

個人專題研究報告

研究主題

基於深度學習-人臉識別考勤系統 v1

研究生：梁家銘

執導教授：郭家旭

目錄

個人專題研究報告	1
研究主題	1
目錄	2
研究動機	3
研究目的	3
系統架構設計	4
系統設計	5
系統部署	7
研究方法與設計	8
辨識畫面的呈現	8
影像處理	8
人臉偵測	8
人臉關鍵點訓練	9
人物識別	9
預期解果	10
系統展示(目前進度)	10
已知人物的識別	10
未知人物的識別	10
問題與解決方案(目前進度)	11
人物識別誤判問題	11
參考資料	12

研究動機

隨著科技日新月異，深度學習在各種不同領域成效令人驚嘆，對於影像辨識的應用亦是普遍廣泛，如人臉辨識考勤系統，許多公司企業也陸續的將這門技術引入自家公司中使用，相比傳統的打卡、點名和簽到等方式，數位化的人臉識別考勤系統，節省了許多成本不必要的浪費，如打卡紙，墨水等。同時人員出勤的數位化也為大型公司省下了不少資料整理時間。

根據目前市場上人臉識別考勤系統使用者給出的回饋，表示目前的系統技術仍然還有改善的空間，例如在鏡頭角度、臉部表情與場景光線等影響下，系統還是很難正確的做出辨識，也因為如此或許我們能夠參考在 2012 年在 imageNet 大賽中拿下優異成績的 AlexNet[1]深度卷積模型架構，以及 Google 在 2015 年 LWF 人臉資料庫以 99.63%的最佳成績刷新了記錄的 FaceNet[2]架構，來建構 CNN 模型，並嘗試不同的模型訓練方法來提高人物辨識的成功率，改善目前市場上系統在識別中受到光線、角度與表情等因素影響的問題。

研究目的

- 學習人工智慧**模型訓練**的相關技術知識。
- 學習**硬體**對神經網路加速的相關知識。
- 學習人工智慧結合物聯網(AIoT)的相關應用。
- 改善市場上人臉識別考勤系統已知的問題，提升正確辨識的成功率。

系統架構設計：

系統架構包含 3 個階段(Phases)，如圖 1-1 所示：

- Phases 1: Web Cam 負責收集影像數據，並搭配顯示器呈現拍攝畫面。
- Phases 2: Client 提供網頁使用者介面，給予使用者下載 User 考勤相關資料。
- Phases 3: Server 負責神經網絡運算與回應 Client 端的資料提取請求。

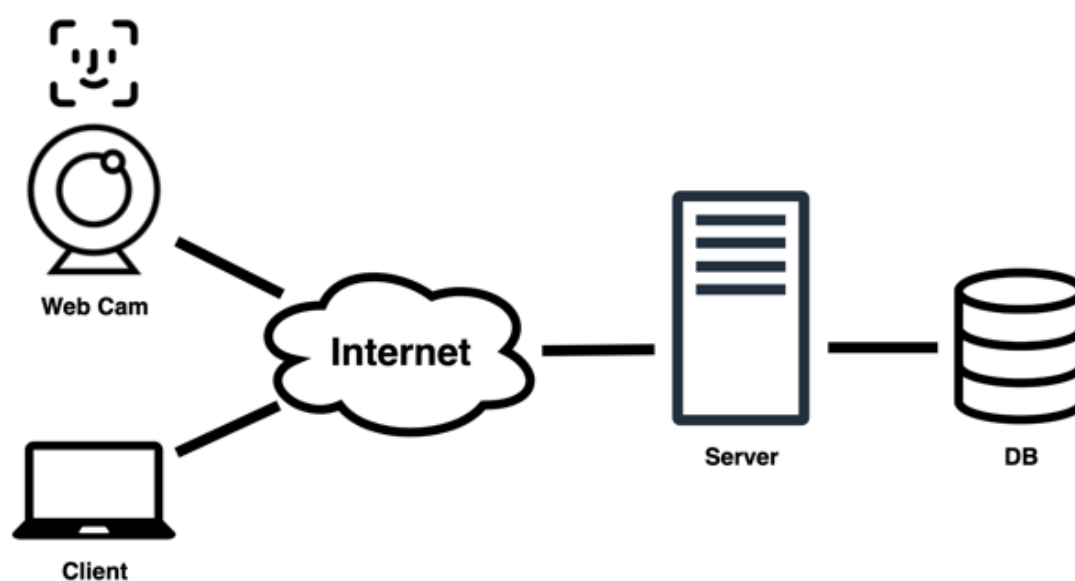


圖 1-1:系統架構 (資料來源:本研究)

系統設計

- Web Cam：採用 C++ 語言開發，負責收集影像數據。設備使用 ESP32-CAM 無線攝像頭收集影像數據，如圖 1-2 所示，並搭配 ILI9341 TFT LCD 螢幕呈現拍攝畫面，如圖 1-3 所示。



項目	規格內容
SPI Flash	默認為 32Mbit
RAM	內置 520 KB +外部 4MPSRAM
藍牙	藍牙 4.2 BR / EDR 和 BLE 標準
Wi-Fi	802.11b / g / n / e / i
IO 端口	9
VCC	3.3V~5V
功耗	6mA @ 5V ~ 310mA @ 5V
工作溫度	-20°C~85°C
尺寸	27 * 40.5 * 4.5 (±0.2) mm

圖 1-2:ESP32-CAM 無線攝像頭模組 (資料來源：本研究)



項目	規格內容
顯示尺寸	2.8 吋
解析度	240 × RGB × 320(TFT) dots
LCD 類型	TFT, 白色, 透射式
驅動晶片	ILI9341
模組介面	4-wire SPI interface
VCC	3.3V~5V
功耗	約為 90mA
工作溫度	-10°C~60°C
有效顯示區域	43.2x57.6(mm)
PCB 底板尺寸	50x86(mm)

圖 1-3:ILI9341 TFT LCD 模組 (資料來源：本研究)

系統設計

- Client 端：採用 **React 網頁框架**以 **Typescript 語言**開發，React 框架透過維護一個虛擬 DOM 的方式，在畫面需要渲染時與真實 DOM 做比較，並且只針對有需要更新的地方才進行運算，以此達到畫面渲染效率的提升，工作原理如圖 1-4 所示。Clnet 端主要負責與 Server 的溝通，提供多種考勤資料下載的相關功能。

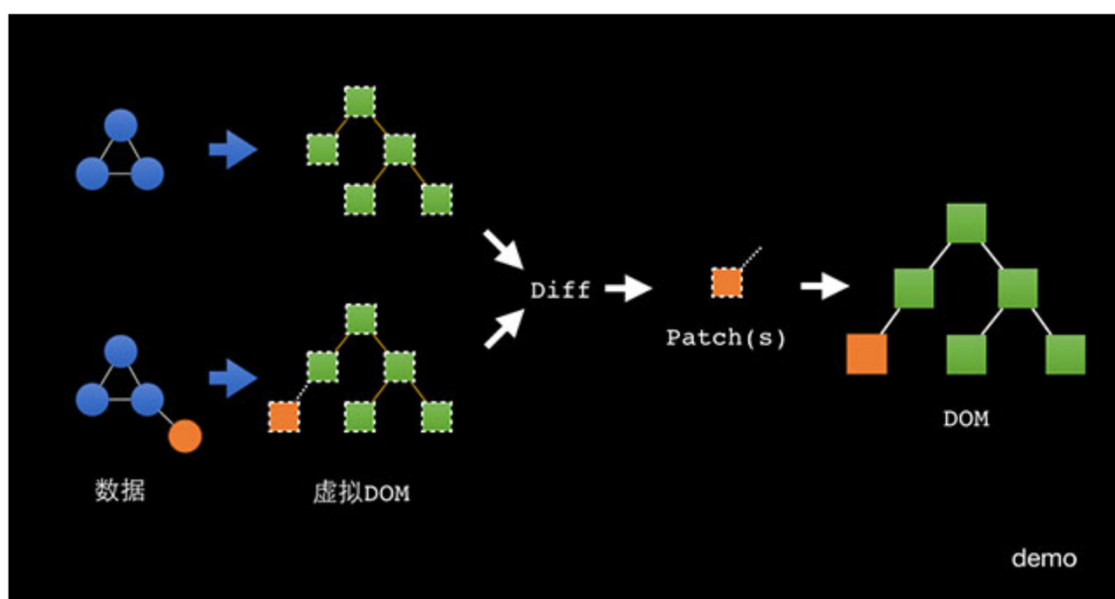


圖 1-4:React 工作原理圖 (資料來源:天天看點, 淺談 React 工作原理)

- Server 端：採用 **Python 語言**開發，負責人物識別神經網路運算，並且提供識別運算與資料下載的相關接口。神經網路模型採用 Dlib 軟件庫所提供的，人臉偵測模型與臉部識別模型，搭配筆記本內建 NVIDIA GTX 1650 顯示卡來加速運算，規格如圖 1-5 所示。資料庫使用 MySQL 關聯式資料庫，更好的管理成員與部門的相關數據。

項目	規格內容
CUDA 核心	1024
圖型提升時脈	1560 MHz
記憶體	4 GB
驅動晶片	ILI9341
記憶體介面	128 位元

圖 1-5: NVIDIA GTX 1650 規格表 (資料來源：本研究)

系統部署

本系統部署圖如圖 1-5 所示，系統各組件工作描述如圖 1-6 所示。

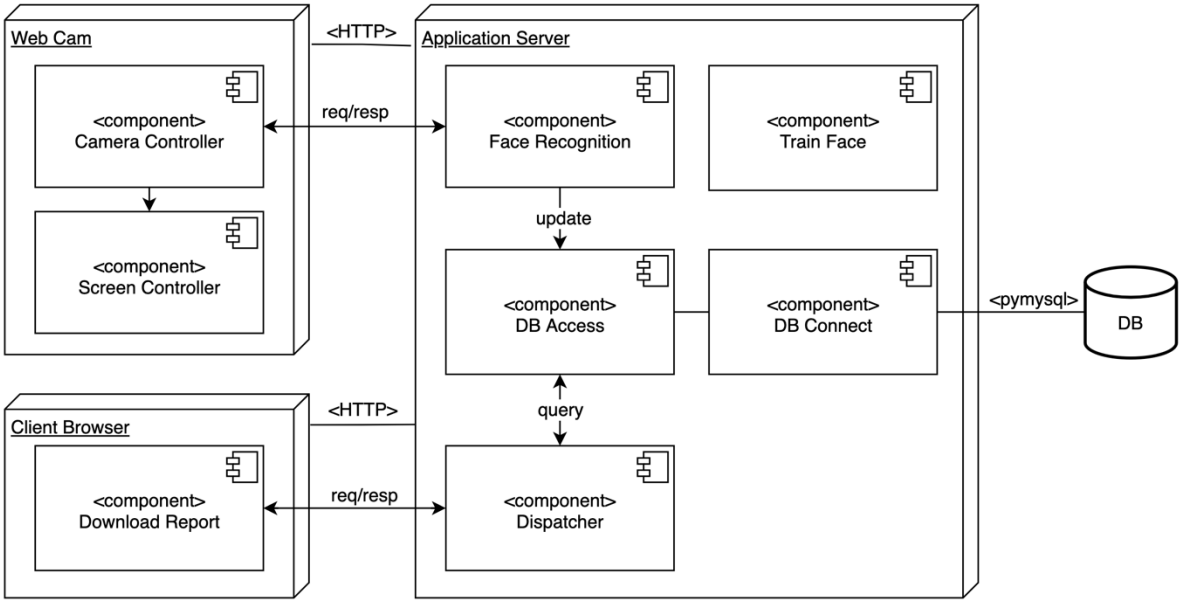


圖 1-5:系統部署架構圖 (資料來源:本研究)

Component	Description
Camera Controller	負責影像數據的收集。
Screen Controller	負責拍攝影像畫面的呈現。
Download Report	提供考勤資料下載的功能。
Face Recognition	負責人臉識別神經網路運算。
Train Face	負責人臉識別模型的訓練。
DB Access	維護資料庫事務，提供封裝好的資料庫操作。
DB connect	負責資料庫的連接，提供資料庫增、刪、改、查操作。
Dispatcher	負責將 Client 端的請求分派給對應的資料庫操作方法。

圖 1-6:系統組件工作描述表 (資料來源:本研究)

研究方法與設計

辨識畫面的呈現

基於 Ajax 工具開發

Web Cam 端利用人物識別的 API 接口，將拍攝畫面送到到 Server 辨識，並在完成辨識後，再將標註好辨識結果的圖片回傳給 Web Cam 呈現畫面。

影像處理

基於 OpenCV 工具開發

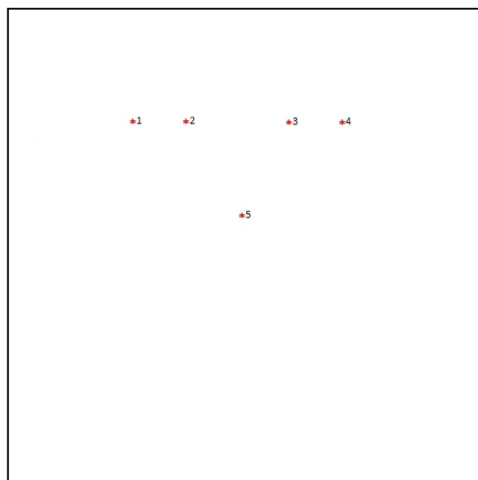
OpenCV 具有豐富的影像處理函式庫，能夠很好的幫助我們在影像上的操作，如解析度放大縮小、臉部辨識結果的畫面呈現。

人臉偵測

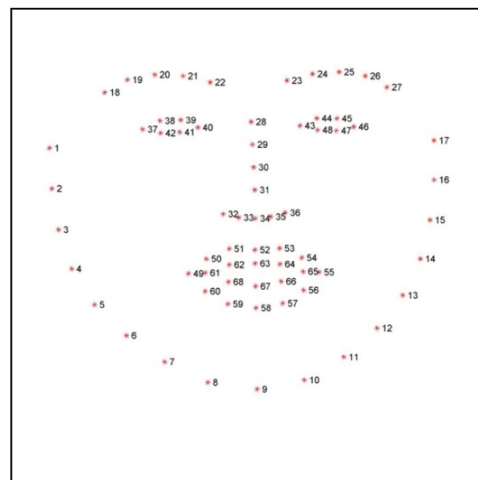
基於 Dlib/OpenCV 工具開發

使用 Dlib 軟件庫所提供的，深度卷積人臉偵測模型，此模型是在一個已標註好關鍵點的資料集上做訓練，最終達到能夠預測出人臉上五官的位置。

人臉關鍵點偵測分為以下兩種，如圖 1-7 所示。



5-points facial landmarks



68-points facial landmarks

Facial Landmarks	Description
5-points	使用左眼頭尾、右眼頭尾、鼻頭這五個點來做辨識
68-points	使用外輪廓、眉毛、眼睛、鼻子、嘴巴共 68 個點來做辨識

圖 1-7:影像關鍵點 (資料來源: iT 邦幫忙, 山姆大叔)

人臉關鍵點訓練

基於 Dlib/Numpy/OpenCV 工具開發

使用 Dlib 軟件庫所提供的，**臉部識別模型**。該模型在 Wild 基準的 Labeled Faces 上的準確率為 99.38%。本系統使用 **68-points 關鍵點**作為模型訓練參數如圖 1-8 所示，利用**影像抖動**的方式增加已知人物訓練樣本數量，來訓練人物的關鍵點數據。

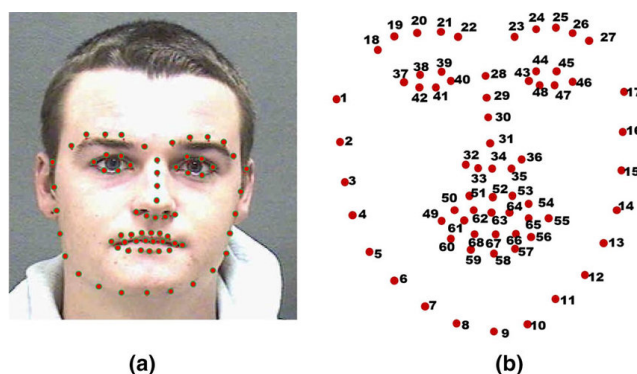


圖 1-8:68-Points 臉部關鍵點圖 (資料來源:ResearchGate, Ali Ahmed Elmahmudi)

人物識別

基於 Dlib/Numpy/OpenCV 工具開發

使用 Dlib 軟體庫所提供的，人臉偵測模型與臉部辨識模型預測，分為以下三步驟：

- **人臉偵測**：將 Web Cam 所拍攝到的畫面作為**人臉識別模型**的輸入，得出畫面中的人臉位置。
- **臉部識別**：在將人臉畫面作為**臉部識別模型**的輸入，得出人物的臉部關鍵點數據。
- **人物識別**：利用已知人物關鍵點數據減去拍攝畫面中的人臉關鍵點數據，得出誤差值，並在系統上**設定一個識別閾值**，若誤差小於該閾值，則視為已知人物，如圖 1-9 所示。

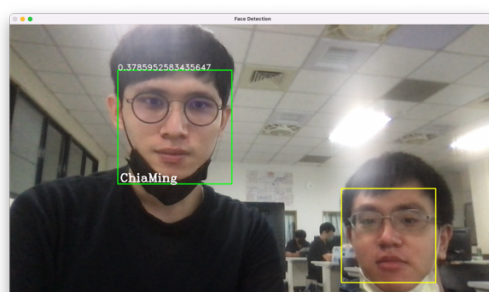


圖 1-9:人物識別結果 綠框為已知人物 黃框為未知人物 (資料來源:本研究)

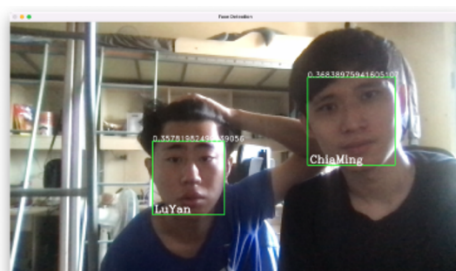
預期解果

- 在鏡頭距離人物 50 cm 內能夠 100%成功擷取人臉特徵
- 在光線充足的空間中能夠達到 95%以上的成功辨識

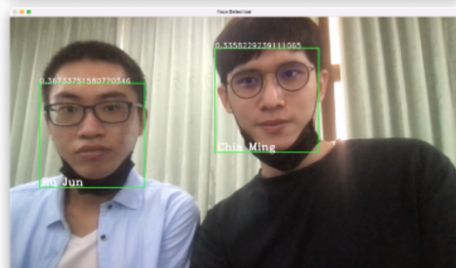
系統展示(目前進度)

已知人物的識別

光線較暗：

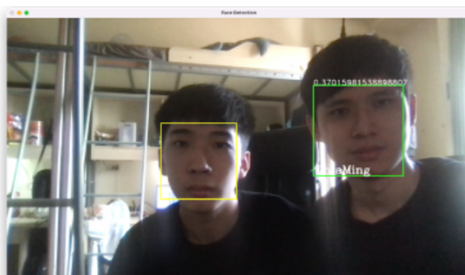


光線較亮：

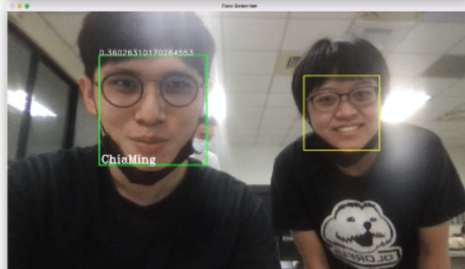


未知人物的識別

光線較暗：



光線較亮：



問題與解決方案(目前進度)

人物識別誤判問題

- 問題描述：在剛開始的時候，已知人物訓練資料採用單一樣本進行訓練，導致會把未知的人物也辨識為已知的某位人物。
- 解決方案：在訓練已知的人物時設置影像抖動參數，將訓練資料透過隨機偏移生成新的訓練資料，以此來增加訓練的樣本數量，並且降低關鍵點誤差的閾值。在設置抖動參數將單一樣本提升至 50 個樣本以及，將關鍵點誤差閾值由原本的 0.6 降至 0.4 後，解決了已知人物誤判的問題。

參考資料

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks

<https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>

[2] Florian Schroff, Dmitry Kalenichenko, James Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering

<https://arxiv.org/abs/1503.03832>