

國立金門大學

資訊工程學系
專題製作(研究)報告

互動式AR穴位解析—岐黃妙訣

Interactive system of acupoint analysis based on
AR

指導老師：趙于翔

專題組員：魏仲彥、徐伯元、莊汶娟、李偉德、鄭智陽

中華民國 112 年 6 月 5 日

國立金門大學資訊工程學系

專題製作(研究)報告審定書

學生： 學號: 110910541 姓名: 魏仲彥

學號: 110910542 姓名: 徐伯元

學號: 110910540 姓名: 莊汶娟

學號: 110910514 姓名: 李偉德

學號: 110910539 姓名: 鄭智陽

本專題研究報告內容符合本學系大學部畢業專題實作標準。

指導教授: 趙于翔

系主任: 馮玄明

中華民國 112 年 06 月

國立金門大學資訊工程學系
專題(製作)研究成果版權授權書

本授權書所授權之成果，為本組成員（以下簡稱本人）

學生 1： 魏仲彥 學號： 110910541

學生 2：徐伯元 學號：110910542

學生 3：莊汶娟 學號：110910540

學生 4： 李偉德 學號： 110910541

學生 5： 鄭智陽 學號： 110910539

在金門大學資訊工程學系111學年度第2學期，完成之專題製作(研究)成果。

專題研究名稱：互動式 AR 穴位解析—岐黃妙訣

學生本人與指導老師，對於專題研究成果及內容精簡版，是否同意放置於資工系網頁內，公開陳覽展示。

同意 不同意

上述同意與不同意之欄位若未勾選，本人同意視同授權。

指導老師簽名：

學生簽名: _____ (親筆正楷)

學號: 110910541 (必填)

日期：中華民國 112 年 6 月 5 日

金門大學資訊工程學系專題成果展示申請表

姓名	魏仲彥	學號	110910541	年級	三年級
姓名	徐伯元	學號	110910542	年級	三年級
姓名	莊汶娟	學號	110910540	年級	三年級
姓名	李偉德	學號	110910514	年級	三年級
姓名	鄭智陽	學號	110910539	年級	三年級

專題實驗成果摘要：

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 與智慧鏡子藍芽連接，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。

- 繳交修正後專題報告書(含電子檔) 繳交網頁格式專題報告書精簡版電子檔
繳交修正後海報(含電子檔) 繳交成果發表授權書

申請人簽名：

指導老師推薦與建議：

簽名：

摘要

按穴道是保養身體的方法之一，其中包含許多具有治癒特效的穴位，不僅能針對單一疾病產生治療效果，還可以調整全身機能，然而，穴位分佈於人體各處，其確切位置通常需要透過專業書籍或專家的協助才能確定。為了協助使用者更輕鬆地定位手部與頭部穴位並了解其功能，以便隨時緩解症狀並預防疾病，我們團隊開發了一套互動式 AR 穴位保健系統。

本系統以 Jetson Nano J1010 作為開發板，並運用 Mediapipe 技術和穴位映射演算法，顯示穴位到螢幕上。透過自定義 DNN 手部識別模型，實現了增強現實（AR）的穴位互動。使用者可以透過手勢，查看對應穴位點在身體上的位置，以及穴位簡述。此外，我們還開發了手機 APP，通過藍芽技術實現與系統的互動功能。使用者可以選擇想要了解的症狀，系統將顯示出緩解該症狀的穴位。本團隊也開發了 AI 問診，使用開源對話人工智能 Rasa，利用自然語言理解（NLU）判斷使用在手機 APP 上輸入的意圖，當判斷意圖為醫療相關內容，便會傳到 BERT 模型進行問診資訊分析，判斷可能的疾病或症狀，並提供相應的建議穴位來緩解症狀，有助於使用者對自身狀況進行初步判斷。所有的 AI 問診資訊都會傳送到伺服器端，醫師可以透過網站查看使用者的資訊。利用圖表化數據，醫師能夠快速了解使用者的狀況，進一步節省醫療資源。這種基於穴位保健的系統為人們提供了一個安全、方便、高效的健康保健解決方案，尤其對於缺乏經驗和醫學知識的人來說，具有重要的價值。

本作品的創新點在於結合手機 APP、AI 和 AR 技術，實現了基於穴位的健康保健解決方案。傳統的健康保健方式可能需要書籍或專家的協助來確定穴位的位置，而我們透過互動式 AR 穴位保健系統和手機應用程式的結合，使得使用者能夠直觀地了解手部穴位在身體上的位置和作用。這種創新的技術結合使得穴位保健更加便利和高效，為使用者提供了一個全新的自我保健方式。

關鍵字：穴位醫療、AR 互動、深度學習、AI 問診、Jetson nano、AIoT

致謝

專題是一項非常具有價值的學習，從決定專題方向、工作分配到實際操作，各個環節都令組員大大成長，並從中學習到技術與人與人之間的溝通能力。我們團隊在這邊要先感謝指導教授，從最初的專題構思，以及跨領域結合的概念，都是趙教授細心監督與耐心協助，我們才可以慢慢從廣大的海量題目中選出最具有價值的專題方向，並帶領我們團隊打入第 27 屆大專校院資訊應用服務創新競賽，並取得兩個佳作的佳績。趙老師也建議我們軟硬結合，把專題作品商品化，雖然起初在軟、硬體結合上遇到很多不相容的問題，但最後我們還是努力從底層開始寫起，捨棄 Unity，從 pyQT 出發，最後成功製作出 AIoT 的產品，並在學期末時，打入第二十三屆旺宏金矽獎總決賽，淘汰掉 284 支優秀隊伍，成為最後入圍《應用組》總決賽前九強的隊伍。感謝趙老師對團隊的付出。

在專題製作過程中，每位組員都有其負責的位置，但往往因為時間不允許或是技術跟不上，而無法達到組長的要求。我們在專題的分工上，學習到了很多，尤其在有限時間內，要完成一個大型系統，實屬不易，我們團隊最後使用切分專題的方式，把大的部分切為很多細項，讓組員能夠一一處裡，如果技術上遇到麻煩，再轉交組長，讓大家能夠很好的參與到專題裡。

感謝魏仲彥同學整理所有專案，從文書處裡、專案報告、UI 介面設計、到伺服器架設、深度學習技術處裡都可以在規定時間內完成，並協助他人沒完成的專案困難內容。感謝徐伯元同學負責架構軟體端，並把細部結構模塊化，從穴位映射技術到 Mediapipe 的架構處裡，都能很完善的處理，並把效能做到最好。感謝莊汶娟在手機應用程式上的付出以及介面設計的努力，從零開始學習 android studio 的技術，以及 Kotlin 語言非常艱辛，但最後還是成功做出成果。感謝李偉德同學在資料庫與硬體外觀上的付出，透過把書籍裡面的資料，寫入到資料庫裡，並在最後嘗試 3D 列印外觀和包裝外觀。感謝鄭智陽同學在網頁上做出的努力，雖然資料庫連接的部分沒有完成，但整體介面做的很完整，讓後續修改可以不用費太大的心力。在專題當中，所有組員都有成長與進步，再次感謝大家對於專題的付出以及努力。

目錄

內容

摘要	I
致謝	II
目錄	III
圖目錄	V
表目錄	VII
第一章 緒論	1
1.2 研究動機與目的.....	1
1.2.1 研究動機.....	1
1.2.2 研究問題.....	2
1.3 工作分配與進度甘特圖.....	2
第二章 背景知識與相關技術介紹	4
2.1 文獻探討與回顧.....	4
2.1.1 Mediapipe	4
2.1.2 pyQT	4
2.1.3 jetson nano	5
2.1.4 BERT	5
2.2 互動式 AR 穴位解析—岐黃妙訣內容簡介.....	7
2.3 MVC AND STATE MACHINE.....	7
2.4 穴位映射原理.....	9
2.5 AI 問診工作原理	10
第三章企劃研究方法及步驟	11
3.1 系統架構.....	11
3.2 企劃研究步驟.....	12
3.3 軟體端.....	13

3.3.1 state machine	13
3.3.2 AR 穴位映射.....	15
3.3.2 手部姿勢辨識.....	18
3.3.3 custom medapipe calculator	20
3.4 硬體端.....	20
3.4.1 產品架構與規格.....	20
3.4.1 硬體效能節省	22
3.5 手機端.....	24
3.5.1 藍芽連線.....	24
3.6 伺服器端	25
3.6.1 BERT 模型	25
3.6.2 醫師伺服器	27
第四章 系統實測	29
4.1 穴位映射實測.....	29
4.2 手機連線實測.....	32
4.3 AI 問診實測	34
第五章 結論及未來展望	35
5.1 結論.....	35
5.2 未來展望.....	36
參考文獻	37

圖目錄

圖 1.1 甘特圖 1	2
圖 1.2 甘特圖 2	3
圖 1.3 分工表	3
圖 2.1 JETSON NANO J1010	5
圖 2.2 BERT 架構圖	6
圖 2.3 MVC 架構圖	8
圖 2.4 STATE 內容簡述	8
圖 2.2 穴位映射架構圖	9
圖 2.3 穴位映射示意圖	9
圖 2.4 AI 問診示意圖	11
圖 3.1 系統架構圖	11
圖 3.2 系統流程圖	12
圖 3.3 FINITE STATE MACHINE 狀態轉移圖	13
圖 3.4 FINITE STATE MACHINE 架構圖	14
圖 3.3 穴位映射展示	17
圖 3.4 MEDIPIPE HAND LANDMARK	17
圖 3.5 DNN 手部辨識模型	18
圖 3.5 DNN 手部辨識模型準確率與損失函數	19
圖 3.6 MEDIPIPE GRAPH	20
圖 3.7 產品架構圖	21
圖 3.8 智慧鏡子設計圖	22
圖 3.9 DNN 頭部穴位模型架構	22

圖 3.10 DNN 頭部穴位模型準確率與損失函數.....	23
圖 3.11 手機藍芽連線.....	24
圖 3.12 AI 問診架構圖.....	25
圖 3.13 BERT LOSS FUNCTION.....	26
圖 3.14 BERT MODEL METRICS.....	27
圖 3.15 網頁查看問診紀錄.....	28
圖 3.15 網頁查看特定用戶問診資訊.....	28
圖 3.15 網頁查看所有用戶問診資訊.....	28
圖 4.1 穴位映射效果展示	29
圖 4.2 呼叫選單	30
圖 4.3 左右手勢控制 UI	30
圖 4.4 選擇手勢 AR 互動	31
圖 4.5 手機藍芽連線.....	32
圖 4.6 利用手機查看穴位分類及穴位詳情	33
圖 4.7 手機 UI 介面展示	34
圖 4.8 手機 AI 問診.....	34
圖 4.9 伺服器問診資訊.....	35
圖 4.10 RASA BERT 伺服器	35

表目錄

表 3.1 穴位映射演算法	16
表 3.2 產品規格	21
表 3.3 DNN 模型運行時間比較	23
表 3.4 BERT 訓練資料	26

第一章 緒論

1.1 前言

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，透過自定義 DNN 手部識別模型，實現了增強現實（AR）的穴位互動，查看對應穴位點在身體的位置。我們還設有手機 APP 與智慧鏡子藍芽連接，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 研究動機與目的

本研究主要以運用穿戴裝置及物聯網的整合，本節主要探討與本研究相關變項之文獻與資料的統整，除了說明現今穿戴式裝置的發展，另透過整合物聯網系統的說明，讓讀者更能了解本篇論文的主要研究理論及實驗方法。

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 Python 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？

1.3 工作分配與進度甘特圖

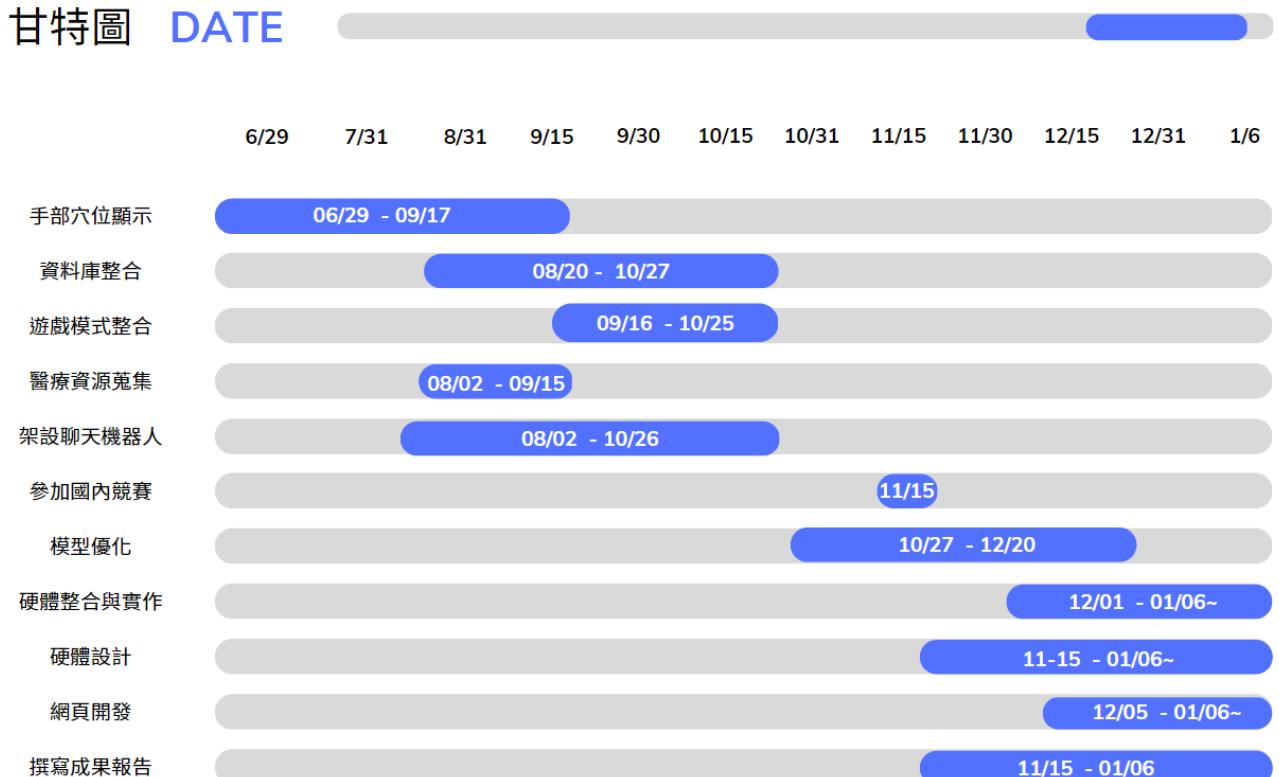


圖 1.1 甘特圖 1

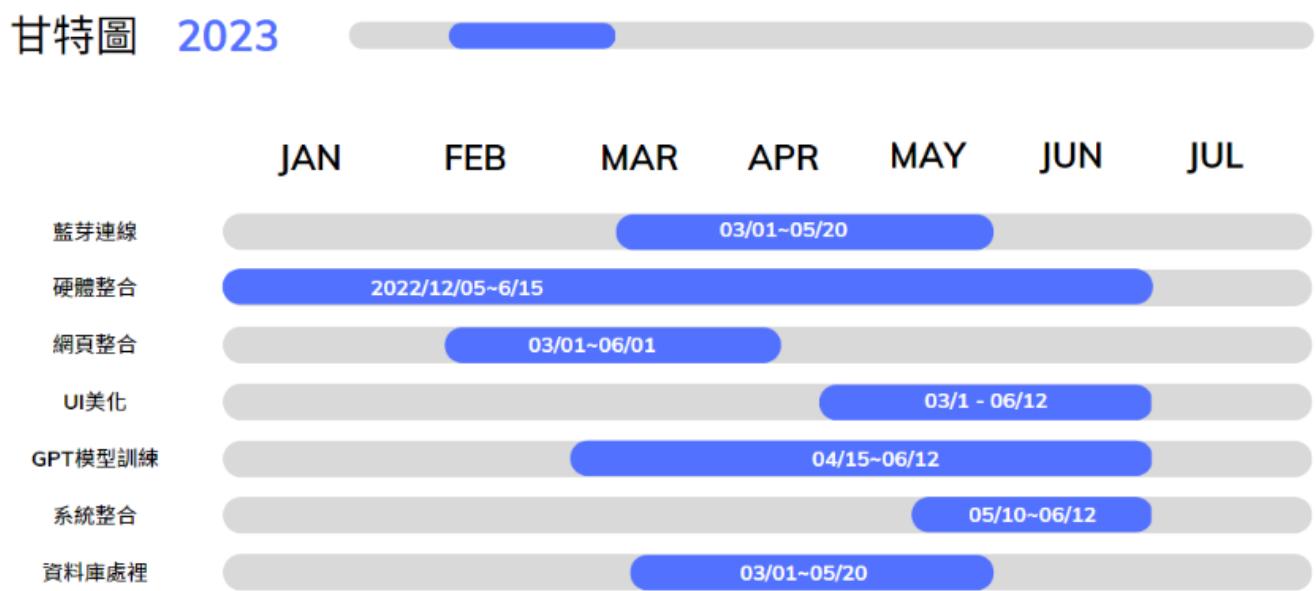


圖 1.2 甘特圖 2

分工表



圖 1.3 分工表

第二章 背景知識與相關技術介紹

穴位是中醫保健的重要元素，近年來人工智慧技術在穴位辨識上有了長足的進步。早期使用 CNN 模型和數學運算[13]或是邊緣偵測[14]推斷穴位，而現在可以取得人體座標點的模型，如 Mediapipe[5]。過去有論文使用 3DMM 模型來達到臉部穴位的顯示[16]，但其複雜的計算需求限制了它實時運行的可能性。Mediapipe Face Mesh 是一個實時面部標誌檢測系統，使用機器學習來預測面部 468 個關鍵點的 3D 位置，相比 3DMM 模型，Mediapipe 效能更好，適用於各種設備上運行。近年也有研究使用 Mediapipe 做臉部穴位顯示[17]，但功能較為單一。本論文在 Mediapipe 基礎上進一步開發，實現了不僅有臉部穴位顯示，還包括手部穴位、穴位分類、手勢互動等功能，讓使用者可以有更完善的體驗。

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 pyQT

PyQt[11]是一個用於開發圖形使用者介面（GUI）應用程序的框架。它是基於 Qt 框架，Qt 是一個跨平台的應用程序開發框架，可用於開發桌面應用程序、移動應用程序和嵌入式系統。PyQt 可以在多個操作系統上運行，包括 Windows、Mac OS X、Linux 和嵌入式 Linux 等，讓 jetson nano 的程序開發簡便許多，且由於 PyQt 基於 Qt 框架，使得它

具有良好的性能。通過優化和設計提供了高效的繪圖和事件處理，pyQT 能夠處理大量的數據和複雜的操作。此外，PyQt 提供了豐富的配置和自定義選項，讓我們能客製化獨特的用戶界面。

2.1.3 jetson nano

Jetson Nano J1010 開發板是一款由 NVIDIA 推出的高性能單板電腦，專為機器學習和人工智能應用而設計，Jetson Nano J1010 搭載 NVIDIA 的四核 ARM Cortex-A57 處理器和 128 核心 NVIDIA Maxwell GPU，具有強大的運算能力。這使得它在處理複雜的機器學習和視覺計算任務時表現出色，能夠實時處理高分辨率圖像和影像。Jetson Nano J1010 的 NVIDIA GPU 提供了額外的計算資源，可用於加速機器學習和深度學習任務。這使得在 Jetson Nano 上執行複雜的神經網絡模型成為可能，並且可以實現即時的本論文要做的臉部偵測與手部辦事。J1010 還針對人工智能和機器學習應用而優化的硬體平台。它支援流行的深度學習框架，如 TensorFlow、PyTorch 和 Caffe，以及 CUDA 和 cuDNN 等 GPU 加速庫。讓我們可以更有效率的使用自定義的模型。



圖 2.1 jetson nano j1010

2.1.4 BERT

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 2.1)，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螢幕上。

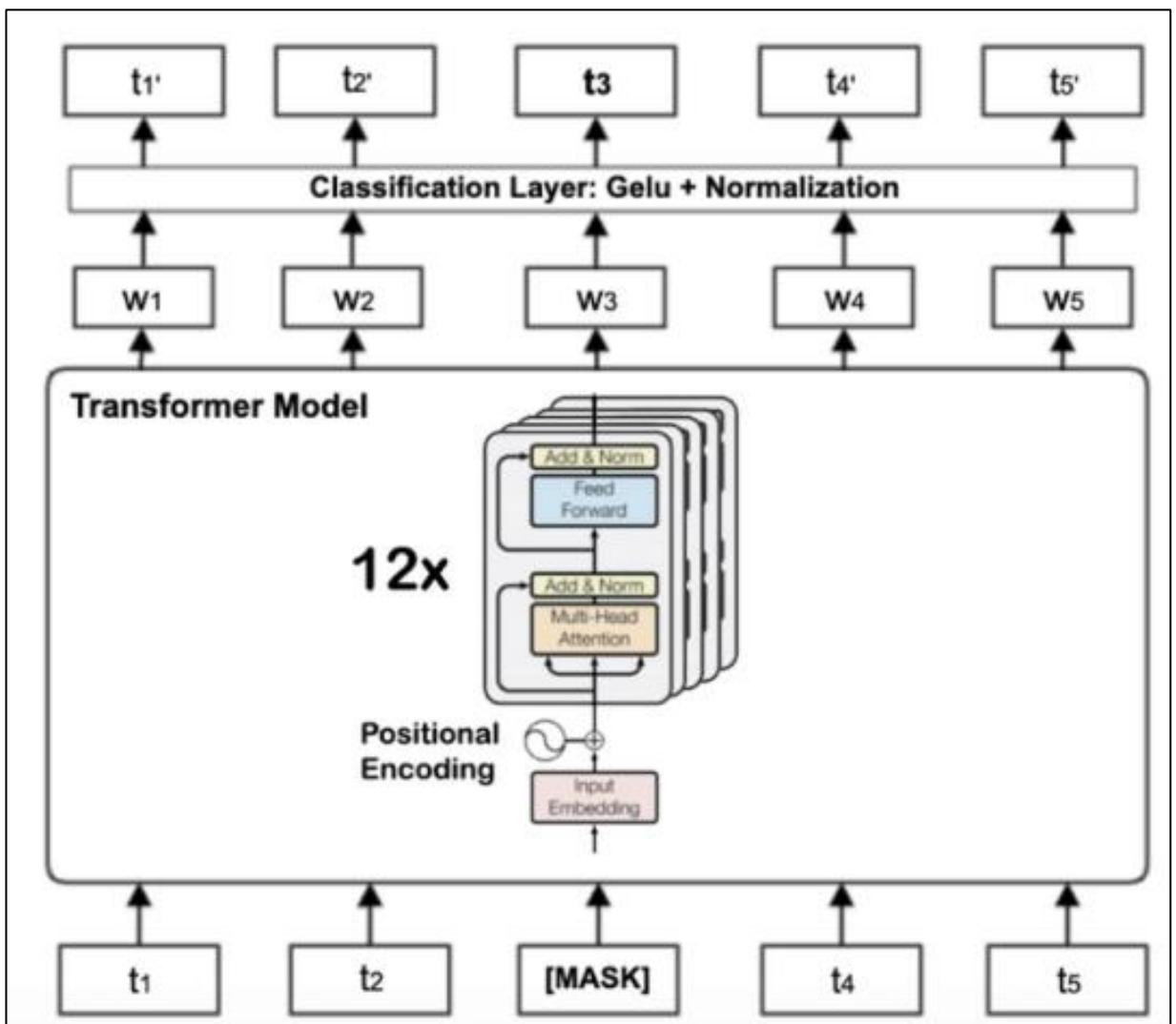


圖 2.2 BERT 架構圖

2.2 互動式 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、往左、往右、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 mediapipe 和穴位映射演算法，將穴位顯示在螢幕上，使用者可以直接在螢幕上查看頭部或手部穴位的簡述功能，增加了使用者與鏡子的互動性。
- (4) 手機連線：透過手機與硬體的連線，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，並查看穴位詳細資訊。
- (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 MVC and State Machine

本專案的軟體端使用 MVC(Model View Controller)架構，用於組織和分離應用程式的不同組件，以實現更好的代碼結構和可維護性。模型(Model)代表應用程式的數據結構、邏輯和狀態。它負責處理數據的操作、驗證、持久化等。模型不直接與用戶界面(View)或用戶交互(Controller)互動，而是通過提供接口來讓它們進行交互。視圖(View)負責呈現介面，讓使用者能夠直觀地看到和操作數據。它通常是用戶界面元素（如按鈕、表單、圖表等）的呈現，並根據模型的狀態進行相應的更新。控制器作為模型和視圖之間的中介，負責接收用戶的輸入、處理用戶的請求，然後根據需要將結果傳遞給模型或視圖。它根據應用程式的邏輯來決定如何更新模型和視圖。在圖 2.3 中可以看到我們的 MVC 架構，而在 controller 端，state machine(圖 2.4)中的 state 會根據需求來控制 UI

controller，讓用戶可以直接與介面進行交互。

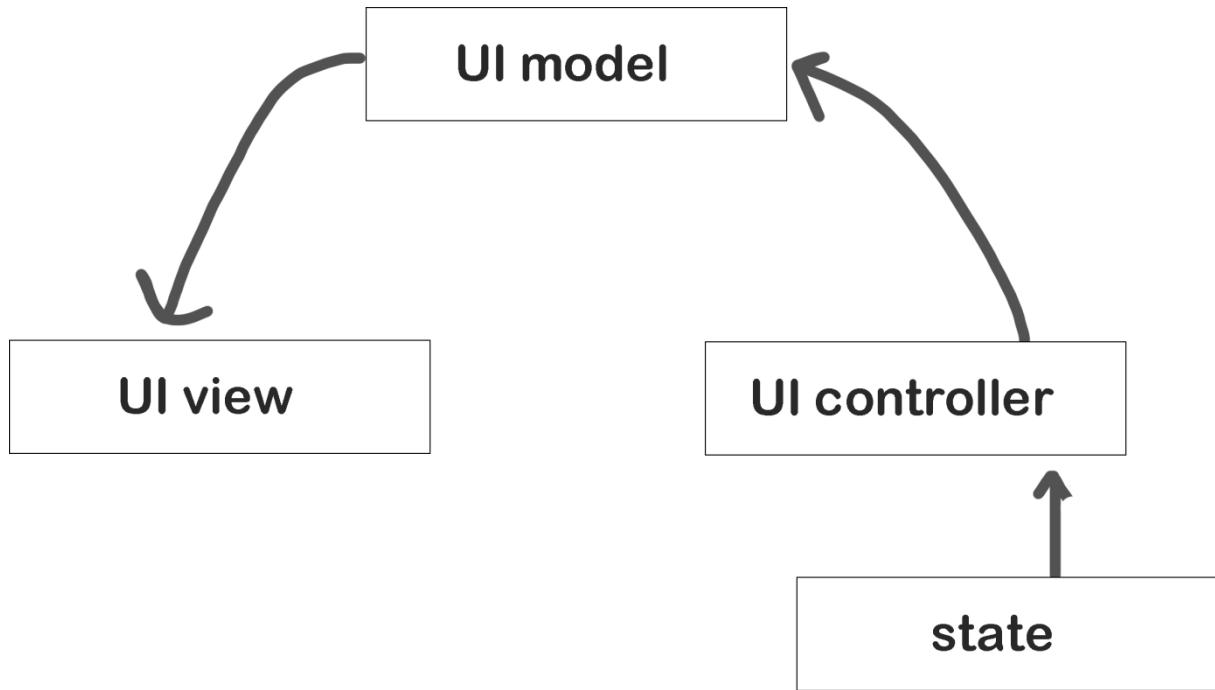


圖 2.3 MVC 架構圖

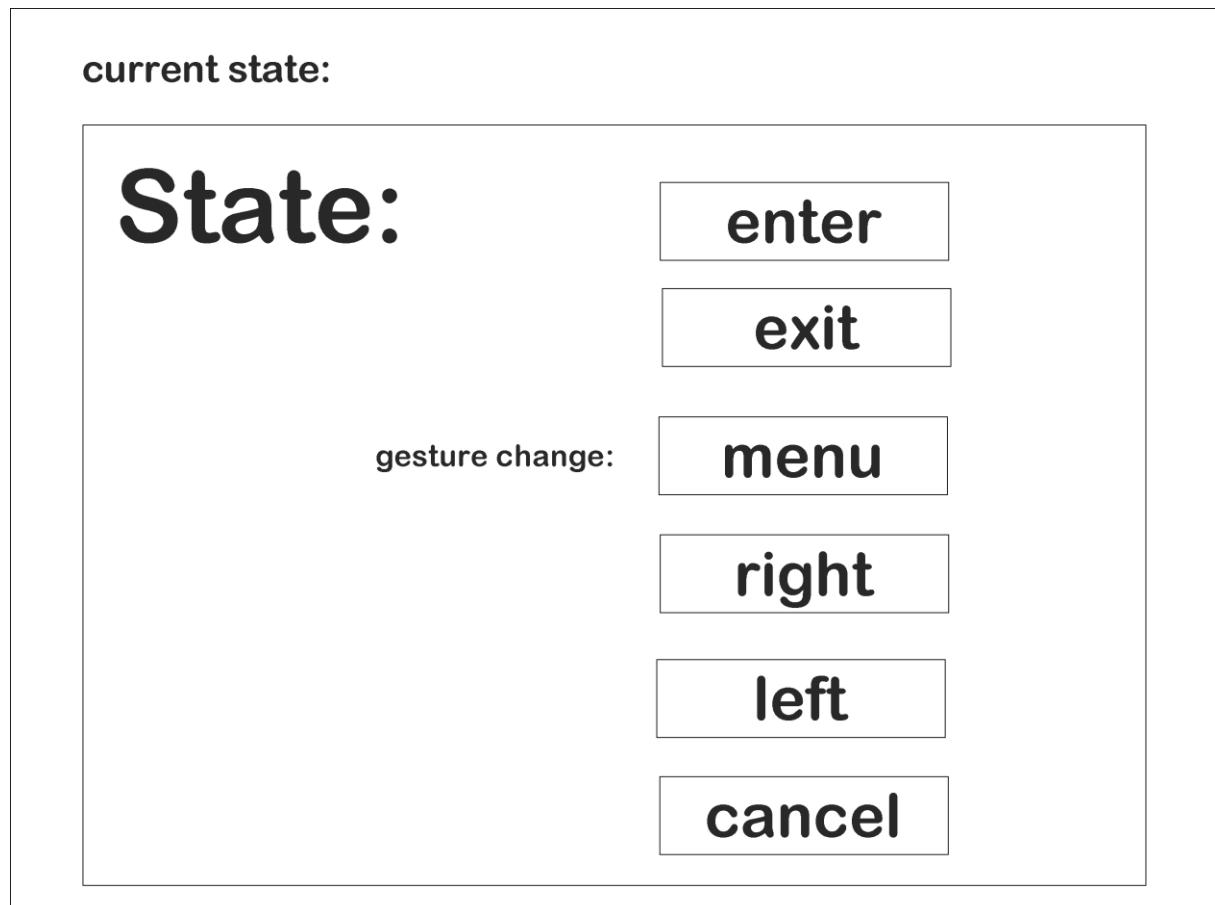


圖 2.4 state 內容簡述

2.4 穴位映射原理

手部定位以及頭部定位是利用資料庫的位置資料，裡面的資訊包括靜態資料與動態資料，靜態資料為參考 MediaPipe 座標位置、與座標位置的相對距離，以及穴位名稱，動態資料則是 MediaPipe 實時傳遞的座標資訊，穴位顯示的步驟可以從圖 2.2 中看到。我們的系統主要使用 MediaPipe 的 hand detections 與 face mesh 模型，這些模型可以幫助

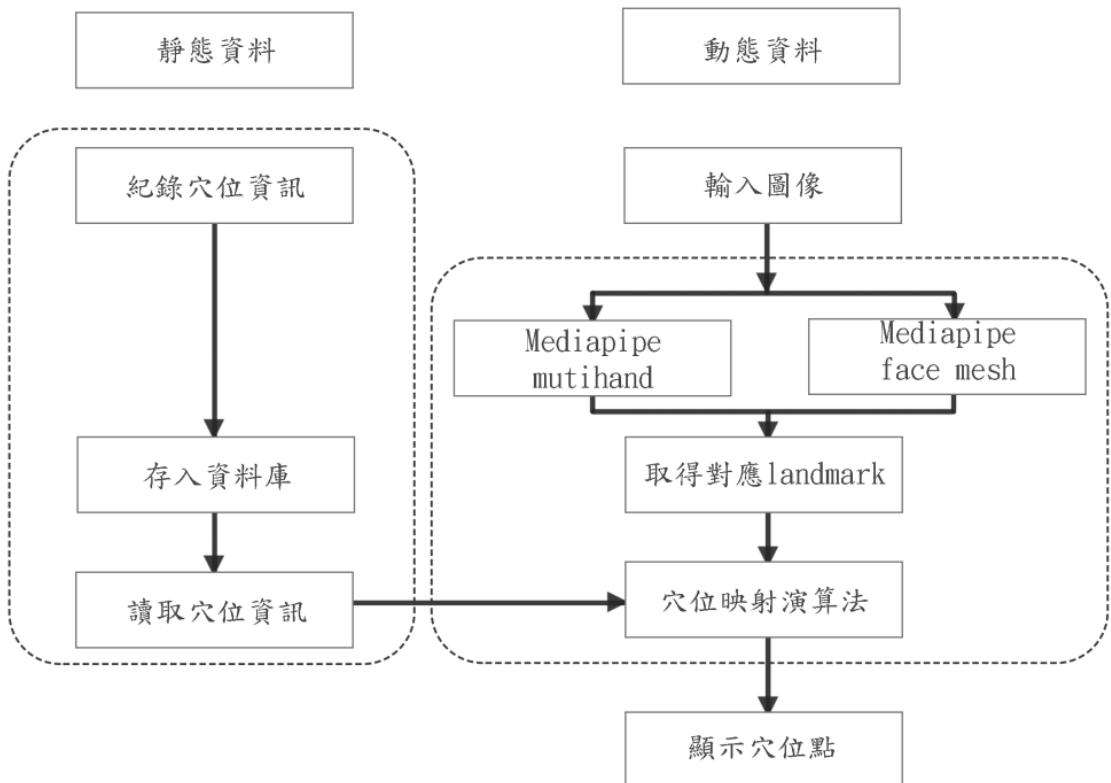


圖 2.5 穴位映射架構圖

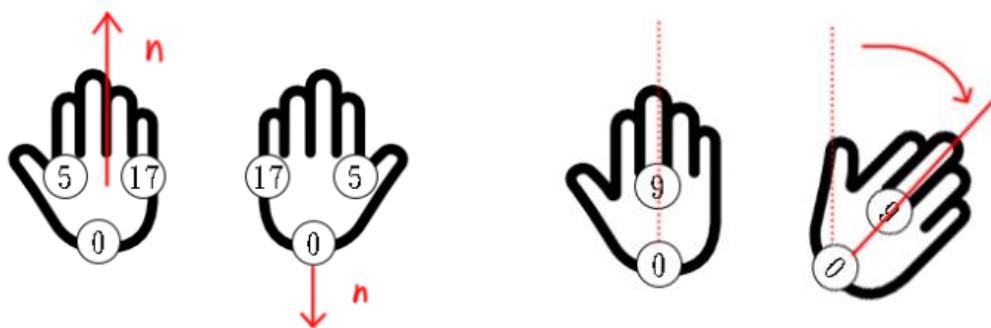


圖 2.6 穴位映射示意圖

我們偵測人體的手部和頭部的位置。透過模型，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。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、往左、往右、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。
- (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。而對於頭部穴位，我們則是使用相對位置坐運算。透過 face mesh 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。

2.5 AI 問診工作原理

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[12]。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。

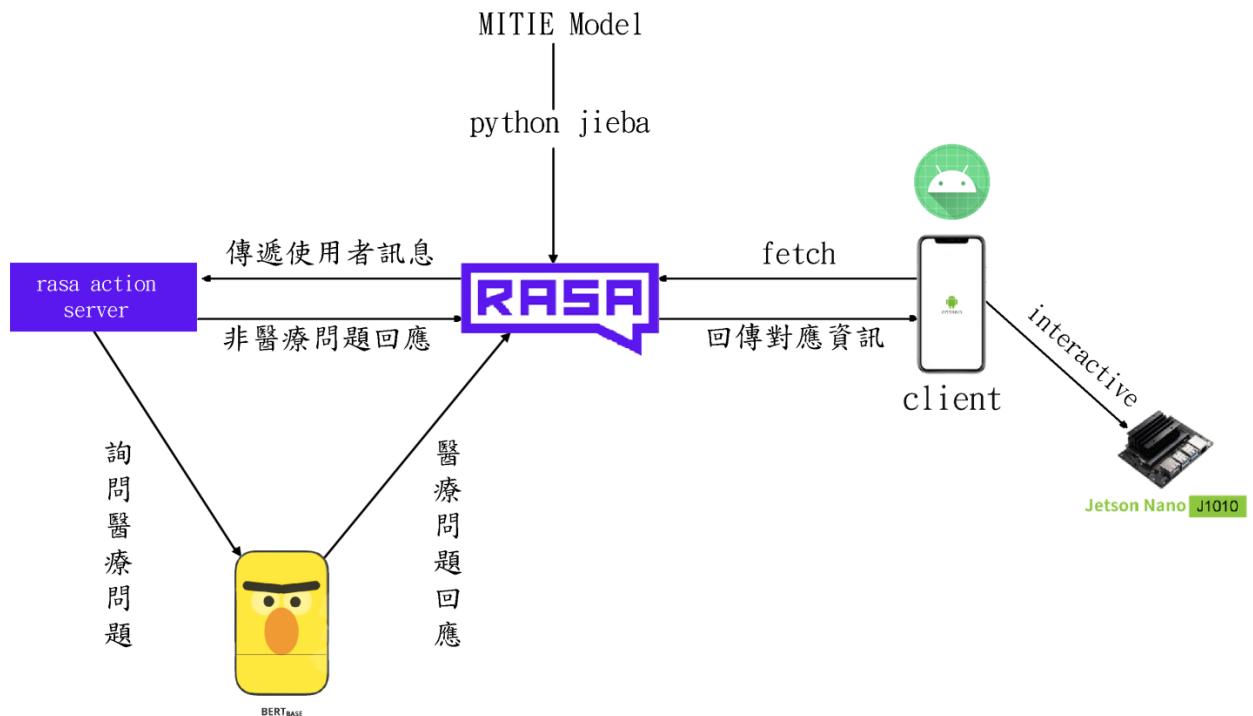


圖 2.7 AI 問診示意圖

第三章企劃研究方法及步驟

3.1 系統架構

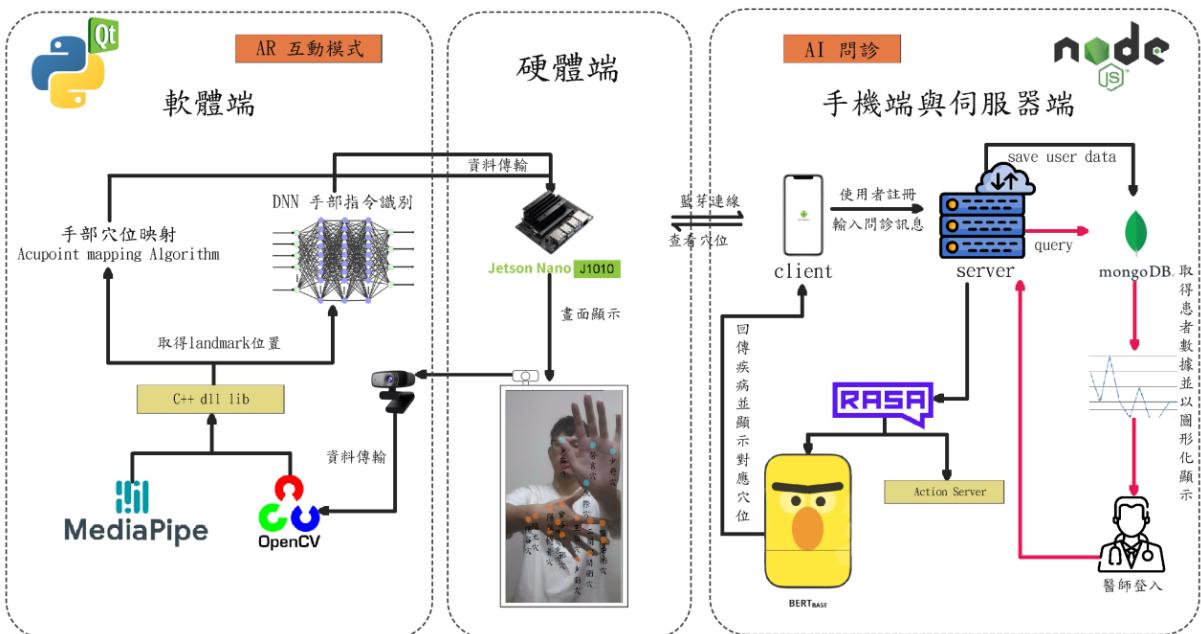


圖 3.1 系統架構圖

在這個計畫中，研發了一個互動式 AR 穴位保健系統，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。這個產品主要包括四個部分：軟體端、硬體端、伺服器端手機端。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 pyQT 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，讓螢幕可以顯示出穴位，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。

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，圖六是我們的系統流程圖。

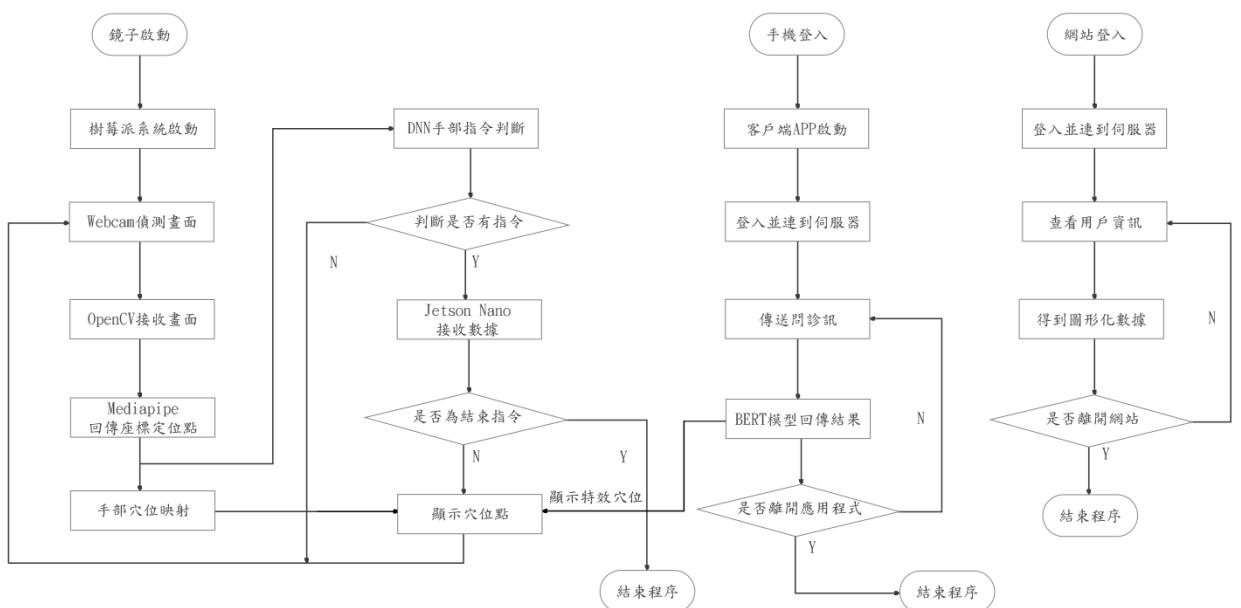


圖 3.2 系統流程圖

3.3 軟體端

軟體端負責處理運算邏輯、AR 穴位顯示邏輯、資料庫連接處裡、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一部分，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 state machine

因為本專案的軟體端有各種場景和不同的事件，所以我們需要 state machine 來接收對應的指令，並反映對應的事件，我們會根據使用者的手勢進行判斷，當作 state machine 的輸入，進行狀態的轉移。如圖 3.3，我們主要狀態有四個：introduction、Qrcode、menu 以及 Normal，introduction 包括穴位介紹以及系統功能簡介；Qrcode 則是與手機進行 bluetooth 配對的場景；menu 是控制場景轉移以及穴位點的開關；Normal 是用來顯示穴位點的位置以及查看穴道資訊簡述的狀態，也是使用者一開始的進入點。

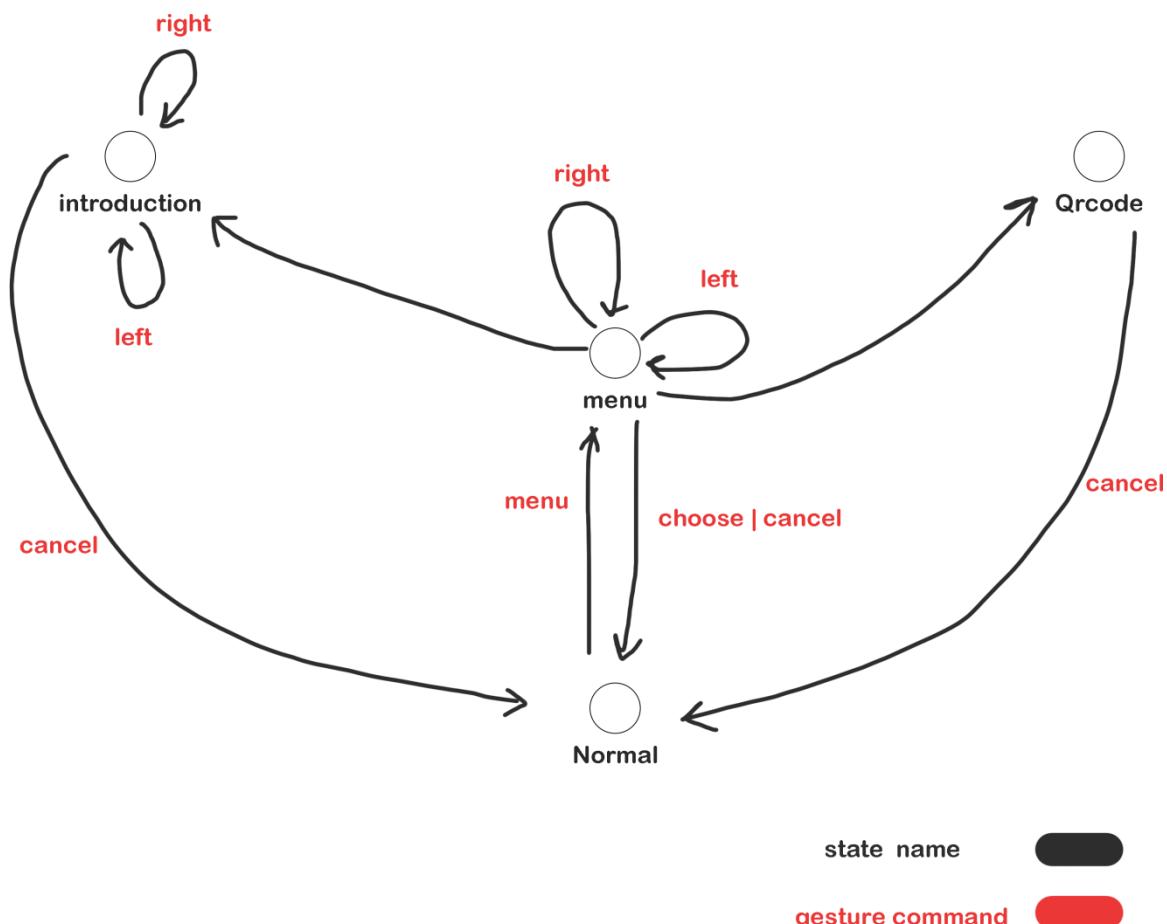


圖 3.3 finite state machine 狀態轉移圖

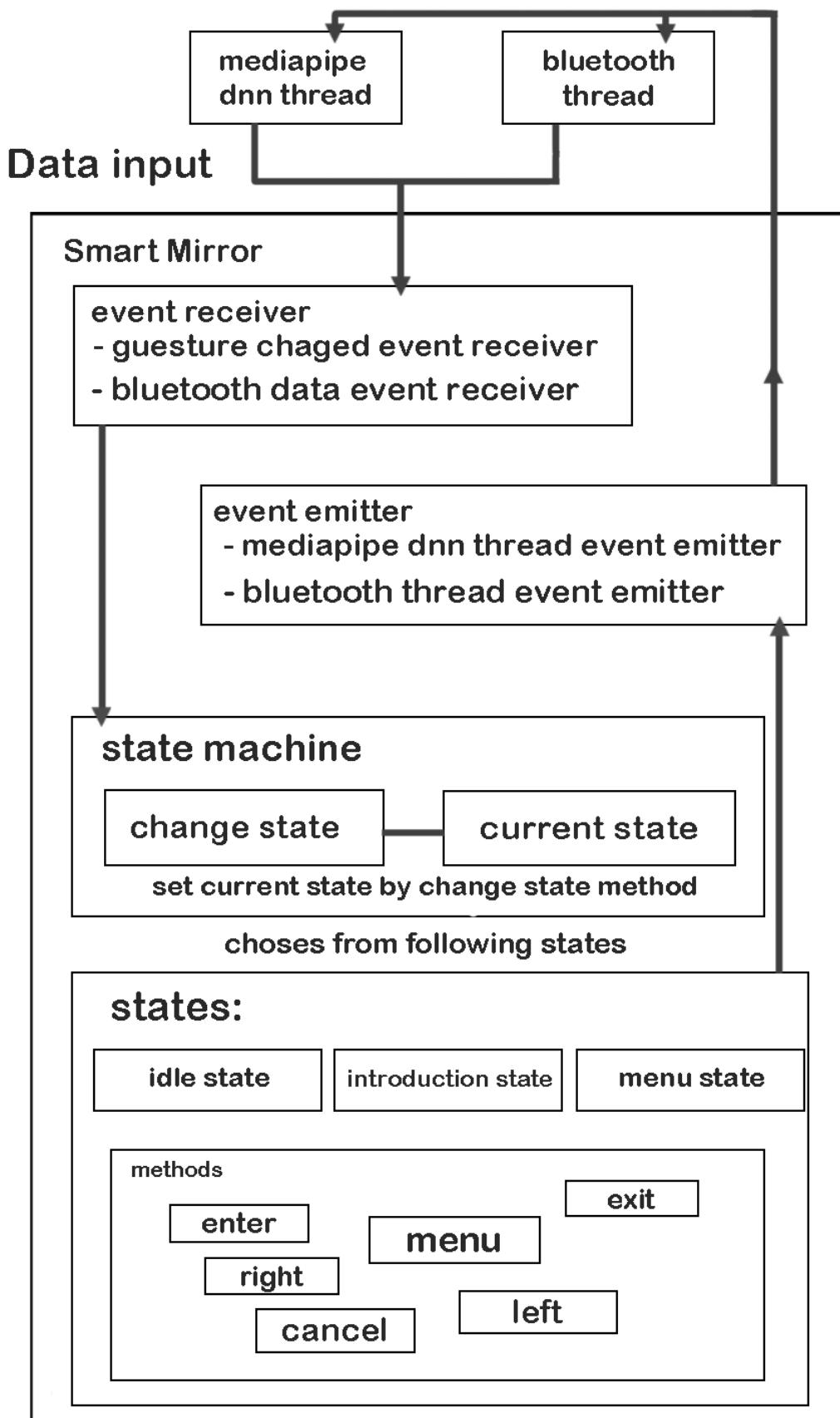


圖 3.4 finite state machine 架構圖

圖 3.4 代表了本專題軟體端程式碼中狀態機（state machine）的主要運行邏輯。該狀態機的目的是根據 mediapipe 的 hand tracking model 模型返回的 landmark 數據作為輸入，進行一系列的操作和決策。下面將對這個運行邏輯進行更詳細的說明。

首先，系統會從 mediapipe 的 hand tracking model 模型中獲取 landmark 數據作為輸入。接下來，這些輸入數據會被傳遞到自定義的 DNN 模型中進行處理。DNN 模型的輸出將透過事件函數傳輸到主線程（smart mirror - main thread）。這樣的設計有助於防止不同線程之間的競爭情況（race condition）和臨界區段（critical section），確保數據的正確處理。

一旦 DNN 模型的輸出到達主線程，smart mirror 將對輸出進行判斷。這個判斷可能涉及基於預定義邏輯或條件的操作。根據判斷結果，state（狀態）將執行相應的方法。這些方法可能涉及更新鏡子的顯示內容、調整鏡子的設置或執行其他相關操作。如果符合特定的轉移條件，state 將告訴狀態機（state machine）執行狀態轉移。狀態轉移可能包括進入另一個狀態或執行不同的操作流程。此外，state 也可以通知 UI 控制器切換 UI 界面。這樣可以根據系統的需求和狀態的變化來改變螢幕的顯示。state 也可以發送事件給其他線程。這些事件可能觸發其他線程中的特定操作或執行特定任務。例如，這些事件可以用於控制模型的開關，通知相關線程執行模型相關的操作。

3.3.2 AR 穴位映射

Algorithm 1 Acupuncture Point Mapping Algorithm

1. procedure HandAcupuncturePointMapping(Landmarks, database, Handness, Scale)
 2. v1 = (HandLandmarks[5] position-HandLandmarks[0]position).normalized
 3. v2 = (HandLandmarks[17]position-HandLandmarks[0]position).normalized
 4. c = crossproduct(v1,v2)
 5. HandAcupointsData[] currentAcupointsData
 6. **if** Handness is left hand && c.z axis value is bigger than 0 **then**
 7. currentAcupointsData = 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. currentAcupointsData = 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.        currentAcupointsData = 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 offest
27.        currentLandmark += offset * referenceLength * Scale
28.        Add currentAcupoint to AcupointsList
29.    until i == size of currentAcupointsData -1
30. end procedure
```

表 3.1 穴位映射演算法



圖 3.5 穴位映射展示

$$\varphi = \frac{\overrightarrow{v_{0to9}} \cdot \overrightarrow{nv_{0to9}}}{\sqrt{(x_0 - x_9)^2 + (y_0 - y_9)^2} \times \sqrt{(nx_0 - nx_9)^2 + (ny_0 - ny_9)^2}} \quad (1)$$

where $\overrightarrow{v_{0to9}} = (x_0 - x_9, y_0 - y_9)$,

$$\overrightarrow{nv_{0to9}} = (nx_0 - nx_9, ny_0 - ny_9)$$

$$[\alpha, \beta] = [x, y] \cdot \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} \times z \quad (2)$$

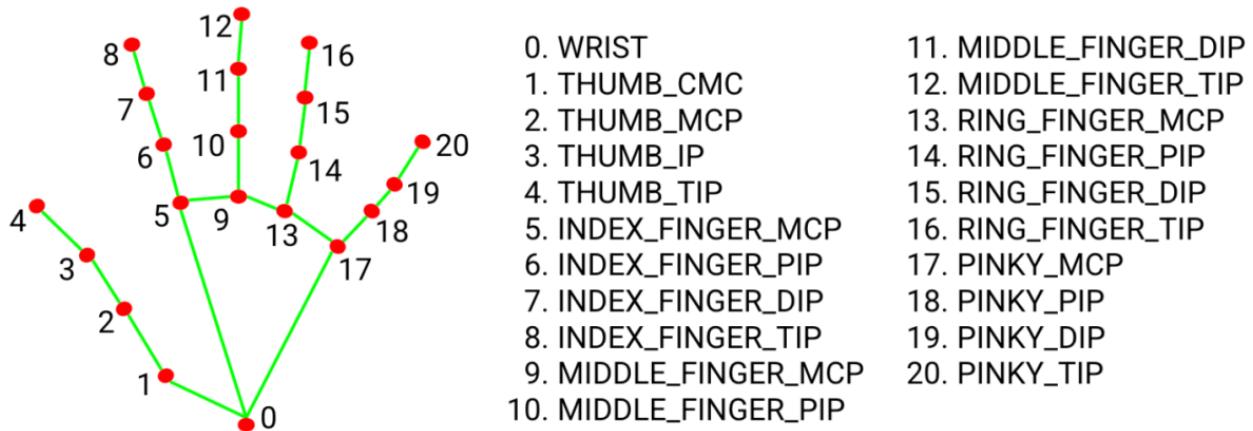


圖 3.6 Mediapipe hand landmark

在判斷手部正反面的的地方，我們使用自定義的手部穴位映射演算法(見表 3.1)，我們用外積、內積多種數學函(公式(1))，並會根據基軸的轉動量生成轉移矩陣(公式(2))，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 3.6)，通過這些技術的結合，讓穴位可以正確顯示在螢幕上(見圖 3.5)，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。

3.3.2 手部姿勢辨識

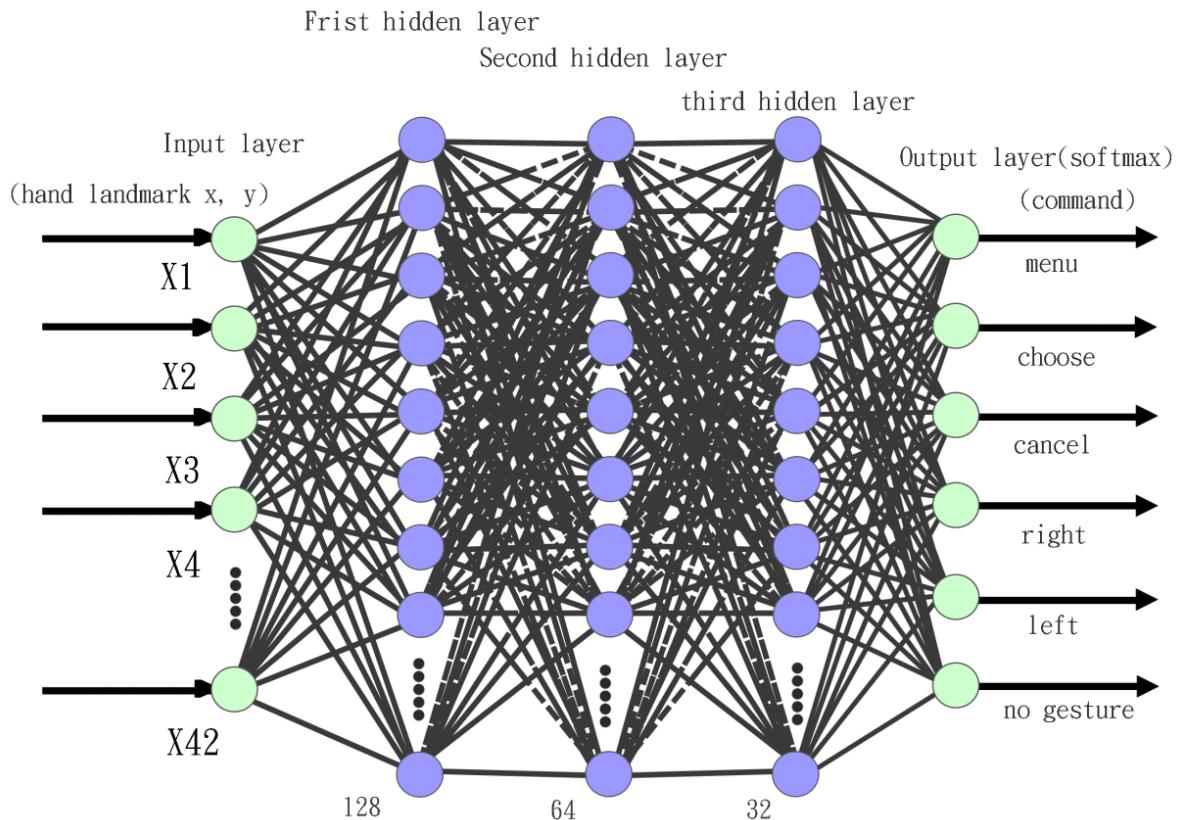


圖 3.7 DNN 手部辨識模型

HaGRID(HAnd Gesture Recognition Image Dataset) [2]是一個手勢識別圖像數據集，本計畫使用其中的資料以及自定義的手部資料進行 DNN (Deep Neural Network) 模型的訓練，資料數量一共 28304 筆。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、選擇手勢、取消動作、往右、往左以及無手勢。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。

$$\text{Loss} = - \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i \quad (3)$$

我們使用 DNN 來進行手部辨識，我們使用的 epochs 為 150，batch size 為 16，損失函數為 categorical crossentropy (公式 3)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的手勢，並根據訓練時所設定的六種類型（呼叫選單、取消動作、選擇選項、往右、往左、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 3.8 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。

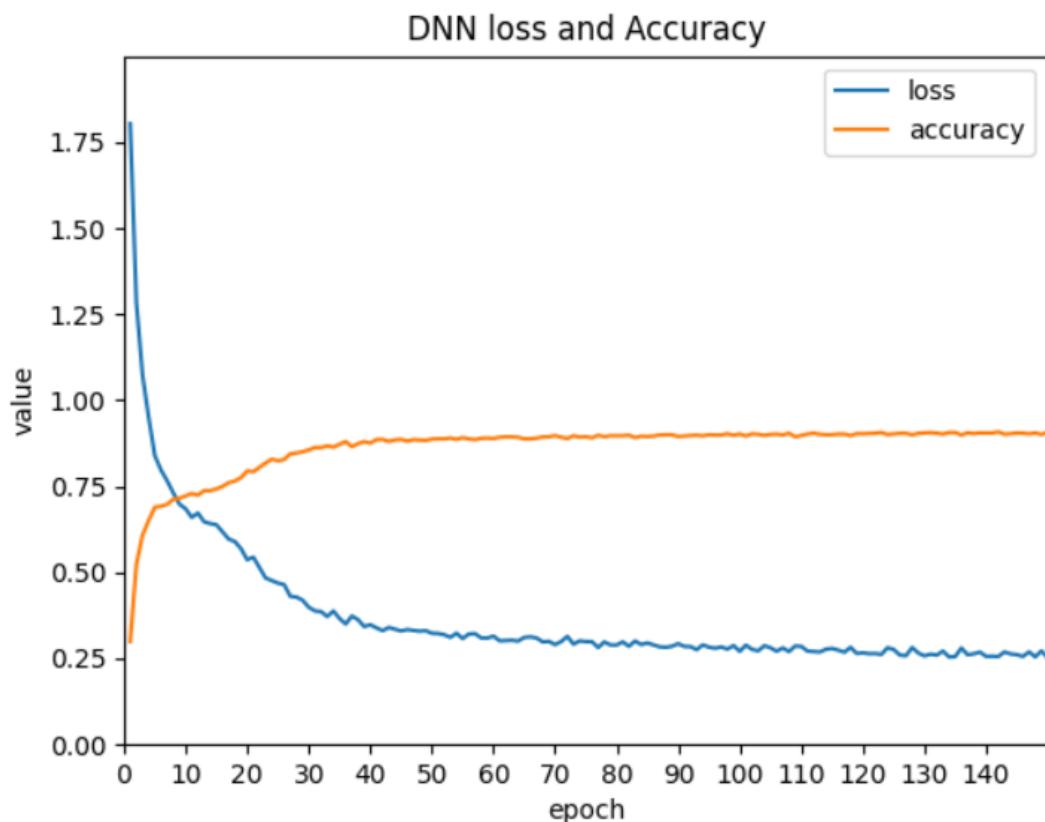


圖 3.8 DNN 手部辨識模型準確率與損失函數

3.3.3 custom mediapipe calculator

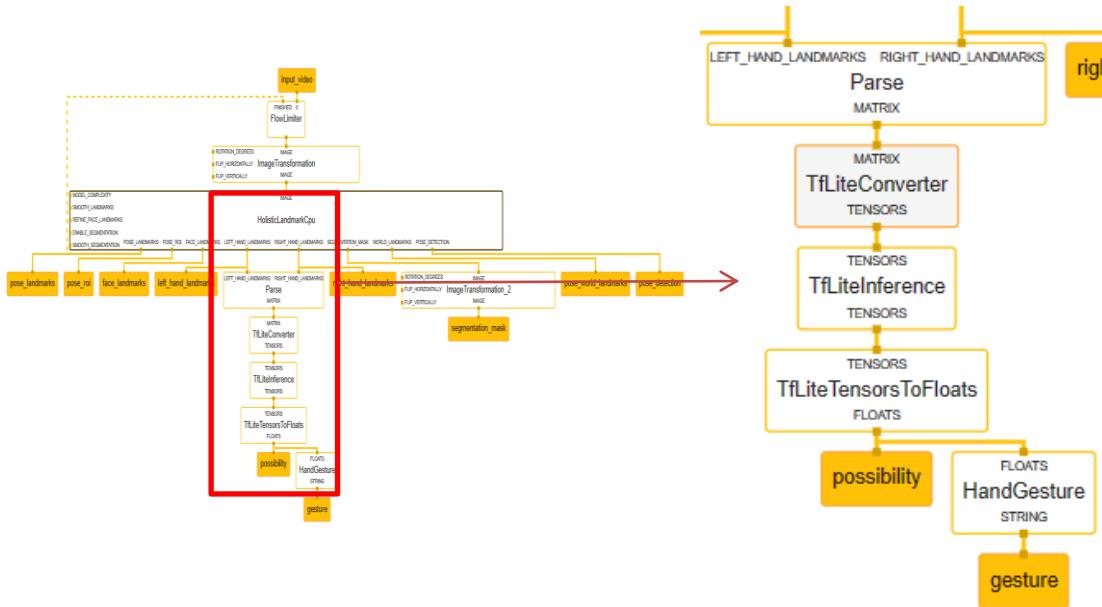


圖 3.9 Mediapipe Graph

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 hand detection landmark 的輸出左右手的 normalized_landmark_list 資料。我們把模型轉成 Tflite 的形式，並對運算結果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。

3.4 硬體端

3.4.1 產品架構與規格

本計畫的產品規格如表二，產品架構如圖 3.10 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。我們提供兩種臉部辨識模型，使用者可以選擇我們自訂義的臉部模型，讓系統效能提升。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。

產品名稱	Jetson nano j1010	21 吋液晶螢幕	logi webcam	Esense D704
------	-------------------	----------	-------------	-------------

產品圖片				
功能	程序運行開發版	顯示畫面	讀取人體資訊	與手機藍芽連線
操作溫度	-40~60 °C	-20~60 °C	-10~50 °C	-20~60 °C
價格	235 USD	80 USD	30 USD	25 USD
重量	3.5 kg			
硬體規格	GPU : NVIDIA Maxwell™ 架構配備 128 個 NVIDIA CUDA® 核心 CPU : 四核心 ARM® Cortex®-A57 MPCore 處理器 記憶體 : 4 GB 64-bit LPDDR4 儲存空間 : 16 GB eMMC 5.1 快閃記憶體			

表 3.2 產品規格

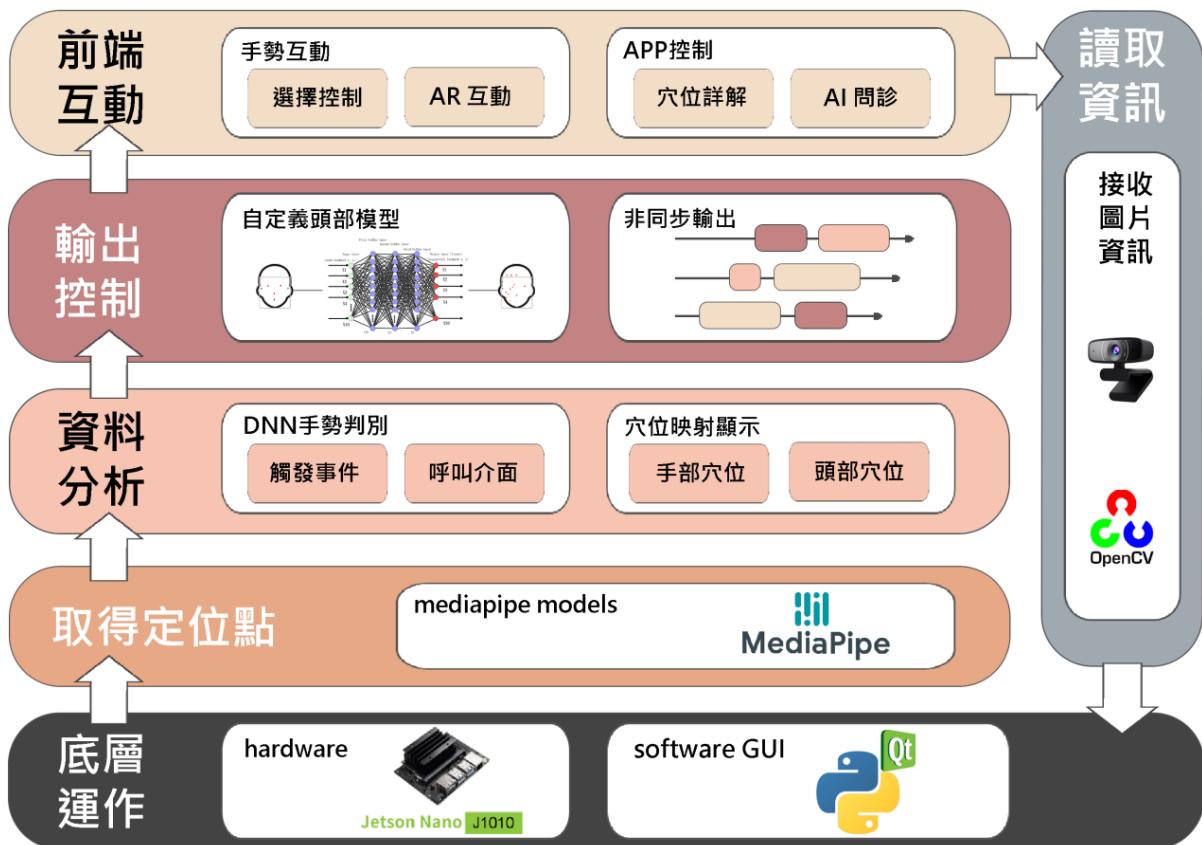


圖 3.10 產品架構圖

智慧鏡子的設計是基於普通家庭鏡子的模板進行延伸，我們利用 Webcam 的輸出，顯示到螢幕上，達到類似鏡子的效果，在螢幕後方有一個盒子，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，並且能隨時檢檢視自身的穴位，進行相應的健康管理和護理。

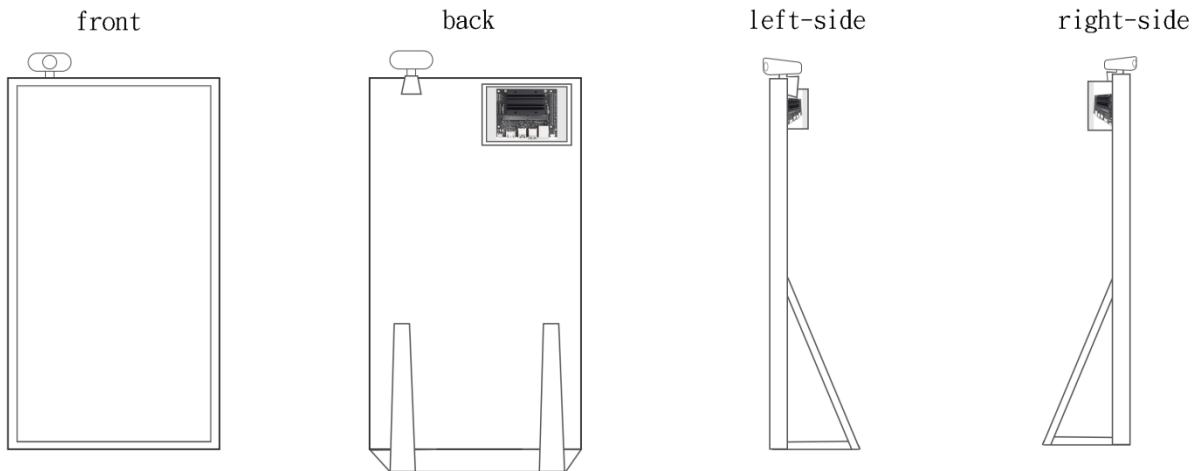


圖 3.11 智慧鏡子設計圖

3.4.1 硬體效能節省

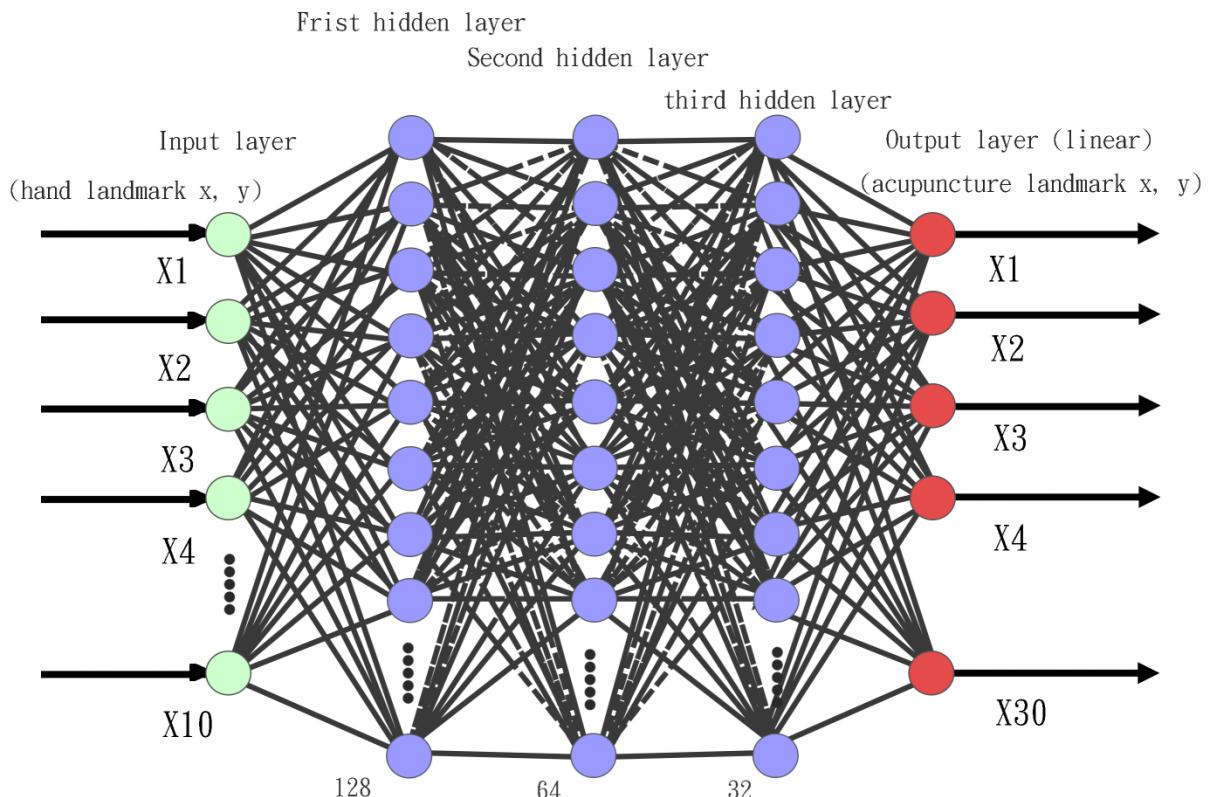


圖 3.12 DNN 頭部穴位模型架構

進行邊緣運算 (edge computing) 是指在裝置本地進行處理和分析資料，而不是依賴遠端伺服器。然而，進行邊緣運算可能會消耗大量的硬體效能，導致使用者體驗下降。為了改善使用者體驗，並達到即時反饋、低延遲，我們提供了兩種模型: Mediapipe face

box 模型和 Mediapipe face mesh 模型。我們透過在 face box 後面加上神經網路，讓模型可以輸出穴位。相較於 face mesh 模型，face box 模型在效能上消耗相對較少，face mesh 所消耗的效能約為我們訓練出來模型的 2.3 倍(表 3.3)。我們使用 face mesh 產生的穴位點座標資訊以及 face box 的座標點資訊來生成訓練集，然後使用深度神經網路(DNN)進行 150 個 epochs 的訓練，其中每個 batch 的大小為 16。在訓練過程中，我們使用均方誤差(MSE)作為損失函數(公式 4)。我們的 DNN 頭部穴位模型的準確度達到了高達九成(如圖 3.13)，這意味著我們的模型能夠將 face box 的穴位轉換為特定的頭部穴位點，同時還節省了效能。然而，如果硬體效能允許的話，我們仍然建議使用者使用 face mesh 模型，因為它能提供更加準確的穴位顯示。

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (4)$$

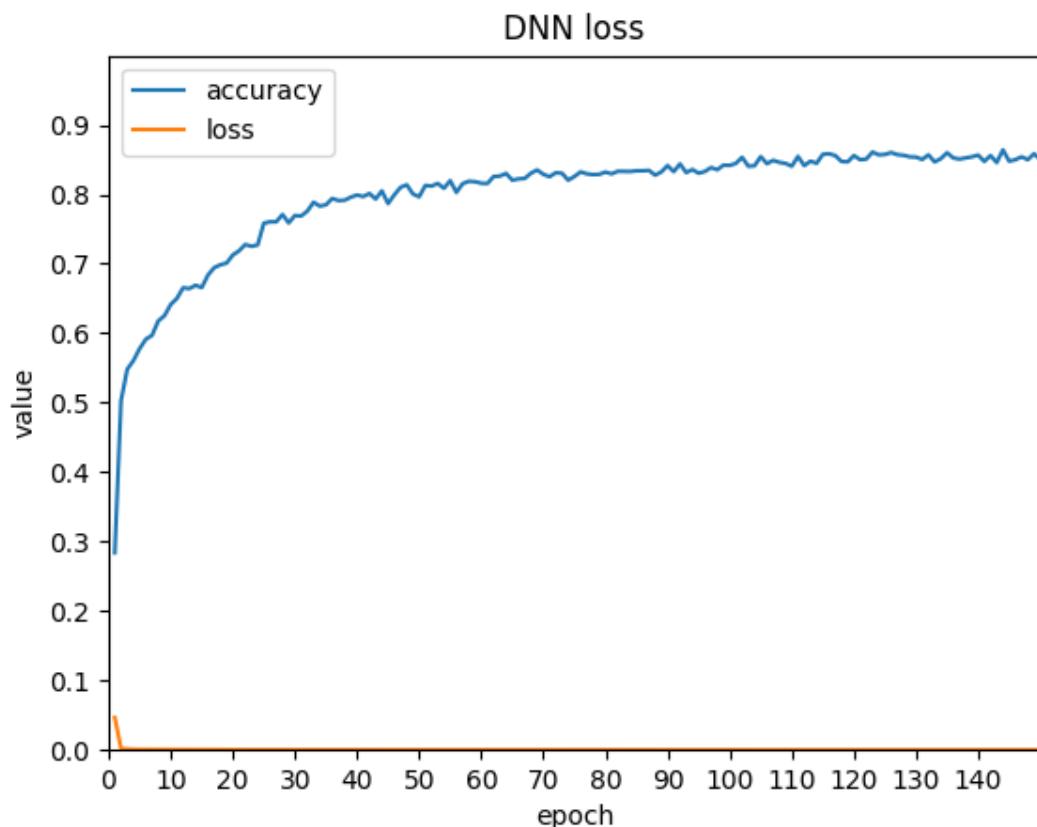


圖 3.13 DNN 頭部穴位模型準確率與損失函數

模型	Mediapipe face mesh	Mediapipe face box + dnn model
運行時間	1.96e-1	5.9e-2

表 3.3 DNN 模型運行時間比較

如果 Webcam 與 Mediapipe，這兩個過程在同一時間進行時可能會導致硬體資源的過度負擔，讓螢幕輸出影響使用者體驗。為了解決這個問題，我們對程式進行了非同步輸出處理。透過將 Webcam 讀取的資料和 Mediapipe 座標運算進行了分開處理，以減少對硬體資源的需求。Webcam 持續讀取視訊資料，而 Mediapipe 座標運算則在後台進行。這樣可以最大限度地利用硬體資源，提高應用程式的效能和響應速度，讓使用者可以享受到更流暢、即時的應用程式體驗，同時還能確保硬體資源的有效利用。

3.5 手機端

手機端透過藍芽連線，使用者可以將手機與智慧鏡子進行互動，查看想要按壓的穴位。也可以進行 AI 問診，AI 能夠根據使用者提供的症狀資訊，進行分析和判斷，並生成對應的疾病，供醫師參考。

3.5.1 藍芽連線

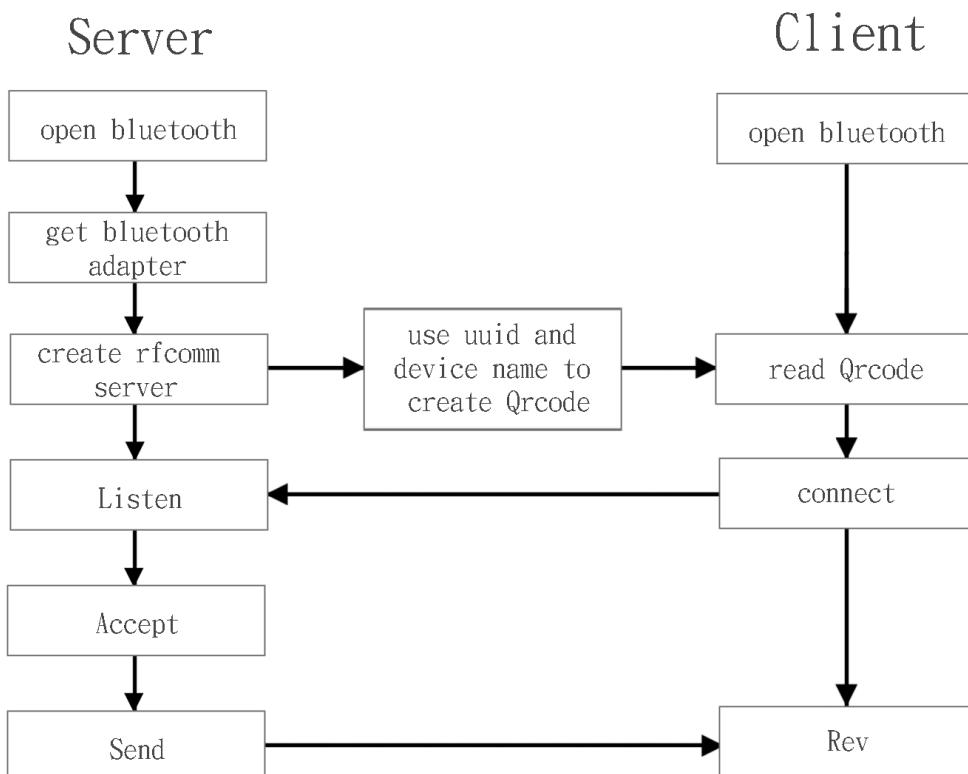


圖 3.14 手機藍芽連線

我們提供了手機端與軟體端之間的藍芽連線功能(如圖 3.14)。透過藍芽連線，使用者可以方便地查看所需的症狀對應的穴位，並且可以獲得所需穴位點的詳細資訊。當使

用者啟動藍芽連線功能時，手機端作為 RFCComm 伺服器會生成一個特定的 UUID，用於識別連線。這個 UUID 和設備名稱會被編碼成 QR code，並在手機上顯示出來。使用者只需將手機上的 QR code 放在 Webcam 的視野範圍內，Jetson Nano 會使用其內建的藍芽功能掃描並解讀 QR code，從而獲取設備的 MAC 和伺服器 UUID。藍芽連線的功能使手機端成為了一個方便的控制界面，使用者能夠輕鬆地操作和存取所需的穴位資訊。這樣的設計不僅提供了更直觀和互動性的界面，同時也增加了使用的便利性。

3.6 伺服器端

圖 3.15 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，當手機端的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，並讓智慧鏡子只顯示這些穴位，讓使用者可以進一步進行症狀舒緩。

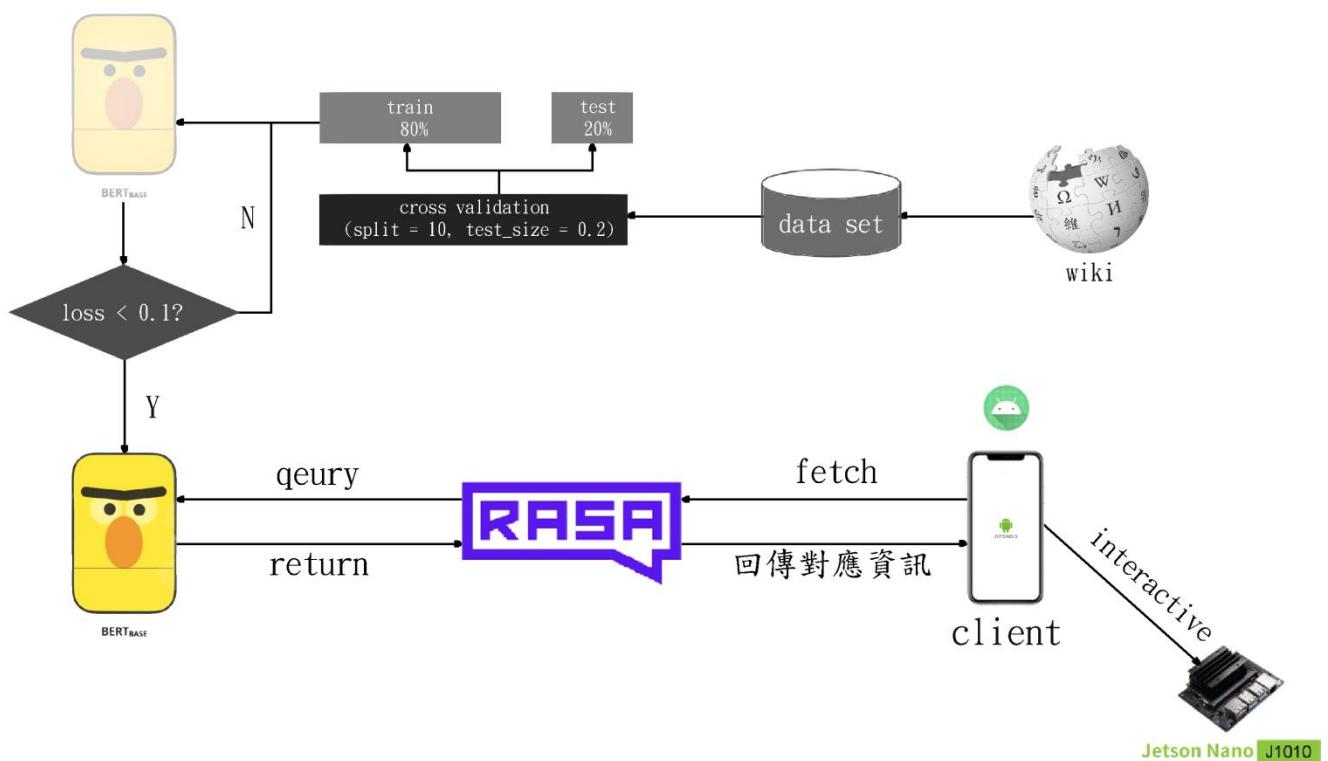


圖 3.15 AI 問診架構圖

3.6.1 BERT 模型

本作品使用了 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 3.4）。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 3.4 BERT 訓練資料

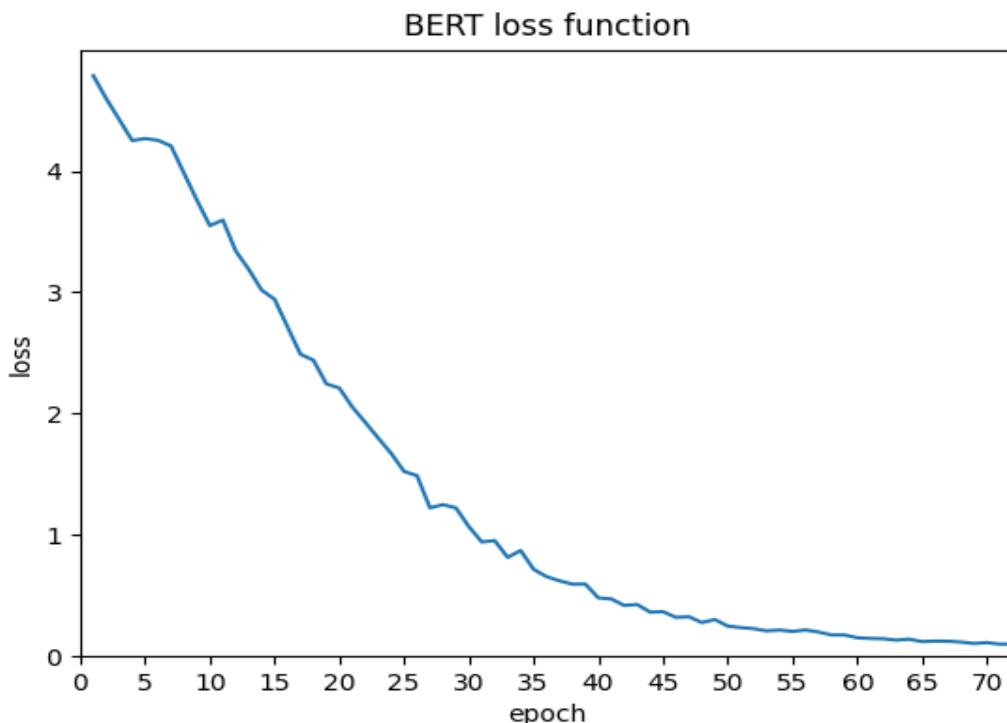


圖 3.16 BERT loss function

在訓練模型時，我們使用 epochs 為 72，batch size 為 10。透過觀察圖 3.16 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。

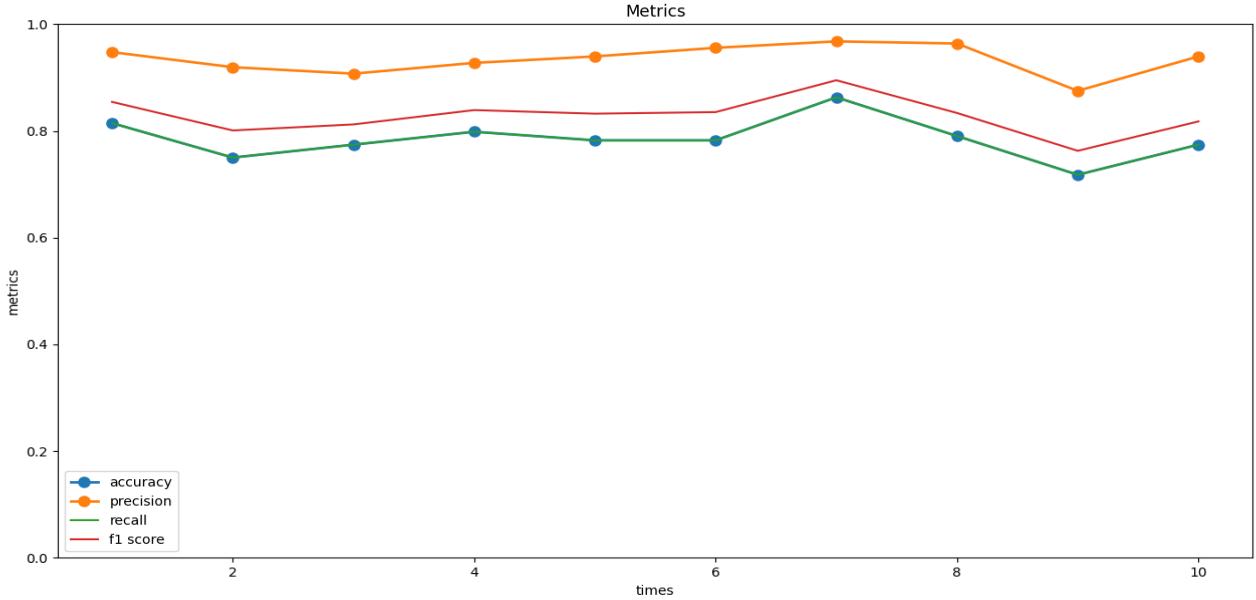


圖 3.17 BERT model Metrics

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 3.17 中的結果進一步驗證我們訓練的 BERT 模型在問診機器人中的有效性和可靠性。

3.6.2 醫師伺服器

AI 問診的結果會存到 Mongodb，醫師透過存取資料庫，查看病患數據，我們會將使用者症狀分類，把患者症狀資訊，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。我們的 AI 問診系統會將問診結果儲存到 MongoDB 資料庫中，有助於醫師更方便地存取和查看病患的數據。當使用者進行問診並提交症狀資訊後，系統會將這些資料整理並存儲到 MongoDB 資料庫中。我們的系統會對使用者症狀進行疾病的分類，並將其整理成圖形資訊。這些圖形資訊能夠提供醫師關於病患目前狀況的

快速概況，節省了醫師整理和分析資料的時間，並且提供了更直觀、清晰的視覺化呈現方式。



使用者名稱	問診內容	推斷疾病	問診日期
user	我的頭好痛	頭痛	2023/5/15 22:30:05
user	鼻子出血怎麼辦？	鼻衄	2023/5/17 23:15:20
user01	鼻子出血怎麼辦？	鼻衄	2023/5/17 23:15:25
admin	我的頭好痛	頭痛	2023/5/14 22:55:05
admin	我有支氣管炎	支氣管炎	2023/5/18 20:18:18

圖 3.18 網頁查看問診紀錄



圖 3.19 網頁查看特定用戶問診資訊

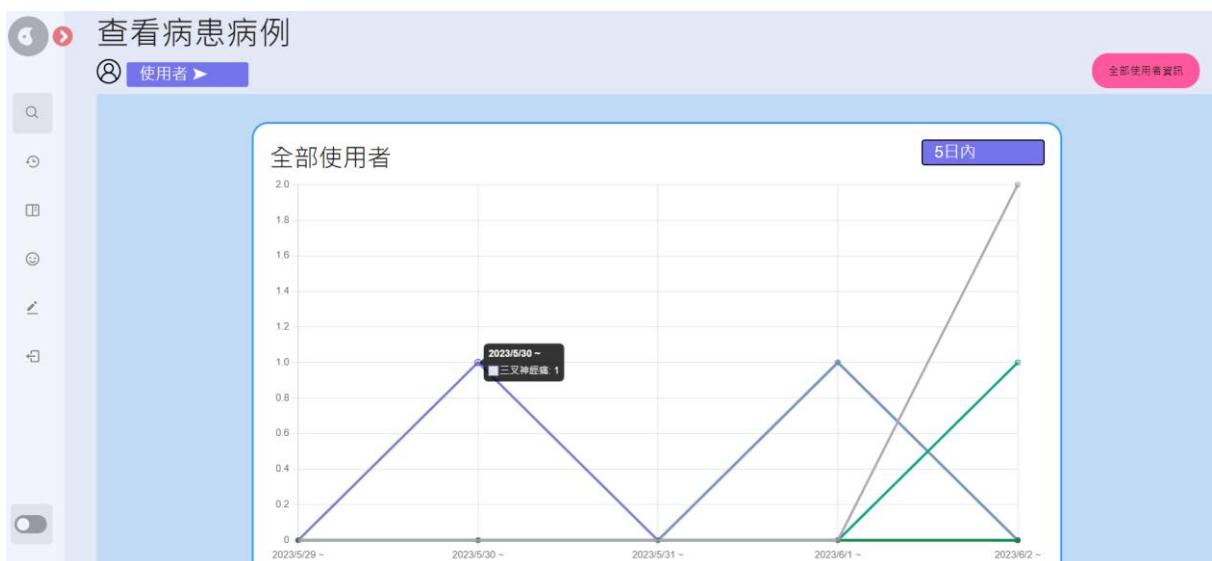


圖 3.20 網頁查看所有用戶問診資訊

第四章 系統實測

4.1 穴位映射實測

可以看到，在實測中，透過我們穴位映射演算法，手部穴位貼合在螢幕上，頭部的穴位也不會因為戴上口罩而影響判斷。



圖 4.1 穴位映射效果展示

透過 DNN 手部辨識網路，判斷使用者目前的手勢，以達到與硬體互動的效果，可以手握拳頭，呼叫選單(如圖 4.2)，利用選擇手勢選擇需要的功能。圖 4.3 演示呼叫選單後，選擇介紹頁面，我們可以使用往右手勢或是往左手勢查看詳細內容。



圖 4.2 呼叫選單

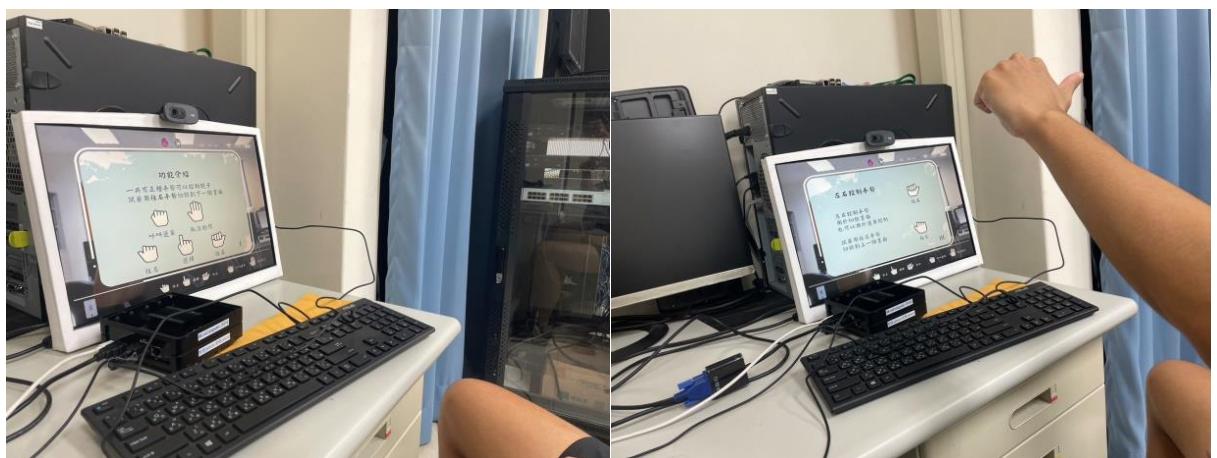


圖 4.3 左右手勢控制 UI

選擇手勢可以與螢幕上的穴位點進行 AR 互動，查看穴位的功能簡介。

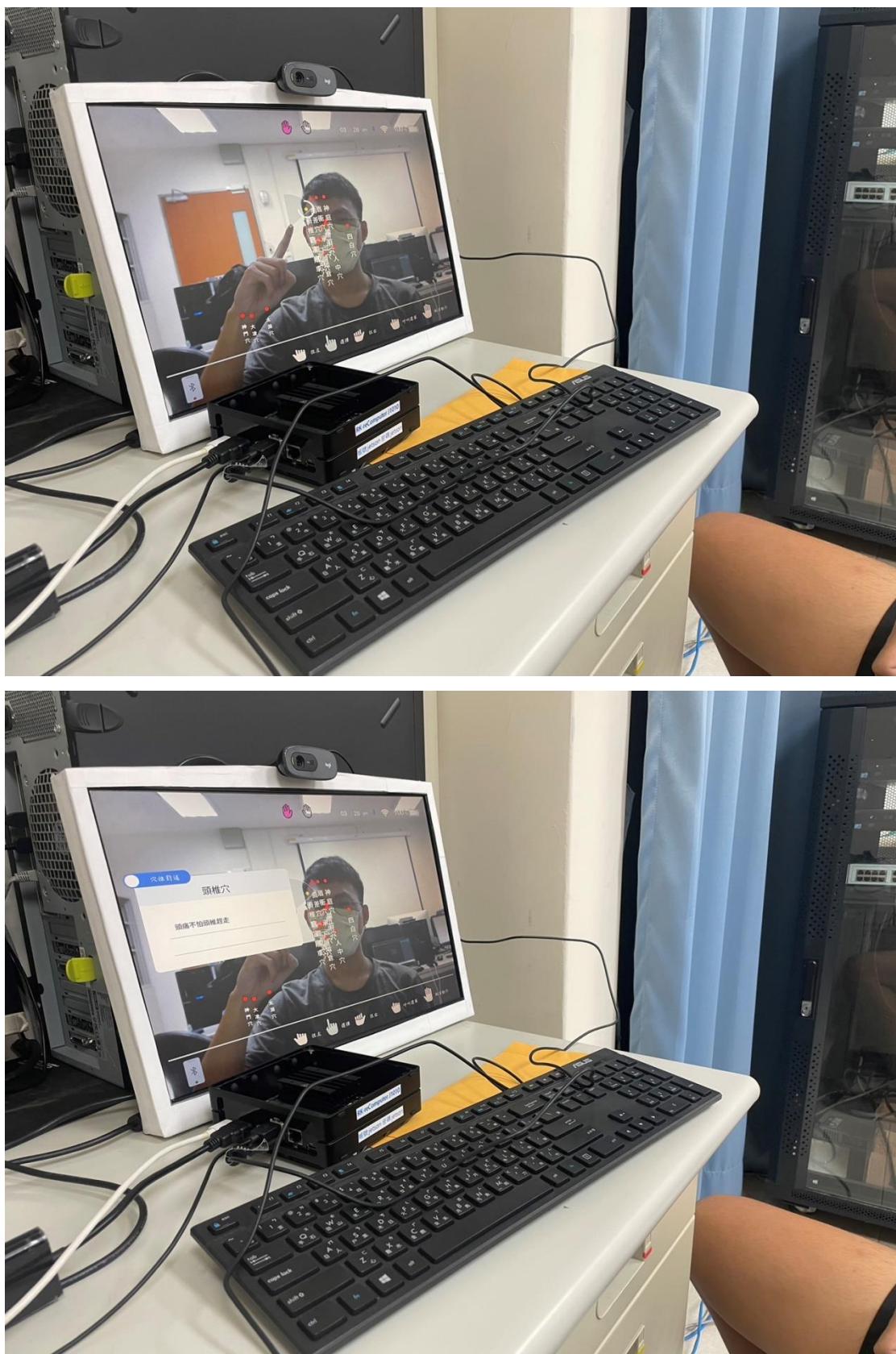


圖 4.4 選擇手勢 AR 互動

4.2 手機連線實測

接下來勢手機連線實測，透過 Qrcode，設備可以與手機進行配對。



圖 4.5 手機藍芽連線

可以透過手機查看想要查看的症狀對應穴位點，或是顯示穴位詳細資訊。

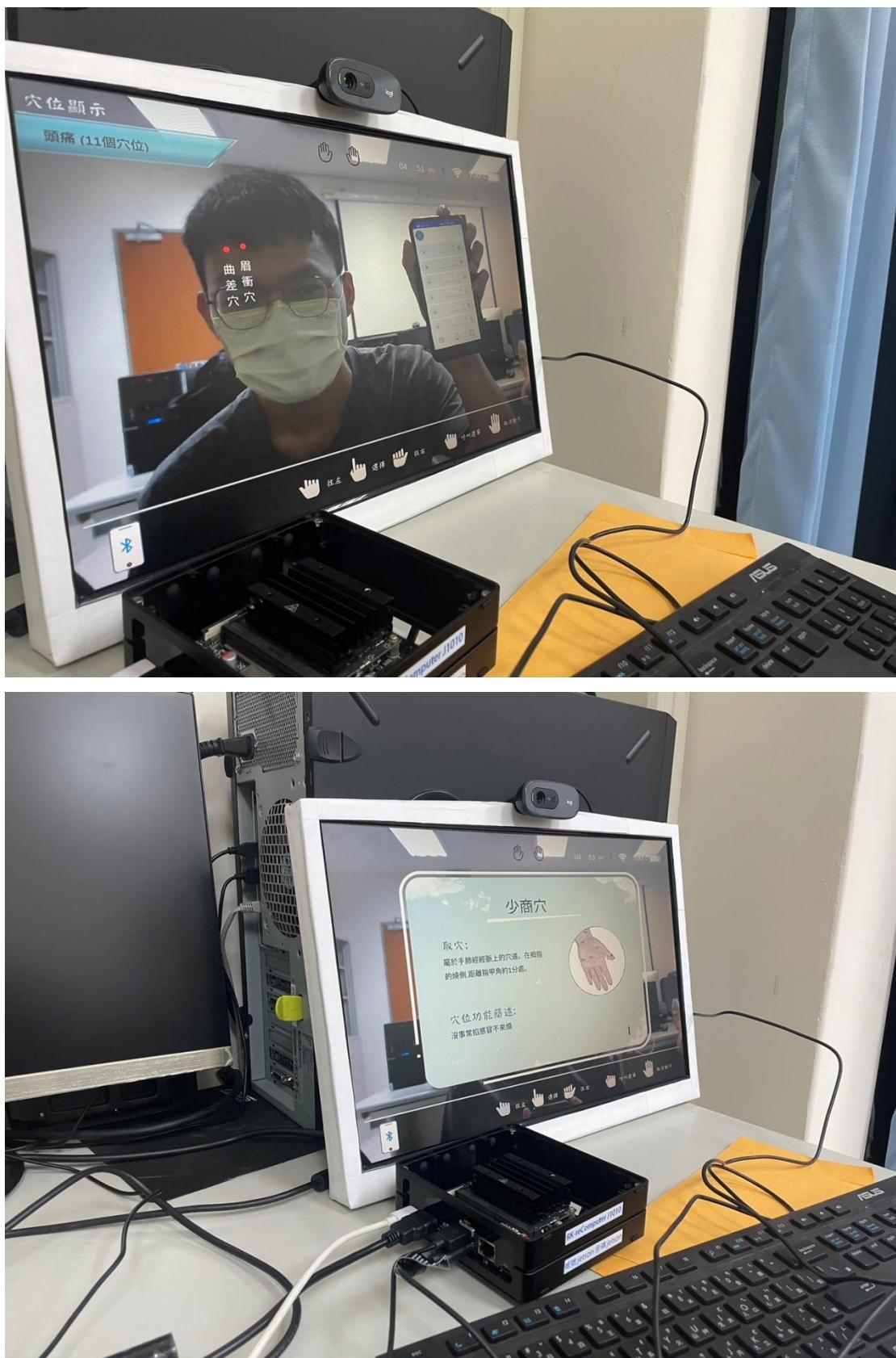


圖 4.6 利用手機查看穴位分類及穴位詳情

手機上的分類畫面以及 UI 介面，手機可以控制目前智慧鏡子要使用的模型。

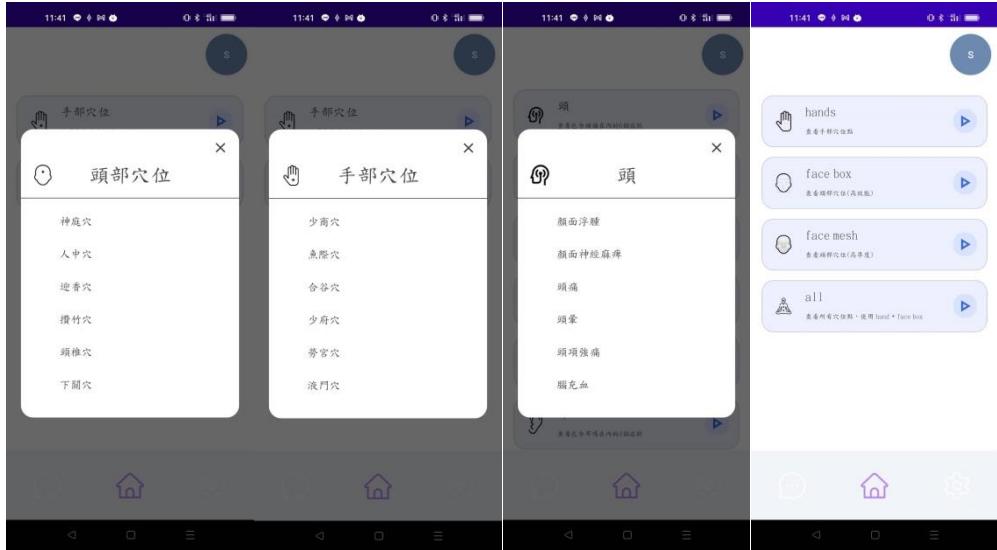


圖 4.7 手機 UI 介面展示

4.3 AI 問診實測

圖是展示手機 APP 進行 AI 問診，手機會先和 Rasa Server 進行 query，得到對應的資訊後，傳送問診訊息到資料庫。如果 BERT 模型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。

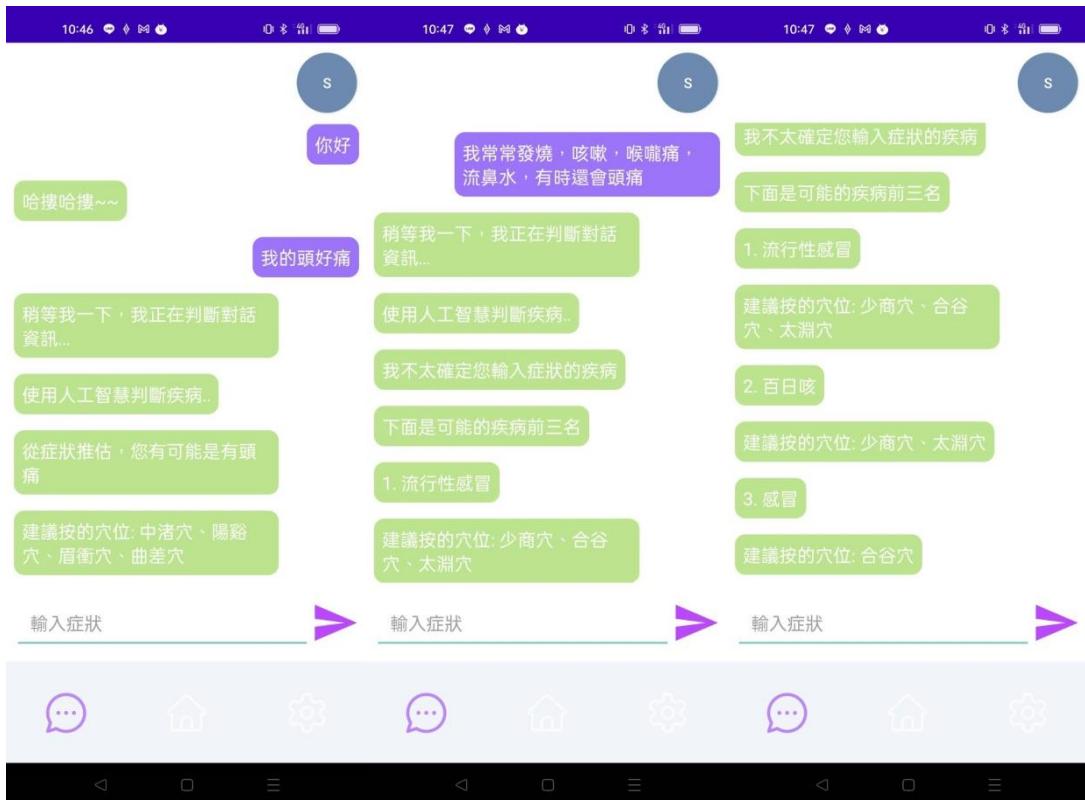


圖 4.8 手機 AI 問診

當手機進行 AI 問診後，可以在網頁端查看使用者的問診資訊，圖 4.9 中可以看到問診資訊已經更新了。

使用者名稱	問診內容	推斷疾病	問診日期
guest	我的頭好痛	頭痛	2023/6/2 22:42:22
guest	我常常發燒、咳嗽、喉嚨痛，流鼻水，有時還會頭痛	流行性感冒	2023/6/2 22:55:30

圖 4.9 伺服器問診資訊

遠端 AI 問診伺服器，使用公開伺服器，架設 Rasa，讓使用者可以在伺服器開啟時隨時隨地 AI 問診。

```

Rasa server connector
(base) C:\Users\user>cd H:\SeniorProject\ChatBot\Rasa-Unity-dev\Rasa
(base) C:\Users\user>conda activate installingRasa
(installingRasa) C:\Users\user>rasa run actions
(installingRasa) H:\SeniorProject\ChatBot\Rasa-Unity-dev\Rasa>rasa run actions
(installingRasa) H:\SeniorProject\ChatBot\Rasa-Unity-dev\Rasa>python -m rasa run actions
(installingRasa) H:\SeniorProject\ChatBot\Rasa-Unity-dev\Rasa>SANIC_VERSION = LooseVersion(sanic_version)
Version classes are deprecated. Use packaging.version instead.
G:\Programfile\ANACONDA\envs\installingRasa\lib\site-pacakge>2022-11-16 00:26:59 INFO  rasa_sdk.endpoint - Starting action endpoint server...
Version classes are deprecated. Use packaging.version instead
SUCCESS TO CONNECT DATA BASE, ALL OPERATION IS SUCCESS
  SANIC_VERSION = LooseVersion(sanic_version)
SUCCESS TO CONNECT DATA BASE, ALL OPERATION IS SUCCESS
G:\Programfile\ANACONDA\envs\installingRasa\lib\site-pacakge>2022-11-16 00:27:00 INFO  rasa_sdk.executor - Registered function for 'clear_certain_slot'.
Version classes are deprecated. Use packaging.version instead
SUCCESS TO CONNECT DATA BASE, ALL OPERATION IS SUCCESS
distutils Version classes are deprecated. Use packaging.version instead
SUCCESS TO CONNECT DATA BASE, ALL OPERATION IS SUCCESS
distutils Version classes are deprecated. Use packaging.version instead
SUCCESS TO CONNECT DATA BASE, ALL OPERATION IS SUCCESS
G:\Programfile\ANACONDA\envs\installingRasa\lib\site-pacakge>2022-11-16 00:27:02 INFO  rasa_sdk.executor - Registered function for 'action_say_disease'.
G:\Programfile\ANACONDA\envs\installingRasa\lib\site-pacakge>2022-11-16 00:27:02 INFO  rasa_sdk.endpoint - Action endpoint is up and running on http://0.0.0.0:5055
onWarning: distutils Version classes are deprecated. Use packaging.version instead.
min_version = LooseVersion(INCLUSIVE_MIN_TF_VERSION)
2022-11-16 00:27:04 INFO  rasa.core.processor - Loading model models\20220918-161920-progressive-sharp.tar.gz...
G:\Programfile\ANACONDA\envs\installingRasa\lib\site-packages\rasa\utils\train_utils.py:531: UserWarning: constrain_similarities is set to 'False'. It is recommended to set it to 'True' when using cross-entropy loss.
category=UserWarning,
2022-11-16 00:27:07 INFO  root - Rasa server is up and running.

```

圖 4.10 Rasa Bert 伺服器

第五章 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 Jetson nano 裡，讓使用者可以透過智慧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) 使用自定義模型，節省 Jetson Nano J1010 需要運算效能，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，使得模機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。

本專題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。

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 Lugaesi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Ubweja, 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.
- [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, Łukasz 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] Joshua Willman, Apress Berkeley, Hampton, USA, December 8, 2020.
- [12] crownpku, Rasa NLU for Chinese , Retrieved September 25 , 2022, from https://github.com/crownpku/Rasa_NLU_Chi
- [13] Gee-Sern Hsu, Hsiao-Chia Peng and Kai-Hsiang Chang, Landmark Based Facial Component Reconstruction for Recognition Across Pose, June 23-28 2014, pp. 34-39
- [14] Menglong Chang, Qing Zhu, Automatic location of facial acupuncture-point based on facial feature points positioning April 1 2017, pp. 545-549
- [15] Camillo Lugaressi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Ubweja, 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.
- [16] Yi-Zhang Chen , Corky Maigre , Min-Chun Hu and Kun-chan Lan, Localization of Acupoints using Augmented Reality, June 20 2017, pp. 239-241
- [17] Menghe Zhang , Jürgen P. Schulze, Dong Zhang, FACEATLASAR: ATLAS OF FACIAL ACUPUNCTURE POINTS IN AUGMENTED REALITY, Mar 29 2022
- [18] seeedstudio (n.d.), reComputer J101 - Carrier Board for Jetson Nano, same size of NVIDIA® Jetson Nano™ Dev Kit carrier board, micro SD card slot, HDMI2.0, USB, CSI, M.2 key E, GPIO, Retrieved September 30 , 2022, from <https://www.seeedstudio.com/Jetson-10-1-A0-p-5336.html>