

國立雲林科技大學電子工程系
實務專題

既有坐姿辨識功能之檯燈

指導教授：夏世昌 特聘教授

專題學生：曹宸維(四電子四 B/B10913154)

陳維翔(四電子四 B/B10913158)

中華民國 113 年 05 月

國立雲林科技大學電子工程系
實務專題研究成果審定書

既有坐姿辨識功能之檯燈

專題學生：曹宸維、陳維翔

本報告書已具備本校實務專題研究訓練之能力

指導教授： 呂世昌 (簽章)

中華民國 113 年 5 月 8 日

國立雲林科技大學機構典藏實務專題報告授權同意書

本授權書所授權之報告為本組在國立雲林科技大學 電子工程 系 系統晶片 組
112 學年度第 1 學期修習實務專題課程之報告。

專題報告名稱：既有坐姿辨識功能之檯燈

本組就具有著作財產權之報告全文資料，同意提供本校圖書館進行機構典藏，並
同意圖書館因典藏之目的就該資料進行必要之數位化重製，且依圖書館法、著作
權法規定以電子形式透過單機、網際網路、無線網路或其他公開傳輸方式，提供
用戶進行檢索、瀏覽、下載、傳輸、列印等。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行
權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。

本組擔保本著作係著作人之原創性著作，並未侵害任何第三人之智慧財產權。倘
若有抄襲行為進而造成侵害他人著作權之情形，願自行負擔相關法律責任。

請勾選電子全文檔授權公開年限(若有商機、專利及發表等考量者，得勾選延後公
開)：

- ☐ 立即公開 ☐ 一年後公開 ☐ 二年後公開 ☐ 三年後公開
☐ 四年後公開 ☐ 五年後公開 ☐ 延後至_____年後公開

指導老師簽名：

廖士昌

學生簽名：

曹宸維

學號：

B10913154

陳維翔

B10913158

(親筆正楷)

(務必填寫)

中 華 民 國

113 年

5 月

8 日

中文摘要

在現代生活中，兒童因長時間使用電子設備、不良的坐姿習慣以及缺乏運動等原因，容易出現姿勢不良的現象，尤其是駝背會造成骨骼發展，為了幫助兒童維持良好的坐姿並預防駝背問題。

創新的解決方案硬體是利用樹梅派 Raspberry Pi 4 +google 開發的 tpu 加速器+加上鏡頭跟檯燈，即執行基於 google 所開發的 coral posenet 人體姿態偵測辨識加以修改成自己所需功能的演算法，能夠實時辨識兒童的骨架結構，特別是關注於頭部、肩部、脊椎等關鍵部位。當偵測到兒童出現不良坐姿或初步的駝背現象時，檯燈將以燈光變化、聲音提示或震動等方式發出即時提醒，引導兒童調整坐姿。

此外，在未來我希望能將我的智慧檯燈將具有個性化的設定功能，讓家長可以根據兒童的特定需求和舒適度進行調整。透過擁有數據追蹤和分析功能，檯燈可以記錄兒童坐姿的變化趨勢，提供給家長有價值的信息，協助他們更好地監測兒童的健康狀態。

關鍵字：姿勢偵測、駝背提醒、posenet



ABSTRACT

In modern life, children often develop poor posture due to prolonged use of electronic devices, bad sitting habits, and lack of physical activity. Particularly, the occurrence of kyphosis can affect skeletal development. To aid children in maintaining good posture and preventing kyphosis, an innovative hardware solution is proposed. The system comprises a Raspberry Pi 4, Google's TPU accelerator, a camera, and a desk lamp. It utilizes an algorithm based on Google's Coral Posenet for real-time recognition of children's skeletal structures, focusing on key areas such as the head, shoulders, and spine. When poor sitting posture or early signs of kyphosis are detected, the desk lamp provides immediate reminders through light variations, sound cues, or vibrations, guiding children to adjust their posture.

Furthermore, future enhancements aim to personalize the smart desk lamp, allowing parents to adjust settings according to their children's specific needs and comfort levels. With data tracking and analysis capabilities, the lamp can record trends in children's sitting posture, providing valuable insights to parents to better monitor their children's health.

Keywords: posture detection, kyphosis reminder, Posenet



誌謝

這份畢業專題研究經過許多的學長姐、同學及老師們、教授們的建議和指導才能夠完成；首先要感謝教授、老師們的指導幫忙，在進行畢業專題製作的過程中遇到了許多的困難，感謝教授、老師們適當的指點，讓我們可以少走許多的彎路，也感謝當我們遇到設備上的問題時，可以提供給我們其他的設備，當我們遇到解決不了的問題時，能提供不一樣的思路，另外也要感謝學長姐們的幫助，藉由許多學長姐們的學習及製作專題的經驗，讓我們能夠更順利的製作並完成專題，還要感謝同學們，在製作專題的過程中，有了其他同學們的幫助，在與同學討論的過程中，每個人的想法都能促進每個人的進步，也讓這份專題有了更多的可能性，把專題做得更好。

在專題製作的一開始，每個人都是一無所知的狀況，在各式各樣人的幫助下學習到的相關知識、透過許多的研究資料及論文、還有組員們的互相合作、學習，讓我們能夠在每天都能收穫到知識，使我們進步，以完成本次的畢業專題。

最後感謝學校提供了完好的學習環境及設備，學校提供了各種硬體設備及實驗室給我們使用，讓我們能在好的環境討論並製作專題，讓我們能順利地完成本次的畢業專題。

目錄

中文摘要	III
ABSTRACT	IV
誌謝	V
目錄	VI
表目錄	VIII
圖目錄	IX
第一章、前言	1
1.1 研究動機.....	1
1.2 構想的情境和場域.....	1
第二章、研究背景與文獻探討	2
2.1 研究背景.....	2
2.2 文獻探討.....	2
2.2.1 PoseNet.....	2
2.2.2 背景比較	3
2.3 套件與工具.....	6
2.3.1 GStreamer.....	6
2.3.2 Pycoral.....	6
2.3.3 tflite_runtime	6
2.3.4 libedgetpu	6
2.4 作品的特色與技術特點.....	7
第三章、系統硬體功能與規格	8
3.1 硬體配置與功能.....	8
3.2 硬體規格.....	9

第四章、系統原理與實作架設	11
4.1 系統流程架構	11
4.2 POSENET	12
4.3 GSTREAMER 流程	13
4.4 實作步驟內容	14
4.4.1 PoseNet 姿勢辨識環境架設	14
4.4.2 駝背判斷演算法	15
4.4.3 Ui 介面設計	18
4.5 解決問題	24
第五章、實驗成果	25
5.1 即時駝背偵測	25
5.2 UI 介面顯示提醒	27
5.3 LED 提醒	28
第六章、結論與未來展望	29
6.1 結論	29
6.2 未來展望與產業發展	29
第七章、參考資料	30

表目錄

表 2-1.四種人體姿態模型比較表	5
表 3-1. 硬體功能說明	8
表 3-2. 嵌入式開發板規格	9
表 3-3. 鏡頭規格	10
表 3-4. 加速器規格	10
表 3-5. USB HUB 規格	10

圖目錄

圖 1-1. 構想情境一	1
圖 2-1. Posenet 關節點圖	2
圖 2-2. Openpose 模型	3
圖 2-3. AlphaPose 網路架構	4
圖 2-4. MoveNet 架構圖	4
圖 3-1. 硬體架構圖	8
圖 4-1. 系統架構流程	11
圖 4-2. PoseNet model	12
圖 4-3. 姿態估計(Pose Estimation) 概念示意圖	13
圖 4-4. Gstreamer 框架的應用簡單分層	13
圖 4-5. GStreamer 串流影音播放工作流程	14
圖 4-6. 樹梅派指令集	14
圖 4-7. 安裝 library 指令	15
圖 4-8. 安裝 TensorFlow Lite library 指令	15
圖 4-9. Install just the TensorFlow Lite interpreter 指令	15
圖 4-10. 安裝 python3-edgetpu 指令	15
圖 4-11. 安裝 posenet 指令	15
圖 4-12. 程式概念示意圖	16
圖 4-13. 程式駝背流程圖	16
圖 4-14. 駝背比例圖	17
圖 4-15. 判斷駝背程式	17
圖 4-16. 控制 PWM 亮燈程式	18

圖 4-17. UI layout.....	20
圖 4-18. 線程概念圖	21
圖 4-19. UI control	23
圖 4-20. 解決方法(1)	24
圖 4-21. 解決方法(1)	24
圖 4-22. 解決方法(2)	24
圖 5-1. 偵測畫面正常坐姿	25
圖 5-2. 偵測畫面駝背坐姿	25
圖 5-3. 偵測畫面正常坐姿(1)	25
圖 5-4. 偵測畫面正常坐姿(2)	25
圖 5-5. 偵測畫面正常坐姿(3)	26
圖 5-6. 偵測畫面駝背坐姿(4)	26
圖 5-7. 偵測畫面駝背坐姿(5)	26
圖 5-8. 偵測畫面駝背坐姿(6)	26
圖 5-9. 即時辨識駝背數據	26
圖 5-10. UI 介面(正常坐姿)	27
圖 5-11. UI 介面(駝背坐姿)	27
圖 5-12. LED 提醒(正常坐姿)	28
圖 5-13. LED 提醒(駝背坐姿)	28

第一章、前言

1.1 研究動機

近年來，隨著電子產品的普及和智慧手機的普及，越來越多的青少年和兒童長時間沉迷於手機、電腦等電子產品，導致了不良的坐姿習慣，加劇了駝背問題的發生，脊椎側彎成了現代人常有的通病，根據一些研究數據顯示，隨著電子產品使用時間的增加，以及不適當的桌椅，青少年和兒童的不良坐姿習慣也在逐漸加重。長時間低頭玩手機、久坐不動等不良習慣對頸椎、背部和肩部的健康造成了嚴重影響，增加了駝背問題的發生率，又缺乏運動，容易出現姿勢不良等現象，尤其是駝背會對骨骼發展造成嚴重的影響。

當人們開始重視健康問題時，特別是與不良坐姿相關的健康問題，對於駝背問題的關注度也逐漸增加。隨著現代生活方式的改變，人們長時間坐姿工作、使用電子產品的情況變得普遍，這些不良姿勢容易導致駝背問題的出現。為了應對這一趨勢，基於 TensorFlow 的 PoseNet 即時人體骨架偵測技術應運而生，而其應用於樹莓派上的智慧檯燈產品，正是將這一技術用於日常生活中的一個例證。

1.2 構想的情境和場域

如圖 2-1 所示為一種使用情境場域，圖一的左側為駝背的狀態會影響到脊椎扭曲，平常學習旁邊如果沒有人提醒你並非知道現在處於駝背的狀態，但經常駝背又會麻煩身邊的人提醒太過於麻煩，因此想到利用智慧檯燈能只要處於駝背狀態它都能即時亮燈提醒你，讓自己意識到要改善駝背坐姿，進而改善掉駝背這個現代人的通病。



圖 2-1. 構想情境一

第二章、研究背景與文獻探討

2.1 研究背景

為了促進小朋友的骨骼健康，我們開發了一款智能台燈，利用攝影機和人體骨架偵測技術，即時辨識小朋友的坐姿，並透過亮度變化提醒他們維持良好的坐姿，以預防駝背對骨骼發育的不良影響。我們所採用的基於 TensorFlow 的 PoseNet 模型，在樹梅派上實時運行，能夠快速且準確地捕捉人體的特徵值。通過這項創新技術，我們致力於改善小朋友的坐姿習慣，提高他們的骨骼健康水平，讓他們健康成長。未來，我們將進一步優化系統功能，提供更多的個性化設置和健康建議，以滿足不同家庭和用戶的需求。

2.2 文獻探討

2.2.1 PoseNet

PoseNet 是一個 real-time human pose estimation 即時人體姿勢估計，來自 Google 創意實驗室開發機器學習模型，借助運行於 TensorFlow.js 版本的 PoseNet，只需藉由具有攝影機取出影像或是圖片中進行人體姿勢估計，它能夠檢測並跟踪給定圖像或視頻幀中關鍵身體關節（如腕部、肘部、肩部、臀部、膝蓋和腳踝）的位置，就能抓取單人姿勢或多人姿勢的關節特徵值，如圖 2-1，總共有 17 個特徵值為 12 個身體 5 個臉部的特徵值。。這項技術有各種應用，包括擴增實境、健身追蹤、手勢識別和人機交互。

PoseNet 使用卷積神經網絡（CNN）架構，在大量標記的人體姿勢示例數據集上進行訓練，以實現準確和強健的姿勢估計。通常與 TensorFlow.js 等網頁技術一起使用，以在網頁瀏覽器中直接實現姿勢估計，從而使其可應用於各種應用場景。



圖 2-1. Posenet 關節點圖【8】

2.2.2 背景比較

常見的背景有：PoseNet、OpenPose、AlphaPose、MoveNet 等做法，每種特點都有不同的特性，其個別說明如下：

1、人體姿態辨識

人體姿態辨識是一種計算機視覺技術，給定一幅圖像或者一段視頻，從圖像或影片中去檢測和分析人體關節點的過程。利用深度學習進行姿態估計的方法大致分為兩種：自上而下的方法和自下而上的方法。自上而下（top-down），即先檢測出人體，再對單個人進行姿態估計；而自下而上（down-top），則是先檢測出人體關節點，如頭部、肩膀、手肘等，再根據檢測出來的關節點連成人體骨架。並估計它們之間的相對位置和關節角度。這項技術有著廣泛的應用，包括運動分析、姿勢評估、姿勢驗證和虛擬試衣間等。人體姿態辨識為各個領域提供了更精準、智能的解決方案，將在未來持續發展並應用於更多的場景中。

2、PoseNet

PoseNet 是由 Google 開發的輕量級人體姿態估計模型，旨在實現即時姿態估計，適用於在移動設備和瀏覽器上進行姿態估計，具有良好的性能和低延遲，主要特點是速度快、易於部署和在瀏覽器中運行。

3、OpenPose

OpenPose 是一種基於深度學習的開源人體姿態估計系統，能夠同時檢測多人姿態，具有高準確性，能夠檢測人體的多個關鍵點，包括身體、臉部和手部的姿態，提供了豐富的功能和彈性的配置，可用於各種應用場景。

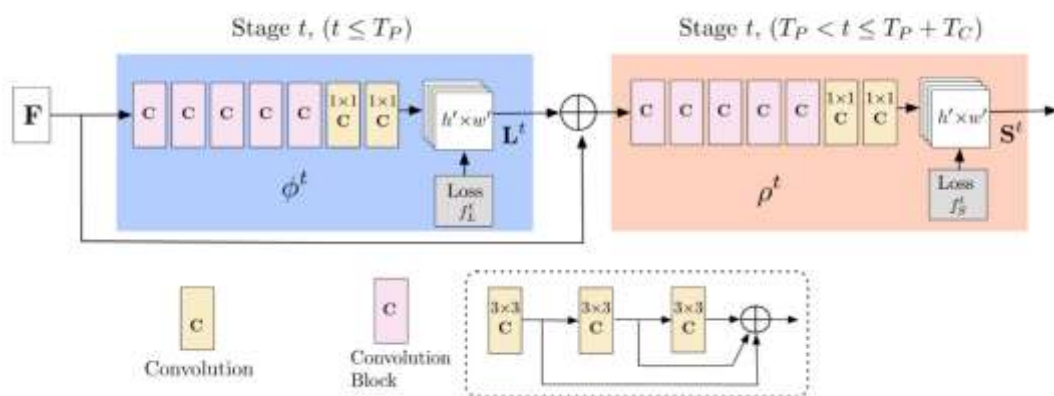


圖 2-2.Openpose 模型【14】

4、AlphaPose

AlphaPose 是一種高效的多人姿態估計系統，能夠實時檢測人體的姿態。它採用了一種混合的自上而下和自下而上的方法，既可以單獨檢測單個人的姿態，也可以同時檢測多人的姿態，具有高度的準確性和效率，特別適用於需要實時多人姿態估計的應用場景。

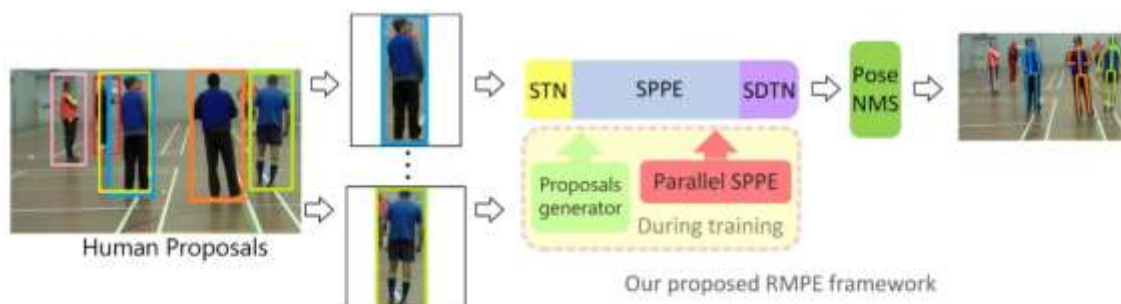


圖 2-3. AlphaPose 網路架構【12】

5、MoveNet

MoveNet 是由 Google 開發的一個輕量級、高效率的人體姿態估計模型，旨在實現即時姿態估計，是一種自下而上的姿態估計模型，使用熱圖來精確定位人體關鍵點。該架構由兩個部分組成：特徵提取器和一組預測頭。預測方案大致遵循 CenterNet，但相較於該架構而言，MoveNet 大幅提升了速度和準確性。所有模型均使用 TensorFlow 物件檢測 API 進行訓練【15】，且它具有快速的推理速度和較低的模型大小，適用於移動設備和嵌入式系統，主要用於單人姿態估計，具有良好的準確性和性能。

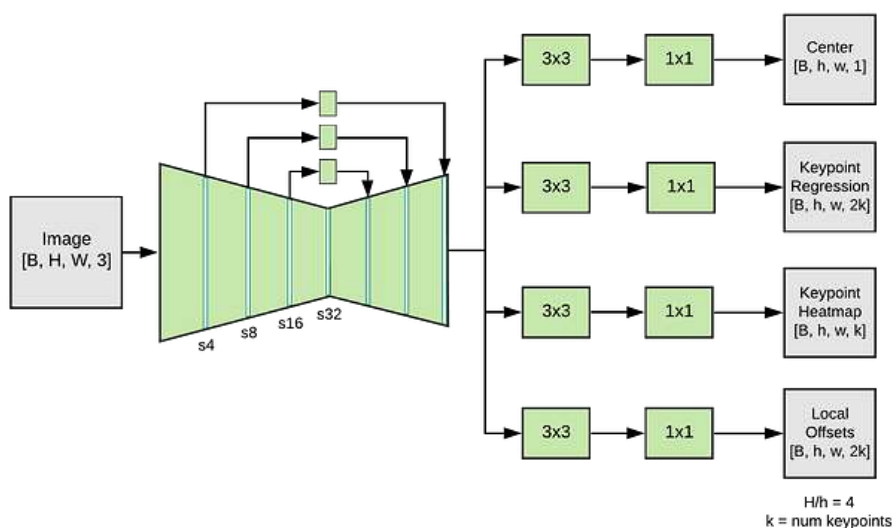


圖 2-4. MoveNet 架構圖【15】

總結，這四種模型在設計上各有其重點，用於不同的應用場景和需求。PoseNet 專注於單人姿態估計，OpenPose 能夠同時檢測多個人的姿態，而 AlphaPose 和 MoveNet 則更注重於效率和輕量化，適用於移動設備和資源受限的環境。

表 2-1.四種人體姿態模型比較表

即時人體估計模型	PoseNet	OpenPose	AlphaPose	MoveNet
比較				
方法	PoseNet 是由 Google 開發的基於深度學習的人體姿態估計模型，主要用於檢測單個人的姿態也有多人姿態。	OpenPose 是一種流行的開源人體姿態估計系統，採用多階段卷積神經網絡（CNN）和 Part Affinity Fields（PAFs）進行姿態估計	AlphaPose 是一種高效的多人姿態估計模型，採用自下而上的方法，先檢測出關鍵點，然後連接成人體骨架。	MoveNet 是由 Google 開發的輕量級人體姿態估計模型，旨在實現快速、準確的姿態估計。
特別點	PoseNet 的優點之一是輕量級和快速，適用於移動設備和瀏覽器上的實時應用。它通常用於單人姿態估計，例如在增強現實應用中追蹤使用者的動作。	OpenPose 能夠同時檢測多個人的姿態，並精確地連接人體的關鍵點，適用於複雜的多人姿態估計場景。	AlphaPose 具有較高的準確性和效率，在多人場景下表現良好，可用於人群密集的姿態估計任務。	MoveNet 採用了 MobileNetV2 和深度可分離卷積等輕量級結構，適用於移動端和嵌入式設備上的實時姿態估計。
缺點	PoseNet 在多人場景下的性能較弱，不太適合多人姿態估計任務。	OpenPose 的模型較為複雜，需要較大的計算資源，在一些低端設備上運行可能效率較低。	AlphaPose 的模型較大，需要較高的計算資源，因此在一些嵌入式設備上可能不太適用。	MoveNet 在精細度上可能不及一些複雜模型，但在速度和資源消耗上具有優勢。

總之，在一開始我們不會考慮 OpenPose 和 AlphaPose，主要是因為這兩個模型過於複雜，需要大量的運算資源。我們只需要單人的姿態估計，而且要能夠在 Raspberry Pi 3 上運行。我們擔心這些複雜的模型在 Raspberry Pi 3 上可能無法達到良好的性能，因此沒有考慮這兩個模型。至於 PoseNet 和 MoveNet，這兩個模型都是由 Google 開發的人體姿態估計模型，而且都可以與 Google 開發的加速器 Google Coral USB Accelerator 搭配使用。但考慮到我們只需要供一位使用者使用，並且希望簡單且高效地實現姿態估計，我們最終選擇了使用 PoseNet。

2.3 套件與工具

2.3.1 GStreamer

GStreamer 是一個基於管道的多媒體框架，基於 GObject，以 C 語言寫成。憑藉 GStreamer，程式設計師可以很容易地建立各種多媒體功能組件，包括簡單的音訊回放，音訊和影片播放、錄音，串流媒體和音訊編輯。基於管線設計，可以建立諸如影片編輯器、串流媒體廣播和媒體播放器等等的很多多媒體應用。

2.3.2 PyCoral

PyCoral 是 Google 推出的用於 Coral 系統的 Python 軟件庫。Coral 是一款由 Google 開發的專為邊緣設備（如嵌入式系統和物聯網設備）設計的硬件平台，旨在實現機器學習模型的高效運行。PyCoral 使開發人員能夠使用 Python 語言與 Coral 加速器交互，從而在邊緣設備上運行機器學習模型，而無需過多關注底層硬件實現。這使得在嵌入式系統中部署深度學習模型變得更加容易和高效。PyCoral 還提供了方便的 API，可用於加載、運行和推理 TensorFlow Lite 模型，以及在 Coral 加速器上進行模型推理。

2.3.3 tf-lite_runtime

tf-lite_runtime 是 TensorFlow Lite 的一個 Python 庫，用於在 Python 中運行 TensorFlow Lite 模型。TensorFlow Lite 是 TensorFlow 的輕量級版本，專門設計用於在諸如移動設備、嵌入式設備和邊緣計算裝置等資源受限環境中執行機器學習模型。使用 tf-lite_runtime，開發人員可以輕鬆地在 Python 中載入、運行和推理 TensorFlow Lite 模型，而無需引入整個 TensorFlow 框架。這使得在資源受限的設備上部署機器學習模型變得更加方便和高效。tf-lite_runtime 提供了 Python API，使得與 TensorFlow Lite 模型的交互變得直觀和容易。開發人員可以使用這個庫來加載 TFLite 模型、將輸入數據傳遞給模型、執行推理運算，並獲取推理結果，從而在 Python 中實現對機器學習模型的使用。

2.3.4 libedgetpu

libedgetpu 是 Google 針對 Edge TPU（邊緣 TPU）所推出的 C++ 庫。Edge TPU 是一種專為邊緣計算設備而設計的硬件加速器，旨在提供高效能的深度學習推理能力。libedgetpu 庫讓開發者能夠輕鬆在邊緣設備上載入和執行 TensorFlow Lite 模型，並利用 Edge TPU 的硬件加速功能來進行模型推理。

透過 libedgetpu，開發者可以利用 Edge TPU 的高性能和低功耗特性，在邊緣設備上實現即時、低延遲的深度學習推理任務，例如物件偵測、圖像分類等。該庫提供了 C++ 接口，讓開發者能夠輕鬆將 Edge TPU 的功能集成到其應用程序中，並通過 TensorFlow Lite 模型來支援各種機器學習任務。

2.4 作品的特色與技術特點

此作品以「即時且精準駝背偵測」為目的進行研究，並且產生 4 點特色功能與技術點，分別為：即時人體骨架偵測、駝背偵測功能、即時反饋和提醒雲端紀錄、優化運行於樹莓派。

1.即時人體骨架偵測

智慧檯燈搭載了 TensorFlow 的 PoseNet 模型，能夠即時從攝影機獲取影像，並快速、精準地偵測人體骨架，特別關注於頭部、肩部和脊椎等關鍵部位。

2.駝背偵測功能

在本作品中，智慧檯燈利用 PoseNet 提取的特徵值，運用特定的演算法進行判斷，從而即時辨識出使用者是否存在駝背現象。一旦偵測到駝背，檯燈將發出警示，提醒使用者調整坐姿。

3.即時反饋和提醒

本作品智慧檯燈具有即時反饋功能，一旦偵測到駝背，將立即通過燈光變化、聲音提示等方式向使用者發出提醒，幫助其改善坐姿。

4.優化運行於樹莓派

智慧檯燈的設計考慮到樹莓派等嵌入式系統的限制，優化了模型的運行效率和資源消耗，確保在樹莓派等硬體上穩定運行。

綜合而言，這款智慧檯燈利用 TensorFlow 的 PoseNet 模型實現了即時的人體骨架偵測，並搭配特定演算法判斷駝背情況，提供即時的警示和提醒功能，以幫助使用者改善坐姿，預防駝背問題的發生。同时，针对樹梅派等嵌入式系统的特点进行了改良，确保了在该硬件平台上的稳定运行。

第三章、系統硬體功能與規格

3.1 硬體配置與功能

此作品整合了多種技術與硬體，旨在幫助使用者改善坐姿，特別是避免駝背的問題。如圖 3-1 所示，首先，我們使用了嵌入式系統開發板 Raspberry Pi 3 Model B 作為系統的核心，並通過 Google Coral USB Accelerator 提供的強大運算能力，實現了 PoseNet 模型的即時影像分析。這使得系統能夠即時捕捉使用者的姿勢並做出反饋。為了獲取影像數據，我們使用了一隻攝像頭，並通過 GPIO 接腳和 USB 介面連接了相關的硬體，例如 LED 燈和其他感知設備。當系統檢測到使用者的坐姿可能存在駝背問題時，LED 燈會通過調節亮度來提醒使用者。這個系統的核心技術是 PoseNet 模型，它可以通過影像處理技術從攝像頭捕捉的畫面中提取骨架特徵值，從而準確地識別出使用者的坐姿狀態。而 LED 燈的警示功能則有效提高了使用者對於自身坐姿狀態的認知和警覺性。

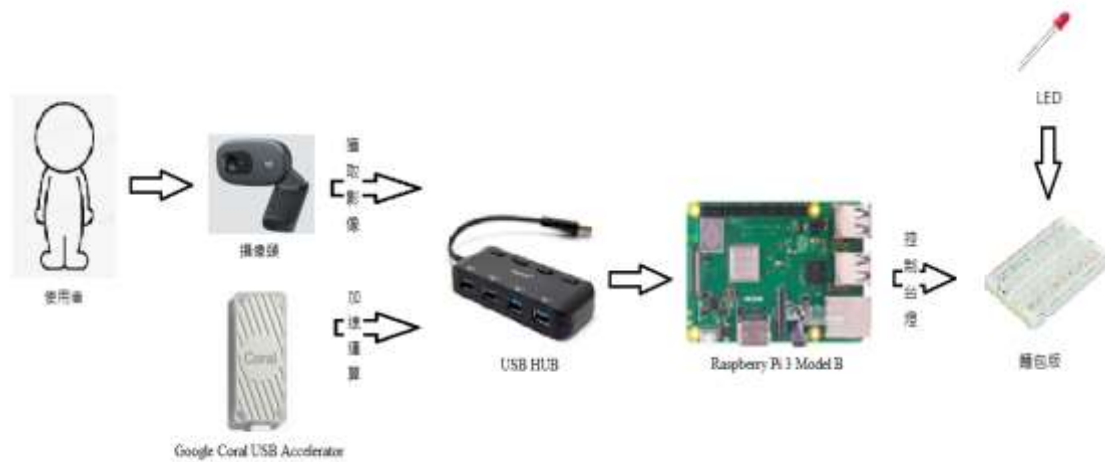


圖 3-1 硬體架構圖

表 3-1. 硬體功能說明

硬體名稱	圖片	功能
Raspberry Pi 3 Model B		接收影像訊號。 PoseNet 運算。 影像辨識運算。 控制 GPIO。

Logitech C270 HD WEBCAM		擷取人正面的坐姿影像，傳送資料至 Raspberry Pi 3 Model B 進行即時影像辨識。
Google Coral USB Accelerator		Google USB Accelerator 是一款 USB 界面的人工智慧加速器，可以為 Linux 或 Raspberry Pi 等單板電腦提供強大的運算能力。
USB HUB		提供 Raspberry Pi 3 擴充 USB 孔之數量，以供鏡頭訊號使用。

3.2 硬體規格

如表 3-2 至表 3-55 所示，為開發檯燈所需之硬體規格。

表 3-2. 嵌入式開發板規格


Raspberry Pi 3 Model B		
	GPU	雙核心 VideoCore IV® Multimedia Co-Processor
	CPU	4 核心 ARM Cortex-A53 64 位元 CPU
	記憶體	1 GB LPDDR2 (和 GPU 共享)
	儲存空間	32GB eMMC 5.1
	視訊輸出	Composite RCA/HDMI
	傳輸介面	USB 2.0、10/100 RJ45、GPIO、Bluetooth 4.1、Wireless 802.11n
	尺寸	85mm x 56mm x 17mm
	功耗	300 mA(1.5W) 1.34A(6.7W)

表 3-3. 鏡頭規格

Logitech C270 HD WEBCAM		
	像素	1280 x 720
	解析度	720P
	幀數	30 幀
	角度	水平 55°
	焦距	固定焦距
	傳輸介面	USB 2.0

表 3-4. 加速器規格

Coral USB Accelerator		
	運算加速卡	5V
	觸發電流	5mA
	最大負載	AC 250V/10A DC 30V/10A
	模組尺寸	73 x 50 x 18.5 (mm)
	控制數量	4 組

表 3-5. USB HUB 規格

INTOPIC 廣鼎 HBC-590 USB3.1 Type-C 高速集線器		
	支援標準	USB1.1/USB2.0/USB3.1
	傳輸速率	最高 5Gbps
	傳輸介面	USB3.1 Type-A *4
	傳輸介面	USB3.1 Type-C

第四章、系統原理與實作架設

4.1 系統流程架構

本作品的系統流程如同圖 4-14 所示，透過雙眼攝影機捕捉使用者的影像，將 RGB 影像傳送至 Raspberry Pi 3 Model B 進行處理。接著，使用 PoseNet 即時人體姿態估計模型對影像進行分析，識別出人體骨架和 17 個特徵點。這個系統的核心目標是辨識使用者是否出現駝背的情況。一旦系統識別到使用者坐姿出現駝背，它會通過檯燈和界面的方式提醒使用者。如果檢測到駝背，檯燈會亮起並在界面上顯示"駝背 (hunchback)"，提醒使用者應該調整坐姿。相反，如果坐姿正常，則檯燈會熄滅並在界面上顯示"正常"。這樣的智能提醒系統有助於使用者保持良好的坐姿習慣，促進健康和舒適。

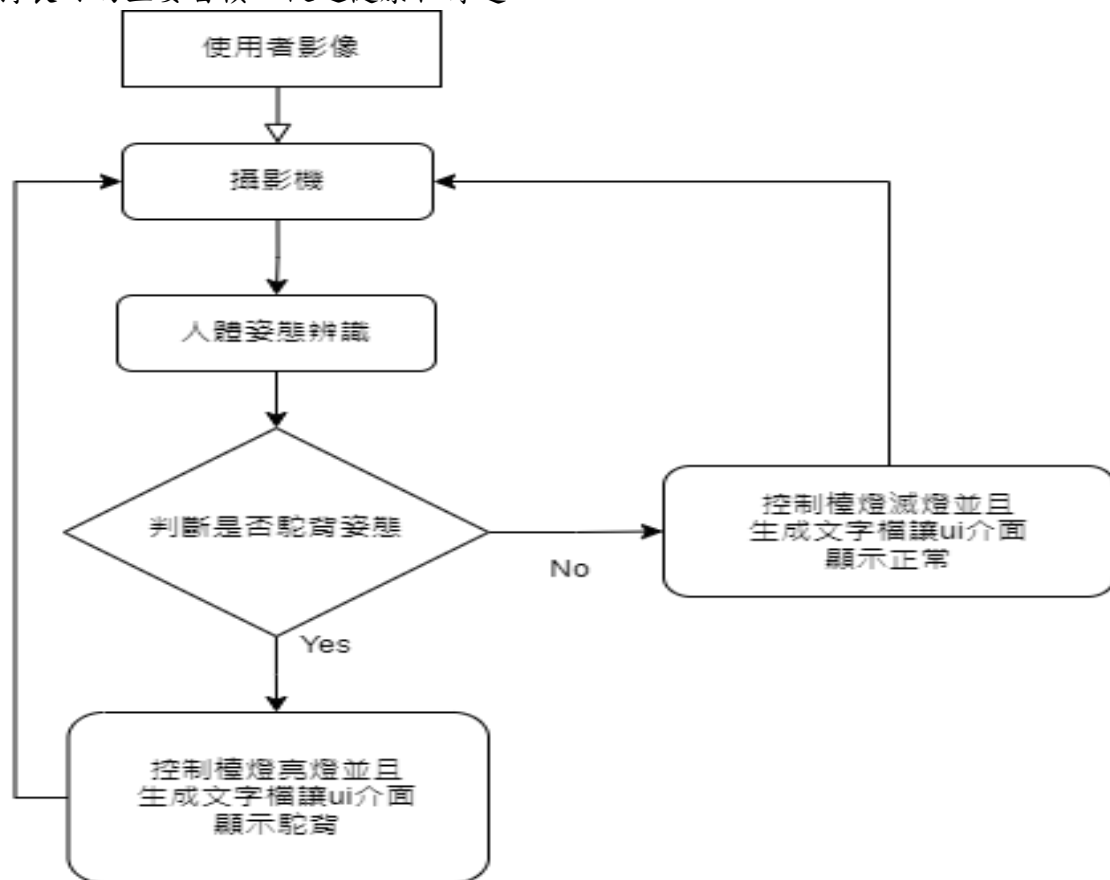


圖 4-1.系統架構流程

4.2 PoseNet

本作品因需要去偵測使用者是否駝背，因此我們想到利用即時人體姿態估計去抓取人的特徵值，姿態估計的實現並非著重於架構的改變，更重要的是如何運用特徵。PoseNet 所運用的 17 個特徵點(key point) [8]；分別為眼睛、耳朵、肩膀、手肘、手腕、臀部、膝蓋、腳踝。簡單來說，就是以 MobileNet 架構去預測這 17 個特徵點的位置資訊。姿態估計概念是由 熱圖(keypoint heatmap) 與 偏移向量(offset vector) 兩大構成。如圖 4-1 所示，熱圖(keypoint heatmap)：表示 17 個特徵點的所在位置，其輸出資訊為 $M \times N \times 17$ 。偏移向量(offset vector)：表示 17 個特徵點的偏移量，其輸出資訊為 $M \times N \times 34$ ，共 34 維度資訊，前 17 個為 x 方向資訊，後 17 個為 y 方向資訊。

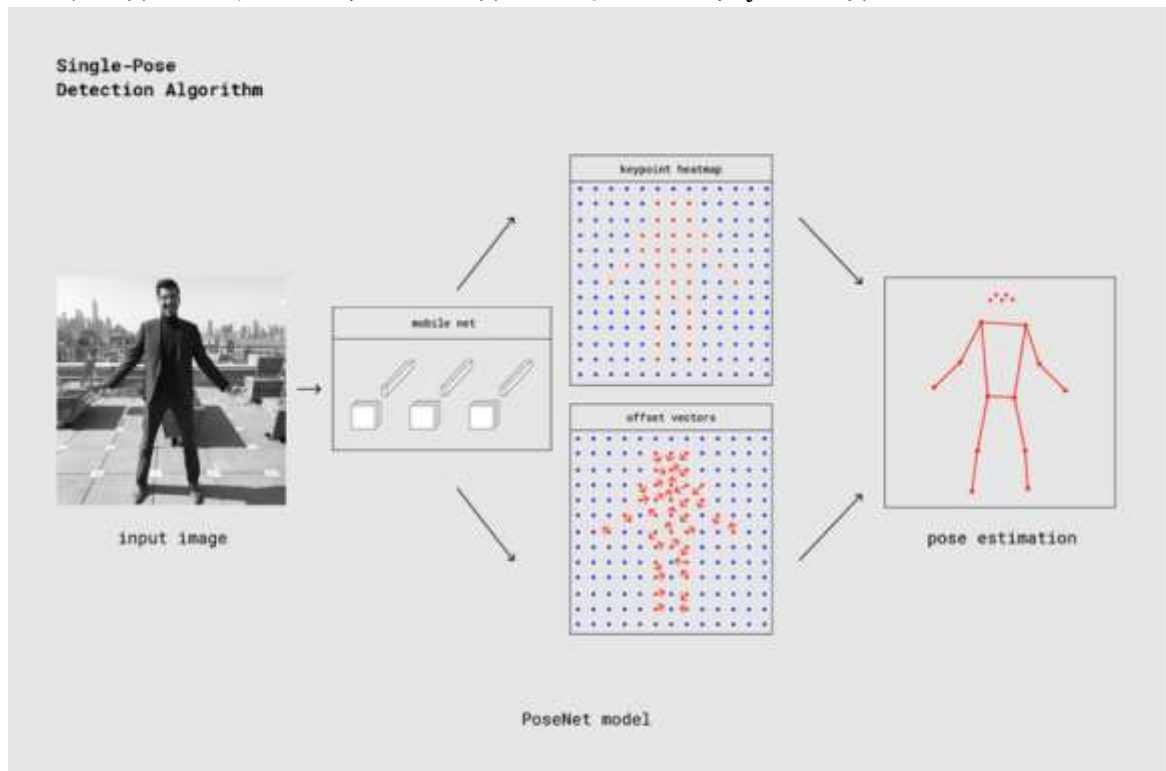


圖 4-2. PoseNet model【9】

以影像大小 224×224 ，步幅 16 作為輸入的話。透過神經網路推理後，將會得到 $14 \times 14 \times 17$ 的熱圖 與 $14 \times 14 \times 34$ 的偏移向量資訊。每一個維度將各自代表對應的特徵(眼睛、耳朵、肩膀..等)，比如在第 10 維度時所代表是手肘位置的資訊與偏移量，如圖 4-2 所示。

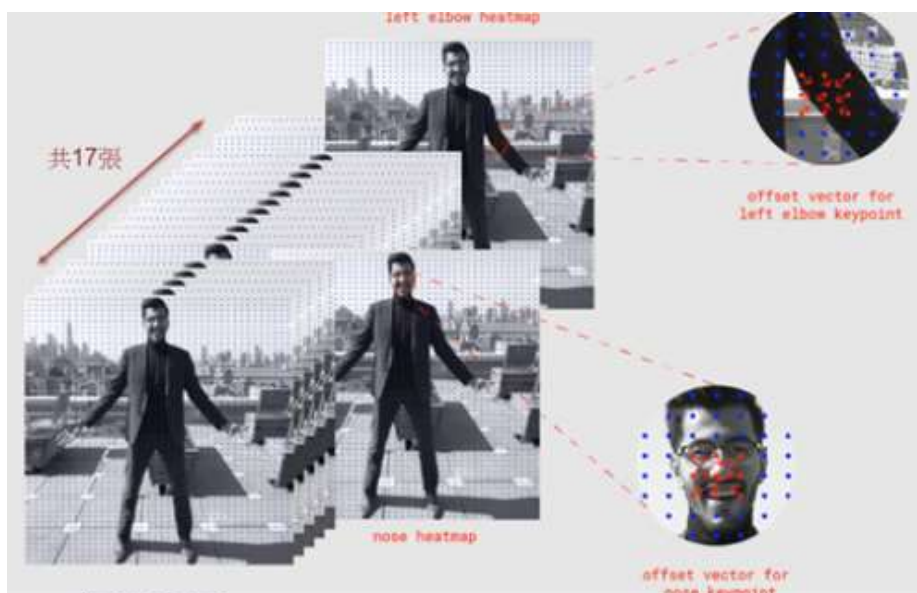


圖 4-3. 姿態估計(Pose Estimation) 概念示意圖【9】

4.3 gstreamer 流程

GStreamer 是一個用創建媒體處理程序的開源多媒體框架，它允許您構建從簡單的音頻/視頻播放器到複雜的影片編譯器等各種多媒體應用程序。在 GStreamer 中，資料流被稱為管道（pipeline）。一個管道由元素（element）組成，元素之間通過連接（pad）進行通訊。元素負責執行特定的任務，如圖 4-3，解碼、編碼、過濾等。管道的資料流從源（source）元素開始，經過一系列處理元素，最終到達匯（sink）元素。

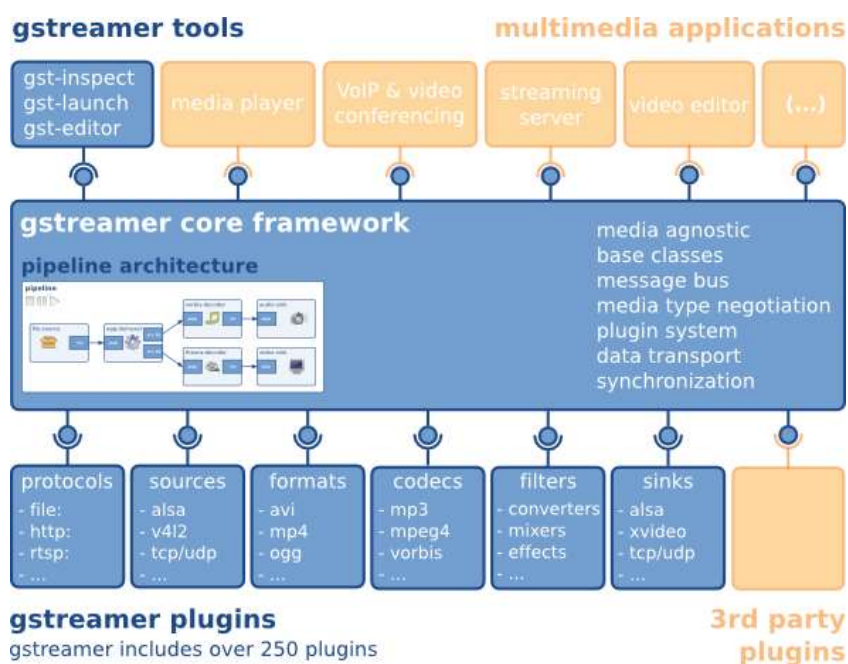


圖 4-4. Gstreamer 框架的應用簡單分層【10】

它主要工作在 Linux 上，有很多硬體(CPU, GPU, DSP, CODEC, ASIC 等)廠商都對其有優化，使得影音內容在壓縮（編碼 Encode） / 解壓縮（解碼 Decode）效率大幅提升。如圖 4-4 所示，GStreamer 採取功能方塊作法，每個方塊依不同工作項目會有不同輸入(sink)和輸出信號源(src)，再依需求將前後串聯起來。以一個影音播放器(player)為例，首先開啟影音檔案，再將其分解(demuxer)成聲音(src_01)和影片(src_02)兩組信號源，再依聲音和影片格式給予對應的解碼器(decoder)，最後再由對應的喇叭和顯示器將聲音和影片播出來，完成串流影音檔案播放工作。當然這裡是採取一邊讀取檔案一邊播放的串流型式存在。

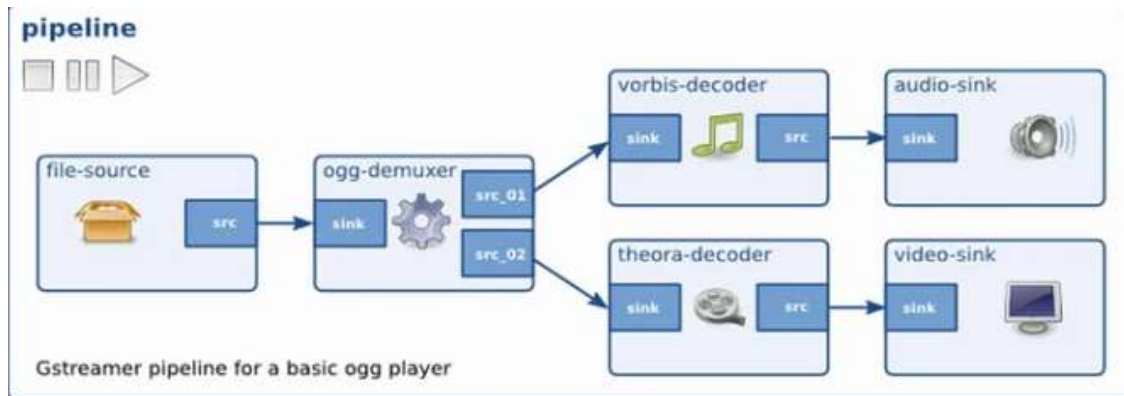


圖 4-5. GStreamer 串流影音播放工作流程【10】

4.4 實作步驟內容

4.4.1 PoseNet 姿勢辨識環境架設

如圖 4-7 至圖 4-11，按照步驟去安裝 PoseNet 所需要的套件，如 pycoral、tflite_runtime、libedgetpu，需要特別注意樹莓派的指令集，確保指令集為 armv7l，如圖 4-6 所示，並相應地修改指令集資料夾，以確保編譯的順利進行。

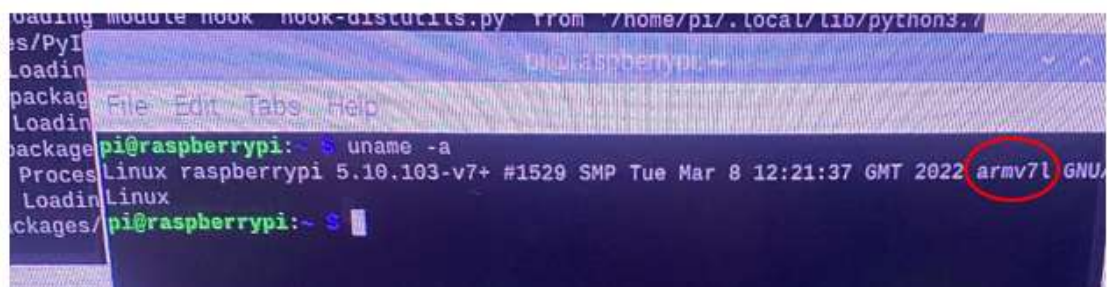


圖 4-6. 樹梅派指令集

```
echo "deb https://packages.cloud.google.com/apt coral-edgetpu-stable main"
| sudo tee /etc/apt/sources.list.d/coral-edgetpu.list

curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key
add -

sudo apt-get update

sudo apt-get install libedgetpu1-std
```

圖 4-7. 安裝 library 指令【7】

```
mkdir coral && cd coral

git clone https://github.com/google-coral/tflite.git

cd tflite/python/examples/classification

bash install_requirements.sh
```

圖 4-8. 安裝 TensorFlow Lite library 指令【7】

```
wget https://dl.google.com/coral/python/tflite_runtime-1.14.0-cp37-cp37m-linux_armv7l.whl
pip3 install tflite_runtime-1.14.0-cp37-cp37m-linux_armv7l.whl
```

圖 4-9. Install just the TensorFlow Lite interpreter 指令【7】

```
sudo apt-get install python3-edgetpu
```

圖 4-10. 安裝 python3-edgetpu 指令【7】

```
git clone https://github.com/google-coral/project-posenet

sh install_requirements.sh

python3 simple_pose.py
```

圖 4-11. 安裝 posenet 指令【7】

4.4.2 駝背判斷演算法

它有 17 個特徵點，我們建立所需的特徵值右耳、左耳、右肩、左肩，我們想說利用人雙耳距離與雙肩距離的比值進行去判斷，如圖 4-11，正常大部分一般人雙耳距離與雙肩距離比例為 1.5，而如果將攝影機架在使用者正前方只要使用者依駝背頭就會靠前接近攝影機，會使雙肩的距離縮小，反而雙耳的距離擴大，因此肩距與耳距的比值會變小，於是我們只要設定好正常坐姿的比值，只要小於正常坐姿的比值就為駝背，反之，就為正常。

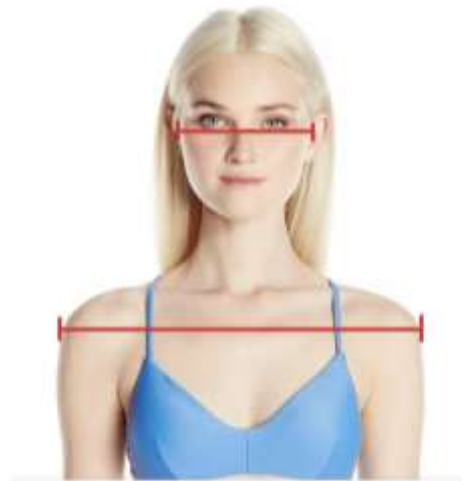


圖 4-12.程式概念示意圖【11】

如圖 4-12，我們藉由預先設定好個人的正常坐姿比值，再來進行判斷，例如我正常坐姿肩距除以耳距為 1.8，因為攝影機是每一幀數(fps) 顯示影格數，因此每一秒都會抓取 PoseNet 的特徵值進行駝背判斷，只要攝影機未中斷，它就算是無窮迴圈在進行，每一幀進行判斷肩距除以耳距的比值只要小於 1.8 就為駝背，大於等於為正常坐姿，如圖 4-13 所示。

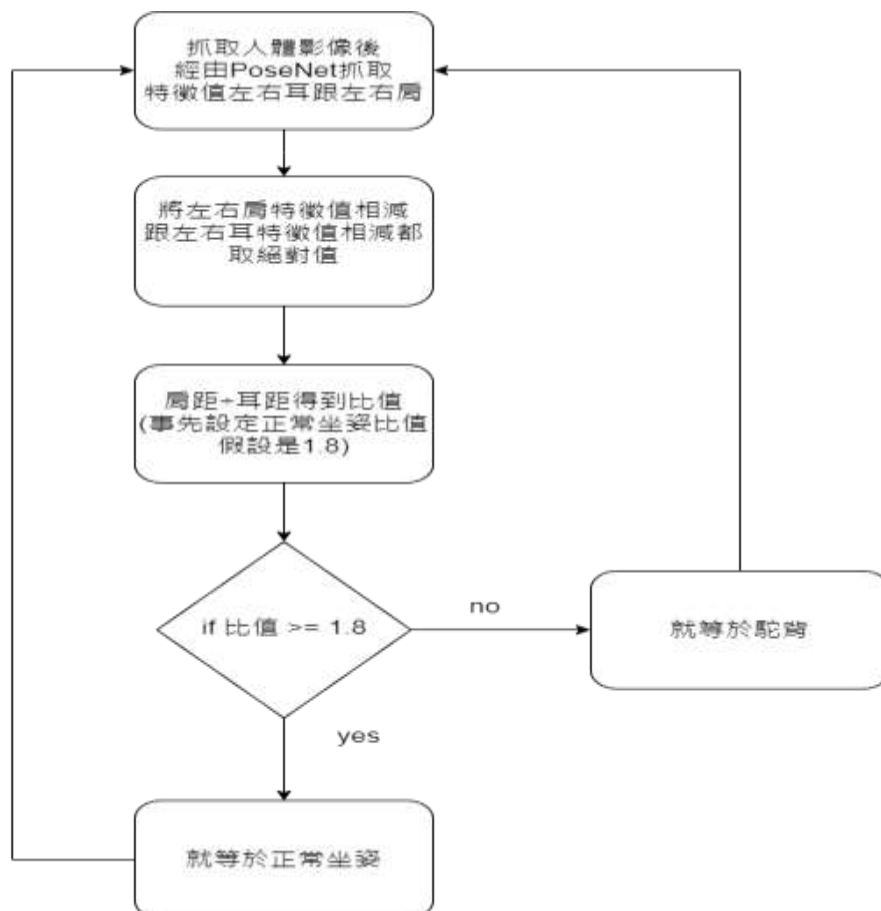


圖 4-13.程式駝背流程圖



圖 4-14.駝背比例圖

根據圖 4-14 的流程，當系統判斷出駝背後，將會呼叫 PWM 副程式，如圖 4-15 所示。如果計算得到的肩距除以耳距的比值小於 1.8，即表示出現駝背，這時會設定 PWM 回傳 0，並將 PWM 週期設為 100，以點亮 LED 燈。同時，系統會生成一個文字檔，讓 UI 能夠偵測到文字檔的存在，從而在 UI 的右下角顯示 "駝背 (hunchback)"。反之，如果肩距除以耳距的比值大於或等於 1.8，表示坐姿正常，系統會設定 PWM 回傳 1，使 PWM 週期為 0，LED 燈熄滅。同時，也會生成一個文字檔，讓 UI 偵測到並在右下角顯示 "正常"。這樣的設計使系統能夠即時反饋使用者的坐姿狀態，並通過 LED 燈和 UI 界面提醒使用者調整坐姿，從而改善坐姿習慣，促進健康。

```
If(ears==1 | shoulder==1):
    print("hunchback")
elif(abs(shoulder/ears)<1.8):
    print("hunchback")
    pwm(0)
    f=open("hunchback.txt","a")//文字檔表示駝背
    f.close()//因為它是每一幀進行判斷，因此在需要關閉讓下一幀顯示
else:
    print("normal")
    pwm(1)
    f=open("normal.txt","a")//文字檔表示正常
    f.close()
```

圖 4-15. 判斷駝背程式

```

def pwm(a)

    pi_pwm=GPIO.PWM(ledpin,1000)

    pi_pwm.start(0)

    if(a==0):

        pi_pwm.ChangeDutyCycle(100)

        sleep(1)

    if(a==1):

        pi_pwm.ChangeDutyCycle(0)

        sleep(0.01)

```

圖 4-16. 控制 PWM 亮燈程式

4.4.3 Ui 介面設計

當我們設計使用者界面時，我們選擇了 PyQt5 函式庫，這是一個功能豐富的工具，可以讓我們按照我們的需求輕鬆建構 UI 介面。我們的目標是提供一個直觀且易於操作的界面，以便使用者可以輕鬆地控制所需的功能。我們將界面分成了幾個區域，其中最主要的是"Start"和"Finish"框。 "Start"框設計為按下後能夠啟動即時人體骨架偵測，這讓使用者能夠輕鬆開始監測的流程。而"Finish"框則是用於結束偵測，當使用者完成觀察或操作時，他們可以輕鬆地結束偵測流程。另外，我們也設計了右邊的框框，這是用來顯示偵測的結果。透過這個區域，使用者可以清楚地看到所捕捉到的人體骨架信息，這有助於他們進行分析和評估。總之，我們的目標是提供一個直觀且易於使用的界面，以便使用者可以輕鬆地控制功能並查看偵測結果。透過 PyQt5 函式庫的支援，我們成功地實現了這一目標，為我們的應用程式帶來了更好的用戶體驗。

```

from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtGui import QFont
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(843, 842)

```

```
self.centralwidget = QtWidgets.QWidget(MainWindow)
self.centralwidget.setObjectName("centralwidget")

self.start = QtWidgets.QPushButton(self.centralwidget)
self.start.setGeometry(QtCore.QRect(30, 750, 151, 41))
self.start.setObjectName("start")
self.start.setFont(QFont('Arial', 16))

self.tv = QtWidgets.QLabel(self.centralwidget)
self.tv.setGeometry(QtCore.QRect(30, 20, 781, 691))
self.tv setFrameShape(QtWidgets.QFrame.Box)
self.tv.setText("")
self.tv.setObjectName("tv")

self.label = QtWidgets.QLabel(self.centralwidget)
self.label.setGeometry(QtCore.QRect(370, 750, 421, 41))
self.label setFrameShape(QtWidgets.QFrame.Box)
self.label.setText("")
self.label.setFont(QFont('Arial', 16))
self.label.setObjectName("label")

self.finish = QtWidgets.QPushButton(self.centralwidget)
self.finish.setGeometry(QtCore.QRect(200, 750, 141, 41))
self.finish.setObjectName("finish")
self.finish.setFont(QFont('Arial', 16))
```

```

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 843, 25))

self.menubar.setObjectName("menubar")

MainWindow.setMenuBar(self.menubar)


self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)


self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)


def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow",
    "MainWindow"))

    self.start.setText(_translate("MainWindow", "Start"))

```

圖 4-17. UI layout

如圖 4-17，這是 Ui layout 程式使用 PyQt5 創建了一個用於姿勢檢測的 GUI 應用程序。它包括了兩個按鈕：Start（開始）和 Finish（結束），以及一個文本標籤來顯示檢測結果。當用戶按下 Start 按鈕時，程式會啟動一個線程，其中運行了一個名為 `_detect` 的函式，該函式通過調用操作系統的 `os.system` 命令來啟動 `pose_camera.py` 腳本，這是用於進行實時影像處理的。同時，主界面上的定時器每 400 毫秒觸發一次 `onTimer` 函式。這個函式檢查兩個文件是否存在：`normal.txt` 和 `hunchback.txt`。如果 `normal.txt` 存在，則在 UI 上顯示 "Pose: Normal"；如果 `hunchback.txt` 存在，則顯示 "Pose: Hunchback"。當用戶按下 Finish 按鈕時，程式調用 `os._exit(0)` 退出。總的來說，這個程式提供了一個簡單的界面，用戶可以通過按下按鈕來啟動或停止姿勢檢測，並在 UI 上即時顯示檢測結果。這裡解釋一下，使用了線程的概念來實現非同步的任務執行。在主線程中運行 GUI 的事件循環，而在獨立的線程中執行耗時的任務，例如啟動 `pose_camera.py` 進行影像處理。在這個程式中，當用戶按下 Start 按鈕時，會啟動一個獨立的線程，該線程的目標是執行 `_detect` 函式。這樣可以使主界面

保持響應，不會因為耗時的任務而阻塞。如果不使用線程，而是直接在主線程中執行 `_detect` 函式，則用戶界面可能會凍結，直到 `_detect` 函式執行完畢。使用線程的好處之一是能夠同時執行多個任務如圖 4-18.所示，從而提高程式的效率和響應速度。在這個程式中，即使 `_detect` 函式正在運行，用戶仍然可以與 GUI 進行交互，例如按下 Finish 按鈕來結束程式。

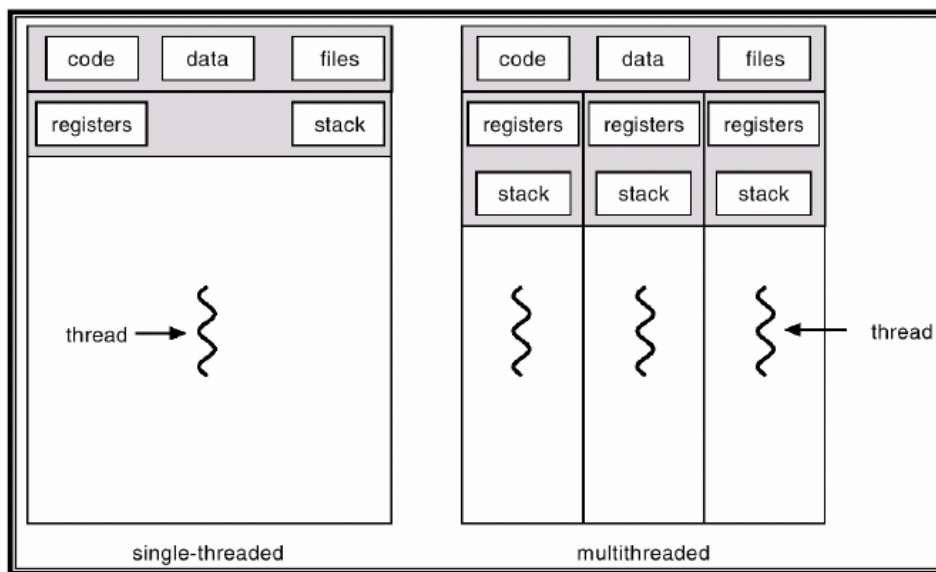


圖 4-18. 線程概念圖【12】

如圖 4-19，這是 UI control 程式使用了 PyQt5 來建立一個 GUI 應用程式，用於姿勢檢測。它包括了兩個按鈕：Start（開始）和 Finish（結束），以及一個文本標籤來顯示檢測結果。當用戶按下 Start 按鈕時，程式會啟動一個線程，其中運行了一個名為 `_detect` 的函式，該函式通過調用操作系統的 `os.system` 命令來啟動 `pose_camera.py` 腳本，這可能是用於進行實時影像處理的。同時，主界面上的定時器每 400 毫秒觸發一次 `onTimer` 函式。這個函式檢查兩個文件是否存在：`normal.txt` 和 `hunchback.txt`。如果 `normal.txt` 存在，則在 UI 上顯示 "Pose: Normal"；如果 `hunchback.txt` 存在，則顯示 "Pose: Hunchback"。當用戶按下 Finish 按鈕時，程式調用 `os._exit(0)` 退出。


```
import sys
import os
import test as ui
from PyQt5 import QtWidgets
from PyQt5.QtCore import QTimer
import threading

def _detect():
    os.system("python3 pose_camera.py")

t = threading.Thread(target=_detect)

class MainWindow(QtWidgets.QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.uiMain=ui.Ui_MainWindow()
        self.uiMain.setupUi(self)
        self.setWindowTitle('Pose Detection')
        self.counter = 0
        self.mytimer = QTimer(self)
        self.mytimer.timeout.connect(self.onTimer)
        self.mytimer.start(400)
        self.uiMain.start.clicked.connect(self.onButtonClick)
        self.uiMain.finish.clicked.connect(self.onButtonClick1)
```

```

def onButtonClick(self):
    t.start()

def onButtonClick1(self):
    os._exit(0)

def onTimer(self):
    path1 = "normal.txt"
    path2 = "hunchback.txt"
    self.counter += 1
    print("Timer "+str(self.counter))

    isExist = os.path.exists(path2)
    isExist1 = os.path.exists(path1)
    if isExist==True:
        self.uiMain.label.setText("Pose:Hunchback")
    elif isExist1==True:
        self.uiMain.label.setText("Pose:Normal")

```

```

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MainWindow()
    window.show()
    sys.exit(app.exec_())

```

圖 4-19. UI control

4.5 解決問題

在我們製作專題的過程中，遭遇了兩個最主要的問題。首先，我們無法成功編譯和執行 `posenet.py` 腳本，如圖 4-20 所示，主要是因為我們使用的環境缺少樹莓派的指令集，我們可以得知樹莓派指令集為 `armv7l`，只需要複製 `armv7a` 裡面指令集過去就能使用如圖 4-21。這導致了編譯和執行過程中的問題，使得我們無法順利進行開發和測試。其次，我們遇到了 `tflite_runtime` 更新以及 `libedgetpu.so` 版本不匹配的問題。它需要 `libedgetpu` version 14 才能正常使用於 linux，但我們是 `libedgetpu` version 16 如圖 4-22，這導致了我們無法正常執行 `posenet.py`。由於更新的 `tflite_runtime` 和現有的 `libedgetpu.so` 版本不相容。

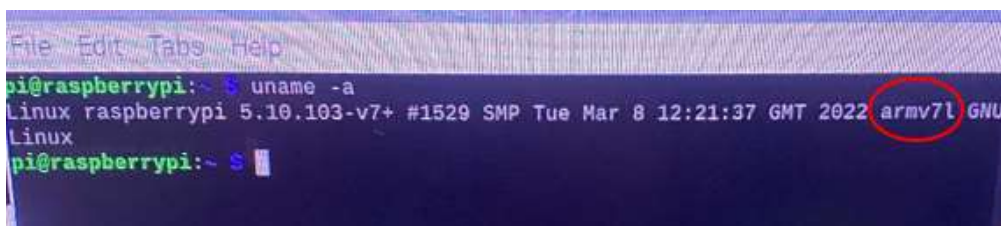


圖 4-20. 解決方法(1)

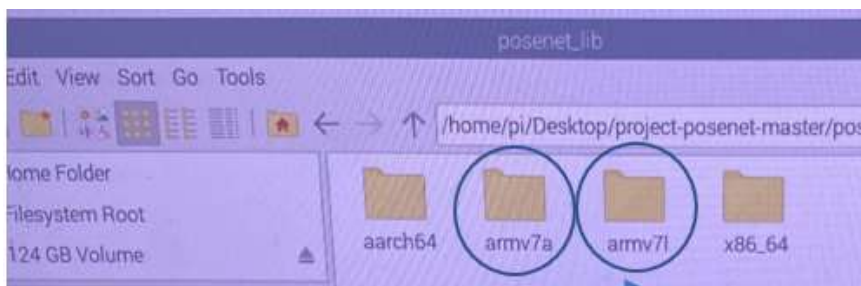


圖 4-21. 解決方法(2)

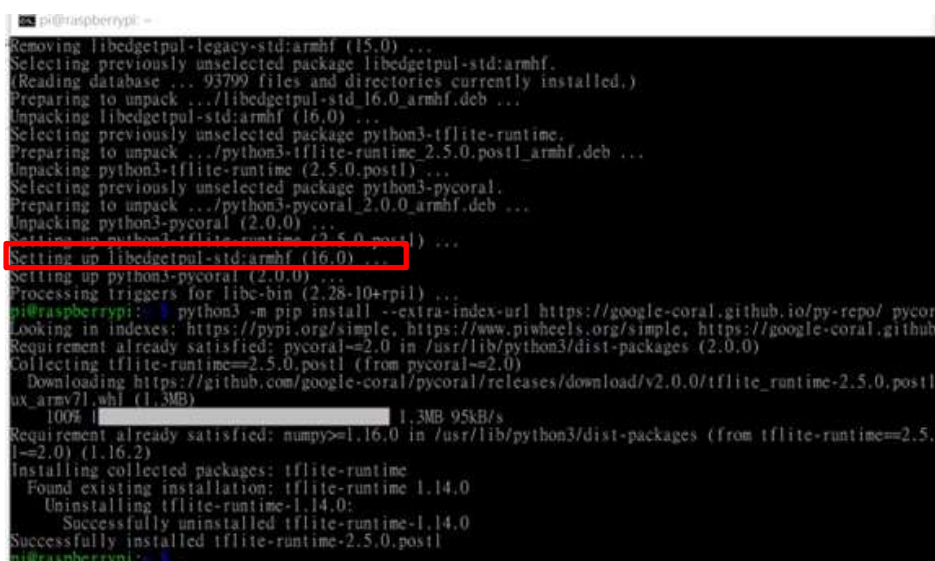


圖 4-22. 解決方法(3)

第五章、實驗成果

5.1 即時駝背偵測

使用機器學習模型進行人體姿勢辨識是一個有效的方法，它可以根據攝影機拍攝的影像，透過 PoseNet 模型對每一幀影像進行分析。我們利用模型輸出的數值來計算每個人不同的耳跟肩距比值，從而準確判斷當前坐姿是否正常或駝背，如圖 5-3 所示。這個功能能夠即時、精確地捕捉使用者的坐姿狀態。在正常坐姿的情況下（如圖 5-1 所示），我們可以在終端機上顯示數值，例如耳距為-133、肩距為-277。然後我們取其絕對值並相除，得到的比值為 2.11，大於設定的閾值 1.6，因此被判定為正常（normal），如圖 5-3 所示。而在駝背坐姿的情況下（如圖 5-2 所示），終端機顯示的數值為耳距-187、肩距-283。我們同樣取其絕對值並相除，得到的比值為 1.51，小於設定的閾值 1.6，因此被判定為駝背（hunchback），如圖 5-3 所示。這樣的系統能夠即時準確地提醒使用者，使其注意坐姿，達到保護脊椎的目的。

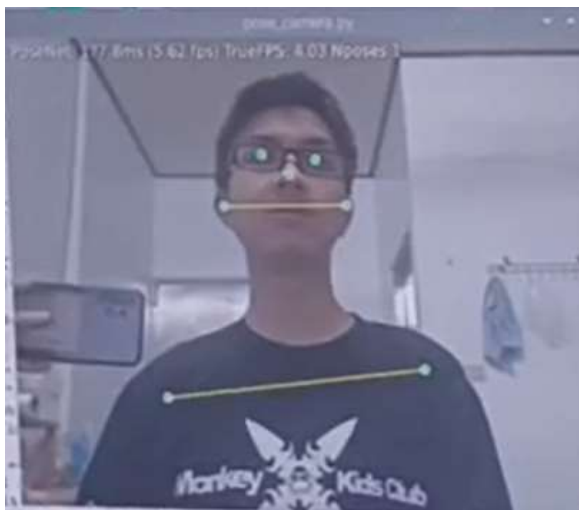


圖 5-1. 偵測畫面正常坐姿

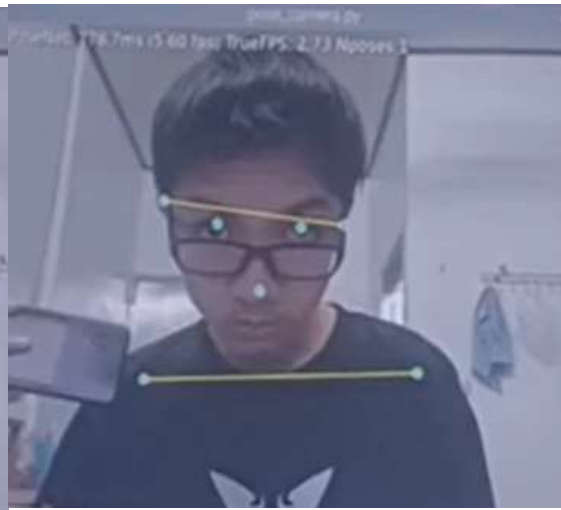


圖 5-2. 偵測畫面駝背坐姿



圖 5-3. 偵測畫面正常坐姿(1)



圖 5-4. 偵測畫面正常坐姿(2)

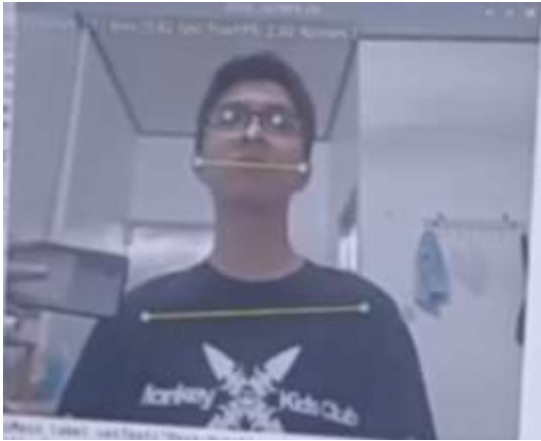


圖 5-5. 偵測畫面正常坐姿(3)



圖 5-6. 偵測畫面駝背坐姿(4)



圖 5-7. 偵測畫面駝背坐姿(5)

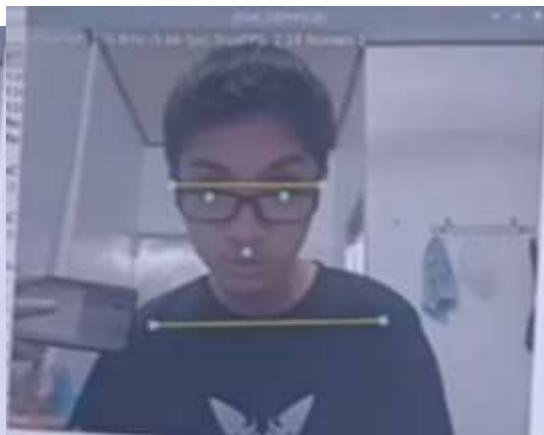


圖 5-8. 偵測畫面駝背坐姿(6)

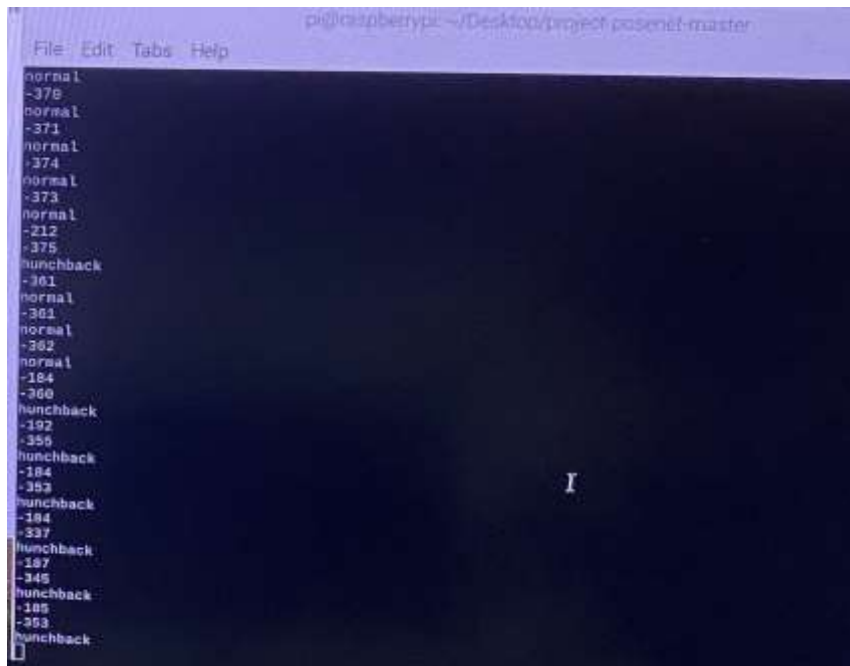


圖 5-9. 即時辨識駝背數據

5.2 UI 介面顯示提醒

為了讓使用者方便且清晰了解自身的坐姿，我們特別設計了一個 UI 介面，以更直觀的方式呈現坐姿狀態。這個 UI 介面通過 PoseNet 模型捕捉特定值，並根據駝背運算的結果，能夠準確且即時地辨別使用者的坐姿狀態。當系統檢測到駝背時，會生成一個標記駝背的文字檔；同樣地，當系統檢測到正常坐姿時，也會生成相應的文字檔。這些文字檔會顯示在 UI 介面的右下角方框中，分別標記為 "normal" 和 "hunchback"，如圖 5-4 和 5-5 所示。這樣的設計使得使用者能夠即時瞭解自己的坐姿狀態，無需專業知識即可清楚地識別。透過這個 UI 介面，使用者可以更加主動地調整自己的坐姿，以保持良好的身體姿勢，避免駝背問題的發生。

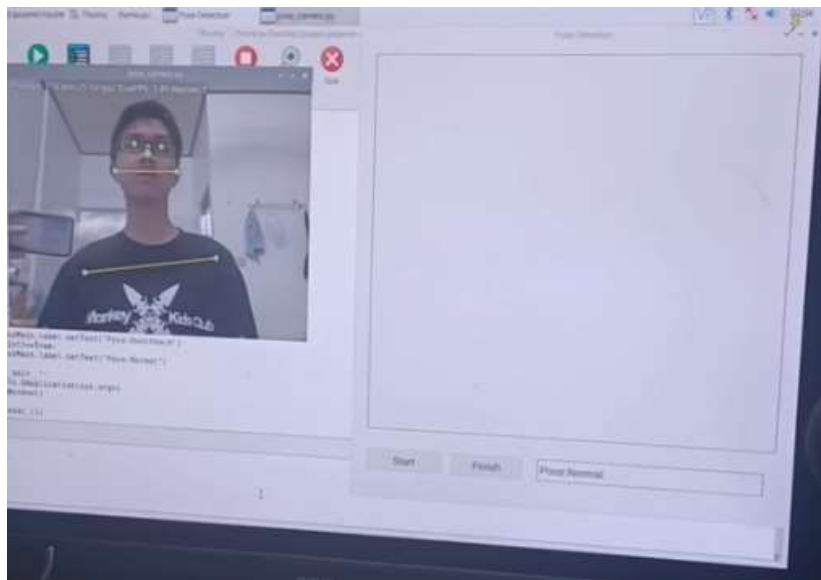


圖 5-10. UI 介面(正常坐姿)

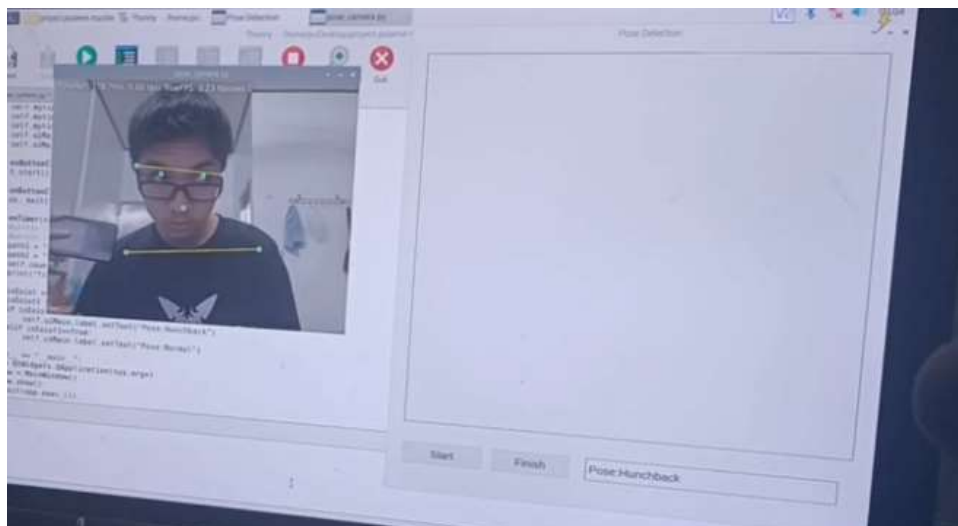


圖 5-11. UI 介面(駝背坐姿)

5.3 LED 提醒

通過攝影機讀取影像，我們能夠即時監測使用者的坐姿狀態。當檢測到使用者呈現駝背姿勢時，系統會啟動燈光提示，如圖 5-7 所示；反之，如果使用者的坐姿正常，如圖 5-6 所示，則燈光處於熄滅狀態。透過這種燈光提示方式，我們能夠有效地提醒使用者改善坐姿，促進良好的姿勢習慣。

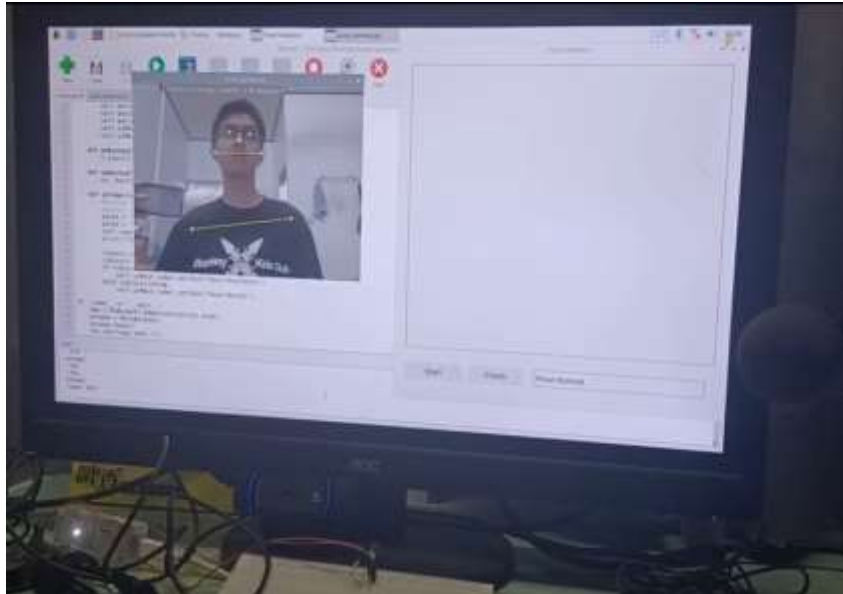


圖 5-12. LED 提醒(正常坐姿)

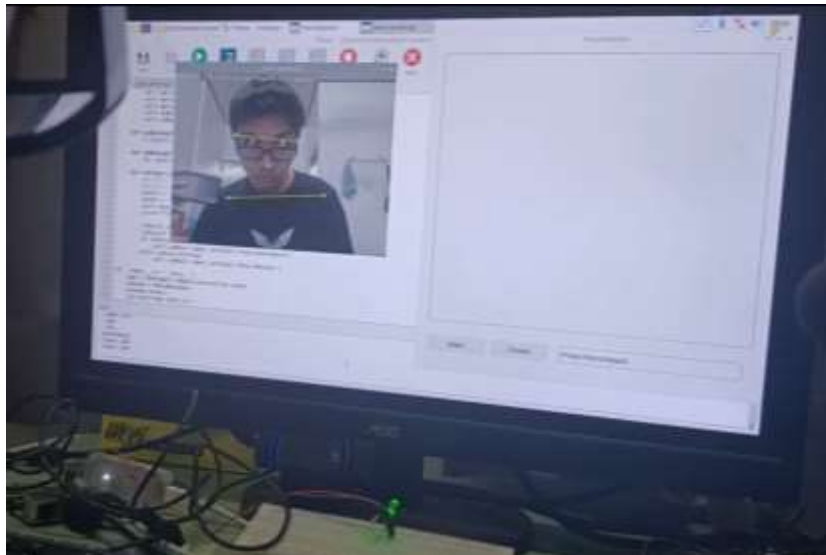


圖 5-13. LED 提醒(駝背坐姿)

第六章、結論與未來展望

6.1 結論

這款智慧檯燈系統經過我們組員間的多次測試，確實能夠即時性判斷使用者是否出現駝背姿勢，並以 LED 燈亮或暗的方式提醒使用者。然而，每個人的身體構造不同，導致雙耳和雙肩之間的距離各異。因此，在系統初次使用時，我們必須先記錄一個正確的坐姿，以計算雙耳和雙肩之間的比例值。系統根據這些設定，當偵測到使用者的坐姿出現駝背時，LED 燈會亮起作為警示燈，同時在介面上顯示駝背狀態。反之，如果坐姿正常，則介面上僅顯示正常狀態，而 LED 燈則保持暗淡。這種智慧檯燈不僅能夠提供即時的坐姿警示，還可以根據每個使用者的身體特徵進行個性化設定，提供更貼心的使用體驗。

6.2 未來展望

因為時間太緊湊我們來不及將 LED 更換成檯燈，將來我們的智慧檯燈系統將進一步演進，像是將 LED 轉換成檯燈且將人機介面更完善讓相機抓取的影像能鑲入 UI 介面的裡面，還有引入更多創新功能，以提供更全面、更人性化的坐姿監測和改善方案。其中，我們計劃加強語音功能，讓系統不僅通過 LED 警示燈提醒使用者，還會通過語音提示方式，更直觀地提醒使用者注意坐姿。同時，我們將實現自動正常狀態設定，無需使用者手動干預，提高系統的便捷性。此外，系統還將支持多方位攝影機安裝，不再局限於正面，以更全面地監測使用者的坐姿。我們還將加強對其他不良坐姿的偵測，如翹腳、低頭等，並針對不同的錯誤姿勢提供相應的語音提示和改善建議。這些改進將使我們的智慧檯燈系統更加智能、實用，更有效地幫助使用者改善坐姿習慣，預防駝背等健康問題的發生，提高生活質量和健康水平。

第七章、參考資料

- [1] “人體姿態辨識 Towards Accurate Multi-person Pose Estimation in the Wild”, 2020, <https://williamchiu0127.medium.com/%E8%AB%96%E6%96%87%E9%96%B1%E8%AE%80%E7%AD%86%E8%A8%98-%E4%BA%BA%E9%AB%94%E5%A7%BF%E6%85%8B%E8%BE%A8%E8%AD%98-towards-accurate-multi-person-pose-estimation-in-the-wild-9af7ae3b467d>
- [2] “Raspberry Pi 3 Model B”, 台灣樹梅派, <https://piepie.com.tw/10684/raspberry-pi-3-model-b>
- [3] “羅技 C270 WebCAM 網路攝影機”, logitech, https://www.thebest.com.tw/product_d.php?lang=tw&tb=1&cid=361&id=1753
- [4] “Coral USB Accelerator”, 萊思禮有限公司, <https://ricelee.com/product/google-usb-accelerator>
- [5] “Real-time Human Pose Estimation in the Browser with TensorFlow.js”, TensorFlow, 2018, <https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5>
- [6] “PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model”, arxiv, 2018, <https://arxiv.org/abs/1803.08225>
- [7] “Raspberry PI4 + Coral USB + posenet 姿態識別”, HackMD, https://hackmd.io/@0p3Xnj8xQ66IEl0EHA_2RQ/H1dgMOx1L
- [8] “project-posenet”, mtyka, <https://github.com/google-coral/project-posenet>
- [9] <https://www.wpgdadatong.com/blog/detail/72102>
- [10] <https://www.cnblogs.com/xleng/p/10948838.html>
- [11] <https://zhuanlan.zhihu.com/p/91435213>
- [12] <https://williamchiu0127.medium.com/%E8%AB%96%E6%96%87%E9%96%B1%E8%AE%80%E7%AD%86%E8%A8%98-%E4%BA%BA%E9%AB%94%E5%A7%BF%E6%85%8B%E8%BE%A8%E8%AD%98-alphapose-rmpe-regional-multi-person-pose-estimation-229e75829bab>
- [13] iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天 (ithome.com.tw)
- [14] “人體姿態辨識 OpenPose”, 2020, <https://williamchiu0127.medium.com/%E8%AB%96%E6%96%87%E9%96%B1%E8%AE%80%E7%AD%86%E8%A8%98-%E4%BA%BA%E9%AB%94%E5%A7%BF%E6%85%8B%E8%BE%A8%E8%AD%98-openpose-f8c2d5d651e2>
- [15] <https://discuss.tf.wiki/t/topic/1844>
- [16] “,AttributeError: 'Delegate' object has no attribute 'library' · Issue #167 · google-coral/edgetpu · GitHub