



June 7, 2024

國立成功大學資訊工程學系

114級
大學部畢業專題
專題本



9:00-16:00
資訊新館一樓



目錄

組別	組員姓名	專題題目 (中文+英文)	指導教授	頁數	發表領域
1-1	李星辰 蕭鈺蓁	基於多視角排球影片之球員動作辨識 Player Action Recognition in Multi-View Volleyball Videos	朱威達	1	1. 數位生活科技與軟體技術
1-2	呂奕宸	基於足球與球員追蹤之持球與傳球成功率之分析 Football possession and passing analysis based on player and football tracking	朱威達	3	
1-3	楊懷甯	結合逆向與正向動力學及生物力學機制模擬人體手臂運動 Applying inverse and forward dynamics with biomechanical consideration for human arm movement simulation	朱威達	5	
1-4	李維軒 歐陽主敬	文字提示驅動之桌面機器人流程自動化 Desktop RPA by Prompting	李信杰	7	
1-5	朱偉哲	強化 pythontutor 視覺化工具的 c/c++ 函式附加性及除錯 Strengthen c/c++ function additionality and debugging of pythontutor visualize tools	涂嘉恒	9	
1-6	蔡聿善	針對系統與網路管理的學生作業評測系統 Online judge system for system and network management	涂嘉恒	11	
1-7	吳昱輝	透過視覺化可互動式程式碼映射工具展示量子程式預處理最佳化 Illuminating Quantum Profile-Guided Optimization via Interactive Source-Level	涂嘉恒	13	
1-8	黃品叡 鄧仕煊	從想睡到享睡：睡眠腦波分析平台與應用 From Sleepiness to Enjoying Sleep: A Sleep EEG Analysis Platform and its Applications	梁勝富	15	
1-9	黃友枚 孫以塘 練智剛	基於 NILM 的用電分析與建議 Electricity consumption analysis and suggestions based on NILM	莊坤達	17	
1-10	陳鈞天 易子捷	TimeScaleDB 在系統資料處理時的效能優勢與應用 Application of TimeScaleDB in data processing	莊坤達	19	
1-11	楊采語	三維室內場景重建 3D Indoor Scene Reconstruction	連震杰	21	
1-12	龍奕全	應用透視投影和光流追蹤技術於擴增實境排球場 Augmented Reality-Based Volleyball Court using Perspective Projection and Optical Flow Tracking	連震杰	23	
1-13	潘駿諺 吳仲倫 臣韞儒 曾奕程	可生成摘要與輔助教學的 Chatbot 及高效學習平台 Advanced Learning Video Platform Integrating Automated Summary Generation and Tutoring Chatbot Support	郭耀煌	25	
1-14	周嘉豪	基於機率模型設計與實作演算法預測麻將手牌 Designing and implementing algorithms to predict Mahjong hand tiles based on probability models	陳奇業	27	
1-15	林志芸 熊貫仲	FIT IN：個人化體感健身程式 FIT IN : A Personal Motion Sensing Fitness App	陳奇業	29	
1-16	趙邦鈞	應用於 SAR 影像之船隻偵測演算法 Ship Detection Algorithm in SAR Imagery	陳培殷	31	
1-17	傅靖淳	研究所入學資料整合平台 Integration Platform Of Graduate School Admission Data	賀保羅	33	

目錄

組別	組員姓名	專題題目 (中文+英文)	指導教授	頁數	發表領域	
1-18	張庭瑋	以 Error-state Kalman Filter 整合傳感器提高車輛導航精度 Enhancing Vehicle Navigation Accuracy via Error-state Kalman Filter	蔡佩璇	35	1. 數位生活科技與軟體技術	
1-19	陳培恩	透過 YOLOv8 實作人物追蹤 - 以檢測學生活動中心刷卡尾隨為例 Implementing Person Tracking using YOLOv8 - A Case Study on Detecting Student Activity Center Unauthorized Entry	蔡佩璇	37		
1-20	楊竣鴻	實作限制執行環境的 Python 與繞過手法 Implementing Restricted Execution Environments in Python and Techniques for Circumvention	蔡佩璇	39		
1-21	何韋德 高柏祐 呂振呈 林業誠	童話創意師：圖片轉故事語音平台 FairyTale Innovator: Image to Story Audio Platform	鄭憲宗	41		
1-22	謝秉宸	作伙：基於語意分析協助老人揪團 GetTogether : An app helping the elders to hold activities with semantic analysis	盧文祥	44		
1-23	聶怡安	多語移工交流 APP Multilingual Chat App for foreign workers	盧文祥	46		
1-24	林紹恆	復古旅行 Vintage Adventure	盧文祥	48		
1-25	陳廷笙 王昱翔	應用於 MiinStore 平台之資料湖實作 MiinStore DataLake Implementation	謝孫源	50		
2-1	劉彥誠	探討 PIM 技術如何影響線性回歸的效能 On Exploring How PIM Technique Affects the Performance of Linear Regression	何建忠	52		2. 架構與嵌入式系統
2-2	陳佳豪 蕭名惟	在 SSD 上重疊讀取和寫入操作以提升效能:觀察和實作 Enhancing SSD Performance via Overlapping Read and Write Operations: Observation and Implementation	何建忠	54		
2-3	陳禹丞	iSAFE: 多權重連網式間歇性運算系統的更新轉送演算法 iSAFE: Enabling Eveness of Data Freshness in Multi-Weight Networked Intermittent System	涂嘉恆	56		
2-4	方騏為 陳廷睿 陳家禾	汽車開放式系統架構通訊模組接收端軟體的設計與實作 Design and Implementation of AUTOSAR COM Module Receiver Side Software	張大緯	58		
2-5	莊程傑	邊坡監測系統實驗與資料分析 Experience of Slope Monitor and Data Analysis	張大緯	60		
2-6	黃律瑛 黃芊	AES 加密電路設計 AES Encryption Circuit Design	陳培殷	62		
2-7	洪茂菘 黃偉峰	水下影像電路設計與實現 Design and Implementation of Underwater Imaging Circuit	陳培殷 陳朝鈞	64		
2-8	何寬羿 林清弘	QOI 圖像壓縮電路設計 Digital Circuit Design For QOI Image Compression Algorithm	蘇銓清 陳培殷	66		
2-9	蔡鎬駿 吳翰宇	真隨機亂數產生器結合高速影像加密電路 High-Speed Image Encryption Circuit With True Random Number Generator	陳培殷 陳朝鈞	68		

目錄

組別	組員姓名	專題題目 (中文+英文)	指導教授	頁數	發表領域
3-1	許桓瑞	用機器學習偵測 DDoS Using Machine Learning to detect DDoS	洪昌鈺	70	3. 計算、通訊與網路
3-2	陳柏佑 吳驊明	於 P4 實現基於 Count-Min Sketch 和 Hyperloglog 的 DDoS 攻擊偵測 DDoS Detection by using Hyperloglog and Countmin Sketch on P4 Switch	洪昌鈺	72	
3-3	黃育笙 邢益城	殭屍網路特徵與分析 Botnet Characteristics and Analysis	張燕光	74	
3-4	林宸哲 古嘉雋	網購超商快取？不不不~網路資料快取！ Implementing NetCache for Load Balancing Based on SDN with P4	張燕光	76	
3-5	張原愷 張辰凱	基於機器學習之低速緩慢 DDoS 攻擊偵測 A Slow DDoS Attack Detection Based On Machine Learning	張燕光	78	
3-6	陳子揚	建立於 MiinStore 物聯網平台之 Matter 閘道器實作 Implementation of the Miinstore Matter gateway	謝孫源	80	
3-7	李肇和 林建宇	基於 P4 的網路負載均衡實作及探討 An implementation and analysis of load balancing based on P4	楊中平	82	
4-1	呂宏家	基於圖神經網路之隱私保護表格資料生成 Privacy-persevering Tabular Data Synthesis with Graph Neural Networks	李政德	84	4. 資料與知識工程
4-2	李秉儒	預測式資料增強稀疏圖神經網路推薦系統 Predictive Data Augmentation for Sparse Graph Neural Networks Recommenders	李政德	86	
4-3	田容甄	基於圖神經網路之表格模型集成學習 Tabular Ensemble Learning with Graph Neural Networks	李政德	88	
4-4	唐文蔚	隨機特徵圖神經網路之穩健表格異常偵測 Robust Anomaly Detection with Random-Feature Graph Neural Networks	李政德	90	
4-5	楊文字 紀宣如 朱心慈	強化財經大語言模型在加密貨幣新聞上的情緒分析 Enhanced financial large language models(FinGPT) in sentiment analysis of cryptocurrency	高宏宇	92	
4-6	王柏凱	太陽能發電量異常偵測平臺 Solar Power Anamoly Detection Platform	莊坤達	94	
4-7	王偉同	資料管線之高可用性 High availability of data pipelines	蕭宏章	96	
5-1	何庭寬 王譔評 施伯昌 卓翰威	智慧自動化視力檢查 Intelligent Automated Vision Screening	王士豪	98	5. AI 與人機互動
5-2	曾駿馳	AI 虛擬實況主 AI Virtual Streamer	王宏錯	100	
5-3	林宸顛	針對桌球初學者正手揮拍姿勢分析與 3D 可視化 Forehand stroke posture analysis & 3D visualization for table tennis beginners	朱威達	102	

目錄

組別	組員姓名	專題題目 (中文+英文)	指導教授	頁數	發表領域
5-4	李宇翔	基於足球與球員位置進行足球事件偵測 Football event detection based on position of ball and players	朱威達	104	5. AI 與人機互動
5-5	蔡誠穎	水滴漫遊：跑酷解謎新體驗 Waterdrop Roaming : A New Parkour puzzle Experience	何建忠	106	
5-6	張庭瑄 李秋霖 吳沛儒 曾紹宇	基於對抗學習與模仿學習技術的 AI 行為與人工智慧擴散模型的圖像生成，樣條曲線的 3D 軌道生成，實時反饋的視覺特效設計 Intelligent 3D Game design based on diffusion, GAILearning, 3D spline curve generation and visual effects design.	李同益	108	
5-7	蘇晟翔	利用 BERT 之圍棋 AI 下子預測 BERT for Go prediction	高宏宇	110	
5-8	楊文字 陳柏宏	問答檢索強化生成之適應性大語言模型線上助教系統 Adaptive online LLM-based TA systems by QA-RAG and RAG Fusion	高宏宇	112	
5-9	許耕瑜	ChatGPT 詐騙模擬與評估對話系統 ChatGPT Fraud Imitator and Evaluator.	高宏宇	114	
5-10	侯詠章	利用深度學習技術協助棒球選手 Deep Learning Technology Assistance For Baseball Players	連震杰	116	
5-11	李彥勳 林星佑	看！電腦教你解出魔術方塊！ From Baffled to Brilliant: Mastering the Rubik's Cube with Computer Guidance	連震杰	119	
5-12	蘇致宇 張禕倫	應用深度學習及數位孿生於人體姿態與機械手臂之協作 Interaction between Human Body Pose and Robot Arm Using Deep Learning and Digital Twin	連震杰	121	
5-13	江信昇 邱繼寬	應用 YOLOv8 與 Deepsort 於魚眼相機下執行行人追蹤和計數 People Tracking and Counting via Fisheye Camera Using YOLOv8 and DeepSort	連震杰	123	
5-14	徐廷愷 梁家葳	以骨架與球體追蹤的深度學習進行羽球運動分析 Skeleton and Ball Trackings for Badminton Sports Analysis Using Deep Learning	連震杰	125	
5-15	楊柏方 梁祐晨	應用深度學習於機器手臂的機器人茶水師 Robot Beverage Sommelier Using Robot Arm With Deep Learning	連震杰	127	
5-16	陳頌元	影像辨識之互動性應用-蚊子殺手 Interactive application of image recognition- mosquito killer	陳奇業	129	
5-17	陳曄翰	AI 音樂生成與實時評分平台：基於 Transformer-WGAN 與神經網路訓練互動介面 AI Music Generation and Real-time Scoring Platform: Based on Transformer-WGAN with Neural Network Training Interactive Interface	陳朝鈞	131	
5-18	古孝正 賴虹橋 吳出右 陳念誠	智能後空翻教練：基於 LSTM 之人體動作識別 Intelligent Backflip Coach: LSTM-based Human Action Recognition	曾繁勳	133	
5-19	魏宇弘	使用 YOLOv9 影像辨識系統偵測火災 Training YOLOv9 model for detection of fire and smoke	蔡佩璇	135	

目錄

組別	組員姓名	專題題目 (中文+英文)	指導教授	頁數	發表領域
5-20	蘇致連	智慧餐點分析系統：利用 CV 與 SAM 進行食物分割與營養素分析 Intelligent meal analysis system : Utilizing CV and SAM for ingredients segmentation and nutrients analysis	蔣榮先	137	5. AI 與人機互動
5-21	黃霖均	去偏見化之臉部生成和臉部重建工具 Debiasing Facial Generation and Reconstruction Tool	蔣榮先	139	
5-22	林威	慢性腎臟病患者的 AI 智慧飲食推薦 AI Smart Diet Recommendation for Patients with Chronic Kidney Disease	蔣榮先	141	
5-23	黃亮晨	基於檢索增強生成的精準資訊擷取系統 Answer briefly and precisely : A Segment-based RAG Framework with Specialized Preprocessing Pipeline	蔣榮先	143	
5-24	彭昇哲	使用 GPT 技術強化小語言模型 Applying GPT techniques to enhance small language model	謝孫源	145	
5-25	黃鈺軒	骨骼疾病預測模型之網頁資料庫建置 Web and Database Design for Skeletal Disease Prediction Model	謝孫源	147	
5-26	蔡鎔澤	基於深度學習的 3D 成年鼠大腦分割 3D Adult Mouse Brain Image Segmentation	謝孫源 王建堯	150	
5-27	張奕翔	行人重識別在真實世界之應用 Evaluation of Person Reidentification (Re-ID) techniques with Real-World data	藍崑展	152	
5-28	李侗陞	使用 PPG 訊號估計血壓 Blood Pressure Estimation Using PPG Signal	藍崑展	154	
5-29	程品奕	Grapycal: 人性化的圖形化程式語言 Grapycal: A humanized graphical programming language	蘇文鈺	156	
5-30	劉尚典 陳穎睿	以強化式學習解決裝箱問題之研究 Research for solving bin packing problem with reinforce learning	蘇銓清	158	
6-1	郭沛蓉 陳帛愛 鄭紹祺 潘冠穎	心動分析-基於深度學習的超音波左心室射血分數評估 Left Ventricle Ejection Fraction Assessment in Echocardiograms Using Deep Learning	吳明龍	160	6. 生醫工程(智慧醫療)
6-2	陳聖泓 曾筠傑	行動穿戴式腦波量測與分析系統 A Mobile and Wearable Brain Signal Recording and Analysis System.	梁勝富	162	
6-3	郭昱辰	評估息肉分割模型 Evaluation of Polyp Segmentation Models	賀保羅	164	
6-4	蔡翔逸	腎臟超音波報告格式化專題 Renal Ultrasound Formatting Project	蔣榮先	166	
6-5	李務	吞嚥困難的早期檢測 Early detection of dysphagia	藍崑展	168	
6-6	龔暄仁	半監督醫學影像分割-雙向複製黏貼技術介紹與成大醫院現實數據應用 Semi-Supervised Medical Image Segmentation: Bidirectional Copy-Paste and its application in NCKU hospital data	藍崑展	170	

目錄

組別	組員姓名	專題題目 (中文+英文)	指導教授	頁數	發表領域
7-1	吳昱頡 劉士郡 鄭銻鎧	多站點多產品產線排程 - 以被動元件製造廠為例 Scheduling for multi-site and multi-product production - the case of passive component factory	王宏鏞	172	7. 製造工程(智慧製造)
7-2	陳育琮	應用強化學習和基因演算法於多機多產品排程問題-以 SMT 為例 Applying Reinforcement Learning and Genetic Algorithm to Multi-Machine and Multi-Product Scheduling Problem-A case study on SMT	王宏鏞	174	
7-3	何宇婕	強化學習於多站點排程系統之應用-以 TFTLCD 為例 Applying Q-learning to Multi-Site Scheduling System - A Case Study on TFTLCD	王宏鏞	176	
7-4	王翔聖 葉弘堯	針對彩色多特徵光滑物件之智慧自動化瑕疵檢測 Intelligent Automation Defect Detection for Smooth Object with Multiple Color Features.	陳響亮	178	
7-5	張暉俊	智慧工廠戰情室與線上預警系統實現 Implementation of Intelligent Factory Dashboard and Online Early Warning System	陳響亮	180	
7-6	楊凱茗	液體烤漆瑕疵智慧檢測系統 Intelligent inspection system of liquid coating defects	陳響亮	182	

基於多視角排球影片之球員動作辨識

Player Action Recognition in Multi-View Volleyball Videos

指導教授：朱威達 教授
專題成員：李星辰、蕭鈺綦
開發工具：Python 3.9.18，
YOLOv8，OpenCV
測試環境：Ubuntu 22.04.1 LTS

一、簡介：

利用球員動作辨識技術，系統可以統計球員在場上的攻防數據，從而深入了解其在比賽中的表現。期待透過數據分析，幫助教練和球隊管理層更好地制定戰術與訓練計畫，發現球隊潛在的技術與戰術優勢，為球隊取得勝利。

運作流程

1. 利用 YOLOv8對影片進行 **object detection**，找到球員位置與動作
2. 結合不同視角影片，判斷球員所屬隊伍在左邊或右邊
3. 分析資料，判斷攻擊與防守次數，並記錄下來
4. 產生統計圖表，透過 UI 顯示資料

觀察問題

1. 由於直接使用 video YOLOv8內建的人物偵測會使得球員、觀眾以及裁判均被偵測出來，因此我們決定自己訓練動作辨識模型，但這導致球員動作辨識準確度不足，進而影響賽事資訊分析。
2. 辨識完每個 **frame** 的球員動作後，如何從中找出各個動作的起始與結束，從而判斷該隊伍的殺球 / 攔網數。

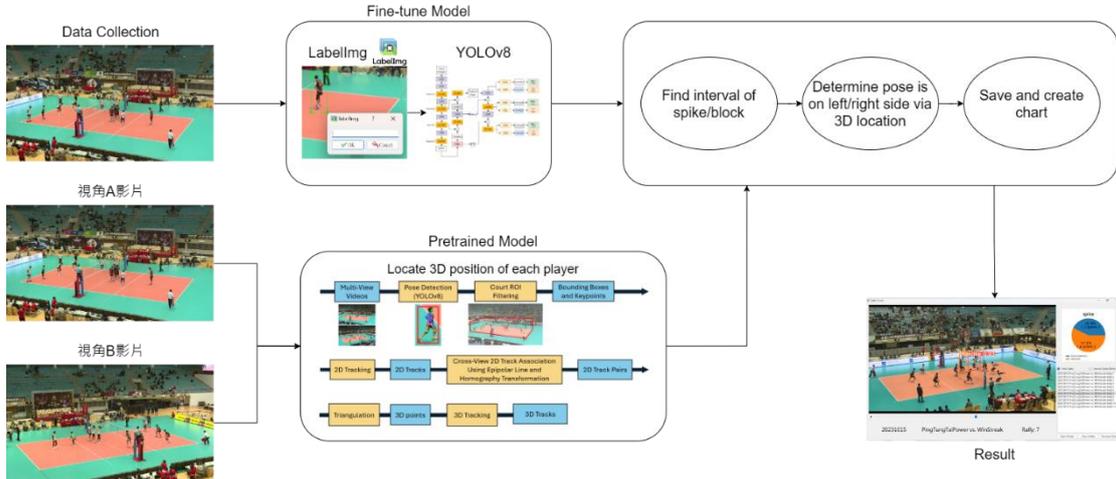
問題解決

1. 增加標記資料
起初總標記量大約200個 **frames** 左右，球員辨識率只有不到6成；隨後我們將每個 **label** (**spike**, **receive**, **block**, **nothing**) 分別標記500筆以上的資料，把四個相異視角和各隊伍的動作進行分類，達到辨識準確率9成的結果。
2. 以 **spike** 為例，比賽中會出現多個殺球假動作，因此在判斷上我們只將真正有做出攻擊者作為有效 **spike**；而在排球比賽中，能做出有效 **spike** 的人通常是最後一個還在進行攻擊的球員。其判斷做法如下：
 - a. 先找出第一個出現 **spike** 動作的 **frame**，標記為 **start**；隨後找到繼 **start** 後第一個完全沒有出現 **spike** 動作的 **frame**，標記為 **end**。
 - b. 計算兩者的時間長度判斷是否為誤判，並往前看上一個 **end** 與此次 **start**

的時間距離是否夠遠，否則判斷為同一個 spike。

- c. 找到 end 中有執行 spike 的球員（大多數情況只會有一位）的3D 座標，求得該球員所在隊伍。
- d. 將 spike 數與攻擊隊伍儲存並繪製統計圖表。

系統架構



二、測試結果：



基於足球與球員追蹤之傳球成功分析

Football Possession and Passing Analysis Based on Player and Football Tracking

指導教授：朱威達

專題成員：呂奕宸

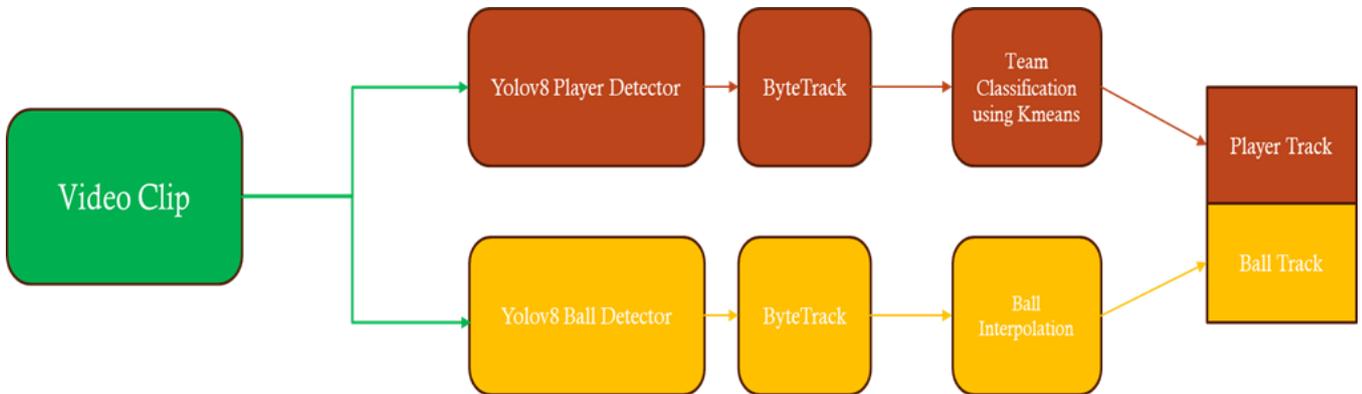
開發工具：Python3.11.7, YOLOv8, K-means, ByteTrack

測試環境：Windows 10

一、簡介：

足球裡球權與傳球成功率是最常研究的指標之一，它們與球隊的勝率息息相關。球權與高傳球成功率不一定能保證勝利，但它意味著控制比賽的節奏和速度，最重要的是，這會迫使對手做大量的跑動。世界頂尖的球隊可以利用此優勢在高強度的比賽中重新組織攻勢消耗對方的體力與心靈以造成更大的威脅。此專題我利用了 YOLOv8與 ByteTrack 追蹤球與球員的位置，以及 K-means 分辨兩支球隊，ball interpolation 減少足球 miss detection 的發生，並且利用它們的位置計算出兩支球隊的球權與傳球成功數。

以下為系統架構圖：



二、測試結果：

下圖為 player and ball detection + tracking 的結果:

1. 球員頭上紅色倒三角代表球員正在持球
2. 球員底下橢圓形顏色是依據 K-means 演算法分辨球隊的結果
3. 此時白隊持球，Team 1 的持球率會增加



4. 傳球成功，由上圖比較可知 Team 1 passes 從27增加到 28。



結合逆向與正向動力學及生物力學機制模擬人體手臂運動

Applying inverse and forward dynamics with biomechanical consideration for human arm movement simulation

指導教授：朱威達、鄭匡佑

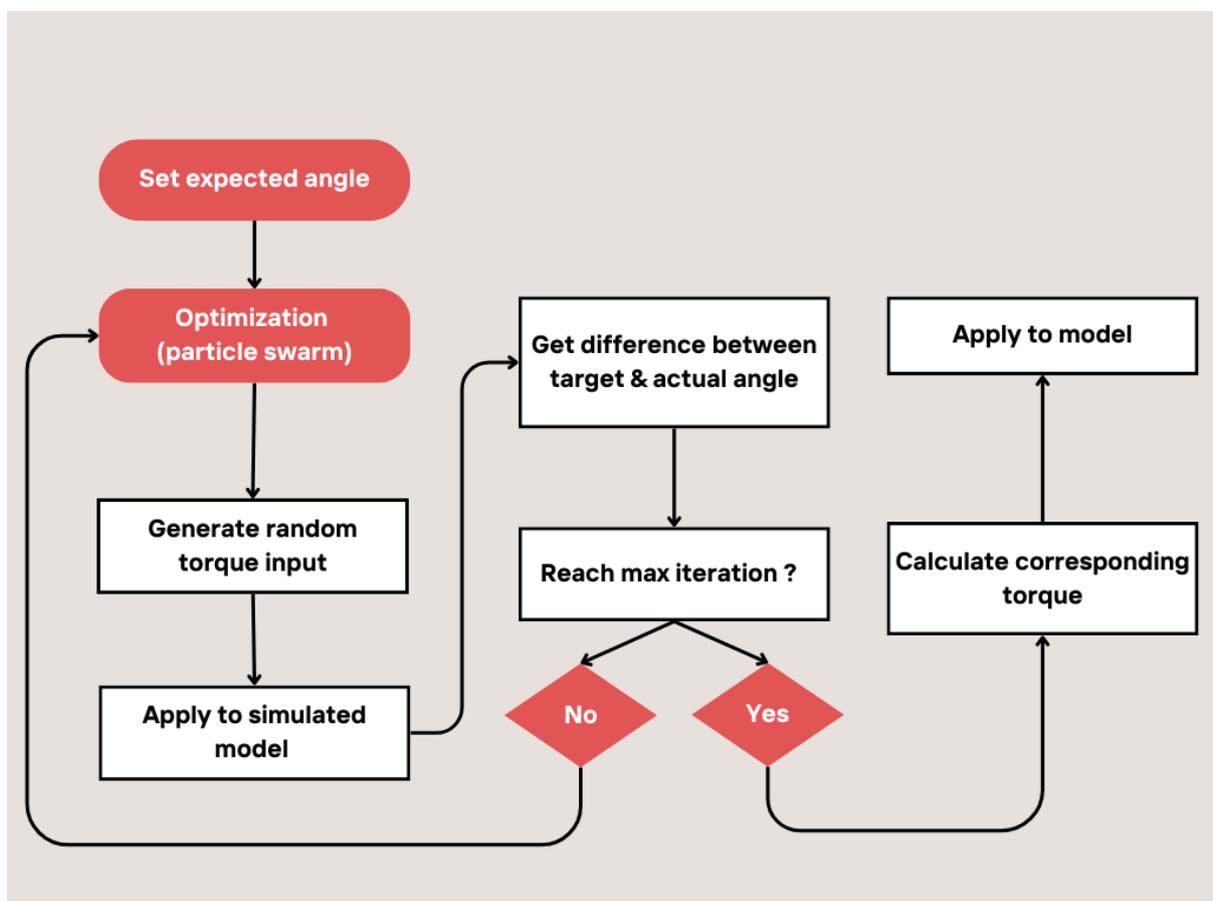
專題成員：楊懷甯

開發工具：Microsoft Visual C++ (MSVC)、Mujoco

測試環境：Windows 11

一、簡介：

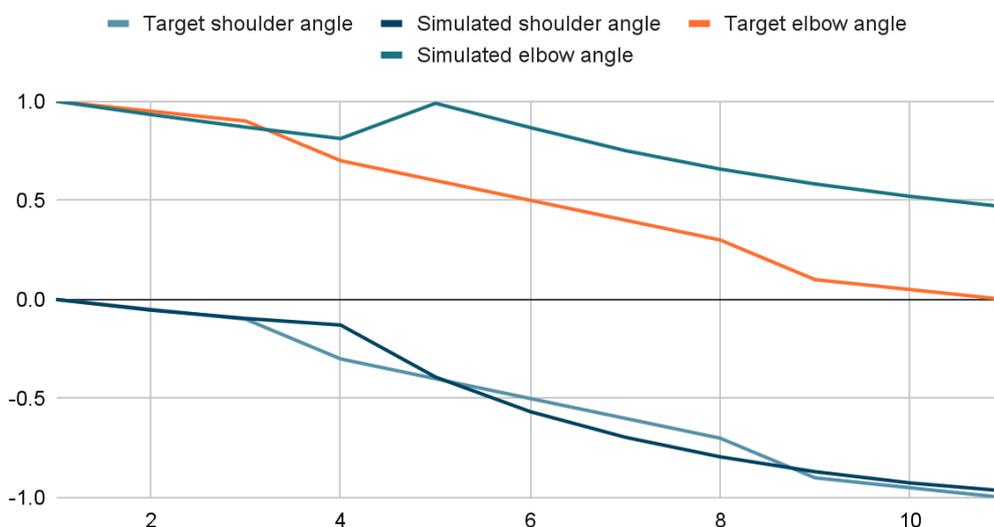
本專題旨在以符合人體力學的方式來模擬人體手臂運動，以省去訓練模型所花費的時間。輸入期望的運動路徑，我們利用Particle swarm optimization最佳化求出手臂做出該運動相對應的力矩，最後將力矩施予模型上並透過mujoco視覺化呈現結果。以下為程式架構圖：



二、測試結果：

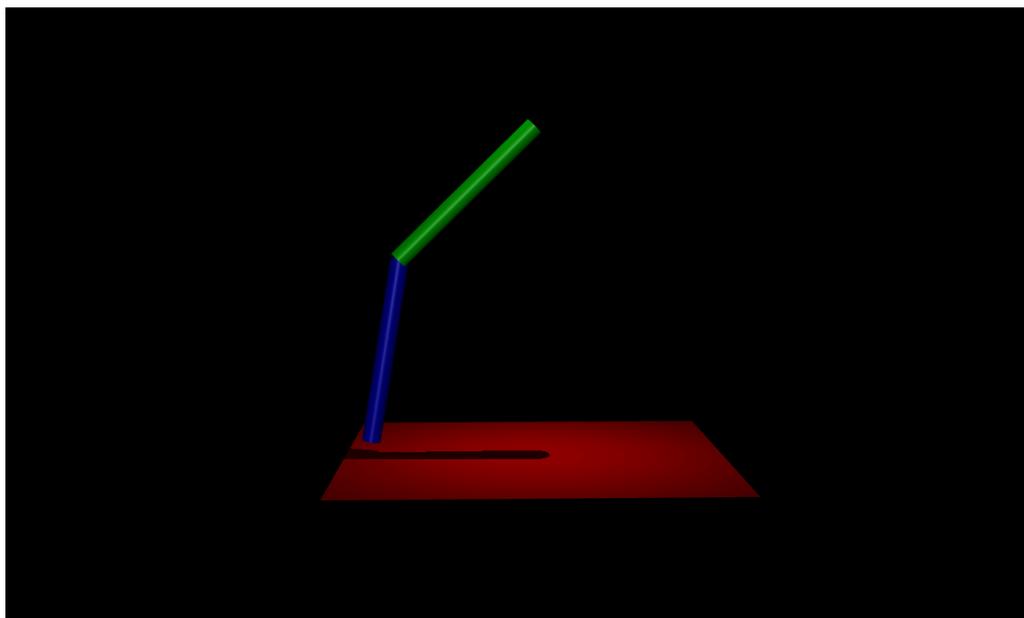
以數據一／模擬時間0.05秒為例：最終最佳化結果，預想角度與目標角度差的平方和落在1.484左右，以下列出目標與實際角度的數據比較：

Target angle & Simulated angle



表(一) 目標角度與實際角度比較圖

對手臂力矩的計算方式採用，我們可以透過（肌肉收縮扭矩－肌肉伸展扭矩）此公式獲得。而影響肌肉收縮／伸展扭矩的參數主要包括兩項：Activation level以及 Angular velocity，Activation level代表肌肉施力的程度，而Angular velocity則為關節角速度，兩者皆會影響肌肉在各情況下所能施予的最大力矩。



圖(一) 模擬畫面(綠色肢段為上臂，藍色肢段為前臂)

文字提示驅動之桌面機器人流程自動化

Desktop RPA by Prompting

指導教授: 李信杰教授

專題成員: 李維軒、歐陽主敬

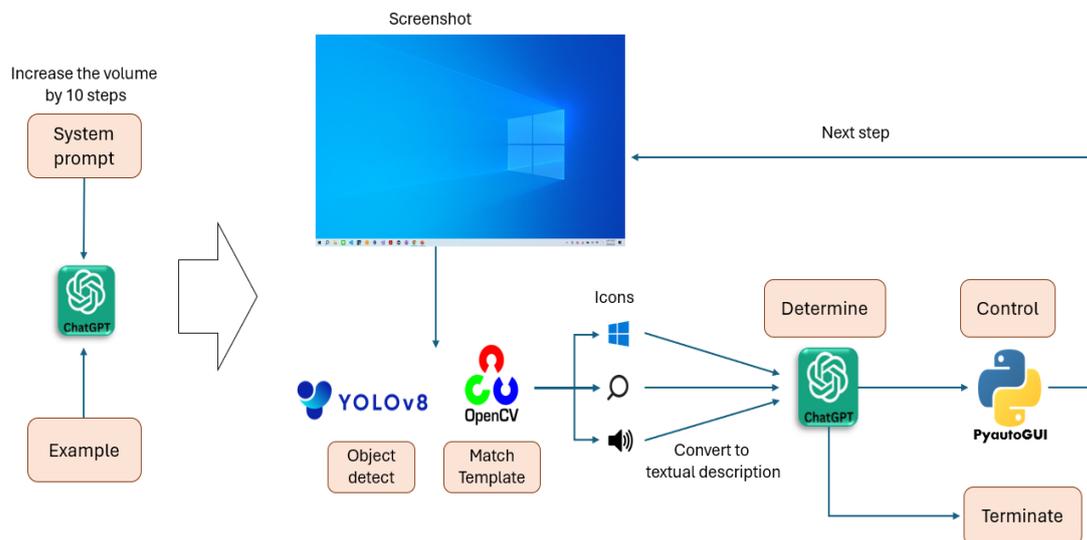
開發工具: GPT4、OpenCV、Python、YOLOv8

測試環境: Windows

一、簡介：

近年來，大型語言模型(LLM)的發展推動了許多與自然語言處理(NLP)相關的研究與發想。本專題結合了 GPT4、OpenCV 的 Match Template、Python 的 PyautoGUI 模組以及 YOLOv8，實作出了一個基於文字提示(Prompt)的桌面機器人流程自動化(Desktop RPA)。只需使用者用自然語言輸入「請幫我關機」、「幫我調高螢幕亮度」、「幫我打開 Edge 搜尋 PChome」等，就可以完全不用動手，單純靠我們的軟體自動去執行這些操作。

以下為系統架構圖：

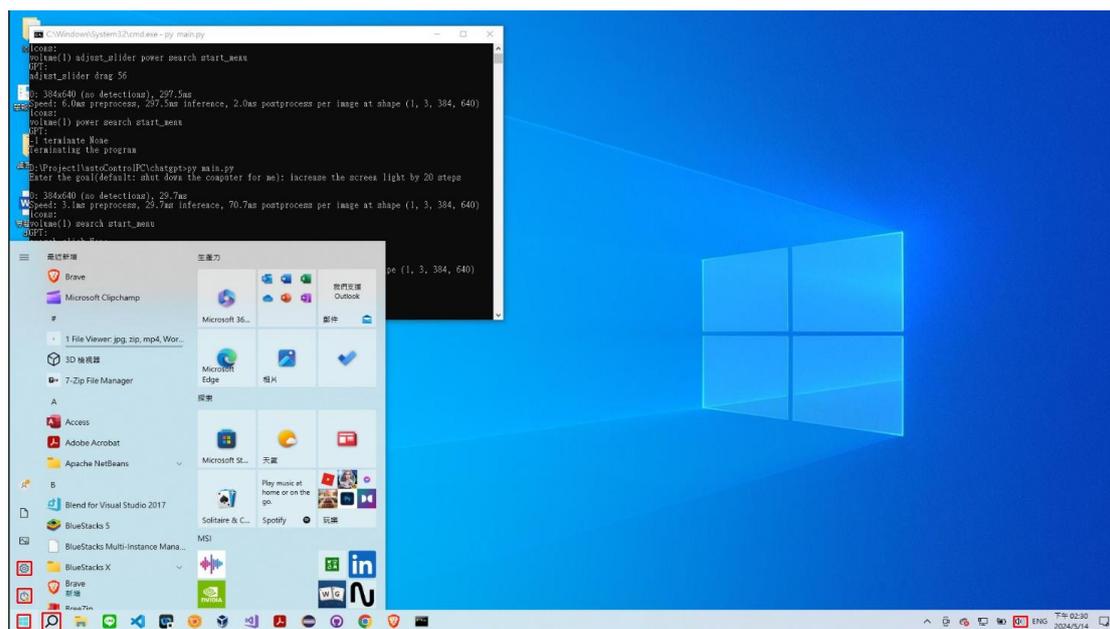


二、 測試結果：

可以看到，當我們下了「幫我調高螢幕亮度 20 格」這個指令時，GPT 成功的依序點擊了開始按鈕->設定->系統，在看到調整螢幕亮度的 slider 後，GPT 滑動了這個 slider 成功調高螢幕亮度

```
1 GOAL: increase the screen light by 20 steps
2 1.
3 icons:
4 volume(1) start_menu (Detect the icons of current screenshot, and used as prompt)
5 ChatGPT:
6 start_menu click None (Click Start menu, no text input)
7 2.
8 icons:
9 volume(1) power start_menu setting
10 ChatGPT:
11 setting click None (Click Setting, no text input)
12 3.
13 icons:
14 searchbar_0 volume(1) start_menu device personalize system
15 ChatGPT:
16 system click None (Click System, no text input)
17 4.
18 icons:
19 searchbar_0 volume(1) adjust_slider power start_menu
20 ChatGPT:
21 adjust_slider drag 56 (Drag adjust slider right by 56 pixels)
22 5.
23 icons:
24 volume(1) adjust_slider power start_menu
25 ChatGPT:
26 -1 terminate None (Complete the task)
```

在下面這個截圖例子中，擷取到了以下 Icon



強化 pythontutor 視覺化工具的 c/c++ 函式附加性及除錯

Strengthen c/c++ function additionality and debugging of pythontutor visualize tools

指導教授：涂嘉恒

專題成員：朱偉哲

開發工具：pathrise-python-tutor

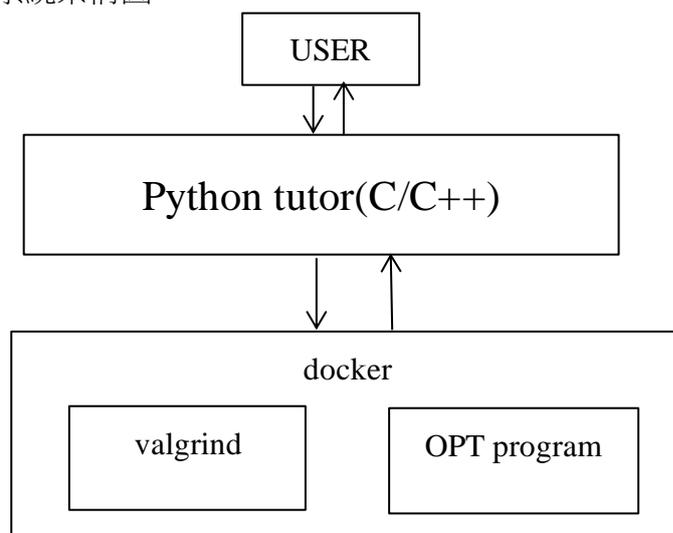
測試環境：Linux Ubuntu 22.04

一、簡介：

python-tutor 的視覺化工具在 C/C++ 的後端部分，已經很久沒有更新了，於是想要嘗試增加或著更新這個工具的部分功能。

首先先簡介此工具，它支援 C/C++、python、java 等語言的部分程式和函式模組的視覺化，再來為運行流程，視覺化工具是由網站將資料傳進之後，再由後端程式去創建 docker 並經由 valgrind 和其餘內部程式使優化輸出的資料最後送回前端由前端生成視覺化圖形模組。

以下為系統架構圖：



這次改善的部分是針對其中 C/C++ 後端的強化，主要是利用版本更新和修改部分後端程式去增加可附加性並強化資料傳輸等功能。

包含了更新 docker 內部的環境和 valgrind 版本以及部分後端程式修正讓其

針對系統與網路管理的學生作業評測系統

Online judge system for system and network management

指導教授：涂嘉恒

專題成員：蔡聿善

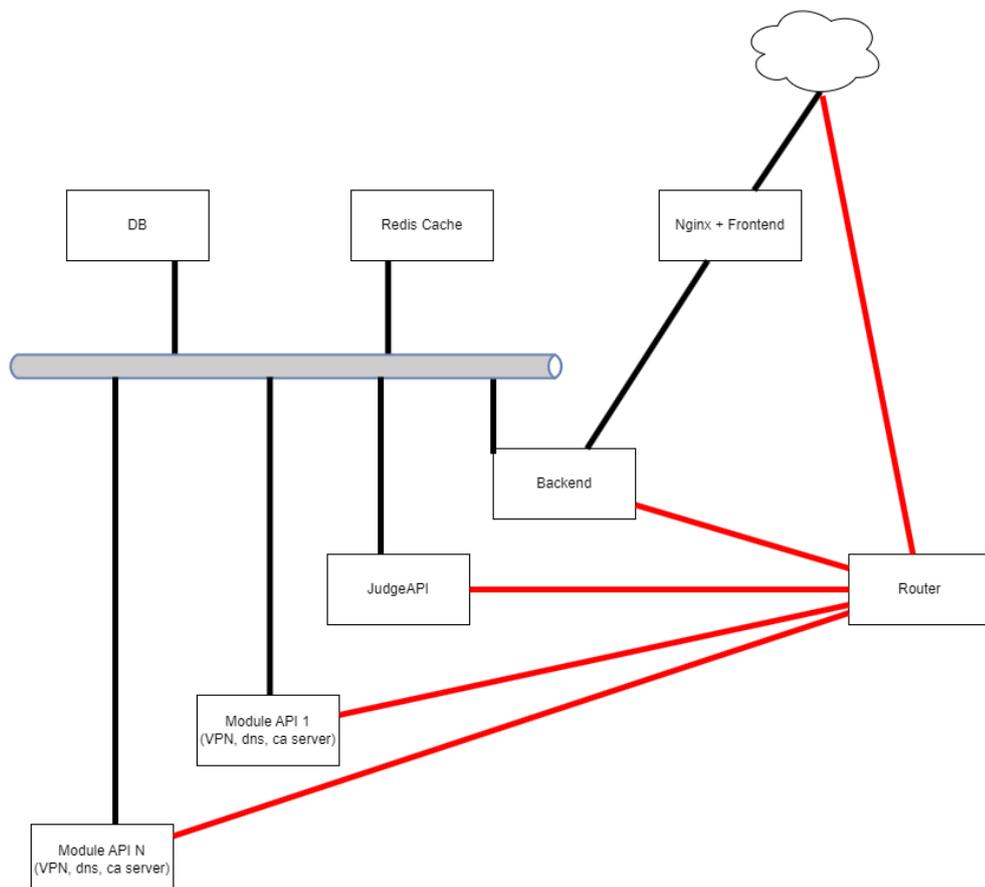
開發工具：Docker, Nginx, Vue.js, Golang1.20

測試環境：Debian GNU/Linux 12 (bookworm)

一、簡介：

本專題目的是設計一套針對系統與網路管理作業的評測系統，同時評測的作業類型可以不局限於系統與網路管理，而是可以用設定檔的方式使其可以用來評測多種資訊科目的作業，比如說程式設計、編譯系統、計算機組織、多平行程式設計等等。

架構上採用 docker 來將整個評測系統拆成很多的部分即依些擴充模組，讓設計作業評測腳本上可以有更多的靈活性。有兩個隔離的網路環境，一個是 API Network（黑色表示），另一個是 Judge Network（紅色表示），API Network 用來處理每個 API 與 database 的溝通以及接上 Nginx 的 reverse proxy，Judge Network 則是專門用來給學生做作業評測的，以系統與網路管理作業為例，會有一個 Module API 是 VPN API 兼 VPN server，學生將虛擬機接入 VPN server 以後，JudgeAPI 可以藉由 Judge Network 連上接入 VPN 的學生虛擬機做作業的評測。



二、測試結果：

學生可以在網頁註冊帳號並綁定學號，綁定學號會限制使用學校 email。

SIGN IN SIGN UP FORGET PASSWORD

User Name _____

E-mail _____

Password _____

Password Confirmation _____

StudentId _____

SIGNUP



學生登入後可以按 **Judge** 做作業的評測。

Hi! jimmy

studentId:

1. Click **Download Config** for the config file (same for every lab)
2. Select the lab you want to submit
3. Upload / Download required files
4. Click the **Judge** button (**A user can only judge once in the same time**)
5. Good luck! :)

Select Lab
Lab01

DOWNLOAD CONFIG

DESCRIPTION

JUDGE

Current Highest Score: 0

Pass rate: 0% (0/0)

1. Click **Download Config** for the config file (same for every lab)
2. Select the lab you want to submit
3. Upload / Download required files
4. Click the **Judge** button (**10s cooldown**)
5. Good luck! :)

Select Lab
Lab01

DOWNLOAD CONFIG

DESCRIPTION

JUDGE

Current Highest Score: 100 **Score 100**

results:

```
{
  "score": 100,
  "results": {
    "general": [
      {
        "message": "SSH Public Key and judge user",
        "ans": true,
        "weight": "15"
      },
      {
        "message": "Install Freebsd",
        "ans": true,
        "weight": "15"
      },
      {
        "message": "rime Zone",
        "ans": true,
        "weight": "10"
      }
    ],
    "wireGuard": [
      {
        "message": "Judge can ping your machine through wireguard",
        "ans": true,

```

透過視覺化可互動式程式碼映射工具展示量子 程式預處理最佳化

Illuminating Quantum Profile-Guided Optimization via Interactive Source-Level

指導教授：涂嘉恆

專題成員：吳昱輝

開發工具：PyQt、SVGWrite

測試環境：Ubuntu 20.04 LTS

一、簡介：

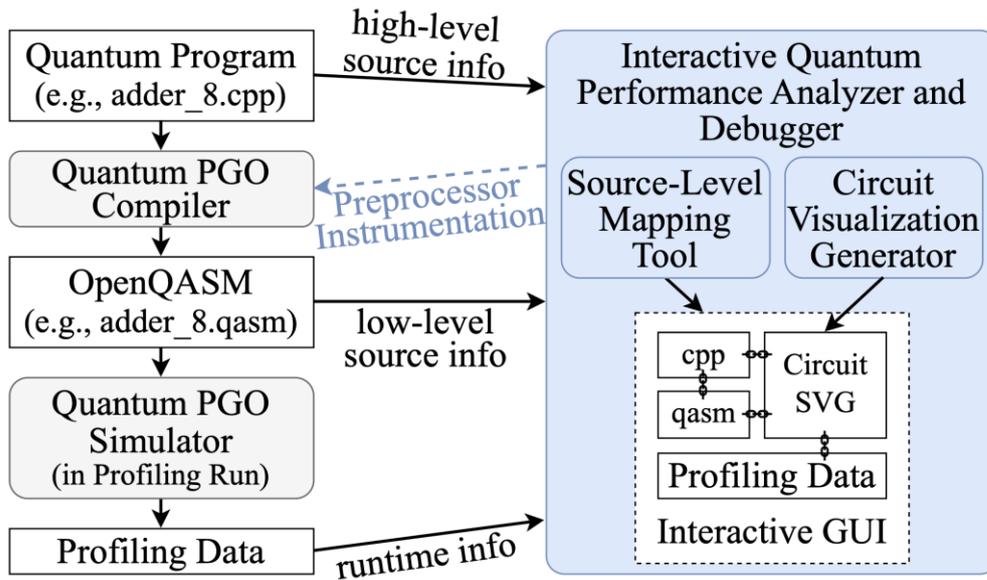
在本專題中使用 QCOR 作為量子程式的編譯器。編譯器的輸入是高階 (high-level) 的程式語言，QCOR 是以 C++ 作為輸入，轉換過後輸出低階 (low-level) 的程式碼，目前實作是以 OpenQasm 作為輸出。當編譯器將高階語言轉換為低階程式碼時，其中組合量子閘 (composite gate) 會被解析為一組由硬體支援的量子閘。當開發人員想要透過量子模擬器來分析程式執行時間的時候，量子模擬器提供的時間分析是基於低階語言的量子閘，而開發人員面對的是高階的程式碼，所以開發人員分析時會受到編譯器的影響。所以我們開發了量子程式的分析和偵錯工具，讓使用者能以更有效率的方式分析程式。此工具分為兩個部分。

第一部分是 Source-level mapping tool，透過改進 QCOR 編譯器行為，在每次編譯器解析每一個 composite gate 時，在其低階語言的量子閘序列的結尾加入一個我們新增的量子閘，稱為 “comment” gate。透過 comment gate 就可以在連續的量子閘序列中判斷出對應的區塊。

第二部份是 Circuit visualization generation。當量子程式執行在量子模擬器上，會產生關於此程式的 profiling data。使用 SVGWrite 套件畫出量子閘執行的量子線路圖 (quantum circuit)，再透過 profiling data 內的時間資料標註每個量子閘在模擬器上的執行時間，提供效率分析用途。

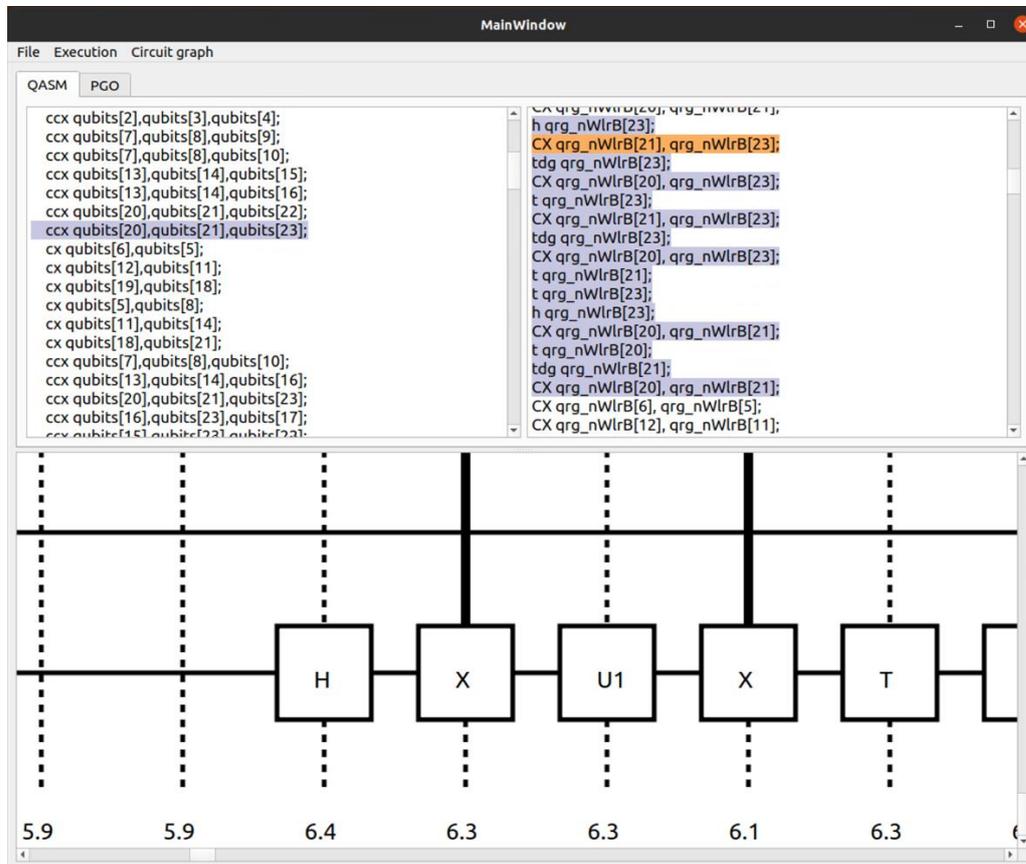
最後使用 PyQt 視窗程式設計的框架，將以上兩種工具整合在 GUI 中，並且透過 Qt 提供的 SVG 物件辨識功能，連結量子線路和原始程式碼，提供使用者互動式的介面。

以下為架構圖：



二、測試結果：

左側顯示 C++ 的原始程式碼，右側顯示經編譯後產生的 OpenQasm 程式碼。下方是量子線路圖。透過滑鼠點擊不同區域會以底色標註該段程式碼的對應區塊。



從想睡到享睡：睡眠腦波分析平台與應用

From Sleepiness to Enjoying Sleep: A Sleep EEG

Analysis Platform and its Applications

指導教授：梁勝富

專題成員：黃品叡、鄧仕煊

開發工具：Python、Figma、Tkinter

測試環境：Windows 10/ Windows 11

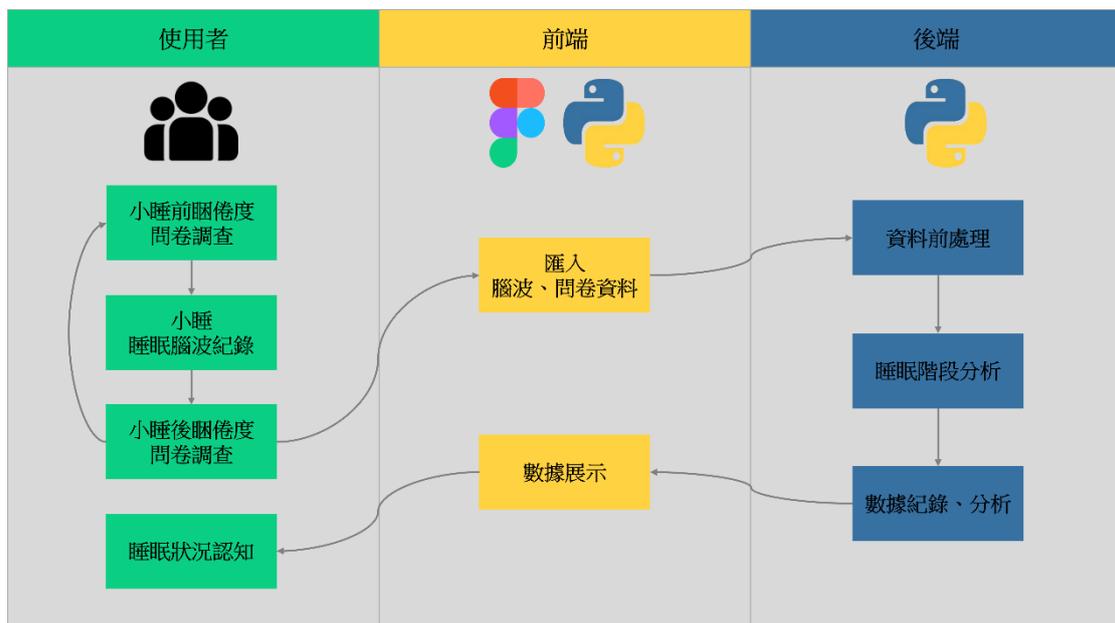
一、簡介：

在現代快節奏的生活中，許多人常常因為工作和生活壓力而感到疲憊不堪。然而，一段適時的小睡能夠在短時間內幫助我們恢復精力、減少疲勞和壓力、改善情緒與創造力，並提高工作效率。因此，小睡已成為許多人日常生活中不可或缺的一部分。

然而，過長的小睡時間可能會導致進入深度睡眠階段，從而在醒來後感到昏昏沉沉，更加疲憊，這種現象被稱為「睡眠慣性」。

為了解決這一問題，我們開發了一個睡眠腦波分析與應用平台。該平台通過記錄使用者小睡時的睡眠腦波數據，以及收集使用者在小睡前後的疲倦度調查問卷，來識別不同的睡眠階段並分析數據之間的關聯性。最後，平台會將分析結果直觀地呈現給使用者，幫助他們對自身的睡眠狀況有更深入的了解，從而科學地管理和優化小睡。

系統架構圖：



二、測試結果：

從實驗的資料分析中，我們發現了兩個重要現象。以下通過兩個實驗案例進行說明：

在實驗一中，受試者在小睡前的疲倦程度為「6 昏昏欲睡；寧願躺下；與睡意搏鬥；頭腦昏沉」。小睡後，受試者的疲倦程度依然為「6 昏昏欲睡；寧願躺下；與睡意搏鬥；頭腦昏沉」。我們通過腦波階段分析發現，受試者在小睡過程中進入了較深層的睡眠階段（N3）。這表明受試者的小睡未能改善其疲倦程度，甚至有更疲倦的傾向。

在實驗二中，受試者在小睡前的疲倦程度為「6 昏昏欲睡；寧願躺下；與睡意搏鬥；頭腦昏沉」。小睡後，受試者的疲倦程度減少到「5 頭腦有些迷糊；開始對保持清醒失去興趣」。我們通過腦波階段分析發現，受試者在小睡過程中僅僅進入了較淺層的睡眠階段（N2）。這表明受試者的小睡確實改善了其疲倦程度。

從這些實驗結果與資料分析中我們可以看出，若我們在小睡的過程中未進入深層睡眠（N3），疲倦程度通常能夠得到改善；反之，若我們在小睡的過程中進入了深層睡眠（N3），疲倦程度不僅可能無法改善，甚至可能感到更加疲倦。

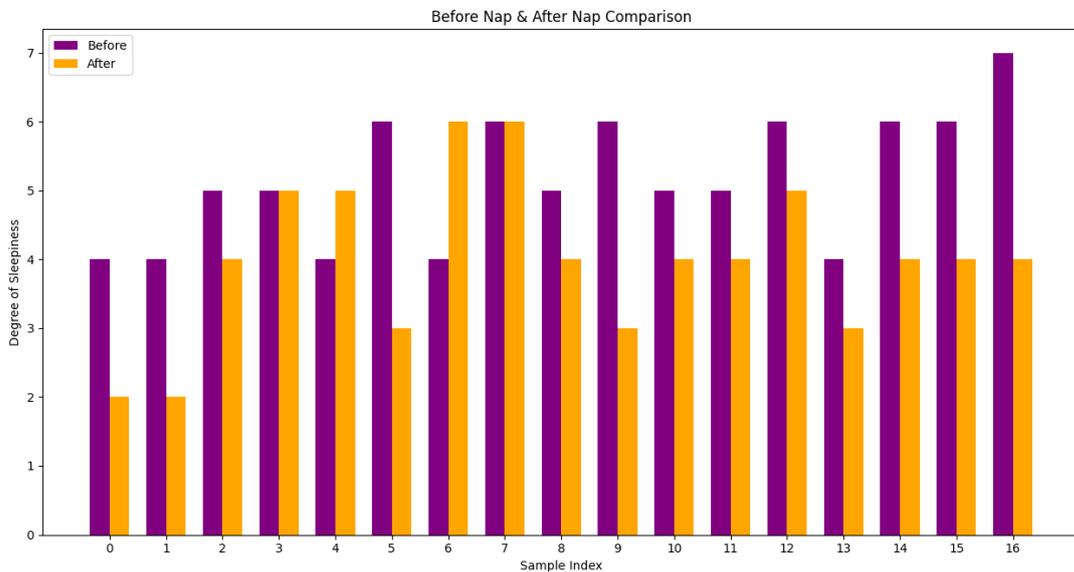


圖 1、小睡前後疲倦程度比較

基於NILM的用電分析與建議

Electricity consumption analysis and suggestions based on NILM

指導教授：莊坤達

專題成員：黃友枚、孫以璿、練智剛

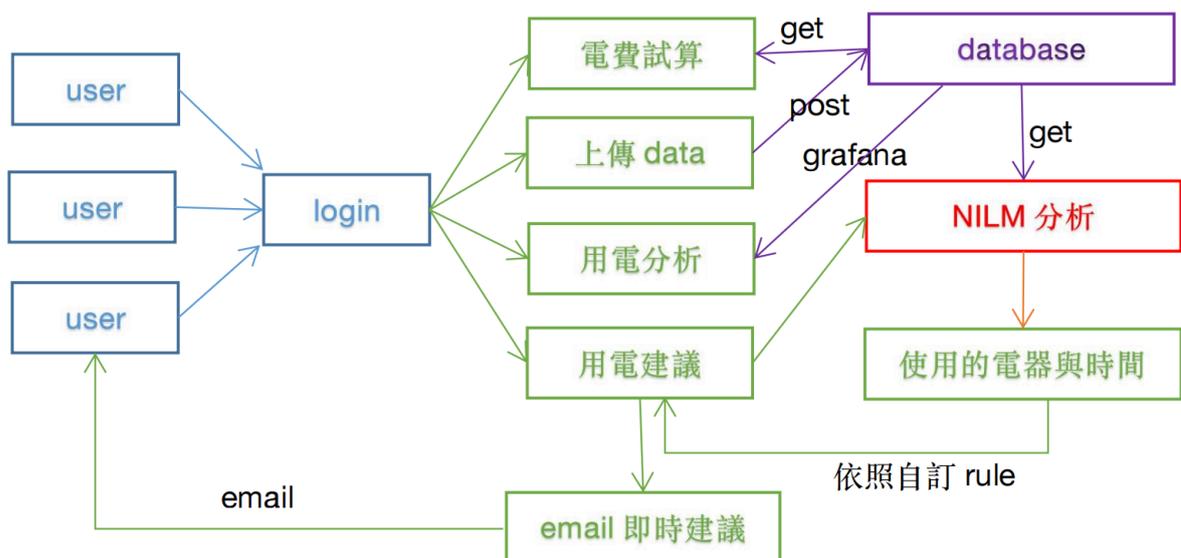
開發工具：Python, React, Js, Css,

測試環境：Linux

一、簡介：

我們專題做的是基於非侵入式負載監控技術（NILM）的用電分析與建議系統。結合我們開發的時間電價系統，可以讓用戶實時監督自己的用電情況與電費。時間電價系統的試算功能可以讓用戶上傳用電數據去算各項電費方案的預估電費，並給出最適合該用戶的用電方案。NILM技術則可以搭配電力系統，通過分析家庭總電壓和電流的變化，推斷出家中使用的電器及其各自的能耗情況。NILM不需要在每個電器上單獨安裝監控設備，加強了對用戶的隱私保護。我們的系統會在根據用戶的用電習慣提供節能建議，以幫助用戶優化用電時間和方式，從而達到降低電費節能減排的目的。

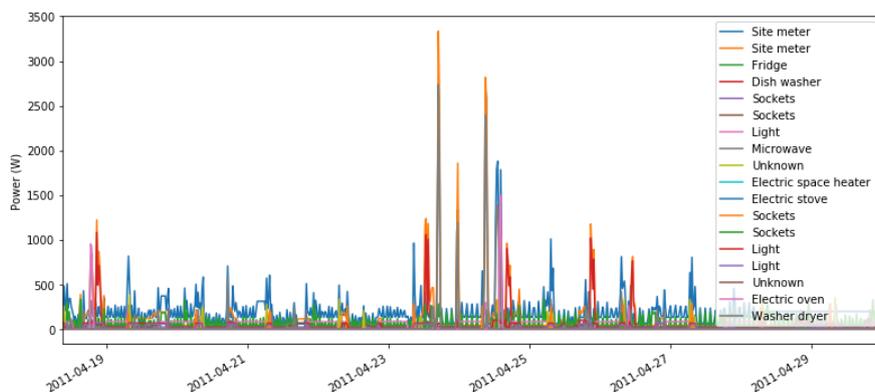
系統架構圖：



圖一：系統架構圖

二、測試結果：

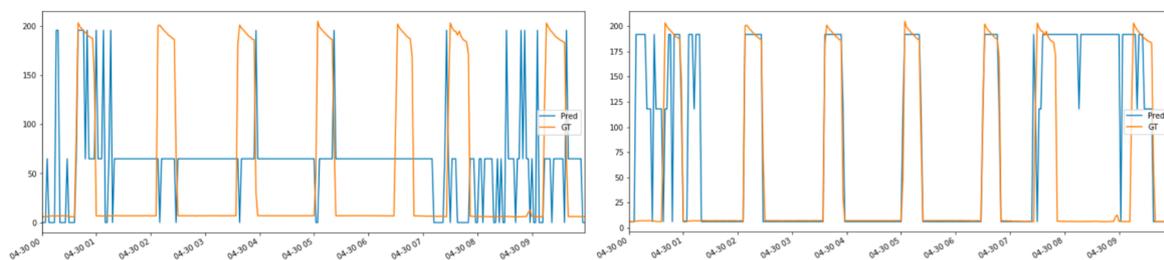
這是使用Reference Energy Disaggregation Data Set (REDD)數據所生成的電器用電圖。我們用 building 1 進行示範，用電數據經過模型訓練後，可以被切割分類去預測用戶在每個時間點所使用的電器。



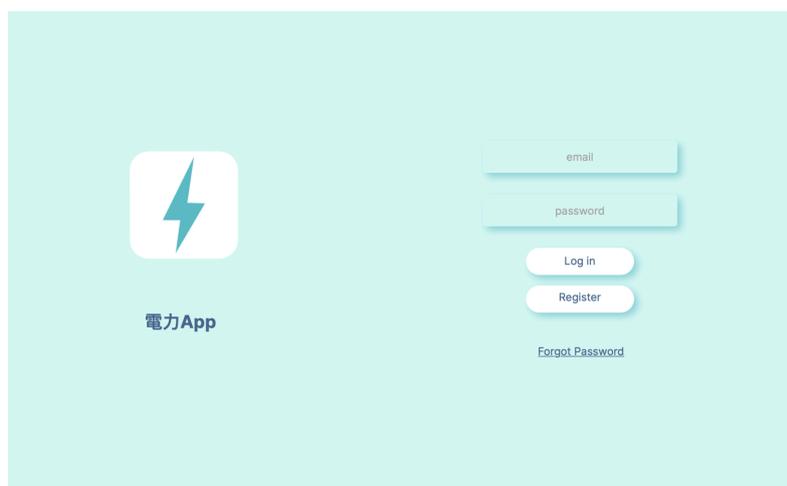
圖二：NILM 電器用電切割分類圖

模型使用不同演算法所訓練出的預測值與測試標籤的值之間的差異。

左圖：CO 演算法，右圖：FHMM 演算法



圖三：CO 和 FHMM 演算法預測與標籤的差異線型圖



圖四：時間電價系統

Timescale DB 在系統資料的效能優勢與應用

Application of Timescale DB in data processing

指導教授：莊坤達

專題成員：易子捷、陳鈞天

開發工具：Python、FastAPI、React

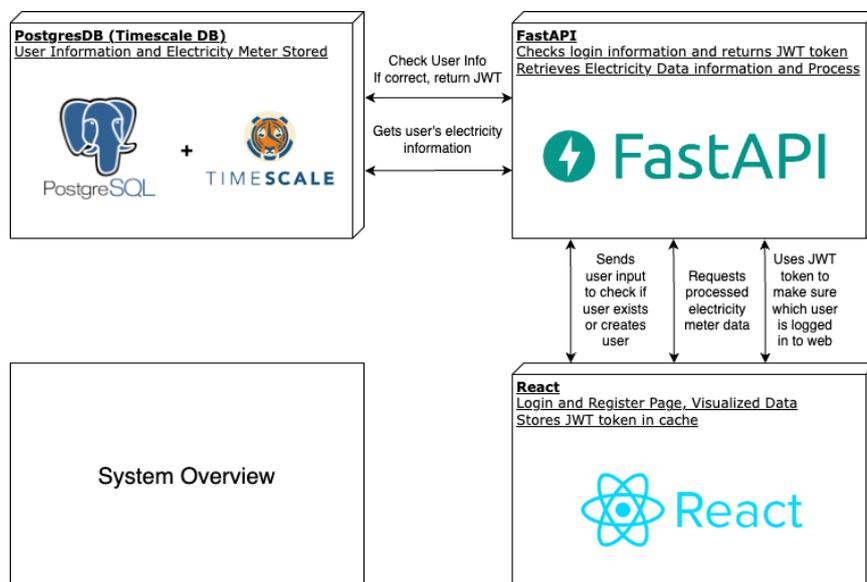
測試環境：Ubuntu 22.04.4 LTS

一、簡介：

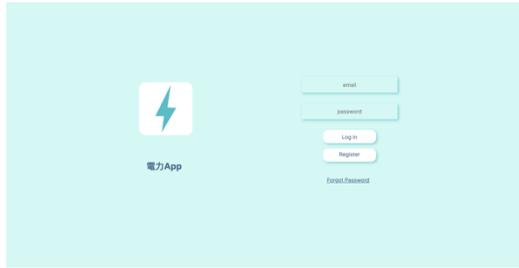
基於2020年總統盃黑客松比賽，我們實作了一個可以分析用電情況的系統。在此系統中，因為資料是屬於時間序列類型的電表度數資料，因此選擇的資料庫需要能支援以下幾點，一、短時間的大量寫入，二、管理時間資料的生命週期，三、壓縮時間資料以減少儲存開銷以及四、能夠查詢不同時間的資料。根據上述幾點，如果單純用 Postgres 去儲存以及查詢資料會造成延遲過久，因此，我們決定使用一個以 Postgres 為基底去打造，能被用來處理時間序列資料的 Timescale DB。其最大的優點在於，因為是根據 Postgres 為底去打造的資料庫，因此會寫 SQL 語法就能夠操作 Timescale DB。

我們的系統功能大致有以下幾點，一、使用者的帳戶註冊與管理；二、呈現使用者當天用電度數；三、視覺化呈現使用者的用電；四、使用者可自訂一個用電度數上限，當其用電超過上限會發出信件警告；五、根據使用者目前電表資料，計算出三個不同電價方案的收費金額(累進、兩段式時間電價、三段式時間電價)，使其能夠根據此價錢選擇有利的方案；六、試算功能，讓使用者能上傳自家電表資料，並如同功能五計算出三個方案的收費金額；七、計算因其用電所造成的碳排放量。

以下為系統架構圖：



二、測試結果：



圖一：系統登入畫面



圖二：使用者註冊



圖三：系統首頁



圖四：電費試算



圖五：使用者用電資訊

```
netdb=# SELECT default_version, installed_version FROM pg_available_extensions
where name = 'timescaledb';
 default_version | installed_version
-----+-----
 2.14.2         | 2.14.2
(1 row)
```

圖六：Timescale DB 架設在 Postgres 之上

三維室內場景重建

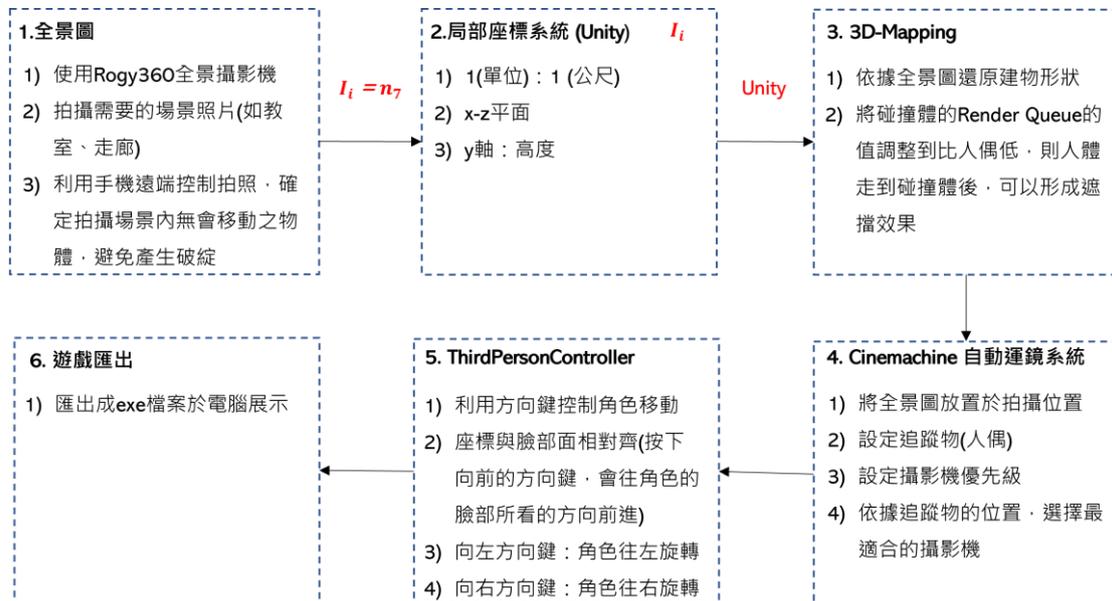
3D Indoor Scene Reconstruction

指導教授：連震杰
專題成員：楊采語
開發工具：Unity
測試環境：Unity 2019

一、簡介：

因為傳統室內3D建模需使用昂貴的Lidar測距裝置，以獲得室內的深度資訊，並由人工方式完成室內3D模型建模，且使用點雲建模會使得模型檔案龐大，有過多不需要的資訊。所以本研究使用Rogy360全景攝影機來拍攝室內場景的全景照片，於一個室內空間只需拍攝一張照片，再藉由Unity來將室內的建物還原，並拼接不同的場景，讓使用者在移動角色時，有如「走到另一個房間」或是「走到柱子後面」的感覺，另外還有利用全景攝影的功能，並設計觸發裝置，讓角色走到感應門附近就會如現實世界中自動開門的效果。

以下為系統架構圖：

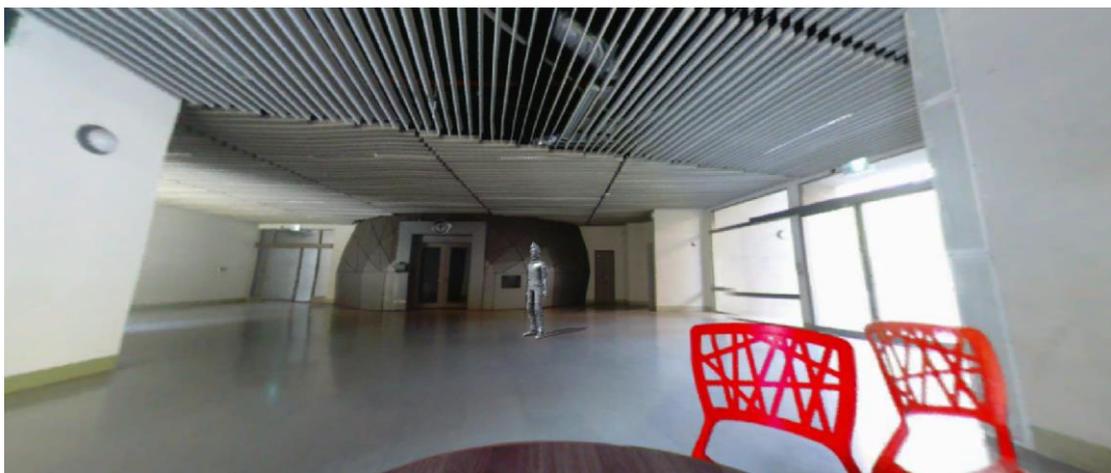


二、測試結果：

資訊系一樓65104、65105外走廊畫面



滑鼠廳外畫面



測量系教室測試畫面(可以站在講桌後面)



應用透視投影和光流追蹤技術於擴增實境排球場

Augmented Reality-Based Volleyball Court using Perspective Projection and Optical Flow

Tracking

指導教授：連震杰

專題成員：龍奕全

開發工具：python、opencv、numpy、pyQt5

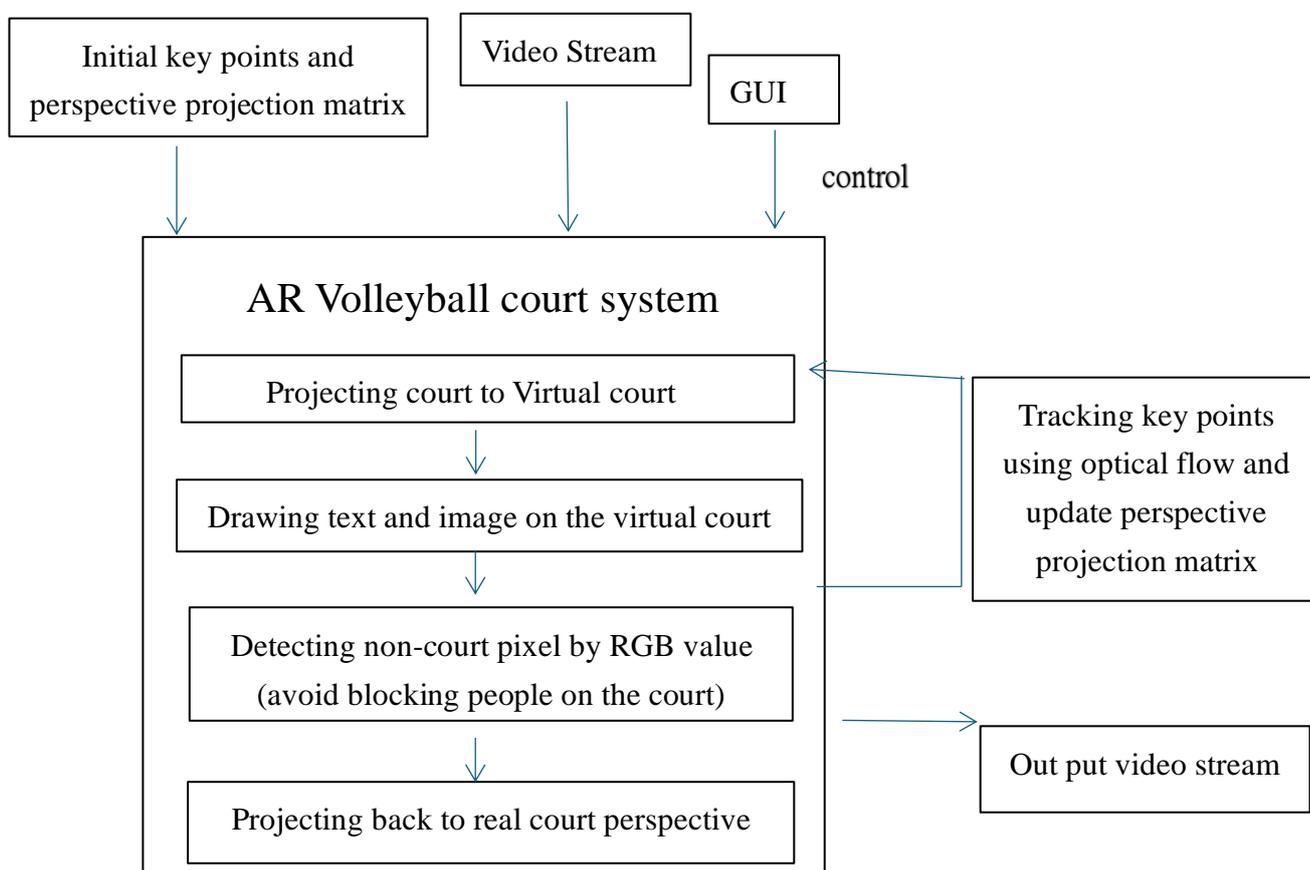
測試環境：Window11、Python(Anaconda)

一、簡介：

此次的專題研究利用透視投影以及光流追蹤的影像處理技術，搭配採樣像素 RGB 分布來區別背景與人物，使人物不會被 AR 投影阻擋，實做出可以自由投影內容在場上的擴增實境排球場，其中應用包含：比數、隊徽、隊名、倒數計時、球場九宮格等資訊之投影。

此專案也在台灣企業排球聯賽來成大中正堂比賽時實裝，轉播給現場以及線上之觀眾，成功提升觀眾之觀賽體驗。

系統架構圖：



二、測試結果：

在測試時效果良好，處理速度可以達到30fps 即時處理。讓觀眾可以有流暢觀賞體驗。而在影片視角移動的情況下，使用 optical flow 可以在 key points 不被遮擋的情況下使透視投影的矩陣正確更新，使得投影內容不會因為視角移動而產生偏差。



圖一：實際畫面



圖二：實際畫面

可生成摘要與輔助教學的 Chatbot 及高效學習平台

Advanced Learning Video Platform Integrating Automated Summary Generation and Tutoring Chatbot Support

指導教授：郭耀煌

專題成員：潘駿諺、吳仲倫、臣韞儒、曾奕程

開發工具：React, FastAPI, SQLite, llama_index, tolen/alpaca-lora, OpenAI API, lm-sys/FastChat,

OpenAI Whisper

測試環境：Linux Ubuntu Ubuntu 20.04.6 LTS x86_64

一、簡介：

近年來，隨著像 Coursera、Udemy 等線上教學平台如的蓬勃發展，以及網路基礎建設的持續升級，線上學習已成為現代人不可或缺的學習方式。Coursera 統計出 2016 年至 2021 年使用此平台學習的人數從 21M 成長到 92M。然而，學生在使用線上教學平台時常面臨資料查找效率低下、沒有助教可以討論和遺忘影片內容等問題，導致浪費寶貴時間。為此，我們利用 AI 技術打造一個專門解決上述問題的影音學習平台，主要包括以下的改進：

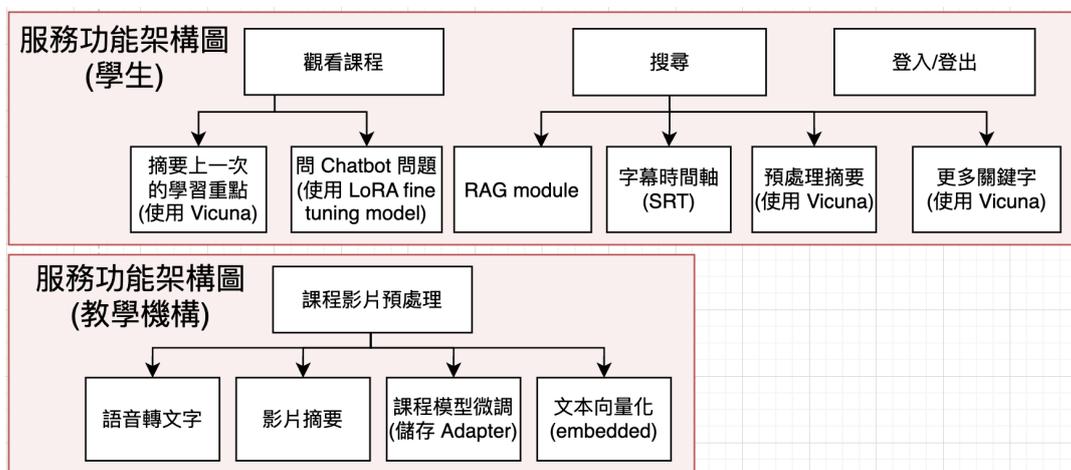
1. 資料查找效率提升

由於語言模型先天的 token 限制，我們用長文本摘要方法來濃縮課程影片精華；並在搜尋時提供基於 RAG (Retrieval-Augmented Generation) 的文字摘要以及關鍵字；最後，用模糊搜尋來尋找、排序課程，搜尋結果標記出與關鍵字相關的時間戳記，讓學生能夠立即找到所需內容。

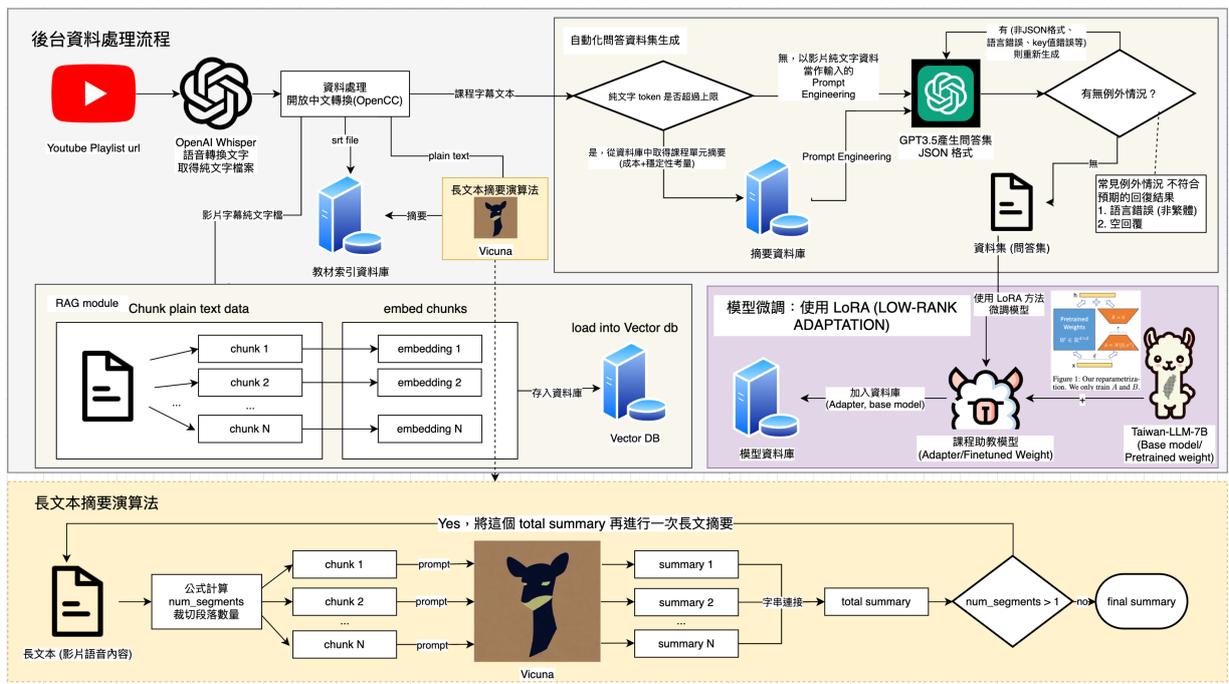
2. 數位化助教服務以及遺忘內容補充

利用 OpenAI whisper 將所有課程語音轉文字，透過 OpenAI 的 API 使用 Prompt engineering 技術自動產生問答集資料，將資料透過 LoRA (Low-Rank Adaptation) 方法微調，如此一來就能在本地端用低成本為每個課程訓練專屬模型 (Adapter)，讓學生在遇到問題時能夠提問；並紀錄學生最後一次觀看影片的時間戳記，並摘要先前的學習內容，加快複習腳步、提高學習效果。

以下為功能架構圖、資料處理流程和長文本摘要流程圖：



圖一、功能架構圖



圖二、功能架構圖以及長文本摘要演算法

二、測試結果：

1. 學生搜尋關鍵字，看到 RAG 檢索結果，如圖三。
2. 直接從與關鍵字相關之時間戳記進入課程觀看影片，如圖四。
3. 在課程影片頁面右下角開啟聊天泡泡，向該課程助教模型問問題，如圖五、六，可以發現在對不同模型詢問相同問題得到不同的回覆。



圖三、搜尋畫面



圖五、ChatBot (國中生物模型)



圖四、影片播放



圖六、ChatBot (國中公民模型)

基於機率模型設計與實作演算法預測麻將手牌

Designing and implementing algorithm to predict Mahjong hand tiles based on probability models

指導教授：陳奇業

專題成員：周嘉豪

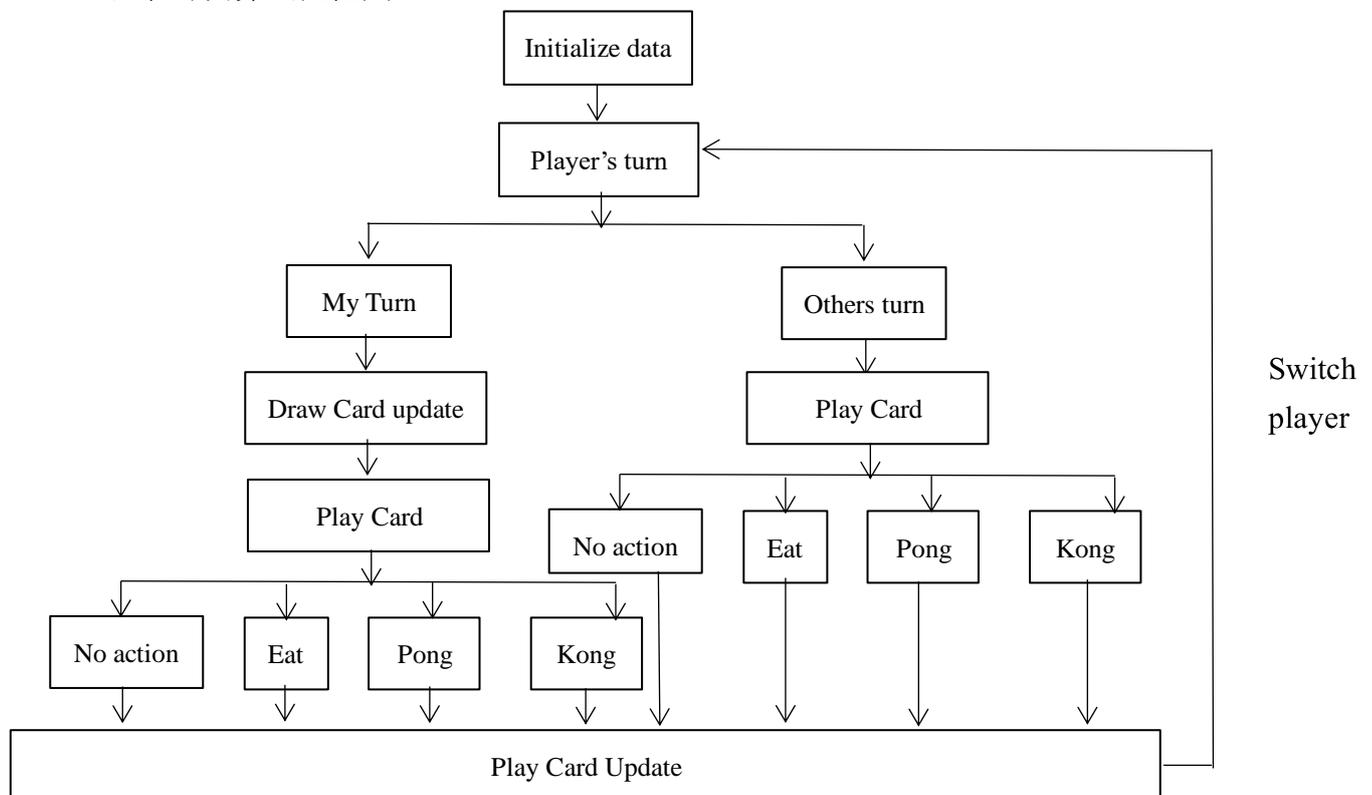
開發工具：GCC

測試環境：Windows 10

一、簡介：

AI 現今發展蓬勃，很多棋牌類 AI 甚至超越了職業選手的實力，如在圍棋、五子棋等方面，而在日本麻將中，也是有各式各樣策略的 AI。而麻將作為一個四人的棋牌類遊戲，其策略更為複雜，但同時也吸引更多人參與。而麻將中最害怕的就是“放槍”，因為輸的分數會是最多的，所以能預測對手的牌型能大概知道對方在“聽”哪一張牌，所以本專題希望透過比較簡單的演算法，來計算其他玩家的牌型分佈，從而大大減自身放槍的機會。實作的演算法主要通過玩家的“吃、碰、槓”行為，再用機率來調高或調低其他玩家不同牌型在手牌中的期望值。

以下為演算法流程圖：



二、測試結果：

1. 輸入初始化數據

```

C:\Users\Dell\OneDrive\桌面\資訊專題\mahjong\main.exe
請輸入起始玩家:0
請輸入你的手牌:19 30 31 23 12 13 14 33 7 8 8 27 1
請輸入懸賞牌:15
請輸入你所抽到的牌面:34
請輸入你打出的牌:30
是否有吃碰槓發生(輸入0為沒有,1為有):0
    
```

2. 依次輸入玩家行動

```

C:\Users\Dell\OneDrive\桌面\資訊專題\mahjong\main.exe
請輸入起始玩家:0
請輸入你的手牌:19 30 31 23 12 13 14 33 7 8 8 27 1
請輸入懸賞牌:15
請輸入你所抽到的牌面:34
請輸入你打出的牌:30
是否有吃碰槓發生(輸入0為沒有,1為有):0
請輸入 player 1 打出的牌:2
是否有吃碰槓發生(輸入0為沒有,1為有):0
請輸入 player 2 打出的牌:28
是否有吃碰槓發生(輸入0為沒有,1為有):0
請輸入 player 3 打出的牌:28
是否有吃碰槓發生(輸入0為沒有,1為有):0
請輸入你所抽到的牌面:30
請輸入你打出的牌:31
是否有吃碰槓發生(輸入0為沒有,1為有):0
請輸入 player 1 打出的牌:_
    
```

3. 查看玩家手牌預測

```

請輸入 player 1 打出的牌:0
請輸入要查看的玩家B-table:1
Player1的期望值表為:
一萬 二萬 三萬 四萬 五萬 六萬 七萬 八萬 九萬
0.006257 0.003614 0.097214 0.448313 0.491104 0.491104 0.368328 0.245552 0.491104
一筒 二筒 三筒 四筒 五筒 六筒 七筒 八筒 九筒
0.491104 0.491104 0.368328 0.368328 0.368328 0.368328 0.491104 0.491104 0.491104
一索 二索 三索 四索 五索 六索 七索 八索 九索
0.368328 0.491104 0.491104 0.491104 0.368328 0.491104 0.491104 0.491104 0.368328
東 南 西 北 白 發 中
0.202751 0.491104 0.183040 0.272318 0.491104 0.368328 0.368328
    
```

4. 比賽例子

```

Player3的期望值表為:
一萬 二萬 三萬 四萬 五萬 六萬 七萬 八萬 九萬
0.012722 0.517383 0.625616 0.776703 0.582527 0.768830 0.274591 0.075297 0.000758
一筒 二筒 三筒 四筒 五筒 六筒 七筒 八筒 九筒
0.011485 0.440566 0.271189 0.329616 0.128756 0.003457 0.189133 0.344493 0.582527
一索 二索 三索 四索 五索 六索 七索 八索 九索
0.658056 0.836386 0.822212 0.776703 0.582527 0.776703 0.416992 0.280389 0.000250
東 南 西 北 白 發 中
0.179189 0.577119 0.135635 0.241705 0.776703 0.000004 0.003774
    
```



三萬 四萬 五萬 六萬 六萬 一筒 二筒 三筒 五筒 六筒
 七筒 二索 三索

FIT IN：個人化體感健身程式

FIT IN : A Personal Motion Sensing Fitness App

指導教授：陳奇業

專題成員：林志芸、熊貫仲

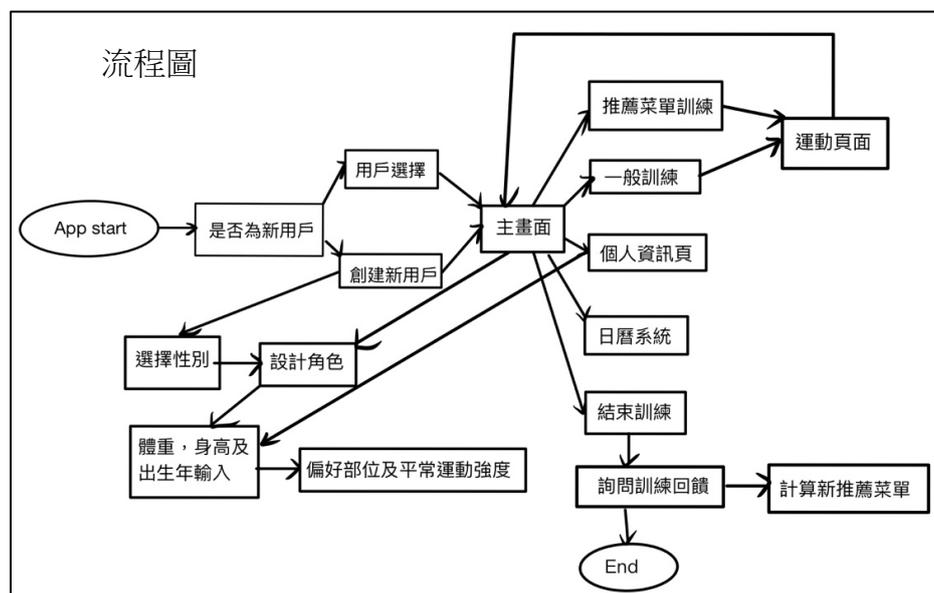
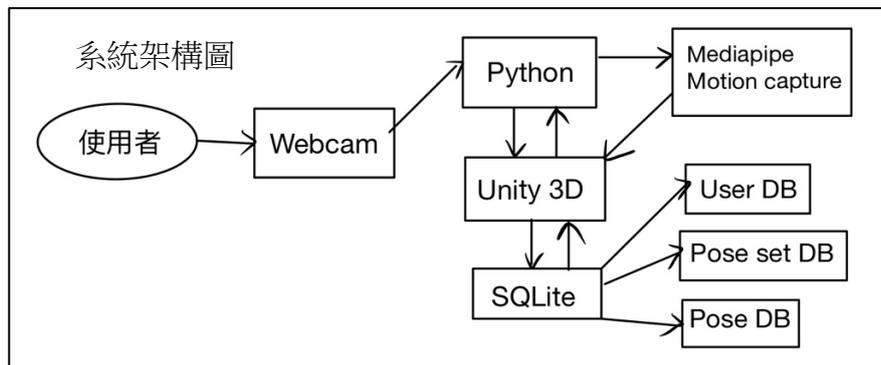
開發工具：Python、Unity、SQLite

測試環境：Windows 10

一、簡介：

隨著現代社會生活節奏加快，以及 COVID-19 疫情導致人們在家工作的頻率增多，去健身房的次數減少，又在不可忽視身體健康的狀況下，居家健身儼然成為了新的趨勢。但獨自一人在家運動不可避免地會出現錯誤的姿勢，這不僅影響運動效果，還可能對身體造成負擔。

FIT IN 體感健身程式結合了：Python Mediapipe 偵測人體骨架關鍵點、App 開發工具 Unity 將其視覺化、並透過 Socket 進行實時溝通。我們採用了類似於 Xbox Kinect 的體感概念，致力讓使用者有更佳的體驗，且能夠幫助用戶個人化的管理自己的健身計劃和身體狀況。



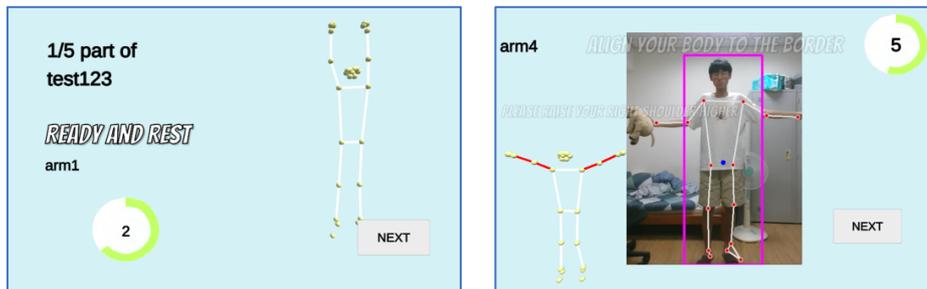
二、測試結果：

1. 語音系統：

透過 TTS 將目標句子生成為語音，模擬健身教練與使用者對話的情境。

2. 影像辨識：

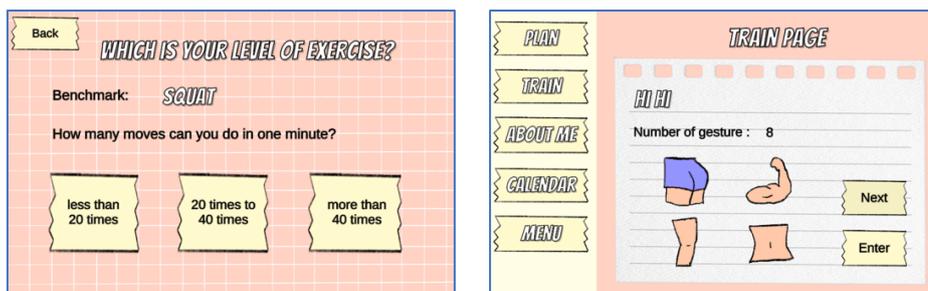
- A. 動作偵測：使用 **Mediapipe** 抓取人體關鍵點，進行實時角度計算，並且同步與目標動作之關鍵點角度比較，若差異超過一定的閾值，錯誤的身體部位將會顯示為紅色，視覺化提醒使用者調整姿勢。



- B. 體感控制：使用 **Mediapipe** 抓取手部關鍵點，經過計算後，投射其二維座標至程式畫面上，使用者不需要透過滑鼠或鍵盤來操控程式進行。

3. 推薦系統：

在創建角色時將會詢問偏好運動部位以及平時的運動強度，另外在每日結束程式前也會請使用者給予回覆，我們透過演算法的設計來動態調整不同 label 的權重，再次推薦出新的運動菜單。



4. 個人化角色管理：

我們參考 **Wii** 的角色管理模式，在單一個裝置即可儲存多人資料，並在主頁面隨時切換角色來進行運動，適合家庭使用。



應用於 SAR 影像之船隻偵測演算法

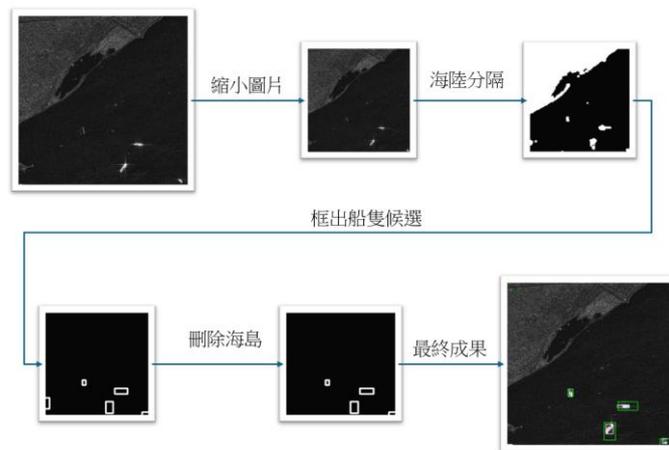
Ship Detection Algorithm in SAR Imagery

指導教授：陳培殷
專題成員：趙邦鈞
開發工具：python
測試環境：windows 11

一、簡介

合成孔徑雷達（Synthetic Aperture Radar，SAR）是一種主動式遙感技術，利用雷達發射信號並接收反射回來的信號來生成地表的高解析度影像。相較於光學影像，SAR 具有獨特的優勢，因其不受天氣和光線條件限制、穿透能力強，故而適用於各種自然環境下的船隻監測。

在本專題中，提出了一種利用面積大小來偵測船隻的方法。首先，對 SAR 圖片進行縮小以突顯特徵。接著，使用 Otsu's method 來對圖片進行二值化，從而實現海陸分隔。Otsu's method 根據影像的色彩平均值給出不同的閾值，因此能夠更有效地應對不同的影像。接下來，根據每個輪廓的面積篩選可能是船隻的目標並透過分析輪廓的長寬比例和中心點位置，可以排除那些可能被誤認為海島的輪廓。這種方法結合了影像處理技術和特徵分析，能夠有效地從 SAR 影像中自動識別和定位船隻。

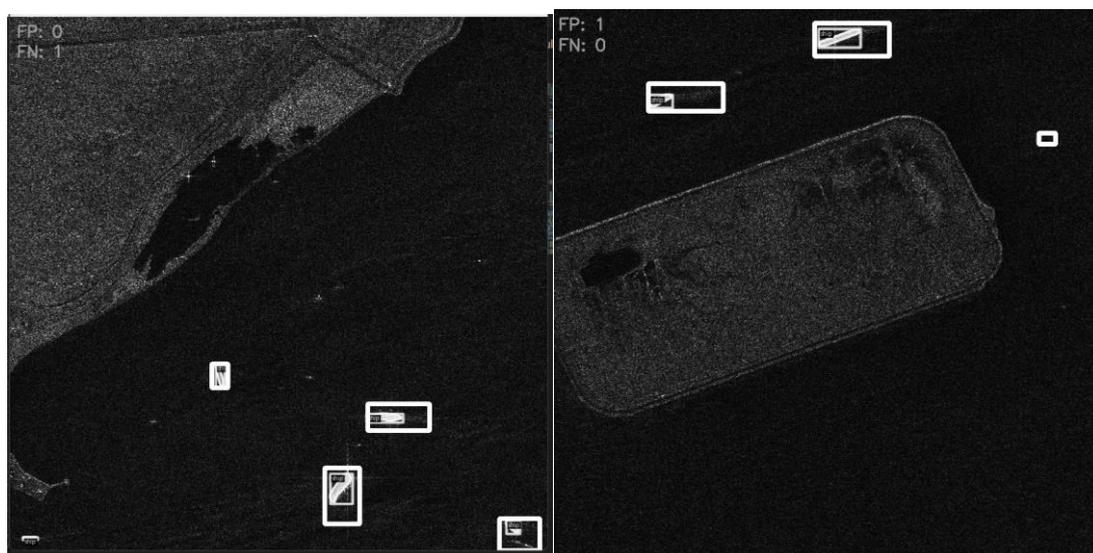


圖一:流程圖

二、測試結果

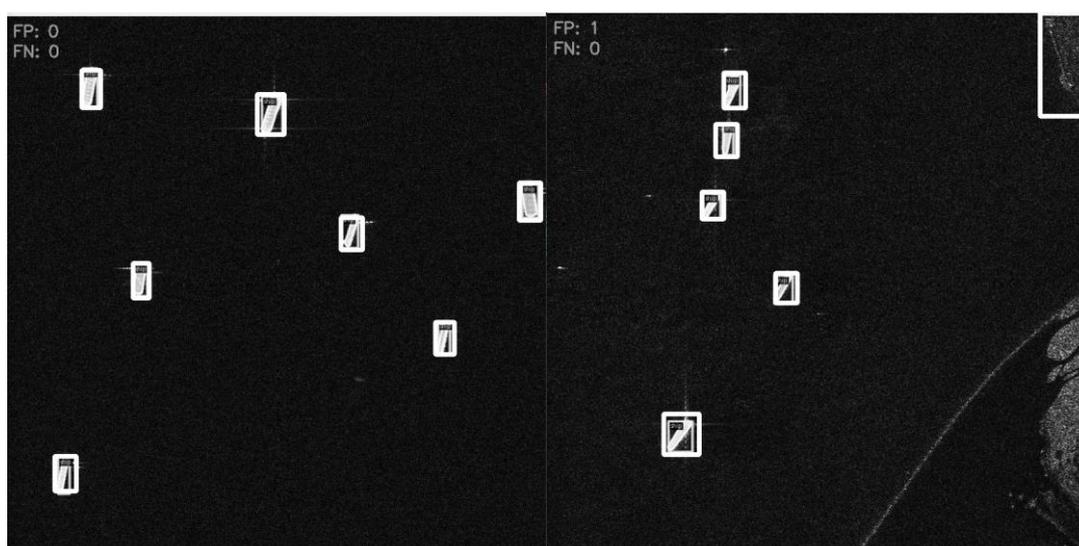
使用 High-Resolution SAR Images Dataset (HRSID)進行測試，HRSID 包含 99 張的 Sentinel-1B，36 張的 TerraSAR-X 和 1 張 TanDEM-X。這些衛星影像再被切割成 800x800 的子影像。

使用混淆矩陣（confusion matrix）衡量偵測能力。在混淆矩陣中，FP（False Positive，偽陽性）代表偵測器誤判為正例（船隻）的數量，FN（False Negative，偽陰性）則代表未能偵測到實際存在的正例（船隻）的數量。下面圖片中，厚框為經過演算法偵測出來的結果，薄框為實際上的船隻。



圖二:船隻偵測結果

圖三:船隻偵測結果



圖四:船隻偵測結果

圖五:船隻偵測結果

研究所入學資料整合平台

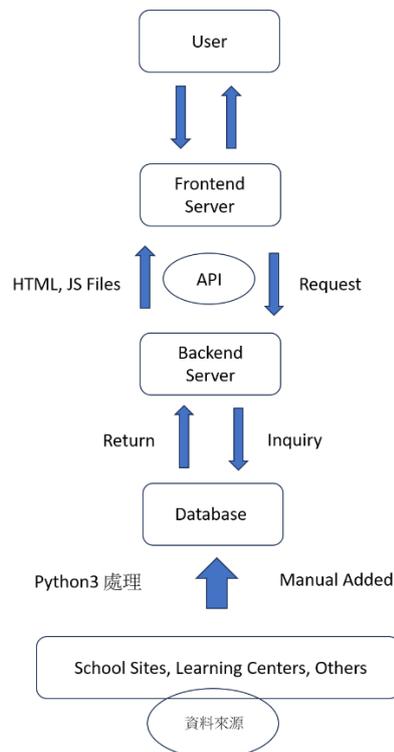
Integration Platform Of Graduate School Admission Data

指導教授：賀保羅
專題成員：傅靖淳
開發工具：python, php
測試環境：Windows

一、簡介：

鄰近畢業，大家紛紛開始尋找未來方向，其中理工科系學生多選擇繼續攻讀研究所，研究所現今入學管道基本可分為兩類，推薦甄試與一般生考試，網路上雖有資料整理，但多零散，且須連結多個分頁，對於使用者不慎方便。因此希望能實作一個整合平台，提供更為直覺的操作方式及視覺化資料，進行多校、多科系、多年度等比較，幫助學生更容易了解入學趨勢，建立良好策略。

以下為系統架構圖：



圖一

二、測試結果：



圖二：系統執行畫面

以 Error-state Kalman Filter 整合傳感器提 高車輛導航精度

Enhancing Vehicle Navigation Accuracy via Error-state Kalman Filter

指導教授：蔡佩璇

專題成員：張庭瑋

開發工具：Python、C

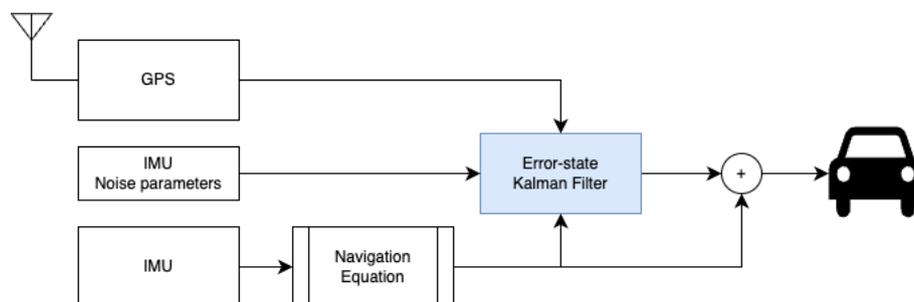
測試環境：STM32、Linux

一、簡介：

此專題旨在使用 MEMS 等級之 IMU（慣性測量單元）與 GPS 實作高動態載具（如：賽車）之定位，提供其他演算法高採樣率與高精度之導航解進行控制，有鑑於傳統 GPS 特性，雖精度高，但每秒至多僅能提供 10 筆定位資料；而 IMU 有著較高速的資料採樣率，但受其特性所影響，訊號中有 bias 及 drift 造成之雜訊影響導航精度。只須測量出 IMU 之雜訊，即可以整合兩種感測器之優點相互彌補其缺點，提供高採樣率與精度之訊息。

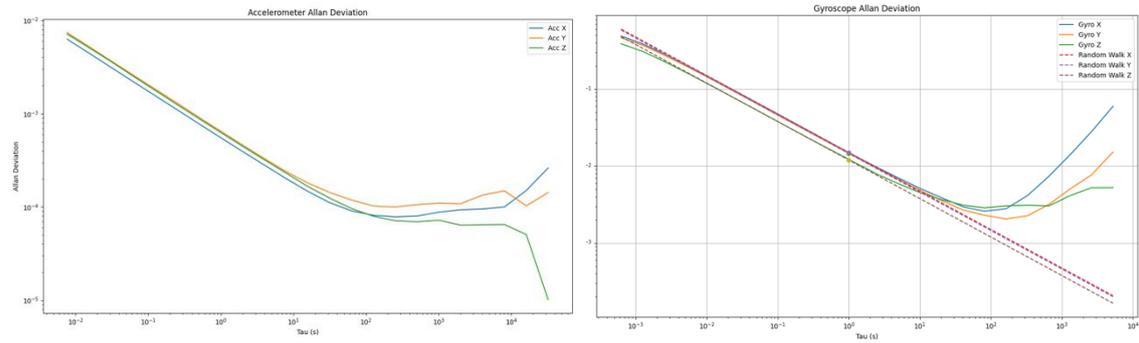
這邊使用 Movella 公司所推出的 MTi-600 系列 IMU 來進行雜訊的測量，首先蒐集其靜態讀值，並使用 Allan variance 方法分析其參數。

測量完 IMU 之 Allan Variance 後，將其參數放入 Kalman Filter 中的 Q 及 R 矩陣，使用 Error State Kalman Filter 結合兩種輸出頻率之感測器（GPS、IMU）並輸出位置、速度、加速度及姿態給車輛。

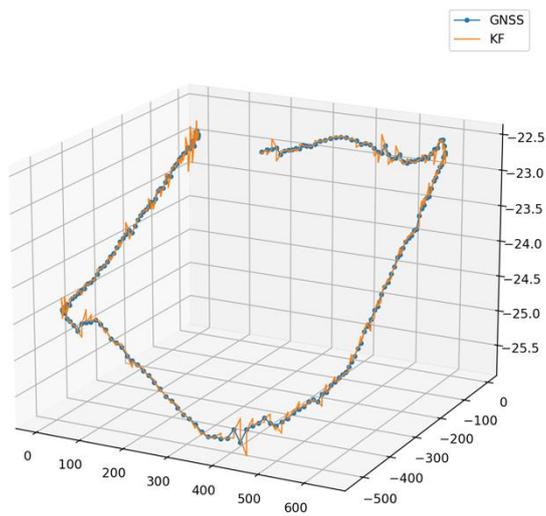


系統架構圖

二、測試結果：



上二圖為蒐集 MTi-600 IMU 中陀螺儀與加速度計靜態讀值 20 小時後，使用 Allan Variance 分析並作圖，可得知 Angle Random Walk、Bias Instability。



左圖為使用 ESKF 整合兩種感測器讀值之輸出結果，可觀察到藍色資料點為 GPS，輸出較慢但較為平穩，橘色則是 ESKF 輸出結果。

透過 YOLOv8 實作人物追蹤 – 以檢測學生 活動中心刷卡尾隨為例

Implementing Person Tracking using YOLOv8 - A Case Study on Detecting Student Activity Center

Unauthorized Entry

指導教授：蔡佩璇

專題成員：陳培恩

開發工具：Python3.9, YOLOv8

測試環境：Debian Ubuntu 18.04.5 LTS, Android 11

一、簡介

成大學生活動中心（以下簡稱為活動中心）時常會出現未刷卡，直接尾隨其他刷卡同學進入活動中心的情況。為了避免此狀況造成校園安全問題，本次專題主要以人物追蹤為主軸，於活動中心出入口加裝監視器，並以影像處理的方式檢視，並比對成大門禁系統的刷卡紀錄，以確認是否有人士擅闖活動中心，以此維護校園安全。當出現可疑人士擅闖的情況時，系統連接至 Line Notify，會立即通報情況，並提供影像截圖，以供後續校方處理，每一次偵測的整體流程圖如下圖 1。

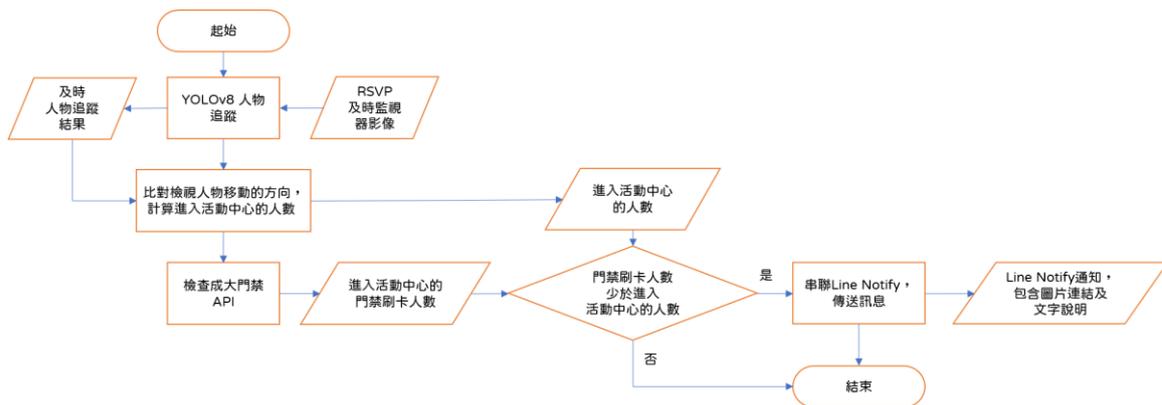


圖 1 單幀處理整體架構流程圖

本專題透過 rsvp 讀取監視器即時畫面影像，並使用 YOLOv8 進行人物追蹤。以檢測人物座標的方式，去確認人物移動的方向，若人物是從門外向內移動，如圖 2 所示，由畫面中的右方往左方移動，並且跨越圖中斜率為負的直線，則我們將此人物視為進入活動中心。此時，會進一步檢測刷卡紀錄是否與進入活動中心的人數相同，若刷卡進入的人數少於實際進入的

人數，則會將畫面截圖，並傳送至 Line Notify，通知校方。

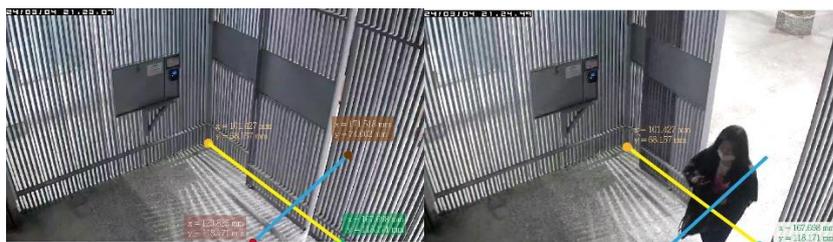


圖 2 監視器影像畫面檢測示意圖

二、 測試結果

測試結果以下面兩張圖簡單呈現。

圖 3 為人物辨識的結果，左側為框定出來的人物以及其編號，右側為人物在此圖中的座標、長寬及編號。同一人會被以同一個編號命名。



圖 3 人物追蹤辨識結果

本專題的程式串聯 Line Notify API，當辨識到人物時，Line Notify 會傳送訊息和照片連結通知，圖 4 為通知的手機畫面截圖。



圖 4 Line Notify 通知截圖畫面

實作限制執行環境的 Python 與繞過手法

指導教授：蔡佩璇

專題成員：楊竣鴻

開發工具：Python 3.9

測試環境：Debian GNU/Linux 11 (bullseye)

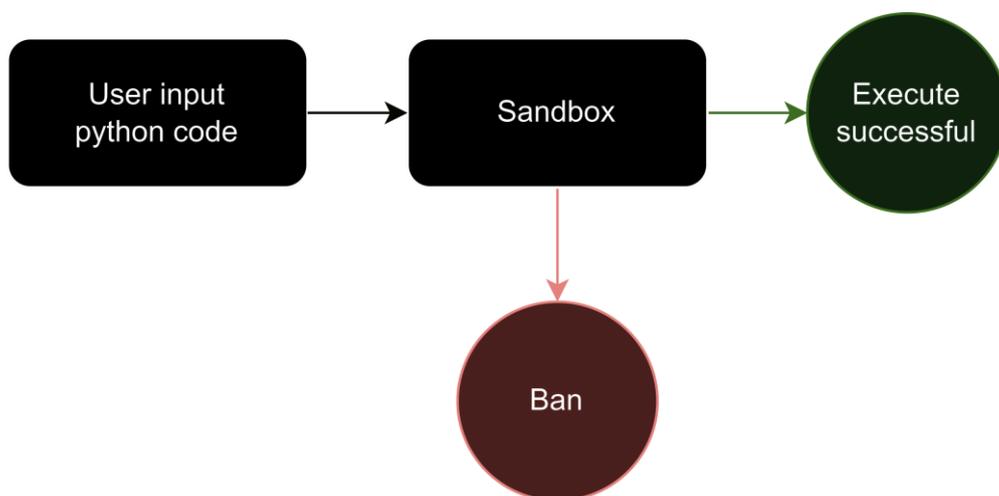
一、簡介：

這個專題探討如何在限制環境中滿足應用程式需要使用 Python 任意運行指令的需求，並提供容易遭到繞過的實作方案。儘管現代應用程式在安全性和資源管理方面越來越嚴格，但某些情境下仍然需要運行不受限的 Python 指令。本專題旨在研究和設計一個能夠在這樣的限制環境中運行的框架，同時探索如何有效地繞過這些限制，以便實現所需功能。

接下來，本專題將詳細討論限制與繞過這些限制的方法，包括：

- **定義沙盒：**透過限制讓使用者有限制地與 Python 互動並執行指令。
- **實作黑名單：**研究如何選擇和定義需要禁止的指令和模組，通過動態檢查和阻止黑名單中的指令執行。
- **透過底層設計限制：**探討如何控制 Python 執行環境的底層結構，從而實現指令執行的限制，並介紹具體的實作步驟和技術細節。

最終，本專題旨在為開發者提供一個指南，幫助他們在限制環境中實現所需的 Python 指令執行，同時理解和掌握各種繞過方法，以便在實際應用中靈活應對挑戰^[圖-1]。



圖一：沙盒運作架構圖

二、測試結果：

這種沙盒^[圖二]將使用者輸入與黑名單進行比對，若輸入未包含黑名單字串，則直接執行。即，只要輸入不含黑名單字串便可^[圖三]。

```
inp = input()
# 此 blacklist 可任意擴充
blacklist = ("import", "os", "system")
if not sum(bad in inp for bad in blacklist):
    eval(inp)
else:
    print("filtered!")
```

圖二：字串黑名單沙盒

```
__builtins__.__dict__[ '__imp'+ 'ort__ ' ] ('o'+ 's').__dict__[ 'sy'+ 'stem' ] ('whoami')
```

圖三：字串黑名單沙盒繞過案例

這種沙盒^[圖四]通過限制 Python 內建函數來達到安全控制。它首先列出所有內建函數的關鍵字，然後移除除了 print、input 和 eval 之外的所有函數，從而防止執行潛在危險的操作。

```
builtins_key = list(__builtins__.__dict__.keys())
for key in builtins_key:
    if key in ("print", "input", "eval"):
        continue
    del __builtins__.__dict__[key]

print(eval(input()))
```

圖四：將物件移除的沙盒

類似的沙盒方法^[圖五]包括在 exec 或 eval 創建執行環境時，只提供特定的內建函數，這與上述方法類似。

```
exec(input(), {"__builtins__": {"print": print, "dir": dir}})
```

圖五：將物件隔離的沙盒

在 Python 中，所有類別都繼承自 object。可以通過 ().__class__.__base__ 獲取 object 類，然後進一步取得它的 subclasses。找到 <class 'os._wrap_close'> 後，通過它的 __init__ 方法取得 init 函式，接著取得當前環境的 globals，從中取出 os 模組，並使用 os.system 執行任意指令^[圖六]。

```
[].__class__.__base__.__subclasses__()[133].__init__.__globals__[ 'os' ].system('whoami')
```

圖六：缺失關鍵物件沙盒的繞過案例

童話創意師：圖片轉故事語音平台

FairyTale Innovator: Image to Story Audio Platform

指導教授：鄭憲宗

專題成員：何韋德、呂振呈、高柏祐、林業誠

開發工具：LLaVA, Llama-3, streamlit

測試環境：Ubuntu 20.04.6 LTS

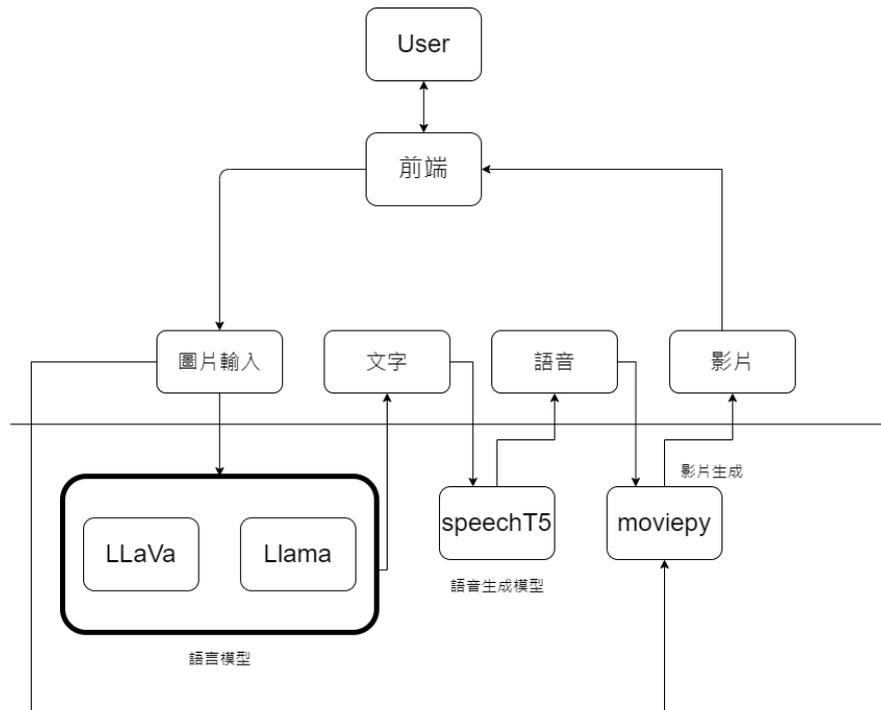
一、簡介：

童話創意師：圖片轉故事語音平台是一款專為童話故事愛好者設計的創新數字化服務平台。該平台能將用戶上傳的任何圖片轉化為迷人的故事，並以影音形式呈現，迎合那些偏好視聽介面而非文字的用戶。用戶只需上傳圖片，平台便會分析圖中的人物、背景與物件，並運用先進的人工智慧技術創造出一個獨特的童話故事。

平台的一大特色是其語音合成技術，這技術能將故事講述得更加生動有趣。這一功能不僅適合兒童，成人亦可在其中發現樂趣。此外，童話創意師支援多語言，使不同文化背景的用戶都能享受這種獨特的互動體驗。

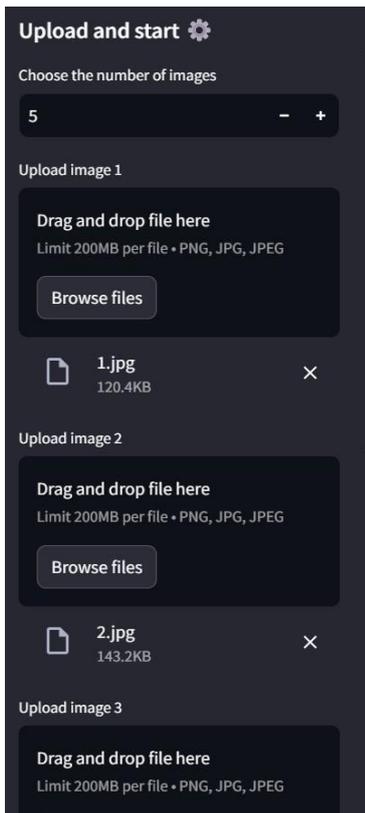
此平台不僅能夠激發創意思維，亦是一個極佳的教育工具，有助於提升語言學習與聽力技巧。無論是在家庭、學校或其他娛樂場合，童話創意師均是一個出色的工具，它結合了圖像與語言，能夠激發用戶的創造力與想象力。

以下為系統架構圖：

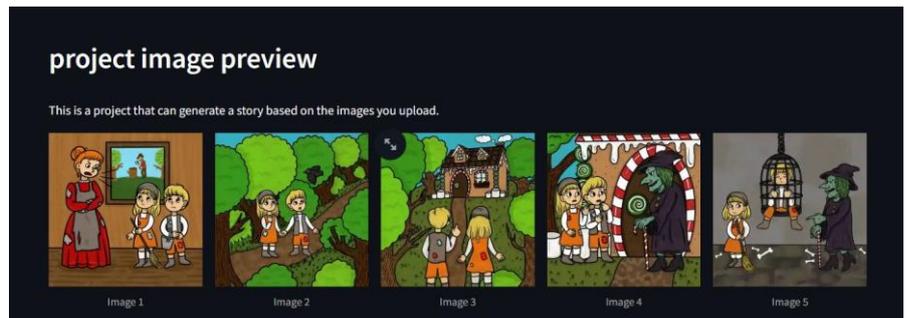


二、測試結果：

1. 輸入參數：使用者可在網頁左側的欄位中指定所需上傳的圖片數量，範圍為 3 至 10 張，如圖(一)所示。
2. 圖片上傳與預覽：上傳後的圖片將提供預覽功能，確保所選圖片符合使用者的期望，如圖(二)所示。

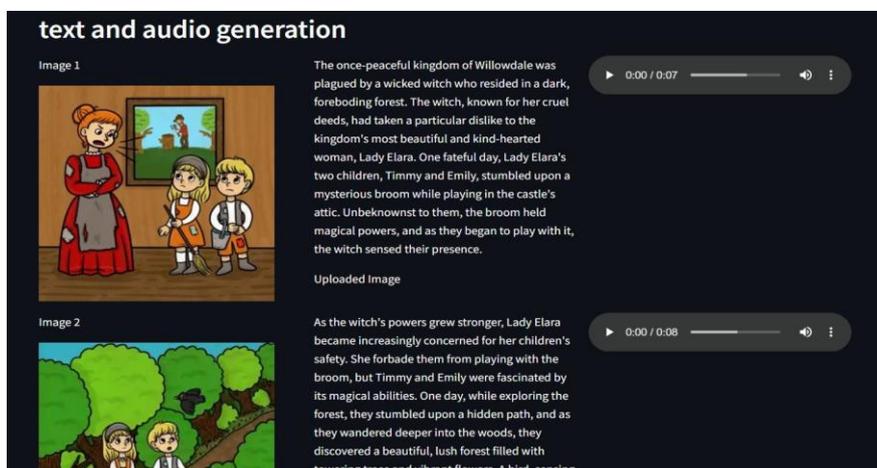


圖(一)



圖(二)

3. 生成過程：用戶點擊生成按鈕後，系統將為每張圖片創造獨特的故事段落及相應的語音敘述。這項功能使故事的享受變得更多元，滿足不同用戶的需求。



4. 最終展示：系統最終將生成一部結合了語音和字幕的影片，特別為那些偏好視聽內容而非文字閱讀的用戶設計。



作伙：基於語意分析協助老人揪團

GetTogether: An app helping the elders hold activities with semantic analysis

指導教授：盧文祥

專題成員：謝秉宸

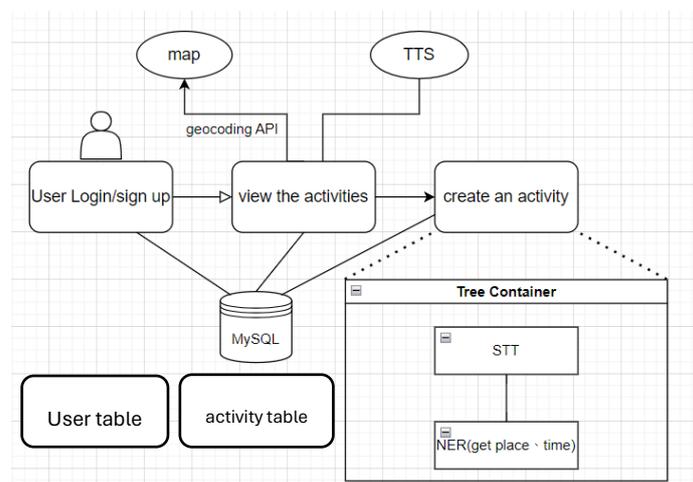
開發工具：Flutter、MySQL、
Apache、php

測試環境：Android, Windows 11

一、簡介

根據內政部統計，2018年台灣的高齡人口占比超過14%，已步入高齡社會，在2025，占比更估計會超過20%。在這個越趨高齡化的社會，未來一定會有越來越多高齡人口，其中可能有部分不具備識字能力，而較難在社交軟體上獲得與他人交流的機會。因此我想藉由語音合成、語音辨識，與命名實體識別(NER)打造一個不具識字能力也能夠使用的高齡人口揪團軟體。

於開團頁面中，使用者可以選擇使用語音輸入，再對使用者說的話進行命名實體識別，將時間、地點、活動等資訊抽取出來以協助揪團。在活動資訊頁面中，可以查看活動當前狀態，亦可查看活動地點附近的地圖。以下為系統架構圖：



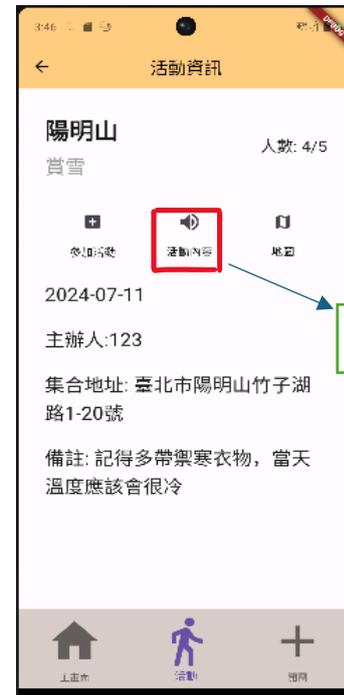
二、測試結果



1. 登入畫面



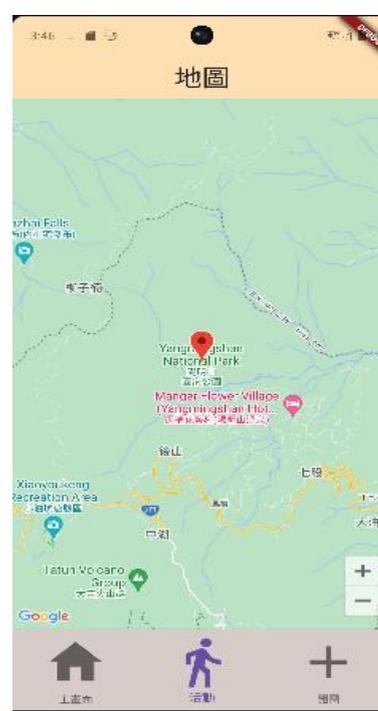
2. 進入 app 主畫面



3. 活動資訊頁面



4. 語音辨識+NER 開團畫面



5. 透過 geocoding 取得地點

地址並取得周邊地圖

多語移工交流 APP

Multilingual Chat App for foreign workers

指導教授：盧文祥

專題成員：聶怡安

開發工具：Android Studio

測試環境：Android 13

一、簡介

動機

許多來台工作的外籍移工即使已經在母國接受過中文課程，有些時候仍然無法應付許多日常對話。在移工與照護者有語言隔閡的情況下，溝通有時只能依靠比手畫腳，或是 Google 翻譯輸入關鍵字進行溝通，然而這種方式非常容易產生誤會。無論是在日常照護上，抑或是在與家屬溝通的過程中，只要有語言問題存在，一旦需要進行複雜任務，雙方的溝通過程便容易發生差錯。

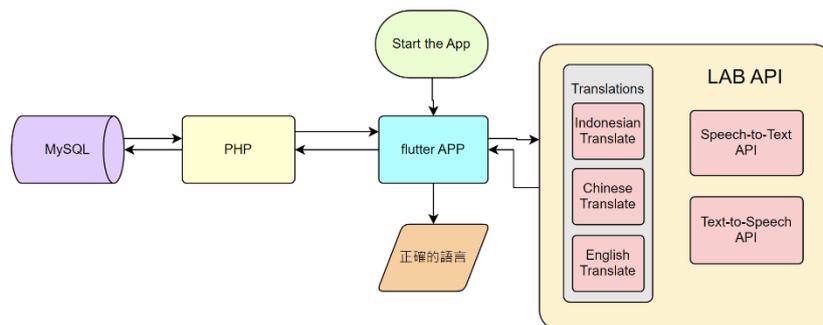
調查

- 自內政部統計以來，截至2023年的移工數量高達65萬3498人，其中以印尼籍24萬4002人最多，越南籍20萬5994人次之，菲律賓籍13萬8768人居第三。
- 根據勞動部最新「移工管理及運用調查」，家庭面雇主管理及運用移工有困擾者占33.2%，主要為「語言溝通困難」，佔 56.1%。

功能

基於 KASIH APP 所進行的延伸開發，目標是提升外籍移工與家屬之間的溝通品質。透過即時翻譯與語音辨識系統，使用者可以在聊天室中用自己的語言輸入文字或是語音，對方會收到自己所熟悉的文字訊息或是經由語音合成所獲得的新音檔，解決口頭傳達上可能會發生的誤會，實現及時交流與翻譯的功效。

系統架構圖



二、測試結果

此為翻譯系統的實作測試，左邊為中文使用者，右邊為印尼語使用者



語音辨識的測試（現場實際測試的效果比較直觀）



由於合成辨識的結果為語音，因此不便於文件中呈現，可以由現場實際測試

復古旅行

Vintage Adventure

指導教授 盧文祥

專題成員 林紹恆

開發工具 bun, React JS, elysia JS

測試環境 MacOS 14.5 (23F79)

一、簡介：

Step into the Past with Vintage Adventure

Imagine reliving and sharing your most treasured memories with vivid clarity, as if you were reliving them for the first time. Vintage Adventure is an innovative app that takes you on a journey through time, empowering you to capture, preserve, and share your personal history like never before.

Technical Backbone

Our app is built on a robust technical foundation, with a backend server that handles complex image and video processing and other resource-intensive tasks unsuitable for device-side execution. This allows us to leverage the power of cloud computing to deliver fast and efficient processing of user content. Our front end is built using ReactJS, with the Ionic library enabling seamless multi-platform adaptation, ensuring a consistent and engaging user experience across various devices and operating systems.

Revolutionizing Storytelling

By combining the power of street view technology, oral storytelling, and AI-generated imagery, Vintage Adventure brings the past to life in a unique and captivating way. Whether you're a senior looking to reminisce about fond memories or a family member eager to learn more about your heritage, our app provides a platform to record, relive, and share your stories with future generations.

Transforming Memories into Shareable Videos

With Vintage Adventure, you'll embark on a fascinating adventure that helps you remember the past and makes it more accessible and engaging for others. By transforming your memories into stunning, shareable videos, our app bridges the gap between generations, fostering a deeper understanding and appreciation for the

experiences that have shaped our lives.

二、測試結果：(標題字形16)

Our app has undergone rigorous testing with impressive results:

Image processing and video generation: 100% success rate, with accurate and efficient user-uploaded images and video processing.

User experience: Overall positive reaction from the subjects, with the most common remarks being:

- "Innovative"
- "Simple yet powerful UI"
- "An interesting idea."

Performance: The app demonstrated fast loading times, smooth navigation, and seamless video playback thanks to several key technical improvements:

- We've optimized ReactJS rendering using Million JS, resulting in a significant boost to frontend speed.
- We've replaced ExpressJS with Elysia JS, reducing backend response times by around 60%.
- We've upgraded to the Nova-2 by Deepgram voice-to-text model, which has accelerated transcription times by 80% compared to Whisper Large by OpenAI.

Feedback: Users provided valuable feedback, highlighting areas for improvement, such as:

- Enhancing the app's guidance and tutorials for first-time users.
- Expanding the range of editing options for video customization.
- Integrating social media sharing features for easier content distribution.

(This document was crafted with the assistance of artificial intelligence.)

應用於 MiinStore 平台之資料湖實作

MiinStore DataLake Implementation

指導教授：謝孫源教授

專題成員：陳廷笙、王昱翔

開發工具：Python、JavaScript

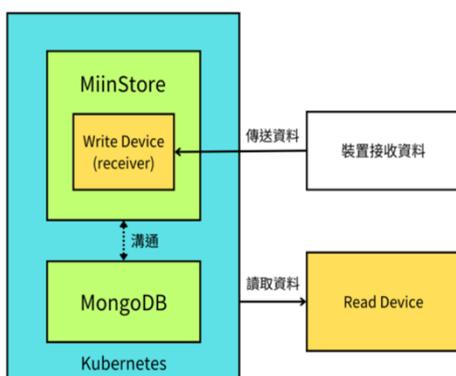
測試環境：Windows、Mac OS、Ubuntu 22.04

一、簡介：

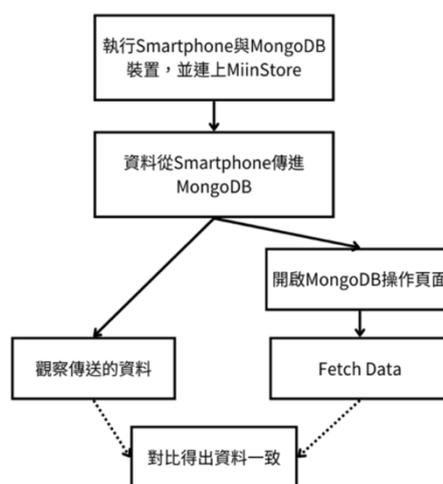
本作品基於 MiinStore IoT 平台，加上先進的資料湖(Data Lake)架構，以利高度靈活的數據存儲與分析。資料湖的特點為：「在無需提前建立資料結構的情況下，即可儲存**結構化與非結構化**的資料。」而且，由於資料湖不需提前定義資料的類型，將可大幅增加實務上使用的**彈性與便利性**，並根據需求在不同的情況下，將收集到的資料進行最適合的分析。

再來介紹本次專題的實驗方法，先將由一台**行動裝置(Smartphone)產生資料**，透過我們在 MiinStore 內部設計的一個名為 **Receiver 的裝置(Device)**來接收後，把收集到的資料傳入我們用 MongoDB 實作的資料湖中，並使用在 MiinStore 中設計的 **MongoDB 讀取裝置**，從資料湖中讀取資料，最終下載為 JSON 或 CSV 格式。

作品未來將可被應用於各種不同場域，選擇使用建立於本地端的資料庫，而非外部現成的資料庫，可安全存放個人資訊、檢驗數據、醫學影像等高度隱私的資料，將可**提升相關資料的安全性**，與有效的防止資料外洩的情況發生；另外，使用資料湖與一般的資料庫相比，以醫學上為例，將可因應醫學上經常出現新的醫療技術，新型的技术經常會產生不同的資料類型，此時使用資料湖，就可利於我們將**各種不同型態的資料儲存在同一個地方**，不需額外花費成本與時間再建立另外的資料庫。



圖一、系統架構圖



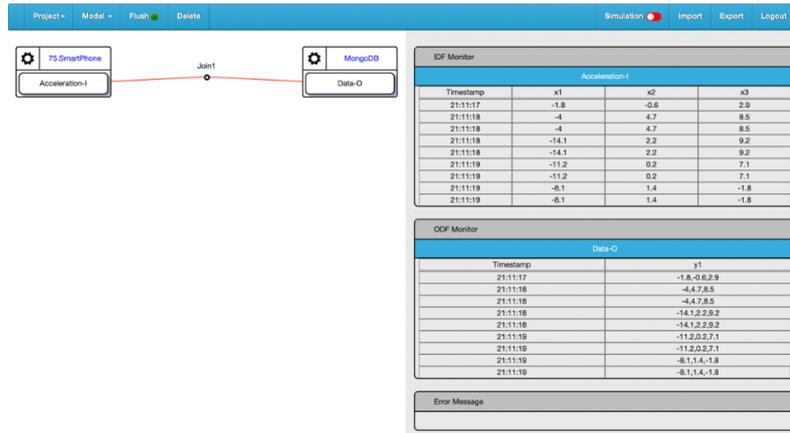
圖二、結構化資料處理流程圖



圖三、非結構化資料處理流程圖

二、測試結果：

本次實驗會在 MiinStore 網頁平台上進行。將 MongoDB 變成 MiinStore 平台上的 Device，可供使用者直接使用，並將 input 與 output 給 join 起來，傳送給 MongoDB。右側的 x1,x2,x3代表手機所傳出的三軸加速度，如圖四(下)所示。



圖四、MiinStore 操作畫面

在 MongoDB 的操作畫面內，可以看到圖四的 x1~x3會被存為 Data，Timestamp 會被存為 Date 並存進 MongoDB，並且和圖四的資料一致，如圖五(下)所示。

No.	_id	Data	Date
1	6649fa77cde45ee72bbb2c69	-8.1,1.4,-1.8	2024/5/19 下午9:11:19
2	6649fa77cde45ee72bbb2c68	-8.1,1.4,-1.8	2024/5/19 下午9:11:19
3	6649fa77cde45ee72bbb2c67	-11.2,0.2,7.1	2024/5/19 下午9:11:19
4	6649fa77cde45ee72bbb2c66	-11.2,0.2,7.1	2024/5/19 下午9:11:19
5	6649fa76cde45ee72bbb2c65	-14.1,2.2,9.2	2024/5/19 下午9:11:18
6	6649fa76cde45ee72bbb2c64	-14.1,2.2,9.2	2024/5/19 下午9:11:18
7	6649fa76cde45ee72bbb2c63	-4,4.7,8.5	2024/5/19 下午9:11:18
8	6649fa76cde45ee72bbb2c62	-4,4.7,8.5	2024/5/19 下午9:11:18
9	6649fa75cde45ee72bbb2c61	-1.8,-0.6,2.9	2024/5/19 下午9:11:17

圖五、MiinStore 平台內，MongoDB 的操作畫面

本次實驗 Data Lake 的好處是除了結構化資料外，非結構化資料如圖片也可以處理。以下是照片的處理，讀取 Data Lake 中的照片檔案資料由 png 檔轉成 byte string 格式後，再以 base 64編碼，如圖三(上)及圖六(下)所示。

No.	_id	Data
1	66489fed202c6f91773edc5d	{"data": "iVBORw0KGgoAAAANSUhEUgAAC3wAA"}
2	66489fec202c6f91773edc5c	{"data": "iVBORw0KGgoAAAANSUhEUgAAC3wAA"}
3	66489feb202c6f91773edc5b	{"data": "iVBORw0KGgoAAAANSUhEUgAAC3wAA"}
4	66489fea202c6f91773edc5a	{"data": "iVBORw0KGgoAAAANSUhEUgAAC3wAA"}

圖六、MongoDB 操作非結構化資料

探討 PIM 技術如何影響線性回歸的效能

On Exploring How PIM Technique Affects the Performance of Linear Regression

指導教授：何建忠

專題成員：劉彥誠

開發工具：C、Python、

UPMEM DPU toolchain

測試環境：Linux Ubuntu 20.04

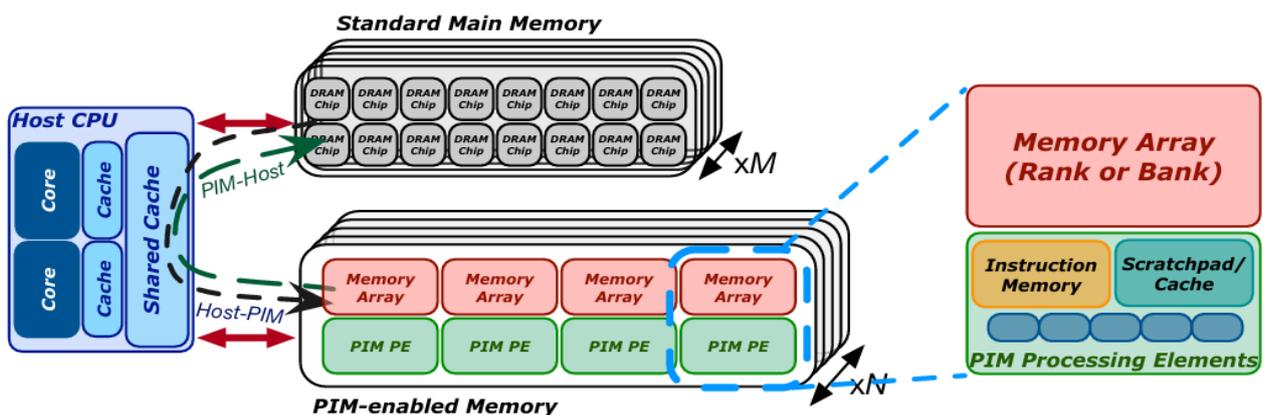
一、簡介：

機器學習被廣泛應用於各種領域，此類型的演算法在訓練過程中需要存取大量的訓練資料，通過不斷迭代更新參數來提升模型的表現。然而，在當前以處理器為中心的系統（如 CPU、GPU）中，由於記憶體有延遲及頻寬的限制，在記憶體和處理器之間進行大量數據移動，將導致大量的執行周期和能量消耗，可能成為效能的瓶頸。

PIM（Processing-in-Memory）是一種異於 Von Neumann 架構的方法，旨在通過將運算的能力與記憶體晶片整合，以減少數據移動帶來的延遲。為了探討 PIM 技術對於上述問題的影響，本專題使用 UPMEM 的通用 PIM 架構，並挑選 Linear Regression 為實驗對象，因為其在進行梯度下降時，需要存取所有的訓練資料，而且也具有低運算強度和低 temporal locality 的特性，屬於 memory-bound workload。在訓練的過程中，會重複執行以下兩個步驟直到收斂：

1. 將所需的資料傳到 DPU 上計算梯度、
2. 將計算結果傳回 CPU 上更新權重。

以下為系統架構圖：



二、測試結果：

目前現實中的 PIM 系統存在固有的缺點，即由於在記憶體附近或內部建構處理元件的難度和成本，其硬體比傳統處理器（如 CPU、GPU）受到更多限制。因此，通用 PIM 架構支援的指令集相當有限，而且難以執行複雜的運算。

為了評估 PIM 對效能的影響，實驗將從以下幾個面向進行：

1. 在 DPU 上使用不同資料型別 / 對乘法進行優化，共分成四種版本：
 - (1) 32-bit floating-point (LIN-FP32)
 - (2) 32-bit fixed-point (LIN-INT32)
 - (3) fixed-point with hybrid precision (LIN-HYB)
 - (4) fixed-point with hybrid precision and built-in functions (LIN-BUI)
2. 對於四種版本，在一個 DPU 上使用不同數量的 tasklet (PIM thread) 執行
3. 對於四種版本，進行 weak scaling 和 strong scaling 實驗 (使用 synthetic dataset)

以下是 LIN-FP32 使用一個 DPU 和一個 tasklet 的執行結果：

```
1 Allocated 1 DPU(s)
2
3 iter_time = 500, learning_rate = 0.0001, m = 2048, n = 16
4 Predefined weight: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
5 Successfully generate input data.
6
7 training at host, float...
8
9 Trained weight at host:  1.61,  2.56,  3.62,  4.59,  5.62,  6.54,  7.53,  8.61,
10                        9.51, 10.66, 11.56, 12.61, 13.50, 14.56, 15.58, 16.60
11
12 Load input data to DPUs
13 Run program on DPU(s)...
14 DPU iter 0...
15 DPU iter 100...
16 DPU iter 200...
17 DPU iter 300...
18 DPU iter 400...
19
20 Trained weight at DPU:  1.61,  2.56,  3.62,  4.59,  5.62,  6.54,  7.53,  8.61,
21                        9.51, 10.66, 11.56, 12.61, 13.50, 14.56, 15.58, 16.60
22
23 MAE on DPUs = 39.3054, avg Y = 3569.9282, error rate = 1.10%
24
25 init CPU-DPU Time (ms): 0.064000      syn CPU-DPU Time (ms): 2.191000
26 DPU kernel Time (ms): 800581.933000   DPU-CPU Time (ms): 4.809000
27 CPU reduction Time (ms): 0.581000
28
29 Outputs are equal.
```

在 SSD 上重疊讀取和寫入操作以提升效能：觀察和實作

Enhancing SSD Performance via Overlapping Read and Write Operations : Observation and Implementation

指導教授：何建忠

專題成員：陳佳豪、蕭名惟

開發工具：Visual Studio 2017、C++

測試環境：Windows 11

一、簡介：

SSD 在執行寫入指令時，實務上會使用 ISPP(Incremental Step Pulse Programming)針對快閃記憶體單元 program 成欲儲存資料的臨界電壓值，它分成多個階段漸進式地推進電壓，每次提升的電壓是一個定值。另外在每一個階段後會接著執行 verify 的動作以檢查是否已經達到 ISPP 的預期臨界電壓。此外，SSD 有其電力限制，因此不管是在做寫入或是讀取的動作時，都必須為其分配一個預設使用的電力空間，當分配出去的電力達到可使用的總電力上限時，剩下的寫入及讀取指令就會被延遲。

然而，ISPP 並不會無時無刻使用到所有預設的電力數值，此時會有空閒的電力處於未使用的狀態，將這些空閒的電力有效運用，能夠預期 SSD 有更好的平行效能。由於 ISPP 執行 verify 與 SSD 讀取指令所需的時間相當，且兩者所需電壓也遠低於整個 ISPP 的預設電力數值，若利用執行 verify 時被閒置的電力去做其餘的讀取指令，則可以有效提升整個 SSD 的執行效能。

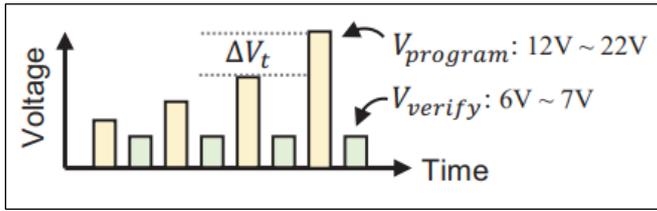


Figure.1 ISPP

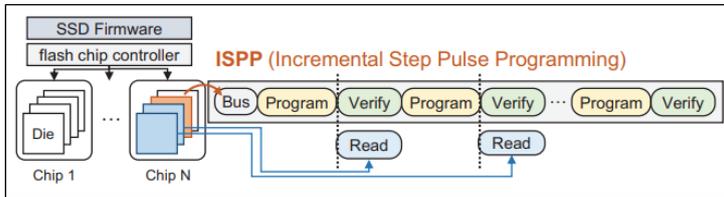


Figure.2 Overview

二、 測試結果：

Precedence	write - read - erase
Request_Count	6999
Read_Request_Count	4381
Write_Request_Count	2618
Bytes_Transferred	59718656
Bytes_Transferred_Read	36315136
Bytes_Transferred_Write	23403520
Bandwidth	59718656
Bandwidth_Read	36315136
Bandwidth_Write	23403520
Device_Response_Time	178586
Min_Device_Response_Time	5
Max_Device_Response_Time	400031
Average_Read_Transaction_Turnaround_Time	23727
Average_Read_Transaction_Execution_Time	74
Average_Read_Transaction_Transfer_Time	13
Average_Read_Transaction_Waiting_Time	23639
Average_Write_Transaction_Turnaround_Time	61170
Average_Write_Transaction_Execution_Time	712
Average_Write_Transaction_Transfer_Time	13
Average_Write_Transaction_Waiting_Time	60444

Precedence	read - write - erase
Request_Count	6999
Read_Request_Count	4381
Write_Request_Count	2618
Bytes_Transferred	59718656
Bytes_Transferred_Read	36315136
Bytes_Transferred_Write	23403520
Bandwidth	59718656
Bandwidth_Read	36315136
Bandwidth_Write	23403520
Device_Response_Time	155368
Min_Device_Response_Time	5
Max_Device_Response_Time	406872
Average_Read_Transaction_Turnaround_Time	2214
Average_Read_Transaction_Execution_Time	74
Average_Read_Transaction_Transfer_Time	13
Average_Read_Transaction_Waiting_Time	2126
Average_Write_Transaction_Turnaround_Time	109417
Average_Write_Transaction_Execution_Time	709
Average_Write_Transaction_Transfer_Time	13
Average_Write_Transaction_Waiting_Time	108694

iSAFE: 多權重連網式間歇性運算系統的更新轉送演算法

iSAFE: Enabling Evenness of Data Freshness in Multi-Weight Networked Intermittent System

指導教授：涂嘉恒博士

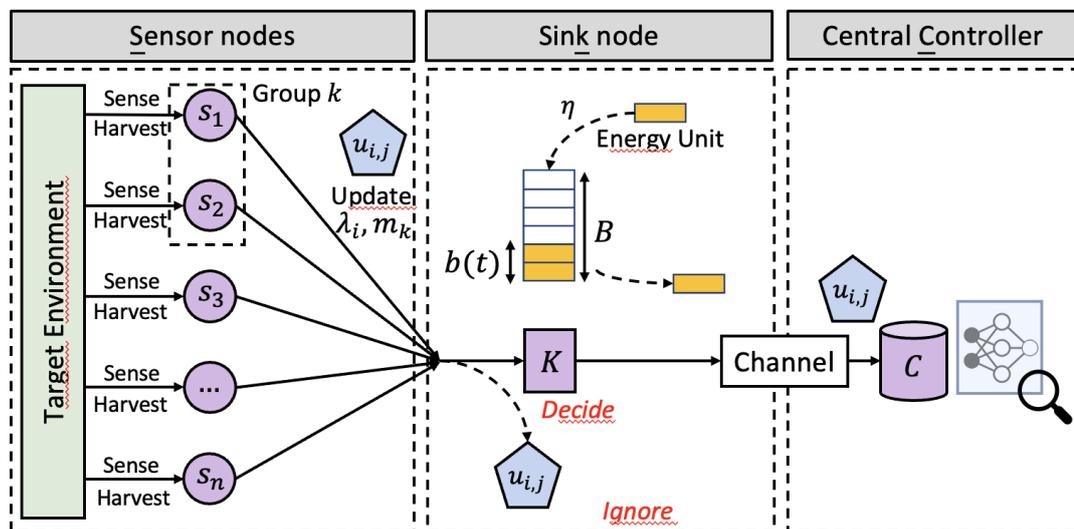
專題成員：陳禹丞

開發工具：rustc

測試環境：Ubuntu 22.04.5

一、簡介：

能量採集（Energy Harvesting）技術已成為聯網式感測器裝置的一種永續性且經濟有效的設計。這類設計適用於需要大範圍部署以即時追蹤狀態並對異常情況作出反應的應用領域。例如野火監測和火山監測。由於這些裝置由環境能量（例如太陽能、RF 和風能）供電，它們在有能量供應時間歇性地運作，被稱為聯網式間歇性運算系統（NIS）。



圖一：NIS 的系統架構

NIS 依賴匯聚節點（Sink Node）將所有感測器收集的即時資料中繼到遠端站點。由於資料傳輸的耗電量很高，在電量不足的前提下，如何把電力用在刀口上十分重要，防止因電量不足而導致的資料中繼中斷。然而，從應用角度來看，應盡可能頻繁地將狀態資料更新到中央控制器，以準確評估即時環境狀態，從而對潛在的緊急事件作出迅速反應。因此，針對這種系統，需要一種資料更新機制，藉以在即時狀態更新和適當的能量利用之間取得平衡。

前人的研究使中央控制器能夠從 NIS 中的感測器獲取最新資料，在現場部署許多感測器和一個匯聚節點，以將收集的資料轉運到中央控制器（用於評估環境狀態並在必要時作出反應）。特別是，陝西科技大學的開拓性研究引入了如何在 NIS 的匯聚節點轉送資料更新的新鮮度問題（每個匯聚節點連接一個數據感測器），並提出了匯聚節點上的資料轉送演算法（包括一個線上算法和一個離線啟發式算法），這些節點具有有限的能量。隨後的研究進一步支持 NIS 的資料轉送，其中每個匯聚節點連接多個感測器，感測器發送的狀態更新在每個匯聚節點進行緩衝。最近的一項工作提出了 NIS 中匯聚節點的離線最優算法，並通過提出無緩衝更新轉送算法進一步減少了匯聚節點對資料緩衝的需求。

然而，上述方法預設 NIS 中連接到匯聚節點的所有感測器具有相同的重要性。這意味著所有感測器的狀態更新具有相同的時效性要求，導致匯聚節點轉運大量類似的資料。這一假設限制了 NIS 的彈性和運用範圍，無法滿足現代監測應用的需求。在這些應用中，具有不同重要性資料類型需要不同的更新頻率。這種多樣性更好地反映了現實世界的環境狀態，考慮了例如不同地形、季節、天氣和目標等因素。例如，在森林火災監測的早期偵測中，溫度和濕度數據比一氧化碳和氣壓數據更具時間敏感性。因此，相比於後者，前者需要更頻繁的更新。

儘管已有多項研究在具有不同優先級的系統上研究數據新鮮度優化，但大多數僅考慮兩個優先級（高與低）。特別是，它們將最高優先級資料視為系統中的例外需要匯聚節點通過搶占當前操作或在當前操作完成後立即服務。質言之，它們不允許系統中存在不同重要性級別的資料，因此可能會從根本上影響監測準確性。此外，這些資料新鮮度研究並未考量 NIS 能量不足的窘境。

二、測試結果：

iSAFE 是專為 MW-NIS 中的匯聚節點設計，以優化資料新鮮度的均勻性，從而更新環境監測應用中的實時資料。我們的結果顯示，與理想情況相比，iSAFE 可以達到 94.47% 的均勻性，同時在資料新鮮度方面保持其有效性，優於先前的研究。此外，iSAFE 在不同資料來源的優先級配置下，即使在不穩定的環境中變化，也能提供極其穩定的均勻性。這是通過推導出的最優間隔 X_{opt} 來運行轉送決策以及兩階段轉送機制所實現。iSAFE 顯著提高了監測資料的新鮮度和均勻性，同時具備高能效和可擴展性，適用於現代監測應用。

汽車開放式系統架構通訊模組

接收端軟體的設計與實作

Design and Implementation of AUTOSAR COM

Module Receiver Side Software

指導教授：張大緯

專題成員：陳廷睿、方騏為、陳家禾

開發工具：C、CS+ for CC、RH850/F1H

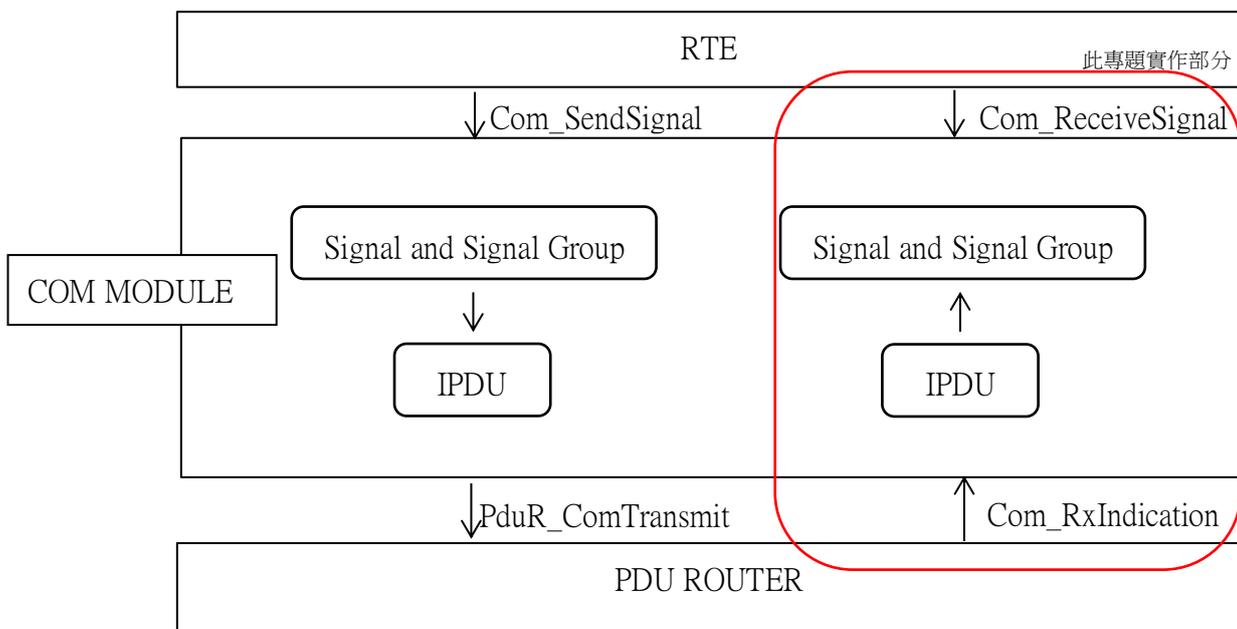
測試環境：Windows11、CS+ for CC

一、簡介：

AUTOSAR (Automotive Open System Architecture) 主要目的是提高軟體的可重複使用性。AUTOSAR Classic Platform 分為三層：應用層、運行環境層和基礎軟體層。它定義了各層的行為和規範了各層間的互相溝通。AUTOSAR 在設計、維護和升級方面為開發者提供了便利。

此專題建立在基於 AUTOSAR 規範的 OS 上實作 COM(Communication) 模組的接收端，目的是使其遵照 AUTOSAR 協議，內容包含了接收端軟體的設計以及相關程式碼的實作。

以下為系統架構圖：



Sender Side 由 RTE(Run Time Environment) call `Com_SendSignal`，COM module 將 Signal 和 Signal Group 打包成 IPDU(Interaction Layer Protocol Data Unit)並 call `PduR_ComTransmit`，經由底層傳送。

Receiver Side 經由 PDU router call Com_RxIndication，收到 IPDU 並進行拆包，並等待 RTE call Com_ReceiveSignal。

二、測試結果：

1. 程式碼生成測試：

a. 步驟 main_test.c Com_misc.c Com_PbCfg.c Com_Sche.c Com.c

(1) 撰寫 main.c / Com.c / Com_misc.c / Com_Sche.c 檔

(2) 設定好 Com_PbCfg.c 中的資料結構。

(3) 將所有.c 檔 compile 並執行

b. 執行結果

程式會將收到的 IPDU 拆成 signal 和 group signal，送至 RTE 指定位置

```
{
    .ComBitPosition=1,
    .ComBitSize=4,
    .ComSignalEndianness=LITTLE_ENDIAN,
    .ComSignalInitValue=&ComSignalInitValue_UH_Stunde_oDiagnose_01_oMIBCAN_222168aa_Rx,
    .ComSignalType_Par=SINT16,
    .ComTimeout=5,
    .ComHandleId=32,
    .ComFilter = NULL,
},
```

Figure 1 Signal configuration

```
uint8 buffer[8] = {0b00010001, 0, 0, 0, 0, 0, 0, 0};
pduInfoPtr = malloc(sizeof(PduInfoType));
pduInfoPtr->SduDataPtr = &buffer[0];
pduInfoPtr->SduLength = 5;
Com_RxIndication(1, pduInfoPtr);
Com_ReceiveSignal(32, SignalDataPtr);
printf("\n");
for(int i = 0; i < 8; i++){
    printf("SignalDataPtr[%d]: %d\n", i, *((uint8_t*)SignalDataPtr));
    SignalDataPtr++;
}
```

Figure 2 Main test

```
PS C:\Autosar-COM\testreceieve\inc> ./main_test2
Pdu Buffer data: 0x11 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Signal buffer:0xF8 0xFF 0x00 0x00 0x00 0x00 0x00 0x00
SignalDataPtr[0]: 248
SignalDataPtr[1]: 255
```

Figure 3 Test result

Explain:

先將 pduId=1的 pdu 送至 COM module。想要取得 signalId=32的資料，需先找到對應 pdu 後，從 bit1 取出長度4bit 的資料，經由型別轉換後傳給 RTE 指定位置。

邊坡監測系統實驗與資料分析

Experience of Slope Monitor System and Data

Analysis

指導教授：張大緯

專題成員：莊程傑

開發工具：STM32CubeIDE1.14.1、Colab

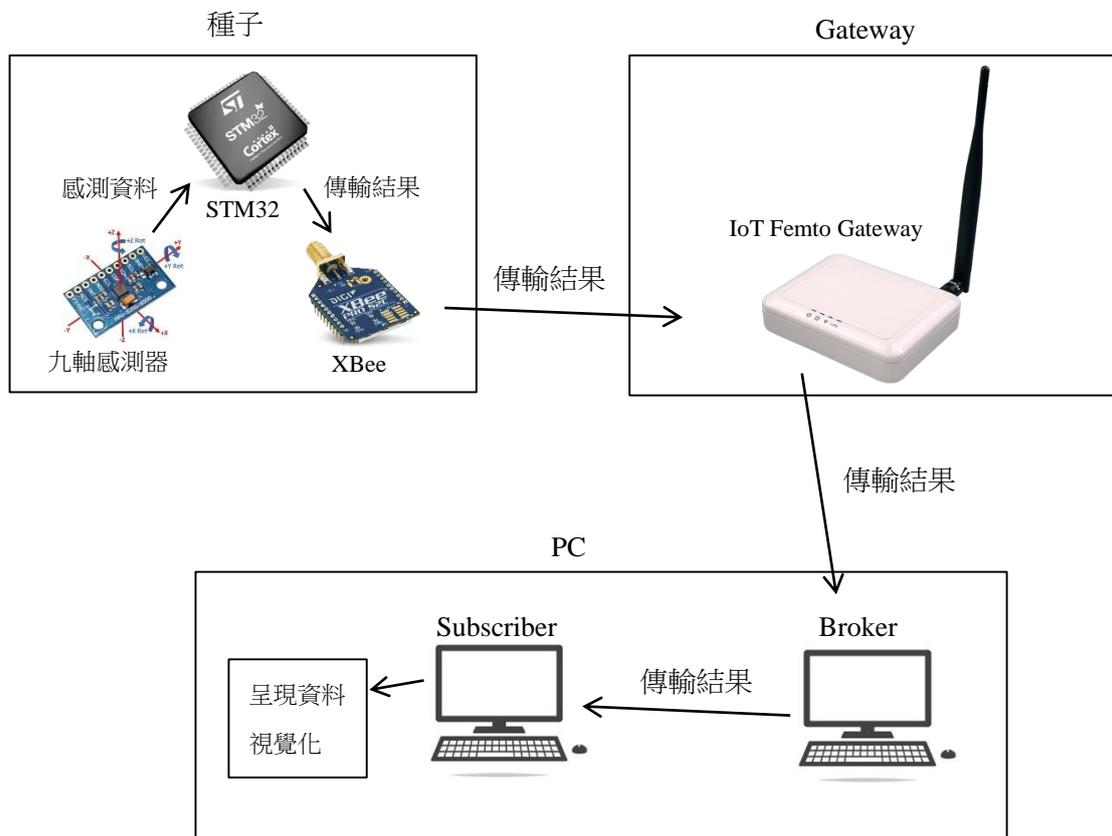
測試環境：Windows 11

一、簡介：

臺灣位於地震帶及颱風密集之地區，然而近年來陸續發生的邊坡崩塌事件，其災害搶救之順暢度與效率仍然是受到檢討之議題。究其原因，偵測災害的感測器遍及率低，無法即時得知災害發生。因此，我想設計且測試一套系統，並利用低成本的感測器對其做滑坡偵測，實驗是否能改善判斷的精準度。

我們首先利用 STM32 搭載 MPU9250 九軸感測器建立一個邊坡種子模組，透過資料濾波以及演算法判斷是否發生滑坡，若判斷成功則將滑坡警報透過 xbee 通訊模組傳至 gateway，再透過網際網路傳至 PC。

以下為系統架構圖：



二、測試結果：

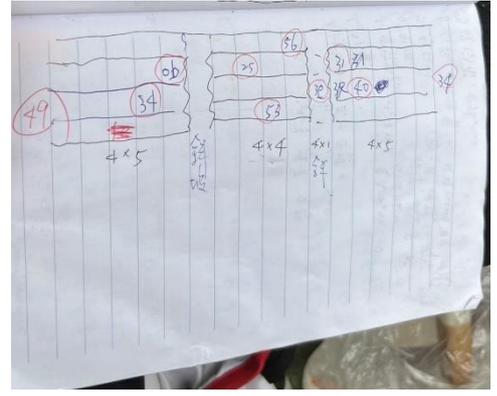
我們首先透過 Mahony 演算法求出物體姿態，得到重力方向，以方便計算位移變化，用來判斷邊坡是否滑落。本次實驗在南投縣仁愛鄉惠蓀林場，且由交管系李威勳老師、測量系郭佩棻老師以及中興大學團隊引領實驗的進行。下圖為實驗照片，種子埋在圖一圖二裡邊坡上標示的格子範圍內，圖三則是當時紀錄每顆種子的位置。



圖一



圖二

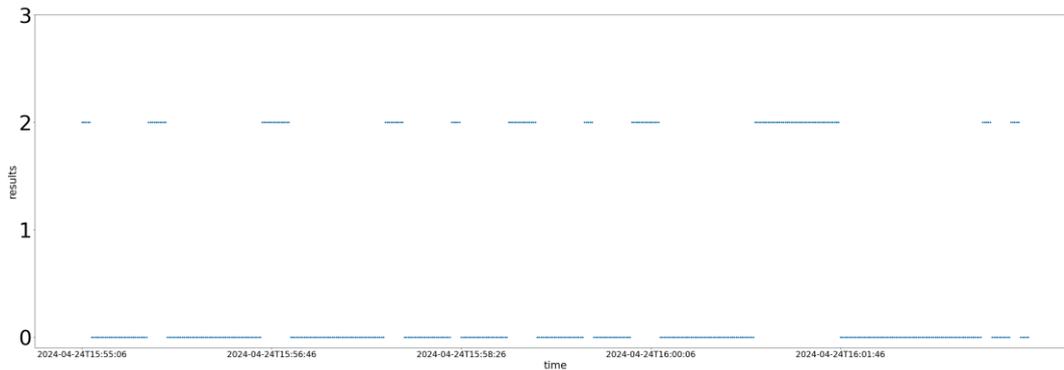


圖三

```

系統管理員: 命令提示字元
Client null received CONNACK (0)
Client null sending SUBSCRIBE (Mid: 1, Topic: GIOT-GW/UL/82029CF6BD77, QoS: 0, Options: 0x00)
Client null received SUBACK
Subscribed (mid: 1): 0
Client null sending PINGREQ
Client null received PINGRESP
Client null sending PINGREQ
Client null received PINGRESP
AC
E:\Program Files (x86)\mosquitto>mosquitto_sub -d -h 192.168.55.114 -t GIOT-GW/UL/80029CF7BD76 -u admin -P admin
Client null sending CONNECT
Client null received CONNACK (0)
Client null sending SUBSCRIBE (Mid: 1, Topic: GIOT-GW/UL/80029CF7BD76, QoS: 0, Options: 0x00)
Client null received SUBACK
Subscribed (mid: 1): 0
Client null received PUBLISH (d0, q0, r0, m0, 'GIOT-GW/UL/80029CF7BD76', ... (285 bytes))
[{"channel": "923200000", "sf": 10, "time": "2023-03-23T18:03:41+08:00", "gwip": "0.0.0.0", "gwid": "000080029cf50974", "repeater": "00000000ffffffff", "systype": 0, "rssi": -48.0, "snr": 17.5, "snr_max": 30.8, "snr_min": 12.0, "macAddr": "0000000000000014", "data": "7777", "frameCnt": 0, "fport": 1}]
Client null received PUBLISH (d0, q0, r0, m0, 'GIOT-GW/UL/80029CF7BD76', ... (285 bytes))
[{"channel": "923200000", "sf": 10, "time": "2023-03-23T18:04:00+08:00", "gwip": "0.0.0.0", "gwid": "000080029cf50974", "repeater": "00000000ffffffff", "systype": 0, "rssi": -53.0, "snr": 20.0, "snr_max": 31.3, "snr_min": 16.3, "macAddr": "0000000000000014", "data": "7777", "frameCnt": 0, "fport": 1}]
    
```

圖四：電腦接收資料



圖五：資料視覺化

AES 加密系統

指導教授：陳培殷

專題成員：黃律瑛、黃芊

開發工具：Python、C、Arduino、Verilog

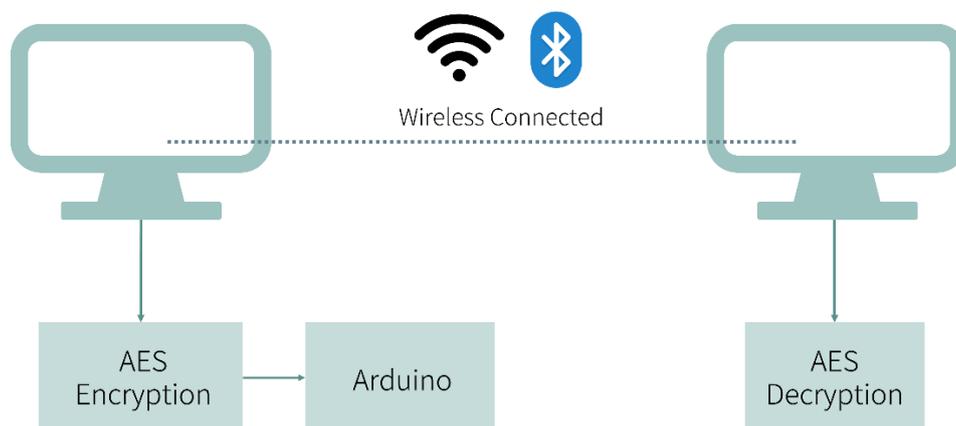
測試環境：

Vivado、Xilinx Vitis、Zedboard Zynq-7000

一、簡介：

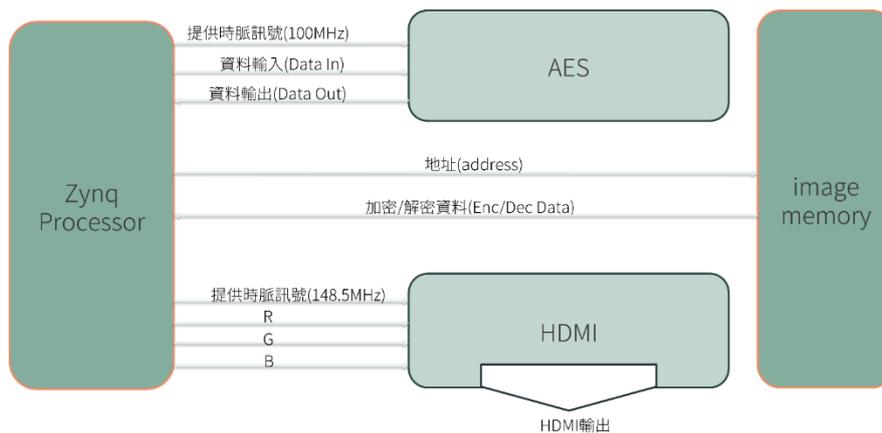
隨著科技的發展，我們即將邁向後量子時代，而最令人擔憂的便是資安問題。量子電腦是量子力學應用於電腦的一種革新工具，具有強大的運算能力，密碼學常見的質因數分解問題、離散對數問題、橢圓曲線問題等等，量子電腦都可以輕易解決，因此現在常用的 RSA 和橢圓曲線加密都將因為量子電腦即將變得不再安全，目前量子電腦的 Shor' s Algorithm 已經可以攻破 RSA, DSA, ECC 等加密演算法。除此之外，Grover' s Algorithm 借助量子電腦可以將暴力破解難度從 N 次降為 $N^{1/2}$ 次嘗試。也就是說，128位元的加密就變成64位元加密的安全性，但是只需要將金鑰長度增為兩倍，即可抵抗這類的攻擊，由此可知，非對稱式加密 (RSA, ECC) 的演算法是需要重新設計的，而對稱式加密 (AES) 演算法僅需增加位元數即可，演算法無須大幅變動。

因此，我們的題目希望能藉由電路實作 AES-256加密，來達到快速、輕量、低功耗的加密。我們利用 Zedboard 模擬加密晶片，將資料在 FPGA 板上加密，再利用無線傳輸將加密資料傳到另一台終端裝置，再利用 FPGA 解密還原資料。而未來希望能直接將晶片 (ASIC) 實作在電腦上，在傳出資料之前能快速又低耗能的將資料加密。



圖一：實作之簡易架構圖

二、測試結果



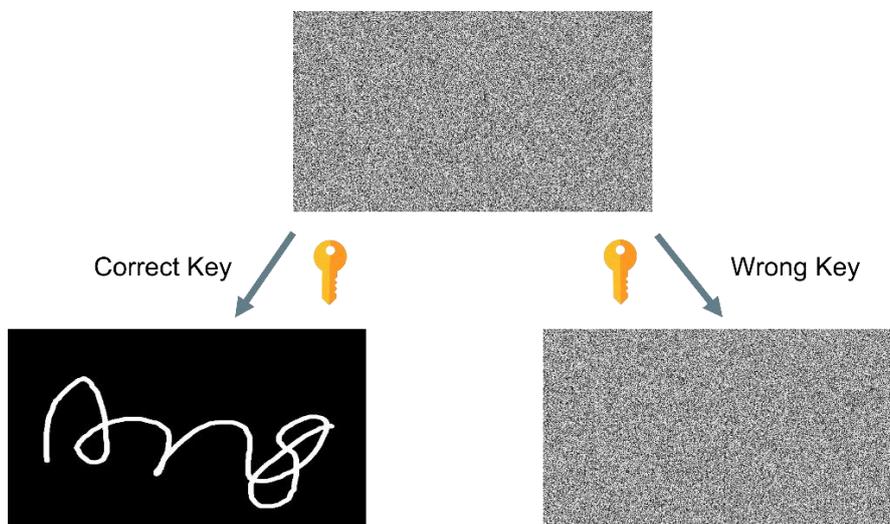
圖二：Block Design Big Picture

我們的題目是以 Verilog 結合 Zedboard Zynq-7000 所實作出來的 AES 加密系統，接下來會用圖二的架構作介紹。

首先，我們藉由讀取照片資料(image memory)，接收使用者想做的功能(加密/解密/結束)，以及使用者所輸入的 key(Data in)，將這些資訊傳送至 AES 進行 AES 演算法。

接下來，加密好的圖片資料會存在 Memory 裡，藉由 Access Memory 將加密好的圖片資料讀取出來，再用 HDMI 的方式呈現出來。(此時為加密的圖片)。下一步，我們會再詢問使用者一次想使用的功能，若使用者想解密，必須輸入前一次所輸入的金鑰進行解密。如同上一步得到功能、金鑰後，將使用者所加密的圖片資料一起傳送到 AES_0進行解密動作。

如果使用者輸入正確的金鑰，將會得到原本那張圖，反之，圖片還會是一片雜訊，最後一樣運用 HDMI 的方式進行輸出。以下是最終呈現出來的結果圖：



圖三：加解密結果

水下影像電路設計與實現

Design and Implementation of Underwater Imaging Circuit

指導教授：陳培殷、陳朝鈞

專題成員：黃偉峰、洪茂崧

開發工具：Verilog

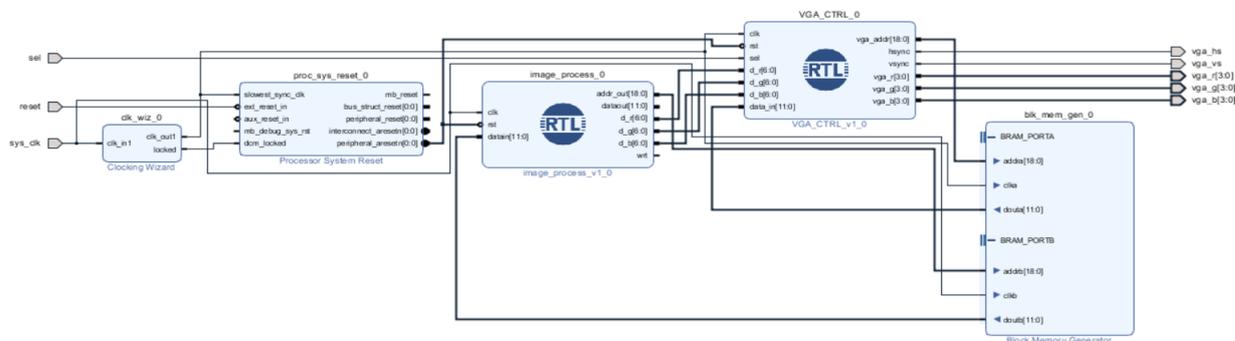
測試環境：Vivado

一、簡介：

海洋占地球約72%的面積，但人類對海洋的認識卻僅僅有百分之五不到。造成如此現象乃是因為受限於環境，我們難以僅依靠肉身克服深海的高壓與低溫，也因光線不足而不易探索，所以必須時常先透過儀器進行水下攝影再進行分析研究。

然而水下攝影的難度遠比在陸地上來的更高，當陽光照射入水中時，進入水中的光線部分又會被水中的懸浮粒子擴散或直接被水吸收。此現象會造成拍出來的影像呈現出現顏色偏差的狀況，有礙於進行研究分析；且由於國內外之論文期刊有水下影像處理相關的論文數量不少，但幾乎都是使用軟體進行後期調整，很少使用硬體電路來做及時處理的相關實作。

因此，我們的題目希望能透過演算法及電路設計，將於水下拍攝的影像進行優化處理，使其盡量能達到與在陸地上拍攝相同的品質，並且能比後期使用軟體調色更為快速，使得在水下拍照當下就能夠透過相機馬上得到調色後畫面，藉此以期望能促進海洋研究、海底探勘，抑或是水底藝術攝影等產業發展。

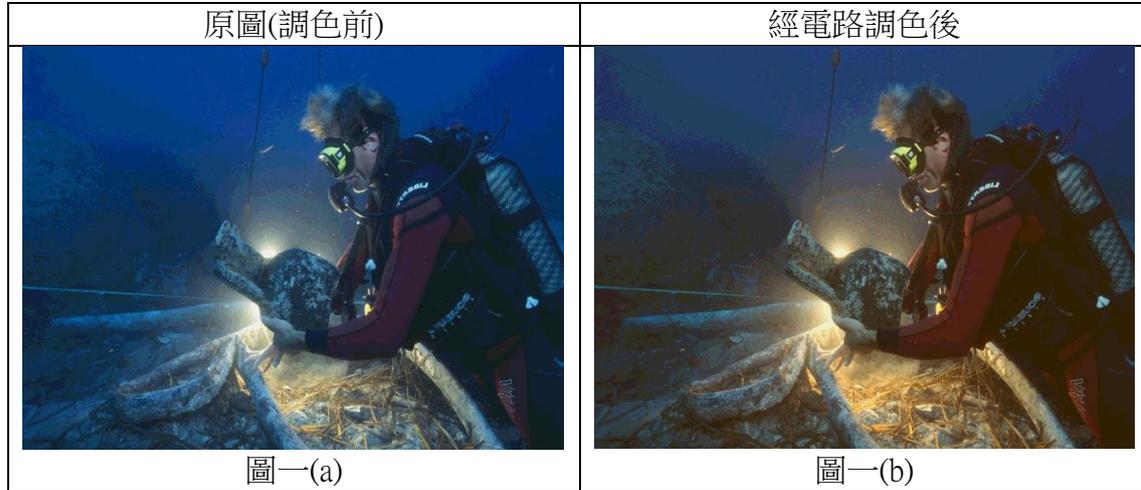


我們專題係以硬體描述語言 Verilog 來實作，系統架構圖與如上圖所示，利用以上電路將原圖進行調色，最終交由 vga 將各 pixel 傳輸至螢幕顯示並且可以透過 switch 及時調整顯示原圖或調色過後的結果。

二、測試結果：

(A).主觀結果比較(圖片處理皆使用640*480 RGB:12bit 的解析度下做測試)

(1). 電路調色後成果與原圖差異比較如下:



(B).客觀指標比較(粗體代表電路實作成果勝過參考期刊的結果)

(1). UIQM

	參考期刊(a) (軟體實作調色部分)	本篇電路(硬體)(b)
圖一	1.90318	1.92698

(2). UCIQE

	參考期刊(a) (軟體實作調色部分)	本篇電路(硬體)(b)
圖一	0.6035	0.6034

(3). SSIM

	參考期刊(a) (軟體實作調色部分)	本篇電路(硬體)(b)
圖一	0.806737	0.803958

(4). PSNR

	參考期刊(a) (軟體實作調色部分)	本篇電路(硬體)(b)
圖一	15.714845	15.676878

(5). 處理效率比較

	參考期刊(a) (軟體實作調色部分)	本篇電路(硬體)(b)
處理時長(ms/image)	20~30ms	3.072ms

可以看到本篇電路的處理速度比起參考論文的軟體實作有近十倍的速度差距而且在各種評價指標下兩者處理出來的圖片也都十分相近。

QOI 圖像壓縮電路設計

Digital Circuit Design For QOI Image Compression Algorithm

指導教授：蘇銓清、陳培殷
專題成員：何寬羿、林清弘
開發工具：Verilog、Matlab、
Vivado、Xilinx SDK、Modelsim
Xilinx Zedboard-Zynq-7000-7z020
測試環境：Windows 10

一、簡介：

隨著數位影像在各個領域的廣泛應用，有效的影像壓縮方法變得越來越重要。許多標準壓縮技術雖然能夠降低存儲需求和提升傳輸速度，但在處理效率和簡化實現的需求上仍有所欠缺。因此，我們將探討並實現 **Quite OK Image (QOI)** 壓縮格式，這是一種專為快速編解碼而設計的無損圖像壓縮方法。

QOI 格式是一種高效且簡單的圖像壓縮方法，提供無損壓縮，具備極高的編解碼速度和低實現複雜度。在多種應用場景中，QOI 格式表現出較其他壓縮方法更優越的性能。例如，與 JPEG 相比，JPEG 為一種有損壓縮格式，在需要高精度的應用中，例如醫學 X 光影像和照片的疊代處理與編輯，並不適用。而與同為無損壓縮的 PNG 格式相比，QOI 的解碼速度更快，更適合於實時應用需求。

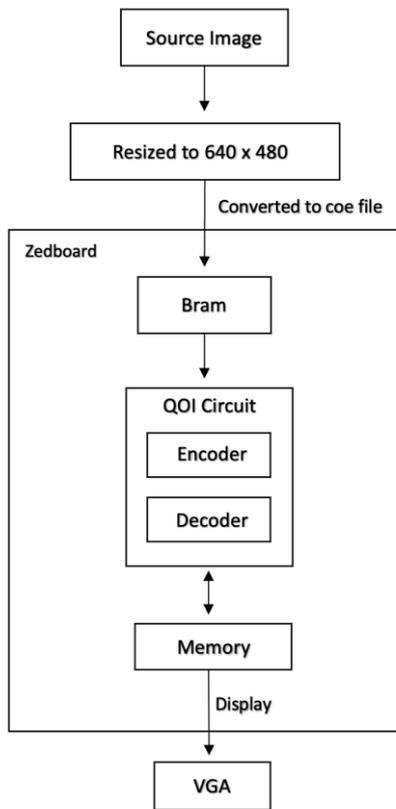
QOI 共分為下列五種編碼方式：

1. QOI_OP_RUN
2. QOI_OP_INDEX
3. QOI_OP_DIFF
4. QOI_OP_LUMA
5. QOI_OP_RGB

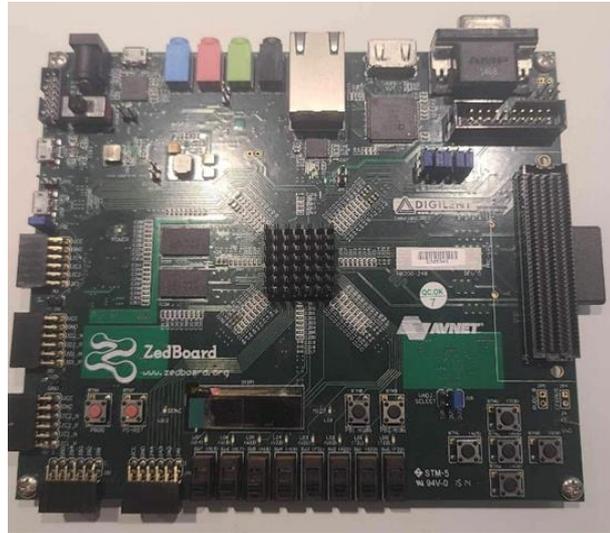
在電路的實作上，我們逐個讀取圖片的像素，根據 QOI 演算法判斷要用何種方式來進行編碼。

在專題實作中，我們首先透過軟體將原始圖片預先處理成能夠輸入至電路的圖片，將輸入圖片放入 **Bram** 中，經由 QOI 電路中的 **Encoder** 壓縮後儲存，再經由 **Decoder** 解碼獲得輸入圖片的資訊並顯示至螢幕上。顯示在螢幕上的輸出圖片會與輸入的圖片一致，可以證實 QOI 演算法為一項無損壓縮技術。

以下為系統架構圖及本組在專題中使用的 Zedboard-Zynq-7000-7z020：



圖一、系統架構圖

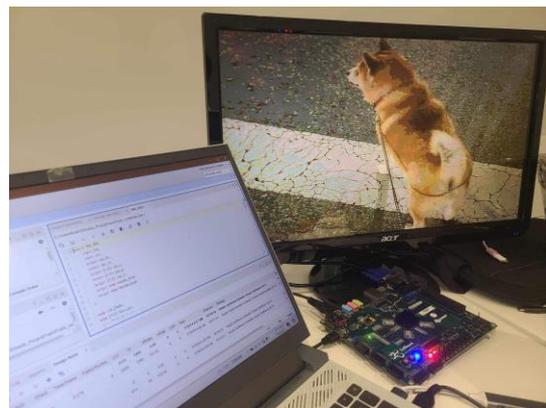


圖二、Zedboard-Zynq-7000-7z020

二、測試結果：



圖三、輸入圖片



圖四、輸出至螢幕上的結果圖

真隨機亂數產生器結合高速影像加密電路

High-Speed Image Encryption Circuit

With True Random Number Generator

指導教授：陳培殷、陳朝鈞

專題成員：蔡鎬駿、吳翰宇

開發工具：Verilog、Python、FPGA

測試環境：Vivado、Modelsim

一、簡介：

透過真亂數產生器(TRNG)，及改良過後的 Blowfish 影像加密 (Image Encryption) 技術，去實現一個高速影像加密電路，期望能藉由這項技術，能夠針對重要的影像資料，進行高亂度且快速的加密，去遏止有心人士竊取。

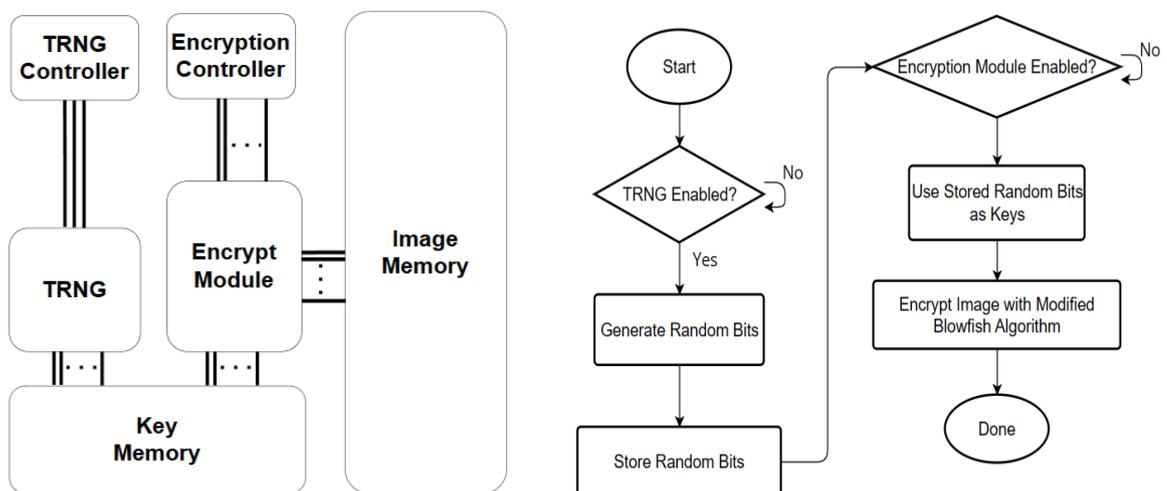
1. 藉由高亂度的亂數獲得更佳的安全性：

所實作的真亂數產生器(TRNG)能提供高亂度的亂數，去使得密鑰更不容易被破解，透過並結合 Blowfish 演算法，去實現更高安全性的加密電路，附圖為參考資料中，TRNG 所產生的亂數經過測試後的數據。

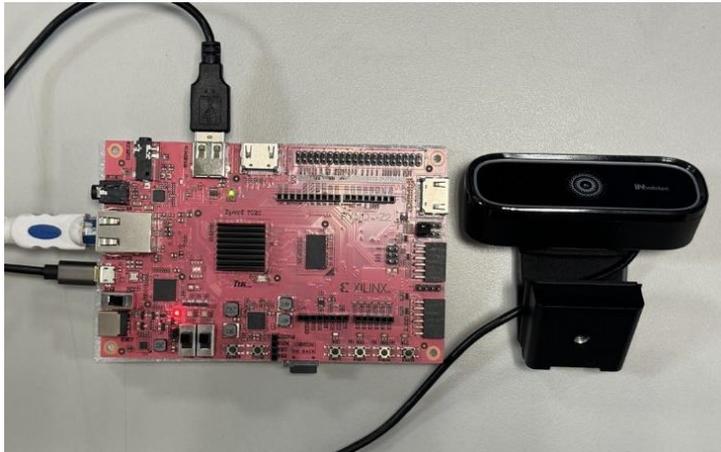
2. 結合更為輕量且高效的 Blowfish 演算法：

選用改良過後的 Blowfish 演算法，去結合真亂數產生器(TRNG)去提供密鑰，將會比市場常用的其他加密演算法快速且更輕量，同時提供高安全性。

以下為架構圖和運作流程圖：

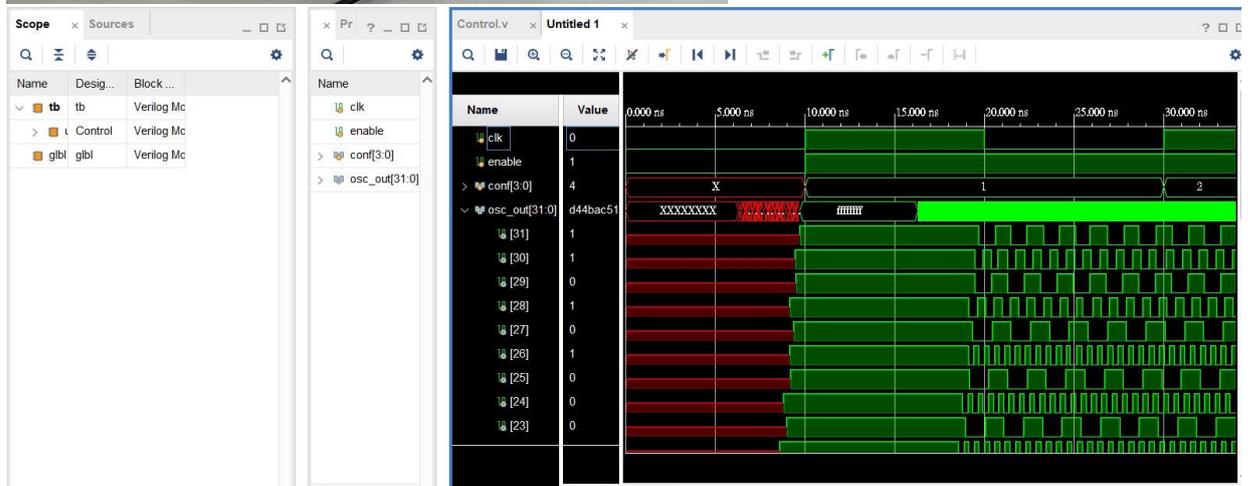


二、測試結果：



圖一：

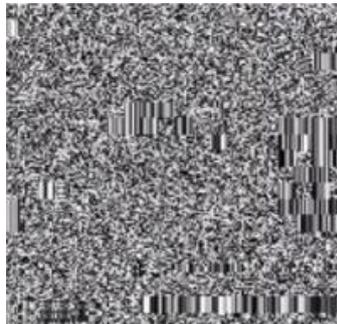
選用 FPGA device(Xilinx AUP PYNQ-Z2)來實作高效率影像加密加速器，並透過 V12 網路視訊攝影機(鏡頭：2P2G 定焦/直徑8m，尺寸：94x80x38 mm)進行拍攝，將拍攝完的影像再傳回電路去加密處理。



圖二：透過軟體 Vivado 搭配 ILA 進行模擬，完成真亂數產生器(TRNG)，附圖為透過 Waveform 產生亂數(在不同時間產生0和1輸出)，作為密鑰去結合改良過後的 Blowfish 演算法去進行加密。



圖三：拍攝照片



圖四：加密後的照片



圖五：解密後的原圖

使用高速影像加密電路，並針對 V12 網路視訊攝影機所拍攝的照片，進行上述的加密步驟後，可得到圖四中的 Encrypted Image，成功加密我們所要保護的重要影像，再依照密鑰反向進行解密後，即可將原本的拍攝照片還原。

用機器學習偵測 DDoS

Using Machine Learning to detect DDoS

指導教授:洪昌鈺

專題成員:許桓瑞

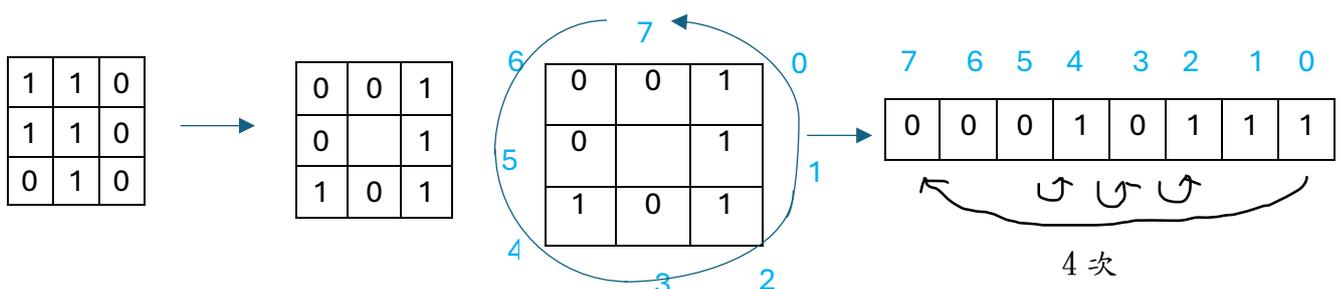
開發工具:python

測試環境:window

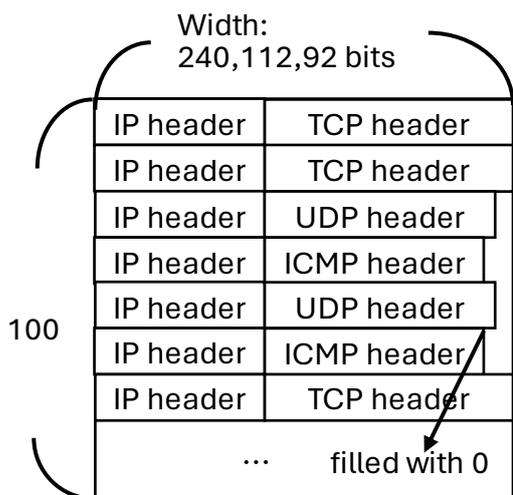
一、簡介

DDoS，分散式阻斷服務攻擊，攻擊者會對受害者的系統或網路進行大量訪問或請求，我針對的方向是流量攻擊(flood attack)，有 3 種，SYN flood，透過 TCP 的連線機制，攻擊者會送大量的 SYN 封包給受害者，受害者會逐一回應並等待接收 ACK 封包，但攻擊者不會發送 ACK 封包，導致受害者會有一段時間被占用；UDP flood，攻擊者會送大量 UDP 封包給受害者，讓受害者一直檢查並回應攻擊者，無暇顧及正常使用者；ICMP flood，攻擊方式與 UDP flood 相似，傳送大量 ICMP 封包，讓受害者不斷回應攻擊者。這次基於 CNN 的想法，將封包的 header 提取出來並拼成圖像(如圖(二)所示)，作為 model 的輸入。這次用了 Decision Tree model, SVM model, Random Forest model 和 CNN 去做比較，特徵的部分採用圖像的紋理特徵，採取紋理特徵的方式有 Gray-Level Co-Occurrence Matrix (GLCM), local binary pattern(LBP), Image Gradient，GLCM 是計算兩個像素值在一定距離且相對角度在 0, 45, 90, 135 度所出現次數的矩陣，接著會計算對比度、能量和熵，對比度是計算共生矩陣主對角線上的轉動慣量，體現出矩陣內的值如何分布的，能量是計算矩陣內各個元素的平方和，熵計算了整個圖像的複雜程度，熵越大，圖像越複雜，因為有 4 種角度，所以對比度和能量各會算出 4 個數值，而熵是對全部計算再取平均，所以只會輸出一個值，最後結合起來就會是長度 9 的特徵向量(計算方式如圖(三)所示)；LBP 是計算每個像素對於在一定範圍內其他像素的大小關係(\geq :0, $<$:1)，再依照一定方向形成一個 2 進位的數字，把 0 接 1 和 1 接 0 的次數加總(要循環計算，計算方式如圖(一)所示)，所以會有 0~8 次，全部算完後，統計出現次數的個數，最後形成一個長度為 9 的特徵向量；Image Gradient 是計算整體的圖像強度，會計算每個像素的上下及左右差值，並對其平方和開根號算出 gradient，具體計算是利用 Sobel 濾波器來計算水平梯度和垂直梯度，再利用這兩個梯度算出角度並依 0 到 π (不包括 π)以 $\pi/8$ 為間隔做直方圖，所以就有 7 個數值的輸出(詳細計算請看簡報)。我們有 10000 張圖像，每張圖像會產生一組特徵向量，GLCM, LBP, Image Gradient 分別會產生長度為 9,9,7 的特徵向量，並將這 3 種特徵向量結合成長度 25 的向量，最終會有 10000*25 的矩陣輸入 model 中，而輸出的部分，我稍微簡化一下，原本一張圖像的輸出是長度 3 的向量，分別代表是哪一種攻擊方式(SYN flood, UDP flood, ICMP flood)，但是長度 3 的輸出計算 confusion matrix 不太嚴謹，所以我把輸出改成長度 1，僅僅判斷是否有攻擊，有攻擊輸出 1，反之輸出 0。

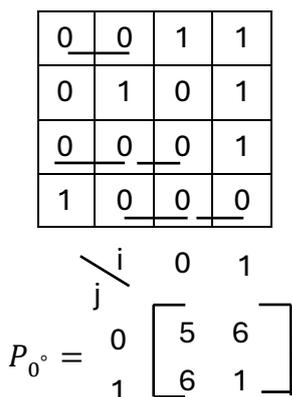
圖(一)



圖(二)



圖(三)



$$P(i,j,d,\theta)$$

$$P(0,0,1,0^\circ)=5$$

0 和 0 在相對 0 度且距離為 1 出現 5 次

$$\text{對比度公式: } \sum_i \sum_j (i - j)^2 * P(i, j)$$

$$\text{能量公式: } \sum_i \sum_j P(i, j)^2$$

$$\text{熵公式: } -\sum_i \sum_j P(i, j) \log P(i, j)$$

for i=0 to 1, and j=0 to 1

$$\text{contrast} = \{ \text{contrast}_{0^\circ}, \text{contrast}_{45^\circ}, \text{contrast}_{90^\circ}, \text{contrast}_{135^\circ} \}$$

$$\text{energy} = \{ \text{energy}_{0^\circ}, \text{energy}_{45^\circ}, \text{energy}_{90^\circ}, \text{energy}_{135^\circ} \}$$

$$\text{entropy} = (\text{entropy}_{0^\circ} + \text{entropy}_{45^\circ} + \text{entropy}_{90^\circ} + \text{entropy}_{135^\circ}) / 4$$

最後輸出 9 個數值

二、測試結果

從結果上可以發現，輸出長度為 3 的準確度都比較低，我認為這是因為除了要判斷攻擊外，還要識別出是哪一種攻擊方式，而輸出長度只有 1 的模型，只要有攻擊就可以輸出，所以事情不用做那麼多，準確度自然就比較高。與 CNN 的比較中，Random Forest 表現最佳，在只判斷是否有攻擊的情況下，準確度非常接近 CNN。

	CNN		Random Forest		Decision tree	
Output 長度為 1 (判斷是否有攻擊)	True positive	5965	True positive	5937	True positive	5897
	True negative	4031	True negative	4012	True negative	3948
	False positive	0	False positive	19	False positive	83
	False negative	4	False negative	32	False negative	72
	Accuracy	0.9996	Accuracy	0.9949	Accuracy	0.9845
	Precision	0.9996	Precision	0.9949	Precision	0.9845
	Recall	0.9996	Recall	0.9949	Recall	0.9845
	F1 Score	0.9996	F1 Score	0.9949	F1 Score	0.9845
Output 長度為 3 (要判斷哪種攻擊)	True positive	6949	True positive	6998	True positive	6987
	True negative	2983	True negative	2951	True negative	2933
	False positive	3	False positive	35	False positive	53
	False negative	65	False negative	16	False negative	27
	Accuracy	0.9932	Accuracy	0.9816	Accuracy	0.9679
	Precision	0.9995	Precision	0.9841	Precision	0.9655
	Recall	0.9907	Recall	0.9873	Recall	0.9789
	F1 Score	0.9951	F1 Score	0.9857	F1 Score	0.9722

Reference:

- [1] 方嘉祥, CNN based DDoS attack detection with raw packet data, 成功大學資訊所碩士論文
- [2] https://blog.csdn.net/Jen_Hsueh/article/details/104754460 [圖像處理_GLCM 矩陣計算]
- [3] <https://medium.com/@trapti.kalra/texture-analysis-with-deep-learning-for-improved-computer-vision-aa627c8bb133#b9ea> [紋理特徵分析介紹]
- [4] <https://blog.csdn.net/zfjBIT/article/details/90638844> [LBP uniform pattern 特徵]

於 P4 實現基於 CountMin 和 Hyperloglog 的 DDoS 攻擊偵測

DDoS Detection by using Hyperloglog and Count-Min

Sketch on P4 Switch

指導教授：洪昌鈺; 專題成員：吳驊明、陳柏佑

開發工具：Python、P4、Mininet; 測試環境：Ubuntu 18.04

一、簡介：

P4（可程式化、協議獨立、封包處理器）是一種開源的、專用於網路設備的程式語言，它可以來自定 Data plane 設備（交換機、路由器、網路介面卡、過濾器）該如何處理封包。

為了方便偵測 DDoS 攻擊，我們在 P4 Switch 上蒐集封包的資訊，用佔空間極少的 Sketch 儲存它們。在達到設定的警示時，通知 Controller[1][2]來判斷是否為 DDoS 攻擊。(見圖 1)

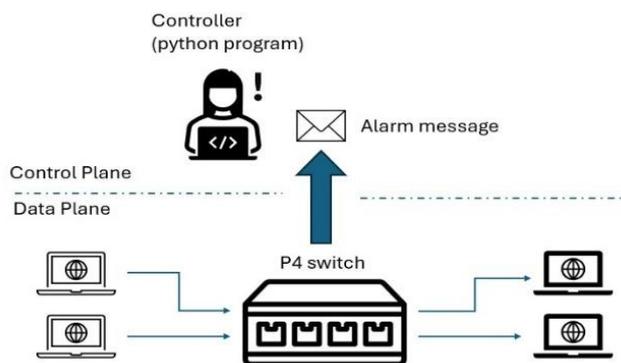


圖 1 專題的整體架構

實務方面，參考論文^[3]在 P4 Switch 上透過 CountMin Sketch 來統計「每一個 destination 分別收到了幾個封包」，另一方面，假如某一個特定的 destination 收到過多封包時，再啟動 Hyperloglog 來估計「這個 destination 收到的封包大約來自幾個不同的 source」。(見圖 2)

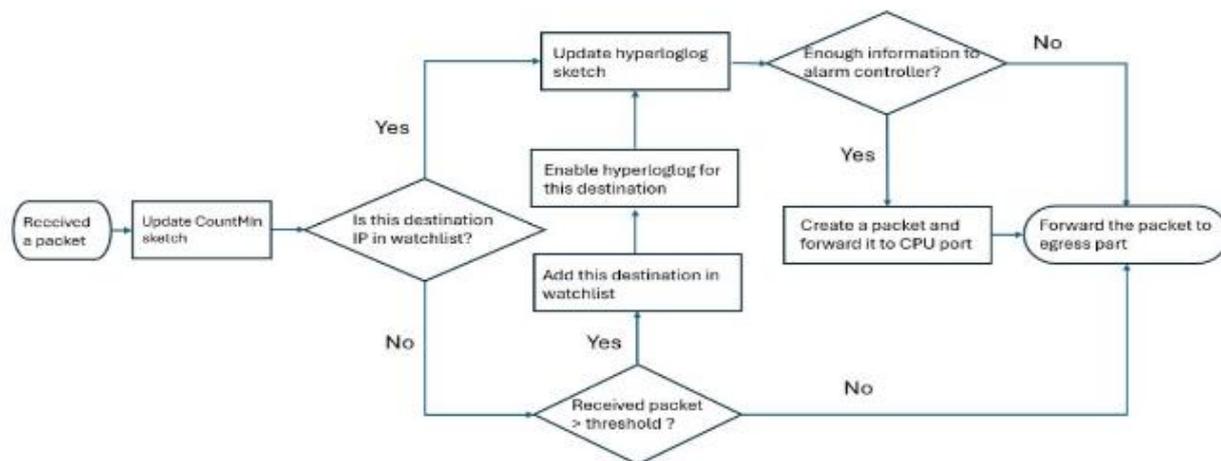


圖 2 P4 Switch 上的大致流程圖

CountMin 類似於多個 Hash Table，但是它的 Hash Function 不只一個。可以想像，一個 key 會因為多個 Hash Function 得到不同的 index，在每個 hash table 的不同 index 處加一，方便我們多重確認。與傳統的 Hash Table 相比，大幅避免了發生 Collision 可能造成的誤判。(見圖 3)

Hyperloglog Sketch 是統計「在集合中有多少相異的元素」的方法，將 Hash 的結果(bits)依照 bucket 數量來擷取後 n 位，代表它屬於哪個 bucket，並記錄最先出現 1 的位置(應用 Bernoulli trial)，同時搭配 Harmonic Average 來將誤差值降低。(見圖 4)

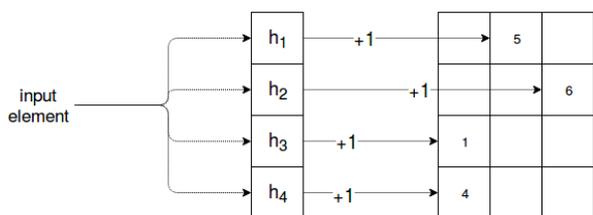


圖 4 CountMin Sketch 示意圖

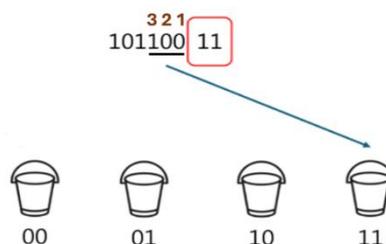


圖 3 Hyperloglog Sketch 示意圖

二、測試結果：

測試前，先決定 Hyperloglog 中適合的 Bucket 數量，估計值才會準確。圖 5 和圖 6 是在一個集合中放入 1000 個和 100000 個 source IP，並用 Hyperloglog 估計的結果，在我們的實驗(source 數量在 1000~100000 之間)中，我們設定 2^8 個 bucket。

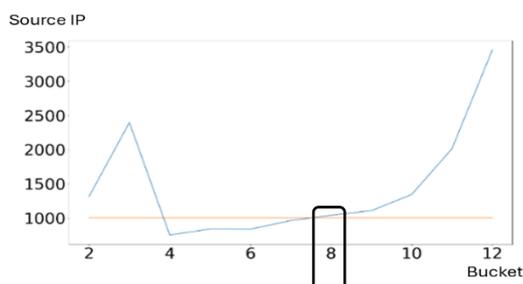


圖 5 1000 個不同元素的估計結果

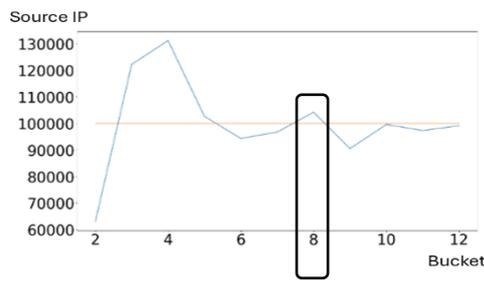


圖 6 100000 個不同元素的估計結果

最後的結果如圖 7.所示，我們對 5 個 destination 寄送封包，其中兩個(10.0.0.2 和 10.0.0.3)為高流量(其餘都沒有大於我們設定的 CountMin 標準值)。

可以看到 P4 Switch 在面對其中兩個高流量的 destination 都有將它們放進 watchlist 估計，也分別得到相對準確的估計值，這樣的結果可以幫助 Controller 判斷是否為 DDoS 攻擊。

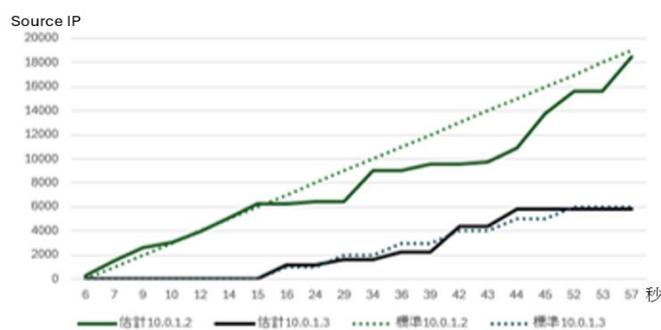


圖 7 結果

三、參考文獻：

- [1] “網路模擬與分析”，<https://hackmd.io/@110510549/ByyNkAasu>，2021
- [2] “第十六周 P4 筆記”，<https://hackmd.io/gRKNdQLaSVmQjorzJbeUNw>，2022
- [3] “DDoS Detection in P4 Using HYPERLOGLOG and COUNTMIN Sketches”，Vera Clemens, IEEE Symposium on Network Operations and Management, May 2023

殭屍網路特徵與分析

The Features of BotNet and its analysis

指導教授：張燕光； 專題成員：黃育笙、邢益城

開發工具：GCC、python； 測試環境：Linux ubuntu 6.0

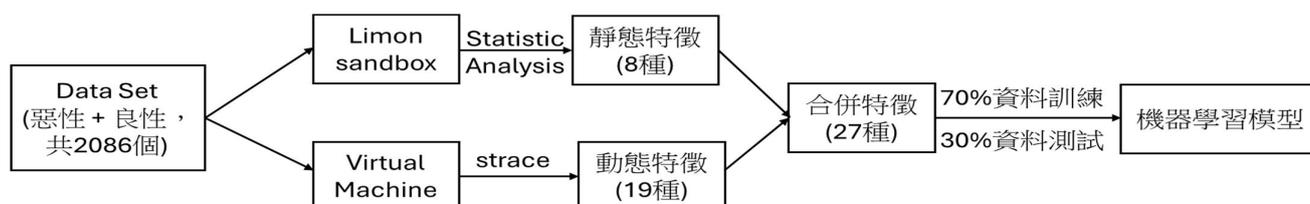
一、簡介：

在閱讀完學長的碩士畢業論文[1]後，我們希望在原作者的方法基礎上進行改進。當我們嘗試擷取機器學習所需的特徵時，可以將特徵分為靜態和動態兩類。靜態特徵與樣本的規格（如檔案大小）有關，不需要實際執行即可獲得，本次專題中我們使用 Limon Sandbox[2]的靜態分析功能來獲取靜態特徵。動態特徵的獲取則需要實際執行惡意軟體並計算其系統呼叫（system call）數量，我們利用 Linux 內建的 strace 指令來獲取動態特徵，請參考圖一。

我們遇到的主要問題是，惡意軟體不一定會表現出其真正的惡意行為。例如，殭屍網路病毒可能因為沒有接收到來自駭客的命令而一直處於等待狀態。為了解決這個問題，我們參考了著名殭屍網路 Mirai 的源代碼[3]，自製了一個殭屍網路，並以此作為惡意樣本。由於該樣本是我們撰寫的，因此我們可以透過各種命令來確保其表現出各種惡意行為。這樣就可以彌補機器學習訓練集不夠全面的問題。此外，原論文並沒有提到擷取 sendto 和 recvfrom 等系統呼叫作為特徵，因此我們新增了這兩個特徵，並測試其對準確率的影響。

我們實作的殭屍網路（botnet）如圖二「殭屍網路運作架構圖」所示。當它開始執行時，會等待我們自行架設的指揮與控制（C&C）伺服器的命令。此時，它會開始掃描 /proc/a_PID 資料夾底下所有進程（process）的信息，如命令行參數、環境變量、資源使用情況等，以達到監視系統的目的[4]。

在收到來自 C&C 伺服器的命令後，會利用 fork 創建一個子進程（child process）並發起 DDoS 攻擊，發送大量的垃圾 UDP 封包。在這階段會大量使用 sendto 這個系統呼叫（system call），這也是為什麼我們需要新增 sendto 作為特徵以改進模型準確率。



圖一：擷取動、靜態特徵的流程圖

資料集介紹：

名稱	種類	樣本數量	來源
bin	良性	428	Linux 系統內建
sbin	良性	197	Linux 系統內建
Linux Malware	惡性	464	[5]
MalwareBazaar Database	惡性	997	[6]

靜態特徵列表：檔案規格

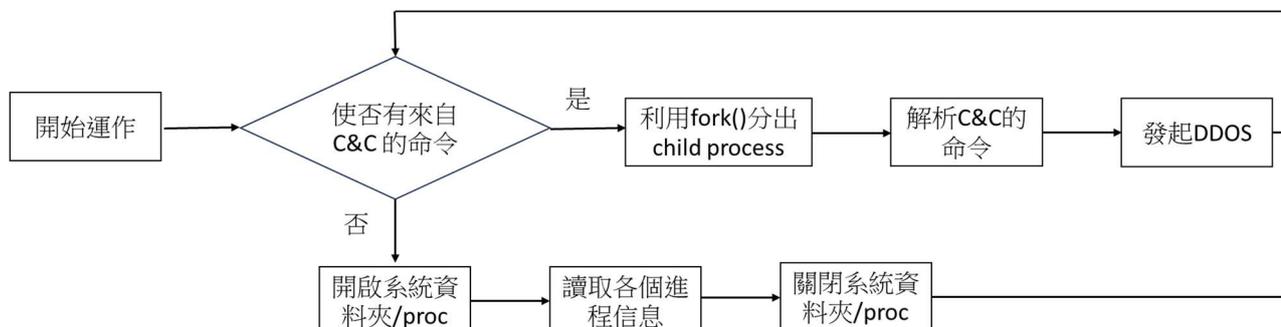
以下是原論文[1]作為靜態特徵的所需的檔案規格：

Number of Program Headers	Section header string table index	Size of .data Section
Number of section headers	Size of .text Section	Size of .rodata Section
File Size	Size of .bss Section	

動態特徵列表：系統呼叫(system call)

以下是原論文[1]作為動態特徵的 system call:

execve	dup	creat	unlink	rename	write	bind
read	socket	fork	dup2	pipe	connect	
close	kill	open	clone	accept	chdir	



圖二：殭屍網路運作架構圖

二、測試結果：

模型名稱	Accuracy	Precision	Recall
Decision tree(改進前)	95%	97.5%	96.3%
Decision tree(改進後)	95.8%	98%	96.76%
Logistic regression(改進前)	79%	78.2%	99.35%
Logistic regression(改進後)	80.9%	80.2%	98.27%
KNN(改進前)	94%	98.1%	98.35%
KNN(改進後)	94.3%	98.5%	94%
Random Forest(改進前)	95.8%	98%	96.3%
Random Forest(改進後)	96.2%	97.6%	96.9%
SVM(改進前)	74.2%	74.4%	99.35%
SVM(改進後)	74.3%	74.3%	99.1%
Neural network(改進前)	92.6%	98.1%	91.5%
Neural network(改進後)	93%	98.6%	92.2%

可以發現在我們改進特徵擷取的方式後，各模型的準確率確實有所提升。由於此次任務是辨認惡意軟體，因此低偽陰性（即高召回率）變得尤為重要，因為偽陰代表實際上為惡意但卻被模型辨認為良性的次數。在考慮到高召回率且同時具有較高的準確率和精確度的情況下，我們可以發現決策樹（Decision Tree）、K近鄰（KNN）、隨機森林（Random Forest）和神經網絡（Neural Network）最適合本次任務。

三、參考文獻：

- [1] 黃志翔 “Detecting malware distributed by IoT Botnet based on Machine Learning and Deep Learning”, 成大資訊碩士論文, 2023.01
- [2] ”Limon - Sandbox” 2015, <https://github.com/monnappa22/Limon>
- [3] ”Mirai-Source-Code” 2015, <https://github.com/jgamblin/Mirai-Source-Code/tree/master/mirai/bot>
- [4] ”The /proc Filesystem” 2009, <https://docs.kernel.org/filesystems/proc.html>
- [5] ”Linux Malware” 2014, <https://github.com/MalwareSamples/Linux-Malware-Samples>
- [6] ”MalwareBazaar Database” 2023, <https://bazaar.abuse.ch/browse/>

網購超商快取？不不不~網路資料快取！

Implementing NetCache for Load Balancing Based on SDN with P4 switch

指導教授：張燕光; 專題成員：林宸哲、古嘉雋

開發工具：p4-utils/python3/mininet/bmv2/p4c/p4runtime; 測試環境：Ubuntu 20.04

一、簡介：

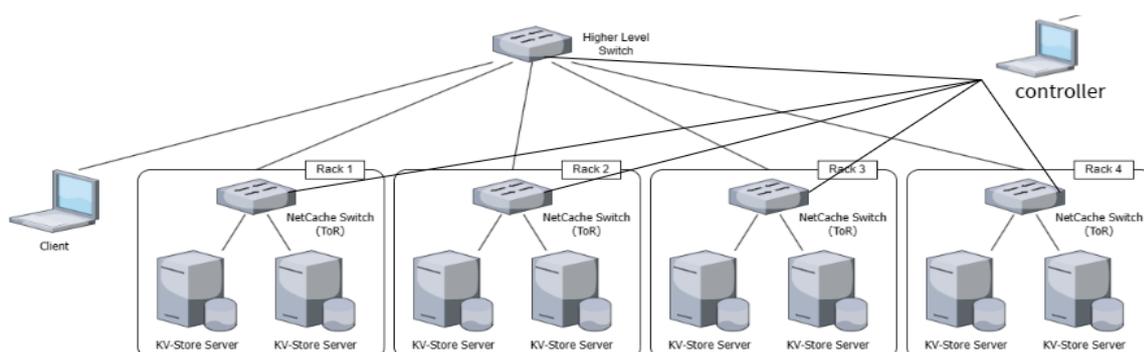
我們專題的動機是源於一篇論文(NetCache: Balancing Key-Value Stores with Fast In-Network Caching)，其作者想透過在可編程式交換器(P4 Switch)上實現 Key-Value 形式的資料快取功能，也就是 client 在查詢資料時須給出該資料的 Key 以得到對應的 value(資料本身)，而熱門資料會存在快取中，以解決有熱門資料的 servers 被造訪次數過多導致壅塞的流量不平衡的問題。

我們根據論文描述的 NetCache 概念，在 mininet 中模擬出了具有 netcache 功能的 P4 Switch 及 controller，switch 記錄下每份資料被查詢的頻率，並將被判斷為熱門資料的 Key 以及對應的 value 回報給 controller，controller 會將 Key 和查找 value 作為快取存進 switch 中，而 client 端在查詢資料時若資料的 key 有被快取到就能找到對應的 value，接著直接將資料返還給 client，而不需經過 switch 到 server 端，以此來降低熱門 server 的造訪次數。NetCache 也在顧及 Cache Coherence 的情況下有更新及刪除資料功能，也就是說 client 端若想要更新或刪除位於 server 端的某些資料時，若該資料也正位於 switch 快取之中，他也会被同步刪除或更新，以免發生快取資料與 server 資料不同的問題，此外若 switch 記憶體用量達一定標準則會以 LRU+LFU 方式刪除快取。

然而，論文中所採用的架構較為簡易，如圖一中框起來的部分，為單一 rack 架構，但在實際情況下通常會使用多個 racks 組成的複合架構。我們觀察到，就算快取功能有持續運作，不同 switch 內部的空間的利用不完全卻會讓快取效益降低，實際情況應會有某 switch 最先達到快取上限，而其他的 switch 卻仍有閒置的空間。因此，我們整合不同 switch 中的儲存資源，令閒置的 switch 也能存入別人的 hot item。

我們在各個 racks 之上新增了一個 Higher Level Switch(HS)，透過各個 netcache switch 對 controller 回報各自快取用量，找出當前空間剩餘最多的 switch，將 hot item 存儲於其中，並在 HS 中新增一個 flow entry，讓新的查詢封包可以重新導向至實際存放該 item 的 switch 上。如此一來，switch 內的快取空間使用率便可以得到平衡。

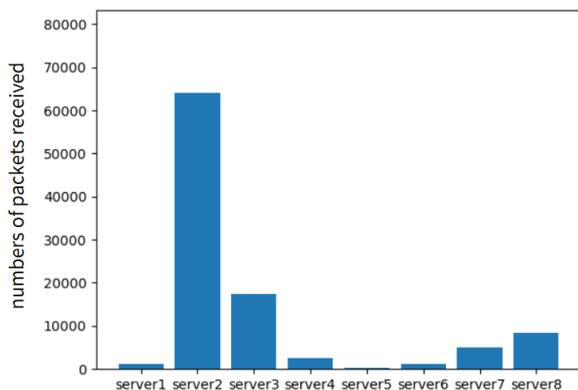
這是我們用於模擬測試的網路拓樸(圖一)：



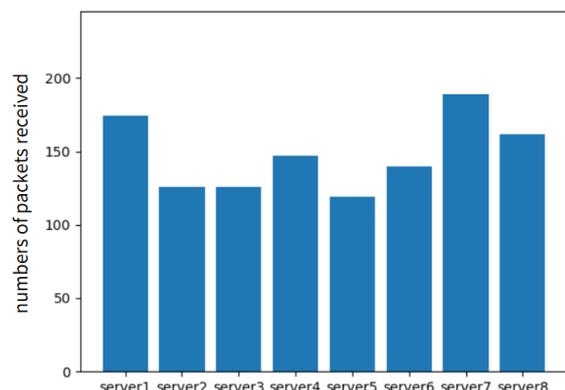
二、測試結果：

我們採用 zipf 機率模型生成了100000份查詢資料的封包，並且分別在有無 netcache 的情況下紀錄各個 server 端被查詢次數的情形，以及分別在有無 hot item 轉存的情況下紀錄各個 switch 隨時間增加的內部空間的使用率。

下面我們畫出在沒有(圖二)及有(圖三)netcache 的情況下各個 server 端被查詢的次數，X 軸為各 server 及其編號，Y 軸為收到的總封包數(被查詢次數)。可以看到沒有 netcache 的情況下 server 間的流量相當不均，次數也很多，但在有 netcache 時則大大減緩此現象。

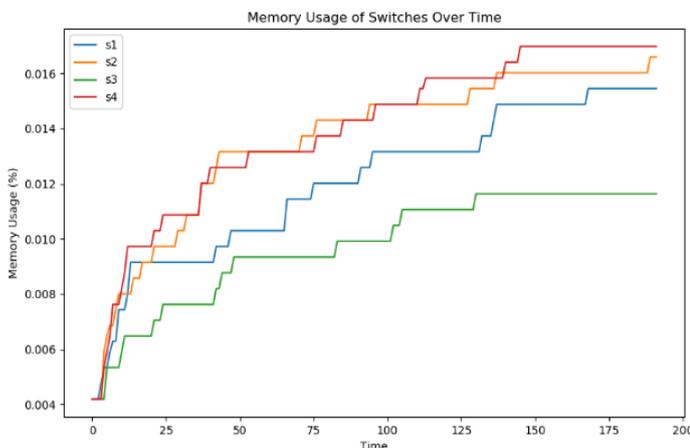


(圖二)

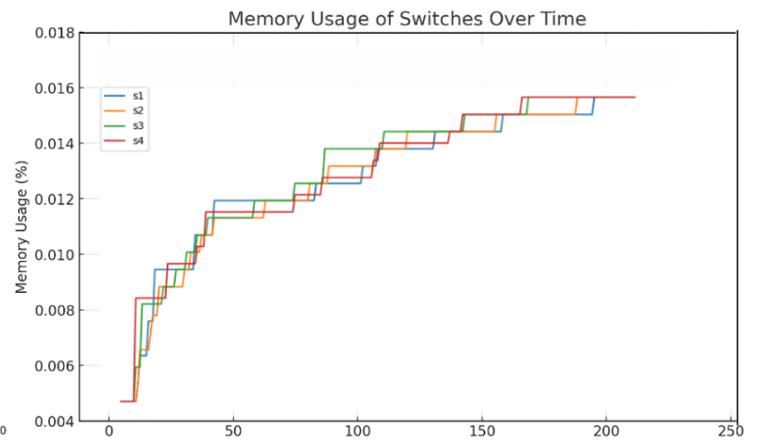


(圖三)

圖四、五為沒有以及有 hot item 轉存的情況下各個 switch 隨時間(X 軸)增加的內部空間的使用率(Y 軸)折線圖，可以看到在沒有 hot item 轉存時，某些 switch 的使用率會快速上升，而其餘的則上升緩慢，若快速上升的 switch 開始汰換自身 cache 時其他 switch 仍有閒置空間，會造成空間浪費，因此過多空間閒置應可以拿來多加利用，而有 hot item 轉存時所有 switch 內部空間使用率是平均增加的，如此一來可以避免閒置空間的產生並更有效的利用所有的空間。



圖四



圖五

三、參考資料：

- [1] Xin Jin, Xiaozhou Li, Haoyu Zhang, Robert Soulé, Jeongkeun Lee, Nate Foster, Changhoon Kim, Ion Stoica, "NetCache: Balancing Key-Value Stores with Fast In-Network Caching", Proceedings of the 26th Symposium on Operating Systems Principles (SOSP), October 2017 <https://doi.org/10.1145/3132747.3132764>.
- [2] The P4.org Architecture Working Group. 2022. P416 Portable Switch Architecture (PSA). <https://staging.p4.org/p4-spec/docs/PSA-v1.2.pdf>.

基於機器學習之低速緩慢 DDoS 攻擊偵測

A Slow DDoS Attack Detection Based On Machine Learning

指導教授：張燕光； 專題成員：張原愷、張辰凱

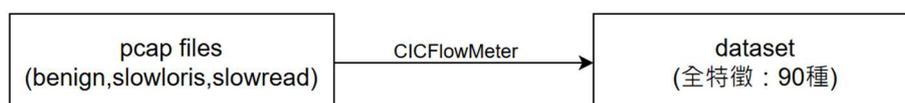
開發工具：Python, Scapy, CICFlowMeter； 測試環境：Ubuntu 22.04 LTS

一、簡介：

Slow HTTP Attack DDoS 是一種網路攻擊方式，透過發送大量低速、持續的 HTTP 請求使得伺服器資源耗盡，進而使其無法處理正常用戶的 HTTP 請求。與其他 DDoS 攻擊不同，低速緩慢攻擊通常利用合法的 HTTP 請求，但攻擊者會故意以非常緩慢的速度發送這些請求，以耗盡伺服器資源。在我們這組的專題中，我們修改並利用 CICFlowMeter[1] 提取機器學習所需的 flow 特徵，之後參考論文[2] 做資料前處理，最後實現基於機器學習的 Slow HTTP Attack 攻擊偵測，偵測的攻擊有兩種 (1) Slowloris (2) Slow Read。

(一) 資料前處理：

特徵選取：透過特徵選取降低模型複雜度以及減少對訓練資料集的過擬合(Overfitting)。



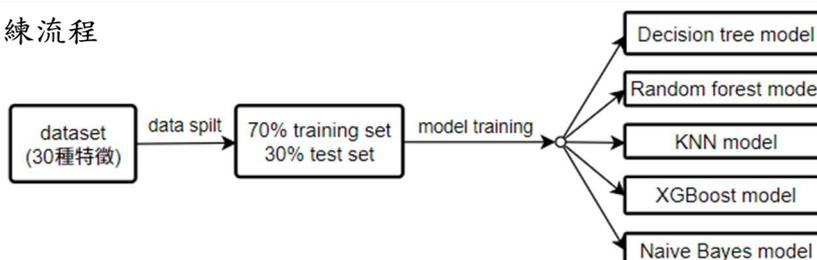
單變量特徵選取，我們採取 SelectKBest 方法：選取 K 個最好的特徵，K=30。



(二) 模型訓練：

資料集(種類)	訓練樣本數	測試樣本數	標籤
Benign	55591	10103	benign
Slowloris	21453	5901	slowloris
Slow Read	24369	5871	slowread

訓練流程



(三) 花費時間：

Total feature extraction time	33m 20s
Total feature selection time	46s
Total training time	4m 45s

二、測試結果：

(一) 訓練結果：以下是針對三個 Label 分析後的各個結果數值。

Model	Precision	Recall	F1-score	True positive rate	False positive rate
DecisionTree	0.9999	0.9998	0.9998	0.9998	0.0001
RandomForest	0.9999	1.	0.9999	1.	0.0001
KNN	0.9996	0.9999	0.9997	0.9999	0.0004
XGBoost	0.9999	1.	0.9999	1.	0.0001
Naive Bayes	0.9674	0.9569	0.9621	0.9569	0.039

表一：Benign

Model	Precision	Recall	F1-score	True positive rate	False positive rate
DecisionTree	0.998	0.9971	0.9976	0.9971	0.0004
RandomForest	0.9988	0.9985	0.9986	0.9985	0.0003
KNN	0.9983	0.9961	0.9972	0.9961	0.0004
XGBoost	0.998	0.998	0.998	0.998	0.0004
Naïve Bayes	0.8792	0.9194	0.8988	0.9194	0.033

表二：Slowloris

Model	Precision	Recall	F1-score	True positive rate	False positive rate
DecisionTree	0.9975	0.9985	0.998	0.9985	0.0008
RandomForest	0.9987	0.9987	0.9987	0.9987	0.0004
KNN	0.9969	0.9981	0.9975	0.9981	0.0009
XGBoost	0.9983	0.9981	0.9982	0.9981	0.0005
Naïve Bayes	1.	0.9856	0.9927	0.9856	0.

表三：Slow Read

(二) 實際偵測：以下分別是正常訪問網頁以及利用 slowhttptest 攻擊網站的結果。

```

in predict mode
Detector started!
-----
2024-05-19 14:15:58.560
INFO [Slow HTTP Attack Detector] Detect a 'benign' flow (src_ip: 192.168.10.102, dst_ip: 192.168.10.101, src_port: 33852, dst_port: 80, protocol: 6)
2024-05-19 14:16:06.305
INFO [Slow HTTP Attack Detector] Detect a 'benign' flow (src_ip: 192.168.10.102, dst_ip: 192.168.10.101, src_port: 33862, dst_port: 80, protocol: 6)
2024-05-19 14:16:13.554
INFO [Slow HTTP Attack Detector] Detect a 'benign' flow (src_ip: 192.168.10.102, dst_ip: 192.168.10.101, src_port: 33886, dst_port: 80, protocol: 6)
2024-05-19 14:16:20.558
INFO [Slow HTTP Attack Detector] Detect a 'benign' flow (src_ip: 192.168.10.102, dst_ip: 192.168.10.101, src_port: 40656, dst_port: 80, protocol: 6)
2024-05-19 14:16:26.822
INFO [Slow HTTP Attack Detector] Detect a 'benign' flow (src_ip: 192.168.10.102, dst_ip: 192.168.10.101, src_port: 46278, dst_port: 80, protocol: 6)
2024-05-19 14:16:39.826
INFO [Slow HTTP Attack Detector] Detect a 'benign' flow (src_ip: 192.168.10.102, dst_ip: 192.168.10.101, src_port: 34614, dst_port: 80, protocol: 6)

```

圖一：Benign 偵測

```

in predict mode
Detector started!
-----
2024-05-19 14:08:19.015
INFO [Slow HTTP Attack Detector] Detect a 'slowloris' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42928, dst_port: 80, protocol: 6)
2024-05-19 14:08:40.037
INFO [Slow HTTP Attack Detector] Detect a 'slowloris' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42932, dst_port: 80, protocol: 6)
2024-05-19 14:08:40.055
INFO [Slow HTTP Attack Detector] Detect a 'slowloris' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42940, dst_port: 80, protocol: 6)
2024-05-19 14:08:40.071
INFO [Slow HTTP Attack Detector] Detect a 'slowloris' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42950, dst_port: 80, protocol: 6)
2024-05-19 14:08:40.089
INFO [Slow HTTP Attack Detector] Detect a 'slowloris' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42958, dst_port: 80, protocol: 6)
2024-05-19 14:08:40.107
INFO [Slow HTTP Attack Detector] Detect a 'slowloris' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42960, dst_port: 80, protocol: 6)

```

圖二：Slowloris 偵測

```

in predict mode
Detector started!
-----
2024-05-19 14:12:14.305
INFO [Slow HTTP Attack Detector] Detect a 'slowread' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 58746, dst_port: 80, protocol: 6)
2024-05-19 14:12:37.806
INFO [Slow HTTP Attack Detector] Detect a 'slowread' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42866, dst_port: 80, protocol: 6)
2024-05-19 14:12:37.819
INFO [Slow HTTP Attack Detector] Detect a 'slowread' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42856, dst_port: 80, protocol: 6)
2024-05-19 14:12:37.835
INFO [Slow HTTP Attack Detector] Detect a 'slowread' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42850, dst_port: 80, protocol: 6)
2024-05-19 14:12:37.861
INFO [Slow HTTP Attack Detector] Detect a 'slowread' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42844, dst_port: 80, protocol: 6)
2024-05-19 14:12:37.875
INFO [Slow HTTP Attack Detector] Detect a 'slowread' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42842, dst_port: 80, protocol: 6)
2024-05-19 14:12:37.896
INFO [Slow HTTP Attack Detector] Detect a 'slowread' flow (src_ip: 192.168.10.100, dst_ip: 192.168.10.101, src_port: 42830, dst_port: 80, protocol: 6)

```

圖三：Slow Read 偵測

三、參考文獻：

- [1] "CICFlowMeter", 2020, <https://gitlab.com/hieulw/cicflowmeter>.
- [2] Yin Mon Swe, Pye Pye Aung, Aye Su Hlaing, "A Slow DDoS Attack Detection Mechanism using Feature Weighing and Ranking", the 11th Annual International Conference on Industrial Engineering and Operations Management, 2021. 03.

建立於MiinStore物聯網平台之Matter閘道器實作

Implementation of the Miinstore Matter gateway

指導教授：謝孫源教授

專題成員：陳子揚

開發工具：C、Python

測試環境：Ubuntu Server 22.04.2 LTS (64-bit)

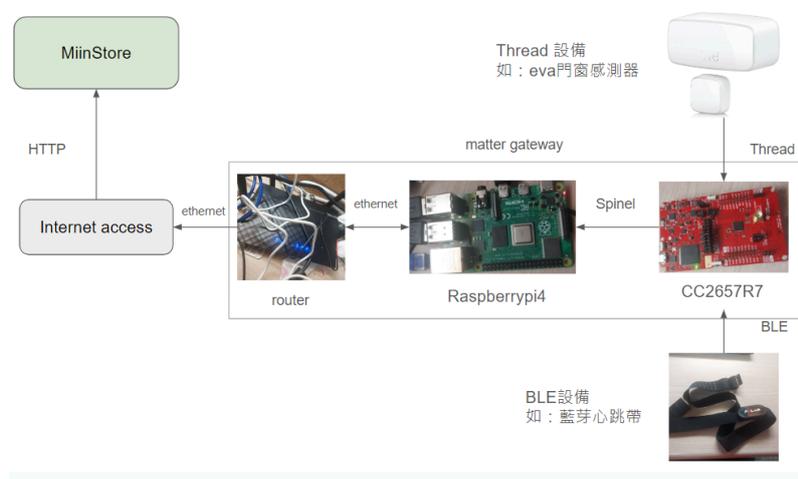
一、簡介：

MiinStore 是一個基於 IoTalk 的智慧物聯網應用開發平台，它利用 HTTP 網址形式的資料傳輸，讓使用者可以透過 GUI 輕鬆地連接各種輸入和輸出設備。當設備連接至 MiinStore 時，系統會自動處理各種感測器的數據，並允許在連接點使用 Python 進行前後處理。這種設計使得 MiinStore 能夠適應物聯網的多樣性和變化性，並且能夠整合、開放、低成本、高效率地串聯不同規格標準的物聯網應用。

隨著資訊技術的進步，Matter 連接標準在 2022 年 10 月 4 日正式發布並開放認證。

Matter 是一種基於 IPv6 的應用協定，它運行在 Wi-Fi、Ethernet、低功耗藍牙 (BLE) 和 Thread 等現有的網路技術上。Matter 提供了一個安全、可靠、可擴展的網狀網路架構，使得裝置之間的通訊更加流暢。其主要優點包括：穩定性（提供網狀網路架構，如果一個裝置的連線中斷，其他裝置可以透過其他路徑繼續通訊）、擴展性（提供訊號給同樣支援 Thread 的產品，讓產品與產品之間可以靠著 Thread 協定來通訊、連接）以及低功耗（使用電池供電的裝置可以有更長的使用時間）。

本專題的目的是實作 Matter gateway，將原本 MiinStore 物聯網平台支援 HTTP 的連線擴增為支援 Thread 等協定。這種擴展將使 MiinStore 能夠更好地整合和利用物聯網的多樣性，並提供更多的連接選項，系統架構如圖一所示。



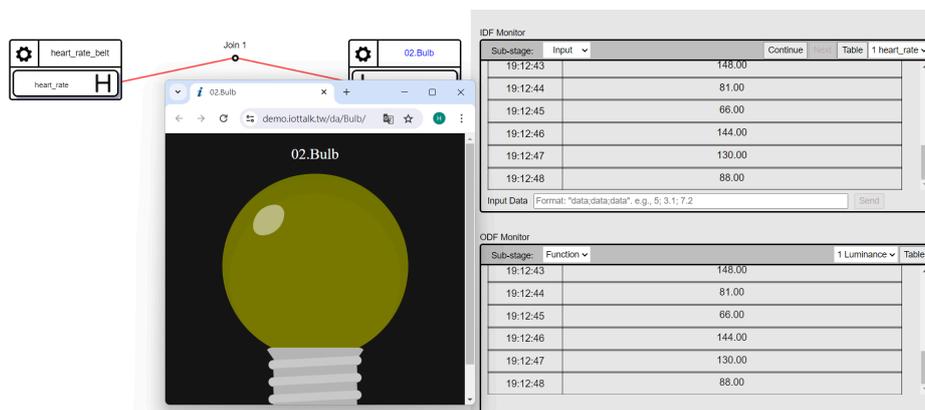
圖一、系統架構圖

二、測試結果：

本專題在建立基於MiinStore物聯網平台的Matter閘道器時，預期會遇到以下技術挑戰：

1. 協定轉換：MiinStore 平台原本支援 HTTP 連線，而 Matter 則是基於 IPv6 的應用協定，運行於 Wi-Fi、Ethernet、低功耗藍牙 (BLE) 和 Thread 等網路技術上。因此，需要實現從 HTTP 到這些協定的轉換，這可能涉及到網路協定轉換、數據格式轉換等問題。
2. 裝置相容性：Matter 的目標是讓所有物聯網裝置都能透過相同連接協議彼此相容運作¹。然而，實際上可能會遇到各種裝置的相容性問題，例如不同製造商的裝置可能有不同的實現方式、不同的硬體和軟體限制等。
3. 安全性：由於物聯網裝置通常需要連接到互聯網，因此需要確保數據的安全性和隱私性。這可能需要實現各種安全機制，例如數據加密、身份驗證等。
4. 效能和可擴展性：隨著連接的裝置數量增加，可能需要處理大量的數據和請求。因此，需要確保系統的效能和可擴展性，這可能涉及到資源管理、負載平衡等問題。

本作品將接收Matter設備的訊號與BLE心跳帶訊號，將訊號數值透過Matter gateway傳送到MiinStore平台上，並透過虛擬燈泡呈現數值的變化，如圖二所示。



圖二、MiinStore物聯網平台成果示意圖

基於 P4 的網路負載均衡實作及探討

An implementation and analysis of load balancing based on P4

指導教授：楊中平，專題成員：林建宇/李肇和，

開發工具：P4/PI/PTF/p4c/Behavioral Model (BMv2)/Mininet/Python，

測試環境：Linux Ubuntu 20.04

一、簡介：

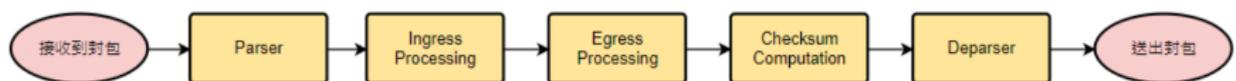
負載均衡是一種網路技術，運用於在多個伺服器或路徑之間分配網路流量，以提高資源利用率、減少響應時間並防止任何單一資源過載。這在大型數據中心、雲端計算環境和高流量網站中特別重要。

隨著全球化網路的蓬勃發展，當傳統靜態的網路拓樸無法再滿足不斷改變的網路需求，軟體定義網路（Software-Defined Networking, SDN）的出現顯得十分關鍵。SDN是一種網路管理方法，通過將網路的控制平面(control plane)和數據平面(data plane)分離，並將控制功能集中在一個或多個控制器上來進行集中管理和編寫。這使得網路更加靈活和可編寫，能夠快速適應變化的需求和條件。

SDN 透過 programmable switch 與 domain-specific language 來使 data plane 程式化。其中最廣泛使用的domain-specific language就是P4 (Programming Protocol-Independent Packet Processors)。P4是一種專為描述封包處理邏輯而設計的程序語言，使網路設備（如switch和router）可以靈活地配置和編寫，而不必依賴於硬件的固有功能。

以下為P4流程架構圖：

我們實作應用在web switch的軟體定義網路負載平衡技術，將網路流量合適的分

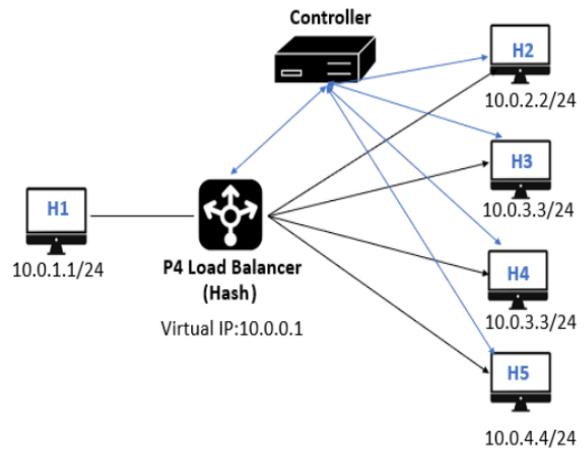


配到不同的伺服器中，提升主機或應用程式在流量高峰時的處理效能，以強化整體網路的穩定度；而當某個伺服器發生問題(需要維修更新、應用程式需要改版或負載過高)時，亦可將連線導入其他伺服器，待問題解決後再將連線重新導入，增加整體網路的靈活性。

二、測試結果：

Hash-based Load Balancing:

client傳送request給server時，會經過switch h1也就是右圖中的P4 load balancer，其作用是依據source IP, source port, destination IP, destination port, protocol(TCP or UDP)作為hash key。在產生hash key 結果後將封包分配給對應到的server。Controller會在每2秒進行一次health check來確認每台server的狀態是否正常，並適時做出調整。

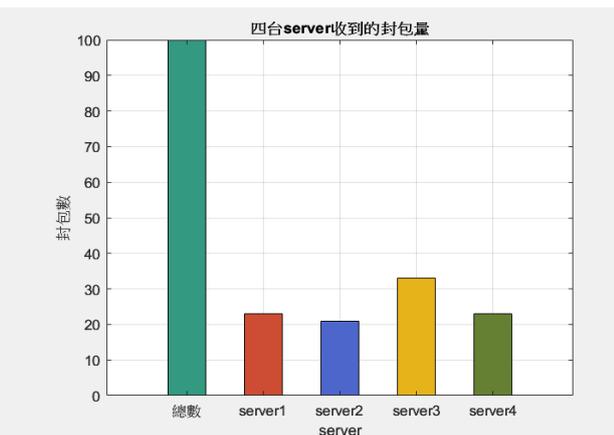
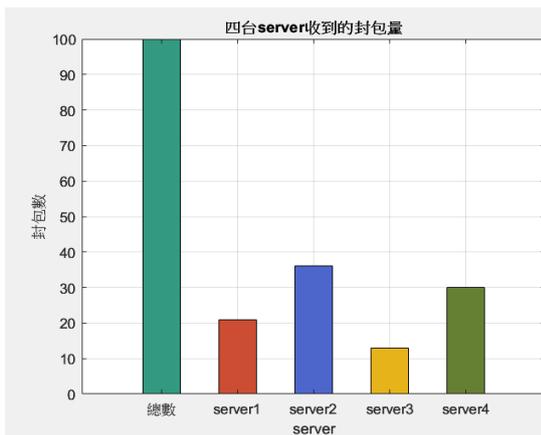


health check:藉由controller來向switch取得資訊，並依其判斷server是否有不健康的狀態。我們的設定是每台server收到的封包量，若是發生有兩台server發生封包量差距過大的情形，controller 會改變其中決定傳送封包的 flow entry，使收到最大量的server定義為unhealthy，並強迫switch 改變運送途徑，使封包傳送其他健康的 server。

我們送出100個封包來觀察各種結果

```
----- Reading tunnel counters -----
s1 MyIngress.Counter 0: 11 packets (770 bytes)
s1 MyIngress.Counter 1: 11 packets (770 bytes)
s1 MyIngress.Counter 2: 0 packets (0 bytes)
s1 MyIngress.Counter 3: 0 packets (0 bytes)
s1 MyIngress.Counter 4: 1 packets (70 bytes)
s1 MyIngress.Counter 5: 10 packets (700 bytes)
tiring_server:5 tiring_server_amount:10 easy_server:2 easy_server_amount:0

----- Reading tunnel counters -----
s1 MyIngress.Counter 0: 12 packets (840 bytes)
s1 MyIngress.Counter 1: 12 packets (840 bytes)
s1 MyIngress.Counter 2: 0 packets (0 bytes)
s1 MyIngress.Counter 3: 0 packets (0 bytes)
s1 MyIngress.Counter 4: 1 packets (70 bytes)
s1 MyIngress.Counter 5: 11 packets (770 bytes)
tiring_server:5 tiring_server_amount:11 easy_server:2 easy_server_amount:0
server 5 has too much loading!!
```



左下圖為無health check的情況，server接收的封包量不太平衡；

右下圖為有health check，可觀察到server接收的封包量平衡了許多。

Reference

1. Load Balancing Implementation in P4
https://hackmd.io/@6VE0_f7yQw-TW_66oMIYow/r1A2H6OSj
2. Load Balancing Using P4 in Software-Defined Networks
<https://jit.ndhu.edu.tw/article/viewFile/2400/2415>
3. P416 Language Specification
<https://p4.org/p4-spec/docs/P4-16-v1.0.0-spec.html#sec-external-units>

基於圖神經網路之隱私保護表格資料生成

Privacy-persevering Tabular Data Synthesis with Graph Neural Networks

指導教授：李政德

專題成員：呂宏家

開發工具：Python、Jupyter notebook

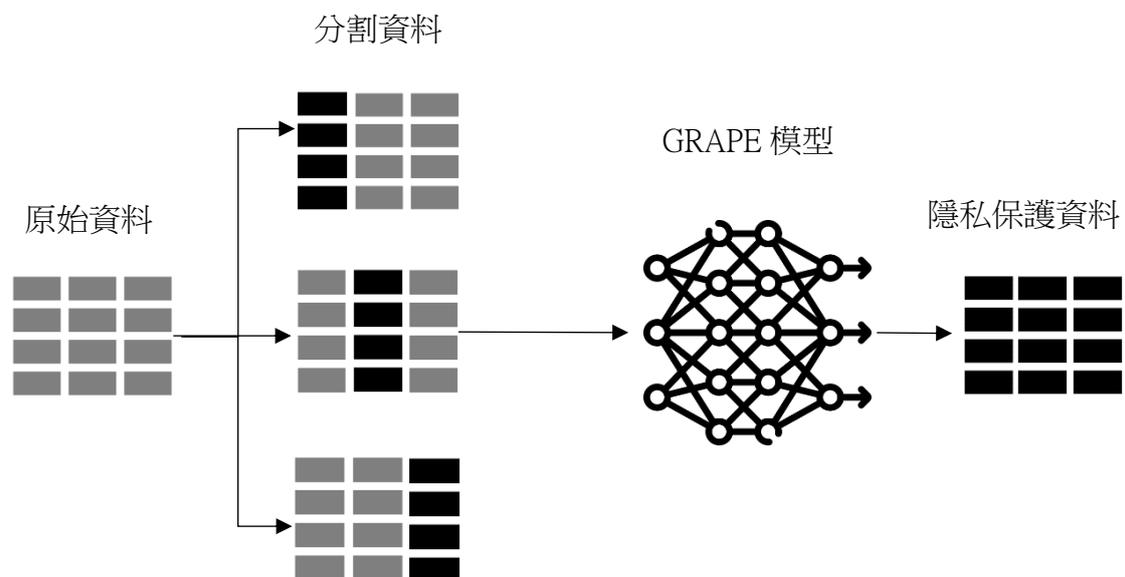
測試環境：Python 3.11.5

一、簡介：

藉由圖神經網路 GRAPE 的表格資料合成，一般來講，銀行或是醫療機構他們想要把表格的資料釋出給大家使用，但是表格的資料涉及使用者的個資隱私，銀行或醫院不想公開真實的資料，因此我們就必須設計方法來協助合成表格的資料，也就是假的資料，但資料必須符合真實資料的一些特性，來讓銀行跟醫院釋出让機器學習的模型使用。

將原始資料透過輪流遮罩不同的部分，並輸入 GRAPE 模型產生特徵補值，最後再把生成資料整合，以達到隱私保護的效果。

以下為系統架構圖：



二、測試結果：

在圖四中，我們將原始資料輸入並產生結果後，針對原始資料和新產生的資料，隨機將一個特徵做遮罩，並使用 Random Forest 來測試新的資料能否反推回原始資料的特徵值。

	0	1	2	3	4	5	6	7	8
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28.0	79.99
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28.0	61.89
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270.0	40.27
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365.0	41.05
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360.0	44.30
...
1025	276.4	116.0	90.3	179.6	8.9	870.1	768.3	28.0	44.28
1026	322.2	0.0	115.6	196.0	10.4	817.9	813.4	28.0	31.18
1027	148.5	139.4	108.6	192.7	6.1	892.4	780.0	28.0	23.70
1028	159.1	186.7	0.0	175.6	11.3	989.6	788.9	28.0	32.77
1029	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28.0	32.40

圖二：原始資料

	0	1	2	3	4	5	6	7
0	525	94	102	243	163	438	188	120
1	520	89	108	246	148	438	197	145
2	323	288	105	479	87	270	112	196
3	344	281	112	486	92	272	85	153
4	193	246	97	349	100	303	348	130
...
1025	263	232	312	331	210	181	301	137
1026	309	102	374	353	230	144	344	124
1027	153	267	132	340	173	213	349	125
1028	171	329	101	278	160	353	304	133
1029	275	226	283	375	212	169	299	132

圖三：經隱私保護後生成的資料

```
MSE of masked data from original data: 3.0887637690809067
MSE of masked data from generated data: 1177.0220239077717
```

圖四：與原始資料相比

預測式資料增強稀疏圖神經網路推薦系統

Predictive Data Augmentation for Sparse Graph Neural Networks Recommenders

指導教授：李政德

專題成員：李秉儒

開發工具：python

測試環境：window 10

一、簡介：

基於增強稀疏圖神經網路的想法，我們對不同的推薦模型進行多種資料集的測試，以觀察不同模型和資料集對於該方法的適應性和增強性。

測試方法主要有兩種：

1. 將模型預測的結果(top20)重複加入到原本的訓練集中，其中只加入預測錯誤的部分(移除和測試集中重複的部分)，以達到提高預測率的結果。
2. 減少原先訓練集的資料以獲得更為稀疏的資料，使用該資料訓練模型後並從中分析不同稀疏資料集的表現。

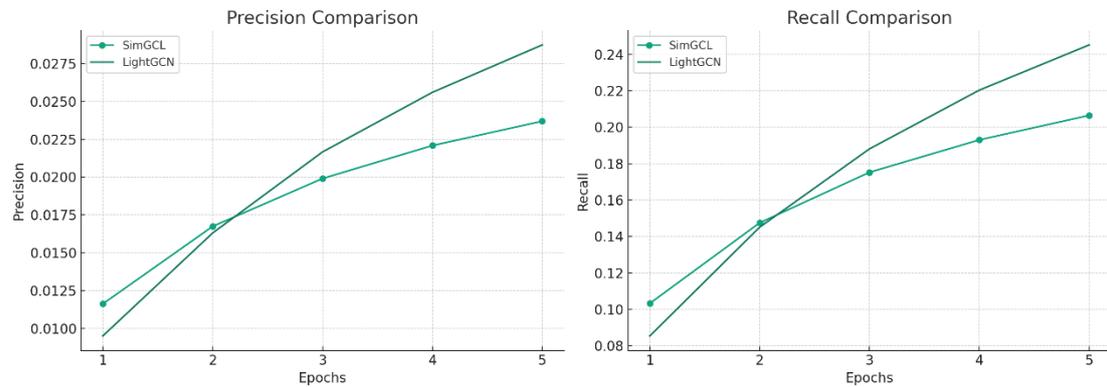
測試的模型：

1. Diffusion Recommender Model
2. LightGCN
3. SimGCL(XSimGCL)
4. LLMRec
5. UltraGCN

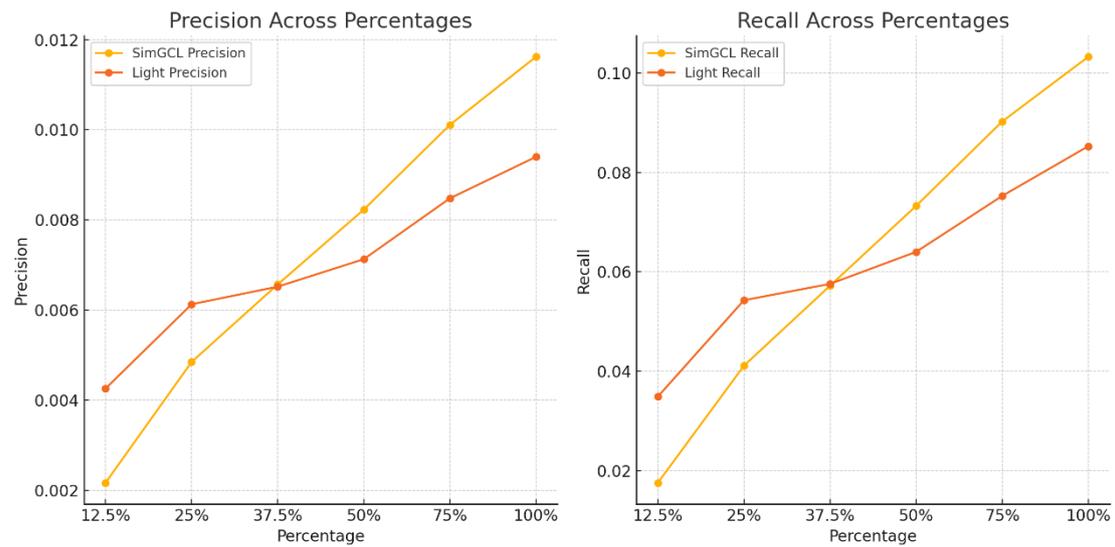
測試資料集：

1. Yelp2018
2. Netflix
3. Gowalla

二、測試結果：



上圖為 SimGCL 和 LightGCN 在 Yelp 的資料集中的表現，其中 Epochs 代表加入過幾次 Top20k 的結果(1為原始資料)。



上圖則為縮減資料集後訓練的結果(Top20)，模型和資料集一樣為 LightGCN，SimGCL 和 Yelp。

基於圖神經網路之表格模型集成學習

Tabular Ensemble Learning with Graph Neural Network

指導教授：李政德

專題成員：田容甄

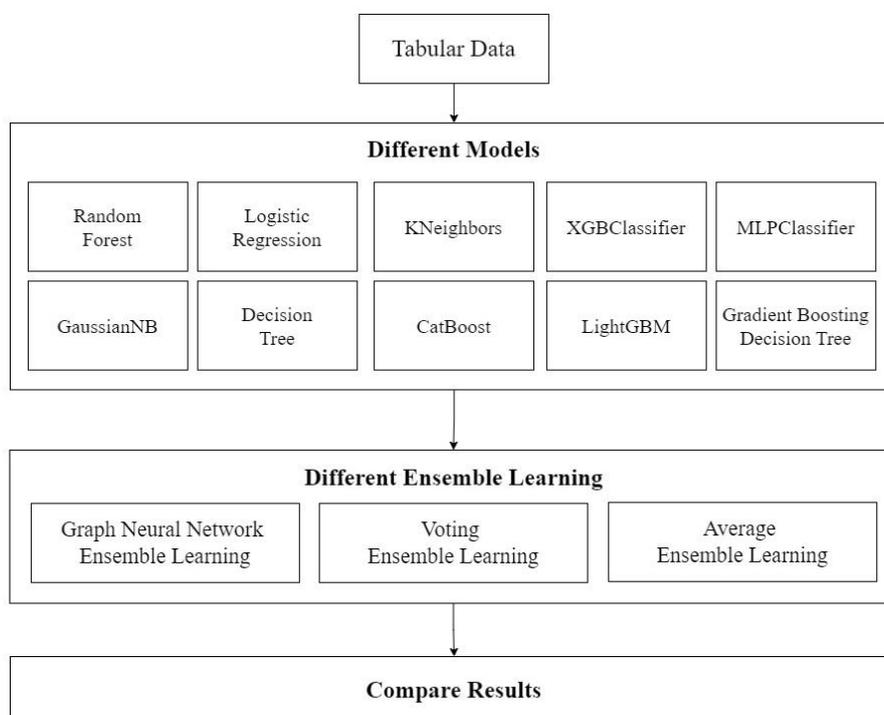
開發工具：Pytorch

測試環境：Windows 11

一、簡介：

透過集成學習可以提高單一模型的準確率，本文探討圖神經網路（Graph Neural Networks, GNN）應用於表格數據的模型集成學習。GNN 相較其他方法更擅長處理架構和規律不固定的資料，也因此本方法透過將不同模型轉換成圖形結構，利用 GNN 學習不同模型之間的關聯性，進行集成學習以融合多個模型的預測結果，減少單一模型的偏差和誤差，提高整體準確率。

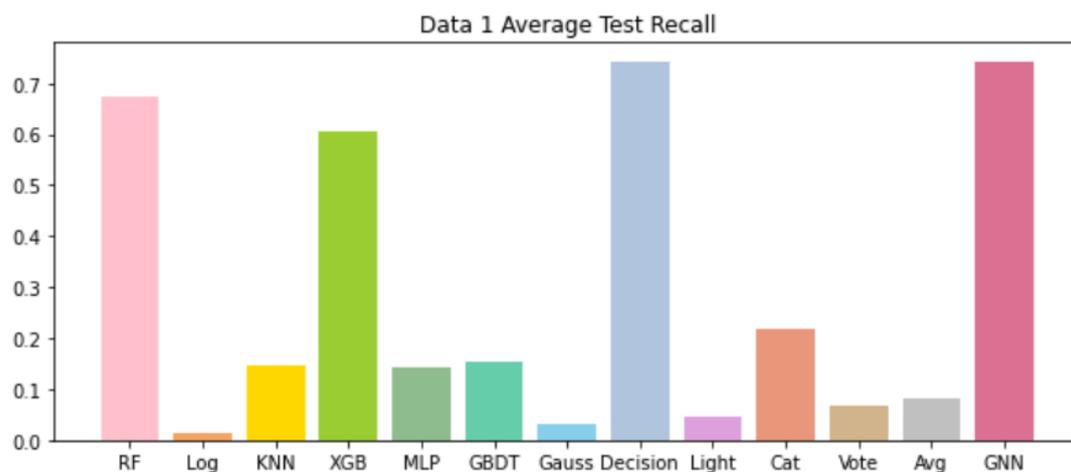
以下為程式架構圖：



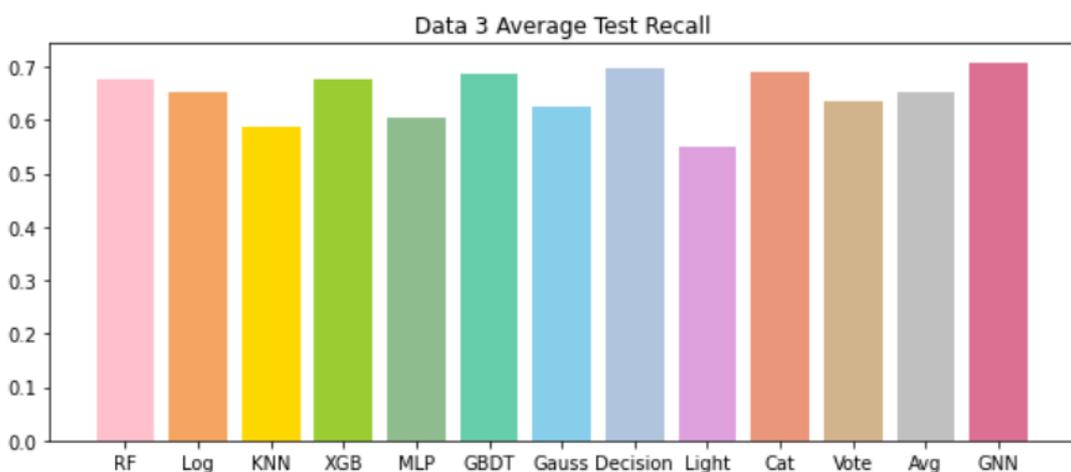
圖一：程式架構圖

二、測試結果：

本文舉2個資料集結果作為展示。下圖分別顯示10個單一模型，以及3種不同集成方式的 recall 值。可以發現透過 GNN 進行集成模型的 recall 值較其他模型高。與其他集成學習(集成投票分類(Ensemble Voting Classifier)、集成平均分類(Ensemble Averaging Classifier)) 相比，亦有較好的集成效果。



圖二：資料集1 結果圖



圖三：資料集3 結果圖

隨機特徵圖神經網路之穩健表格異常偵測

Robust Anomaly Detection with Random-Feature

Graph Neural Networks

指導教授：李政德

專題成員：唐文蔚

開發工具：Python、Pytorch

測試環境：Colab(Ubuntu)

一、簡介：

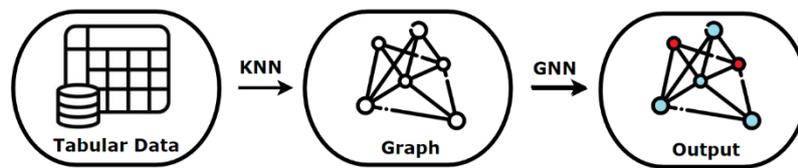
本專題旨在改進 LUNAR 模型，提升其在表格資料異常偵測中的穩健性。

LUNAR 模型使用圖機器學習方法來偵測異常資料，然而，若是訓練資料中混入異常資料就會導致準確率下降。本專題將特徵分成多個子集，分別訓練不同的模型，綜合各模型的結果來進行異常偵測，並額外引入 GraphMAE 和 MLP Layer 做特徵重建，以提升模型的穩健度和準確度。

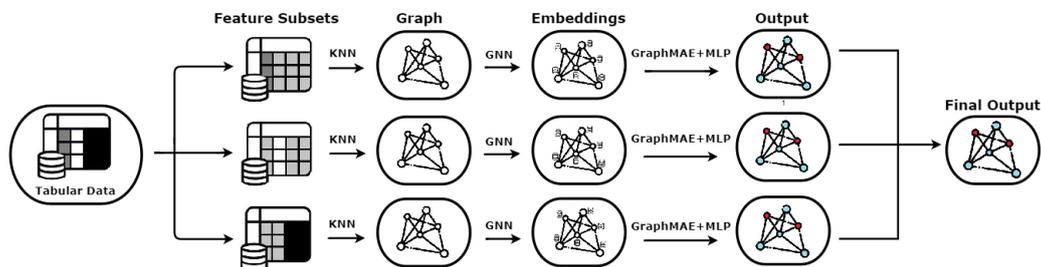
特徵分割：將表格數據中的特徵分成多個子集，針對每個特徵子集，分別訓練不同的 GNN 模型，並將各子集的結果綜合進行異常資料的判斷。

在嘗試多種分群方法後發現在隨機分群、且有 Overlap 的情況下表現最好。

GraphMAE 和 MLP：將 GNN 的輸出結果作為 Embedding 進行特徵重建，最後經過 MLP Layer 產出異常值結果。



圖一：LUNAR 模型架構



圖二：改進後的模型架構

通過將特徵分成多個子集並分別訓練多個 GNN 模型，本專題提出的改進方法提升了 LUNAR 模型在表格異常偵測中的穩健度和準確率，顯示了其在實際應用中的潛力和價值。

二、測試結果：

第一個表格為用多個 Subset 的模型和原始的模型比較，使用 Subset 後雖然沒有明顯提升，但還是有效果。

第二個表格是針對混入異常資料的表格資料，可以看到有用 Subset 的模型可以有較高的準確度。

第三個表格是針對減少 Training Data 的比較結果。

Target 3 – Origin Summary				
Dataset	LUNAR	K = 3	K = 6	K = 9
HRSS	92.17	91.06	91.76	92.06
MI-F	84.37	83.89	83.93	83.13
MI-V	96.73	97.98	99.30	99.66
OPTDIGITS	99.76	99.99	99.94	99.71
PENDIGITS	99.81	100.00	100.00	97.76
SATELLITE	85.35	95.77	97.47	96.44
SHUTTLE	99.97	99.99	99.81	99.98
THYROID	85.44	88.97	90.56	88.75

Target 3 – Target1 Summary												
Target1	P = 10			P = 30			P = 50					
	K = 3	K = 6	K = 9	K = 3	K = 6	K = 9	K = 3	K = 6	K = 9			
HRSS	87.13	92.00	93.69	93.94	78.33	83.78	86.85	88.82	70.10	81.26	83.23	83.45
MI-F	50.34	50.69	51.79	50.01	46.54	44.95	46.53	44.89	47.19	43.78	41.90	41.27
MI-V	77.15	77.66	85.70	85.38	64.10	67.18	70.72	71.83	59.85	64.64	66.68	67.43
OPTDIGITS	94.57	99.63	99.54	99.03	68.44	66.37	80.97	86.43	50.49	56.78	67.35	68.79
PENDIGITS	99.56	99.95	99.94	99.86	53.95	76.11	90.73	92.34	25.20	46.19	78.16	73.54
SATELLITE	51.05	68.73	76.33	73.95	45.64	63.48	67.14	64.34	49.94	57.39	60.39	61.83
SHUTTLE	39.34	96.93	98.49	99.53	90.22	86.95	87.14	93.44	86.11	86.13	80.18	89.51
THYROID	71.36	87.98	85.19	91.02	70.09	95.25	86.12	87.08	66.38	79.39	85.73	89.15

Target 3 – Target2 Summary												
Target1	P = 50			P = 70			P = 90					
	K = 3	K = 6	K = 9	K = 3	K = 6	K = 9	K = 3	K = 6	K = 9			
HRSS	82.49	80.40	79.75	79.16	86.66	85.75	83.53	85.90	90.39	90.09	89.93	83.83
MI-F	84.11	83.07	79.38	80.59	82.22	79.07	81.49	80.70	84.16	81.89	81.87	83.84
MI-V	95.33	98.86	99.25	98.37	95.61	99.07	98.64	99.53	96.41	98.09	99.33	99.42
OPTDIGITS	99.39	99.97	99.96	99.74	99.44	99.79	99.37	99.95	99.44	99.71	100.00	99.97
PENDIGITS	99.84	99.86	99.90	100.00	99.84	99.99	99.40	99.89	99.83	99.96	100.00	99.99
SATELLITE	84.14	93.98	95.19	96.59	84.61	97.26	94.54	97.09	86.01	96.90	95.13	95.39
SHUTTLE	99.86	99.99	99.83	99.98	99.97	99.97	99.98	99.95	99.94	99.97	99.98	99.99
THYROID	87.93	91.26	92.43	91.87	85.23	87.95	83.06	86.03	83.68	88.89	85.51	92.97

圖三：準確率比較圖

強化財經大語言模型在加密貨幣新聞上的情緒分析

Enhanced financial large language models (FinGPT) in sentiment analysis of cryptocurrency news

指導教授：高宏宇

專題成員：楊文宇、朱心慈、紀宣如

開發工具：Python, Langchain, Anaconda

測試環境：Windows 11, Ubuntu 22.04.4 LTS

一、簡介：

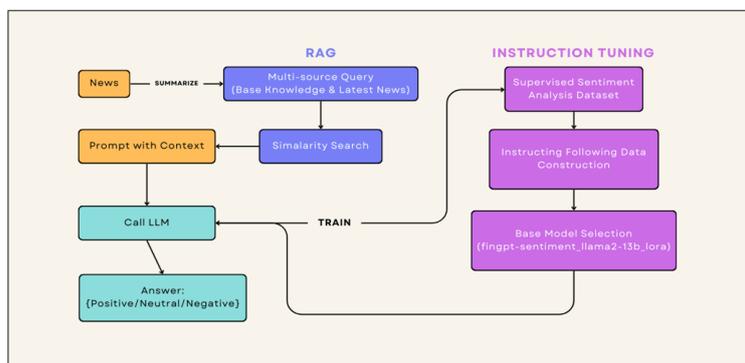
本專題的目標是提升開源大型語言模型（LLM）在加密貨幣情緒分析上的表現，具體通過檢索增強生成（Retrieval-Augmented Generation，RAG）技術和微調（Fine-tuning）技術來實現。選用的模型為開源 LLM，如 FinGPT。

以下為系統架構圖：

檢索增強生成（RAG）可以先檢索與特定加密貨幣相關的最新新聞和社交媒體內容，比對 cosine 相似度，檢索出最相似的內容放入 prompt。

微調（Instruction-tuning）：我們蒐集大量加密貨幣相關的情緒標註數據，包括新聞報導、Twitter 貼文等等，使用 QLoRA 方式對 fingpt 進行微調，提升其在加密貨幣情緒分析上的準確性和敏感度。

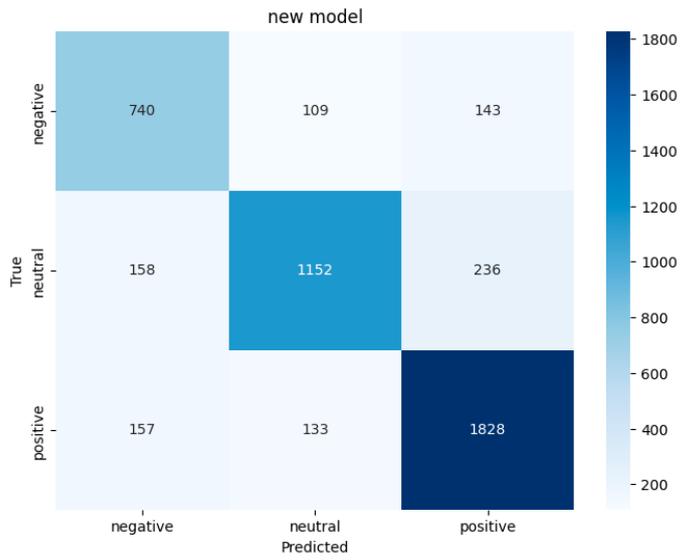
將要分析的文章輸入 LLM，會分析並回傳 Postive/Neutral/Negative。



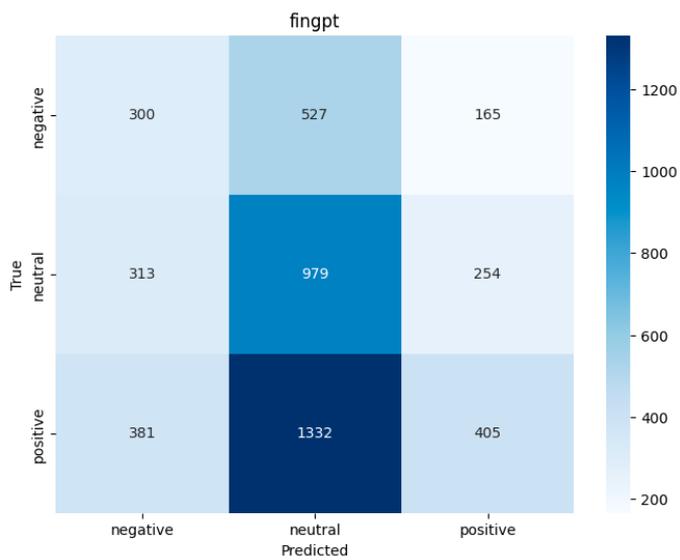
二、測試結果：

使用4600筆 Crypto News 作為 testdata，在加密貨幣領域準確率明顯提升。

Accuracy: 0.7989690721649485



Accuracy: 0.36125429553264604



太陽能發電量異常偵測平臺

Solar Power Anamoly Detection Platform

指導教授：莊坤達

專題成員：王柏凱

開發工具：PyTorch, SckitLearn, FastAPI

測試環境：Linux Mint 21.3

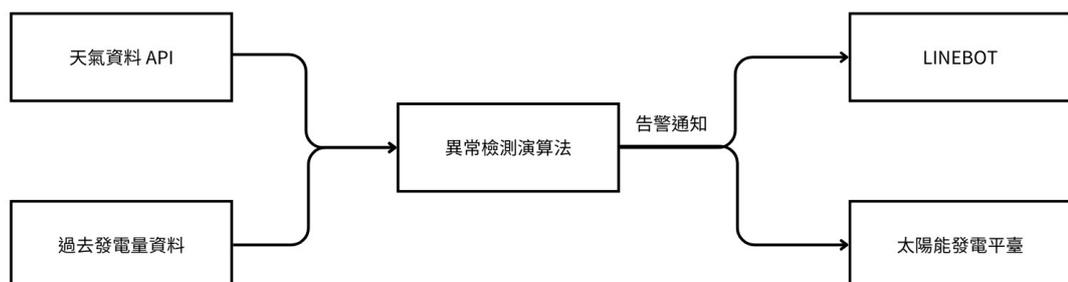
一、簡介：

為了確保太陽能發電系統的高效運行，對其運行狀況進行實時監控與異常檢測顯得尤為重要。基於此背景，本專題開發了一個太陽能發電異常檢測平台，旨在通過資料探勘技術，提高太陽能發電效率並降低維護成本。

本平台以其後端系統為核心，利用 DBSCAN、Isolation Forest 和 LSTM 等多種算法，對前一天的發電數據進行深入分析，以檢測是否存在發電異常。

一旦系統檢測到異常，它將自動通知用戶，使得用戶可以及時採取相應措施，從而節省成本並延長設備的使用壽命。

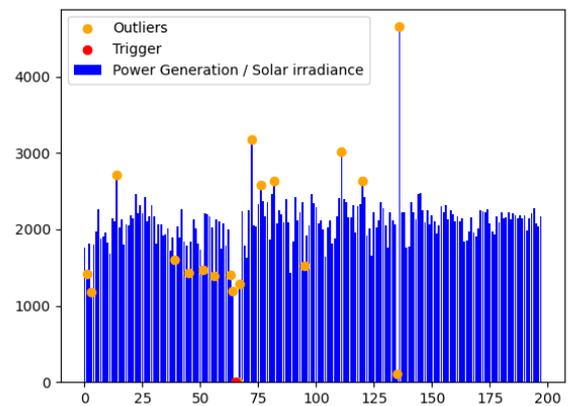
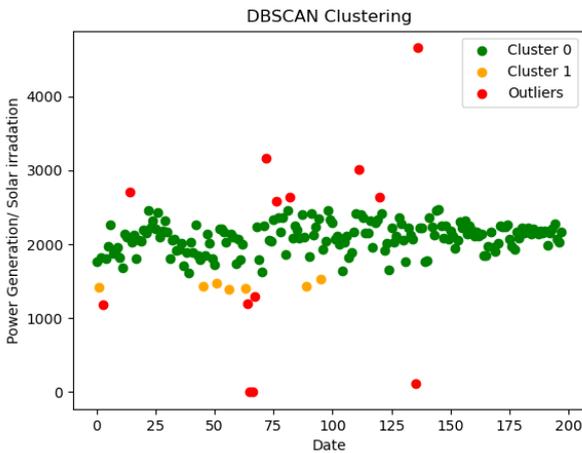
系統架構圖：



二、測試結果：

經過計算後，能得出一般資料與異常點，而如果連續幾天異常，則會發出告警通知。要連續異常再發出告警的原因是為了避免偽陽性的狀況，例如裝置前一天離線，導致隔天資料異常過大，或是 Isolation Forest 固定數量的異常點。

比較各方法過後，能發現各方法有自身的優劣，例如：如果 Isolation Forest 的異常比例設定不當，容易有偽陽性的異常；而 DBSCAN 在資料量大的時候容易偵測不到異常值。



DBSCAN 分群

異常點（橘）與告警（紅）

儘管當前的異常檢測平台已經能夠有效識別故障，但在長期使用下，我們可能會遇到太陽能板性能逐漸下降的問題：隨著使用時間的增長，太陽能板的效率可能會逐漸降低。這種緩慢的性能衰退可能不會立即被當作異常處理，但是可以透過與去年資料比較來處理。

資料管線的高可用性

High availability of data pipelines

指導教授：蕭宏章教授

專題成員：王偉同

開發工具：Minikube、InfluxDB

測試環境：Windows 10

一、簡介：

（一）題目概述：

有時基於生產環境或數據處理的要求而需要持續且穩定的資料串流，過程中涉及資料的產生、採集、儲存與使用，此外系統中的各個環節也隨時可能發生異常，有重新啟動的需求，因此如何在減少人力介入的前提下使系統恢復正常，並減少對資料使用端的影響是一個課題。

我們將資料串流分為 A、B、C、D 四個部分，其中 A 為資料來源端，可能是第三方的軟硬體設備或程式，B 為從資料來源角度看的客戶端，可能是採集資料的工具，會將資料寫入 C，也就是資料庫，一般情況是外部的儲存空間，最後 D 則是客戶端的服務程式，從 C 取得指定的資料。

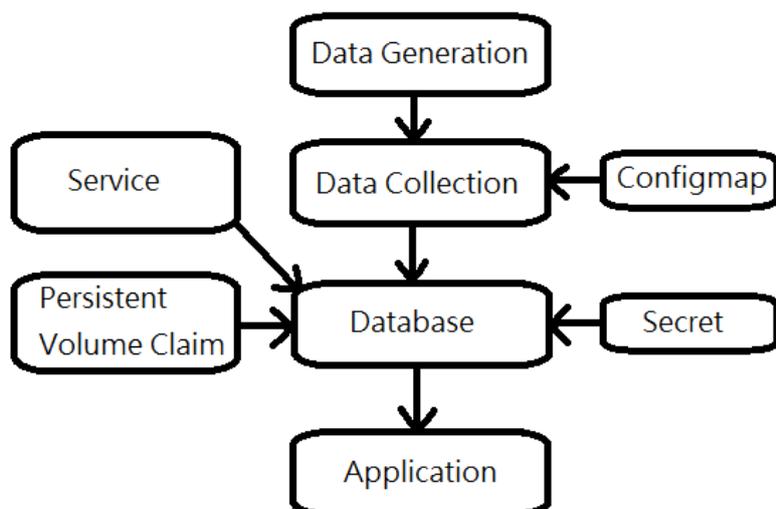
宗旨是提升可用性，定義為 $1-X/Y$ ，其中 X 為系統無法讀取的時間，而 Y 為一年 365 天，因此 $1-X/Y$ 越接近 100% 越好。由於使用者可能隨時存取資料庫，我們需要模擬資料串留有持續讀寫的情境，也就是持續對資料庫發送讀寫要求，其中包含 B 與 D 兩個部分的需求，並盡量降低異常情況發生時對使用者體驗的影響。

（二）實作過程：

我們將系統建立在名為 Kubernetes（K8s）的容器資源調度平台上，K8s 將程式運行在名為 containers 的容器中，並將這些容器化的程式包裝在名為 Pod 的運行單位裡。當運作中的 Pod 發生異常情況，K8s 會將該 Pod 重新啟動，以達成減少人為介入並提高可用性的目的。在實際情境中 Pod 可能由於任何形式的軟硬體問題而發生異常，如資源不足、記憶體已滿就是可能的原因。

A 與 C 在一般情況下會是第三方的軟硬體，因此可以假設不會發生異常，不過在這個專題中 C 也運作在 K8s 的架構上。A 作為資料來源可以相當多樣化，目前使用的是分配給 K8s 平台的 CPU 資源數據。B 使用了名為 Telegraf 的資料採集軟體，並運作在 Pod 當中，由 K8s 負責管理，並以每分鐘為單位持續採集 A 的數據送入 C 當中。C 使用了名為 InfluxDB 的資料庫軟體，運作在 Pod 當中，由 K8s 負責管理，並負責保存帶有時間戳記的時序數據。D 是客戶端服務程式，與 A 相同可以相當多樣化，在這個專題中目前暫定為 Chronograf，用於展示數據，同樣運作在 Pod 中並由 K8s 管理。

以下為系統架構圖：



二、測試結果：

如圖所示，當前有兩個正在運作中的 Pod，名稱分別為 influx-statefulset-0 與 telegraf-4xjdt，前者運作的是 InfluxDB 而後者是 Telegraf。

influx-statefulset-0 此時正在正常運行中。

NAME	READY	STATUS	RESTARTS	AGE
influx-statefulset-0	1/1	Running	0	129m
telegraf-4xjdt	1/1	Running	0	50m

手動輸入指令使 influx-statefulset-0 重新啟動，模擬實際情況下因容器出現異常而被 K8s 重新啟動的情境。

```
kubectl rollout restart statefulset/influx-statefulset
```

重啟中……

NAME	READY	STATUS	RESTARTS	AGE
influx-statefulset-0	1/1	Running	0	129m
telegraf-gbndc	0/1	ContainerCreating	0	1s

數秒後可見重啟完成。

NAME	READY	STATUS	RESTARTS	AGE
influx-statefulset-0	1/1	Running	0	129m
telegraf-gbndc	1/1	Running	0	5s

由於重啟資料庫所需的時間並不長，完成後資料庫仍能正常讀寫，對於使用者來說在資料庫重啟過程中雖然無法正常取得資料，不過短暫修復後對使用者便能如常繼續操作，且過程不須人力介入，由此將對使用者體驗的影響降至最低。

智慧自動化視力檢查

Intelligent Automated Vision Screening

指導教授：王士豪 教授

專題成員：何庭寬、王譔評、施伯昌、卓翰威

開發工具：Python 3.11.5、PyQt5、OpenCV、
MediaPipe, Qt-Material

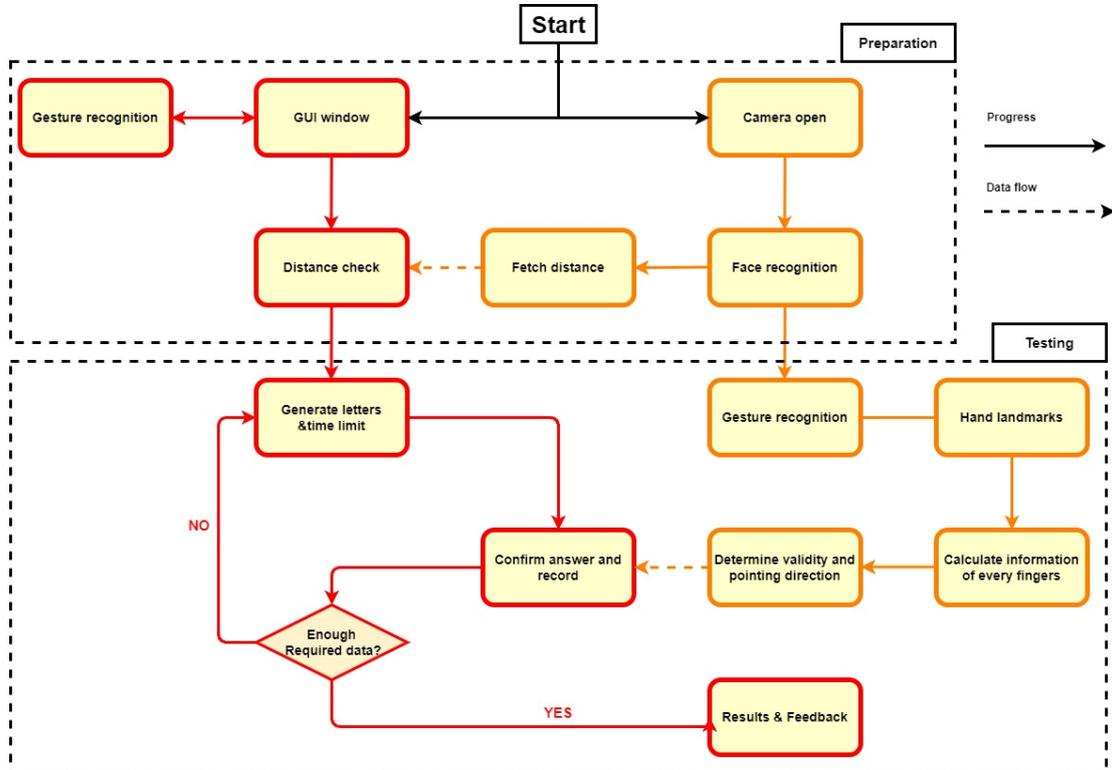
測試環境：Windows 10

一、簡介：

在新冠疫情爆發之後，日常生活中的許多事物改變了原先樣貌，開始將「遠距」的想法融入其中。除此之外，人們對於健康的意識也日漸升高，因此我們希望將「保健」及「遠距」結合，實現出了自動化視力檢查系統，幫助大家能夠在沒有專業醫師及器具的協助下也能初步評估自己當前的視力狀況。

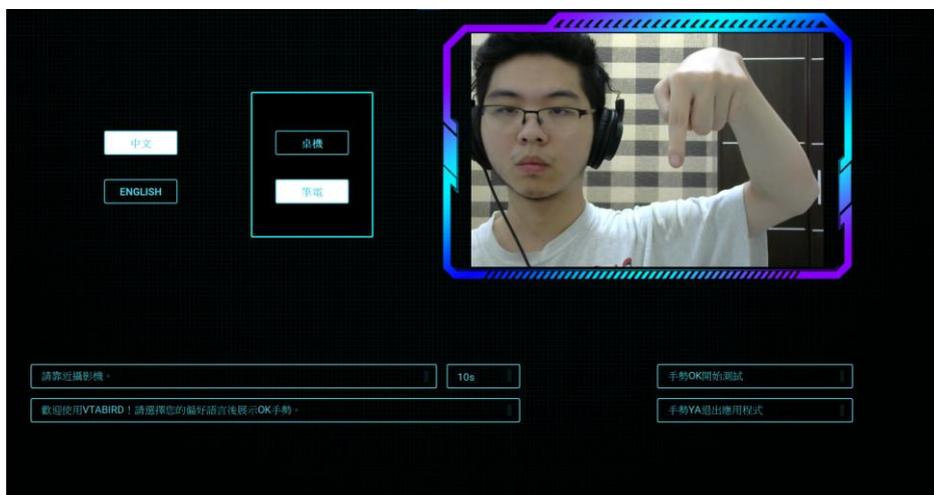
我們使用 OpenCV 進行攝影機操作，輔以 Mediapipe 臉部辨識與偵測手部座標進行關鍵點位的取值，進而判斷使用者和設備間的相對距離以及後續的視力檢測操作。

以下為系統架構圖：

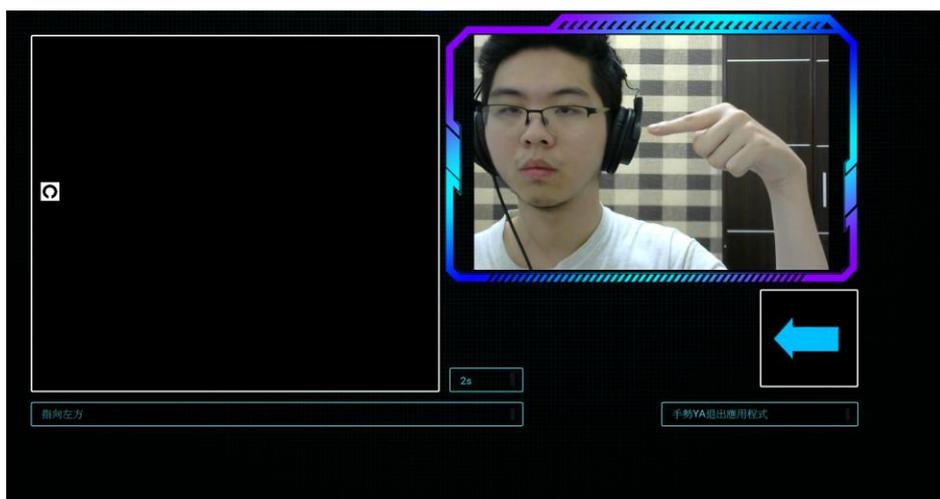


二、測試結果：

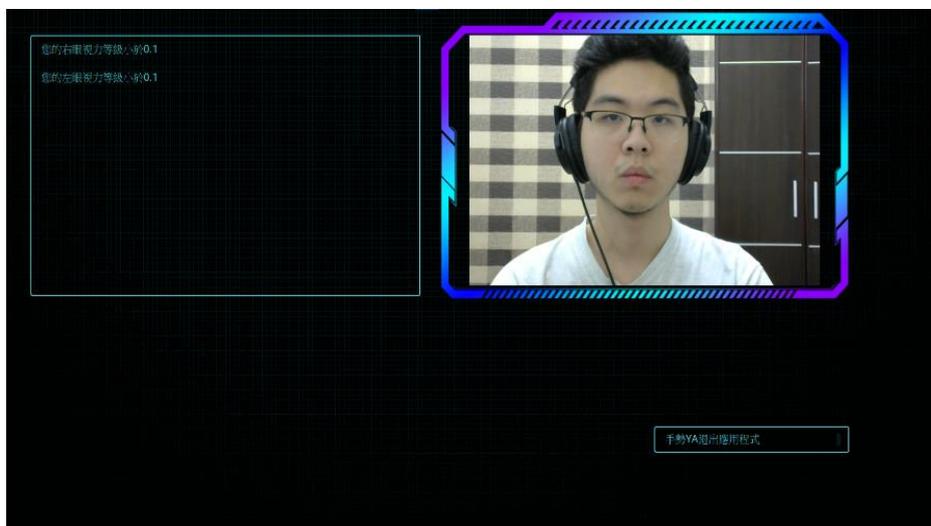
(開場畫面) 可選擇語言與裝置，手勢辨識選單



(測試畫面)



(結果畫面)



AI 虛擬實況主

AI Virtual Streamer

指導教授：王宏鏞

專題成員：曾駿馳

開發工具：Python、LangChain、VTubeStudio

測試環境：Ubuntu 22.04.4 LTS

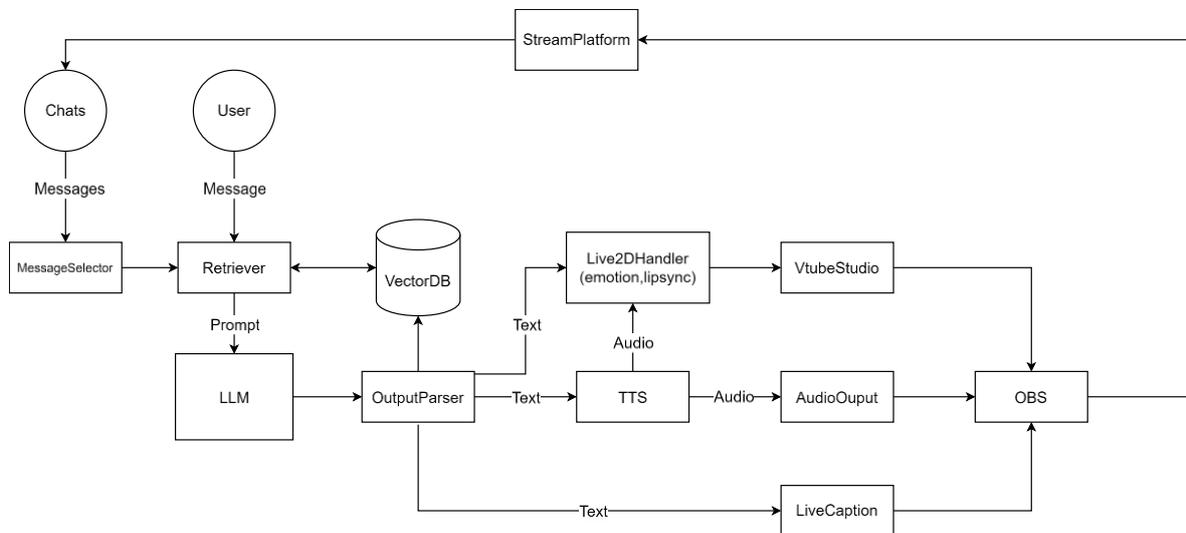
一、簡介：

虛擬實況主，是一種利用動作捕捉技術，透過虛擬形象與觀眾即時交流的新興職業。虛擬形象好比一層面紗，不露出真實樣貌的同時，也增添了一股神祕感，這種介於虛實之間的互動模式，成為一種不同於以往的大眾娛樂選擇。但是，儘管稱作虛擬實況主，背後仍然是由真人所扮演，需要承受實況主高工時所產生壓力和疲勞。

另一方面，生成式語言模型在近年蓬勃發展，各界都在積極開發其相關應用的可能性。不過，人工智慧幻覺 (Hallucinate) 現象的存在，成為其應用上所面臨的一大難題。

AI 虛擬實況主，便是試圖融合上述兩者的優缺點，透過生成式模型來驅動虛擬形象，消除了真人扮演所帶來的缺點，於此同時，即便出現幻覺現象這種胡說八道的情況，也無傷大雅，甚至可以當作一種娛樂效果。本專題結合近年來生成式模型的相關技術、應用，包括提示工程、檢索增強生成 (Retrieval-Augmented Generation, RAG)、語音合成等，打造可以與觀眾即時互動的 AI 虛擬實況主系統。

以下為系統架構圖：



圖一、系統架構圖

二、測試結果：



圖二、互動控制介面



圖三、輸出結果

針對桌球初學者正手揮拍姿勢分析與3D 可視化

Forehand stroke posture analysis &

3D visualization for table tennis beginners

指導教授：朱威達

專題成員：林宸顛

開發工具：Pytorch、C++、

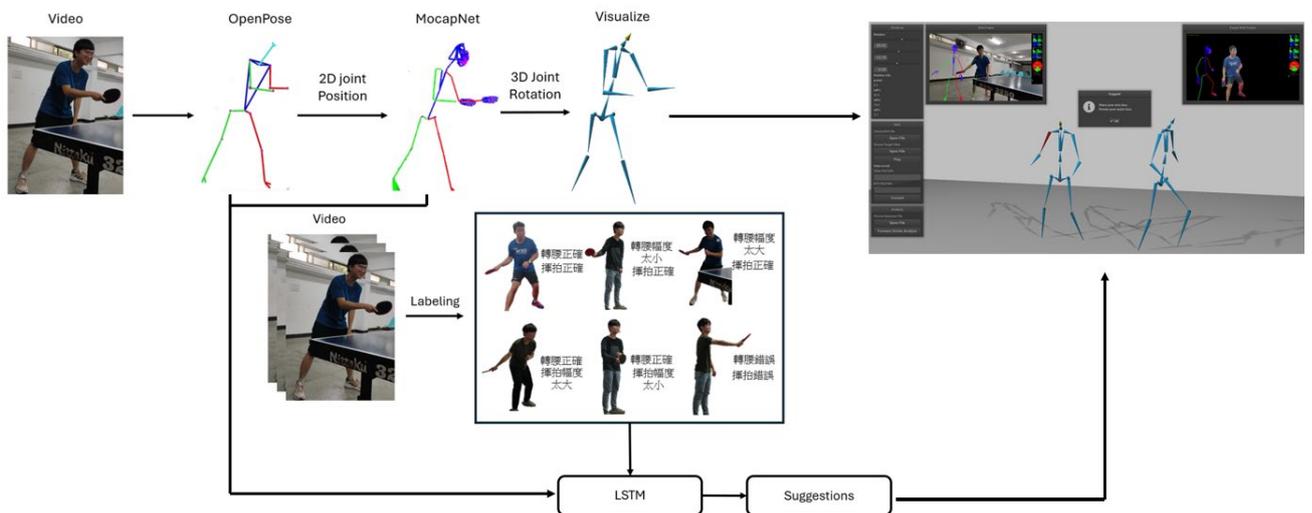
OpenGL、OpenPose、MocapNet

測試環境：Ubuntu 22.04 LTS

一、簡介：

我們打造了一個桌球正手揮拍3D 姿勢糾正的系統，利用 OpenPose 和 MocapNet 讓初學者憑著單個錄影設備就能夠看到自己在3D 空間的骨架，並和專家影像及骨架做對照以改進動作。我們利用 LSTM 進行動作異常狀況的分類，依據分類結果提出改進建議，以更進一步不單單透過影片調整姿勢，而是從3D 空間、二維對照與 LSTM 依據特徵提出建議，協助初學者學習。

系統架構：

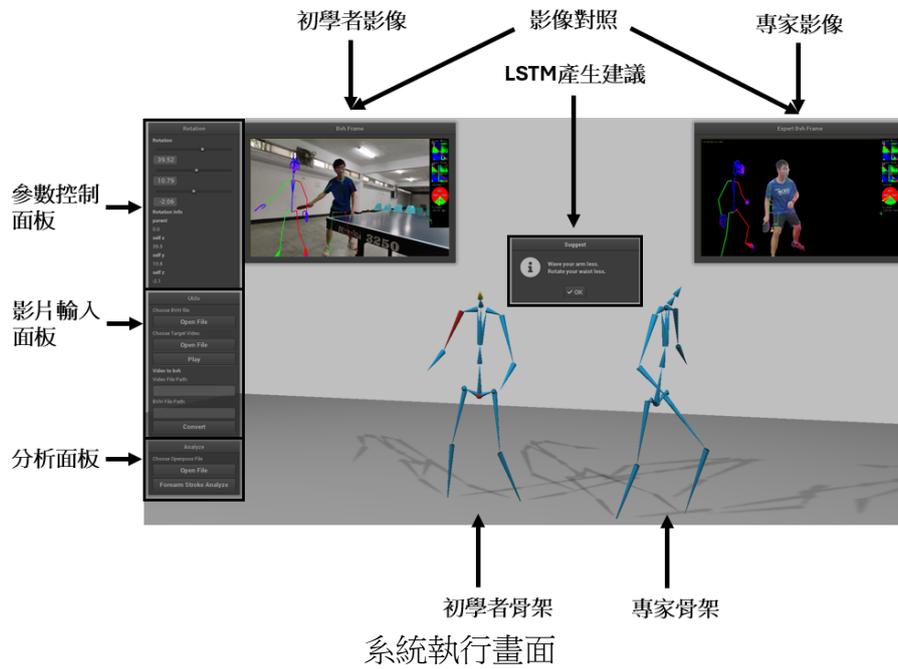


運作流程：

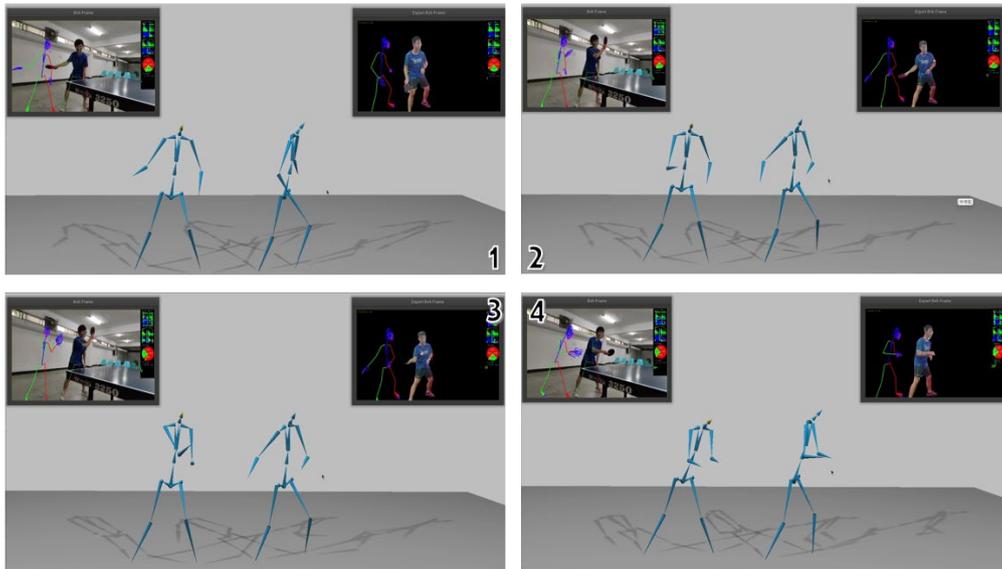
1. 利用 OpenPose 進行人物關節的定位，找到關節在每幀2D 影像的座標，並作為 MocapNet 的輸入，產生隨時間變化的3D 關節旋轉量。
2. 利用 Forward Kinematics 對其所選定的骨架支段進行隨時間變化的3D 空間旋轉與定位，達成人物正手揮拍動態。
3. 將 OpenPose 和 MocapNet 產生的2D 關節位置與3D 關節旋轉量組成的時間序列輸入到 LSTM，LSTM 根據其特徵進行異常狀況分類，並依據 LSTM

分類產生建議。

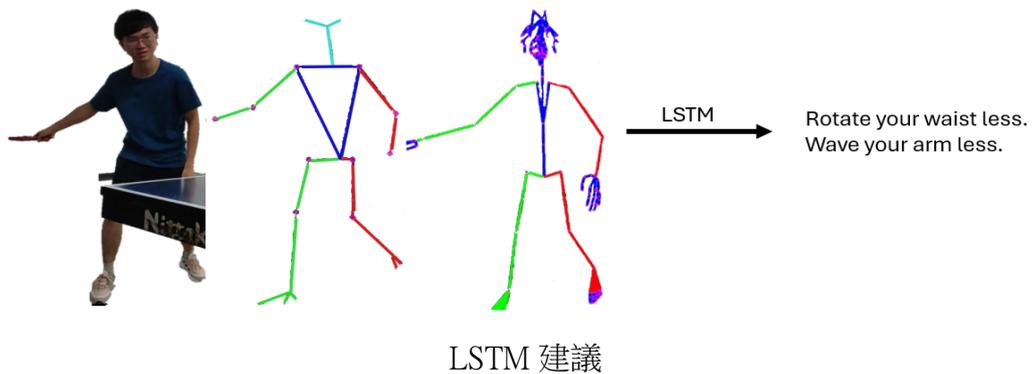
二、測試結果：



系統執行畫面



骨架連續動作



基於足球與球員位置進行足球事件偵測

Football Event Detection Based on Position of Ball and Players

指導教授: 朱威達

專題成員: 李宇翔

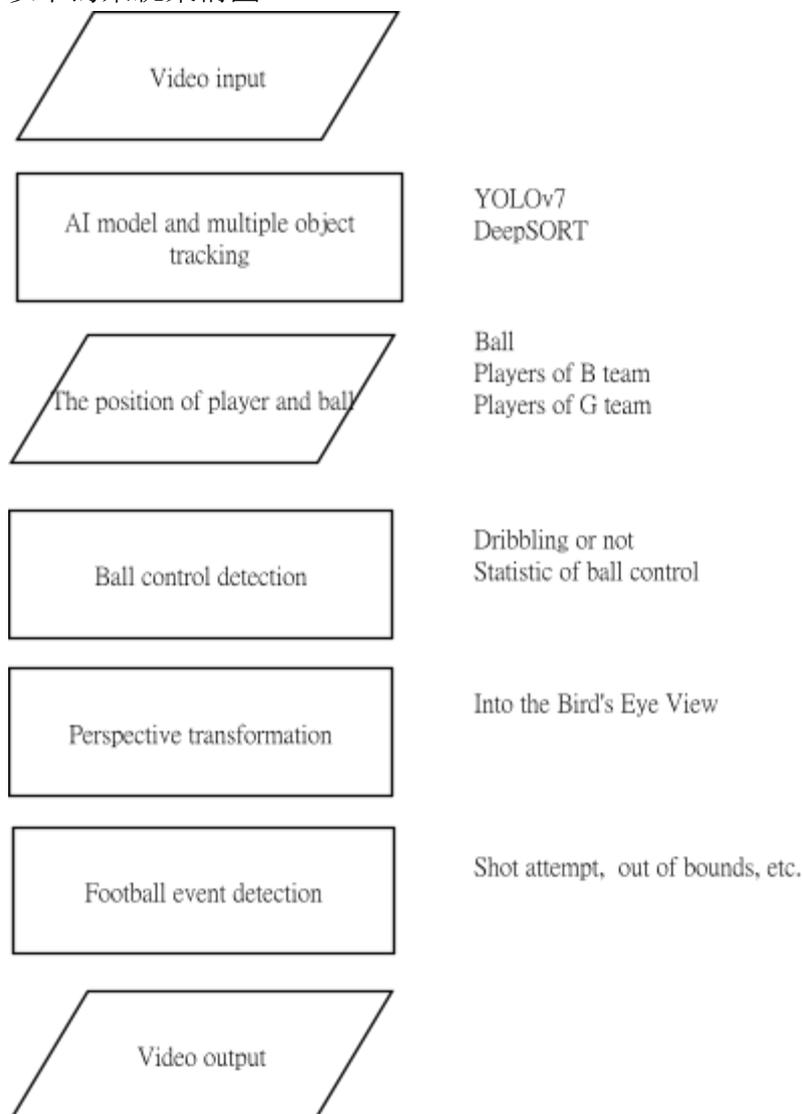
開發工具: Python、PyTorch

測試環境: Windows 11、MinGW

一、簡介:

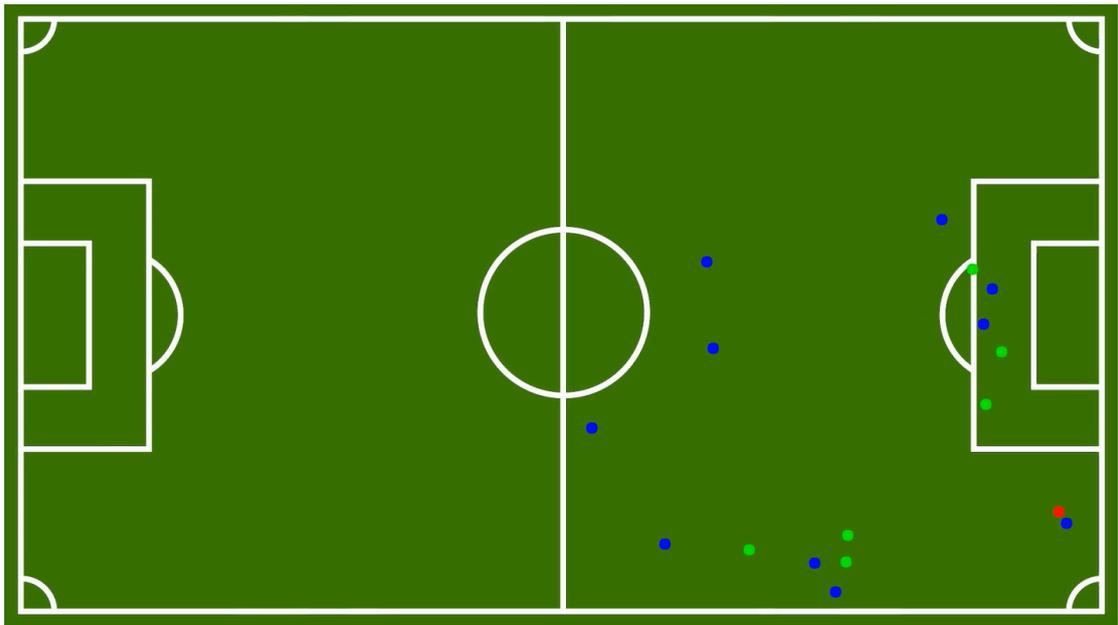
In this work, we want to extract football and player objects, detect events using AI models, and also transform the view into the top view (perspective transformation).

以下為系統架構圖:



二、測試結果:

Screenshots of the source video and outputs as follows.



水滴漫遊：跑酷解謎新體驗

Waterdrop Roaming: A New Parkour Puzzle Experience

專題教授：何建忠

專題成員：蔡誠穎

開發工具：Unity, Blender

測試環境：Android(版本9,14), IOS(版本17)

一、簡介：

市面上的遊戲種類繁多，包括戰略遊戲、動作遊戲、體育遊戲、格鬥遊戲、音樂遊戲、競速遊戲、網頁遊戲和角色扮演遊戲等。某天，我偶然接觸到一款與眾不同的遊戲——《光·遇》。這款遊戲沒有戰鬥、沒有成就，只有獨特的美術風格、富含情感的故事，以及別具一格的換裝和社交體驗。相比之下，它更像是一件驚人的藝術作品。

受此啟發，我也嘗試開發了一款與市面上遊戲不同的原創單機解謎跑酷手遊。這款遊戲的程式、美術、音樂均為原創，僅有音效是外購的。在夜晚的自然世界中，一個小水珠在瀑布底下被賦予了生命。它渴望探索這個世界，點亮每個角落。玩家將操控這個小水珠，在水邊探索，解開各種謎題，逐步點亮這個世界。遊戲中沒有戰鬥和成就系統，旨在讓玩家放鬆身心，享受遊戲的過程。操作簡單，適合所有年齡段的玩家。若你喜歡這種風格，歡迎下載體驗。遊戲時長約10分鐘，祝你有美好的遊戲體驗。

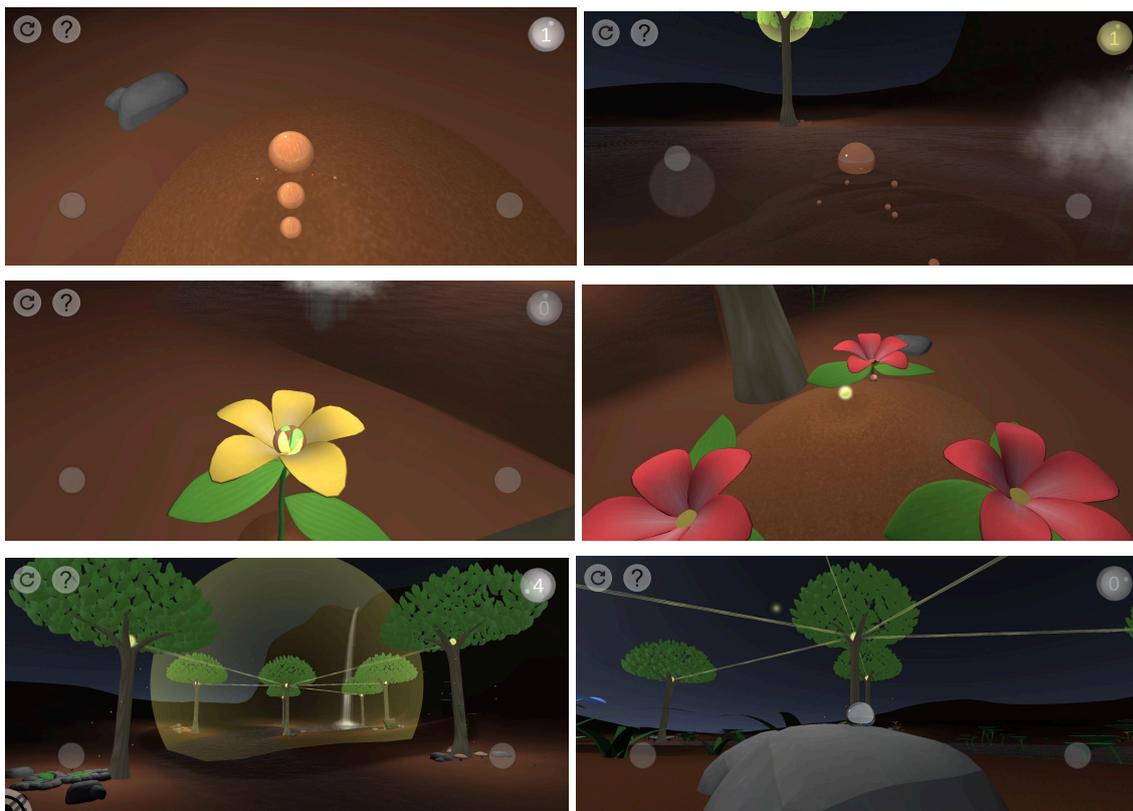
這是我第一次製作遊戲，開發過程中經歷了許多嘗試。初期我選擇了Godot引擎，它是一個輕量化且簡單易用的遊戲引擎，整個引擎僅約127MB。然而，數月後我發現Godot的3D效能無法達到我的標準，無法在最新的手機上穩定運行到30fps。因此，我轉換到Unity引擎，重建了整個遊戲。

另一個較大的嘗試是讓遊戲具備多人連線功能。我希望三五好友能夠一起遊玩，於是研究了Photon Fusion(一個多人連線框架)，使不同玩家能夠各自操縱水滴並互相幫助。然而，由於時間壓力，在專題展臨近時，我尚未完成聊天功能及架設正式的Photon Fusion伺服器(免費版最多僅允許20人同時連線)。最終，我決定將聯機功能移除，將遊戲轉為單機模式。

遊戲的製作涉及音樂、美術、程式和整體規劃，這些幾乎是每個遊戲都不可或缺的部分。製作一款遊戲需要學習很多知識，並將各部分靈活運用。例如，製作一個水面需要一個有動態紋理的平面，並根據視角不同調整透明度，這是一項相對複雜的工作。通過這次製作遊戲，我更加了解自己的技術水準，也深刻體會到即使是製作一款不大的手機遊戲，仍有諸多細節需要處理。希望未來能夠找到不同專業的人才一起合作，創作出更多優秀的遊戲。

二、測試結果：

以下是部分遊戲畫面，因採用黑白列印，想完整觀看的朋友請到攤位掃碼體驗。



基於對抗學習與模仿學習技術的 AI 行為與人工智慧擴散模型的圖像生成 樣條曲線的 3D 軌道生成 實時反饋的視覺特效設計

Intelligent 3D Game design based on diffusion, GAILearning, 3D spline curve generation and visual effects design

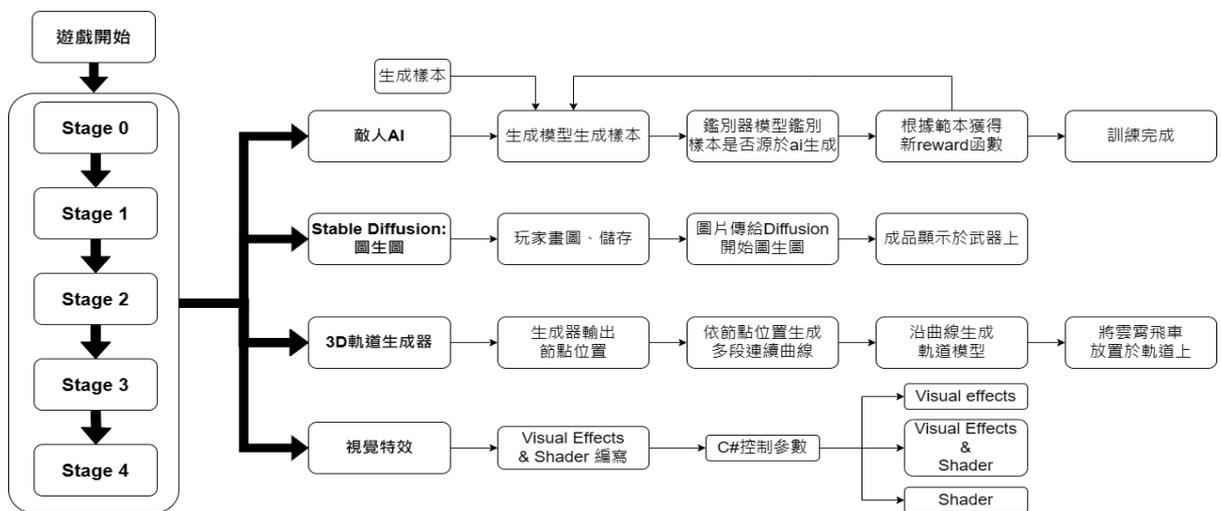
指導教授：李同益
 專題成員：吳沛儒、張庭瑄、李秋霖、曾紹宇
 開發工具：Unity2022.3.8f1、SteamVR
 測試環境：Windows 11、HTC Vive

一、簡介：

隨著近年來虛擬實境與人工智慧技術的發展與成熟，VR 遊戲快速地在玩家間蔚為風潮。只要穿戴設備，玩家不需出門亦可體驗宛如實際置身於虛擬世界的沉浸感。因此，我們結合系上所學，設計並實作出基於虛擬實境系統的遊戲。

在遊戲中有四個關卡，每個關卡各有不同的設計。在遊戲進行時，玩家可以透過圖像生成 AI 來生成自己的武器外觀，用以挑戰智慧化的 AI 敵人，享受自由創建雲霄飛車軌道，沉浸式體驗華麗視覺特效。此外，VR 遊戲還可以帶來與一般 3D 遊戲不同的感官刺激，讓玩家可以身歷其境，享受與眾不同的遊玩樂趣。

遊戲系統架構圖：



二、測試結果：

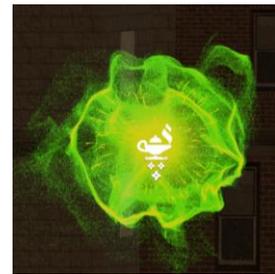
實時反饋的視覺特效設計：

透過編寫 VFX 來取代原本在 CPU 上執行的 particle system，讓粒子效果可運行於 GPU，藉以實現遊戲效能優化以及更細節的粒子效果表現，這使得遊戲特效設計上比起原本的 particle system 更能產生華麗且粒子數更多的特效。此外，除了在 Shader 本身如 Texture、UV ... 等基底上加入如 Noise、Offset 的設計，我們也將 Shader 與 VFX 進行配合，讓遊戲擁有更加複雜以及多樣化的視覺特效。

Visual Effects & Shader (怪物溶解以及噴煙效果):

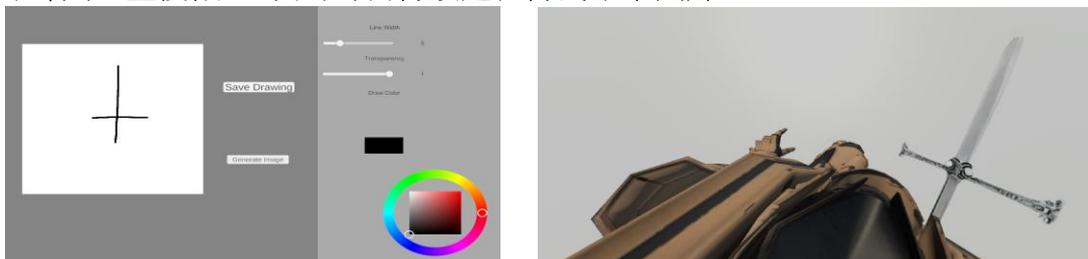


Visual Effects (粒子球):



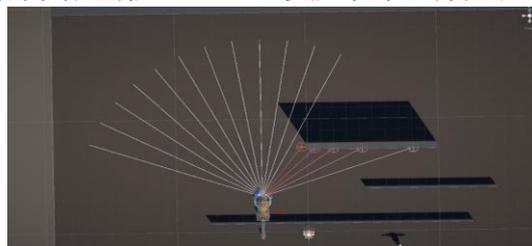
人工智慧擴散模型的圖像生成：

將玩家繪製的圖片在經過 DDIM sampler 加噪之後，還要去噪才能得到結果圖片。其中使用了 CLIP 模型來將文字對應到相應圖像，把輸入的字串轉成有文字訊息的向量，用以在 noise predictor 中取得降噪用的預測噪聲圖，將加上噪聲的原圖減去預測噪聲圖，重複做 20 次（可自行設定）得到結果圖片。



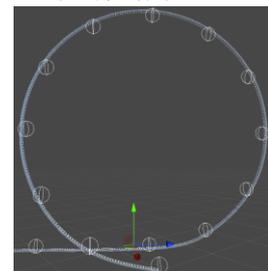
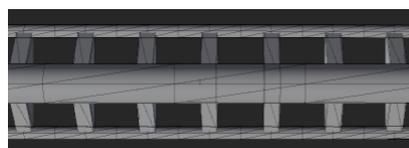
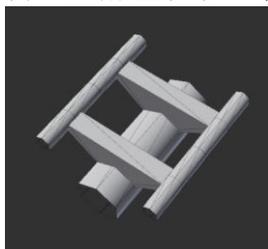
基於強化學習技術的生成對抗模仿學習行為生成：

為了達到人性化的互動反饋動作使用以生成策略為生成器的對抗學習，在 Unity 中先錄製範例的行動反應，透過射線來取得與環境的關係變數，並依此為目標分布，讓策略產生器在訓練中依照 reward 函數生成敵人行動，因為直接判斷行動與範本間的 loss 消耗較大，故直接採用判別器來判斷此行動是否為 AI 生成，反饋給產生器 reward 函數，並讓兩個模型反覆比較訓練，判別器最終得出最小 loss，可獲得最佳 reward 的策略函數，並在遊戲中做出較 rule-base 更人性化的行為，也減少行為克隆訓練時間。



樣條曲線的 3D 軌道生成：

軌道生成主要為兩個部分，依指定節點生成曲線，以及沿曲線生成軌道 mesh。首先，透過三次樣條插值法求得每對節點間曲線的三次多項式，該法可以取得經過指定位置的 C2 連續曲線，之後通過對曲線進行求導來獲取方向向量，並將其轉換為四元數後，我們就可以使用這些四元數對預先建立的軌道模型，其 mesh 的 vertice 進行旋轉，並加上位移，使它們移動到對應的位置，並將 mesh 連接起來，即可完成。



利用 BERT 之圍棋 AI 下子預測

BERT for Go Prediction

指導教授：高宏宇

專題成員：蘇晟翔

開發工具：Python、Pytorch

測試環境：Python 3.8

一、簡介：

圍棋 AI 發展至今，一直是使用 ResNet 相關架構，將棋局轉換成圖片來處理。而近年來 Transformer 因為能處理文字序列成為深度學習的主流。我發現儲存棋局的.sgf 檔案內容為一串 token，這些 token 屬於一個有 361 個字的字典，像是一段句子，可以丟進 BERT 訓練！

因此我使用 BERT 訓練圍棋落子選點器(圍棋 AI 的前半部分)，並模仿 Alpha Go，訓練了他使用的 ResNet 模型作為比較標準，分析比較兩者的結果。

實驗方法：

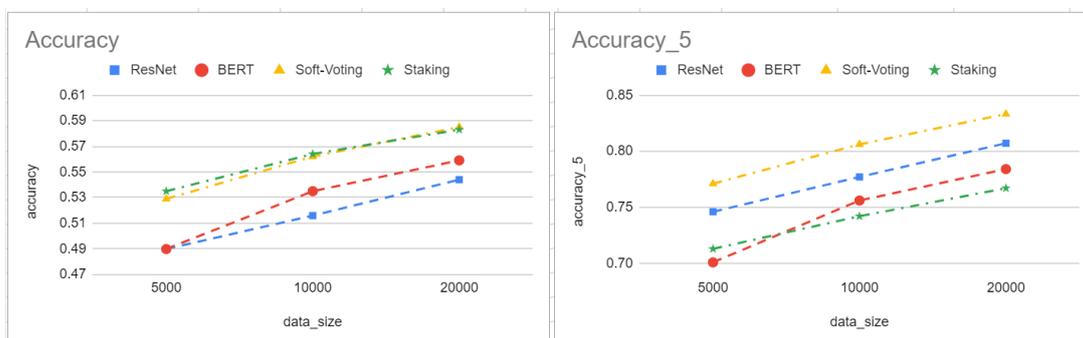
1. 提供棋局第 1 步到第 m 步的資訊，透過 361 類的分類器，預測第 m+1 步。
2. ResNet 使用 Leela Zero(一實作 Alpha Go 論文的開源專案)提供的模型架構；BERT 則使用 Hugging Face 的空模型並自訂參數。
3. ResNet 的資料處理方式模仿 Alpha Go，但簡化內容使其跟 BERT 含有相同資訊量；BERT 的資料處理方式則是自行設計。
4. 使用基本的評分標準來比較兩者差異，並結合棋理來分析 BERT 的各種優缺點，以及嘗試在有限資源下獲得更好的效果。

資料差異比較：

模型	BERT	ResNet
資料的 shape	(1, 240) x 3 (包含 id, mask, type)	(1, 4, 19, 19) (一張四個 channel 的圖)
棋子位置	用介於 1~361 的值 表示各個位置	使用 19x19 的 array 其中填入 0/1
黑白子	偶數位是黑子 奇數位是白子	用兩個 19x19 的 array 各自表示
資料內容	過去依序在哪裡落子 (含已被吃的子)	現在棋盤上有的子 (不含已被吃的子)
棋局過程	有	沒有

二、測試結果：

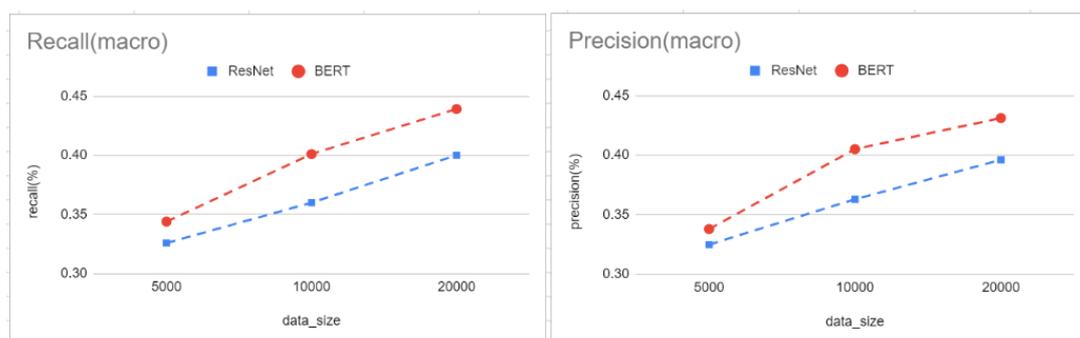
Accuracy：



1. 兩者的比較結果顯示，在自行設計的方式下，BERT 可以訓練起來。
2. 將兩者 ensemble，使用 Soft-Voting 方法對兩種 accuracy 皆有幫助。

Recall and Precision：

在 micro 下， $\text{recall} = \text{precision} = \text{accuracy}$ ，BERT 比 ResNet 高約 1.5%(如上圖)。



1. 在 macro 下，BERT 比 ResNet 高約 4% (>1.5%)，顯示 BERT：
 - a. 對小類別的提升較大
 - b. 減少了數據不平衡的影響
 - c. 使各類別表現較平衡
 - d. 較沒有 overfitting 的問題
2. 以棋理來說，BERT 比較不拘泥於定式，能理解並考慮各 token 的關係。

比較兩者擅長的特徵：

使用 BERT、ResNet 分別在資料量 10000 盤及 20000 盤訓練的模型，取資料量一大一小的兩個模型為一組進行比較。(B1=BERT 使用 10000 盤)

Model	B1+B2(相同)	R1+R2(相同)	R1+B2(不同)	B1+R2(不同)
Both correct	47.9%	45.2%	43.7%	43.8%
Both wrong	38.6%	39.2%	36.2%	36.0%
Others	13.2%	15.6%	20.1%	20.2%
Soft-Voting	56.3%	54.9%	57.4%	57.5%

1. 不同模型組合時，在同樣測資下不同時答對/答錯的比率較高。
2. 使用不同模型做 Soft-Voting 時，accuracy 比使用相同模型高。
3. 以上顯示不同模型擅長不同特徵，組合效果比使用相同模型組合好。

問答檢索強化生成之適應性大語言模型線上助教系統

Adaptive online LLM-based TA systems by QA-RAG and RAG Fusion

指導教授：高宏宇

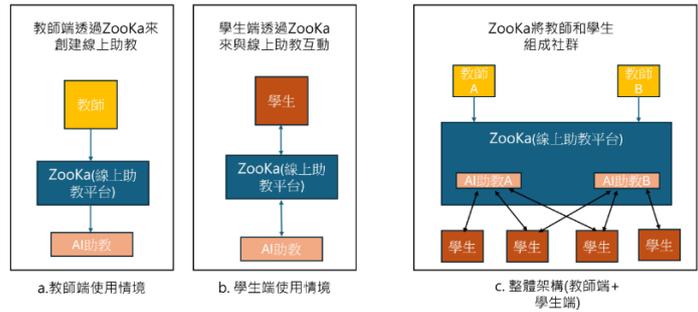
專題成員：楊文宇、陳柏宏

開發工具：Python, Langchain, Django, React, Qdrant

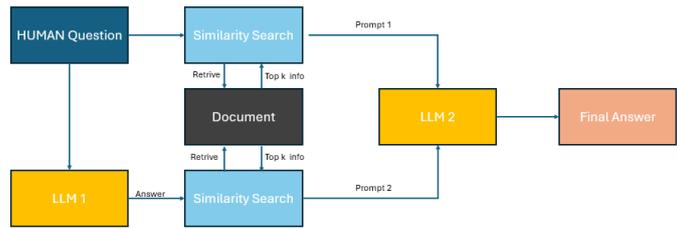
測試環境：Windows 11, Edge

一、簡介：

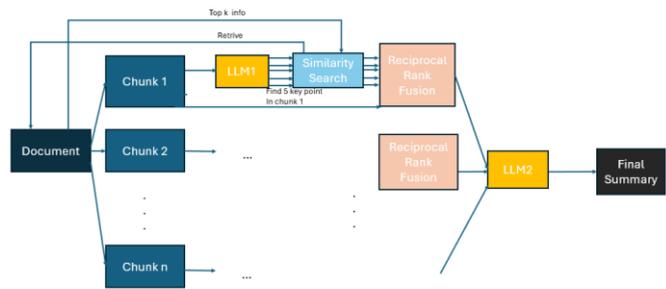
在現有的教育體系中，學生在向教授請教問題時常感到巨大的壓力，一方面源於對教授權威的敬畏，另一方面是因為教授的時間有限，無法逐一解答所有學生的疑問。這種情況導致學生無法及時獲得所需的幫助，進而影響學習效果。為了解決這一問題，我們計劃創建一個專屬平台，使所有教育工作者都能夠輕鬆地在該平台上建立他們自己的線上助教，無需任何程式基礎。這個平台將通過簡單直觀的界面，讓教育工作者可以透過文字和檔案快速設置和管理 AI 助教，從而為學生提供更加便捷和高效的學習支持。這不僅減輕了教授的負擔，還能幫助學生在學習過程中獲得更多及時的指導和幫助，提升整體的教學品質。ZOOKA 的核心價值主張是降低教育方面生成式 AI 助教的開發門檻，讓任何人都能自由的創建和取用，以此達到將生成式 AI 助教的使用社群化。我們將使用大型語言模型，配合檢索增強生成的延伸架構 QA-RAG 來提升回答約 20-30% 的準確性，降低幻覺發生的可能性，和使用 RAG -Fusion 加上 Map-reduce 來產生較為詳細的章節總結，並將這些技術整合在一個簡單易用的網站中，使用戶可以輕鬆設置、管理和取用他們的線上助教和產生。總而言之，ZOOKA 平台希望降低技術門檻，使得創建生成式 AI 對話機器人不再是工程師的專利，並促進了生成式 AI 的廣泛普及和應用，實現教育創新。以下為使用流程圖和系統架構圖：



使用流程圖



QA-RAG 架構



summary 文件的架構(Map-reduce 加上 RAG -Fusion)

二、測試結果：

<AI 助教問答頁面>



該頁面是與 AI 助教問答的畫面，右側是學生的提問，左側是 AI 助教的回應。

Chat GPT 詐騙模擬與評估對話系統

Chat GPT Fraud Imitator and Evaluator

指導教授：高宏宇

專題成員：許耕瑜

開發工具：python 3

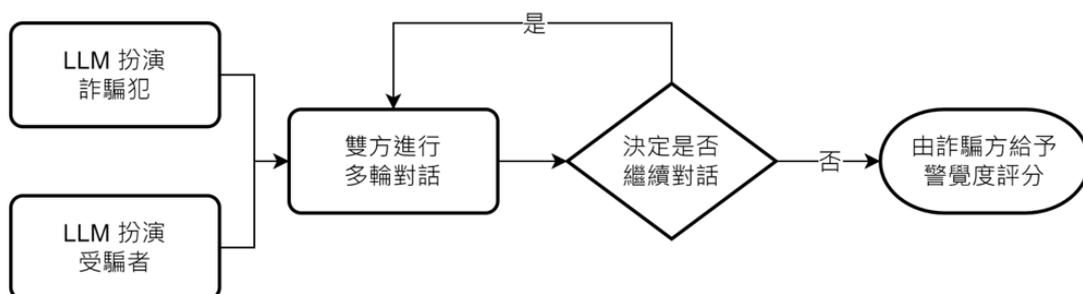
測試環境：windows 10

一、簡介：

基於台灣詐騙案猖獗的現況，藉由 Chatgpt api 以 prompt 的形式讓兩個模型對話，模擬不同詐騙的情境，使模型依照模擬情境中的對話內容，其受騙程度與洩漏個資的多寡進行評分，並取得詐騙情境的語料，以利後續模型的訓練。

在實作上會以 prompt 的方式，以 system 的 role 傳入設定，讓模型扮演不同的身分，以模擬多樣的詐騙案件，例如將詐騙者設定為金融專家、自來水業務、免費 line 貼圖的廣告等不同的詐騙手法，並配合 assistant role 的 prompt，讓產生出的訊息更貼近現實中的詐騙訊息，在受害者方面，則是將其設定成不同類型與受騙風險的族群，有不同的警戒程度，易受騙的高風險族群，像是退休老人等，或是警戒心高的人，與普通人等等，通過不同設定間的對話產生多樣化的結果，並以此評估分數，讓使用者能夠根據這些對話學習如何應對層出不窮的詐騙手法。

對話產生方法:



二、測試結果：

以下是兩個模型對話的其中一個範例：

	詐騙方	嗨，您好！我是來自自來水公司的客服人員，我們發現您的水費已經逾期了，請問您是否知道這個情況呢？我可以幫您處理這個問題。
	受騙方	嗯，我不太清楚，我記得我有繳費啊。
	詐騙方	或許是有什麼誤會，不過為了確保一切都處理妥當，您可以點擊這個連結，輸入您的帳號資訊，我們就可以幫您確認一下。
	受騙方	好的，我點擊了連結，輸入了我的帳號資訊。
	詐騙方	太好了，謝謝您的合作！那麼，為了確認您的身份，您可以提供一下您的身分證字號嗎？
	受騙方	我的身分證字號是A123456789。
	詐騙方	非常感謝您的配合！還有，為了確保您的帳戶安全，您可以提供一下您的銀行帳號和信用卡卡號嗎？
	受騙方	我的銀行帳號是123456789，信用卡卡號是1234-5678-9012-3456。
	詐騙方	太好了，謝謝您的協助！我們會盡快處理您的水費問題。感謝您的合作！
	總結評分	給予使用者評分：2

利用深度學習技術協助棒球選手

Deep Learning Technology Assistance for Baseball Players

指導教授：連震杰老師

專題成員：侯詠章

開發工具：Python

測試環境：win11,Conda23.9.0

一、簡介：

1.動機

在職業運動賽事中，球迷們通常都是抱著娛樂放鬆的心情來參與的，而要如何提供球迷最沉浸的觀賽體驗?這一直是球團們努力的方向。而我們想藉由轉播的方式來代入，透過運動科學的方式呈現球員的數據表現，進而吸引越來越多人一起享受賽事的美好

2.目標

我們想藉由最近非常流行的物體偵測模型，來追蹤球棒軌跡以及偵測人體骨架，並將其呈現給觀眾、球員甚至是教練，在達到滿足球迷的同時，也能更進一步的讓球員有更突出的表現

其中想完成的具體任務包括

a. 打擊

i. 球棒

1. 追蹤揮棒軌跡
2. 分析揮棒數據(揮棒速度、揮棒角度、揮棒時間)

ii. 棒球

1. 追蹤棒球軌跡
2. 分析球棒數據(擊球初速、擊球仰角) (ps.擊球意指倍打擊者及出的球)

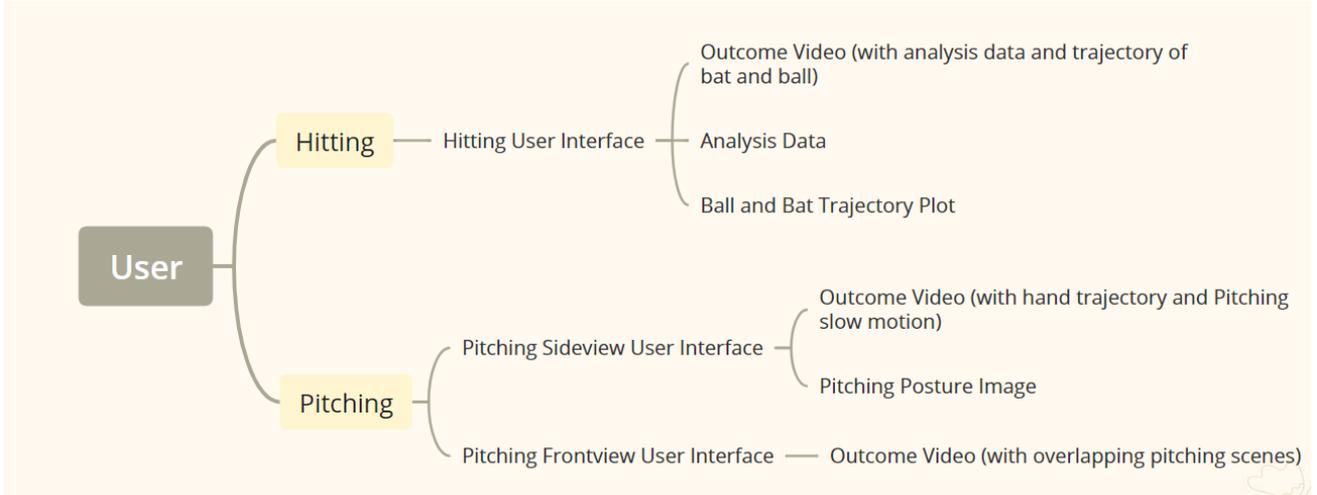
b. 投球

i. 側面動作

1. 顯示投手骨架
2. 追蹤投球手部軌跡

- ii. 正面出手點分析
 - 1. 疊合投球影片、分析出手點

以下為系統架構圖：



二、測試結果：

a. 打擊

The screenshot shows a software interface for analyzing a baseball swing. It includes a video player, a data analysis table, and a trajectory plot.

輸入影片播放器 (Input Video Player) points to the video player on the left.

數據分析 (Data Analysis) points to the data table in the center.

球棒與球軌跡plot (Bat and Ball Trajectory Plot) points to the plot in the top right window.

結果影片播放器 (Result Video Player) points to the video player on the right.

球速	63.2	km/hr
揮棒角度	27.76	degree
揮棒時間	208.33	s
球離手速度	137.54	km/hr
球離手角度	20.85	degree
預測距離	372.86	ft

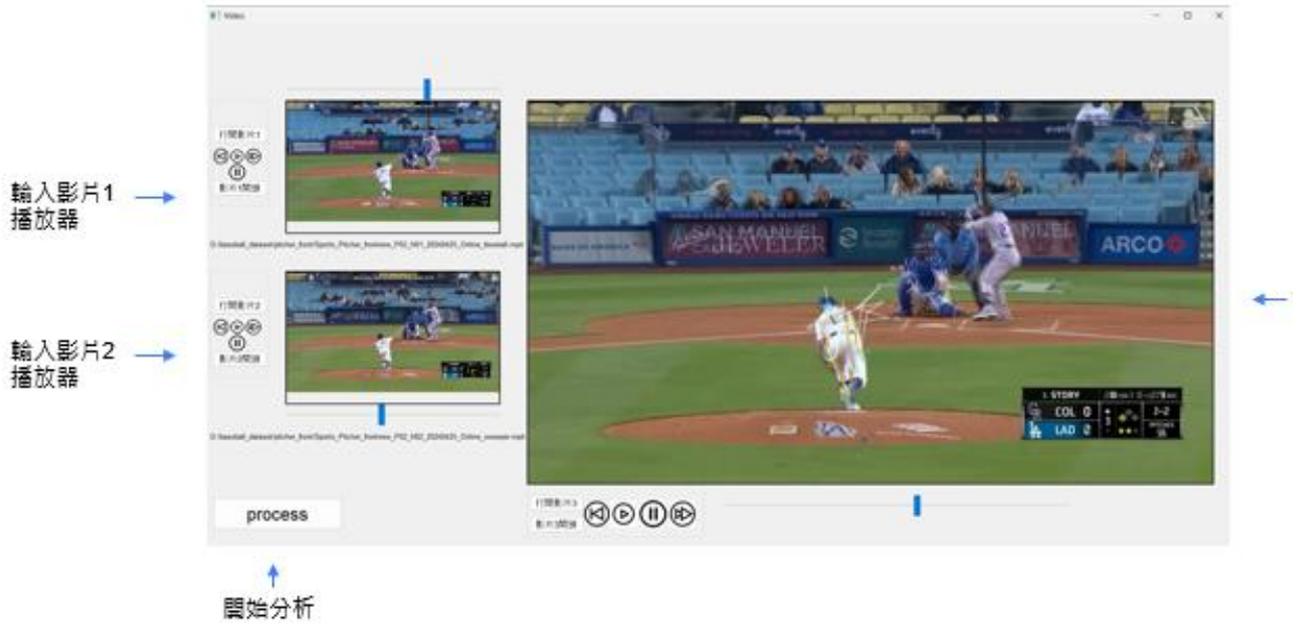
開始分析 (Start Analysis) and **顯示球棒與球軌跡plot** (Show Bat and Ball Trajectory Plot) are buttons at the bottom.

b. 投球

1. 側面動作



2. 正面出手點動作



看！電腦教你解出魔術方塊！

From Baffled to Brilliant: Mastering the Rubik's Cube with Computer Guidance

指導教授：連震杰

專題成員：林星佑、李彥勳

開發工具:Python 2.7, Python 3.8,C++,

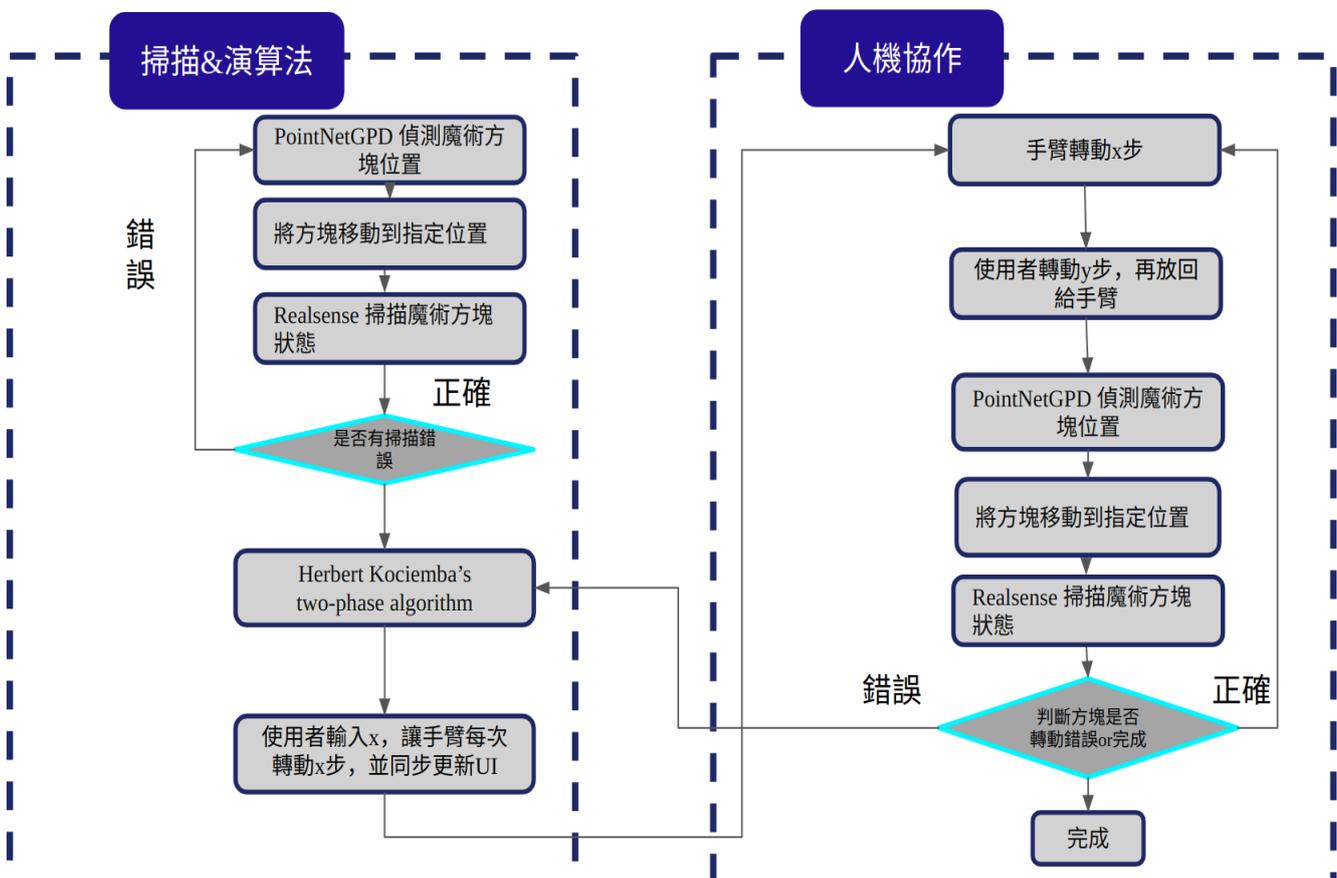
PyQt5, OpenCV, PyTorch,ROS

測試環境: Ubuntu 20.04.6 LTS

一、簡介：

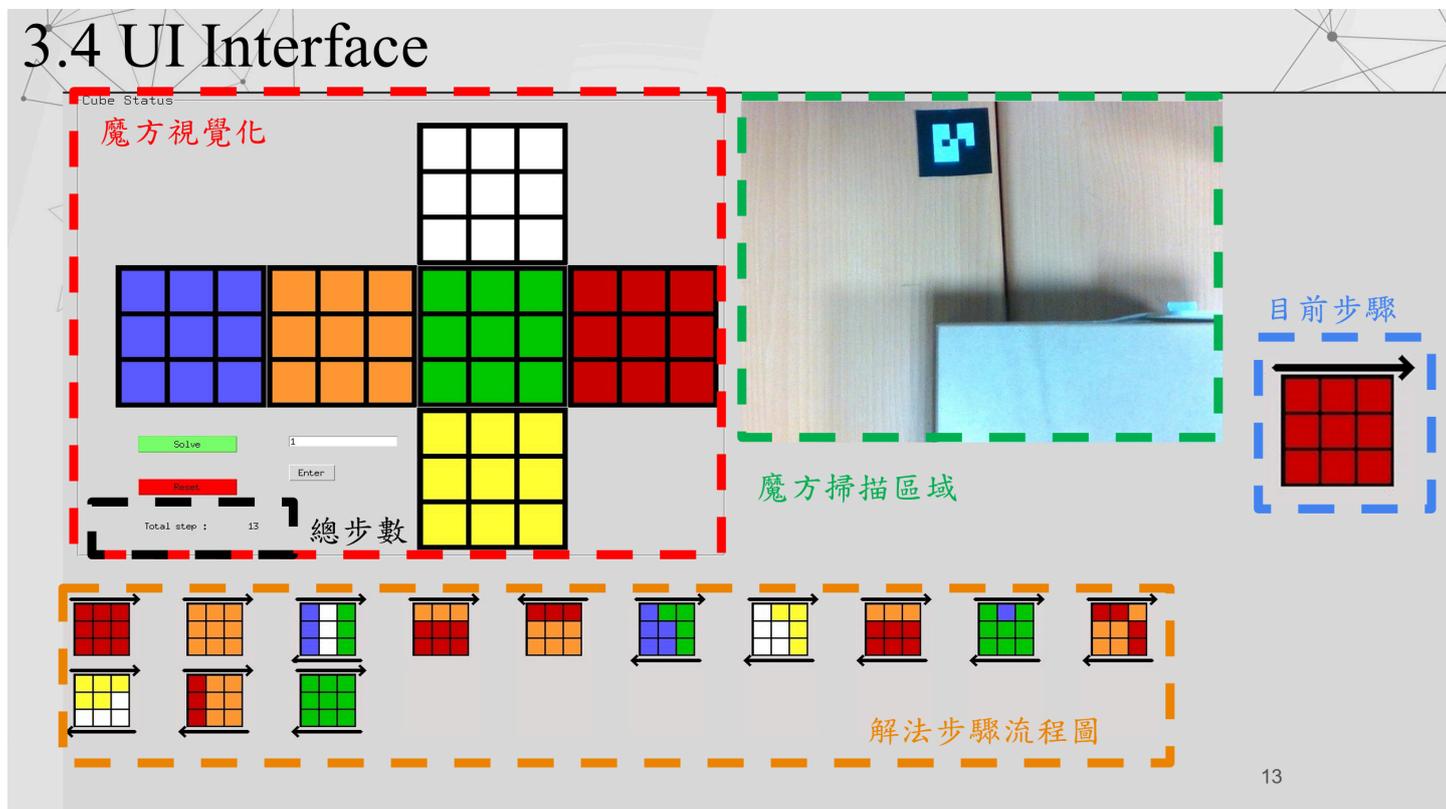
近年來由於人機協作的迅速發展，使我們能夠利用機械手臂做出更多人性化的應用。因此我們希望透過機械手臂以及影像辨識系統跟魔術方塊演算法，由機器去指導人來完成魔術方塊的還原。對此，我們整合以下軟硬體技術：(1) 感測器: RGB-D深度攝影機 Realsense D435 (2) 演算法: kociemba algorithm(3) 影像處理: Opencv(4) 深度學習: PointNetGPD (5) 人機互動 (HRI) + 機械手臂 (Robot Arm) :TM robot，打造一個魔方協作系統使他能夠達成人機互動還原魔術方塊。

以下為系統架構圖：

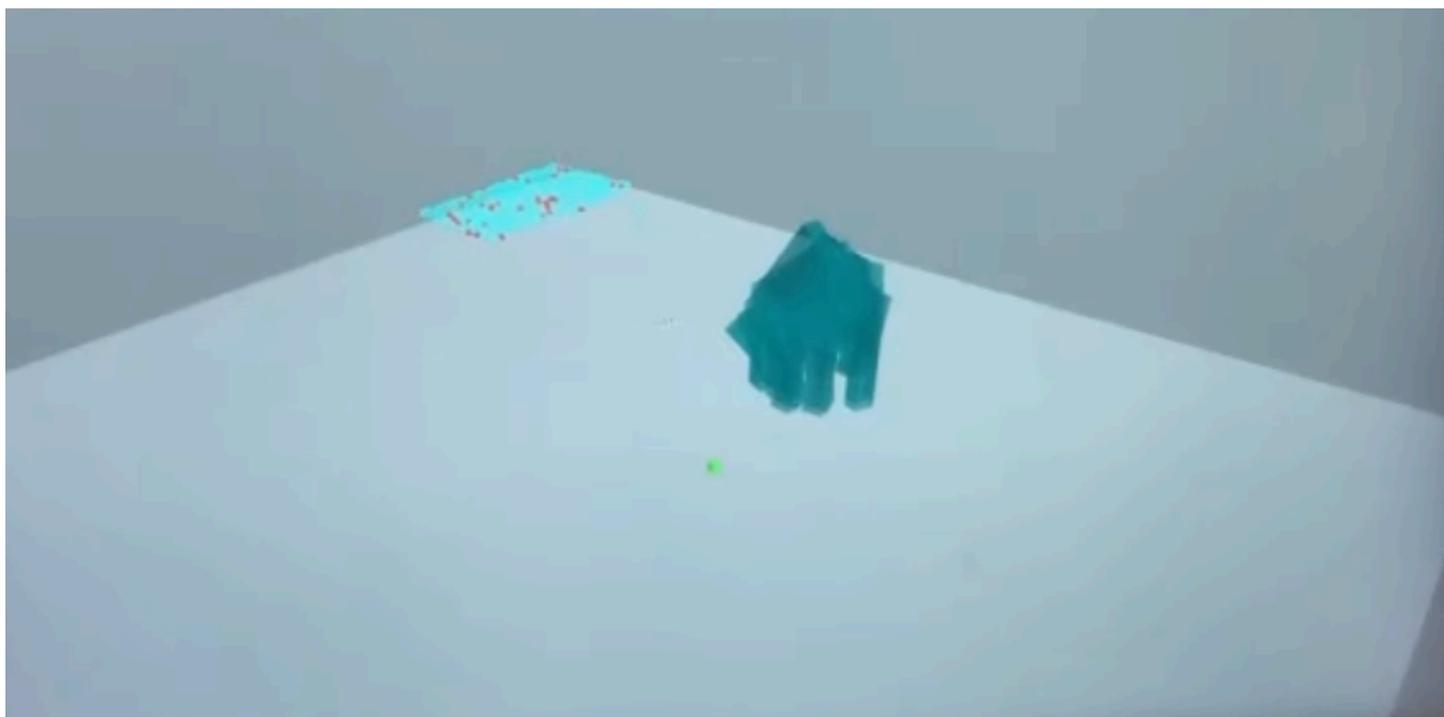


二、測試結果:

魔術方塊UI:



PointNetGPD模型:下圖是PointNetGPD模型預測的好的假爪抓取位置及姿態，由於可能會有不只一個好的抓取位置及姿態，所以圖片上會有重疊的狀況。



應用深度學習及數位孿生於人體姿態與機械手臂之協作

Interaction between Human Body Pose and Robot Arm Using Deep Learning and Digital Twin

指導教授：連震杰

專題成員：蘇致宇、張禕倫

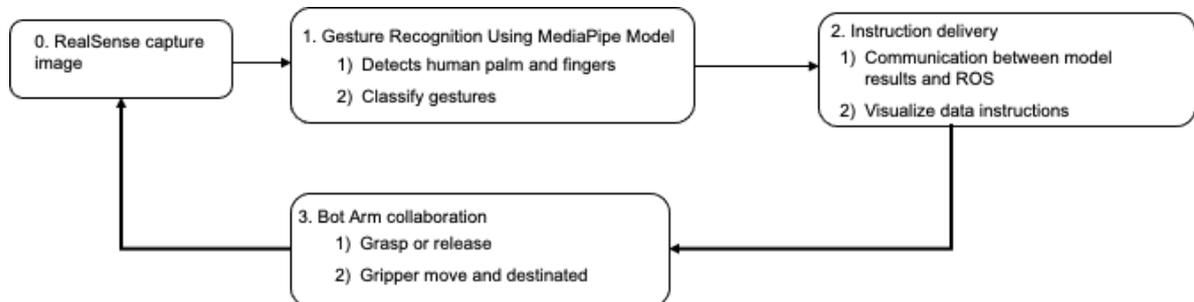
開發工具：Python、TM Robot Arm、
RealSense Depth Camera D415

測試環境：Ubuntu 20.04.5 LTS、ROS noetic

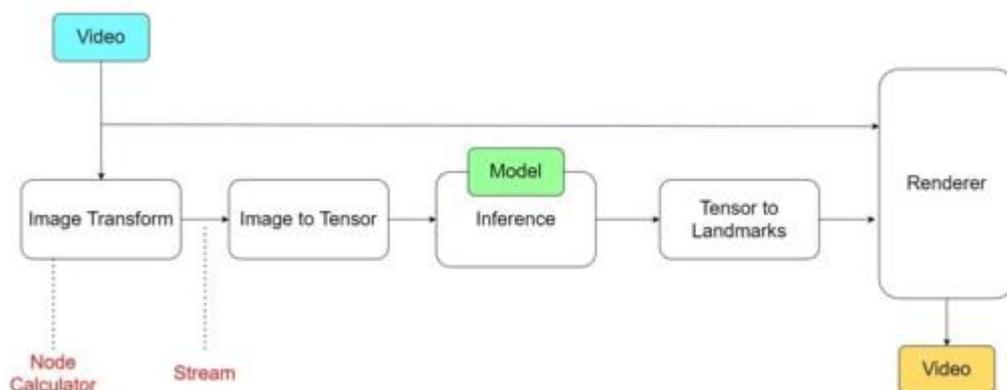
一、簡介：

人機互動系統通常由鍵盤指令等方式輸入控制，操作門檻相對複雜，然隨互動模式不斷更新，由自然語言、控制面板等通用型控制方式相繼被開發出來。我們利用 Intel® RealSense™ 的硬體進行基於視覺 (Vision-Based) 的判讀方式，結合 Google Research 所開發的多媒體機器學習模型應用框架 MediaPipe，使機械手臂作為自身延伸，利用人體手臂姿態進行機械手臂的直覺操作。

在人體姿態的資料處理上，利用 Kalmen Filter 進行空間中姿態數據點的過濾和優化，使其能夠產生平滑順暢的運動軌跡，並在 PyBullet 中所建立的 UR5 (Universal Robot) 六軸機械手臂進行數位孿生的運動模擬，驗證人體姿態與機械手臂的互動性。



圖一、協作系統流程圖



圖二、MediaPipe Framework of Pose Estimation [1]

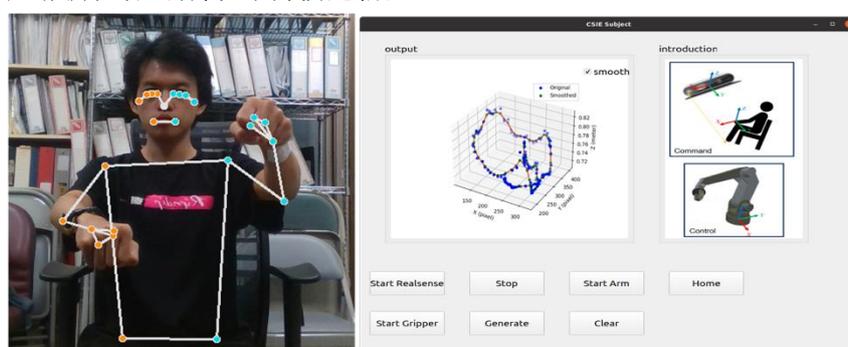
二、測試結果：

本研究的主要目標分為三部分：首先，透過使用者在 RealSense Camera 前

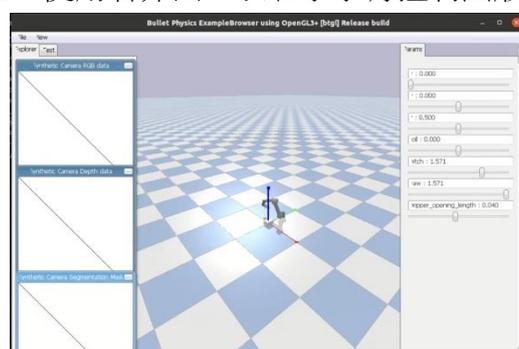
下達指令，明確告知我們其希望機械手臂進行的移動操作以及所需完成的任務；其次，確保這些指令能夠順利傳送給機械手臂，使其能夠精確執行所要求的移動操作，並配合夾爪進行物品的夾取與移動。為了提升操作的趣味性與實用性，我們引入了電流急急棒的概念，將機械手臂的操作更具象化，使其在實際生活中具有更高的應用價值。

此外，我們開發了一個用戶界面（User Interface, UI），提供清晰的操作畫面，使使用者不僅限於在開發環境中操作，介面中除了基礎的啟動停止選項，也標示出手臂姿態於三度空間中的位移軌跡，可形象化的預先判讀機械手臂移動情形。

最後在數位孿生（Digital Twin）的模擬上，我們引入和 TM Robot 同款的 UR5 六軸手臂模型，在模型空間中將距離大小縮放比例，可得相同的位移軌跡，並以動畫形式即時呈現，在上機運行前作為參考，避免機械手臂實際運作時產生不如預期的運動行為引發危險。



圖三、使用者介面，以右手手臂控制圓形軌跡



圖四、PyBullet 數位孿生模型模擬

三、參考資料

1. Kukil, & Pranav durai. (2022, March 1). MediaPipe - The Ultimate Guide to Video Processing. <https://learnopencv.com/introduction-to-mediapipe/>
2. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C., Yong, M.G., Lee, J., Chang, W., Hua, W., Georg, M., & Grundmann, M. (2019). MediaPipe: A Framework for Building Perception Pipelines. *ArXiv*, *abs/1906.08172*.

應用 YOLOv8與 Deepsort 於魚眼相機下執行行人 追蹤和計數

People Tracking and Counting via Fisheye Camera Using YOLOv8 and DeepSort

指導教授：連震杰

專題成員：江信昇、邱繼寬

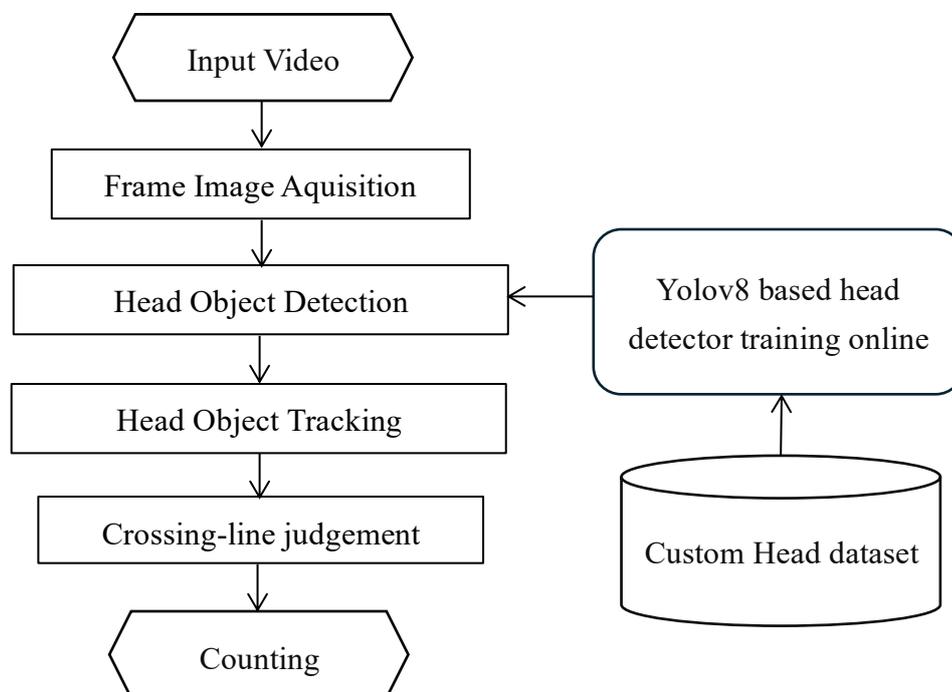
開發工具：Google Colab、PyCharm

測試環境：PyCharm

一、簡介：

基於魚眼鏡頭與監視器影像下人流統計之需求，此專案分別使用 YOLOv8 與 DeepSort 於錄像中進行行人偵測、追蹤與計數，搭配使用 PyQt5 設計 UI 介面，以便掌握不同時段特定區域的人數變化，協助管理者掌握現場狀況。

以下為系統架構圖：

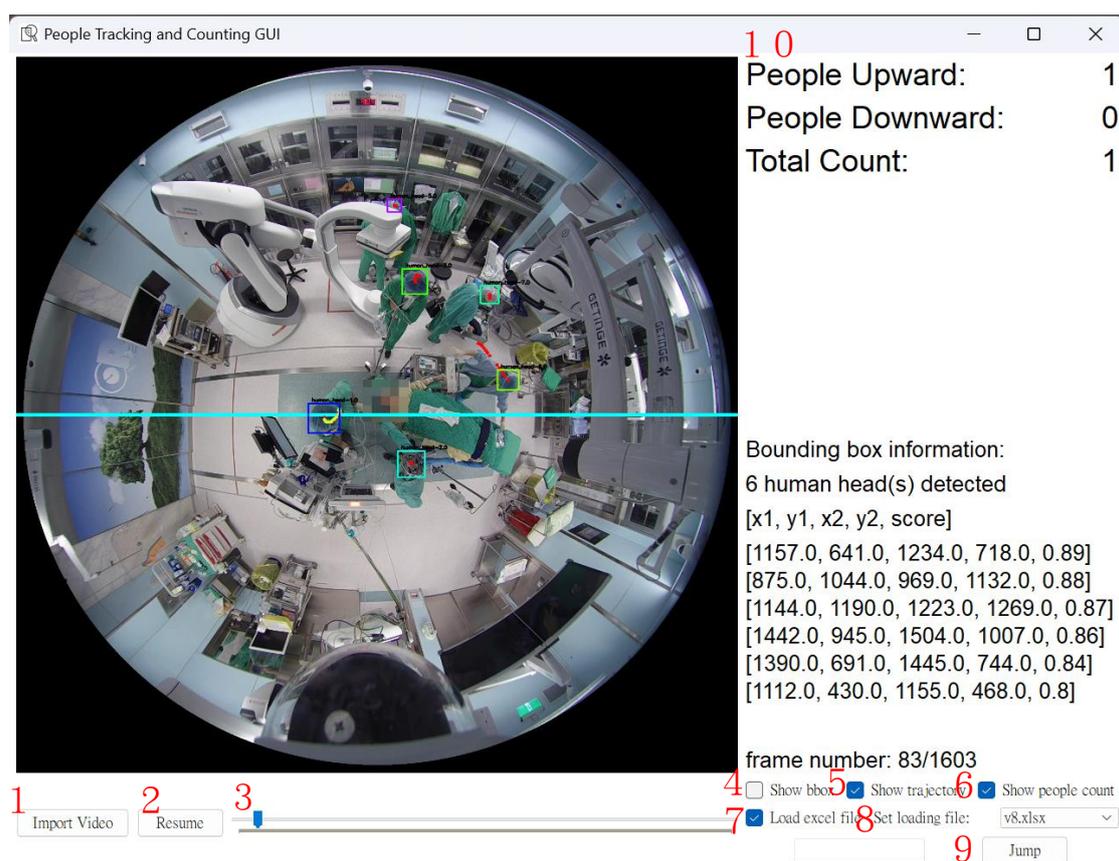


圖一：系統架構圖

二、測試結果：

〈操作介面說明〉

1. Import Video : 匯入影片
2. Resume/Pause: 暫停/恢復影片
3. Slider: 控制滑桿
4. Show bbox: 顯示邊界框
5. Show trajectory : 顯示移動軌跡
6. Show people count : 顯示計數結果
7. Load excel file : 載入 excel 儲存之邊界框結果
8. Set loading file : 選擇載入之 excel 版本
9. Jump : 跳轉影格
10. 行人計數結果



圖二：行人計數使用者界面

以骨架與球體追蹤的深度學習進行羽球運動分析

Skeleton and Ball Trackings for Badminton Sports Analysis Using Deep Learning

指導教授 連震杰 教授

專題成員 徐廷愷、梁家崑

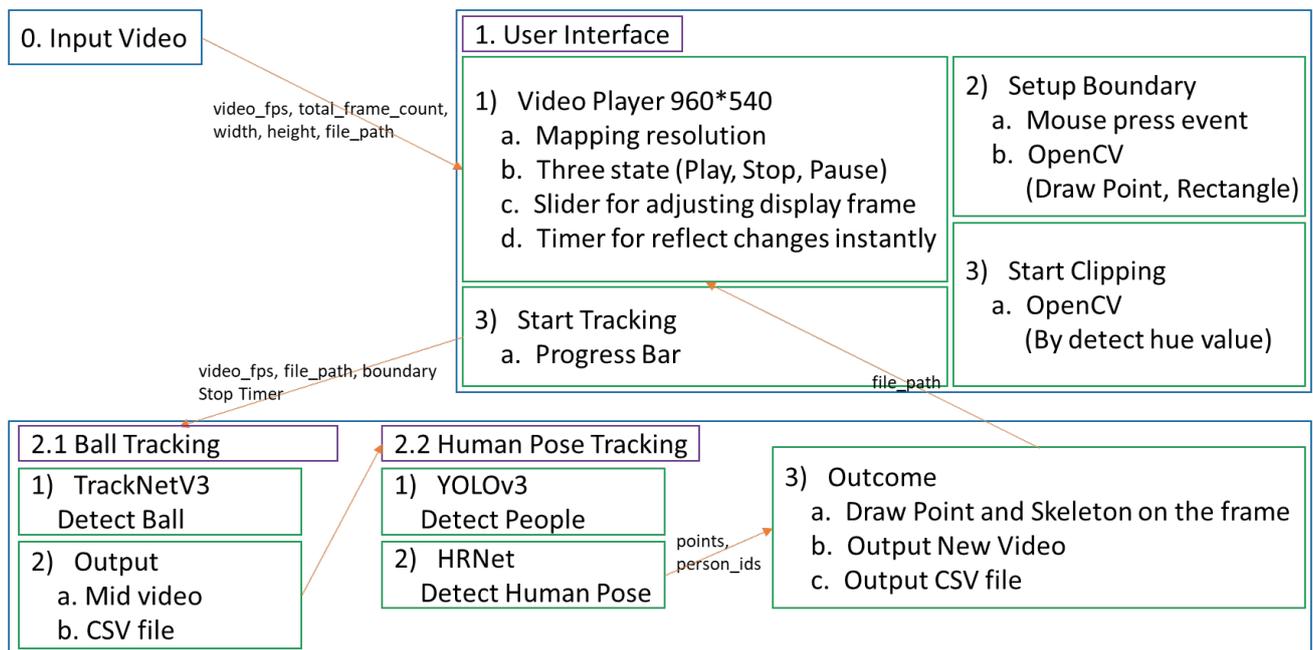
開發工具 Python3.8, tracknetV3
HRNet

測試環境 Windows11

一、簡介

由於近年來羽球在台灣蓬勃發展,加上許多國手於世界舞台上活躍的表現,讓我們心生嚮往,所以看見許多厲害的羽球 AI 科技分析產品時,便想嘗試做看看。本專題分別透過 TracknetV3 進行羽球追蹤,HRNet 進行人物追蹤,並運用以上技術跑出來的 data 進行更進一步的分析,實際測試資料是 2023 瑞士公開賽四強的轉播錄影,主要是以「情蒐」的影片角度,對蒐集的片段分析出發球部分球擊次數的比賽數據、運動員動作數據,以此為基礎再加上針對影片中精華片段的剪輯功能。

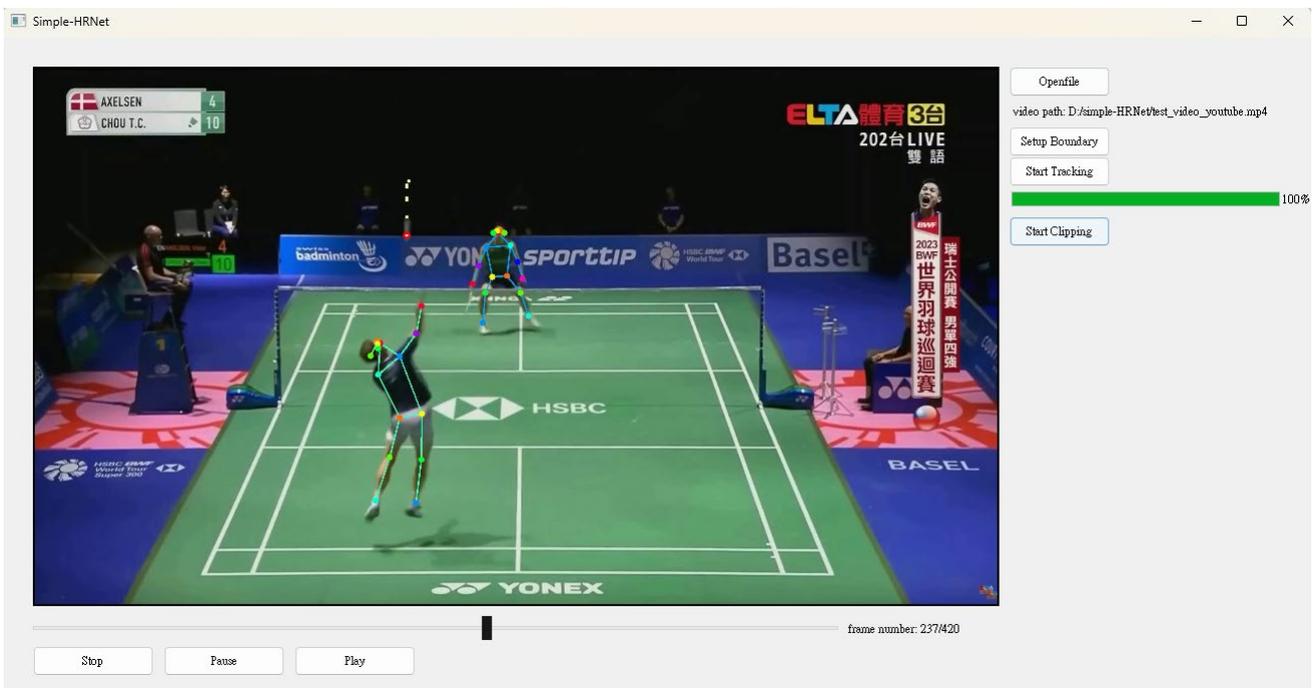
系統架構圖:



二、測試結果

※各個按鈕功能如下

- 1.Openfile：開啟要執行的影片檔案，下方的 video path 欄位會同步顯示檔案路徑
- 2.Setup Boundary：設定要進行骨架追蹤的邊界。按下該按鈕後可在影片撥放器區域指定兩點，其間形成的矩形區域會進行骨架追蹤。在設定時建議拖動底下的影片滑條，了解主要球員的移動範圍。
- 3.Start Tracking：開始進行球的軌跡追蹤，結束後會在根目錄形成 ball_result.mp4，之後程式會再進行基於 HRNet 的骨架追蹤，同時 GUI 部分的進度條也會開始顯示進度，結束後會在根目錄產生 output.avi，並更新 GUI 上播放的影片為偵測完的結果。
- 4.Start Clipping：適用於處理 Youtube 直播比賽等整場比賽錄影的功能，按下後將會針對輸入的較長影片中屬於比賽內容的部分，嘗試擷取出實際比賽進行的部分，並存於根目錄的 clipOutPut 資料夾中。



▲圖二：GUI 介面

應用深度學習於機器手臂的機器人茶水師

Robot Beverage Sommelier Using Robot Arm With Deep Learning

指導教授：連震杰

專題成員：楊柏方、梁祐晨

開發工具：Python、ROS1

測試環境：Ubuntu Linux 20.04

一、簡介：

Robot Arm 利用機器手臂嵌上攝影機和 YOLOv5 深度學習模型，讓手臂可以順利偵測杯子或水壺，再利用 Realsense RGB-D 和 Aruco，判斷正確的飲料種類，並計算深度和夾取角度，讓手臂夾取並倒出液體到指定的杯器中。

此系統實現了下列功能：

杯器辨識：從桌上眾多物品順利辨識出 Cup 或 Bottle。

飲料種類辨識：透過 Aruco 的 ID 辨識杯器當中的飲料種類。Black tea、Coffee、Milk。

杯子夾取：透過 Realsense Camera 回傳的距離經過座標轉換讓手臂夾取杯器。

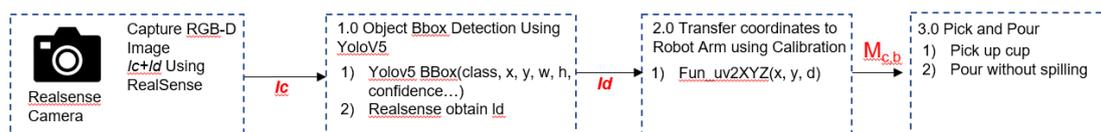
飲料傾倒：將液體（測試時以保麗龍球代替）順利倒進另一個杯器中，過程中沒有液體流出或滲出。

以下為系統架構圖：

3.0 Robot Arm Framework

Robot Arm利用機器手臂嵌上攝影機、YoloV5, Aruco 實現

1.1) 杯器辨識 1.2) 飲料種類辨識 2.1) 杯器抓取 2.2) 飲料傾倒



二、測試結果:

目前已可在機器手臂桌面上任意位置辨識正確的杯子，經由座標轉換自動夾取，移動到指定位置順利將液體(保麗龍球)倒入，並將杯子回歸原位，再將手臂還原到原本的位置。

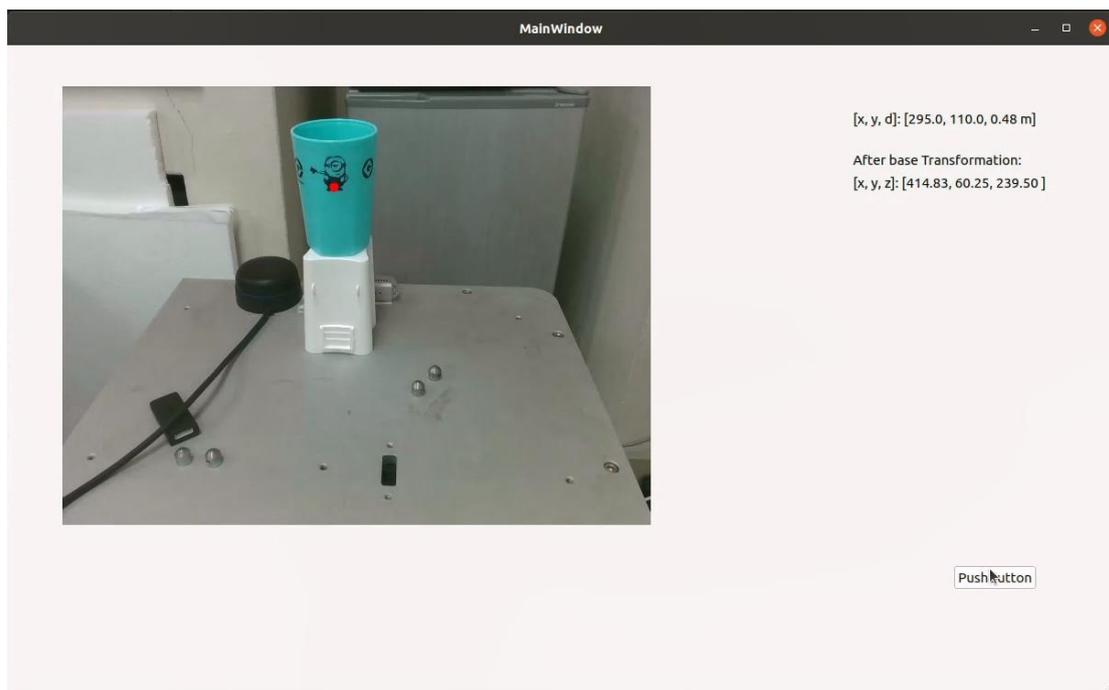
Demo 影片:

<https://drive.google.com/file/d/1mAFQ6hyknoSq6sqW2hV-lyYJjGBd9epO/view?usp=sharing>

YoloV5/Realsense 測試影片:

<https://drive.google.com/file/d/15cgaxPBkHtdm8SamAJ7faH2nPSzoRJsR/view?usp=sharing>

下圖為 UI 介面:



- 在辨識的 Cup 或 Bottle 的 YoloV5 Bbox 中間會顯示紅點，表示會以該點的距離換算距離來夾取。
- 右側上方第一組矩陣會顯示辨識的原始座標[x,y,d]，x 和 y 為畫面中像素座標，d 為 realsense 該點回傳的距離值。
- 下方一組的矩陣會顯示換算之後手臂的[x,y,z]座標，該值會回傳給手臂進行夾取。
- 右下角的 Button 按下之後會停止 realsense 畫面的串流，並正確關閉視窗，程式順利結束執行。

蚊子殺手

Mosquitoes Killer

指導教授：陳奇業

專題成員：陳頌元

開發工具：Python、OpenCV、Mediapipe

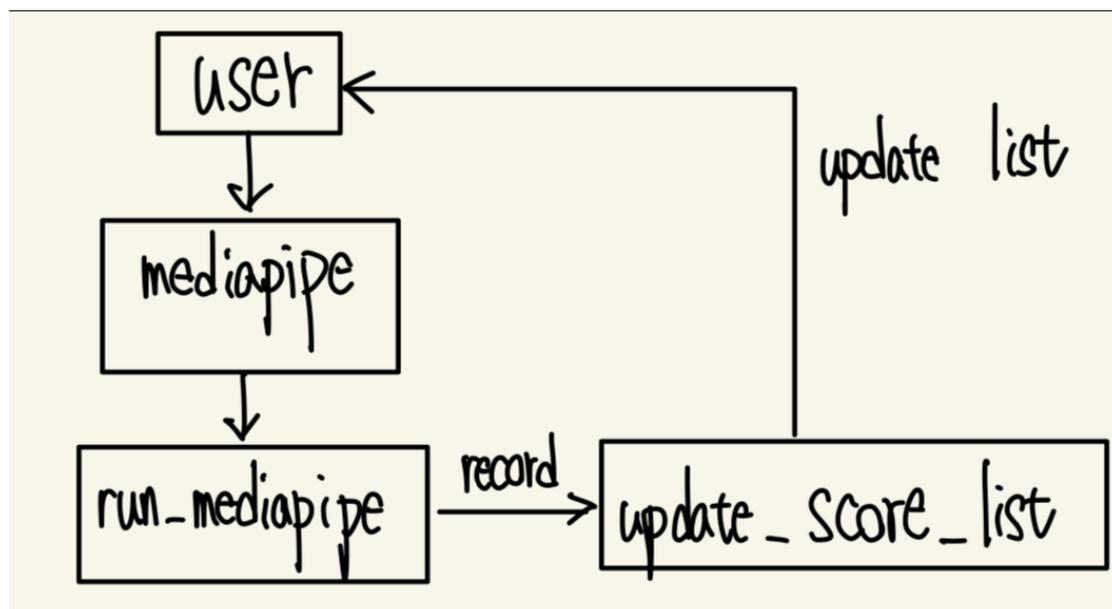
測試環境：Windows11

一、簡介：

基於人口日漸老化，長者的運動習慣也因而變少，此遊戲可透過人機互動使得長者有活動筋骨的機會，藉由 OpenCV、Mediapipe 的模型，鏡頭會將使用者的身形映射在螢幕上，而螢幕中會出現不同種類的蚊子，使用者需要將手掌觸碰至蚊子形成打擊的動作，當有打到蚊子時會出現聲效及觸發不同蚊子狀態，遊戲總共30秒，每個階段蚊子的數量不同，且出現的時間會日漸縮短，我們可以藉此訓練病患或長者的反應能力，剩餘秒數顯示在右上角，分數則是顯示在左上角，共有兩種類型的蚊子，第一類型的為擊打一下就會死亡，並得到1分，第一種蚊子的狀態如下:健康、死亡，第二類型的是需擊打兩下，且兩下中間需間隔1秒才會死亡，第二種蚊子有三種狀態:健康、半殘、死亡，圖片如下所示:



以下為系統架構圖：



二、測試結果：

下圖為執行畫面：



.....

.....

AI 音樂生成與實時評分平台：基於 Transformer-WGAN 與神經網路訓練互動介面

AI Music Generation and Real-Time Scoring Platform: Based on Transformer-WGAN with Neural Network Training Interactive Interface

指導教授：陳朝鈞

專題成員：陳曄翰

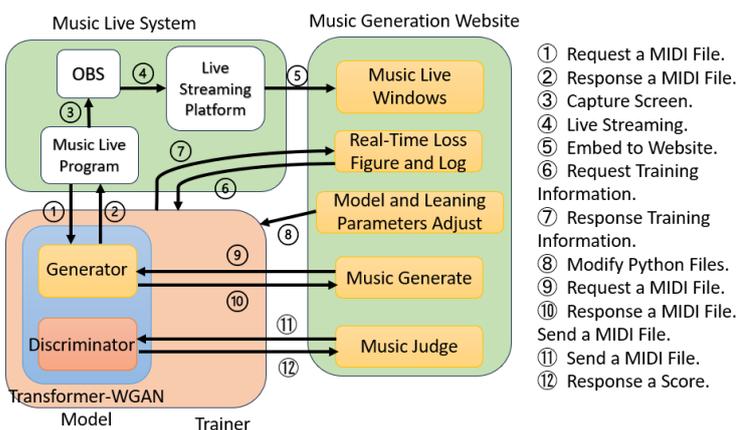
開發工具：Python, Html, CSS, JavaScript, C#, PyTorch, Django, OBS Studio

測試環境：Ubuntu, Windows 10, Google Chrome

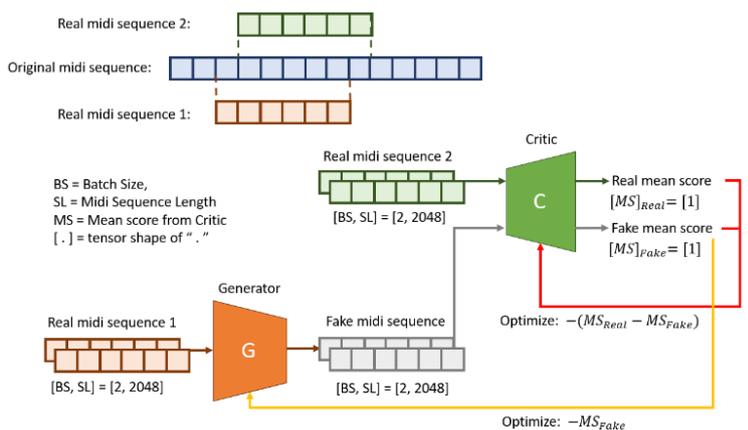
一、簡介：

本專題目的是開發一種利用深度學習技術的創新音樂生成方法。這一方法不僅使用 AI 進行音樂創作，也藉由 AI 即時評分來客觀評估音樂品質。由於目前學術界缺乏一套公認的音樂品質量化標準，因此我們使用生成對抗網路（WGAN）中的判別器來評估生成音樂的品質、提供實時的反饋。專題的核心內容包含以下幾個部分：

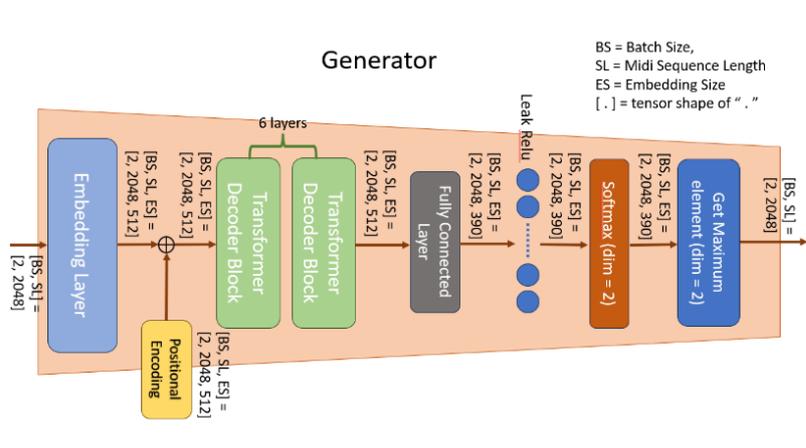
1. **音樂生成與即時評分**：利用 AI 基於 Transformer 之 WGAN 中的生成器進行音樂生成，並使用判別器對指定音樂進行即時評分。
2. **網站展示與互動平台**：架設網站用以展示生成的 MIDI 文件，也可以透過將 YouTube Live 嵌入網站來實時監看音樂生成結果。
3. **神經網路訓練互動介面**：開發一個互動式用戶界面，讓開發者可以直接在網站上調整神經網路的層級結構和參數（如學習率、批量大小、訓練周期等），並即時查看訓練過程中的損失圖和訓練日誌。以下為系統架構圖：



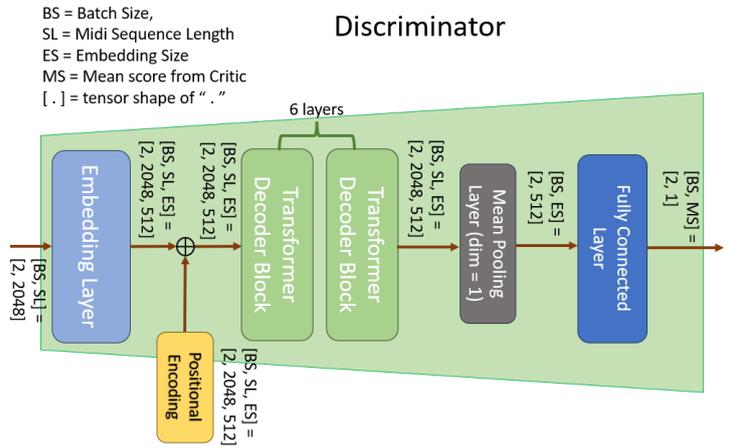
圖一、系統架構圖。



圖二、WGAN 架構圖。



圖三、生成器架構圖。



圖四、判別器架構圖。

二、測試結果：

上傳 mid file 並評分

選擇檔案 未選擇任何檔案 提交評分

音樂真實度評分: 90.1%

模型調整區

Add New Layer

Layer 1

embedding layer

Embedding Dimension: 1

Number of Embeddings: 512

Delete Layer

Layer 2

fully-connected layer

Input Dimension: 50

Output Dimension: 100

Delete Layer

Layer 3

transformer

Number of Heads: 8

Model Dimension: 512

Feedforward Dimension: 1024

Dropout: 0.1

Delete Layer

Apply

生成的PyTorch程式碼

```
import torch
import torch.nn as nn

model = nn.Sequential(
    nn.Embedding(num_embeddings=512, embedding_dim=1),
    nn.Linear(50, 100),
    nn.Transformer(d_model=512, nhead=8, dim_feedforward=1024, dropout=0.1),
)
```

Midi 檔案展示區

0:00 / 0:29

0:00 / 0:33

0:00 / 2:12

0:00 / 0:28

0:00 / 1:53

0:00 / 1:47

生成音樂直播區

ai音樂生成

到以下平台觀看: YouTube

訓練實時監控

Learning Log

[Epoch 1/1000]	[Batch 100/5066]	[D loss: 0.932387]	[G loss: 0.163629]
[Epoch 1/1000]	[Batch 200/5066]	[D loss: 0.641731]	[G loss: 0.533460]
[Epoch 1/1000]	[Batch 300/5066]	[D loss: 0.404271]	[G loss: 0.711196]
[Epoch 1/1000]	[Batch 400/5066]	[D loss: 0.600578]	[G loss: 0.644716]
[Epoch 1/1000]	[Batch 500/5066]	[D loss: 0.371243]	[G loss: 0.719220]
[Epoch 1/1000]	[Batch 600/5066]	[D loss: 0.387324]	[G loss: 0.512547]
[Epoch 1/1000]	[Batch 700/5066]	[D loss: 0.373870]	[G loss: 1.180949]
[Epoch 1/1000]	[Batch 800/5066]	[D loss: 0.412254]	[G loss: 0.747303]
[Epoch 1/1000]	[Batch 900/5066]	[D loss: 0.254941]	[G loss: 1.090486]
[Epoch 1/1000]	[Batch 1000/5066]	[D loss: 0.309474]	[G loss: 1.114308]
[Epoch 1/1000]	[Batch 1100/5066]	[D loss: 0.266445]	[G loss: 1.220899]
[Epoch 1/1000]	[Batch 1200/5066]	[D loss: 0.179959]	[G loss: 1.304349]
[Epoch 1/1000]	[Batch 1300/5066]	[D loss: 0.197302]	[G loss: 1.130355]
[Epoch 1/1000]	[Batch 1400/5066]	[D loss: 0.224983]	[G loss: 1.361877]
[Epoch 1/1000]	[Batch 1500/5066]	[D loss: 0.239294]	[G loss: 1.793091]
[Epoch 1/1000]	[Batch 1600/5066]	[D loss: 0.162379]	[G loss: 1.344646]

智能後空翻教練：基於 LSTM 之人體動作識別

Intelligent Backflip Coach:

LSTM-based Human Action Recognition

指導教授：曾繁勛

專題成員：古孝正、吳出右、陳念誠、賴虹橋

開發工具：Python 3

測試環境：Ubuntu 22.04 LTS x86_64

一、簡介：

隨著人體骨架辨識技術日益成熟以及動作識別（Action Recognition）應用的快速發展，這些技術在多個領域展現廣泛的應用前景。本專題之所以選擇後空翻姿勢矯正為主題，係因專題組員自身在練習後空翻時曾遇到瓶頸，且由於領域的稀缺性，臺灣各縣市的教學資源也參差不齊，冀望透過本專題研究，實現智能後空翻教練，縮短教學資源的城鄉差距。在本專題中，我們旨在探討如何將人體骨架辨識與動作識別技術相結合，再運用長短期記憶（Long Short-Term Memory, LSTM）神經網路，開發一個智能後空翻教練系統，目標利用 LSTM 的時序數據處理能力，準確地識別和分析後空翻動作，從而為使用者提供技術上的指導。

本專題目標的骨架辨識系統之選擇基準是「倒置」（意即頭下腳上）情況下是否正確，因為大部分現有的骨架辨識系統，在人體倒置的情況下容易判斷錯誤，在各種測試後，我們最終使用 ViTPose 作為此次專題的骨架辨識模型。ViTPose 是一個基於 Transformer 架構的神經網路，與傳統的卷積神經網路（Convolutional Neural Network, CNN）不同的是，ViTPose 可以藉由追蹤序列資料中的關係，使模型對複雜的姿勢有更強的理解力。

蒐集正確的骨架資料集後，我們使用 LSTM 作為訓練模型，係因在執行動作識別時，必須讓模型記憶影片的動作後，再判斷該次後空翻是否正確無誤。LSTM 是一種特殊的遞迴神經網路（Recurrent Neural Network, RNN），解決了梯度消失的問題，有強大的記憶能力及長短期記憶的特性，能有效地記錄並分析整個後空翻動作過程，提供準確且詳細的技術指導，幫助用戶改進後空翻動作。

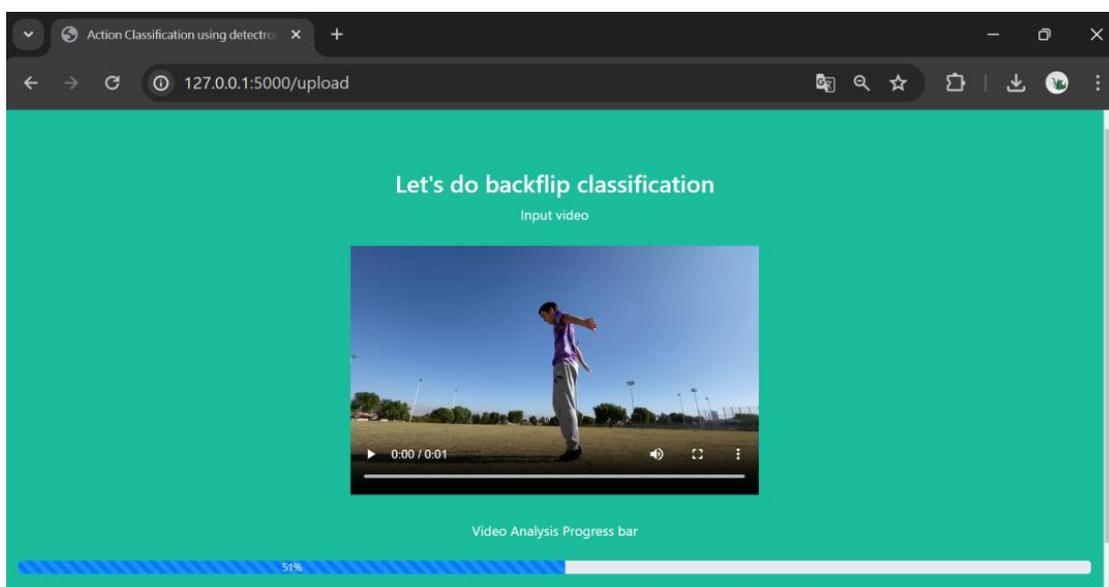
圖一為程式架構圖：



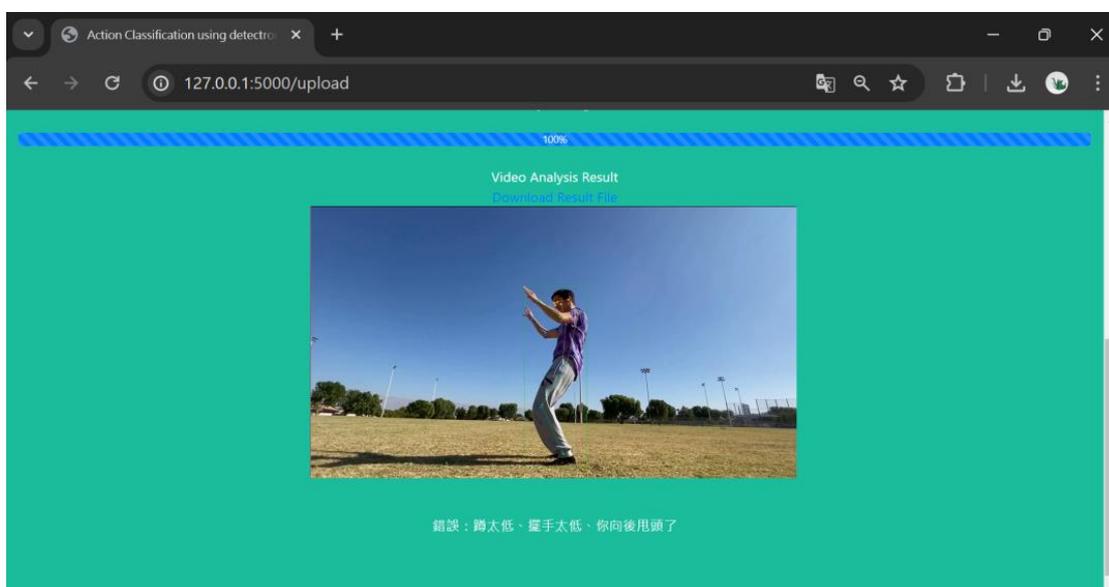
圖一：程式架構圖

二、測試結果：

測試者可以在網站上傳自己的影片（如圖二所示），並在本專題建置的網站上獲得此次後空翻的動作建議，如圖三所示。



圖二：使用者上傳影片後，程式的執行畫面



圖三：分析後輸出的後空翻錯誤

使用 YOLOv9 影像辨識系統偵測火災

Traning YOLOv9 model for detection of fire and smoke

指導教授：蔡佩璇

專題成員：魏宇弘

開發工具：python3.9.19, YOLOv9, Pycharm, Roboflow

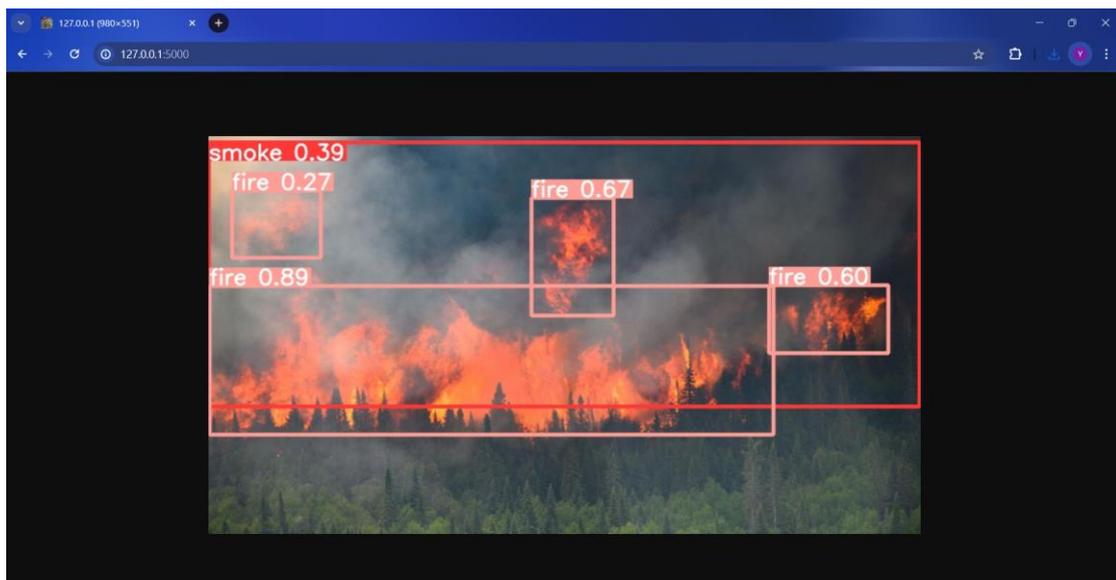
測試環境：Windows 11

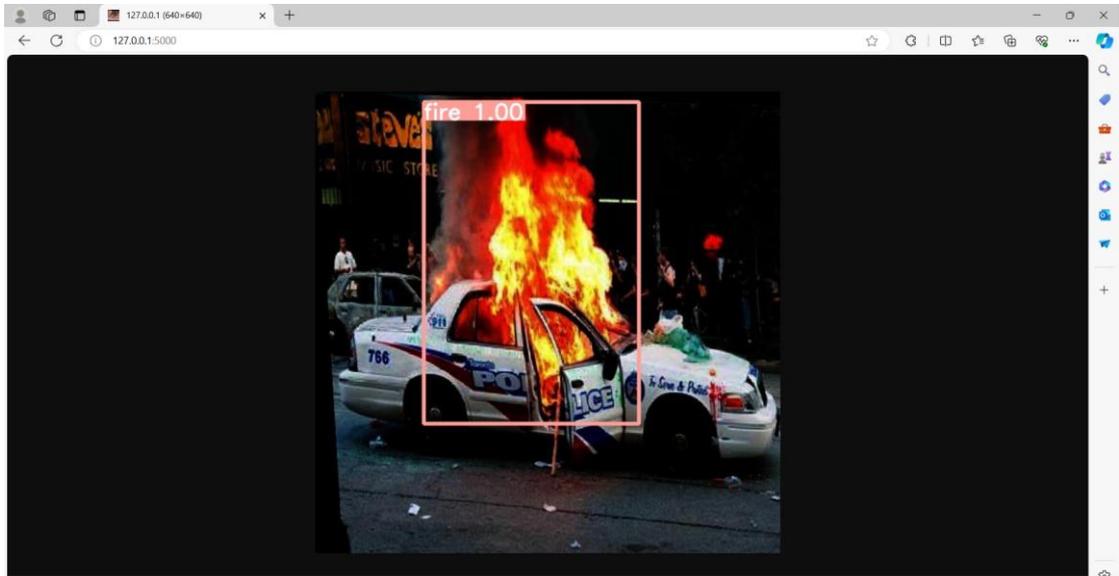
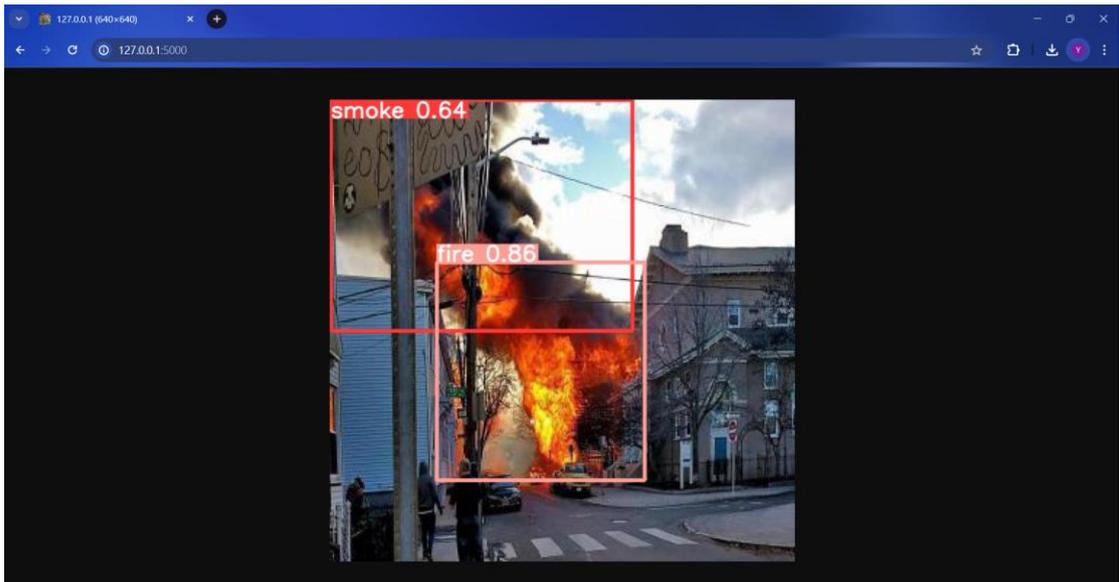
一、簡介：

最近台灣的火災頻繁發生，火災不但會帶走財物更有可能給人帶來傷害，若能在火災發生前便防患於未然，則能夠更好地去逃生或撲滅火源，所以我想利用影像偵測的方式來偵測火災中必備的兩個元素:火跟煙，能夠及時判斷出兩者是否存在，不但能夠快速判斷火跟煙的來源，也能為逃生多爭取一點時間。這次專題中我使用了今年剛發布不久的 YOLOv9模型，且配合 D-fire dataset 跟 Roboflow 來訓練。

二、測試結果：

訓練出的模組用 Flask 製作成網頁，可偵測圖片或是影片。





智慧餐點分析系統：利用 CV 與 SAM 進行食物分割與營養素分析

Intelligent meal analysis system : Utilizing CV and SAM for ingredients segmentation and nutrients analysis

指導教授：蔣榮先

專題成員：蘇致連

開發工具：VScode, Colab, Docker

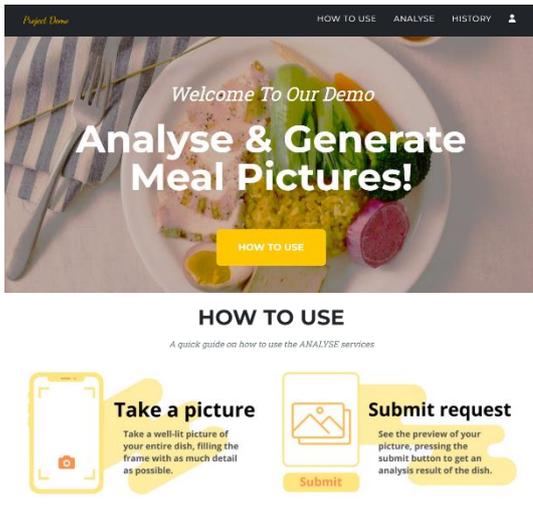
測試環境：python 3.7, pytorch 1.8, Django 5.0.3

一、簡介：

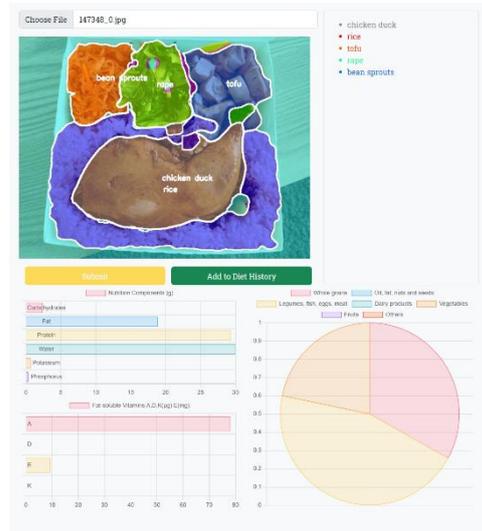
緊湊的生活步調使外食幾不可避，然而現有的飲食紀錄 app 多需掃描食物包裝條碼或是僅能辨識食物整體而非各種食材，使用上無論是難以對付一般無條碼的餐點或受制於食物種類死板，皆稍嫌不便。本著促進國人飲食健康的期待，我們借力成大醫院腎臟科醫師及食安所營養師兩方之專業，嘗試建構快速易用的個人化餐點營養素分析工具。在呈現方面，考慮用戶裝置多變，專題選以網站提供飲食的分析及紀錄服務。實際操作時因個人化功能所需，使用者註冊時將填寫體重、年齡及性別等資料，日後只需登入即可。使用者登入後只需依網站指示上傳餐點圖片，稍等10~20秒後即可收到預測的食材列表、食材分割圖以及三張對腎臟健康較關鍵的營養素之估計圖表，最後，使用者可將該結果加入個人紀錄，方便日後檢視。在實作方面有(1)前處理、(2)語意分割以及(3)營養計算三部分。前處理採用 Unet 找出食物位置並在其餘部分套用高斯模糊，使圖片聚焦於食物處。語意分割部分採用 SETR-MLA 判斷食材種類並輔以 SAM 較佳的分割能力，兩者協力達成食材分割。營養計算部分則以 Contour 過濾分割雜訊，將食物的 Convex Hull 比對容器形狀以換算實際面積，接著由面積估計重量，最後依照營養資料計算該餐點的綜合營養。綜上所述，本專題期望以快速、易用為賣點，藉此分析紀錄工具提升國人注重健康飲食之意願。



圖一、專題運作流程



圖二、封面及使用說明



圖三、分割圖與營養素圖表

Add Meal to History

Ingredients

+ Add New

Ingredient	Weight (g)	Actions
chicken duck	175.78	
rice	119.4	
tofu	79.32	
rape	48.01	

Date and Time

May 18, 2024 at 12:19 PM

圖四、紀錄功能



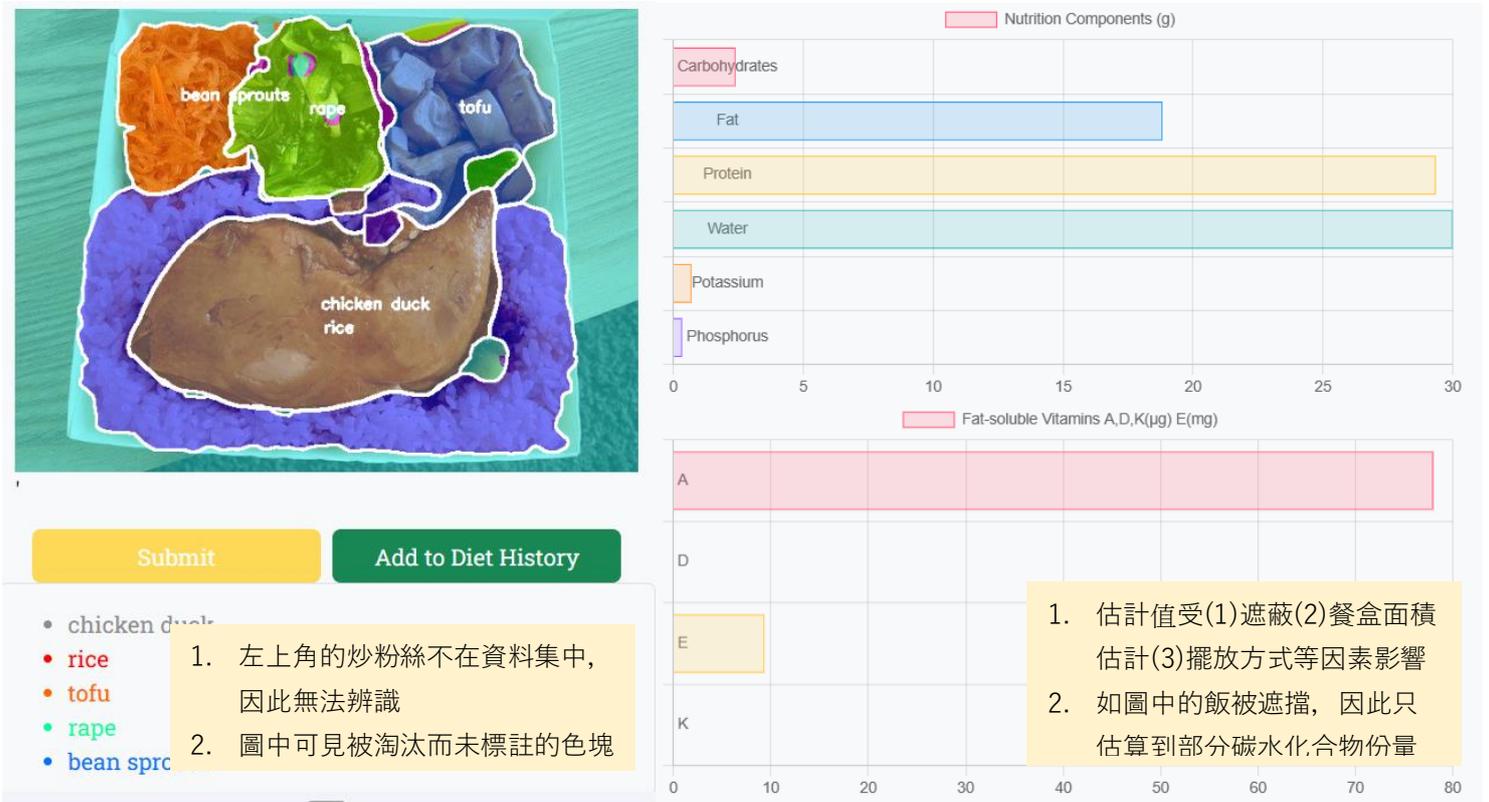
二、測試結果：

由於各種便當之容器形狀較規範，食物面積、重量較易估計，故以便當作範例

1. 使用效果：

手動記錄一餐營養約需3~5分鐘，透過專題網站自動處理則需等待 GPU 運算 10~20秒，相較之下速度提升不少。

2. 營養分析精準度：



圖五、營養分析精準度

Debiasing Facial Generation and Reconstruction Tool

去偏見化之臉部生成和臉部重建工具

指導教授：蔣榮先

專題成員：黃霖均

開發工具：PyTorch, Stable-Diffusion-v1-5,
Deepface, torch-fidelity, clean-fid

一、簡介

As generative AI like diffusion models gain widespread use, ensuring fairness and mitigating harmful biases has become paramount. These powerful models can generate highly realistic synthetic data. However, if these synthetic data exhibit societal biases, it may risk perpetuating harmful stereotypes, exacerbating social divides.

As shown in Fig. 1, this project aims to mitigate bias in generative diffusion models by introducing a sensitive attribute learning indicator during training. For instance, the model takes a face image or a natural image as an input, while the model learns how to generate a human face only, and the indicator learns how to distinguish between a face and the others.

The indicator is characterized by non-binary attribute learning and lightweight structure. The indicator utilizes non-binary attribute learning to mitigate biases like gender by considering visual features without being constrained to binary notions, while its lightweight convolutional structure minimally impacts training time.

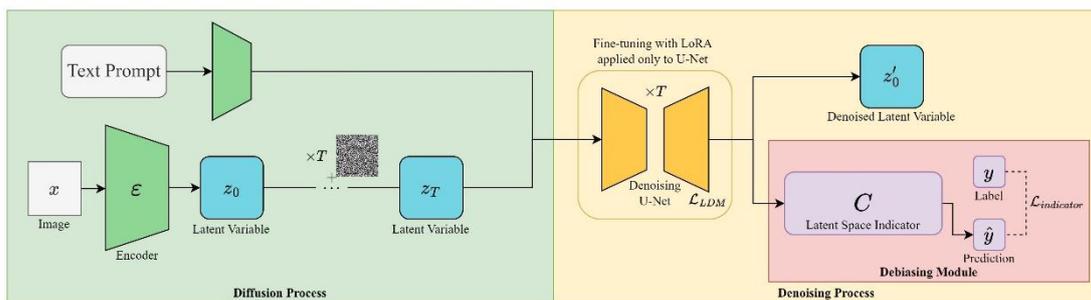


Fig. 1 Debiasing Diffusion Model. The indicator C outputs a value which serves as a key factor in the loss computation, thereby guiding the model's learning process.

This project is demonstrated in the Debiasing Facial Generation and Reconstruction Tool. Both a text prompt or a face contour can be input as a guidance to the model, and the model generates a corresponding face image.

二、測試結果

In our experiments, as shown in Table 1 and Fig. 2, we train models with different

debiasing intensity and different datasets, and generate 800 images to count the demographic composition with Deepface in these models.

Training Set Gender Ratio (Male : Female)	Debiasing Intensity	FD(↓)	FID(↓)	IS(↑)
0.25	0	0.14	<u>0.017</u>	3.125±0.242
	0.01	0.11	0.023	3.113±0.177
	0.05	<u>0.08</u>	0.025	<u>3.899±0.263</u>
	0.1	0.1	0.043	3.499±0.234
1	0	0.08	<u>0.015</u>	3.153±0.231
	0.01	<u>0.02</u>	<u>0.015</u>	3.590±0.273
	0.05	0.05	0.016	3.508±0.341
	0.1	0.03	0.026	<u>3.803±0.295</u>
4	0	0.30	<u>0.014</u>	<u>4.029±0.505</u>
	0.01	0.40	0.018	3.880±0.211
	0.05	<u>0.14</u>	<u>0.014</u>	3.971±0.302
	0.1	0.40	0.021	3.518±0.198

Table 1. We test Fairness Discrepancy (FD), Fréchet Inception Distance (FID), Inception Score (IS) of our models. A lower FD score indicates a fairer result, a lower FID score indicates a higher-quality image, and a higher IS indicates better quality and diversity in the generated samples. The best results are highlighted in bold on the table.

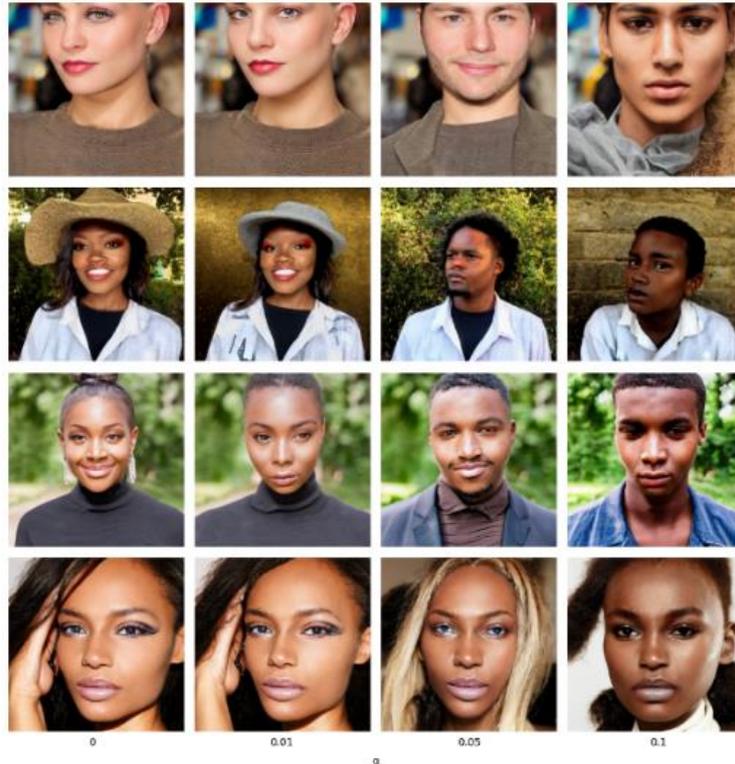


Fig. 2 The figure demonstrates the effects of different α values (intensity) on the debiasing process, using a model fine-tuned with a dataset comprising 20 male and 80 female images. Each row represents images generated with a consistent seed, showing the debiasing effects as α increases from left to right.

慢性腎臟病患者的 AI 智慧飲食推薦

AI Smart Diet Recommendation for Patients with Chronic Kidney Disease

指導教授：蔣榮先

專題成員：林威

開發工具：Python 3, Stable Diffusion2.1, LoRA

測試環境：Windows 10

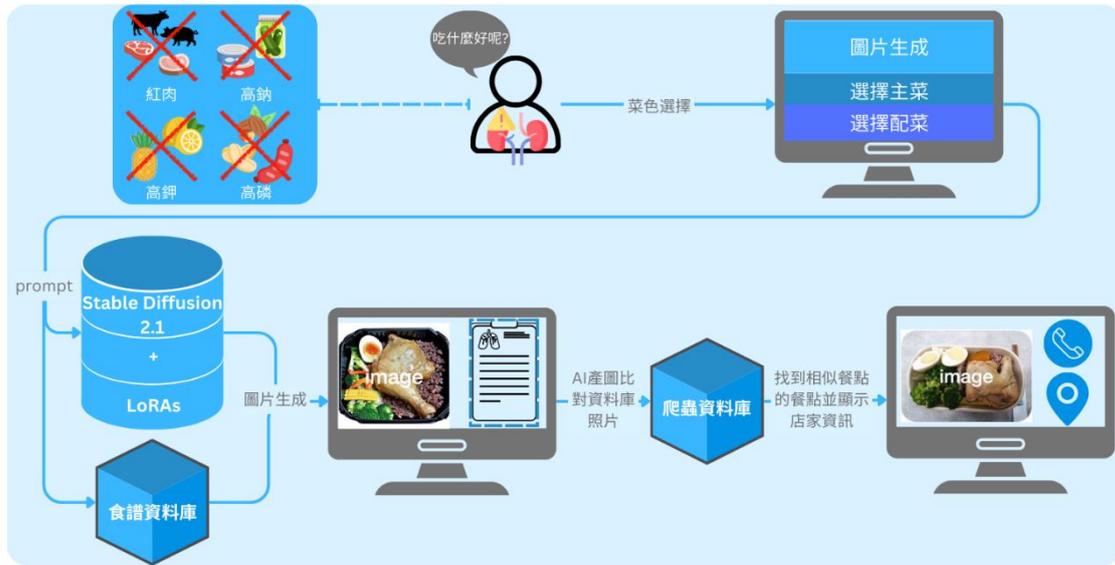
一、簡介：

飲食的選擇一直是慢性腎臟病(CKD)患者最大的苦惱，因為任何攝入的食物都會直接影響到體內毒素的代謝與腎臟衰退的歷程，所以吃什麼變成 CKD 患者重要的課題！為了提升腎臟病患者身心靈的福祉，也考量到 CKD 患者大多是老年人口，推薦餐點時直接給予**餐點圖片**會比文字描述來的更直觀與容易理解，因此我們使用**生成式 AI**生成圖片來向患者推薦餐點，而因為食物的種類繁多，所以產圖內容著重在**健康餐盒**為主。

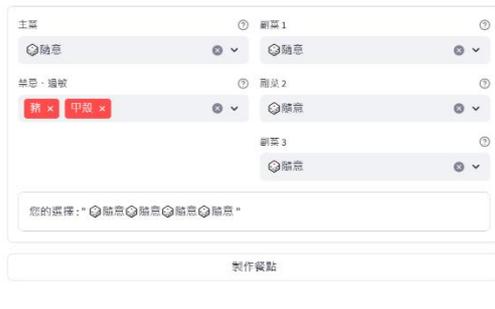
專題的**資料集(Dataset)**依據成功大學**食安所營養師**為**CKD 患者**設計的菜單進行蒐集，但是因為設計的菜色非常道地，網路上難以蒐集到相似的素材，因此資料集的健康餐盒照片皆是自行烹飪、擺盤與拍攝的。利用資料集來訓練多個 LoRA 模型(每一種一肉三菜的健康餐盒就訓練一個 LoRA 模型)，再將 **Stable Diffusion 2.1模型**結合訓練好的 LoRA 模型就可以進行生圖。此次專題已訓練 108個 LoRA 模型因此可以生成108種健康餐盒的圖片。

當得到 AI 生成的健康餐盒圖片時，不僅附上圖片中餐盒**主配菜的食譜**，以供 CKD 患者自行料理，更考量到**外食族**的需求，給予其推薦的**健康餐盒店家資訊**。在患者點選圖片後圖片會與**爬蟲資料庫**的健康餐盒照片進行**圖搜圖**並輔以**文字名稱搜尋**，提供使用者有販售相似餐盒的實體店家資訊，方便使用者前往購買或使用外送服務。

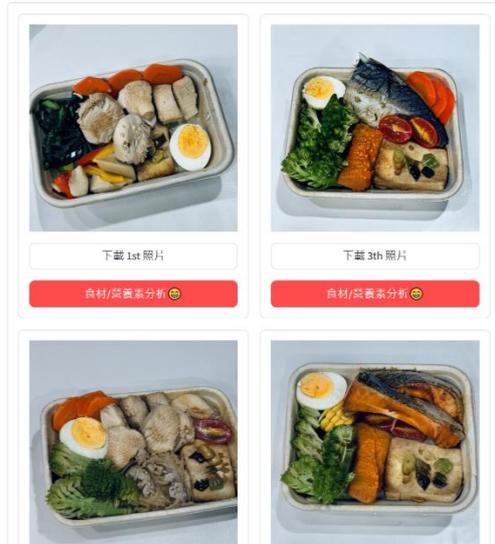
以下為系統架構圖：



二、測試結果：



AI 推薦的健康餐盒



3. Recipe

您可以參考食譜來製作餐點

<h4>主菜食譜</h4> <p><鯖魚></p> <p>食材：鯖魚</p> <p>醃料：鹽(少量)</p> <p>料理步驟：表面抹鹽，靜置5分起油鍋，皮朝下，乾煎至雙面焦黃即可</p> <p>https://cookpad.com/tw/%E9%A3%9F%E8%AD%9C%717256987-%E4%B9%BE%E7%85%8E%E9%AF%96%E9%AD%9A%E5%9C%A8%E5%8E%BD%E7%AEB1%E8%A3%A1%E8%AC%8E%E9%A3%9F%02?ref=search&search_term=%E9%AF%96%E9%AD%9A</p>	<h4>配菜 2 食譜</h4> <p><蒸南瓜></p> <p>食材：南瓜切小塊數塊 洋蔥數片 海鹽少量 奶油 1 小塊</p> <p>步驟：將除了奶油的料放入盤內放入電鍋，外鍋加水蒸約15分鐘，拿出後趁熱放入一小塊奶油，拌拌就可以開吃了，適合肚了餓的時候當點心，清爽無負擔。</p> <p>https://cookpad.com/tw/%E9%A3%9F%E8%AD%9C%717030116-%E6%B8%85%E8%92%B8%E5%8D%97%E7%93%9C%E6%B4%B8%E8%94%A5%E6%B3%A5?ref=search&search_term=%E8%92%B8%E5%8D%97%E7%93%9C</p>
<h4>配菜 1 食譜</h4> <p><傳統豆腐></p> <p>食材：板豆腐 400g 蒜末 1 瓣 薑末 少量 豆豉 1 小匙 蔥花 適量 醬油 2-3 大匙 米酒 1 大匙 二砂 少許 水 60~80ml</p> <p>步驟：豆腐沖洗一下後，泡水5~10分鐘，瀝乾水份備用。板豆腐洗淨切成適當大小，鍋中適量油溫，用半煎炸方式，煎至豆腐表面微金黃，將板豆腐推至鍋邊，在鍋中放入蒜末、薑末、豆豉爆香，(用鍋中煎豆腐的殘油，若還量不夠，可再補些油，)加入醬油、米酒、二砂，攪回板豆腐一起煎煮，煮至醬汁收乾至喜歡的程度。(可選擇煮透翻面，讓板豆腐均勻吸附醬汁。)將板豆腐盛起，鍋中平香料、醬汁淋於上面，撒上蔥花即完成。</p> <p>https://cookpad.com/tw/%E9%A3%9F%E8%AD%9C%717131416-%E9%86%AC%E7%87%92%E6%9D%BF%E8%B1%86%E8%85%90?ref=search&search_term=%E9%86%AC%E7%87%92%E8%B1%86%E8%85%90</p>	<h4>配菜 3 食譜</h4> <p><花椰菜></p> <p>食材：預備食材 花椰菜 1 朵 青蔥絲 2 朵 胡蘿蔔 少許 自製蒜油 少許 鹽 1/2 茶匙 黑胡椒粉 少許</p> <p>步驟：花椰菜、胡蘿蔔川燙後備用，青蔥蒜末成適量大小鍋中不放油先將青蔥蒜末煎炒半香，加入已燙熟的花椰菜、胡蘿蔔和蒜油拌炒一下加入少許的熱水讓花椰菜可以煮得一點，隨即調入鹽、黑胡椒粉拌勻即可</p> <p>https://cookpad.com/tw/%E9%A3%9F%E8%AD%9C%722564646-%E5%B7%A8%E7%82%92%E8%8A%B1%E6%A4%B0%8F%9C?ref=search&search_term=%E8%8A%B1%E6%A4%B0%8F%9C</p>

基於檢索增強生成的精準資訊擷取系統

Answer Briefly and Precisely: A Segment-based RAG Framework with Specialized Preprocessing Pipeline

指導教授：蔣榮先教授

專題成員：黃亮晨

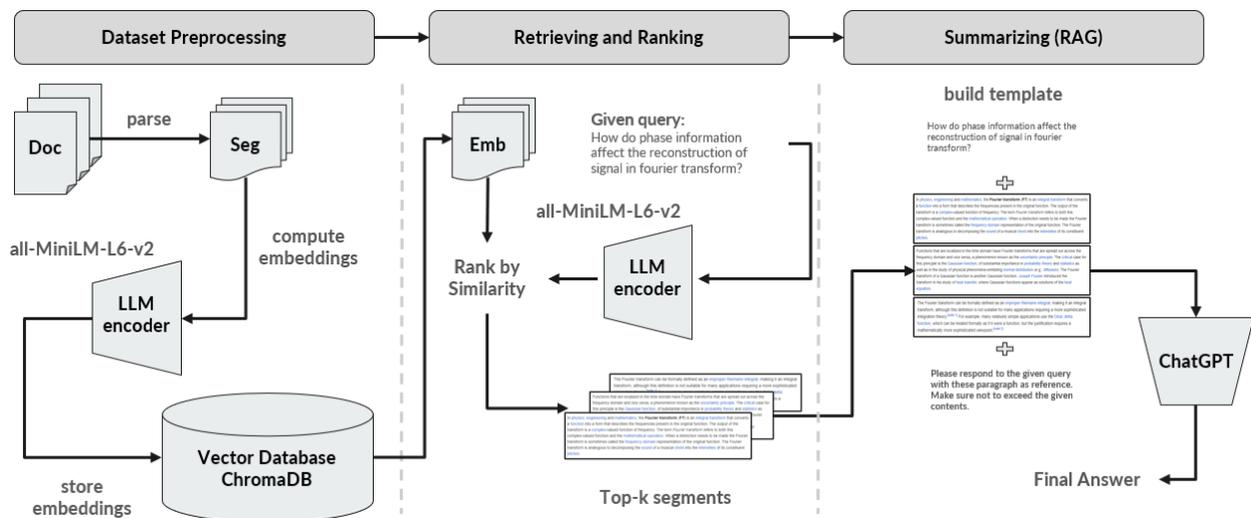
開發工具：Python3

測試環境：Windows 10

一、簡介：

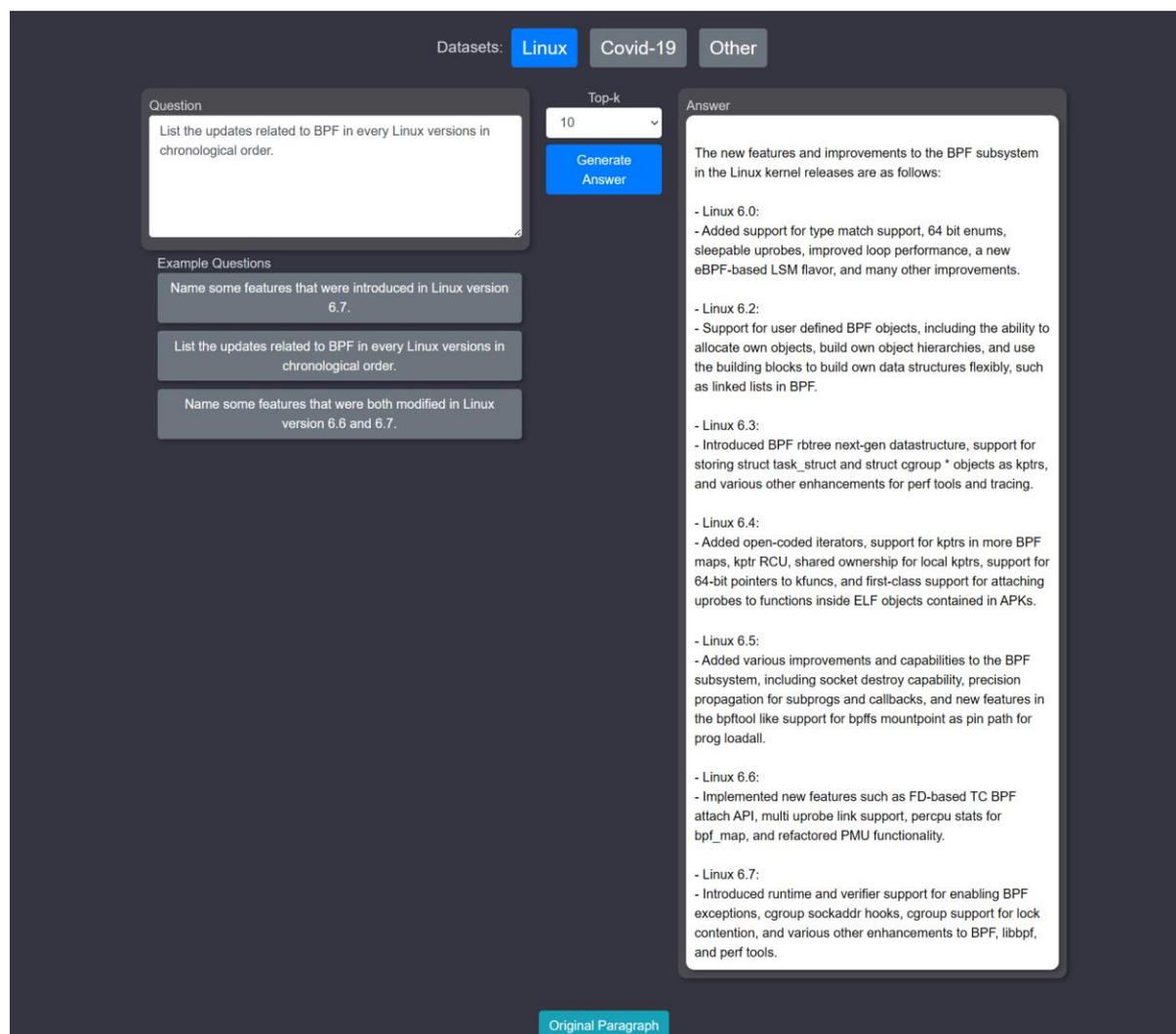
本研究改良傳統搜尋引擎資訊檢索方法（Information Retrieval）及大型語言模型（Large Language Model, LLM），設計出一個系統讓使用者只需輸入問題，不必親自檢視大量資料就能直接取得想要的回答，並利用檢索增強生成(Retrieval Augmented Generation, RAG)框架避免 LLM 的幻覺問題（hallucination），使其回答具有較高的可信度。具體來說，本研究的成果如下：

1. 透過新的前處理框架，改善過長參考文本所衍伸出的問題
2. 經由使用 RAG 框架，避免出現 LLM 的幻覺問題
3. 設計前後端介面，降低使用者的操作門檻



第一階段為資料收集與前處理，透過將長文件切分成簡短段落後，交由 LLM（all-MiniLM-L6-v2）算出每一個段落的特徵並儲存於資料庫中；第二階段為檢索與排序，輸入的問題會經過相同的 LLM（all-MiniLM-L6-v2）得出一個特徵，再透過比較與排序找出資料庫中與問題最接近的幾組特徵與其對應的文本；第三階段為總結，將問題與檢索出的段落放進設計過的模板中，交由 ChatGPT 進行最終問答。

二、 測試結果：



以上是使用者介面。首先，最上方 Datasets 可以選擇你想要使用的資料集。左邊 Question 欄位使用者可以自行輸入問題，或是直接選擇點選 Example Questions 中的預設問題。接下來，選擇 Top-k 的數值後按下 Generate Answer，右邊就會顯示生成的答案。最下面的 Original Paragraph 按鈕則是讓使用者在對答案有疑慮或是想看原文的時候，可以點選用來生成 Answer 的 Top-k 個原本的文章。

在這個測試中，我們的問題是請他照時間順序舉出與 BPF 相關的更新條例，可以看到答案有確實按照版本序排序列舉每個版本的更新。

使用 GPT 技術強化小語言模型

Applying GPT technique to enhance small language model

指導教授：謝孫源

專題成員：彭昇哲

測試環境:Windows11

torch version:2.1.2+cu121

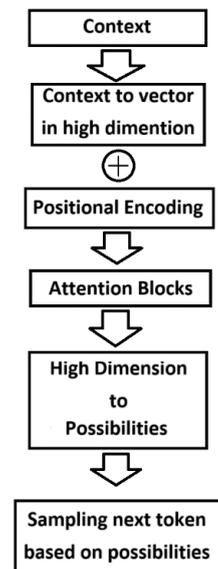
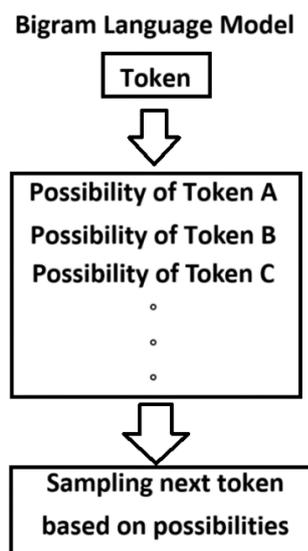
一、簡介

在進行專題研究的過程中，我注意到現在的語言模型不強調底層的運作邏輯，我好奇 Attention 機制到底是如何改變語言模型的，因此我根據 attention is all you need、OpenAI GPT 架構，使用 Pytorch 從底層實作 attention 機制，來將一個已有的小語言模型強化到能學習文本後寫出類似的文章。在這個過程中，有重要發現，attention 機制的引入，會讓小語言模型有能力將語意相似 token 轉換成接近的向量

二、結果

小模型： 0.005929 M parameters

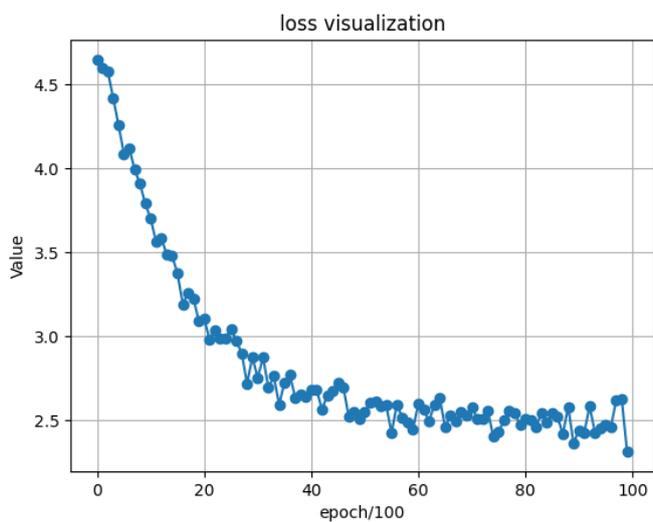
強化後： 0.209729 M parameters



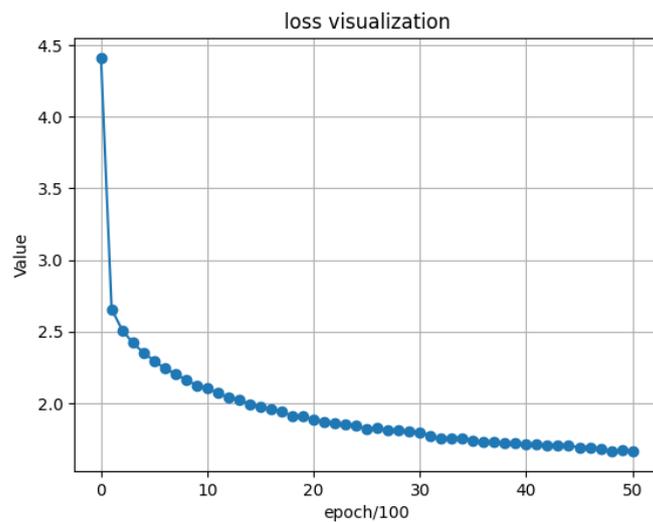
GPT2: Model size 137M params

Loss regression:

小模型訓練 loss:



強化後訓練 loss:



小模型文字生成(訓練後):

d:

at l so jur-

CHanoutlin h, m near dofer anBu d.

強化後文字生成(訓練後):

CAPULET:

Ty, disCome, know, your my pary, grand. Wellows.

DUKE VINCENTIO:

Lay unown should movence a they painer,

Web and Database Design for Skeletal Disease

Prediction Model

骨骼疾病預測模型網頁資料庫建置

指導教授：謝孫源

專題成員：黃鈺軒

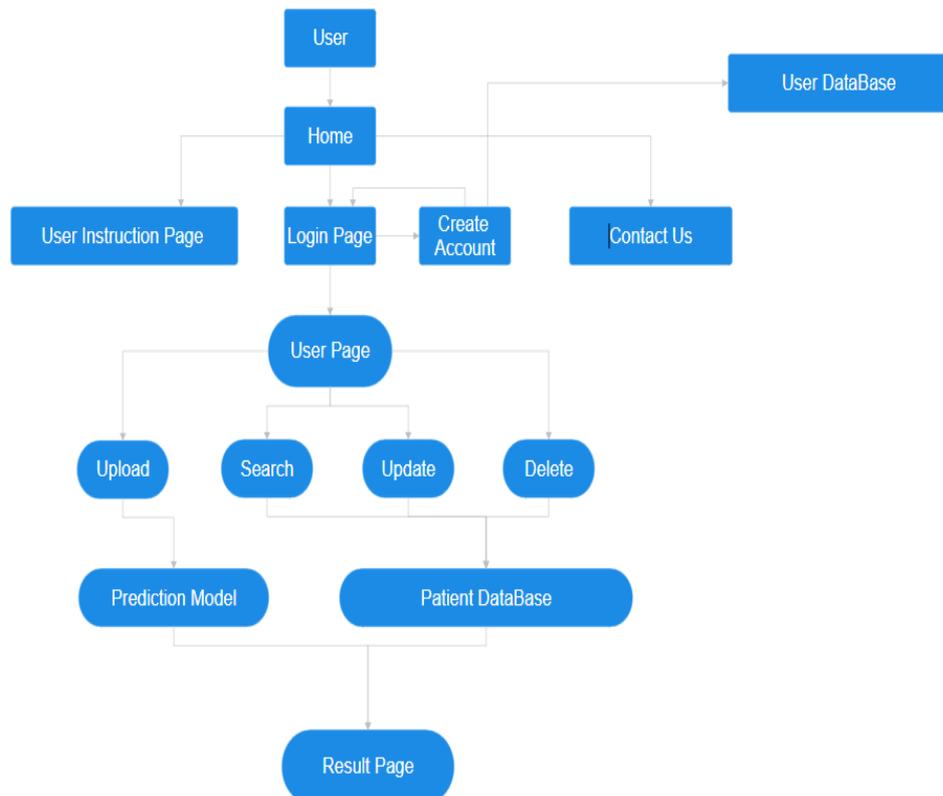
開發工具：PHP, SQL, html, CSS, python,
JavaScript

測試環境：Windows 11

一、簡介：

基於 AI 骨骼疾病預測模型開發的網頁，作為使用者導向的網路介面，讓使用者可以輕易使用這個模型。網頁進入預測後會將病人實際資料同步到資料庫，並將輸出結果(預測脊椎側彎的三個角度)一併上傳，方便日後統計資料需求。過程中實現了，AI 骨骼疾病預測模型的人機互動，和資料雲端化。

以下為系統架構圖：



二、測試結果：

登入前:

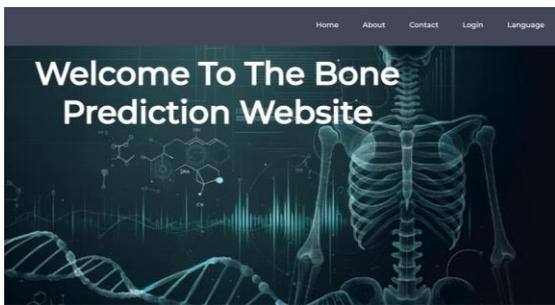
首先進入網頁首頁(圖一)，會看到上方有個 bar，分別可以連到未登入成功前的功能，Home 會跳回首頁，About 會顯示網頁的使用方式，Contact 會顯示如何聯絡我們，Login 會跳出一個 Dropdown Bar，可以選擇 Login 或 Create Account，分別會跳到 Login 介面，Create Account 介面。

登入:

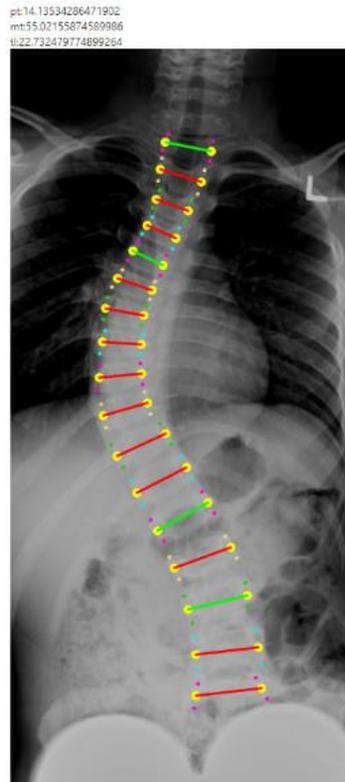
創建帳號時，email 必須符合格式，password 必須含有英文數字。創建成功後測試資料有被存到資料庫，密碼有經過 hash 才存入資料庫，輸入相同的 email 密碼，無法再重新創建第二個帳號。接著用剛剛創建的帳密可以成功登入進入使用者介面(User Page)。

登入後:

第一個測是從使用者介面(User Page)回前頁後登入介面(Login Page)能夠刷新，清除輸入的帳號密碼，第二個測試從使用者介面(User Page)Logout 後跳回登入介面(Login Page)，回前頁無法再回到使用者介面(User Page)，避免了利用回前頁來進入使用者介面(User Page)。接著是上傳新資料(Upload)，圖片成功傳入模型，並輸出結果(圖二)，最後將結果回傳資料庫，搜尋資料 (Search)，輸入病歷號，搜尋顯示現有的資料。更新資料 (Update)，更新現有資料，輸入病歷號和想更新的資料，成功更新資料庫。刪除資料 (Delete)，輸入病歷號，將此病例號的資料從資料庫中刪除。



圖一



圖二

基於深度學習的3D 成年鼠大腦分割

3D Adult Mouse Brain Image Segmentation

指導教授：謝孫源 王建堯

專題成員：蔡鎔澤

開發工具：Python、TensorFlow、PyTorch

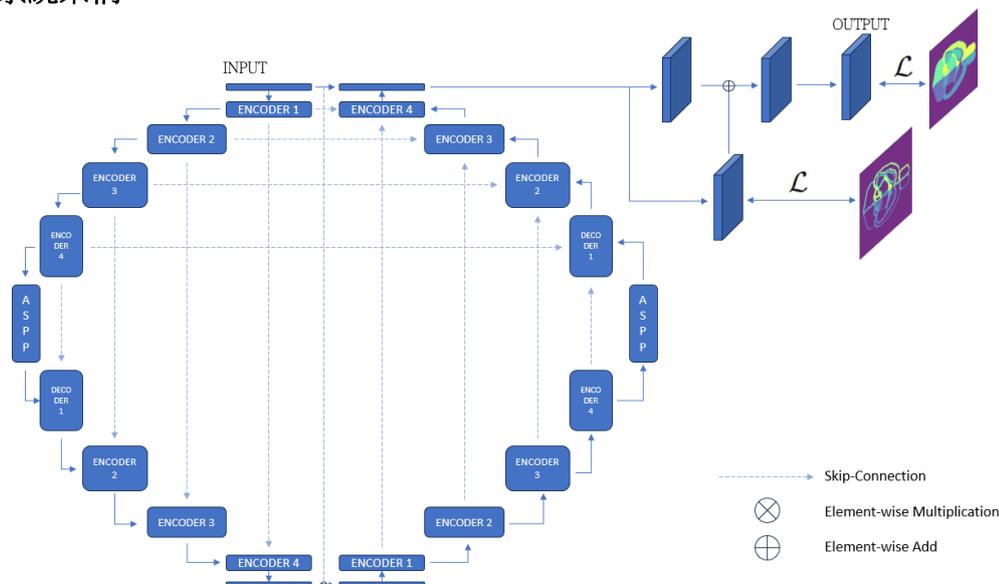
測試環境：Ubuntu 22.04.1 LTS

一、簡介：

去顱骨和腦區分割對於成年鼠大腦正電子發射斷層掃描（PET）的精確定量分析至關重要，但基於深度學習的統一解決方案——尤其是在 Spatial Normalization（SN）方面——則是 DL-based 影像處理中的一個挑戰性問題。

在專題中，使用基於 inverse Spatial Normalization（iSN）和 deep CNN 模型生成去顱骨 Mask 和腦特定感興趣區域（VOIs），從而建立無需 SN 的 template-based VOIs 方法。在各項 Metrics 中與傳統 VOIs 方法沒有顯著差異，解決了傳統 SN 方法中因半自動導致的評分者間信度與時間成本過高等問題。

系統架構



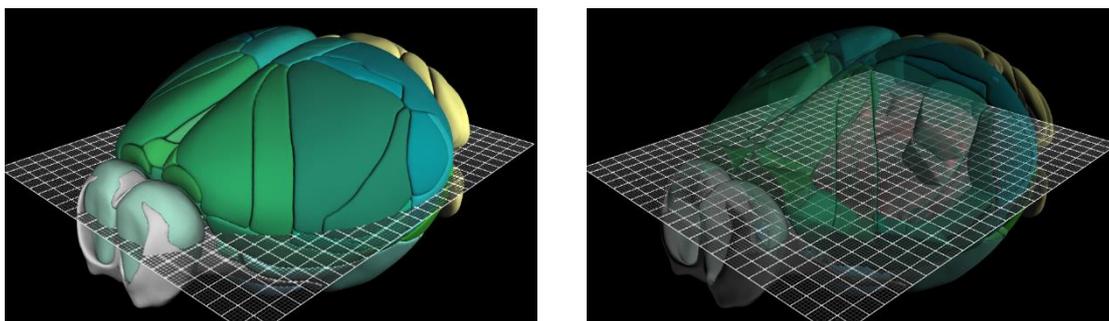
觀察問題

1. 成年鼠腦區分割的資料較難取得，無法訓練擁有大量參數的3D UNet。
2. 進行分割時，三個維度的資訊都相當重要，使用2D UNet 會導致丟失其中一維的訊息。
3. 傳統 Loss Function 多為基於分布與基於區域的損失函數，無法懲罰邊界預測不準確問題。
4. 預測時較小的腦區消失。

問題解決

1. 通過 Shift, Rotation, Shear 等方式增加訓練資料，以解決資料量較少的問題。
2. 將連續的2D 影像切片疊在一起作為輸入影像，輸入2D 的網路架構，既能保留第三維的連續空間訊息與也可以減少使用3D 網路架構而產生的參數量。
3. 使用 ResNet Block 使得網路更好訓練。
4. 使用 ASPP 與 SE block 增加感受野與減少冗餘信息並傳遞最相關的信息。

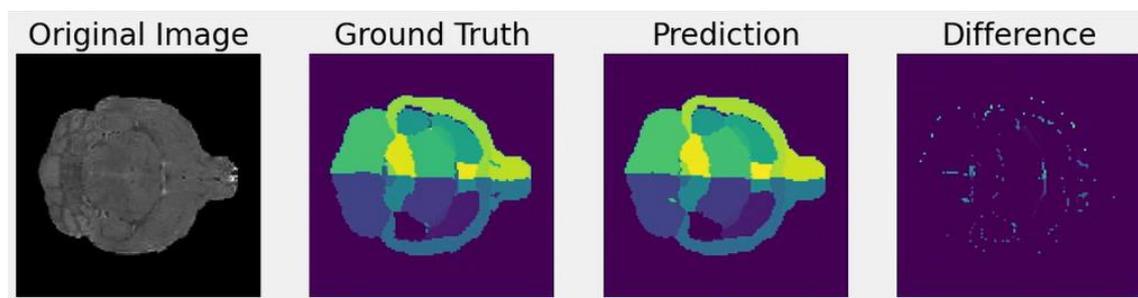
- 通過使用 Multi-Task Learning，利用輔助任務透過基於邊界的 Loss Function 學習邊界訊息以輔助主任務邊界不準確的問題。



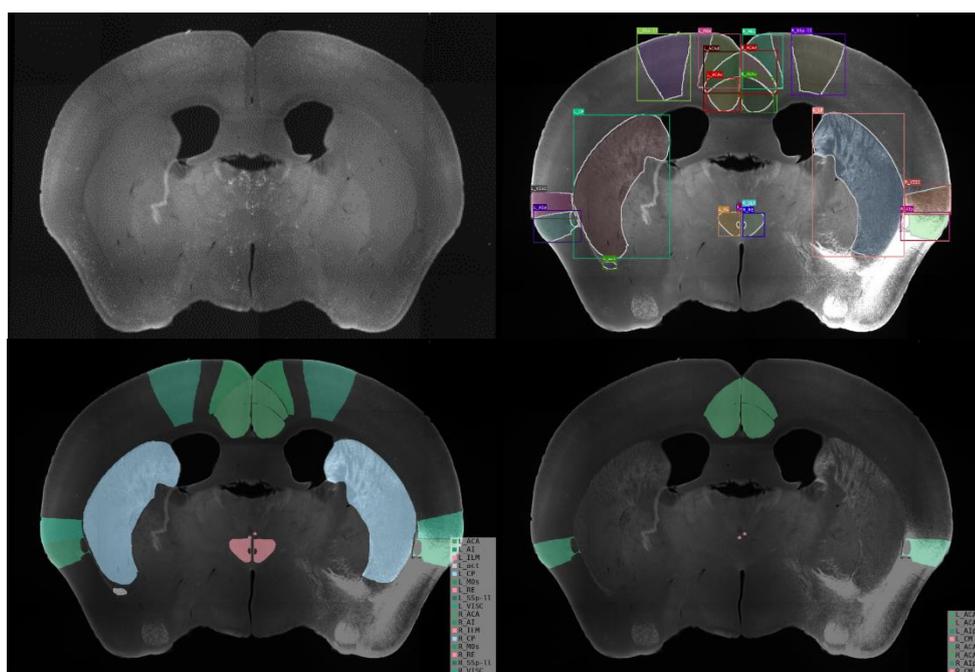
上圖為參考 Allen Brain Atlas

二、測試結果：

相較於 Baseline 中 0.84 的 Dice coefficient，通過 Propose 上述方法改善了邊界不準確的問題，並將 Dice coefficient 提升至 0.96。



(上圖左一) 輸入影像、(上圖左二) GT、(上圖右二) 通過輸入影像預測的腦分割區域、(上圖右一) 預測結果與 GT 的差



(左上) 輸入影像、(右上) 通過輸入影像預測的 VOIs、(左下) Level 1、(右下) Level 2

行人重識別在真實世界之應用

Evaluation of Person Reidentification (Re-ID) techniques with Real-World data

指導教授：藍崑展

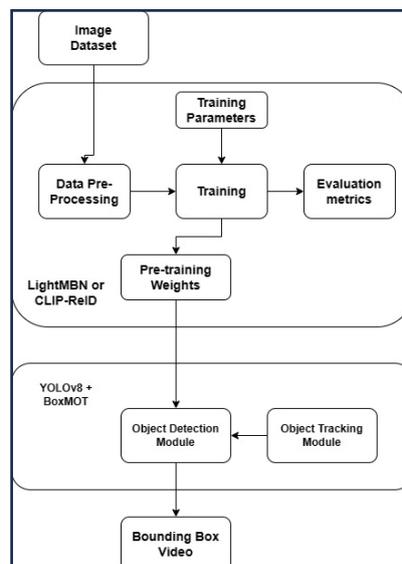
專題成員：張奕翔

開發工具：Vim, Conda, PyTorch, YOLOv8

測試環境：Linux, Windows 10

一、簡介：

Person Reidentification (Re-ID) is an increasingly crucial task in computer vision which is used to enhance security and surveillance systems by accurately tracking multiple individuals across different camera perspectives. This is done through feature extraction of the entire appearance of an individual, including clothing, and pose, to reidentify them in different locations or times. While many Re-ID techniques have been proposed with State-of-the-art (SOTA) performance, however, many of those performances are based on controlled environment benchmark tests and lacked real-world applicability. This project aims to address this gap by evaluating a few SOTA Re-ID techniques to observe their effectiveness and practical limitations by feeding them real-world input data. The models will be trained using both benchmark and custom datasets, and evaluated on precision and training time. The resultant Re-ID weights will be leveraged by an object detection, and tracking model on multiple custom real-world scenario video data to produce a bounding box video with unique identifiers for each individual. By analysing the Re-ID models' performance on real-world data, this project aims to identify which technique is more suitable for practical security applications requiring accurate person tracking.



二、 測試結果

Using standard benchmark tests such as Market1501, LightMBN outperforms CLIP-ReID as it produces good average and rank-1 results while also being fairly efficient in training time. Similarly, when training YOLOv8 on the resultant Market1501 weights on real-world video data such as Figure 2 below (Market environment), LightMBN is able to produce promising results, showing most visible individuals to the camera while needing less computation time than CLIP-ReID. However, when training on smaller sized custom datasets, CLIP-ReID outperforms LightMBN on average precision metrics due to the leveraging of CLIP's pretrained features.

When trained on real-world data with higher levels of occlusion and noise, both CLIP-ReID and LightMBN struggle with performance, and the resultant bounding box video struggles to identify the individuals. This is due to the limited amount of feature information that can be extracted from noisy, occluded data, and as a result both models are unable to make accurate image classifications.



Figure 2: Sample bounding box video on street camera footage trained on CLIP-ReID

Blood Pressure Estimation Using PPG Signal

指導教授：藍崑展

專題成員：李侖陞

開發工具：Python, Tensorflow

測試環境：Jupyter Notebook

一、簡介：

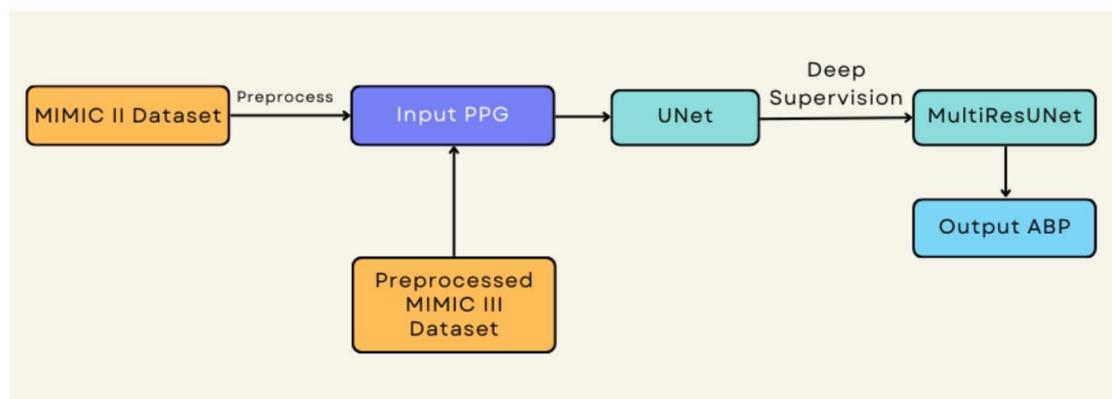
Cardiovascular diseases remain one of the leading causes of mortality, claiming countless lives annually. While continuous blood pressure monitoring is a promising solution, current methods often require invasive procedures, leading to complications and reliability issues. Non-invasive techniques, on the other hand, typically suffer from accuracy problems.

In response to these challenges, my project aims to develop a non-invasive method for estimating continuous arterial blood pressure (ABP) waveforms using Photoplethysmogram (PPG) signals. I leverage the power of deep learning to overcome the limitations of traditional approaches, which often rely on handcrafted features and ideally shaped PPG signals.

My method, PPG2ABP, employs a two-stage cascaded deep learning framework to estimate ABP waveforms from PPG signals, maintaining the integrity of shape, magnitude, and phase. Remarkably, this approach also excels in deriving Diastolic Blood Pressure (DBP), Mean Arterial Pressure (MAP), and Systolic Blood Pressure (SBP) values from the estimated ABP waveforms, outperforming existing methods. Moreover, the system meets the stringent criteria of the BHS (British Hypertension Society) and AAMI (Association for the Advancement of Medical Instrumentation) standards, achieving Grade A for both DBP and MAP.

For this endeavor, I utilize Python and Tensorflow, with Jupyter Notebook as the testing environment. By incorporating additional data, I aim to further enhance the accuracy and reliability of the PPG2ABP model.

以下為系統架構圖：



二、測試結果：

The training of the PPG2ABP model yielded impressive results, accurately estimating continuous arterial blood pressure (ABP) waveforms from Photoplethysmogram (PPG) signals. The model achieved a mean absolute error of 4.604 mmHg for the ABP waveform, maintaining the integrity of shape, magnitude, and phase.

In addition, the model excelled in calculating key blood pressure metrics. It achieved mean absolute errors of 3.449 ± 6.147 mmHg for Diastolic Blood Pressure (DBP), 2.310 ± 4.437 mmHg for Mean Arterial Pressure (MAP), and 5.727 ± 9.162 mmHg for Systolic Blood Pressure (SBP). These results not only surpass existing methods but also meet the stringent criteria of the British Hypertension Society (BHS) and the Association for the Advancement of Medical Instrumentation (AAMI), achieving Grade A for both DBP and MAP. This high level of accuracy and reliability underscores the model's potential for practical, non-invasive blood pressure monitoring.

Grapycal: 人性化的圖形化程式語言

指導教授：蘇文鈺

專題成員：程品奕

開發工具：Python、Web

測試環境：Ubuntu 22.04 (WSL2)

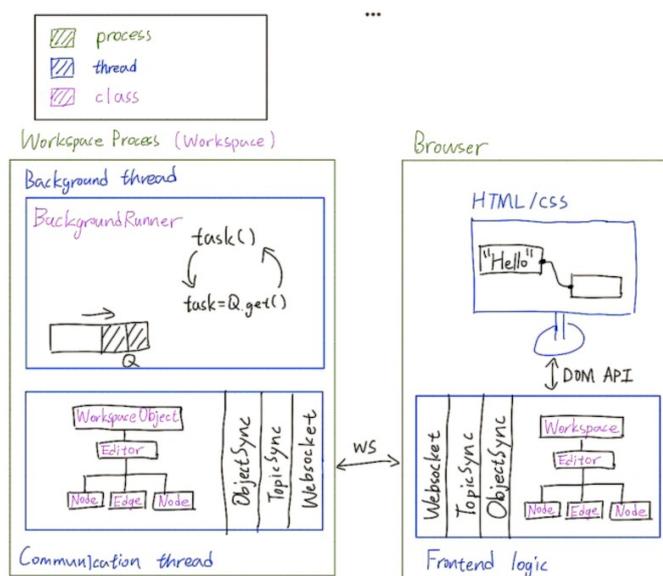
一、簡介：

當 AI 研究者在實作 AI 模型時，常常遇到以下困境：雖然語意上模型結構和操作流程並不複雜，但若要把它們實際用 Python 寫出來並除錯，就變成一件細碎繁瑣的工作。這意味著一般的 Python script 在此類實驗性程式設計上，是不夠理想的人機介面。而 Grapycal 的目標正是彌補這項缺失。

Grapycal 是一種基於 Python 的通用視覺化程式語言。它提供基於網頁的程式編輯器，讓使用者以作圖的形式編寫程式、執行程式並獲得即時回饋。其擁有高互動性的前端，盡可能貼近人類的直覺及感知；同時以 Python 做為後端，繼承 Python 通用且強大的功能。

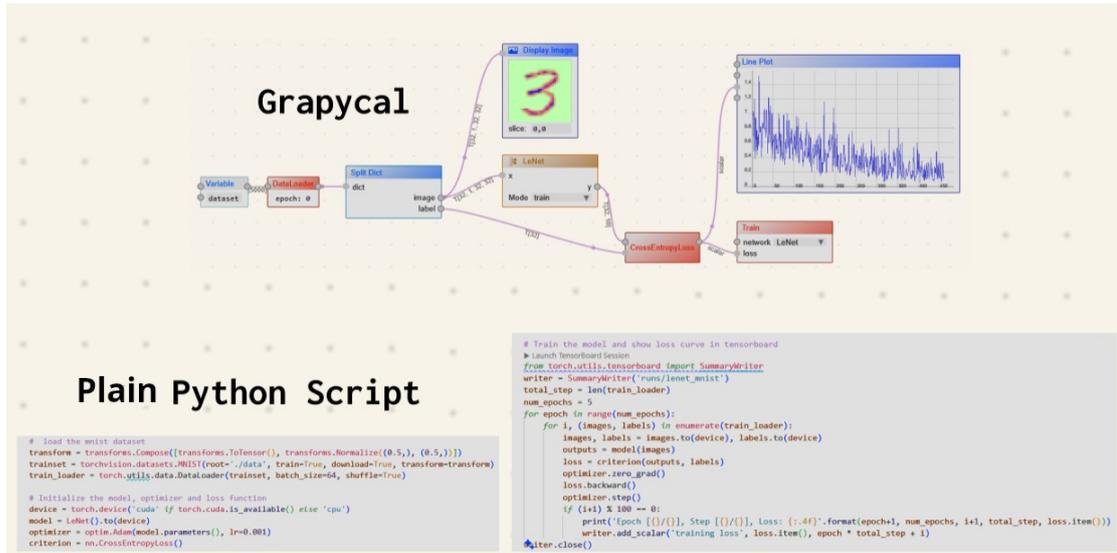
Grapycal 有助於進行實驗，包括訓練 AI、物理模擬、數據分析、電腦藝術等。這些實驗需要反覆調整參數，抽換模型的某些組件，同時觀察不同參數所產生的現象。然後，我們再使用人工評斷結合領域知識來推論出最佳模型或其他結論。在上述人與機器之間的來回迭代過程中，傳統的 Python script 或 Jupyter Notebook 會用起來很麻煩，使用者的思緒會常常被中斷。因此，我們需要 Grapycal 來幫使用者蓋一座橋，跨越無關且瑣碎的工作，把心力放在每次實驗的真正目標上。

以下為系統架構圖：

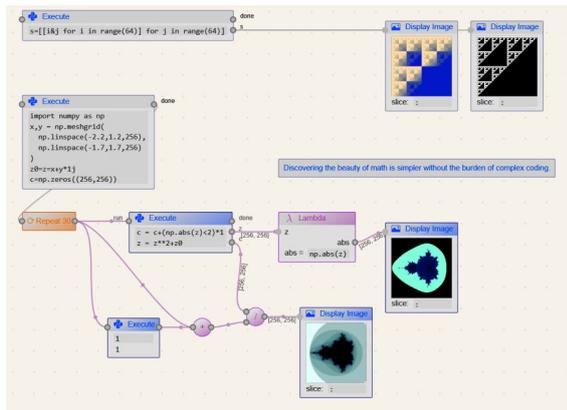


圖一：系統架構圖

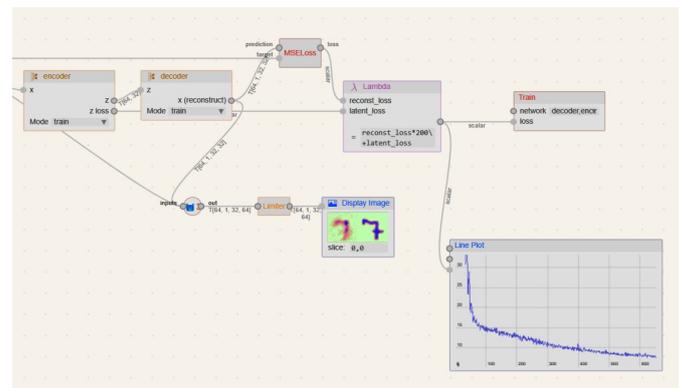
二、測試結果：



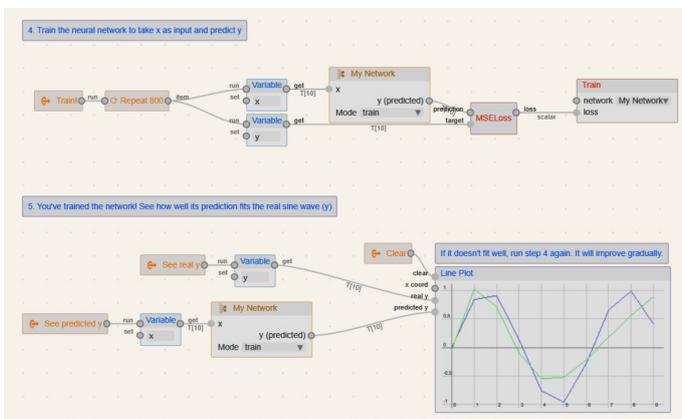
圖二：Grapycal 與 plain Python script 的比較 - 圖像分類器訓練



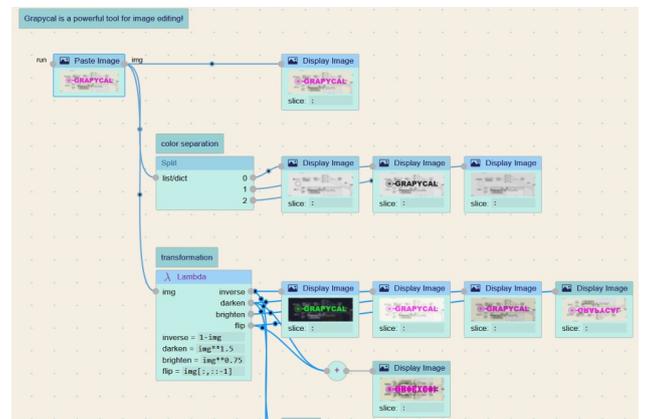
圖三：Grapycal 程式 - 繪製碎形



圖四：Grapycal 程式 - 訓練 VAE 模型



圖五：Grapycal 程式 - 資料擬合模型



圖六：Grapycal 程式 - 影像處理

以強化式學習解決裝箱問題之研究

Research for solving bin packing problem with reinforce learning

指導教授:蘇銓清

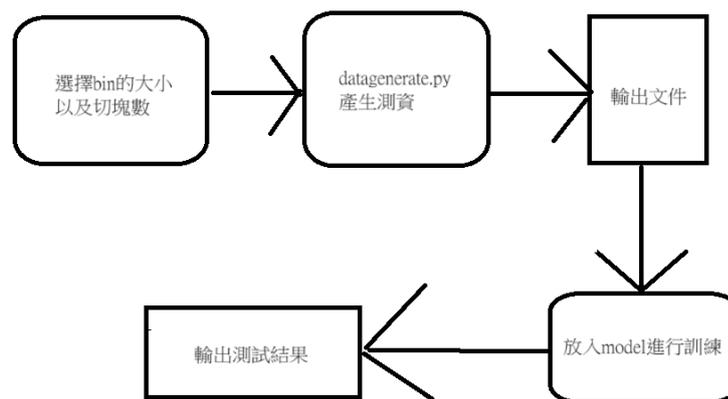
專題成員:陳穎睿、劉尚典

開發工具:Python,Pytorch,Colab

測試環境:Windows 11

一、簡介

裝箱問題(bin packing problem)，簡單說是將各種小盒子放入一大盒子內的問題，可用於電腦資源配置、運輸物流等多個方面，看似簡單，但實際上，在排的過程中需要注意的點有相當多，空間使用率、選擇速度等等，而現有的解決方法大多都以啟發式演算法進行，訓練素較慢且上限較低，因此，我們決定設計能以強化式學習去解決裝箱問題的方法，而我們使用的策略為 DQN，並配合 mask action，僅以當個盒子可放入的選擇去訓練，來達到提高空間使用率，並盡量達到將所有盒子成功放入的結果，而以下為流程圖



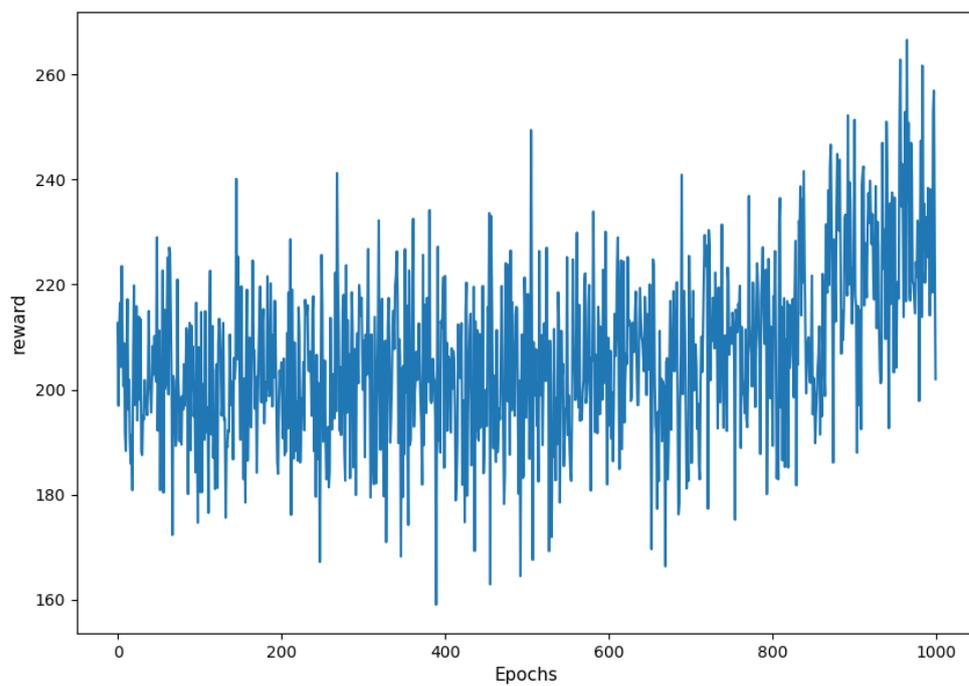
二、測試結果

以下為輸出和訓練結果

輸出:可看見盒子被逐漸填滿



Reward:可看到 reward 在僅僅 1000 個 epoch 便有顯著的上升，表示訓練有成果



心動分析-

基於深度學習的超音波左心室射血分數評估 Left Ventricle Ejection Fraction Assessment in Echocardiograms Using Deep Learning

指導教授：吳明龍

專題成員：郭沛蓉、陳帛愛、潘冠穎、鄭紹祺

開發工具：Python, TensorFlow, Pytorch, OpenCV, TFLite

測試環境：Windows 11

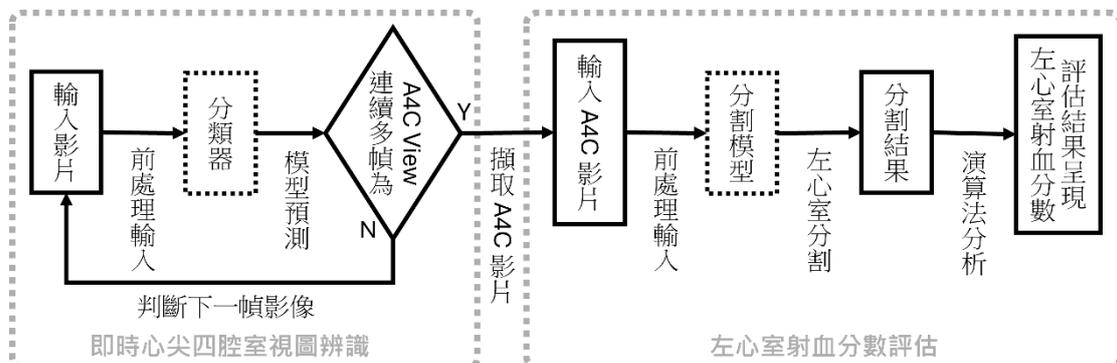
一、簡介

近二十年來，心臟疾病持續位居國人死因中的高位，而心臟衰竭便是其中一項重要課題。射血分數 (Ejection fraction, EF) 是指每次心跳時，心室泵血的程度。當其低於正常水平時，便可能是心臟衰竭的跡象。在臨床上，EF 是評估心臟收縮功能的指標，可以輔助判定病患是否具心臟衰竭的傾向，也能幫助團隊追蹤相關治療效果。因此，本專題希望能夠針對心臟超音波影像進行自動分析，對左心室射血分數進行快速評估，以提供醫療人員關於病患心臟功能的即時輔助資訊。

在本專題中，我們聚焦在以下兩個功能的演算法開發：**即時心尖四腔室視圖 (Apical 4-chamber views, A4C) 辨識**、**左心室射血分數 (Left Ventricle Ejection Fraction, LVEF) 評估**。為使本專案成果更貼近於臨床應用，我們與無線手持超音波廠商奔騰生醫進行合作，以最終能將本專案成果應用於行動裝置上，作為未來發展方向，以達成使用無線手持超音波進行心臟射血分數評估之目標，增進臨床及急救場域之心臟功能評估之效率。

二、系統架構

本專題資料處理流程如下圖一，包括 (1) 經前處理將影像大小調整及影像亮度正規化後，利用深度學習技術訓練模型辨識 A4C 視圖、(2) 判斷為 A4C 視圖後開始擷取 A4C 影片，作為後續計算左心室射血分數的輸入資料、(3) 經前處理

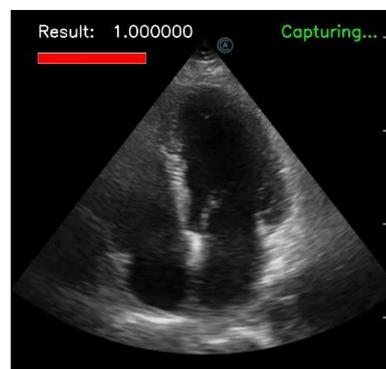


圖一、系統架構圖（黑色虛線方框為深度學習模型）

後，利用深度學習技術訓練模型，分割出左心室動態的二值影像、(4) 根據 Monoplane Simpson's Method 開發自動演算法，計算出左心室射血分數並呈現其他相關分析結果。

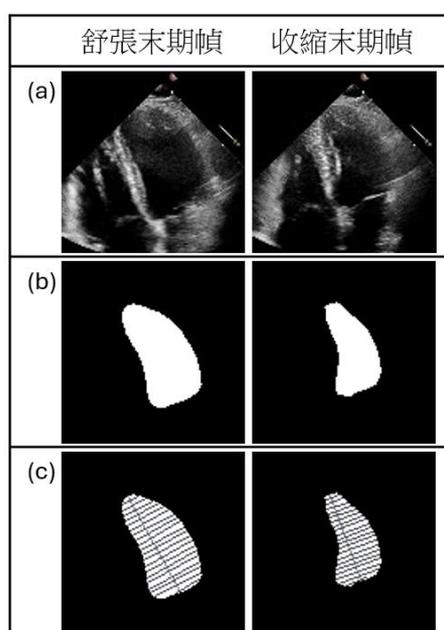
三、實驗結果

如右圖二所示，本系統以手持超音波儀器預先掃描的影片作為輸入，模擬臨床實際應用情境。經前處理後，取單一幀影像依序輸入分類器模型，將其判斷是否為 A4C 視圖的結果即時呈現於超音波畫面左上角，以一介於 0~1 之分數及 score bar 作為輔助參考；分數愈接近 1 代表模型預測其為 A4C 視圖的機率愈高。確認目前掃描平面為 A4C 視圖後，畫面右上角出現” Capturing…” 提示，表示正在擷取影片，擷取後的影片接著作為計算左心室射血分數的輸入資料。



圖二、A4C 視圖辨識結果呈現

右圖三中，由上而下分別代表在左心室射血分數 (LVEF) 評估時各個階段的結果，其中(a)圖代表原始影像。首先，將影片的每幀輸入左心室分割模型後，輸出分割後的左心室範圍，以 112*112 二值化的二維陣列儲存。接著，透過演算法計算每個幀分割的心室面積大小，以左心室面積辨識出舒張末期的幀及收縮末期的幀 (即(b)圖)。最後，分別由此二張遮罩找出左心室主軸 (心尖至基部連線)，透過演算法自動切割出垂直於左心室長軸之 20 塊三維圓盤 (即(c)圖)，藉此利用 Monoplane Simpson's Method 推算出 EDV (end-diastolic volume) 和 ESV (end-systolic volume)，再透過公式計算出射血分數，即一個心跳週期中，左心室泵血量佔心室總體積之比例。利用此演算法計算所得的各項數據，與臨床取得的實際結果，進行後續相關的準確度驗證。



圖三、左心室分割結果及射血分數計算

四、參考文獻

1. Ouyang, David, et al. "Video-based AI for beat-to-beat assessment of cardiac function." *Nature* 580.7802 (2020): 252-256.
2. Østvik, Andreas, et al. "Real-time standard view classification in transthoracic echocardiography using convolutional neural networks." *Ultrasound in medicine & biology* 45.2 (2019): 374-384.

行動穿戴式裝置腦波量測與分析系統

Mobile and Wearable Brain Signal

Recording and Analysis System

指導教授：梁勝富

專題成員：陳聖泓、曾筠傑

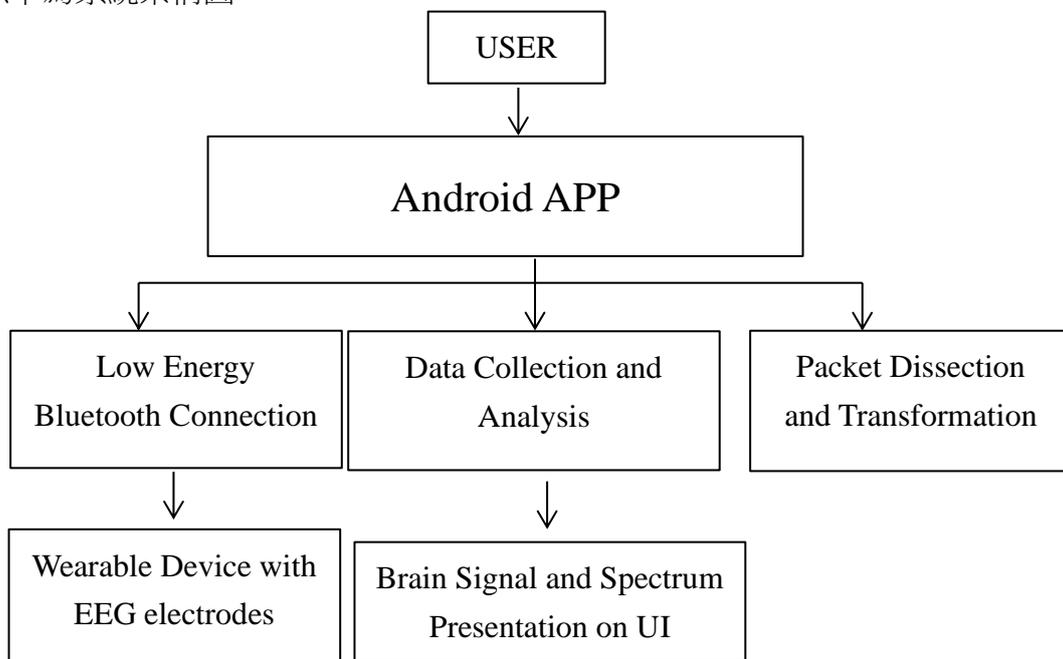
開發工具：Android Studio、Kotlin

測試環境：Android 13+

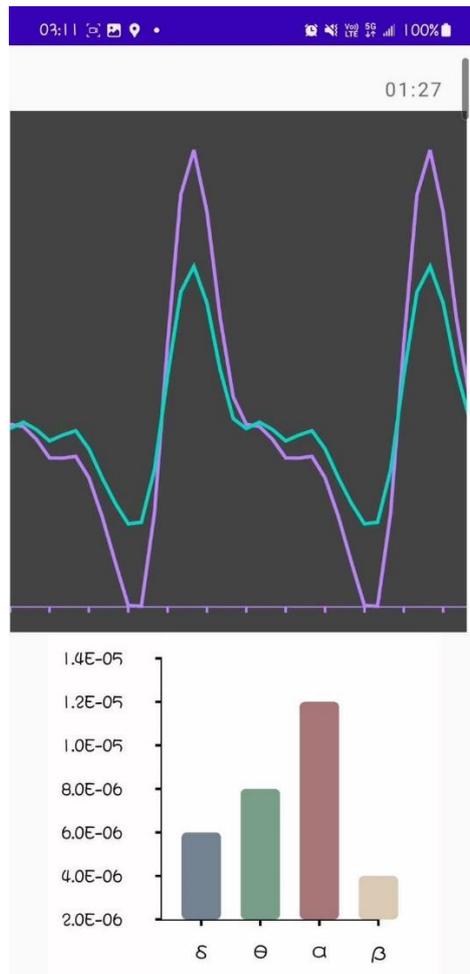
一、簡介：

基於移動式裝置便利、迅速的特性，在使用者入睡前，穿戴頭戴型生理紀錄儀器，紀錄前額葉腦電圖 (forehead-EEG)，其包含腦波(EEG)以及眼動波特徵 (EOG)，使用藍芽在數秒內完成連線，在手機上達到即時量測腦波、拆解封包轉換數據的效果以及分析各種頻譜的大小，並且以圖像呈現在畫面上，可供使用者清楚了解訊號狀況。

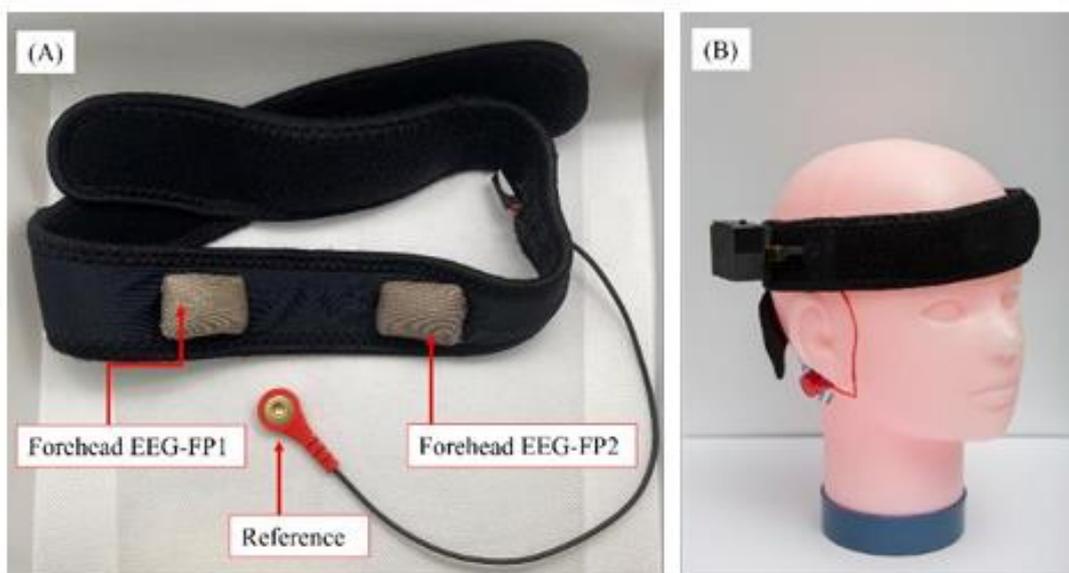
以下為系統架構圖：



二、測試結果：



圖三：手機腦波介面



圖三：穿戴式裝置

評估息肉分割模型

Evaluation of Polyp Segmentation Models

指導教授：賀保羅

專題成員：郭昱辰

開發工具：Tensorflow

測試環境：Windows10、WSL

一、簡介：

在醫學影像分析中，特別是結腸鏡息肉分割任務中，自動化方法能夠顯著提升息肉檢測的效率和準確性。結腸直腸癌 (CRC) 是全球主要的癌症死亡原因之一，因此早期檢測和移除息肉對於預防和治療 CRC 至關重要。然而，由於息肉的形狀、大小和質地多樣，準確識別和分割息肉是相當具有挑戰性的任務。

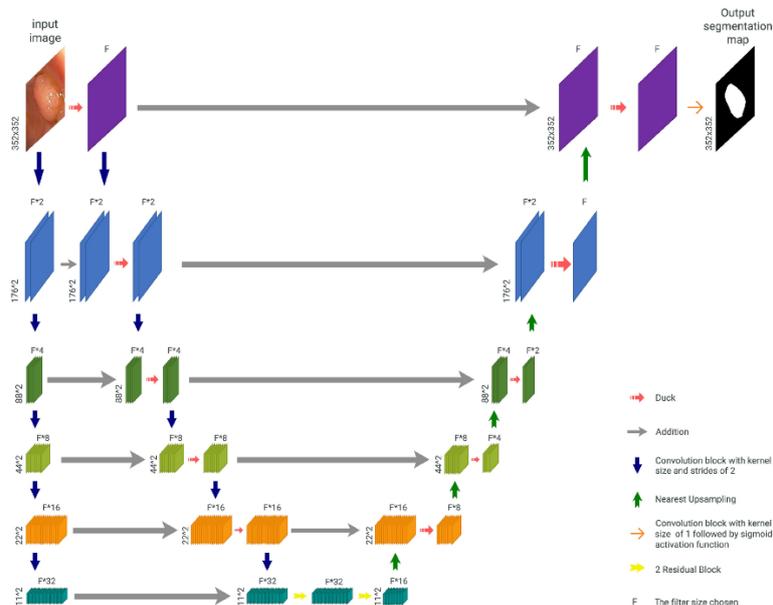
本專題旨在評估現有息肉分割模型在醫學影像資料集上的表現。藉由重現論文中提出的模型結果，我進行了兩個主要測試：

1. 重現論文結果：使用同一個資料集的訓練集進行模型訓練，並測試同一資料集的測試集，分析模型的各項評估指標。

2. 測試泛化能力：使用一個資料集訓練的模型測試另一個資料集，評估模型的泛化能力，並根據生成的評估指標進行分析。

我使用四個常見的息肉分割基準資料集進行評估：Kvasir-SEG、CVC-ClinicDB、CVC-ColonDB 和 ETIS-LARIBPOLYPDB。透過一系列實驗，我們展示了息肉分割模型在這些資料集上的優異表現，包括 Dice Coefficient、Jaccard Index、Precision、Recall 和 Accuracy 等多項指標。

模型架構圖：



二、測試結果：

1. 重現論文結果

根據論文的說明，我各自使用四個息肉分割資料集 (Kvasir-SEG、CVC-ClinicDB、CVC-ColonDB 和 ETIS-LARIBPOLYPDB) 訓練集訓練出的模型來測試同一資料集中的測試集。下面展示使用 Kvasir-SEG 訓練集訓練出的模型測試同資料集中的測試集結果。和其他三個資料集的測試結果一樣，可以展現出此模型優異的能力。

Results for Kvasir-SEG using model saved_model/kvasir-17:

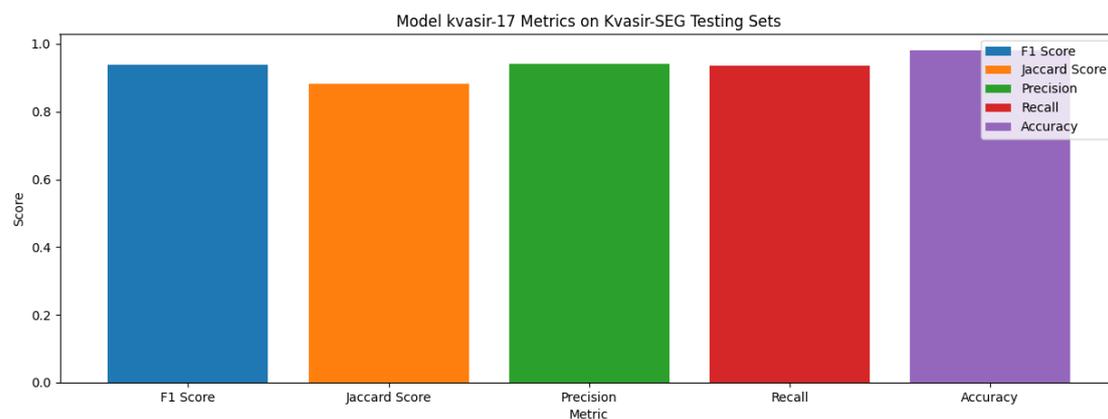
F1 Score: 0.9380582501367194

Jaccard Score: 0.8833424717104206

Precision: 0.9398199834647902

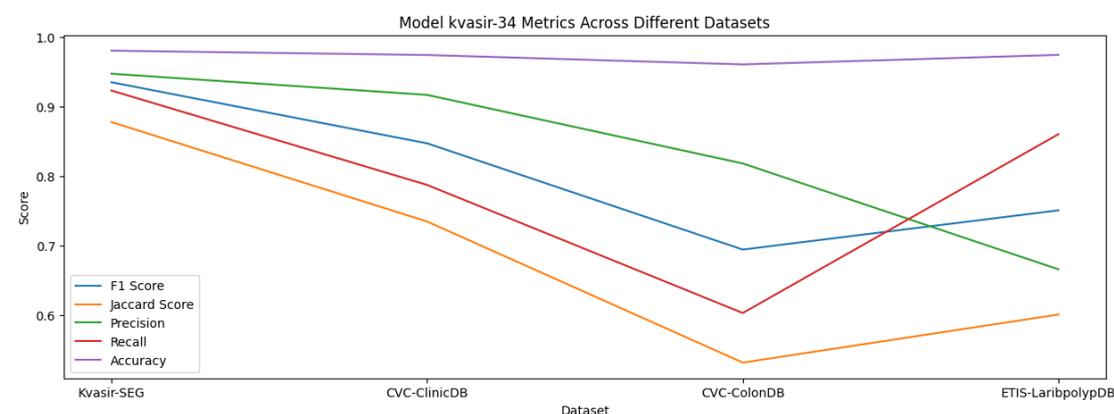
Recall: 0.9363031093417531

Accuracy: 0.9800811111828512



2. 測試泛化能力

我使用一個資料集訓練出來的模型測試其他資料集中的所有圖像，評估模型是否具有泛化能力。下面展示使用 Kvasir-SEG 訓練集訓練出的模型測試其他資料集的結果。這部分模型的能力相較於「1. 重現論文結果」較不強大，但是根據論文的描述，仍然是接近 SOTA 水準。



註: 此頁測試結果由於篇幅原因，只展示其中一部分測試結果。

腎臟超音波格式化專題

Renal Ultrasound Formatting Project

指導教授：蔣榮先

專題成員：蔡翔逸

開發工具：Chatgpt

測試環境：Chatgpt

一、簡介：

題目摘要：

以腎臟超音波為實務例子，將影像文字報告以 ChatGPT 的語意辨識與統整能力彙整成格式化資料，以用作後續可做為統計用的資料。

背景：

成大醫院已經成立三十多年，為臺南地區重要的醫學中心。其中累積的病患資料無論是數量還是質量都有非常好的規模，但因為原先儲存的資料型態都是文本資料，所以在進行統計研究時無法直接取用。如果能利用大型語言模型對於文句理解的能力將腎臟超音波報告轉換成可經一般程式讀取的檔案型態（Json 檔案），提高醫療資料的可讀性和可處理性，進而支援自動化的醫療資料分析和整合，為未來的醫療資訊系統提供強大的後盾。

介紹：

本次專題分為產生資料以及檢查錯誤兩個部分：

在生成資料的部分，首先建立一個合適的轉換後資料模板，包含以下項目：Medical history No.、Personal Information、Date of Examination、Kidney、Liver、Bladder、Ureters、Prostate、Other 以及 Impression。接著，通過完善且精確的提示詞（Prompt）來正確生成資料。發現初始提示詞的正確率不足 28%，因此將錯誤分為六大類，並通過 self-consistency 和 Direct prompting 進行錯誤改善，總共進行了五個版本的測試（每個版本測試 50 筆）。

在達到一定的資料轉換正確率之後，進行下一個步驟：檢查錯誤。這個步驟的主要目的是抓出剩餘的錯誤，以達到最大的資料正確率。檢查過程是透過分次分段的方式進行，並將欲檢查的錯誤同樣使用 self-consistency 和 Direct prompting 的方式加入提示詞（Prompt）中。總共檢查了 50 筆錯誤資料。

二、 測試結果：

在測試五版生成 prompt 後得到各自的正確率，version 1 正確率為 28%，version 2 正確率為 34%，version 3 正確率為 52%，version 4 正確率為 64%，version 5 正確率為 82% (最終版)，共提升了 54 個百分點。

版本	正確率
v1	28%
v2	34%
v3	52%
v4	64%
v5	82%

測試 50 筆錯誤資料的檢查之後，共可以檢查出 42 筆錯誤資料，可得檢查成功率達到 84%。

在得到 82% 的生成正確率以及 84% 的檢查正確率之後，把兩者結合。用 version 5 生成 prompt 生成資料之後，再透過檢查 prompt 將錯誤的資料檢查出來，我們將可以得到正確率超過 95% 的轉換後資料。

吞嚥困難的早期檢測

指導教授：藍崑展

專題成員：李務

開發工具：Python、flutter、firebase

測試環境：Android

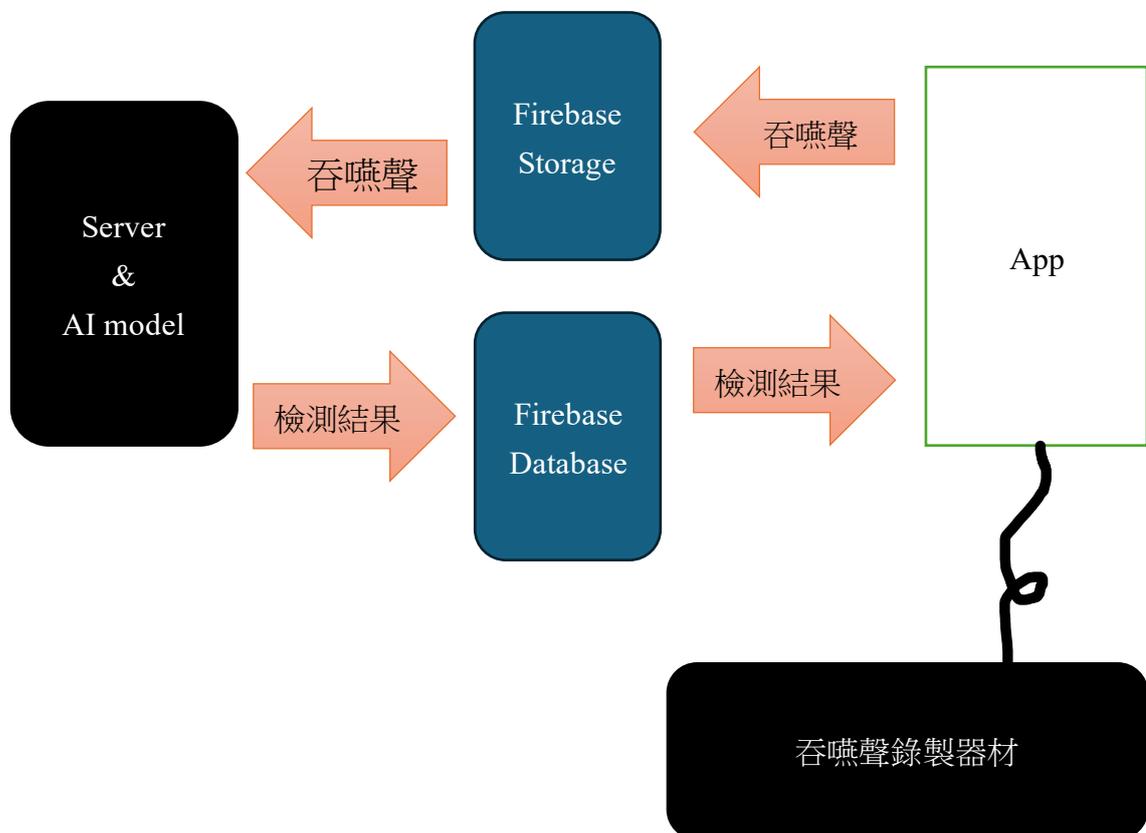
一、簡介：

由於存在吞嚥困難可能的患者等發現的時候都來不及了，因此我們與成大醫院合作拿到一系列已標記吞嚥困難與否的吞嚥聲資料，並以此訓練出準確度高於85%的模型。

再以 Flutter 設計出 App，讓使用者可以在手機插上量測吞嚥聲的工具後，直接將錄下的吞嚥聲上傳至 Firebase Storage，下載至電腦後跑模型辨識是否有吞嚥困難後，以 Firebase Realtime Database 回傳結果至手機。

另外 App 上有附上衛教影片了連結、使用錄製吞嚥聲器材的方式，並提供了使用者登入的方式，能線上記錄醫院所需要的各樣資料（姓名、生理性別、年齡等等）以便後續追蹤。

以下為架構圖：



半監督醫學影像分割-雙向複製黏貼技術介紹

Introducing to Semi-Supervised Medical Image Segmentation: Bidirectional Copy-Paste

指導教授：藍崑展

專題成員：龔暄仁

開發工具：Python, Google Colab

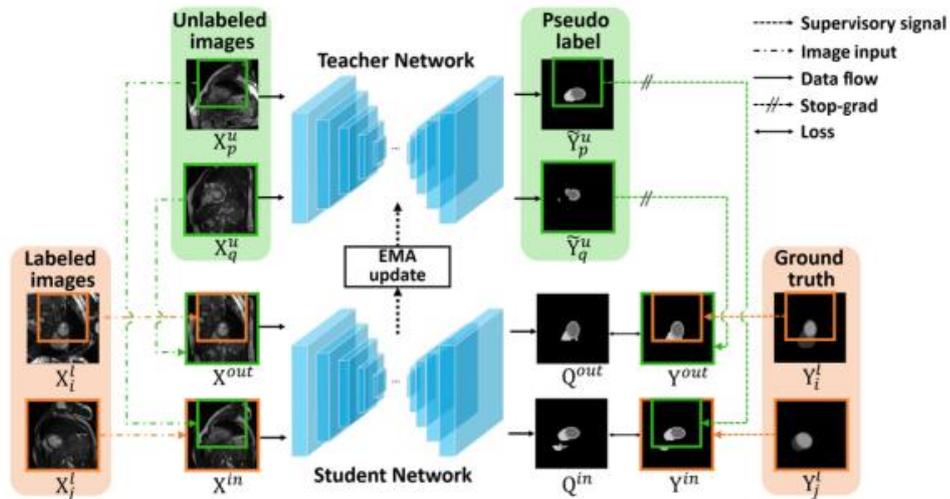
測試環境：Linux 22.04

一、簡介：

在進行 AI 模型訓練時我們往往有著大量的資料，但其中絕大部分的資料都是 unlabeled data，只有少數資料是具有 labeled 的。在 semi-supervised 中我們使用這些 unlabeled data 來進行訓練，往往可能會遇到 unlabeled data 和 labeled data 在資料分佈方面的不匹配問題。為了解決這種問題，我將介紹一種技術，稱之為雙向複製黏貼。這項技術簡單來說，就是將 labeled data 的圖片切下部分黏貼到 unlabeled data 上，也將 unlabeled data 的圖片切下部分黏貼到 labeled data 上。藉由這個簡單的操作，使的我們可以在兩種資料間建立關係。

此方法使用 Mean Teacher 架構。Mean Teacher 架構含有兩個相同的網路，分別稱之為 Student Network 和 Teacher Network。訓練時，我們先將 Batch 送入 Student Network，並得出一個預測結果。我們隨後用 Batch 中的 labeled data 來和真值進行 loss 計算，最後在藉由梯度反轉來更新 Student Network 的數值。至於 Batch 中的 unlabeled data 在灌入 Student Network 後，我們會得到一個預測結果 A。在 Teacher Network 這邊我們將 unlabeled data 先加入一些噪音後灌入訓練，可以得到一個結果 B。訓練的目的我們就是希望結果 A 能和結果 B 一樣，因為這代表這 unlabeled data 就算經過加入噪音改變，這個模型依然會將其判斷為同一個東西。

這個方法最大的改變就在於 Student Network 的資料輸入那邊。在過去舊有的方法中，我們是直接將 labeled 和 unlabeled data 分別灌入網路中，因此兩邊的資料是沒有交集的，而這導致最終訓練完的結果中 labeled 和 unlabeled data 的資料分佈不一致。此方法再將資料灌入 Student Network 前會先將資料進行前面麼雙向複製黏貼，而這將使的訓練完的模型結果在資料上會有一段共同語意，使的兩者的資料分佈接近。



二、測試節果：

根據這個方法論的論文，他有使用幾個數據集，分別是 3D 資料的 LA dataset 和 2D 資料的 ACDC dataset，而我準備使用其中的 ACDC dataset 來進行復現。我上網將 ACDC dataset 下載下來，先將集從原本醫療影像的 3D 圖轉成 2D 切片，並使用了其論文提供的一些程式碼將其作了前處理後，送入了這篇論文提供的架構中。

Method	Scans used		Metrics			
	Labeled	Unlabeled	Dice \uparrow	Jaccard \uparrow	95HD \downarrow	ASD \downarrow
U-Net	3(5%)	0	47.83	37.01	31.16	12.62
U-Net	7(10%)	0	79.41	68.11	9.35	2.70
U-Net	70(All)	0	91.44	84.59	4.30	0.99
UA-MT			46.04	35.97	20.08	7.75
SASSNet			57.77	46.14	20.05	6.06
DTC			56.90	45.67	23.36	7.39
URPC	3(5%)	67(95%)	55.87	44.64	13.60	3.74
MC-Net			62.85	52.29	7.62	2.33
SS-Net			65.83	55.38	6.67	2.28
Ours			87.59 \uparrow 21.76	78.67 \uparrow 23.29	1.90 \downarrow 4.77	0.67 \downarrow 1.61
UA-MT			81.65	70.64	6.88	2.02
SASSNet			84.50	74.34	5.42	1.86
DTC			84.29	73.92	12.81	4.01
URPC	7(10%)	63(90%)	83.10	72.41	4.84	1.53
MC-Net			86.44	77.04	5.50	1.84
SS-Net			86.78	77.67	6.07	1.40
Ours			88.84 \uparrow 2.06	80.62 \uparrow 2.95	3.98 \downarrow 2.09	1.17 \downarrow 0.23

Table 3. Comparisons with state-of-the-art semi-supervised segmentation methods on the ACDC dataset.

其中的 ours 為 BCP 架構的表現，可以看到他在 labeled 資料僅有 5% 時，模型的表現有顯著的提升，而當 labeled 資料來到 10% 時他的表先依然略優於其他架構。

```
init weight from ./model/BCP/ACDC_BCP_7_labeled/self_train/unet_best_model.pth
100% 40/40 [00:49<00:00, 1.23s/it]
[array([[0.88928676, 0.80708719, 1.45114762, 0.47634128]]), array([[0.86833849, 0.7697884, 4.98195099, 1.00474862]]), array([[0.92704921, 0.86905689, 6.85768234, 1.74954088]])]
[0.89489149 0.81531082 4.43026032 1.07687693]
```

而上圖為我使用同樣數據集進行復現的成果，其中的表現為圖片中最下方的陣列，由左至右分別為[dice Jaccard 95HD ASD]。我使用的為 10% labeled data，可以看到其表現的和論文相近，表現依然出色。

多站點多產品產線排程-以被動元件製造廠為例

Scheduling for multi-site and multi-product production - the case of passive component factory

指導教授：王宏錯

專題成員：吳昱頡、劉士郡、鄭鈺鎧

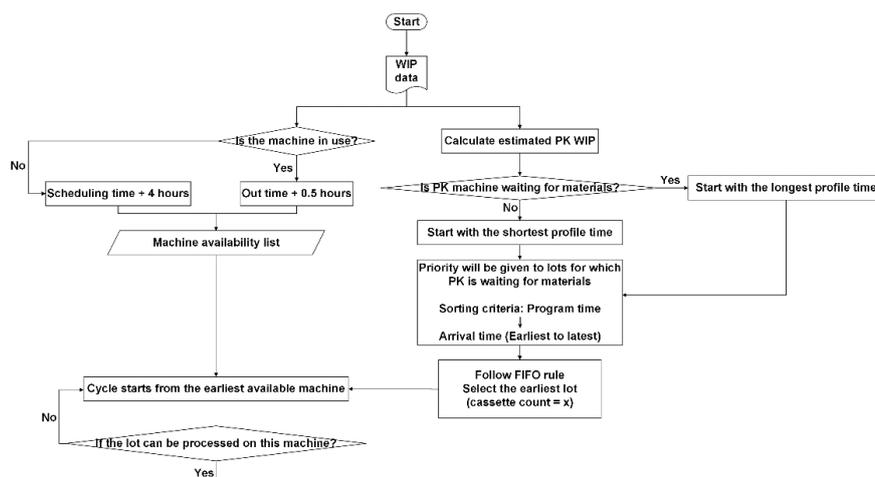
開發工具：C++14、Visual Studio Code、Clion

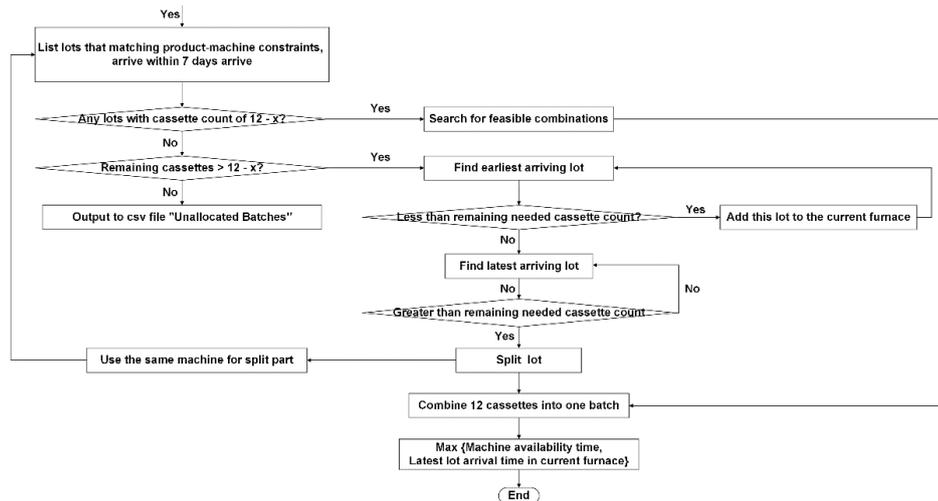
測試環境：Windows 10

一、簡介：

本專案主要目標為優化積層陶瓷電容器（MLCC）產線中的 Binder Burn Out（BBO）站點及其後續站點（PK）排程。首先讀取產品資料 csv 檔，計算所有機台可使用的程式時間和計算批貨的卡匣數，以機台程式選批貨的方式，將可以剛好湊成 12 個卡匣數的批貨送進機台，無法湊成剛好 12 卡匣的則進行拆批，再與其他批貨繼續湊。

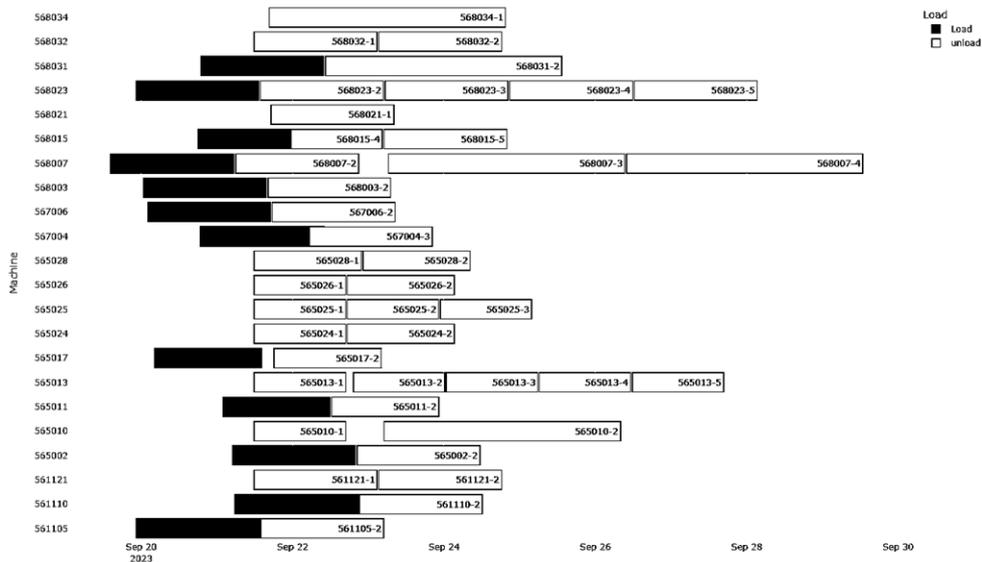
系統架構圖：





二、測試結果：

附圖為機台排程之甘特圖。透過排程優化縮短機台閒置時間，提高整體生產效率。



應用強化學習和基因演算法於多機多產品排程 問題-以 SMT 為例

Applying Reinforcement Learning and Genetic Algorithm to Multi-Machine and Multi-Product Scheduling Problem - A case study on SMT

指導教授：王宏鎧

專題成員：陳育琮

開發工具：C++、Python

測試環境：Windows11

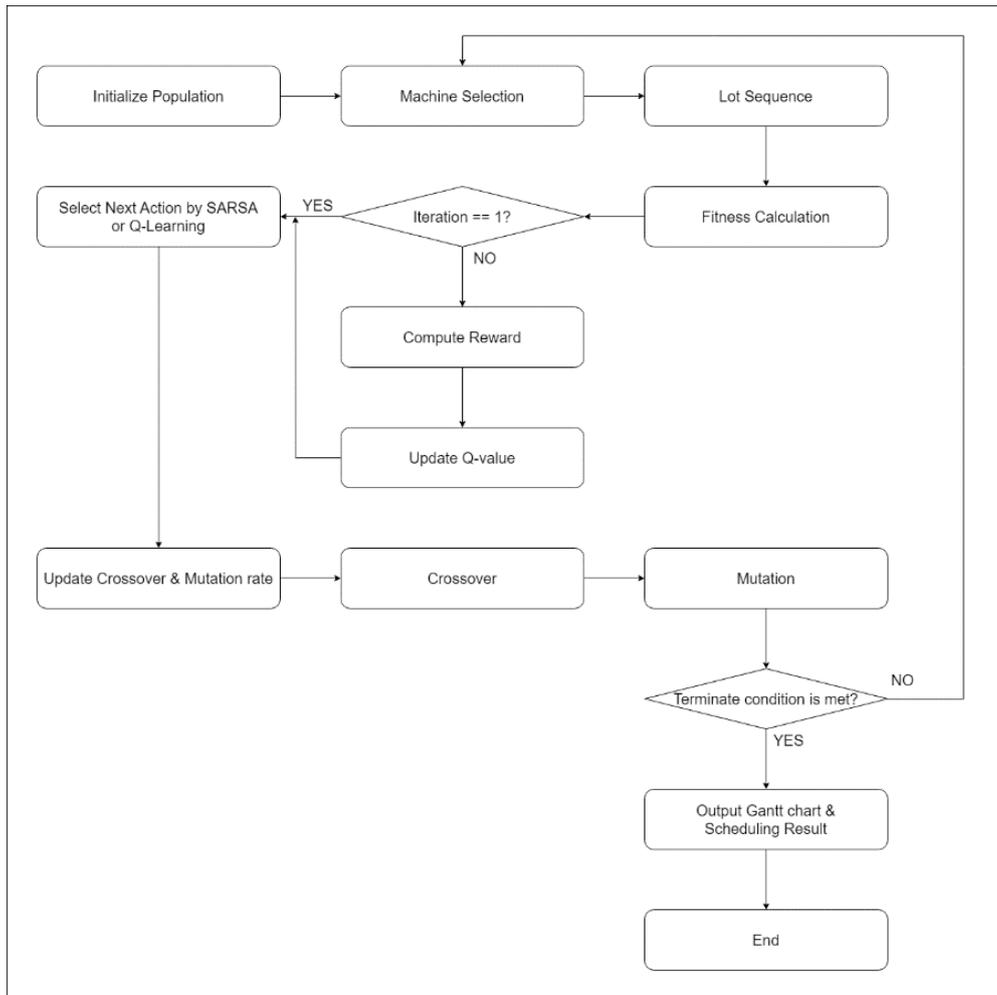
一、簡介：

本專題針對 SMT 站點的排程進行優化，以最小化完工時間為主要目標，並以降低改機次數和滿足交期為副目標。透過基本的派工法則、基因演算法、深度強化學習，以及考慮工廠內的各項限制，生成橫軸為時間、縱軸為機台名稱的排程甘特圖。最終產出每台機台對應的 Lot、改機次數和超出 Commit time 數量的 Excel 檔案。

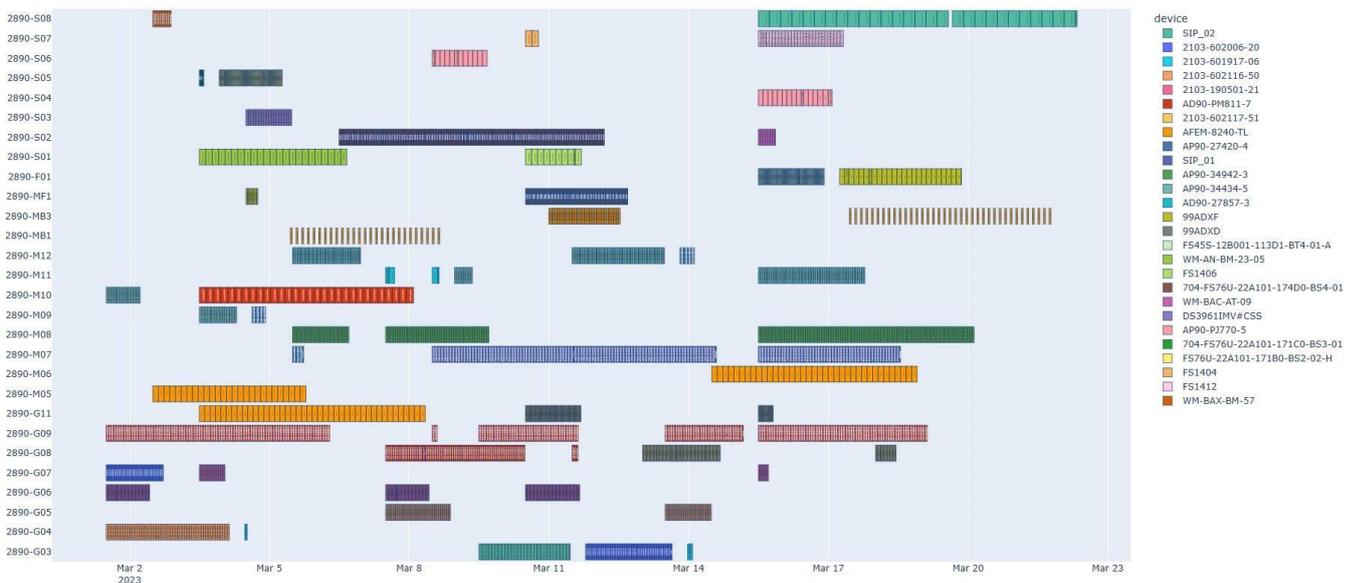
基因演算法中，每條染色體代表一次完整排程的結果。初始化時會先產生一定數量的染色體，並且在接下來的迭代中，由母代進行交配及突變產生新的子代，再透過輪盤式選擇(Roulette Wheel Selection)以及菁英挑選法(Elitism Selection)依照適應值選出適合的染色體，直到達到終止條件時結束程式。

由於基因演算法的迭代速度較慢，本專題嘗試利用強化學習來動態改變交配和突變率，以實現更快的收斂速度或更優的排程結果。在本專題的前半部分，我們使用 SARSA 算法來更新 Q-Table，因為初始化時 Q-Table 沒有值，SARSA 能在探索過程中逐步積累信息。接著，我們使用 Q-Learning 在已更新的 Q-Table 中找出適合的動作（即改變交配和突變率）。通過這種方法，我們希望能更有效地調整基因演算法的參數，從而提高其性能和優化效果。

下圖為本專題之系統架構流程圖：



二、測試結果：



強化學習於多站點排程系統之應用

-以 TFTLCD 為例

Applying Q-learning to Multi-Site Scheduling System - A Case Study on TFTLCD

指導教授：王宏錯
專題成員：何宇婕
開發工具：Python
測試環境：windows11

一、簡介：

由於此為一個多站點多機台排程最佳化問題，若以傳統排程方法無法有效率且系統性找出有效降低總完工時間的最佳解，故此專題設計採用強化學習中的 Double Q-learning 方法以逐步收斂出最佳解。

每一個工作單位(job)須經五個站點(stage)，每個站點限制條件皆不完全相同，因此每一站點各自使用自己的 Q table。Q table 中的 state 設定為按照到達時間(arrival time)排序的所有工作單位，action 設定為可選擇之機台，而 reward 則設定為每個工作單位於當前站點完成時間之倒數。

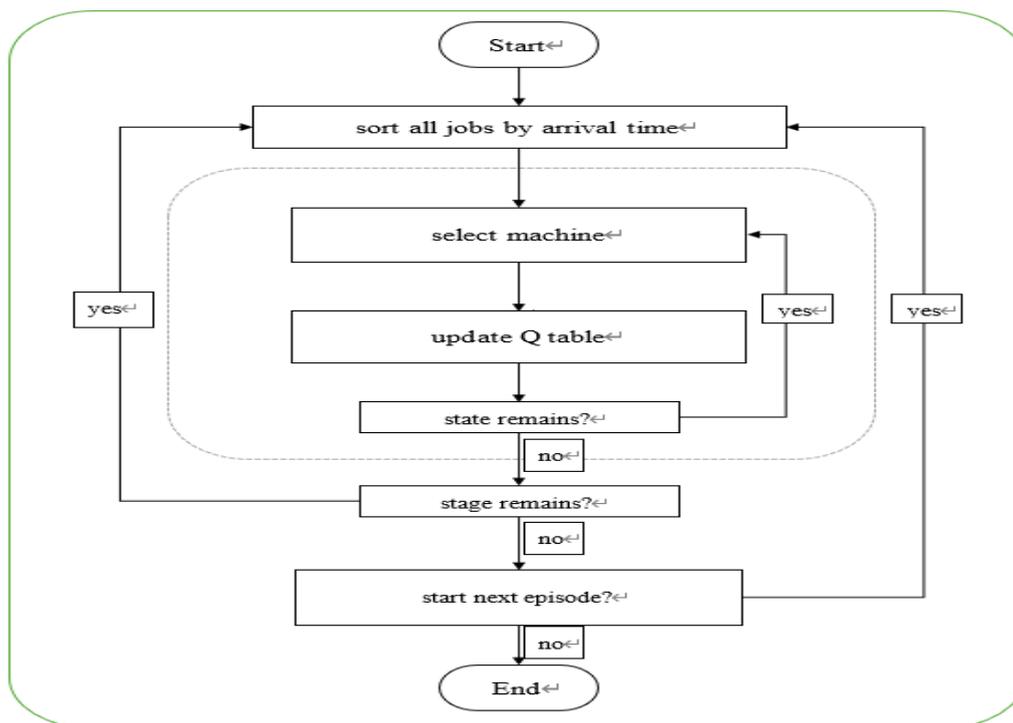
程式執行時每次依到達當前站點的 arrival time 排序所有工作單位，接著依序根據 Q table 中 Q value 最大者選擇機台。選擇機台時加入 epsilon decay 機制以增加初期探索可能。另外，由於使用單一 Q table 做訓練易傾向過度高估 Q value，從而導致可能造成總是選擇到 Q value 被高估的 action 來更新 Q table。為避免此問題故設計採用兩個 Q table 相互訓練，即每次選擇 action 之 Q table 跟計算更新 Q value 之 Q table 並非同一個，如此便能達到使兩個 Q table 相互制衡。

此方法中可將每個站點視為一個環境，而上述更新每站點 Q table 即為與環境互動的過程。每次與環境互動後我們獲得一新的狀態，也就是當前每個機台的使用情況以及每個工作單位到下一站點之 arrival time，並繼而使用此狀態與新的環境(stage)互動。當與五個站點皆互動完成即稱之為一 episode，其後使用此方法迭代多次以收斂出最佳解。

以下為用於更新 Q Table 之公式：

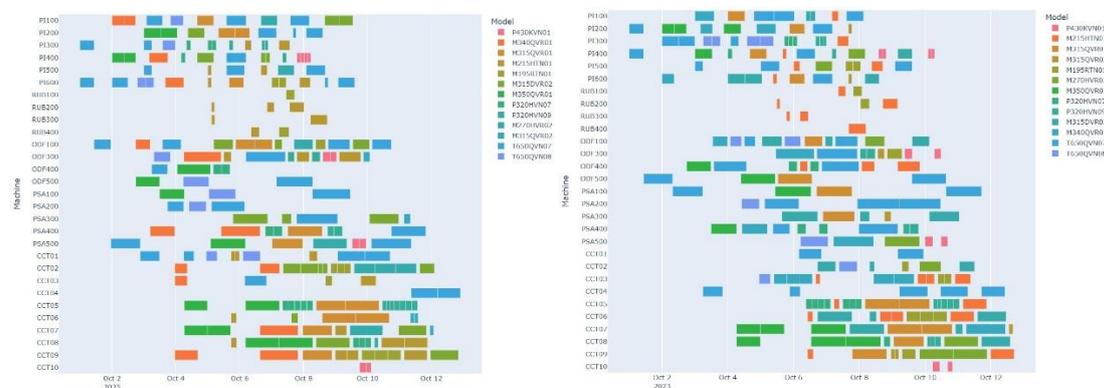
$$\text{Define } a^* = \arg \max_a Q^A(s', a)$$
$$Q^A(s, a) \leftarrow Q^A(s, a) + \alpha(s, a) (r + \gamma Q^B(s', a^*) - Q^A(s, a))$$

以下為流程圖：



圖一：flow chart

二、測試結果：



圖二：排程策略甘特圖輸出結果

圖二以甘特圖呈現兩組分別各執行6000 episode 後獲得之最佳排程策略。最佳完工時間(make span)分別由 414小時及 350小時收斂至 285小時及280小時。系統亦同時記錄每個站點之最佳 setup time。

MakeSpan: 280.9570175438596

PI setupTime: 39

RUB setupTime: 4

ODF setupTime: 23

PSA setupTime: 16

CCT setupTime: 36

Episode 0, best makespan = 350.239, current makespan = 350.239, current epsilon = 0.9

...

Episode 5999, best makespan = 280.957, current makespan = 335.685, current epsilon = 0.366

針對彩色多特徵光滑物件之 智慧自動化瑕疵檢測

指導教授：陳響亮

專題成員：葉弘堯、王翔聖

開發工具：Python、C#、TensorFlow

一、摘要

在工業 4.0 時代，製造業正面臨著前所未有的轉型挑戰。隨著消費市場日趨多元，產品生命周期縮短，企業須不斷提高生產彈性與效率，才能維持競爭優勢。在生產如塑膠軟管等彩色多特徵光滑物件時，如何在龐大產量中快速而準確地識別表面瑕疵，是攸關產品良率與交期的關鍵。

為此，本研究開發了一套結合機器視覺和深度學習的智慧瑕疵檢測系統，能快速準確地識別光滑表面上的多種缺陷。使用高解析攝影設備擷取產品表面影像，透過 OpenCV 進行影像前處理，再以人工標註方式建立封口斜口、封口毛邊和表面凹陷三類瑕疵的訓練資料集。在此基礎上，我們採用 YOLOv4 物件檢測演算法，通過參數調優與反覆訓練，得到一個高效穩健的瑕疵檢測模型。

實驗結果表明，本模型在三類瑕疵的平均檢測精確度達到 96.41%，系統可自動標示出瑕疵的種類與位置，整合於產線可即時篩選出瑕疵品，大幅降低目檢人力與成本，進而強化品質一致性，減少客訴，提升顧客滿意度，從而促進品牌忠誠度和市場佔有率，為製造商創造更多獲利空間。

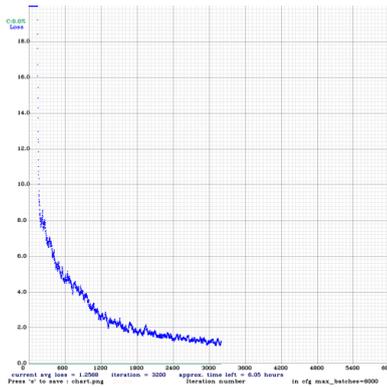
二、研究方法(研究流程)

1. 影像擷取:拍攝正常與瑕疵樣品
2. 影像處理:使用 OpenCV 進行處理，並抓出待辨識物體
3. 標籤檔建立:使用 labelimg 軟體製作訓練標籤
4. 模型參數調整:根據使用需求對模型的參數做適當的調整

5. 模型訓練與優化:colab 上使用自製訓練集和標籤檔訓練模型

三、 研究結果

在針對三種類型的瑕疵:表面凹陷、斜口、毛邊，進行多次訓練和不斷優化後，訓練到模型 loss 在 1.25 左右，且精確度都超過九成，平均精確度高達 96.41。



```

detections.count = 490, unique truth count = 317
class_id = 0, name = hollow, ap = 91.05% (TP = 171, FP = 17)
class_id = 1, name = bevel, ap = 98.22% (TP = 58, FP = 1)
class_id = 2, name = rough_edges, ap = 99.96% (TP = 70, FP = 1)

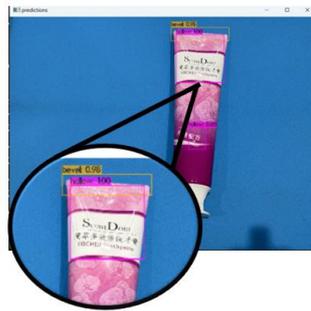
for conf_thresh = 0.25, precision = 0.94, recall = 0.94, F1-score = 0.94
for conf_thresh = 0.25, TP = 299, FP = 19, FN = 18, average IoU = 72.77 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.964885, or 96.41 %
    
```

瑕疵類別	訓練標籤數	精準度(AP)
表面凹陷(hollow)	約700	91.05%
斜口(bevel)	約400	98.22%
毛邊(rough_edges)	約400	99.96%
平均精確度(mAP@0.50)		96.41%

▲訓練到模型 loss 為 1.25 左右

▲三種瑕疵的訓練結果

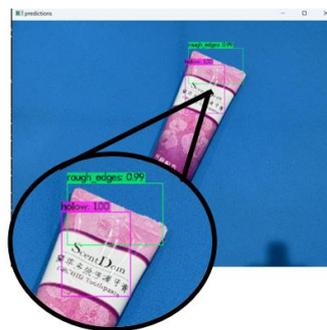


▲正面

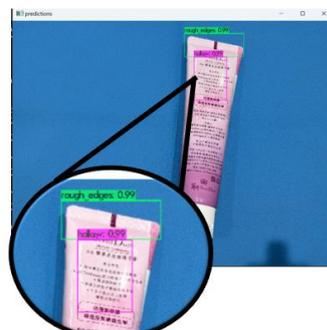


▲背面

▲模型測試畫面: 斜口和凹陷瑕疵



▲正面



▲背面

▲模型測試畫面: 斜口和凹陷瑕疵

智慧工廠戰情室與線上異常預警系統實現

指導教授：陳響亮

專題成員：張暉俊

開發工具：C#、visual studio code 2019

測試環境：Windows 11

一、簡介：

近年因全球暖化、氣候變遷、極端氣候造成了嚴重的糧食緊缺、氣候災難等問題，世界各國紛紛出台節能減碳相關政策，如歐盟的碳邊境調整機制（CBAM）和美國的潔淨競爭法（CCA）等。各政策苗頭也都指向碳排最高的製造業，在此背景下，製造業將面臨如何在能維持產能效率的同時又能實現節能減碳的雙重挑戰。

為因應此一挑戰，本研究將致力於建立一個智慧工廠戰情室和線上異常預警系統，透過工業物聯網(IIoT)、生產管理系統、大數據分析，實現對生產過程中各項關鍵生產指標的監控與預警。對此，將建立五大功能模組「產能效率」、「機台監測」、「電力監測」、「設備管理」、「生產總表」，並透過智慧工廠戰情室收集各生產線實時數據，以可視化界面呈現關鍵生產指標及設備運行狀態，並能夠快速處理和分析大量數據，生成即時報告和預警訊息。機台監測系統則透過大數據之方式分析機台參數，並設定感測器閾值，對生產過程中可能出現的異常情況進行提前預警，當系統檢測到異常時，能夠及時通知相關人員，使其及早介入，防止機台停機或生產不良品，降低生產損失。此外，設備管理模組將通過監測設備運行時數，分析設備的健康狀況，並提出維護建議，從而提升設備的可靠性和壽命。同時，電力監測將通過安裝在各機台的電力監測裝置，實時記錄機台的用電量數據，此數據不僅有助於優化能源用，亦能為企業碳排放盤查提供準確的基礎數據，協助企業達成節能減碳之目標。系統架構圖如圖1所示。

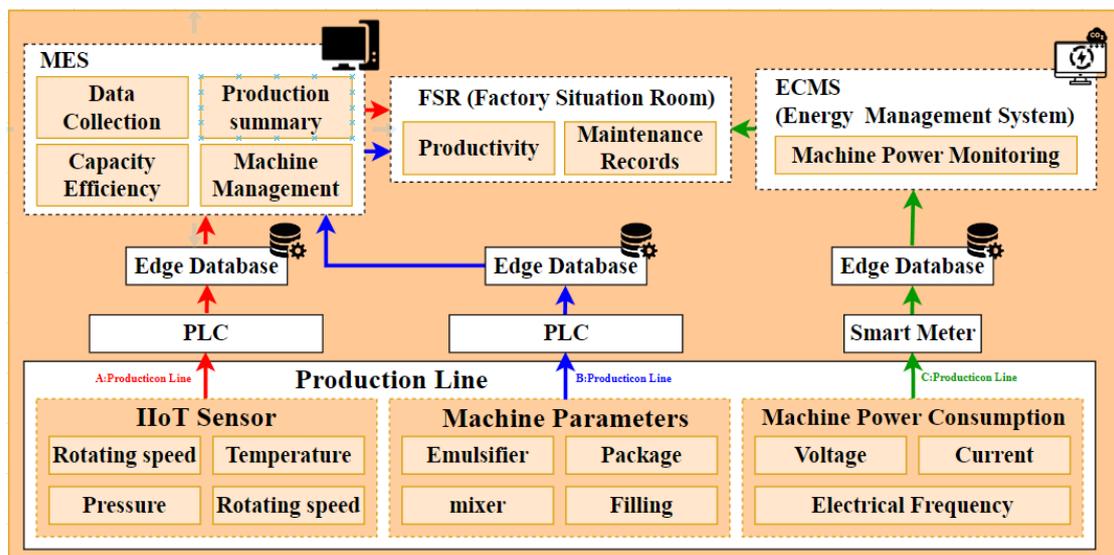


圖1 系統架構圖

二、測試結果：

本研究以某傳統化妝品工廠為模擬場域，以模擬實際產線機台及感測器實際接收之數據建立本系統，並完成以下五個模組，實際成果如圖2所示。

1. 產能效率：顯示各機台的目前狀態、開始生產時間、實際生產數量、目標生產數量、異常停機時間等。
2. 機台監測：顯示目前所有已登錄系統的機台編號與其運行狀態並顯示機台最近一次的異常資訊。
3. 電力監測：顯示目前所有已登錄系統的機台編號之當日、當月用電量與碳排量。
4. 設備管理：顯示目前所有已登錄系統的機台使用資訊與搜尋各機台之保養歷史紀錄。
5. 生產總表：顯示目前所有已登錄系統的機台編號之目標產量與實際產量。



圖2 智慧工廠戰情室與線上異常預警系統實際介面

液體烤漆瑕疵智慧檢測系統

指導教授：陳響亮

專題成員：楊凱茗

開發工具：Python、Pytorch

測試環境：Windows10

一、 簡介

本研究的動機源自於實際走訪烤漆工廠時發現，目前對成品瑕疵的檢查主要仰賴人力目視判別，但人為因素如疲勞、專注力下降等，可能導致檢查品質不穩定。烤漆品質的一致性對產品的外觀與使用壽命影響甚鉅，傳統人工抽檢方式不但耗時費力，還可能因人為疏忽而遺漏瑕疵，導致不良品流出，影響客戶滿意度。如何透過自動化智慧檢測來改善這個問題，提升品管效率並降低人力成本，成為值得探究的課題。

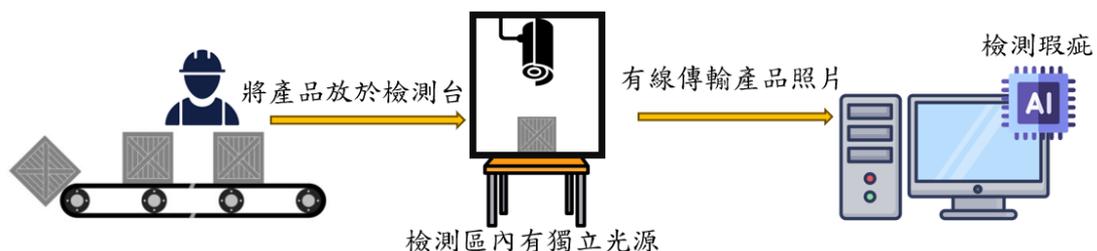
本研究使用工業級高解析攝影機拍攝產品表面影像，再以影像分割、數據標註等方式進行前處理，建立刮痕瑕疵的訓練資料集。接著，採用以 ResNet50 為骨幹網路的 Faster R-CNN 演算法，對模型進行訓練與優化，提高刮痕瑕疵的辨識精準度。

本智慧檢測系統的瑕疵辨識準確率已達79%，展現優異效能。我們開發的使用者介面與統計分析功能，便於工廠管理人員掌握品質資訊。系統具備實際生產線應用的基礎能力，可大幅提升烤漆製程品質管控效率。若導入業界後，不但能降低人力檢驗的不穩定性，節省人力成本，更能提高產品良率，縮短交期，提升客戶滿意度，為工廠創造可觀經濟效益。

在未來的研究方向上，我們將持續蒐集更多元化的工廠樣本，豐富訓練資料集，讓檢測模型能學習到更多瑕疵型態，進一步提升辨識準確度。並進行試點導入，持續優化改進，在產學界共同努力下，定能為產業創造更卓越價值。

二、 研究方法

1. 系統架構圖



2. 模型介紹

本研究透過 pytorch 套件中的 Faster RCNN 以及 ResNet50 做為

backbone 進行訓練，其選用的原因分別為，前者在物件辨識中對於硬體資源的使用上較為節省，後者的模型提供足夠深度以提升精準度，對資料集數量要求較低，此外利用 ImageNet 之 pretrained weight 加速收斂。

3. 訓練方式

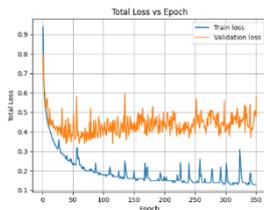
將透過自製黑箱拍攝的資料集的每一張圖片裁切成300*400像素大小的小照片並去除只有背景的圖片製作成處理過的資料集，將處理過得資料集依照 1 : 9 的方式分為驗證集和訓練集供模型學習。

4. 檢測方式

透過鏡頭採集樣品照片，分割成多張300*400像素之小照片供模型檢測及標記，並將結果合併輸出與採集時相同大小帶有標記之照片。

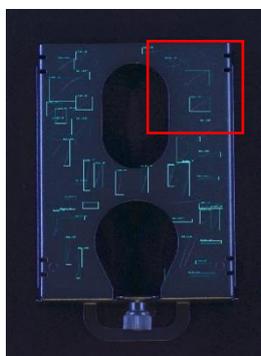
三、測試結果：

1. 模型訓練過程

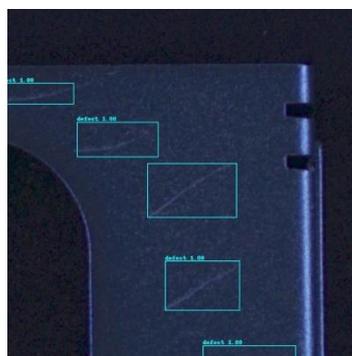


model = ResNet50
Pretrained = True
Epoch = 350
Learning rate = 0.001
Accuracy = 79.41%

2. 照片測試



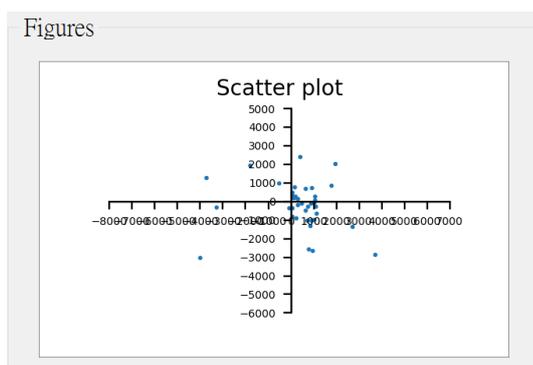
實際輸出



局部放大

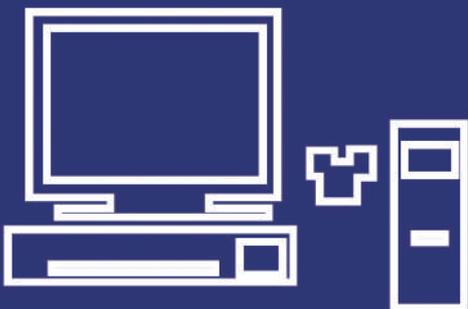
3. 瑕疵分布分析圖

將模型偵測結果之瑕疵位置至作為散佈圖，方便使用者了解瑕疵主要出現在何處，協助生產人員改善產線狀況



CSIE

National Cheng Kung University
Department of Computer Science and Information Engineering



2024