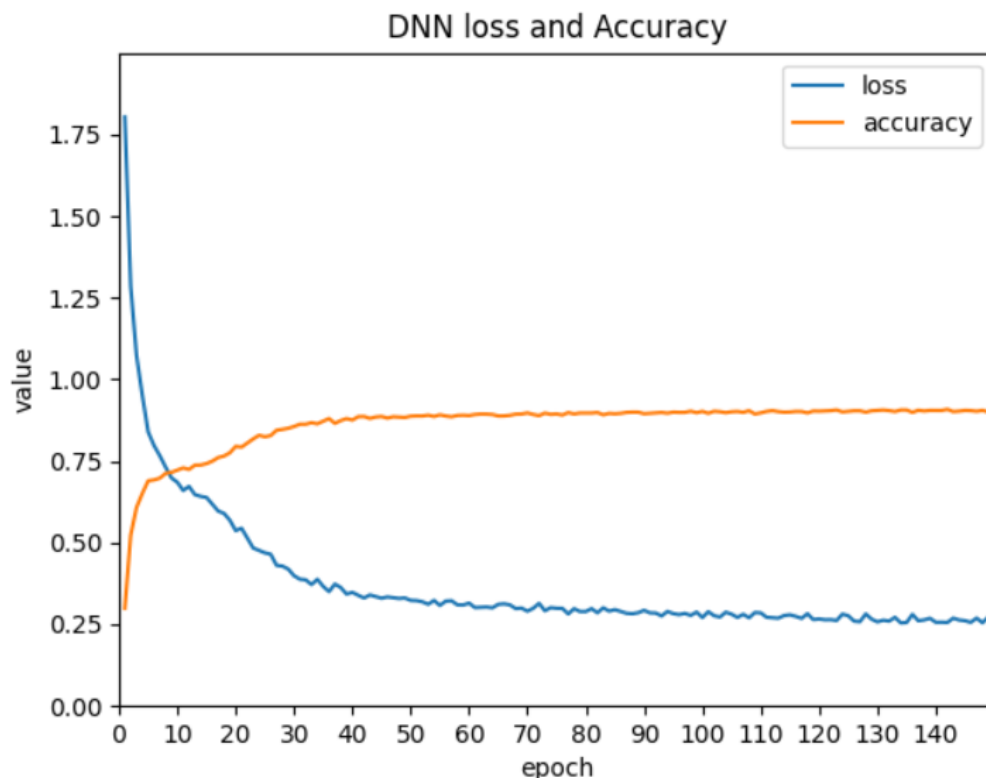
3.3.2 手部姿勢辨識

HaGRID(HAnd Gesture Recognition Image Dataset) [2]是一個手勢識別圖像數據集，本計畫使用其中的 10807 筆資料進行 DNN (Deep Neural Network) 模型的訓練。這個模型的輸入共有 42 個特徵，包括 Mediapipe 回傳的 21 個手部定位點的 X 和 Y 座標。因為輸入的特徵是座標位置，所以我們只需要簡單的三層 DNN 模型，就可以快速的分類出目前手勢是屬於哪一類型。我們的輸出一共有四類：呼叫選單、取消動作、選擇手勢以及無手勢。這樣的手勢識別功能可以讓使用者更加方便地與智慧鏡子互動，提高產品的使用體驗。



圖六 DNN 手部辨識模型

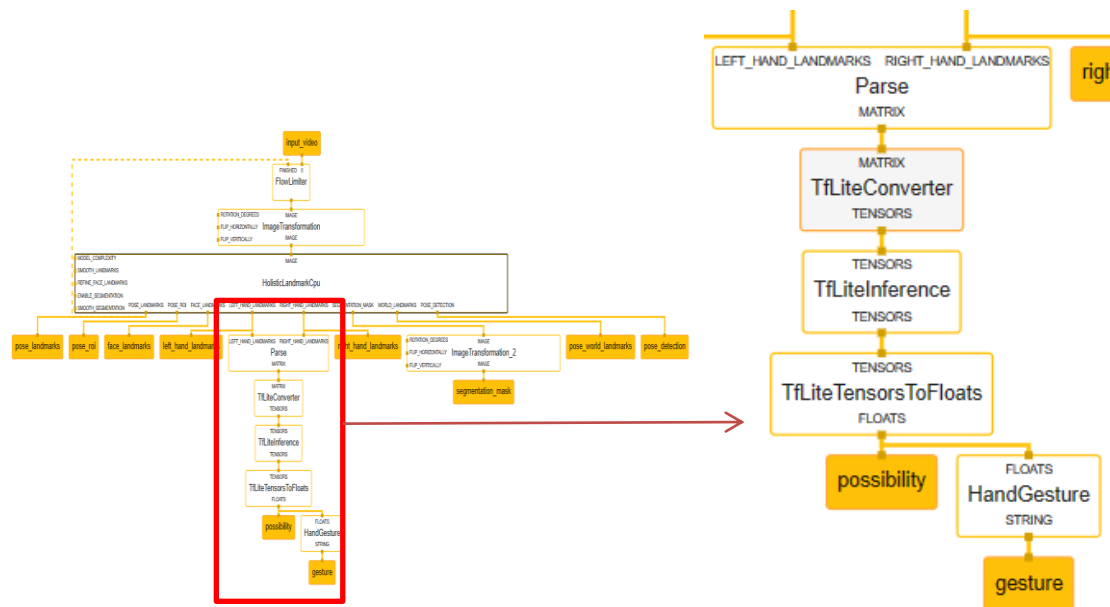
HaGRID(HAnd Gesture Recognition Image Dataset) [2]是一個手勢識別圖像數據集，本計畫使用其中的 10807 筆資料進行 DNN (Deep Neural Network) 模型的訓練。這個模型的輸入共有 42 個特徵，包括 Mediapipe 回傳的 21 個手部定位點的 X 和 Y 座標。因為輸入的特徵是座標位置，所以我們只需要簡單的三層 DNN 模型，就可以快速的分類出目前手勢是屬於哪一類型。我們的輸出一共有四類：呼叫選單、取消動作、選擇手勢以及無手勢。這樣的手勢識別功能可以讓使用者更加方便地與智慧鏡子互動，提高產品的使用體驗。



圖六 DNN 手部辨識模型準確率與損失函數

我們使用 DNN 來進行手部辨識，我們使用的 epochs 為 150，batch size 為 16，透過訓練過程中的損失函數的觀察，我們可以看到損失函數在訓練過程中慢慢下降，而準確率則慢慢上升，顯示模型的學習效果良好。本計畫的 DNN 手部姿勢辨識模型達到了高達 90%的準確率。這表示我們的模型能夠準確地判斷使用者手勢，並根據訓練時所設定的四種類型（呼叫選單、取消動作、選擇選項、無手勢）進行分類。這樣的高準確率為我們的應用提供了可靠的手部姿勢辨識功能，使得使用者可以順利進行相對應的操作。透過損失函數和準確率的監控，我們能夠確保模型的訓練過程和效果達到了預期的目標。圖六中的趨勢圖也直觀地顯示了我們模型訓練的進步和效果。這項技術在手部姿勢辨識方面表現出高度的準確性，為本計畫的成功實施提供了堅實的基礎。

3.3.3 custom mediapipe calculator

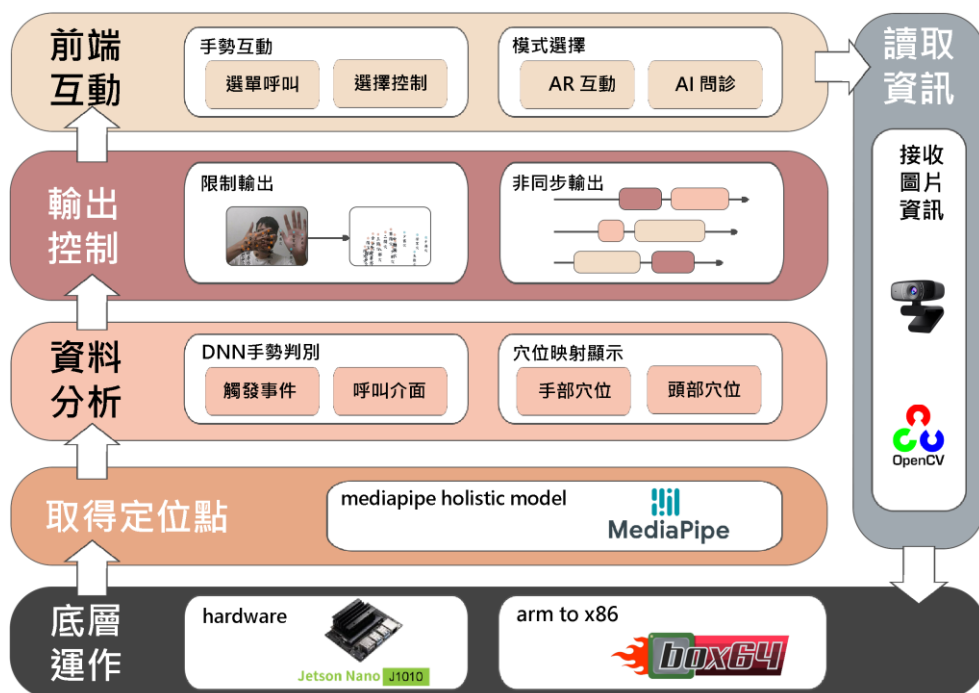


圖七 Mediapipe Graph

手勢判斷系統使用的模型是 DNN。為了使用這個模型進行手勢判斷，需要先取得 holistic landmark 的輸出左右手的 normalized_landmark_list 資料。接下來，需要修改資料型態，將左右手的資料轉換成矩陣(Matrix)格式，以便餵給 Google Mediapipe TFLite Converter Calculator 的 ImageFrame、GpuBuffer 或 Matrix 等計算器進行處理。然後，這些資料會進入 Google Mediapipe TFLite Inference Calculator，這個計算器會根據提供的模型位置載入對應的模型並進行運算。最後，運算結果需要進行資料型態的轉換，從 TFLite tensors 轉成 vector 的形式，並輸出最高機率對應的手勢字串，作為手勢判斷系統的最終輸出。

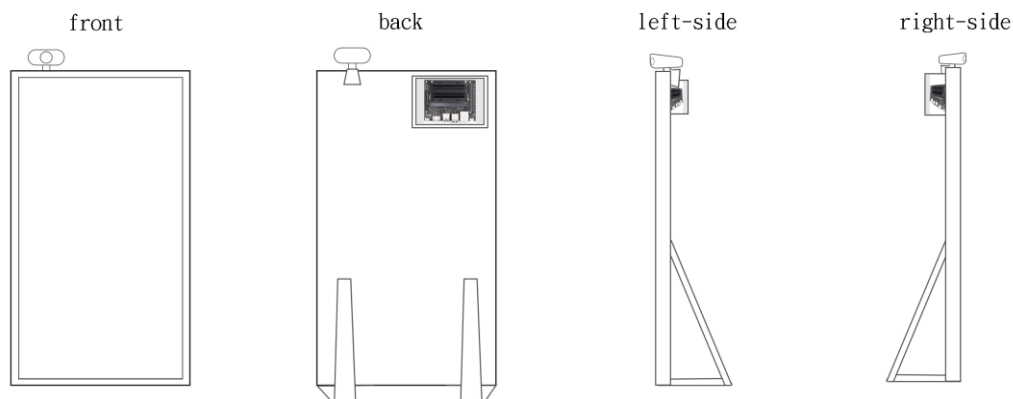
要讓 Unity 使用新的 Graph，需要增加其他相依性，以及新的 Calculator 的 cc 檔案。在進行編譯之前，需要先使用的工具 Bazel 和檔案總攬。Unity 與 Mediapipe 連接，讓中間的 Mediapipe 在執行狀態下溝通。Bazel 是編譯 Mediapipe 與 Unity Mediapipe 轉接層的工具，可以說他像是管理檔案者一樣，包括描述一個 cc (cpp 檔)他需要的相依性文件，不同檔案 txt、tflite、cc...的壓縮、轉移等操作。例如，在完成自訂的 Calculator cc 檔案後，我們需要在 Graph 中引入它。這時，Bazel 就可以幫助我們告訴這個 Graph 檔案要在哪裡找到我們需要的檔案。其中一項 Calculator (HolisticTrackingToRenderData)的相依性描述在 Bazel BUILD 中。在檔案操作方面，這個步驟是把 Mediapipe 的 tflite 的檔案直接改名成 bytes 檔案供 Unity Mediapipe Plugin[12]使用，然後進行壓縮、複製和移動檔案給 Unity 使用。總結來說，這些步驟是 Mediapipe 與 Unity 轉接層的工作。因為這是我們使用的最上端 Unity 與 Mediapipe 的工作端口，所以這些步驟十分的重要。

3.4 硬體端



圖七 產品架構圖

本計畫的產品架構如圖七所示。首先，我們會利用 Webcam 來讀取使用者的手部數據。這些數據會傳送到 Jetson Nano J1010 進行運算。我們使用 MediaPipe holistic 模型來進行手部定位，並獲得 Landmark 座標。經過資料分析後，我們會對輸出進行控制，因為在邊緣運算中需要低耗能且低附載的解決方案。為了節省繪圖的功耗，我們使用單面鏡，省去 Webcam 的輸出，使螢幕只顯示穴位點和其他必要資訊，而不顯示相機訊息，從而達到節省繪圖資源的效果。此外，我們使用非同步輸出，使使用者在視覺上能夠更順暢地與自身的手部穴位進行擴增實境互動。

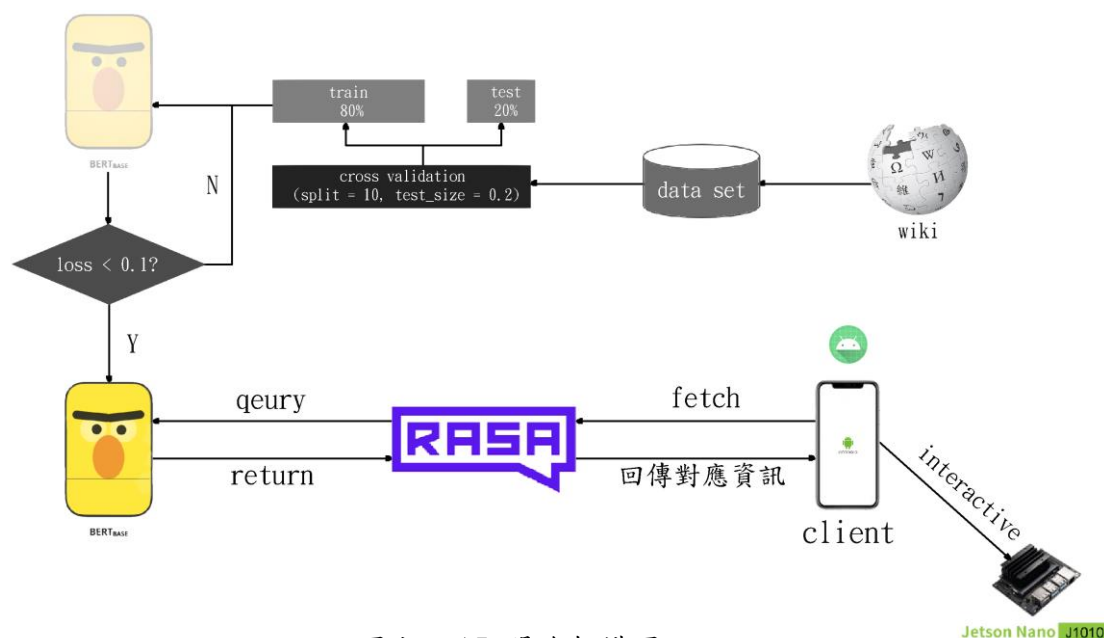


圖八 智慧鏡子設計圖

智慧鏡子的設計是基於普通家庭鏡子的模板進行延伸，使鏡子具備智能化的功能。如圖八所示，鏡子貼附在螢幕的前面，而外框將螢幕與鏡子包裹起來。在鏡子的後方，有一個盒子，裡面放置著 Jetson Nano J1010 的電路板，而在上方則安裝了用來獲取使用者影像的 Webcam。智慧鏡子的設計可以使居家更加有趣，藉由整合手部姿勢辨識技術，使用者可以透過鏡子上顯示的穴位點學習穴位知識，並且能夠隨時檢視自身的穴位，進行相應的健康管理和護理。

3.5 伺服器端

圖九展示了本計畫的AI問診架構，本計畫自行架設伺服器，並在伺服器上建設簡單的問診機器人。我們透過Rasa來建立聊天機器，當使用者的意圖為詢問疾病或是症狀，本計畫訓練的BERT模型就會根據使用者輸入，給出對應的特效穴位，並讓智慧鏡子指顯示這些穴位，讓使用者可以進一步進行症狀舒緩。



圖九 AI 問診架構圖

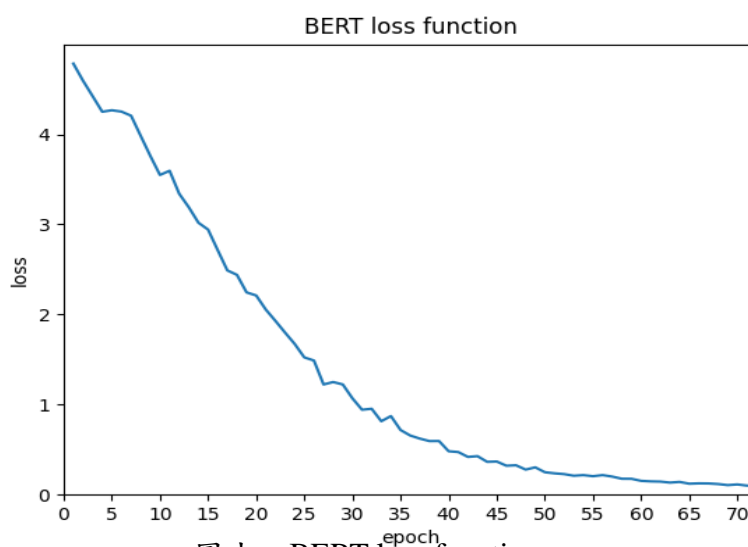
本計畫使用了 BERT 的預訓練模型使用 bert-base-chinese[10]，該模型具有 12 層 layer、768 層 hidden、12 個 heads，參數規模達到 110MB。在進行模型訓練前，我們對 BERT 模型進行 token 的輸入初始化，並加入我們從維基百科蒐集的資料。目前我們的資料集中包含 109 種類別的疾病資料，以及與這些疾病可能發生的症狀相關的對話資訊（見表二）。總共有 623 筆資料。我們使用這些資料進行模型的 fine-tuning，即在 BERT 的最後一層接上一個新的簡單的分類器，用於識別使用者輸入的疾病類別。

Id	text
流行性感冒	發燒、咳嗽、喉嚨痛、流鼻水、肌肉酸痛、疲倦和頭痛
流行性感冒	我最近常嘔吐和腹瀉

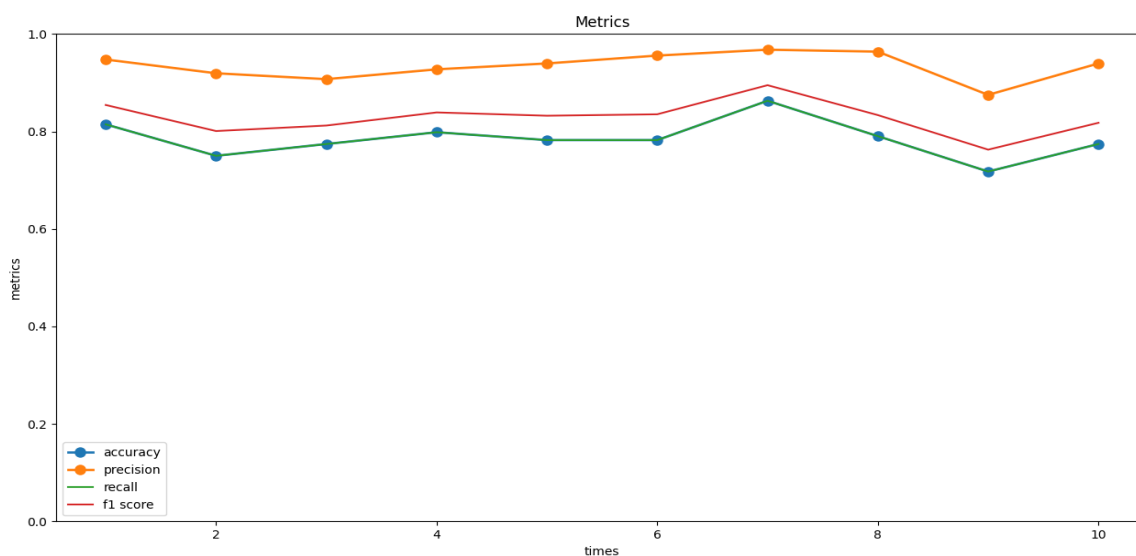
急性咽喉炎	咽部疼痛，吞嚥疼痛及吞嚥困難
....
中暑	頭暈、頭痛、口渴、呼吸急促、意識不清等症狀
消化不良	持續性食慾不振、無故消瘦

表二 BERT 訓練資料

在訓練模型時，我們使用epochs為72，batch size為10。透過觀察圖十中的損失函數變化，我們可以看到損失函數在訓練過程中逐漸下降，顯示模型的學習效果逐漸改進。這表明我們的模型在訓練過程中逐漸學會了從資料中抽取特徵，並在預測疾病類別時逐漸提高了準確性。這使得我們的模型在識別使用者輸入的疾病類別方面表現出高度的準確性。透過fine-tuning BERT 模型，我們能夠在疾病識別方面取得優越的結果。



圖十 BERT loss function



圖十一 BERT model Metrics

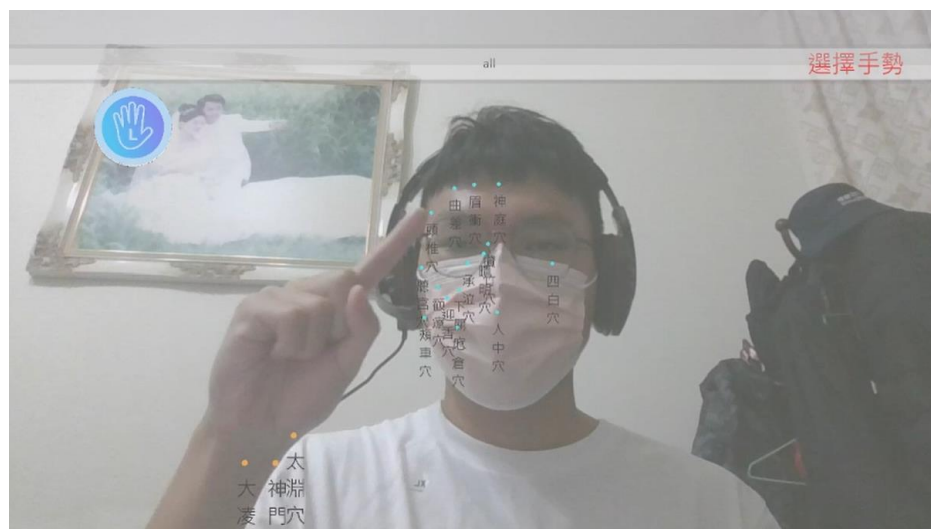
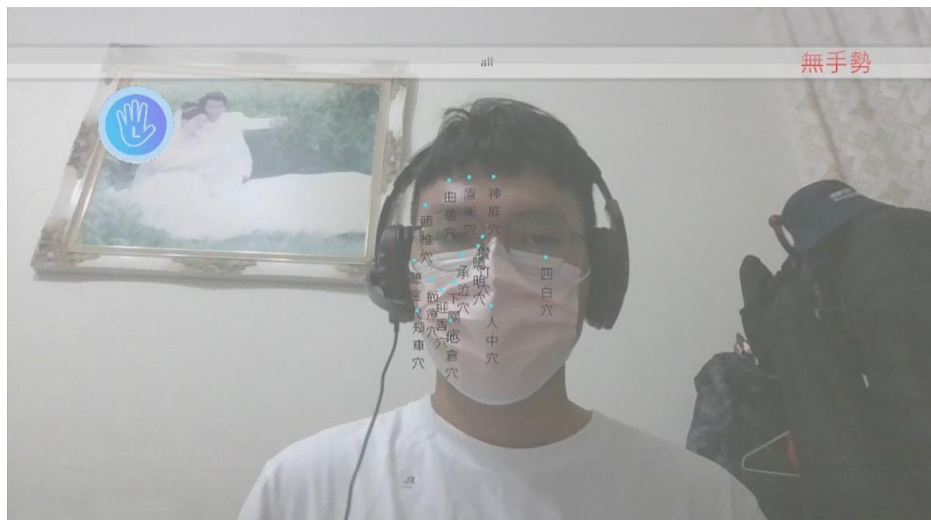
在本計畫中，我們使用了 Cross Validation 進行十次的訓練。每次訓練完畢後，我們將得到四個模型指標，包括 Accuracy、Precision、Recall、F1-Score。這些指標是用來評估模型預測的準確性和性能的重要指標。我們將十次的模型指標結果相加後取平均，作為最終的模型指標。這樣的做法有助於減少單次訓練的隨機性對結果的影響，並且能夠更全面地評估模型的性能。根據我們的結果，在預測疾病方面，我們的模型表現出色，達到近八成的準確率。這意味著我們的模型對於預測疾病的能力非常強大，對於使用者輸入的症狀和疾病類別能夠給出準確的預測結果。圖十一中的結果進一步驗我們訓練的 BERT 模型在問診機器人中的有效性和可靠性。

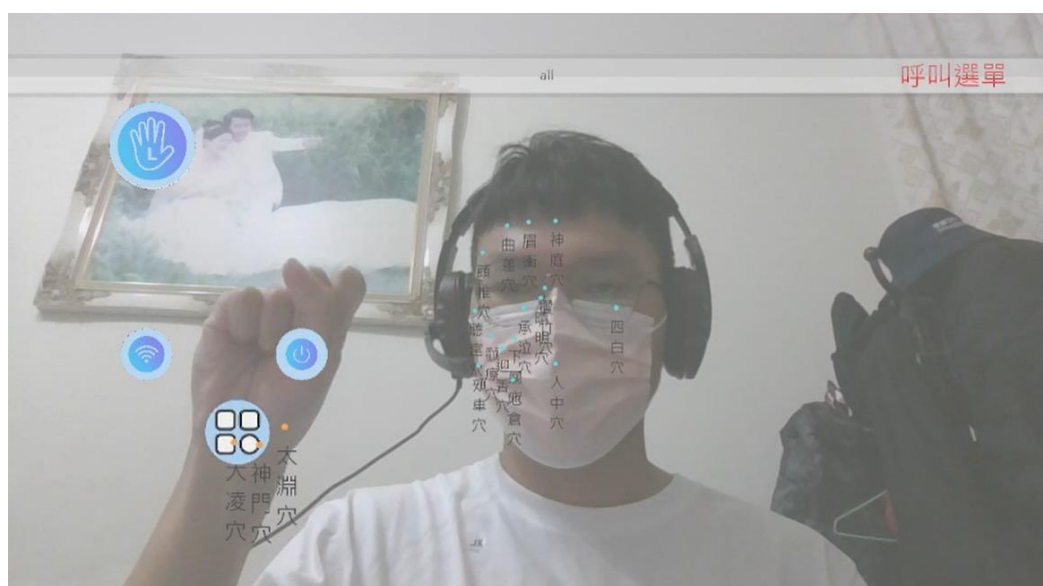
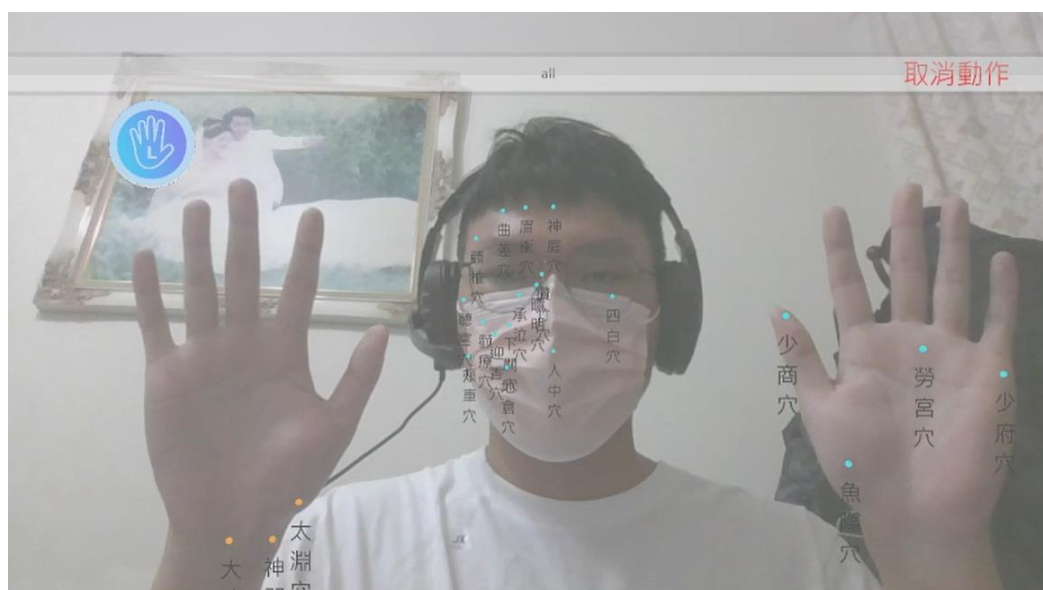
在伺服器上，預計會將使用者輸入資料分類，整理成圖形資訊，讓看診醫生能夠快速的了解病人目前的概況，節省醫療資源。

（四）系統實測

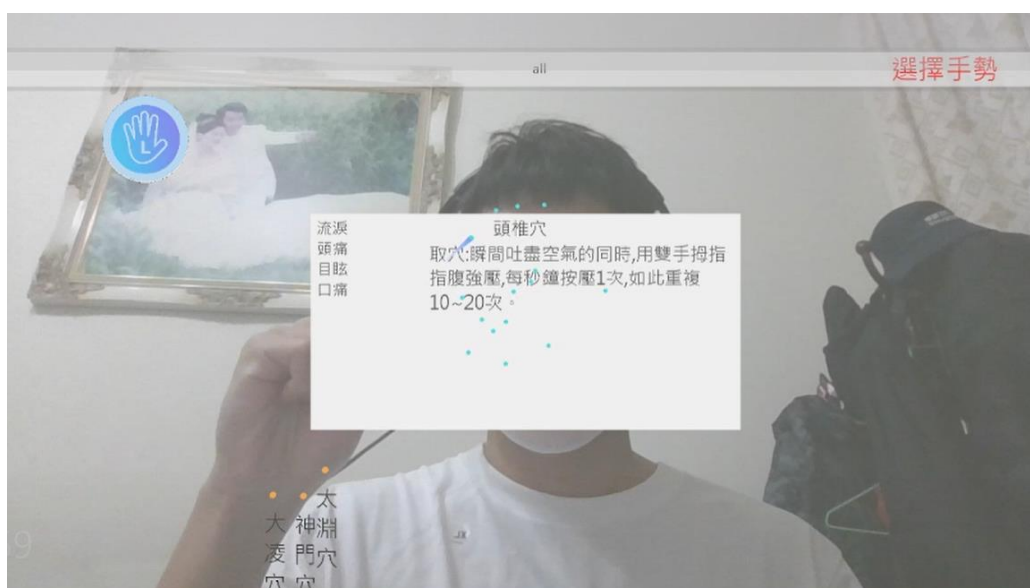
4.1 軟體端測試

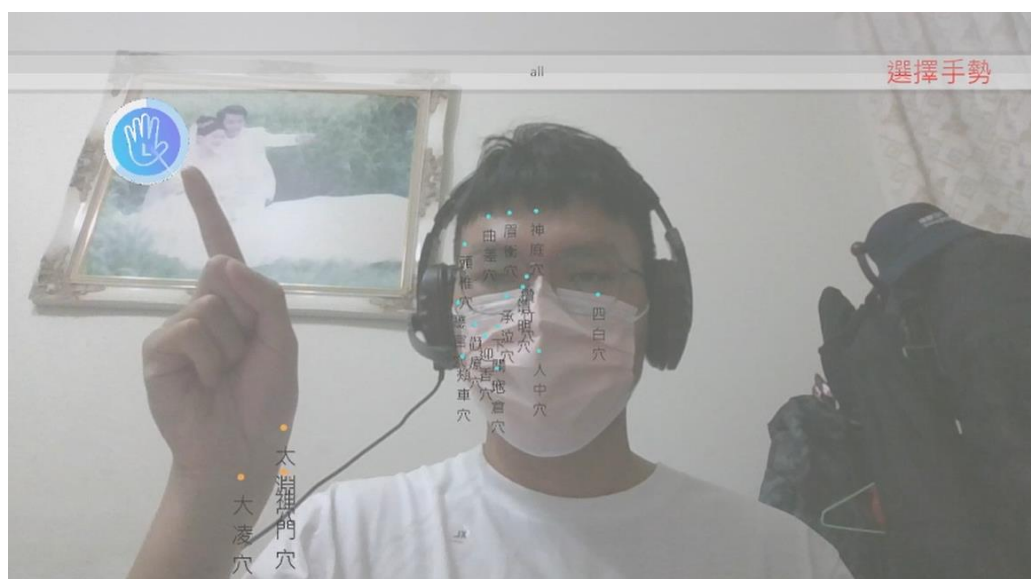
以下是軟體端實際測試，可以看到穴位點貼合在人體上，並可以判斷手勢，與智慧鏡子進行交互。



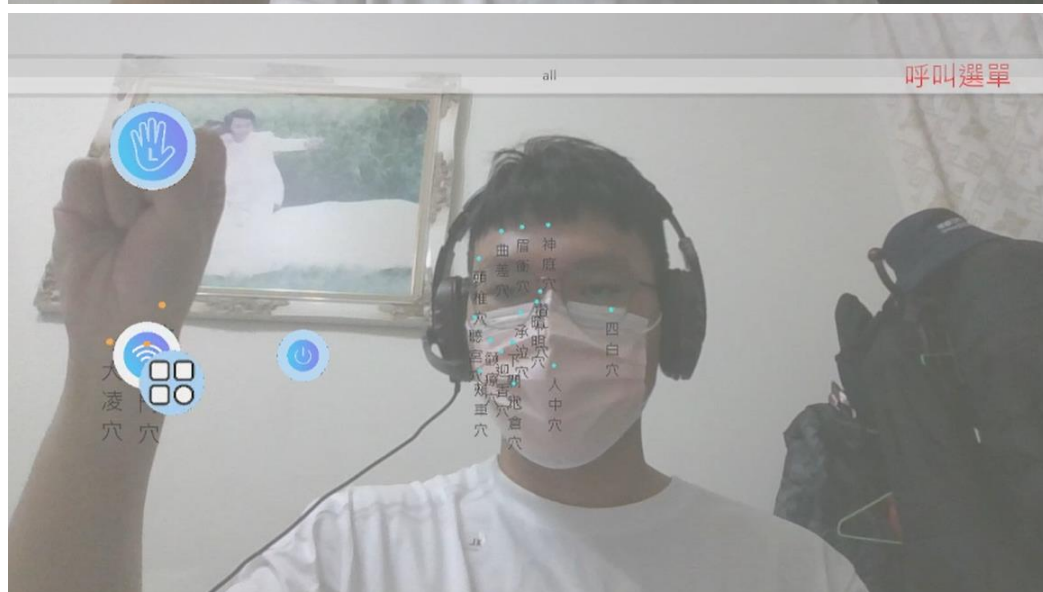


可以利用選擇手勢查看穴位資訊，或是換手判定控制手勢





呼叫選單可以控制穴位的顯示，或是控制與手機連線



4.2 伺服器端測試

伺服器主要測試網站以及 AI 問診伺服器，網站的部分，醫生可以透過登入介面登入網站，並填寫回饋資訊給我們的團隊。

用戶登入

使用舊帳戶

l23@l23

密碼

⋮

Next

標題

如有使用上問題，或是問題反饋，請填寫這個表單。

目前一個帳號只能在一天內發送一次訊息
否則系統將視為垃圾郵件

回饋給我們

內容

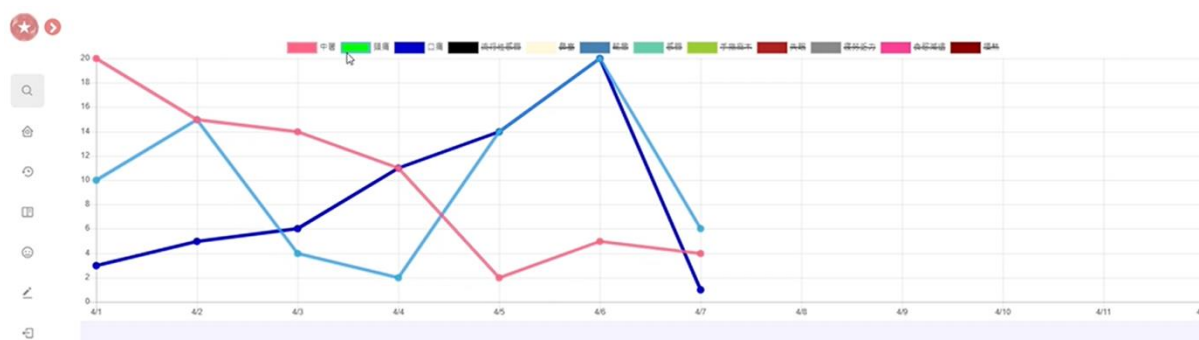
使用者名稱

電子郵件

訊息在這裡。

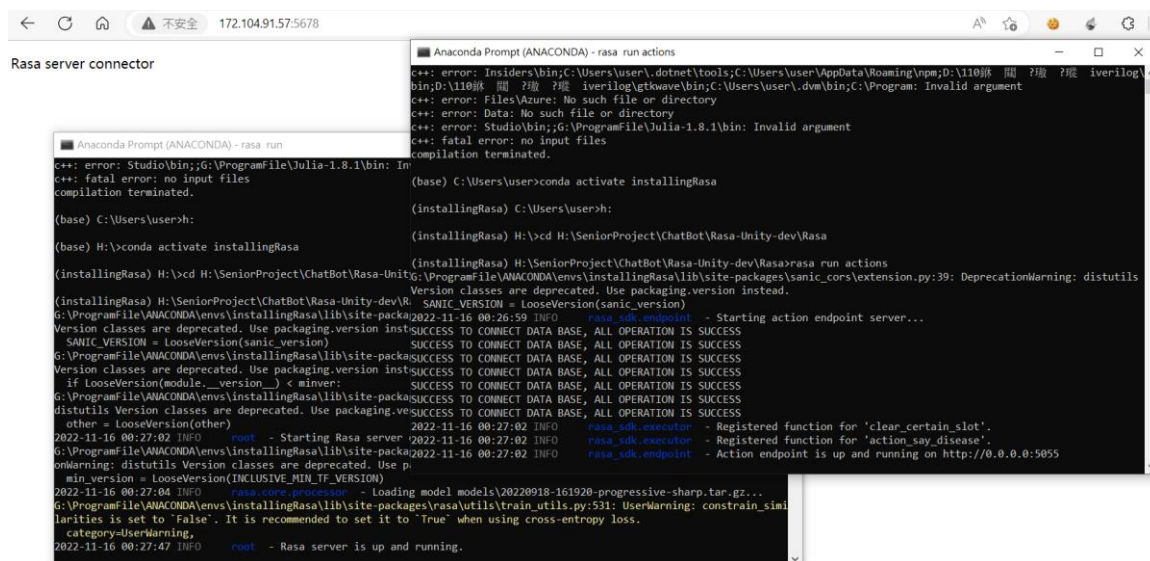
發送

醫師也可以從資料庫裏面讀取使用者數據，得知使用者的疾病發生次數



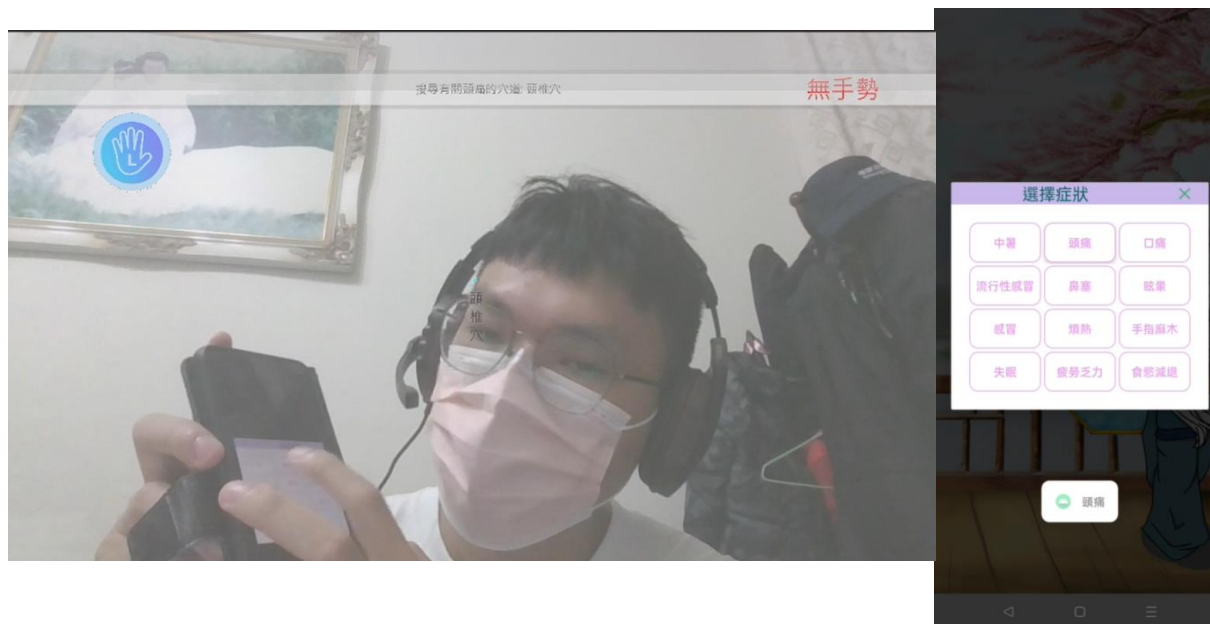
遠端 AI 問診伺服器，使用公開伺服器，架設 Rasa，並在裡面加入 BERT 模

型，讓手機可以進行 AI 問診

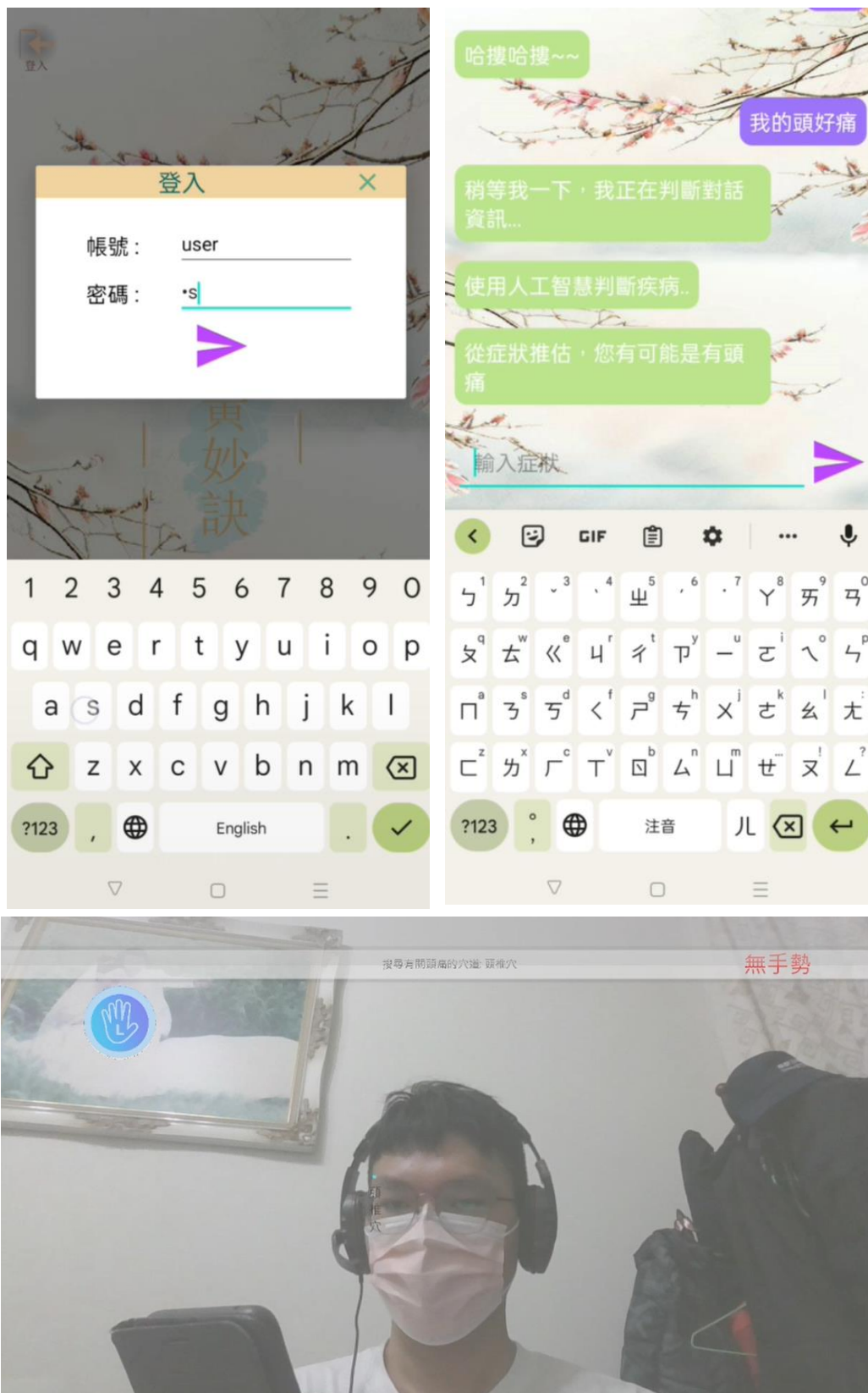


4.3 系統整合測試

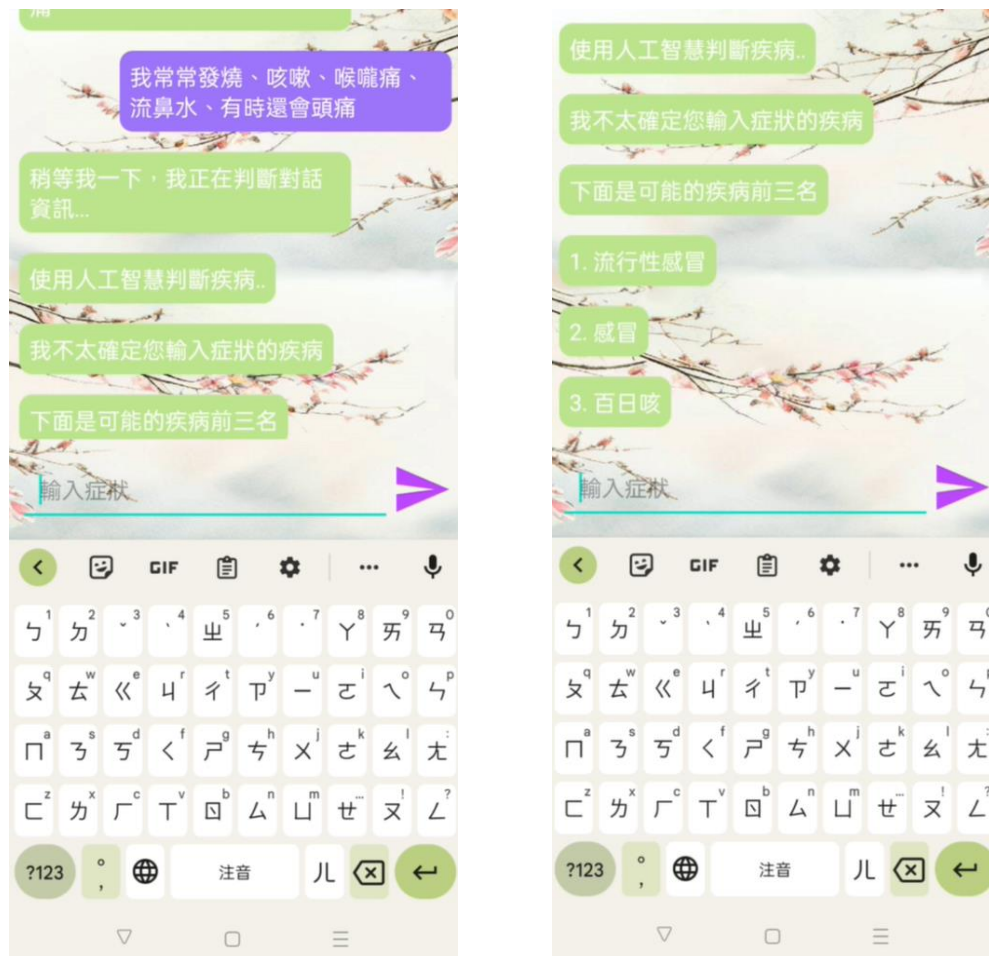
系統整合測試主要是展示整個系統是否有正常運作，以及連線是否有正常，設備之間是否可以互相通訊，下圖是展示當手機 APP 按下症狀選擇時，軟體端會收到訊息，並只顯示頭痛相關的穴位



使用者可以進行 AI 問診，手機會先和 Rasa Server 進行 query，得到對應的資訊後，再透過 TCP 傳輸給軟體端，讓螢幕顯示對應的特效穴位



當 Rasa Server 收到訊息，但是不確定使用者輸入症狀的疾病時，Rasa Server 會傳回這些症狀有可能的疾病前三名



(五) 結論及未來展望

5.1 結論

本計畫開發嵌入式系統，把技術溶入到鏡子，讓使用者可以透過一面鏡子就可以觀看穴位，以達到商品化。我們結合MediaPipe實現擴增實境的呈現，並使用BERT模型進行疾病對話推斷。具體而言，本計畫之預期結果如下：

- (1)建立演算法，讓穴位映射到對應的位置。
- (2)使用 DNN 達到手部姿勢辨識的簡單神經網路。
- (3)利用單面鏡減少 Jetson Nano J1010 需要的繪圖的複雜度，並會對圖像輸出控制，使用非同步輸出，以達到使用者流暢度的提升。
- (4)提出一種基於 BERT 的疾病分析技術，討論本專題對於疾病推斷的準確度，

和應用成果。

- (5)本專題未來會加入更多模型訓練，並反饋到資料集，期望本專題的資料集可以延續使用，並讓系統可以造福更多使用者。

5.2 未來展望

未來，本系統可以進一步整合更多的診斷技術和健康管理功能，以提供更全面的健康管理方案給使用者。例如，可以加入病歷管理功能，讓使用者能夠輕鬆地保存和管理自己的病歷資料，包括就醫紀錄、檢查結果、用藥記錄等，以便日後查詢和分享給醫生。進一步的，本系統也可以提供飲食建議，根據使用者的健康狀況和需求，推薦適合的飲食方案，以促進健康飲食習慣。

此外，本系統可以進一步擴充資料庫，加入更多的醫生患者的對話數據，以提升模型的診斷準確性和完整性。透過不斷更新和擴充資料庫，使得模型能夠不斷學習和改進，提供更精準的診斷結果。除了個人使用外，本系統也可以應用在醫療機構中，幫助醫護人員更好地進行診斷和治療。例如，可以將本系統整合到醫院的電子病歷系統中，讓醫生能夠更方便地使用本系統進行診斷，提供更快速和準確的診斷結果，並支援醫護人員的臨床判斷。

未來本系統具有許多潛在的應用和發展空間，可以透過不斷的改進和擴展功能，為使用者和醫護人員提供更全面和便捷的健康管理和診斷工具。這將對改善健康管理和醫療診斷的效率和準確性產生積極的影響。進一步加強模型的訓練和優化，並考慮應用在實際的臨床實踐中，進行實際效果的驗證和改進。相信這將對改善健康管理和醫療診斷的效率和準確性產生積極的影響。

(六)參考文獻

1. John Aponte, Kelly Bienhoff, Bob Black, Freddie Bray, Zoe Brillantes, Stephanie Burrows, Diana Estevez, Juliana Daher, Jacques Ferlay, Marta GacicDobo, Patrick Gerland, , Philippe Glaziou, Lucia Hug, Kacem Iaych, Robert Jakob, Li Liu, Rafael Lozano, Mary Mahy, Colin Mathers, Ann-Beth Moller, William Msemburi, Mohsen Naghavi, Abdisalan Noor, Minal K. Patel and Danzhen You, WHO methods and data sources for country-level causes of death 2000-2019. Global Health Estimates Technical Paper WHO/DDI/DNA/GHE/2020.2, pp.3-5, 9-15, 26.
2. Alexander Kapitanov, Andrey Makhlyarchuk, Karina Kvanchiani, HaGRID -- HAnd Gesture Recognition Image Dataset, Jun 16 2022. [CrossRef][PubMed]
3. Baohua Sun, Lin Yang, Wenhan Zhang, Patrick Dong, Charles Young, Jason Dong, Michael Lin, Demonstration of Applications in Computer Vision and NLP on Ultra Power-Efficient CNN Domain Specific Accelerator with 9.3TOPS/Watt, 2019, pp.1213-1223.
4. Andreas S. Panayides, Amir Amini, Nenad D. Filipovic, Ashish Sharma, Sotirios

- A. Tsaftaris, Alistair Young, David Foran, Nhan Do, Spyretta Golemati, Tahsin Kurc, Kun Huang, Konstantina S. Nikita, Ben P. Veasey, Michalis Zervakis, Joel H. Saltz, Constantinos S. Pattichis, AI in Medical Imaging Informatics: Current Challenges and Future Directions, 2020, pp.1-15.
5. Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, Matthias Grundmann , MediaPipe: A Framework for Building Perception Pipelines, Jun 14 2019, pp.1-8.
 6. Google(n.d.), MediaPipe Holistic, Retrieved December 2 , 2022, from <https://google.github.io/mediapipe/solutions/holistic.html>.
 7. Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Oct 11, 2018 , pp.1~16.
 8. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, Jun 12 2017, pp.1-15.
 9. Tom Bocklisch, Joey Faulkner, Nick Pawlowski, Alan Nichol, Rasa: Open Source Language Understanding and Dialogue Management, Dec 15 2017, pp 1-9.
 10. Hugging Face (n.d.), BERT model pre-trained for Chinese , Retrieved December 2 , 2022, from <https://huggingface.co/bert-base-chinese>.
 11. UnityTechnologies (n.d.), Unity open project , Retrieved September 19 , 2022, from <https://github.com/UnityTechnologies/open-project-1>
 12. homuler, MediaPipe Unity Plugin , Retrieved September 19 , 2022, from <https://github.com/homuler/MediaPipeUnityPlugin>
 13. crownpku, Rasa NLU for Chinese , Retrieved September 25 , 2022, from https://github.com/crownpku/Rasa_NLU_Chi