

運用深度學習技術於圖書館  
人流監控

銘傳大學

一一二學年度

資 訊 工 程 學 系

# 運用深度學習技術於圖書館人流監控

## 摘要

圖書館被認為是大學的心臟，且隨著功能趨於多元化，一直是校園中人潮匯集的學習場域。面對每日進進出出的龐大人流，館員如何有效管理並確保館內安全，乃是重要的課題。近年來，人工智慧蓬勃發展，已被廣泛應用於諸多領域。如人臉辨識常運用在資安或金融等領域。而該項技術結合物件偵測及追蹤，再透過與已知的人臉資料庫進行匹配，最終實現身分辨識的目的。

本專題旨在利用深度學習技術，實現對圖書館內人流的偵測與分析。以<<銘傳大學桃園校區圖書館>>為對象，設計館內人流監控系統。利用鏡頭捕捉即時影像，並透過深度學習技術偵測館內人員。使用者藉此系統取得館內人流數據。從而更好理解與滿足讀者需求，進而提升閱覽品質。

本專題的成果有望為圖書館管理方式帶來更多的改進方向，藉此提升讀者體驗並提高圖書館的行政效率。

關鍵詞：人臉辨識、物件追蹤、人臉偵測。

# 目錄

摘要.....	I
目錄.....	II
圖目錄.....	V
表目錄.....	VII
第一章 緒論.....	1
1-1 研究背景與動機 .....	1
1-2 研究目的 .....	1
第二章 文獻探討 .....	2
2-1 機器學習(Machine Learning) .....	2
2-1-1 監督式學習(Supervised learning) .....	3
2-1-2 非監督式學習(Unsupervised Learning).....	3
2-1-3 半監督式學習(Semi-supervised Learning) .....	3
2-1-4 強化式學習(Reinforcement Learning) .....	3
2-2 深度學習(Deep Learning).....	4
2-2-1 Tensorflow .....	4
2-2-2 PyTorch.....	4
2-2-3 DarkNet .....	4
2-3 人工神經網路(Artificial Neural Network, ANN) .....	5
2-3-1 遞迴神經網路(Recurrent Neural Network, RNN) .....	5

2-3-2 卷積神經網路(Convolutional Neural Network, CNN) .....	6
2-3-3 MTCNN.....	6
2-4 人臉辨識(Face Recognition).....	7
2-4-1 OpenCV .....	7
2-4-2 FaceNet.....	7
2-5 物件追蹤(Multiple Object Tracking).....	8
2-5-1 DeepSort .....	8
2-5-2 Re-ID .....	8
2-6 系統開發工具與技術 .....	8
2-6-1 相關程式語言 .....	9
2-6-2 相關函式庫 .....	10
2-6-3 相關軟體與工具 .....	11
<b>第三章 統分析與設計 .....</b>	<b>14</b>
3-1 需求調查 .....	14
3-2 系統功能分析 .....	14
3-3 介面設計 .....	19
3-3-1 主畫面設計 .....	19
3-3-2 查詢畫面設計 .....	21
3-3-3 人流統計畫面設計 .....	21
3-3-4 管理員設定畫面設計 .....	22

3-4 資料庫邏輯設計 .....	23
<b>第四章 系統實作與測試 .....</b>	<b>26</b>
4-1 開發環境 .....	26
4-1-1 硬體設備 .....	26
4-1-2 使用軟體與工具 .....	26
4-2 系統架構 .....	27
4-2-1 MTCNN 人臉偵測.....	28
4-2-2 DeepSort 物件追蹤.....	29
4-2-3 FaceNet 人臉辨識.....	29
4-3 系統測試 .....	30
<b>第五章 結論.....</b>	<b>33</b>
5-1 研究成果 .....	33
5-2 未來展望 .....	33
<b>參考文獻.....</b>	<b>35</b>
<b>附錄.....</b>	<b>41</b>
附錄一 系統安裝說明.....	41
附錄二 專案時程與分工.....	44

# 圖目錄

圖 2-1 AI 與機器學習關係圖 .....	2
圖 2-2 CNN 架構圖 .....	6
圖 2-3 FaceNet 運作流程圖 .....	7
圖 2-4 Mac OS 版 Qt Designer 介面.....	13
圖 3-1 即時人流監控活動圖 .....	15
圖 3-2 偵測特殊人士活動圖 .....	16
圖 3-3 閉館人流檢查活動圖 .....	17
圖 3-5 主畫面 .....	19
圖 3-6 偵測特殊人士 .....	20
圖 3-8 閉館檢查結束 .....	20
圖 3-9 查詢畫面 .....	21
圖 3-10 人流統計畫面 .....	22
圖 3-11 管理員設定畫面 .....	23
圖 3-12 資料庫邏輯設計 .....	24
圖 4-1 系統架構圖 .....	28
圖 4-2 MTCNN 人臉偵測 .....	29
圖 4-3 DeepSort 物件追蹤 .....	29

圖 4-4 FaceNet 人臉辨識.....	30
附圖 1 程式組工作時程甘特圖.....	44
附圖 2 文件組工作時程甘特圖.....	45



# 表目錄

表 4-1 系統測試準確率 .....	30
表 4-2 進出館各原因錯誤率占比 .....	32

# 第一章 緒論

## 1-1 研究背景與動機

大學乃知識的殿堂，而圖書館則是知識累積與傳播的核心。銘傳大學圖書館提供豐富的圖書資源及舒適的閱覽環境，已成為校園中深受歡迎的學習場域。但館內秩序僅靠人力管控卻也難保萬無一失，維持館內秩序除了能保證閱讀環境，亦可便於館方維護所提供的各項服務及資源。

此外據圖書館統計，每日進館人次約兩千人，入館者除校內教職員生外，也有不少校外人士。面對眾多入館者，如何掌握館內人流狀況並有效管理特殊人士，乃圖書館的重要課題。希望這項開發能提高圖書館的運作效率，以協助讀者及館方人員共創更美好的閱讀環境。

## 1-2 研究目的

近年來，人工智慧相關技術深受各界重視，尤其是深度學習的應用更如雨後春筍般蓬勃發展。本專題擬整合現有物件辨識與追蹤等技術，以銘傳大學圖書館為實驗對象，建置一套人流監控系統。本系統的主要目的有三：首先，館員可以瞭解館內人流的進出狀況，以作為人力調度與服務調整的依據；師生可藉由圖書館網頁自遠端瞭解館內人流，以作為是否親自到館的參考。其次，本系統還能針對特定人士進行管理，一旦有列管人士入館，館員能在第一時間掌握與因應。最後，因銘傳大學師生出館時無需刷卡，本系統能協助館員在閉館時清查館內師生是否皆已正常離館。

## 第二章 文獻探討

機器學習與深度學習技術在影像處理領域已然取得顯著的進展，使人臉辨識與物件追蹤等應用表現卓越。兩項技術的結合不僅對各領域，如安全監控、自動駕駛、機器人技術等，帶來重大的影響，同時也為我們的生活提供諸多便利。本文將深入討論人臉辨識及物件追蹤等相關技術。

### 2-1 機器學習(Machine Learning)

機器學習是人工智慧(AI)的子集，如圖 2-1 [1]，是一門開發演算法和統計模型的科學，透過這些演算法和模型，人類無須下達精確指令，電腦系統即可根據模式和推理來執行任務 [2]。其主要訓練電腦運用演算法及模型持續在龐大的資料中進行學習，同時透過每次的經驗改善以達到逐步提升預測準確率的目的，並將所獲得的資訊運用至當下的情況，藉此做出合適的決策 [3]。

接下來將分別介紹機器學習根據訓練方式可大致分成四種學習演算法：監督式學習、非監督式學習、半監督式學習及強化式學習等 [2]。

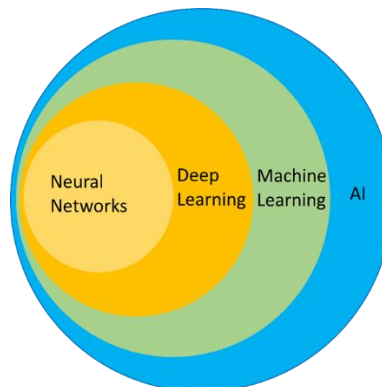


圖 2-1 AI 與機器學習關係圖

資料來源：SAP [1]

### **2-1-1 監督式學習(Supervised learning)**

監督式學習需要自行提供分類並標記好的資料給電腦，讓它透過演算法及模型來找出資料中的特徵及關聯，且訓練用的資料須包含正確的輸入輸出，隨著電腦訓練的資料量增加，其正確率也會逐漸上升 [4]。

### **2-1-2 非監督式學習(Unsupervised Learning)**

非監督式學習使用未經分類及標記的資料進行訓練，先輸入所有資料並透過演算法或模型嘗試找出其中的特徵及關聯以進行分群。而它的特性使其在資料探勘初期是十分好用的工具，因所需的資料無須事先進行人工標記，可大幅降低人力需求並快速找出規律，但同時也可能因此造成較大的功耗，更甚至所找到的特徵不具重要性，最終導致無意義的分群結果 [5]。

### **2-1-3 半監督式學習(Semi-supervised Learning)**

半監督式學習結合監督式學習及非監督式學習的訓練方式與優點，當開始訓練時僅取部分訓練資料進行人工標記後，電腦再根據該部分資料找尋特徵並用於辨識及分類剩餘資料。透過半監督式學習可降低人力需求，同時確保仍有較高的準確率，因此逐漸受到重視 [6]。

### **2-1-4 強化式學習(Reinforcement Learning)**

強化式學習近似於非監督式學習的延伸，所有資料皆未經人工標記，須由電腦自行分類，與非監督式學習不同的則是當電腦分類錯誤時，將透過人為設計的獎懲機制引導電腦進行修正，以此逐步改善並提高其預測準確率 [2]。

## 2-2 深度學習(Deep Learning)

深度學習是機器學習的一種方法，主要是以人工神經網路為架構並利用其概念來提取資料特徵，對資料進行表徵學習的演算法，以此透過所得特徵來辨識資料 [7] [8]。

### 2-2-1 Tensorflow

Tensorflow 是一個開源、高性能的深度學習框架，可應用於處理各種感知與自然語言，可透過 Python、C++、Java 等多種語言進行開發，由 Google 於 2015 年推出 [9]。它被設計用於建立及訓練各種機器學習及深度學習模型，Tensorflow 的高階 API，是按照 Keras API 用以定義和訓練類神經網路的標準而建立 [10]。

### 2-2-2 PyTorch

PyTorch 以 Torch 為基礎打造，是一個開源的 Python 機器學習框架，底層由 C++ 實現，主要由 Meta Platforms 的人工智慧研究團隊開發 [11]。PyTorch 的優勢是可以使用切換 CPU 和 GPU 的計算 [12]。由於它語法簡潔、好上手，且後續有推出 TorchScript 以協助使用者在未安裝 Python 的情況下，仍能正常執行 PyTorch，也因此在學術界愈發流行 [13]。

### 2-2-3 DarkNet

DarkNet 是由 YOLO 作者(Joseph Redmon)用 C 和 CUDA 編寫的開源神經網路框架，其特點為易安裝、沒有任何依賴項，同時移植性極佳，相較於 Tensorflow 更加輕型，且支持 CPU 和 GPU 計算。基於上述特性，在影像處理的領域中可以透過結合 OpenCV 以達到更好的成效 [14]。DarkNet 的本意是為了結合 YOLO 用於物體檢測及追蹤等，因此也使得

DarkNet 的應用層面能夠更加廣泛 [15]。

## 2-3 人工神經網路(Artificial Neural Network, ANN)

人工神經網路(ANN)是一種模擬生物神經網路結構與功能的數學模型，採用類似於人腦分層結構中的互連節點或神經元以建立計算模型 [16]。模型中透過神經網路的多層神經元對函式進行估計及近似運算，其中包含一層輸入層、一或多層隱藏層及最後的輸出層，而神經元間具有相應的權重和閾值，當任一節點輸出高於指定閾值，此節點便會被激發，並將資料傳送至下一層，反之則不予以傳送 [17] [18]。

人工神經網路中每層皆可透過不同的設計以應付不同的需求，大致可分為池化層、平坦層及全連接層。池化層主要用於降低參數量級（降維），可降低過擬合問題，其運作方式大致為每節點僅選取上層特定範圍內的資料並進行運算，且過程中可以透過不同的運算方式來決定池化結果，以便提取特徵；平坦層用於將池化層所輸出的特徵拉平及維度的轉換，以便於後續輸入全連接層進行分類；全連接層又稱密集層，表示該層所有神經元皆與上一層的所有神經元相連，主要用於分類，將前面的輸出結果輸入至全連接層，即可透過調整權重及偏差來取得結果 [19]。

### 2-3-1 遞迴神經網路(Recurrent Neural Network, RNN)

遞迴神經網路(RNN)與一般神經網路的區別在於每次都會將上一層的輸出結果輸入到下個隱藏層中一起訓練，因此所有的輸入皆會影響到未來的輸出 [20]。而 RNN 也經常結合長短期記憶模型(long short-term memory, LSTM)使用，相較於一般的 RNN，結合 LSTM 的架構中將會多一個 state 儲存資料，及三個閥門用來設定閾值以控制資料的傳遞與否 [21]。如若僅使用 RNN，則容易發生遞迴所導致的權重指數級爆炸或梯

度消失問題，善用 LSTM 控制資料的傳遞可以很好的抑制這種情況發生 [22] [23]。

## 2-3-2 卷積神經網路(Convolutional Neural Network, CNN)

卷積神經網路(CNN)屬於一種人工神經網路，由最頂層的全連接層及一或數個卷積層所組成，同時亦包含池化層，如圖 2-2 [24]。卷積層靈感主要參照動物視覺皮層組織的神經連接方式，每個神經元僅對特定範圍內的刺激作出反應，與池化層的運作方式類似，但卷積層的神經元感知區域可相互重疊從而覆蓋整個視野，而池化層的神經元感知區域並未重疊，因此卷積神經網路在圖像及音訊處理方面較為出色 [25]。

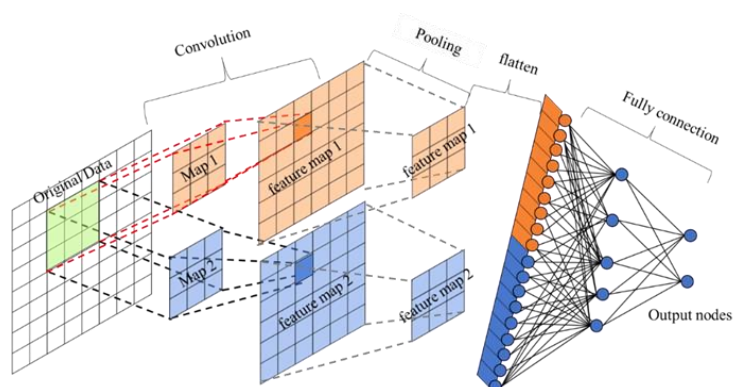


圖 2-2 CNN 架構圖

資料來源：維基百科 [24]

## 2-3-3 MTCNN

MTCNN 主要由三個 CNN 架構的子網路所組成，其中包含偵測人臉並生成邊界框的 P-Net、用來排除誤判的 R-Net 及最後對 R-Net 所輸出的結果再進一步回歸校正的 O-Net [26] [27]。因其獨特的架構，MTCNN 對於計算資源的需求較高，但也因此具有快速且高效的優點，同時亦可辨識各種大小種類不一的人臉 [28]。

## 2-4 人臉辨識(Face Recognition)

人臉辨識又稱臉部辨識，是透過分析比較人臉特徵進行身分鑑別的技術，屬於生物特徵辨識技術之一 [29]。此技術包括人臉的圖像採集、人臉定位、臉部辨識、身分確認等，由於生物特徵具有獨特性，因此適合用來辨識身分，其主要優勢在於可靈活應用於各種場景，雖說有容易受個體及環境因素影響等缺點，但整體仍是利大於弊，因此深受青睞 [30]。

### 2-4-1 OpenCV

OpenCV 於 1999 年由 Intel 公司開發，它是基於 BSD 許可發行的開源跨平臺的電腦視覺庫，它為影像處理與辨識、三維重建、物體追蹤、機器學習和線性代數提供各式各樣的演算法 [31]。透過 OpenCV 進行圖像辨識與應用，可以做到影像取得、圖片預處理與特徵提取等 [32]。

### 2-4-2 FaceNet

FaceNet 提出於 Google 在 2015 年間發表的一篇論文 [33]。它可以直接學習從人臉圖像到歐幾裡得空間的映射，在此空間中距離可直接對應於人臉相似性的度量 [34]。其過程大致為：輸入兩張圖片並透過深度學習網路提取特徵，分別進行 L2 標準化以取得各自的 128 維特徵向量，將兩向量映射至歐基裡德空間，隨後計算兩者間的距離，即可判定兩張人臉照片的相似度，如圖 2-3 [35] [36]。



圖 2-3 FaceNet 運作流程圖

資料來源：生物辨識小團體 [36]



## 2-5 物件追蹤(Multiple Object Tracking)

物件追蹤主要包含物件偵測(Object detection)及追蹤器(Tracker)等兩大部分。物件偵測的主要功用是在影像中抓取所要的目標物件生成偵測值並賦予邊界框；而追蹤器則是用於判斷前後幀所抓取到的物件是否屬於同一個，若為同一物件便賦予相同 ID，反之則賦予全新 ID [37]。

### 2-5-1 DeepSort

DeepSort 於物件追蹤中常用於改善追蹤效果，尤其是有遮蔽物導致目標丟失的情況。其最大的特點就是額外提供外觀資訊，並借用 Re-ID 領域中的模型來提取特徵，以此減少 ID 切換的次數 [38]。其主要流程為檢測器取得邊界框生成檢測值，轉由卡爾曼濾波進行預測，再利用匈牙利演算法將預測後獲取的路徑與當前幀中的檢測值進行匹配並更新卡爾曼濾波，周而復始，便可達到改善追蹤效果的目的 [39] [40]。

### 2-5-2 Re-ID

Re-ID 被稱為行人重辨識，也稱作行人再辨識，主要用於對特定行人進行跨視域匹配和檢索，而所謂跨視域匹配即是當目標出現在不同畫面中時，仍將其視為同一目標。在單一鏡頭下所追蹤的人物經常會因為離開拍攝範圍或被障礙物遮蔽而丟失該人物。因此在這種狀況下，如何才能確保此人再度出現時仍能判斷出他是曾經出現過的人物，便是 Re-ID 的主要目的。而該項技術亦可利用於多鏡頭的狀況下，在目標人物跨鏡頭出現時仍將其判斷為同一人 [41]。

## 2-6 系統開發工具與技術

本章節將深入研究人工智慧領域中常見的程式語言，並探討相關的

函式庫。在這個日益發展的領域中，選擇合適的程式語言以及瞭解相應的函式庫是至關重要的，有助於有效地建立、訓練及運用人工智慧模型。

## 2-6-1 相關程式語言

本節將著重介紹三種相關的程式語言：Python、Java、C++。這三種語言各自擁有獨特的特性和優勢，在人工智慧的開發與應用上扮演著重要而不可替代的角色。

### (一) Python

Python 一種廣泛使用的直譯式高階程式語言，具有龐大的標準函式庫及眾多協力廠商函式庫，且本身可跨近乎所有的作業系統上執行，能大幅提升編寫時的效率；語法近似於英語的基本語法，且社群上具有眾多用戶可供提問，因此對於初學者十分容易上手。此外 Python 還強調程式碼的可讀性及簡潔的語法，可便於開發人員的閱讀及理解 [42] [43]。

Python 在人工智慧領域中，因其可提供大量且高效的函式庫，且能夠協助使用者快速且有效地學會結合並運用相關的資料結構及其他人工智慧演算法。此外，利用 Python 開發時，其可擴充性與可攜性可使開發者更加輕鬆的透過其他程式語言編寫擴充函式庫以應付各自需求，因此深受大家推崇 [44]。

### (二) Java

在人工智慧領域中，Java 是一個極為合適的程式語言選擇，它具有高度安全性、可攜性及可伸縮性，這些特性均可滿足人工智慧應用的多樣需求，且其易懂的語法和豐富的函式庫使得開發者能夠輕鬆應對複雜的系統開發。而其中它的可攜性和內建的垃圾回收機制更是大幅提升開發者在大型 AI 專案中採用 Java 的吸引力，也因此使其成為人工智慧領域中十分熱門的開發工具 [45]。

### (三) C++

在人工智慧領域中，時間敏感性至關重要，而 C++相較於其他常見的程式語言，其執行效率更高，因此能夠協助開發人員優化程式執行時間，使得演算法能更高效地運行，特別是在遊戲開發中，C++經常被用於實現遊戲中的人工智慧，以確保更快的執行速度與即時的響應時間。同時 C++亦能更直接且精確地控制及分配硬體資源，能極大程度地滿足 AI 領域中對於資源分配的需求 [46]。

## 2-6-2 相關函式庫

本節將著重介紹四種相關的函式庫：Numpy、PyQt5、MySQL 及 Threading。這些函式庫在不同領域展現出強大的功能，提供豐富的工具和資源，協助解決各種複雜的問題。從資料處理、圖形介面開發到資料庫管理和多執行緒處理，這四個函式庫的結合提供全方位的支援，讓人工智慧專案更加高效且靈活。

### (一) Numpy

Numpy 是 Python 的一個擴充函式庫，可支援高階大規模的多維陣列與矩陣運算，此外也針對陣列運算提供大量的數學函數 [47]。Numpy 能用於定義任意的資料型態，便於進行多種資料庫的整合。此外亦可根據自身需求結合運用各種 Python 擴充函式庫 [48]。

### (二) PyQt5

PyQt5 是 Python 語言的 GUI 編程解決方案，可用來代替 Python 內建的 Tkinter [49]。使用者可利用 Qt Designer 設計並匯出 .ui 檔，因 Qt Designer 是使用拖拉操作來設計 GUI，所以在操作方面十分人性化，且後續可透過命令提示字元輸入 pyuic5 指令可將其轉換成 .py 檔以便引用 [50]。

### (三) MySQL

MySQL 是一種開源關聯資料庫管理系統，會將資料儲存在資料表中。使用者可使用結構化查詢語言(SQL)來定義、操作、控管及查詢資料。MySQL 速度快、可靠、可擴充且易於使用，具有 ACID（完整性、一致性、獨立性和耐用性）特性，無論錯誤、失敗或其他潛在危害，都能確保資料有效性 [51]。而本系統是透過 MySQL Connector 連接使用 MySQL 資料庫，為 MySQL 官方提供的驅動器，提供連接 Python 和 MySQL 資料庫所需的所有功能，同時兼具進階功能，如事務管理、批量操作和連接池等 [52]。

### (四) Threading

Threading 是 Python 標準函式庫裡的模組，可用於撰寫多執行緒的平行計算程式，利用多顆 CPU 核心加速運算，發揮硬體的運算能力 [53]。Threading 也可以讓原本同步的執行改為非同步，兩者差異在於前者只有單一執行緒下，必須按照執行順序完成每個任務，而後者可同時分成不同執行緒，各自執行任務，減少等待前一個任務完成的時間，所以可大幅減少工作時間 [54]。

## 2-6-3 相關軟體與工具

本節將深入探討與領域相關的軟體與工具，著重介紹五種相關的軟體與工具：Visual Studio Code、Notepad++、Colab、Anaconda、Windows 11 及 Qt Designer。這些工具的選擇和使用直接影響到最終的成功與效能。透過深入瞭解這些軟體和工具，能更好地應對複雜的挑戰。

### (一) Visual Studio Code

Visual Studio Code 是由 Microsoft 開發的一個免費、輕量且開源的程式碼編輯器。它支援多種編程語言，並提供豐富的擴充功能，使開發

者能夠個性化地擴展和配置環境，以滿足不同的開發需求，其可以在 Windows、macOS 和 Linux 等多個平臺上運行，並內建了 Git 版本控制工具，使開發者能夠輕鬆進行版本控制操作 [55]。

## (二) Notepad++

Notepad++是進階版的 Windows 記事本文字編輯器，主要用於一般檔及程式碼的撰寫，同時可為程式語言的語法上色以增加可讀性，亦能開啟大型的檔案不會延遲。它是由 GNU 通用公共授權條款(GNU General Public License, GPL)許可的自由開發軟體，以 Win32 API、STL 及 C++ 程式語言撰寫而成 [56]。

## (三) Colab (Colaboratory)

Colab 是一個能用網頁瀏覽器編寫程式且能執行的平臺，亦能整合 Google 服務儲存於雲端，便於使用者共用。其架構以 Jupyter Notebook 的開發環境為基礎，並預先安裝常用的 python 套件，提供免費的 GPU，可高效運算機器學習演算法，適用於模型訓練 [57][58]。

## (四) Anaconda

Anaconda 是一個開源的 Python 和 R 語言的發行版本。Anaconda 擁有自己的套件管理系統和應用程式，主要集中在數據科學、機器學習、人工智慧等領域，其中包含 Conda 和虛擬環境管理，其亦可在多個作業系統上運行，由於其方便性和功能豐富性，成為了現今工程師的首選工具之一 [59]。

## (五) 作業系統(OS)：Windows 11

Windows 11 是由微軟公司的團隊設計和開發。Windows 11 是微軟推出的作業系統，在設計和功能上進行多項改進，包括重新設計的使用者介面、更優雅的視窗管理、新的應用程式商店等。在此版本的 Windows 中已無 IE 瀏覽器，由 Microsoft Edge 取而代之，其也導入 Teams 成為內

建通訊軟體，因此 Skype 將不會在預載於系統當中 [60]。

## (六) Qt Designer

Qt 是應用程式開發的框架，而 Qt Designer 是其中的一個模組，可運用 Qt 的工具設計和建置圖形使用者介面(GUI) [61]。其可視化設計界面，可將元件拖入畫面編輯區進行介面設計，如圖 2-4 [62]。其也有自定義屬性和信號槽，可使用戶與介面之間交互更加靈活 [63]。

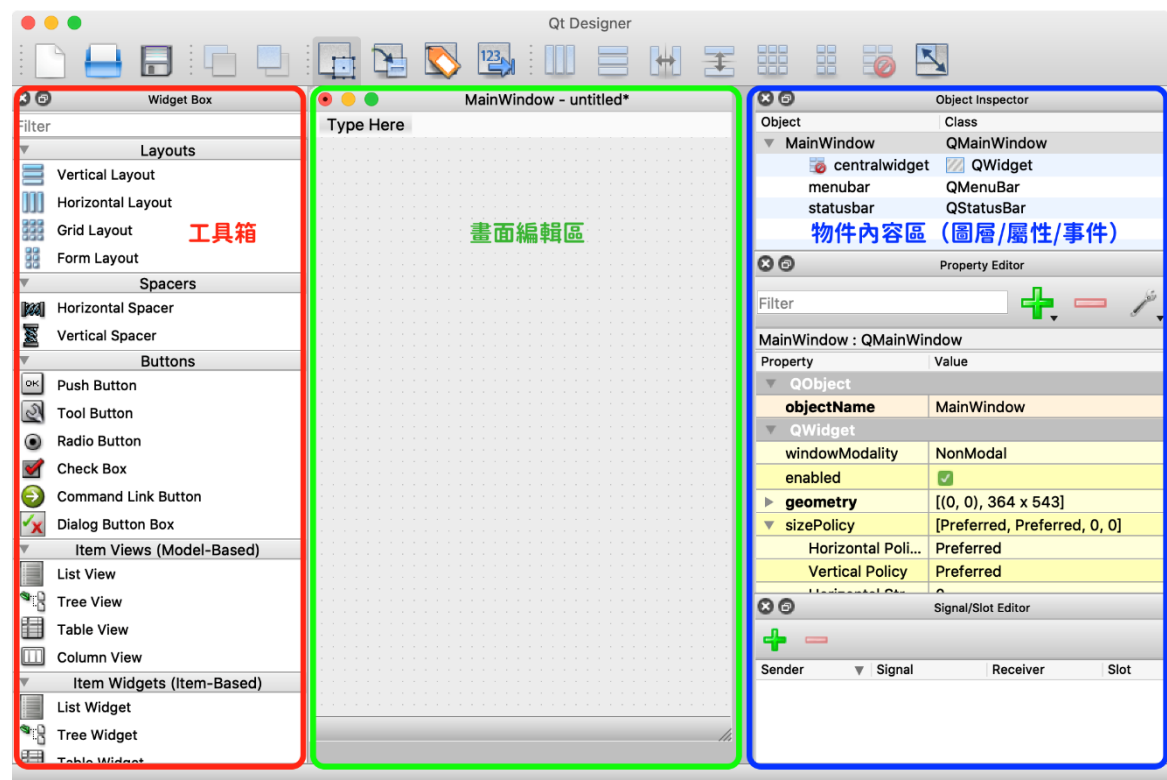


圖 2-4 Mac OS 版 Qt Designer 介面

資料來源：Johnny Zhung [62]

## 第三章 系統分析與設計

本章將針對使用者需求進行系統分析，並在瞭解使用者需求後設計本系統，因此同時也會對系統設計進行詳細的說明。系統分析分為需求調查和功能分析兩節。在需求調查部分，將詳細說明使用者對系統的需求。而在功能分析部分，將闡述各項系統功能的具體用途，並透過活動圖形式呈現功能與使用者之間的關聯。系統設計將探討本系統的介面與資料庫設計。介面一節將詳細說明其畫面與功能，以提供全面的瞭解。資料庫方面將分為當日進館人員資料庫、歷史資料庫、特殊名單資料庫、通知對象資料庫及人數紀錄資料庫等五個主要部分深入解說。

### 3-1 需求調查

此章節將分析使用者對於本系統的需求。為了確保圖書館館員能夠順暢地使用本系統，並且為入館者提供更舒適的體驗，本專題將以圖書館館員的角度來設計系統。

由於校內師生眾多，人流控管變得較為複雜，需要追蹤進出館的人員紀錄及特殊名單上的人是否已進入圖書館；另外，閉館時也需確保沒有人員逗留在館內。這些工作皆需要館員手動處理，因此希望透過本系統來協助館員更有效地管理圖書館。

### 3-2 系統功能分析

本章節主要說明本系統的功能，再以活動圖加以細述。主要分為四個功能，其中包括即時人流監控、偵測特殊人士、閉館人流檢查及人流統計數據管理。本研究各項功能及其說明如下：

(一)即時人流監控：

本系統可供使用者查看館內當前人數及今日進館總人數，以便於瞭解當前館內狀況。管理者開啟系統後，系統將持續進行人物偵測，當入館者進入鏡頭範圍，便將畫面中該人員臉部截圖，並為其賦予編號存入當日進館人員資料庫，與此同時變動館內人數並即時更新系統數據，如圖 3-1。

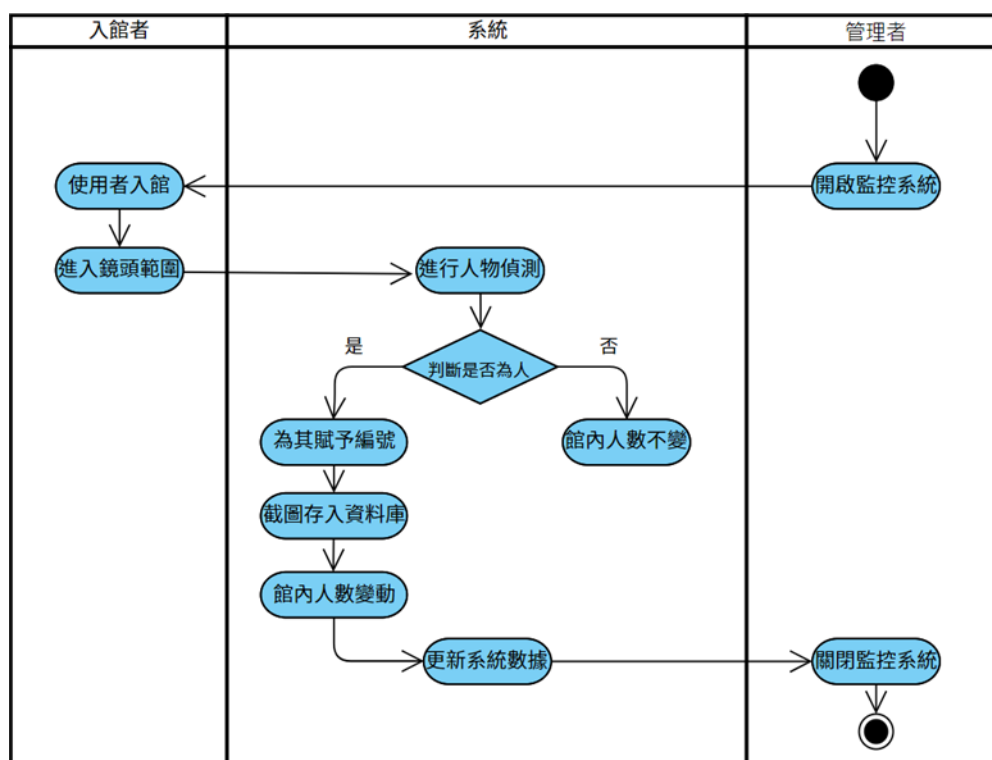


圖 3-1 即時人流監控活動圖

## (二) 偵測特殊人士：

本系統透過管理人員或櫃檯人員新增特殊名單，並設定當特殊人士入館時須發送通知的對象，便於館方及時對名單人員採取措施。系統開啟後，一旦偵測到入館者，便對其進行臉部截圖，並與特殊名單資料庫中每筆資料進行比對。經判定結果為特殊人士時，將發送警示郵件給通知對象資料庫中的每位人員，隨後變動館內人數並更新系統數據，如圖 3-2。



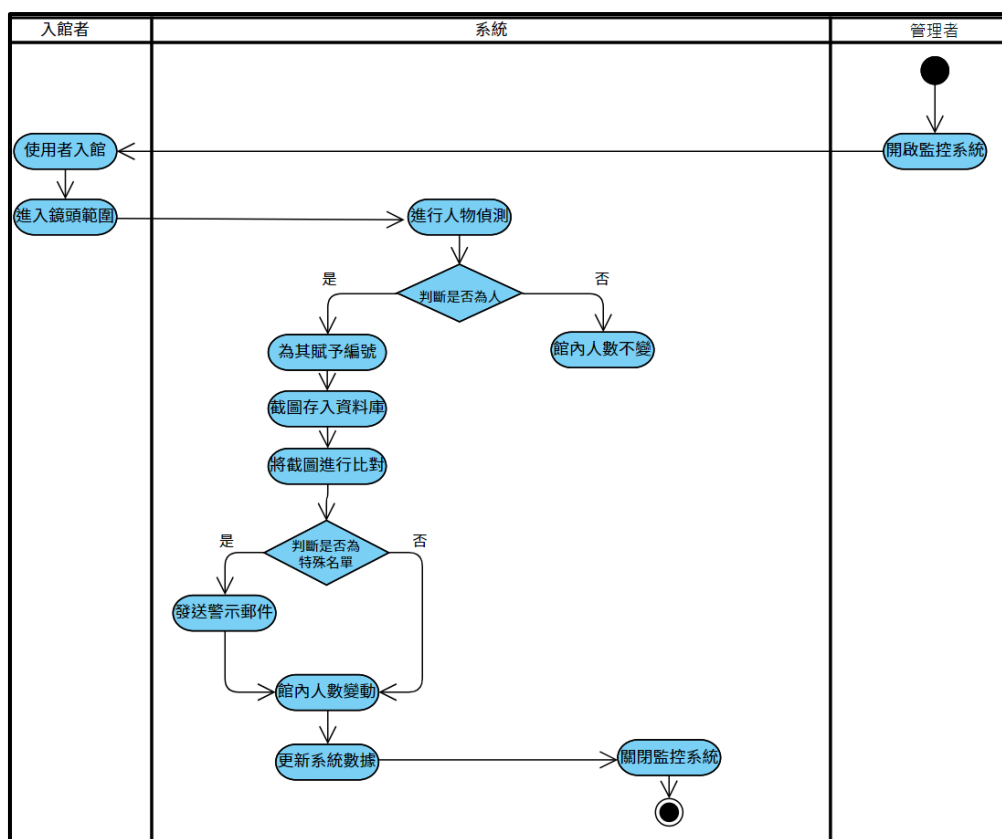


圖 3-2 偵測特殊人士活動圖

### (三) 閉館人流檢查：

本系統可於閉館時協助館員快速確認館內狀況，如發現有逗留人員便回報提醒館員。當圖書館將閉館時，開啟閉館檢查，系統將會確認當日進館人員資料庫，若庫中數據已清空則回報閉館檢查完畢並關閉系統，反之則顯示館內剩餘人員並於畫面中顯示其進館的臉部截圖及進館時間，以便館員前去請離，如圖 3-3。

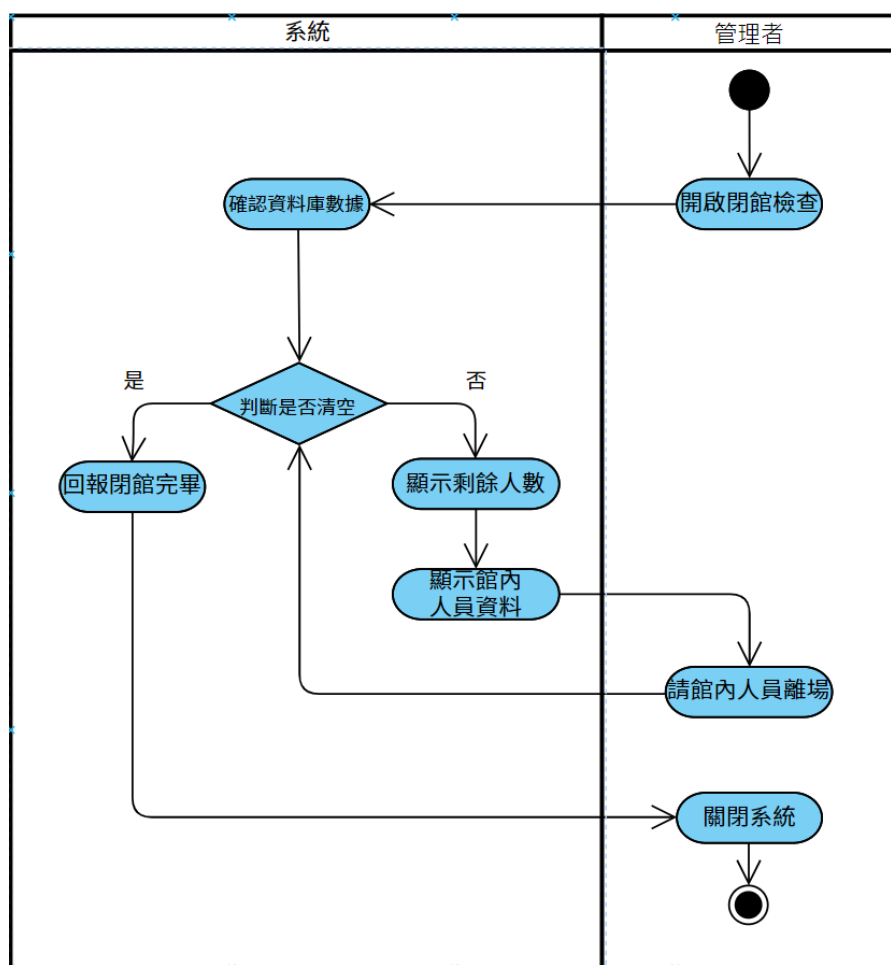


圖 3-3 閉館人流檢查活動圖

#### (四) 人流統計數據管理：

本系統存取過往的人流數據，便於管理人員調閱及匯出使用。管理人員於統計頁面進行查詢，選定目標時段、圖表格式及時間單位，系統將輸出詳細數據並繪製成圖表以便管理人員查閱；匯出功能將數據轉換為指定檔案類型並匯出當前圖表以供管理人員取用，如圖 3-4。

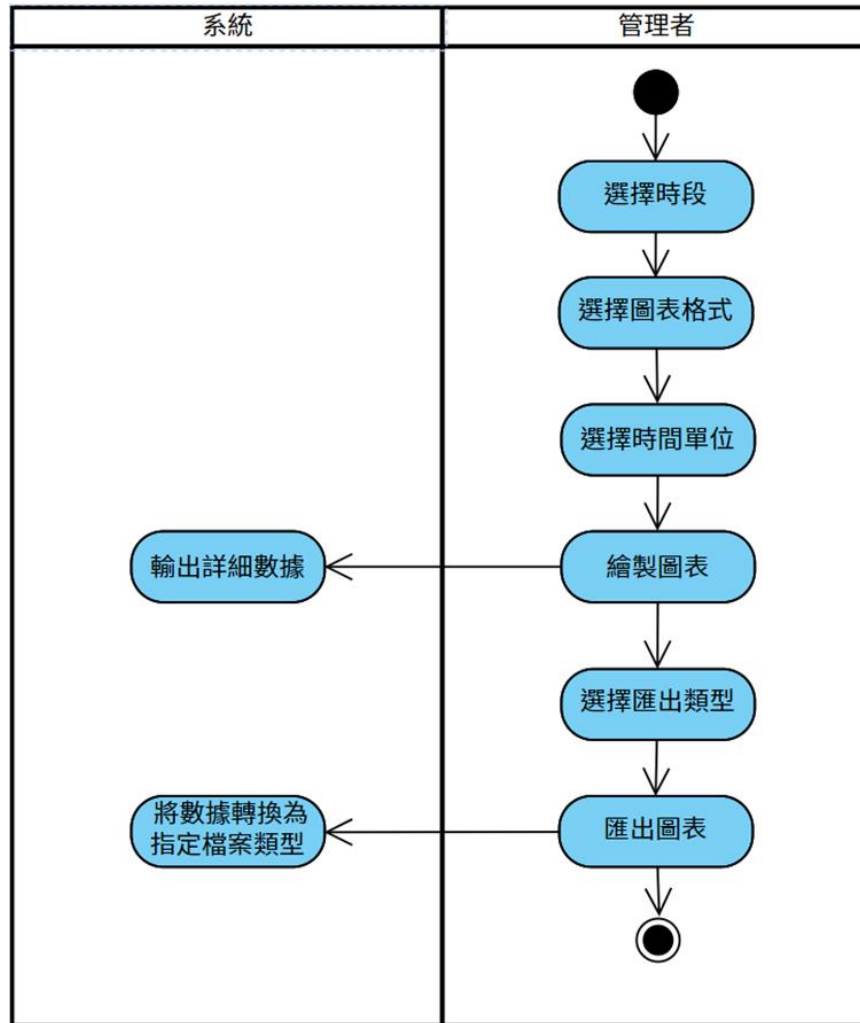


圖 3-4 人流統計數據管理活動圖

### 3-3 介面設計

此節將透過圖片呈現本專題介面及功能配置。其中包括主畫面、查詢畫面、人流統計畫面及管理員設定畫面。

#### 3-3-1 主畫面設計

主畫面（如圖 3-5）為本專題系統的核心頁面，蘊含著即時人流監控、進出館人數統計、偵測特殊人士以及閉館人流檢查等關鍵功能。接下來，將逐一詳細說明其各項功能。



圖 3-4 主畫面

##### (一) 即時人流監控：

該功能於主畫面中即時顯示進出館的鏡頭畫面，並偵測畫面中進出館人員，經統計偵測結果後於下方顯示自系統開啟的進出館人數及當前館內人數。

##### (二) 偵測特殊人士：

當偵測到特殊人士入館時，於即時監控畫面中特別標記並顯示該目標的姓名，隨後立即發送警示郵件給通知對象，如圖 3-6。



圖 3-5 偵測特殊人士

### (三) 閉館人流檢查：

按下 close 按鍵，若館內尚有人員未離館，即顯示該人員的資料，如圖 3-7，待館內逗留人員離開後即可手動刪除資料並關閉系統，如圖 3-8。



圖 3-7 閉館人流檢查



圖 3-6 閉館檢查結束

### 3-3-2 查詢畫面設計

查詢畫面（如圖 3-9）可提供以時間段、個人資料及圖片查詢特定目標於當日進館人員資料庫或歷史資料庫中的紀錄，並將結果輸出在畫面右側的表格中。

此外於畫面右下方可勾選僅顯示出特殊名單，則右側查詢結果即會隱藏所有非名單中的人員，亦可直接於查詢結果選取非特殊名單的人員，提供其姓名及學號/員編並新增至特殊名單。

勾選	ID	Name	number	Image Path	In Time
<input type="checkbox"/>	745				2023-09-10 ...
<input type="checkbox"/>	746	daisy	09360085	/special_list/...	2023-09-10 ...
<input type="checkbox"/>	747	daisy	09360085	/special_list/...	2023-09-10 ...
<input type="checkbox"/>	748	kira	09360990	/special_list/...	2023-09-10 ...
<input type="checkbox"/>	749	jerry	09360874	/special_list/...	2023-09-10 ...
<input type="checkbox"/>	750				2023-09-13 ...
<input type="checkbox"/>	751				2023-09-13 ...
<input type="checkbox"/>	752				2023-09-13 ...
<input type="checkbox"/>	753				2023-09-13 ...
<input type="checkbox"/>	754	kira	09360990	/special_list/...	2023-09-13 ...
<input type="checkbox"/>	755	jerry	09360874	/special_list/...	2023-09-13 ...
<input type="checkbox"/>	756				2023-09-13 ...
<input type="checkbox"/>	757				2023-09-13 ...
<input type="checkbox"/>	758	jerry	09360874	/special_list/...	2023-09-13 ...
<input type="checkbox"/>	759				2023-09-15 ...
<input type="checkbox"/>	760				2023-09-15 ...
<input type="checkbox"/>	761	jerry	09360874	/special_list/...	2023-09-15 ...
<input type="checkbox"/>	762				2023-09-15 ...
<input type="checkbox"/>	763				2023-09-15 ...

圖 3-7 查詢畫面

### 3-3-3 人流統計畫面設計

人流統計畫面（如圖 3-10）可以輸出特定時段內的人流統計數據。此畫面中可以設定並繪製該圖表。

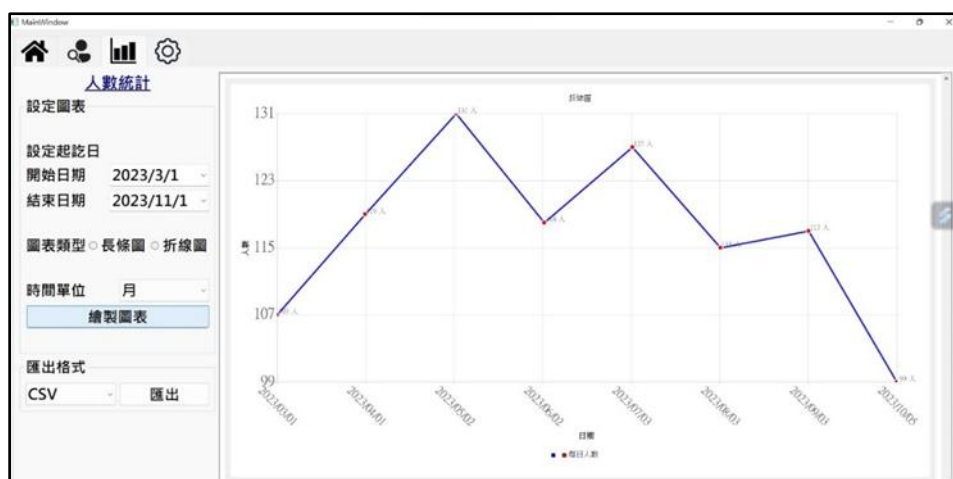


圖 3-8 人流統計畫面

#### (一) 設定圖表：

該功能可設定起訖日、圖表類型及時間單位。此外系統亦設有限制使用者繪製圖表時可選擇的時間單位以提高圖表的可讀性。一週內時間單位為”日”，一個月內時間單位為”日、週”，一年內時間單位為”週、月”，一年以上時間單位為”月、年”。

#### (二) 匯出格式：

該功能可選擇匯出圖表的檔案格式，其中可匯出的檔案格式分別為 CSV、Excel、PDF 及 JPG 等選項。

### 3-3-4 管理員設定畫面設計

管理員設定（如圖 3-11）可以設定及調整警示郵件的傳送對象及特殊名單。

圖 3-9 管理員設定畫面

#### (一) 管理通知對象：

本功能用於管理特殊人士進館時應傳送警示郵件的對象。新增時需填寫通知對象姓名及其電子郵件。而後可於右側表格中單獨勾選以進行修改及刪除。

#### (二) 管理特殊人士：

本功能用於管理特殊名單，新增時特殊人士的姓名、學號/員編及正臉照為必填項目，因此畫面中有加 “\*” 提醒，而其他項目為選填。未來可於右側表格中單獨勾選以進行修改及刪除，亦或補齊選填項目。

### 3-4 資料庫邏輯設計

為求方便管理者掌握館內人流狀況，本系統共設計五個資料庫：當日進館人員資料庫、歷史資料庫、特殊名單資料庫、通知對象資料庫及人數紀錄資料庫，以便管理者查閱及設定。透過圖 3-12 可清楚瞭解各個資料庫彼此之間的關係。



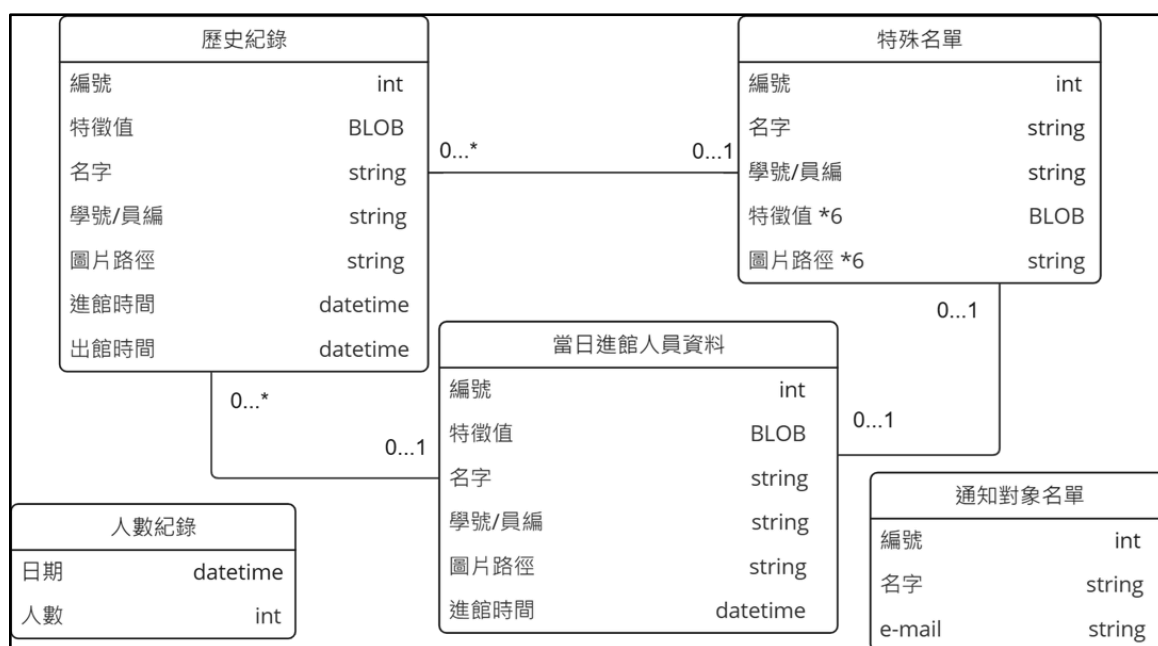


圖 3-10 資料庫邏輯設計

(一) 當日進館人員資料庫：

當日進館人員資料庫用於存取當日進館人員，內容包含人員編號、姓名、學號或員編、進館臉部截圖（圖片路徑）及其特徵值和進館時間。人員於進館後，資料將會暫存於該資料庫，並於離館後將其移至歷史資料庫。

(二) 歷史資料庫：

歷史資料庫用於存取過往進館人員，其內容包含姓名、學號或員編、進館臉部截圖（圖片路徑）及其特徵值和最近一次的進出館時間。該資料庫於管理者有需求時，可提供協助以快速查詢到目標人員並取得其外觀或進出館時間等細項資訊。

(三) 特殊名單資料庫：

特殊名單資料庫用於存取特殊人士，其內容包含編號、姓名、學號/員編、口罩配戴與否的各角度照片、及其特徵值。當鏡頭畫面中出現人員時，可供系統用於比對是否為特殊人士。

(四) 通知對象資料庫：

通知對象資料庫用於存取系統寄送警示郵件進行通知的對象，其內容包含姓名及該人員的電子郵件。當特殊人士進館時，系統於此同時會寄送警示郵件提醒該資料庫中的人員，以便管理者即時採取相關措施。

(五) 人數紀錄資料庫：

人數紀錄資料庫用於存取系統每天所記錄下的進館總人數，以便館員後續於統計畫面繪製圖表及匯出取用。

## 第四章 系統實作與測試

本章將詳細介紹系統的開發環境與架構及實測結果分析。在開發環境方面，將深入探討所使用的硬體和軟體工具。系統架構部分將逐步解析系統運作流程，並於最後進行實測及分析結果，為後續的系統開發提供有力的參考。

### 4-1 開發環境

本章節將會介紹本專題中所使用到的硬體設備、軟體及工具。其中包含處理器、作業系統、記憶體、顯示卡、鏡頭、程式碼編輯器、程式語言及系統運行環境。

#### 4-1-1 硬體設備

硬體設備是系統運作不可或缺的基石，其性能直接關係到系統的運算效能與處理能力。本研究採用性能較高的硬體配置，以確保系統能夠順利執行各項任務。

(一) 處理器(CPU)：11th Gen Intel(R) Core(TM) i7-1165G7

(二) 記憶體(RAM)：16GB DDR4 3200Mhz

(三) 顯示卡(GPU)：NVIDIA GeForce MX330

(四) 鏡頭(Webcam)：羅技(Logitech) C270i HD

#### 4-1-2 使用軟體與工具

在系統的軟體與工具方面的選擇是確保開發環境的穩定性和效能。程式碼編輯器選擇 Visual Studio Code，其多元的擴充套件能夠提升開發效率，如 Live Share 可即時共用協作。同時，透過選擇 Anaconda 作為系

統運行環境，確保豐富的 Python 開發工具和函式庫的支援，對於系統的開發和測試提供穩定的基礎，其亦可透過 Anaconda 虛擬環境安裝 CUDA Toolkit 切換使用 GPU。在程式語言採用 Python，其廣泛的應用範疇，能確保系統開發的靈活性與可擴展性。最後，在介面設計上使用 Qt Designer，其可視化界面便於轉換成 XML 格式，再透過 PyQt5 轉換成.py 檔，以便後續操作。這些軟體與工具的結合為本系統的開發提供一個有效的工作環境，確保開發環境的穩定性和效能。這樣的整合不僅解決各種需求，同時也在系統構建階段確保軟硬體間的順暢溝通。

## 4-2 系統架構

本章節將詳細介紹整個系統的架構流程，如圖 4-1。一旦啟動系統，即時監控影像將顯示在主畫面上。透過 MTCNN 進行人臉偵測，當有入館者被捕捉到時，系統將截取其臉部截圖，接著利用 FaceNet 進行特徵值提取，再透過 DeepSort 進行人臉追蹤。當人員離開館時，再次利用 FaceNet 進行人臉比對，以確認離館者在當日進館人員資料庫中的身分，隨後將其移至歷史資料庫。

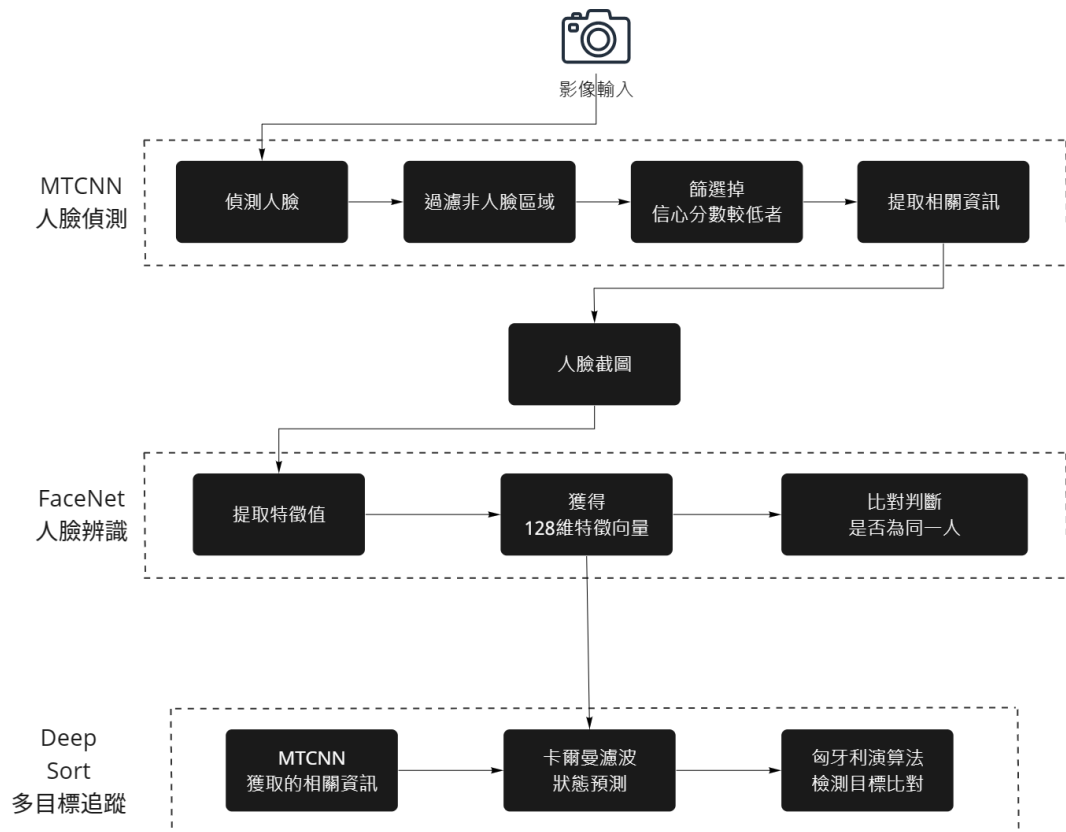


圖 4-1 系統架構圖

### 4-2-1 MTCNN 人臉偵測

MTCNN 於讀取影像後並檢測人臉的位置，提供多項人臉相關資訊。其中包括檢測框的位置和尺寸（左上角座標  $x, y$ 、寬度  $w$ 、高度  $h$ ）、人臉可信度分數（分數接近 0 表示為非人臉，接近 1 表示為人臉），及關鍵點（如左眼、右眼、鼻子、左嘴角和右嘴角的位置），如圖 4-2。

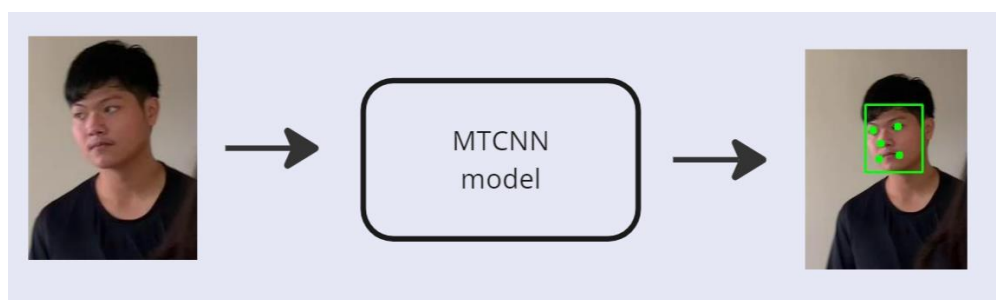


圖 4-2 MTCNN 人臉偵測

## 4-2-2 DeepSort 物件追蹤

DeepSort 利用 MTCNN 提供的資訊和 FaceNet 提供的特徵向量，可以生成一系列該目標的檢測結果。系統會提取前一幀的資訊，預測檢測結果在當前幀中的位置，並為每個檢測結果賦予一個追蹤辨識編號。接著，將當前幀的所有檢測結果與前一幀的追蹤辨識編號進行匹配，以判斷前一幀的某一目標是否與當前幀的某目標相對應。如此的追蹤機制有助於特定目標的穩定追蹤，使系統能夠在動態環境中有效地追蹤人臉，提高整體追蹤的準確度和穩定性，如圖 4-3。

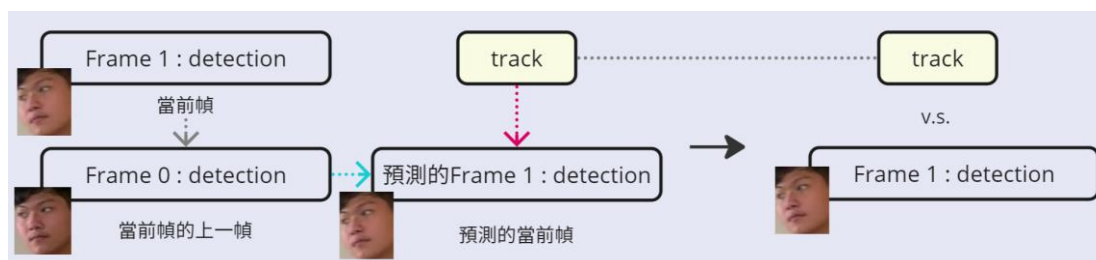


圖 4-3 DeepSort 物件追蹤

## 4-2-3 FaceNet 人臉辨識

FaceNet 透過 MTCNN 獲取的人臉位置進行特徵提取，獲得 128 維的特徵向量。該特徵向量與 MTCNN 提供的人臉相關資訊結合，為 DeepSort 提供準確的人臉追蹤工具。同時，該特徵向量亦可用於將檢測到的人臉與特殊名單中的人員進行比對，實現對特定身份的辨識與驗證，

如圖 4-4。

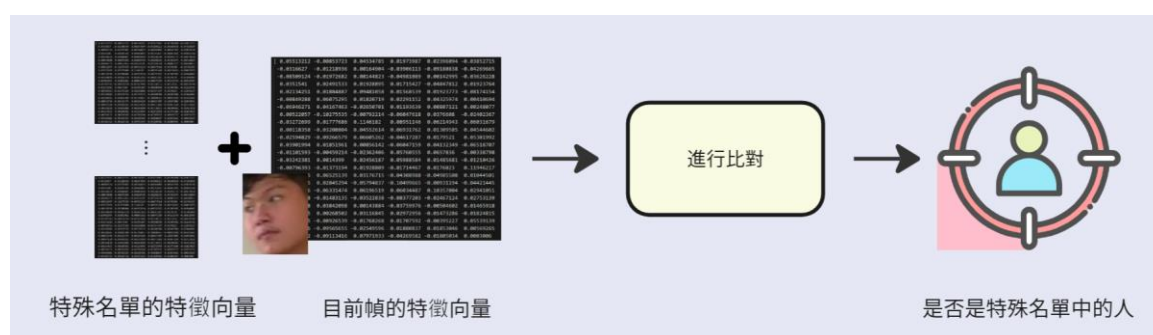


圖 4-4 FaceNet 人臉辨識

## 4-3 系統測試

系統測試為本系統經測試後，於各種情境下可達到的成效。本章節將探討準確率、穩定性及可靠性測試，透過測試將更瞭解本系統的問題，以尋求未來系統優化方向。

### (一) 準確率測試

本段將呈現本系統經測試後，於各種人流情況下可達到的準確率。表 4-1 為本系統準確率測試的結果，進出館的平均準確率均達 80% 以上，偵測特殊人士的平均準確率為 68%，閉館人流檢查的平均準確率為 73%。綜上所述，準確率測試結果為尚可，且本系統未來仍有待優化的空間。

表 4-1 系統測試準確率

人流情境	進館人流監控	出館人流監控	特殊人士偵測	閉館檢查
正常時段	81%	83%	71%	70%
尖峰時段	75%	70%	56%	67%
離峰時段	87%	89%	77%	82%
總平均	80%	81%	68%	73%

## (二) 穩定性測試

穩定性測試為評估系統在長時間運行下的穩定性。本系統測試最長的運行時間為 1 小時，期間系統未有出現任何異常卡頓的問題，雖說持續時長稍嫌過短，但未來可透過新增定時清除背景運行垃圾及減少非必要的程式片段等方式以提高穩定性，因此經評估後本系統的穩定性測試結果為尚可。

## (三) 可靠性測試

可靠性測試為評估系統在面對故障或異常情境時的性能。當鏡頭畫面過於複雜時，系統可能會陷入卡頓的狀態，透過重啟系統即可恢復。如若過於嚴重亦可能導致系統崩潰，這種情況下只能透過使用者手動重新執行程式。因此經評估後本系統的可靠性測試結果為尚可，且在應對複雜畫面的表現仍有改善空間。

## (四) 測試分析

經實測後，發現會造成本系統準確率下降的問題有三（如表 4-2）：首先，人物相互遮擋的錯誤占比約為 5.5%，較容易發生在尖峰時段；其次，部分人員進館時，會配戴口罩及帽子或給予鏡頭非正臉，該原因的錯誤占比約為 89%，還有其餘約 5.5%則瑣碎分佈於其他原因；最後，晚上的光線影響減弱，該問題占錯誤總數的 30%，白天時易造成曝光，導致白天的錯誤率相較於晚上提高 2.3 倍。以上三個問題都容易造成臉部截圖不完整，導致無法準確辨識身分，進而使本系統的準確率下降。綜合上述三個測試，未來圖書館可以考慮規劃進出口動線以便系統偵測人臉，或調整館內佈置以降低室外光線影響，進而提高整體準確率，因此本系統的總體測試結果為尚可。



表 4-2 進出館各原因錯誤率占比

影響原因	配戴口罩	遮擋鏡頭	其他
進館	52%	3.2%	1.5%
出館	37%	2.3%	4%

## 第五章 結論

### 5-1 研究成果

現如今各科技已然十分成熟，許多過去僅能依靠人力達成的工作，皆可透過結合各項技術來完成，其中更以人工智慧最為突出。以管理者的角度來說，善加利用人工智慧，不僅能節省人力資源且更有效率外，更可大幅降低人為疏失的發生率。

本專題所開發的系統以銘傳大學桃園校區圖書館為實驗場域，利用深度學習技術，實現對圖書館內人流的監控與分析。本系統使用的技術主要包括人臉偵測(MTCNN)、物件追蹤(DeepSort)及人臉辨識(FaceNet)。透過 MTCNN 進行人臉偵測，當鏡頭捕捉到入館者時，將其臉部截圖，交由 FaceNet 進行特徵提取，同時比對是否為特殊人士及發送警示，並藉由 DeepSort 進行人臉追蹤。

本系統透過特殊名單的設置，管理者能有效掌握特殊人士的進出館動向；即時人流監控提供即時的館內人數統計，使館員得以輕易獲取人流數據，以便隨時調整及管理人力資源；而閉館人流檢查則可於閉館時，協助館員快速取得館內逗留人員的相關資料，以提高館內的整體效率。

然而本系統在面對背景環境較為複雜的情況時，辨識率較容易受其影響，導致未能達到預期成效，其中又以光線明暗程度影響最為明顯。此外也會因進出人員的個體因素而有所影響，如穿搭或行走路線，而出現鏡頭遭遮擋或無法準確辨識目標的狀況，導致系統誤判降低準確率。

### 5-2 未來展望

本系統提供圖書館員及組長實測後給予良好的評價，然而本系統仍

有進步空間，期望本系統能更貼近他們的使用需求。未來除了持續提升各項技術的辨識準確度外，也考慮結合學校圖書館門禁系統取得學號，並與系統所截人臉照進行身分確認。人流統計數據方面也將詳細統計各時段，以便進行人員調配及開閉館時間調整。此外，也建議圖書館可以於落地窗張貼玻璃紙，以降低光線問題，也可規劃出入館的動線指引，以便於系統偵測，降低錯誤率的發生，以期推廣至各種大型場館之人流管理與監控。

## 參考文獻

- [1] SAP，〈什麼是機器學習〉，網址：  
<https://www.sap.com/taiwan/products/artificial-intelligence/what-is-machine-learning.html>，上網日期：2023 年 9 月 23 日。
- [2] Tn 科技島，〈「機器學習」究竟是什麼？定義、四大類型一次看〉，網址：  
<https://www.technice.com.tw/issues/ai/2517/>，上網日期：2023 年 9 月 23 日。
- [3] AWS，〈什麼是機器學習〉，網址：  
<https://aws.amazon.com/tw/what-is/machine-learning/>，上網日期：2023 年 9 月 23 日。
- [4] OOSGA 的研究團隊，〈機器學習(ML)定義為何？演算法有哪些？〉網址：  
<https://zh.oosga.com/docs/machine-learning/>，上網日期：2024 年 1 月 22 日。
- [5] Design Thinking 團隊，〈三大類機器學習：監督式、強化式、非監督式〉，  
網址：  
<https://ai4dt.wordpress.com/2018/05/25/%E4%B8%89%E5%A4%A7%E9%A1%9E%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%EF%BC%9A%E7%9B%A3%E7%9D%A3%E5%BC%8F%E3%80%81%E5%BC%B7%E5%8C%96%E5%BC%8F%E3%80%81%E9%9D%9E%E7%9B%A3%E7%9D%A3%E5%BC%8F/>，上網日期：2024 年 1 月 22 日。
- [6] 中文百科全書，〈半監督學習〉，網址：  
<https://www.newton.com.tw/wiki/%E5%8D%8A%E7%9B%A3%E7%9D%A3%E5%AD%B8%E7%BF%92>，上網日期：2024 年 1 月 22 日。
- [7] 維基百科，〈深度學習〉，網址：  
<https://zh.wikipedia.org/zh-tw/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0>，上網日期：2023 年 9 月 26 日。
- [8] CrazyFire，〈Day-01 深度學習是什麼〉，iT 邦幫忙，網址：  
<https://ithelp.ithome.com.tw/articles/10264238?sc=iThelpR>，上網日期：2023 年 9 月 26 日。
- [9] 維基百科，〈TensorFlow〉，網址：  
<https://zh.wikipedia.org/zh-tw/TensorFlow>，上網日期：2023 年 9 月 30 日。
- [10] TensorFlow，〈TensorFlow 是用於機器學習的端對端開放原始碼平臺〉，網址：  
<https://www.tensorflow.org/overview?hl=zh-tw>，上網日期：2023 年 9 月 30 日。
- [11] 維基百科，〈PyTorch〉，網址：  
<https://zh.wikipedia.org/zh-tw/PyTorch>，上網日期：2023 年 9 月 30 日。

- [12] 三阿毛，〈PyTorch 簡易介紹〉，網址：  
<https://medium.com/@auwit0205/pytorch-%E7%B0%A1%E6%98%93%E4%BB%8B%E7%B4%B9-45e25d3269b2>，上網日期：2023 年 10 月 5 日。
- [13] pyliaorachel，〈Day7/PyTorch/深度學習框架之亂〉，iT 邦幫忙，網址：  
<https://ithelp.ithome.com.tw/articles/10241809>，上網日期：2023 年 10 月 5 日。
- [14] xunan003，〈Darknet 框架簡介〉，網址：  
<https://blog.csdn.net/xunan003/article/details/79932888>，上網日期：2024 年 1 月 29 日。
- [15] 路同學，〈介紹一個相對小眾的深度學習框架 Darknet，其 YOLO 神經網路演算法對目標檢測效果顯著〉，網址：  
<https://cloud.tencent.com/developer/news/76803>，上網日期：2024 年 1 月 29 日。
- [16] AWS，〈什麼是神經網路？〉，網址：<https://aws.amazon.com/tw/what-is/neural-network/>，上網日期：2023 年 10 月 9 日。
- [17] K. Hsieh，〈神經網路是什麼：應用和案例說明〉，網址：  
<https://nordvpn.com/zh-tw/blog/shenjing-wang-lu/>，上網日期：2023 年 10 月 9 日。
- [18] 維基百科，〈人工神經網路〉，網址：<https://zh.wikipedia.org/zh-tw/%E4%BA%BA%E5%B7%A5%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>，上網日期：2023 年 10 月 9 日。
- [19] 李馨伊，〈卷積神經網絡 Convolutional Neural Network (CNN)〉，網址：  
<https://medium.com/ching-i/%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E7%B5%A1-convolutional-neural-network-cnn-d7246d24ff3e>，上網日期：2024 年 1 月 29 日。
- [20] 打不死的小強，〈循環神經網路 – Recurrent Neural Network | RNN〉，網址：  
<https://easyai.tech/ai-definition/rnn/>，上網日期：2023 年 10 月 11 日。
- [21] sarahwei0804，〈【Day 28】長短期記憶網路 Long short-term memory〉，網址：  
<https://ithelp.ithome.com.tw/articles/10308248?sc=rss.iron>，上網日期：2024 年 1 月 29 日。
- [22] 維基百科，〈循環神經網路〉，網址：<https://zh.wikipedia.org/zh-tw/%E5%BE%AA%E7%8E%AF%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>，上網日期：2023 年 10 月 11 日。
- [23] 王柏鈞，〈RNN（遞迴神經網路）全面認識〉，Medium，網址：

<https://medium.com/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E7%9F%A5%E8%AD%98%E6%AD%B7%E7%A8%8B/rnn-%E9%81%9E%E8%BF%B4%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF-%E5%85%A8%E9%9D%A2%E8%AA%8D%E8%AD%98-27e4dd150bdd>，上網日期：2023 年 10 月 11 日。

- [24] T. Huang，〈卷積神經網路(Convolutional neural network, CNN) — CNN 運算流程〉，網址：<https://chih-sheng-huang821.medium.com/%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF-convolutional-neural-network-cnn-cnn%E9%81%8B%E7%AE%97%E6%B5%81%E7%A8%8B-ecaec240a631>，上網日期：2023 年 10 月 9 日。
- [25] 維基百科，〈卷積神經網路〉，網址：<https://zh.wikipedia.org/zh-tw/%E5%8D%B7%E7%A7%AF%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C>，上網日期：2023 年 10 月 9 日。
- [26] AI 之路，〈MTCNN 演算法及代碼筆記〉，CSDN 博客，網址：<https://blog.csdn.net/u014380165/article/details/78906898>，上網日期：2023 年 10 月 10 日。
- [27] 邱學文，〈人臉檢測—MTCNN 從頭到尾的詳解〉，知乎，網址：<https://zhuanlan.zhihu.com/p/58825924>，上網日期：2023 年 10 月 10 日。
- [28] 武維，〈MTCNN 人臉檢測 — ONet 網路訓練〉，知乎，網址：<https://zhuanlan.zhihu.com/p/32163660>，上網日期：2023 年 10 月 10 日。
- [29] 維基百科，〈臉部辨識系統〉，網址：<https://zh.wikipedia.org/zh-tw/%E8%87%89%E9%83%A8%E8%BE%A8%E8%AD%98%E7%B3%BB%E7%B5%B1>，上網日期：2023 年 10 月 14 日。
- [30] GoFace，〈人臉辨識是什麼？不可不知的人臉辨識系統 6 大優缺點〉，網址：<https://www.goface.me/zh-TW/blog/zh-TW/b32.html>，上網日期：2023 年 10 月 14 日。
- [31] 維基百科，〈OpenCV〉，網址：<https://zh.wikipedia.org/wiki/OpenCV>，上網日期：2023 年 10 月 15 日。
- [32] 王春仁，〈[OpenCV]基礎教學筆記：影像讀取、前處理(with python)-001〉，Medium，網址：<https://medium.com/jimmy-wang/opencv-%E5%9F%BA%E7%A4%8E%E6%95%99%E5%AD%B8%E7%AD%86%E8%A8%98-with-python-d780f571a57a>，上網日期：2023 年 10 月 15 日。
- [33] 山姆大叔，〈[Day 22] Facial Recognition: Google FaceNet〉，iT 邦幫忙，網址：<https://ithelp.ithome.com.tw/articles/10275219>，上網日期：2023 年 10 月 17 日。

- [34] 我不愛機學習，〈人臉特徵學習 FaceNet 簡介〉，知乎，網址：  
<https://zhuanlan.zhihu.com/p/493521556>，上網日期：2023 年 10 月 17 日。
- [35] Bubbliiiiing，〈神經網絡學習小記錄 32——facenet 詳解及其 keras 實現〉，  
台部落，網址：<https://www.twblogs.net/a/5e5449d0bd9eee2117c4e84a>，上網日  
期：2023 年 10 月 17 日。
- [36] 生物辨識小團體，〈FaceNet 人臉辨識 (CVPR2015)〉，網址：  
[https://hackmd.io/@esun-face/ryCTdssU\\_](https://hackmd.io/@esun-face/ryCTdssU_)，上網日期：2023 年 10 月 18 日。
- [37] 履歷軌，〈Multiple Object Tracking (MOT)〉，網址：  
<https://hackmd.io/@computerVision/S18nD20Vq>，上網日期：2023 年 10 月 18  
日。
- [38] pprp，〈Deep SORT 多目標跟蹤代碼解析（上）〉，知乎，網址：  
<https://zhuanlan.zhihu.com/p/133678626>，上網日期：2023 年 10 月 18 日。
- [39] 可樂，〈目標跟蹤初探(DeepSORT)〉，知乎，網址：  
<https://zhuanlan.zhihu.com/p/90835266>，上網日期：2023 年 10 月 19 日。
- [40] P. Wu，〈Deep Sort 論文筆記：SIMPLE ONLINE AND REALTIME  
TRACKING WITH A DEEP ASSOCIATION METRIC〉，Medium，網址：  
[https://pedin024.medium.com/deep-  
sort%E8%AB%96%E6%96%87%E7%AD%86%E8%A8%98-simple-online-and-  
realtime-tracking-with-a-deep-association-metric-c6a9f01392ec](https://pedin024.medium.com/deep-sort%E8%AB%96%E6%96%87%E7%AD%86%E8%A8%98-simple-online-and-realtime-tracking-with-a-deep-association-metric-c6a9f01392ec)，上網日期：  
2023 年 10 月 19 日。
- [41] 上杉翔二，〈Person Re-Identification (ReID 行人重識別)〉，CSDN 博客，  
網址：[https://blog.csdn.net/qq\\_39388410/article/details/108173767](https://blog.csdn.net/qq_39388410/article/details/108173767)，上網日期：  
2023 年 10 月 19 日。
- [42] 維基百科，〈Python〉，網址：<https://zh.wikipedia.org/wiki/Python>，上網日  
期：2023 年 10 月 21 日。
- [43] AWS，〈什麼是 Python？〉，網址：[https://aws.amazon.com/tw/what-  
is/python/](https://aws.amazon.com/tw/what-is/python/)，上網日期：2023 年 10 月 21 日。
- [44] Shawn 來科普，〈AI 時代開發語言的新標準：Python，為何人工智慧不能沒  
有它？〉，網址：[https://www.potatomeia.co/post/a554e7d4-4208-4527-8466-  
34f3207859bf](https://www.potatomeia.co/post/a554e7d4-4208-4527-8466-34f3207859bf)，上網日期：2024 年 1 月 29 日。
- [45] atyun，〈最適合人工智慧的程式語言：JAVA 人工智慧程式編程〉，網址：  
<https://read01.com/zh-tw/3GjjALN.html>，上網日期：2024 年 1 月 29 日。
- [46] 海棠如醉，〈人工智慧領域對 C++的需求〉，網址：  
<https://blog.csdn.net/feelinglikeyou/article/details/133813369>，上網日期：2024  
年 1 月 29 日。

- [47] 維基百科，〈Numpy〉，網址：<https://zh.wikipedia.org/wiki/NumPy>，上網日期：2023 年 10 月 21 日。
- [48] S. Lo，〈Hello NumPy！〉，網址：<https://medium.com/python4u/hello-numpy-b5ebe67a1ada>，上網日期：2023 年 10 月 22 日。
- [49] 維基百科，〈PyQt〉，網址：<https://zh.wikipedia.org/wiki/PyQt>，上網日期：2023 年 10 月 22 日。
- [50] Zhung\_，〈PyQt 入門，用 Python 寫第一支 GUI〉，網址：<https://zhung.com.tw/article/pyqt%E5%85%A5%E9%96%80%E7%94%A8python%E5%AF%AB%E7%AC%AC%E4%B8%80%E6%94%AFgui/>，上網日期：2023 年 10 月 22 日。
- [51] Google Cloud，〈什麼是 MySQL？〉，網址：<https://cloud.google.com/mysql?hl=zh-tw>，上網日期：2023 年 10 月 22 日。
- [52] 稀土掘金，〈mysql connector 是什麼〉，網址：<https://juejin.cn/s/mysql%E2%0connector%E6%98%AF%E4%BB%80%E4%B9%88>，上網日期：2023 年 10 月 22 日。
- [53] G. T. Wang，〈Python 多執行緒 threading 模組平行化程式設計教學〉，網址：<https://blog.gtwang.org/programming/python-threading-multithreaded-programming-tutorial/>，上網日期：2023 年 10 月 23 日。
- [54] STEAM 教育學習網，〈threading 多執行緒處理〉，網址：<https://steam.oxxostudio.tw/category/python/library/threading.html>，上網日期：2023 年 10 月 23 日。
- [55] 維基百科，〈Visual Studio Code〉，網址：[https://zh.wikipedia.org/zh-tw/Visual\\_Studio\\_Code](https://zh.wikipedia.org/zh-tw/Visual_Studio_Code)，上網日期：2023 年 11 月 22 日。
- [56] WEDO 網頁設計公司，〈台灣之光 Notepad++好免程式編輯軟，支援 50 多種程式設計語言〉，網址：<https://rwd.wedo.com.tw/index.php/knowledge-articles/free-software/513-notepad-plus-plus-text-editor>，上網日期：2024 年 1 月 29 日。
- [57] J. Wu，〈Google Colaboratory 介紹〉，medium，網址：<https://johnny-notes.medium.com/google-colaboratory-%E4%BB%8B%E7%B4%B9-dbd09ebf077e>，上網日期：2024 年 1 月 29 日。
- [58] 好豪，〈為什麼我超愛用 Google Colab？Python 菜鳥與老手都適合的利器〉，網址：<https://haosquare.com/why-google-colab-python/>，上網日期：2024 年 1 月 29 日。
- [59] 維基百科，〈Anaconda〉，網址：[https://zh.wikipedia.org/zh-tw/Anaconda\\_\(Python%E5%8F%91%E8%A1%8C%E7%89%88\)](https://zh.wikipedia.org/zh-tw/Anaconda_(Python%E5%8F%91%E8%A1%8C%E7%89%88))，上網日期：



2023 年 11 月 22 日。

- [60] MikaBrea，〈Windows 11 完整體驗、實裝搶先看：Windows11 新介面、用法完整介紹〉，I T 客邦，網址：<https://www.techbang.com/posts/89029-big-change-windows-11-get-a-first-look-real-world-experience>，上網日期：2024 年 1 月 20 日。
- [61] The Qt Company Ltd. "Qt Designer Manual." Available: <https://doc.qt.io/qt-6/qtdesigner-manual.html>. (accessed Jan 30, 2024).
- [62] J. Zhung，〈用 Qt Designer 來設計 PyQt GUI 應用程式界面〉，網址：<https://zhung.com.tw/article/%E7%94%A8qt-designer%E4%BE%86%E8%A8%AD%E8%A8%88pyqt-gui%E6%87%89%E7%94%A8%E7%A8%8B%E5%BC%8F%E7%95%8C%E9%9D%A2/>，上網日期：2024 年 1 月 30 日。
- [63] The Qt Company Ltd . "Qt Designer's Signals and Slots Editing Mode." Available: <https://doc.qt.io/qt-6/designer-connection-mode.html>. (accessed Jan 30, 2024).

# 附錄

## 附錄一 系統安裝說明

為了確保使用者在操作本系統時能清楚地瞭解安裝過程，本附錄將提供詳盡的安裝步驟說明。透過此附錄，使用者將能夠更得心應手地完成系統的安裝，建立一個穩定且可靠的運行環境。以下是安裝步驟的具體指南：

### (一) Github 下載系統程式

在 Windows 系統中使用命令提示字元克隆系統程式

```
git clone https://github.com/daisy00000/Pedestrian-flow-monitoring-Detections-in-Library-by-Using-Deep-Learning-Technology.git
```

### (二) 設置 OpenCV 影片處理庫環境變數

在 Windows 系統中使用命令提示字元執行以下命令以設置環境變數，這能幫助快速找到鏡頭影像配置（優先順序）

```
setx OPENCV_VIDEOIO_PRIORITY_MSMF 0
```

### (三) 創建和啟用 Conda 虛擬環境

1. 需先安裝 anaconda

2. 創建虛擬環境

在程式所在的資料夾中開啟命令提示字元並輸入以下指令

```
conda create --name myenv python=3.9
```

3. 啟用虛擬環境

在命令提示字元並輸入以下指令

```
activate myenv
```

(四) 安裝 Nvidia CUDA Toolkit (如有使用 GPU 再執行)

在命令提示字元使用 Conda 安裝 CUDA Toolkit，確保使用版本 11.7 (電腦本身須內建 Nvidia)：

```
conda install cuda -c nvidia==11.7
```

(五) 安裝系統所需套件

在命令提示字元使用以下指令安裝系統所需的 Python 套件：

```
pip install -r requirements.txt
```

(六) 更改資料庫參數

在 MySQL 中建立新的資料庫並更改 main.py 中的指令(換成你所建置的資料庫密碼及名稱)

```
conn = mysql.connector.connect(  
    host='127.0.0.1',  
    user='root',  
    password='your_password',  
    database='your_database'  
)
```

(七) 執行程式

```
python main.py
```

(八) 備註：測試帳號密碼(發送通知郵件)

以下為預設通知 gmail 帳號，可自行更改 main.py 中的設定

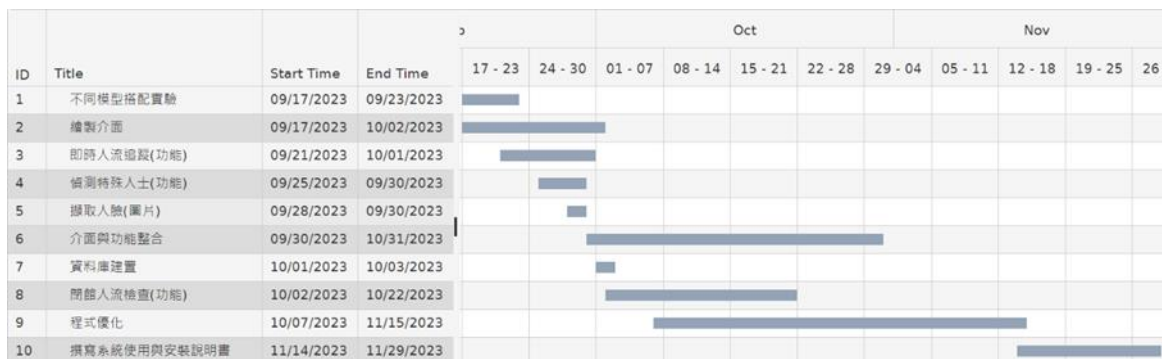
帳號：mcutest000@gmail.com

應用程式密碼：aijd aolm jyfu ahmo

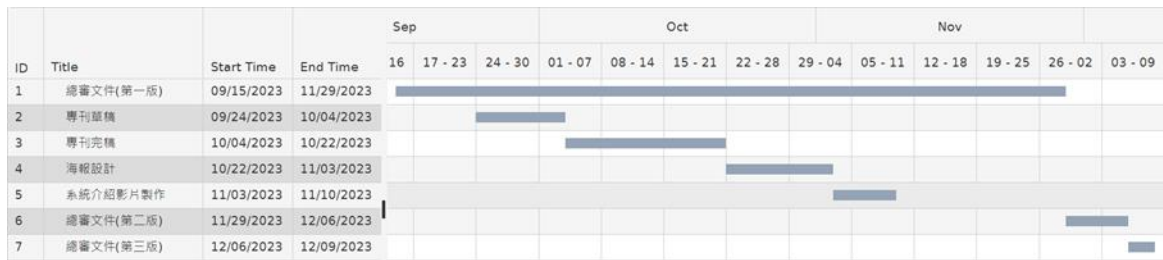
## 附錄二 專案時程與分工

此附錄為本專案的工作時程與分配。將以甘特圖呈現本專案的工作時程。工作分配分為兩組進行，分別為程式組（陳誼瑄、蔣亞軒、楊宸榕）及文件組（蔡彙苡、李明輝），程式組即為負責撰寫本系統程式，文件組即為撰寫相關檔，彼此相互合作使專案能順利進行。

程式組工作時程主要分為三個階段進行，前期以撰寫系統功能為主，中期加入介面設計與功能整合，後期進行優化及測試系統。文件組工作時程根據系所規劃分為兩個部分撰寫相關檔，分別為成果展及專題畢業審查。透過甘特圖（附圖 1、2）可明確知曉專案的預期進度及方便追蹤當前進度。



附圖 1 程式組工作時程甘特圖



附圖 2 文件組工作時程甘特圖