

國立東華大學教學卓越中心
113-2IDEAS 教學課程計畫成果報告書

計畫主持人:陳震宇、廖慶華
單位:電機工程學系、藝術與設計學系

目錄

壹、113-2 期末成果報告確認·····	002
貳、執行成果總報告·····	003
參、活動紀錄表-1(附件一)·····	006
肆、活動紀錄表-2(附件二)·····	008
伍、活動紀錄表-2(附件三)·····	010
陸、計畫執行成果 跨領域組別(附件四)·····	012
柒、計畫執行成果 智慧計算組別(附件五)·····	039

國立東華大學-IDEAS 教學課程計畫 113-2 執行成果報告書確認表

課程/學程名稱：理工學院 智慧科技跨領域應用與實作學程 [含智慧計算實驗、現代書藝研究、TTPBL 跨領域整合專題(二)、 (四)、(六)]		
授課教師：陳震宇、廖慶華		
服務單位：電機工程學系/副教授兼國際事務處副處長、藝術與設計學系/教授兼藝術學院院長		
班級人數:智慧計算實驗 37 人、現代書藝研究 8 人		
勾選	檢核項目	說明內容
<input checked="" type="checkbox"/>	本確認表	請確實填報，以俾利核對
<input checked="" type="checkbox"/>	執行成果總報告表-電子檔 (Word)	字型：標楷體 (中文)； Times New Roman (英文) 行距：單行間距 字體大小：12 號字
<input checked="" type="checkbox"/>	活動記錄表	當期程全部活動紀錄，如講座、參訪、期末成發展等
<input type="checkbox"/>	AI 培訓講座/工作坊	<input type="checkbox"/> A 類 素養導向、 C 類 跨領域課程 <input type="checkbox"/> B 類 AI 科技運用(2 場)
依據 IDEAS 教學課程計畫辦法第四條，受補助計畫主持人 有義務參加舉辦之 AI 培訓講座/工作坊 ， A 類 素養導向課程與 C 類 跨領域課程 1 場講座 ； B 類 AI 科技運用為 2 場講座 ，以培養教師 AI 應用的能力，因應未來發展趨勢。		

- 繳交期末成果報告時，請確認繳交項目是否齊全
- 本年度所有受補助課程/學程之成果報告，將上述資料匯集成冊(封面、目錄、內容、附件)，做為本期成果報告書
- 若有相關疑問，請與承辦人郭心怡助理聯繫
(#6591；imyeee@gms.ndhu.edu.tw)

IDEAS 教學課程計畫-執行成果總報告

跨領域課程

一、教學/計畫目標

本課程強調結合素養導向理念與前瞻科技應用，融合 AI 科技與跨域實作精神，透過課程設計讓學生能夠：

- 掌握 AI 科技（如機器學習、影像辨識、智慧決策等）於真實情境中的應用方法，強化問題解決能力。
- 建構跨領域合作平台，促進不同專業學生之間的協同學習與溝通表達能力。
- 培養創新思維與設計思考素養，強化學生對未來技術發展與應用場域的整合能力。
- 融入工程倫理與社會責任，提升學生在實作過程中的批判思考與系統分析能力。
- 培育具備面對未來產業挑戰的跨域人才，培養團隊合作、創造力與實作能力兼備之新世代工程師。

課程同時引導學生探索生活中真實問題，並將 AI 技術應用於解決具體場域的挑戰，達到知識與實踐並重之教學目標。

二、課程內容特色

Innovation 創新

結合在地文化與書法藝術，鼓勵學生從真實生活與社區需求中發想，並運用 Gen-AIoT 技術設計智慧書藝系統。學生將自主構思應用主題，發展具創造力與文化價值的創新解決方案，例如智慧城市、智慧教育與智慧藝創的融合應用。

Design 設計

透過完整設計流程，讓學生從概念、筆劃構圖到互動介面設計皆能參與。結合感測器、AI 模型與書法美學，發展出強調科技和美感整合的智慧系統設計，培養具備跨域美感與工程邏輯思維的創新人才。

Explore / Experience 探索/體驗

學生需進行實作與展示，透過作品驗證與修正設計方案，從中認識技術與藝術應用間的落差與挑戰。強調在實務中學習與反思的過程，激發學生主動學習、創新思考與問題解決的能力，並培養其持續探索與改良技術的動能。

AI 人工智慧應用實踐

學生將開發具書法筆跡生成、姿態追蹤與互動控制功能的 AI 系統，並運用生成式 AI、Edge AI 與 AIoT 技術，將人工智慧融入藝術創作與書藝教學中，實踐跨領域智慧應用。

Skills 技能建構與專業養成

課程強調「電機 × 藝設」雙軌養成，包含：

- 電機技術：感測器、資料處理、AI 模型訓練
- 藝術設計：書法結構、美感設計、視覺介面
- 學生從創意發想到實作成果全程參與
- 培養跨域溝通能力、團隊協作與專案執行力

融合科技與書法，創造具文化深度與實用價值的智慧應用。

學生不僅掌握 AI 與 AIoT 技術，更能透過書藝創作實踐數位轉型與人文科技的跨域整合。

促使學生成為未來引領智慧藝文創新與地方文化科技發展的關鍵人才！

三、整體活動執行成果效益

主要教學法	課程大綱		學習(質化/量化)成果
	學習主題	執行過程	
VIP	跨年級合作專題設計與實作	1. 由研究生與大學生各年級學生組成跨層級專題團隊 2. 透過高年級生引導低年級生進行專題規劃、資料蒐集與技術實作 3. 分工合作，整合書法與電機領域知識，共同完成創新智慧系統專題 4. 定期進行進度簡報與成果展示，強化專案管理與知識傳承	學習成果： • 實際完成 9 項具有創新價值之專題成果 • 培養學生的專題企劃能力與問題解決能力 • 強化跨年級的團隊合作與領導統御能力 • 提升學生實務經驗與職涯競爭力
TTBL	跨領域整合與問題導向實作專題（電機 × 藝術設計）	1. 結合電機工程與藝術設計相關師生，組成跨域合作社群 2. 聚焦於「智慧計算實驗」與「現代書藝研究」之實務問題，透過腦力激盪與討論，提出具有創意與實用性的整合型專題構想 3. 學生以小組形式進行資料蒐集、技術整合與創作實踐，落實跨領域融合應用 4. 強調以學生為學習主體，藉由專題實作深化技術應用與藝術表達之跨域整合能力	學習成果： • 實際完成 4 組結合電機與藝術設計的跨領域合作專題 • 強化學生整合技術知識與藝術思維的跨域應用能力 • 培養學生針對真實問題進行需求分析與創意解決的能力 • 提升團隊協作與跨專業溝通能力，累積解決複雜問題之實作經驗 • 增進學生對在地議題的關注與社會參與意識，深化學用合一的連結

四、多元評量尺規

【質化指標】

- 學生能融合 AI、AIoT 技術與藝術設計思維，提出具創意與實用價值的跨域解決方案。
- 學生展現跨領域溝通與協作能力，能有效分工並整合專業技術與設計表達。
- 學生具備將知識轉化為原型作品之能力，能從設計、開發到展示完整表達專題核心價值。

【量化指標】

- 期末專題組別共 13 組，包含 4 組兩系跨領域專題組
- 兩系跨領域專題組學生共 18 人，藝設系 6 人、電機系 12 人
- 兩系聯合跨領域專題進度審查共 3 次

五、學生整體意見與回饋（整體活動滿意度、文字意見回饋等）

根據期中與期末的學生課程回饋調查與文字意見整理顯示，本課程透過電機與藝術設計領域的跨域合作，有效激發了學生的學習熱情與投入程度。特別是在生成式 AI 與智慧物聯網（AIoT）技術結合藝術創作的應用場景中，學生普遍認為課程內容具備挑戰性，同時也帶來高度的成就感。透過團隊合作與專題實作的歷程，學生不僅學會整合兩大領域的知識，更在提案規劃、技術實現與成果展示等階段中，培養出自主學習與解決問題的能力。

多數學生也提到，課程提供了充分機會發揮自身專長，並從來自不同領域的組員身上獲得新視角與創意激盪。許多學生認為其專題成果具備延伸應用的潛力，並期待後續能進行公開展示、申請專利，甚至參與創新競賽。

儘管整體評價良好，仍有少部分學生反映部分技術教材對初學者而言有一定門檻，建議未來能提供更多基礎入門資源或調整教學進度以因應差異。此外，有學生希望課程中段的評量能更明確列出標準與回饋時程，以便及時調整學習策略。

針對上述建議，教學團隊也規劃了具體的因應措施，包括：

- 調整課程節奏與難度，讓學生能更順利銜接並掌握學習內容；
- 加強課堂互動與引導，透過提問、小組討論等方式促進異質背景的學生交流；
- 提供技術支援，為初學者補充基礎知識，安排助教協助及設立教學諮詢時段；
- 擴展成果發表機會，協助學生將優秀專題作品推向校內外平台，進一步申請專利或參賽，提升學習動機與實作價值。

整體而言，本課程成功整合跨域知識與實作訓練，不僅拓展了學生的學習視野，也奠定了其面對未來挑戰的核心能力。

六、檢討與建議

當前困難/問題	未來改善/精進
跨領域小組合作中，部分學生對異領域溝通與專業理解有所落差，產生協作障礙與作品分工不均情形。	未來課程可於初期安排雙方專業導論課程與成功合作示例，幫助學生建立基本認知與合作語言，減少學期中溝通摩擦。
成果展示機會有限，學生缺乏實體舞台呈現學習成果，無法獲得外部回饋。	可爭取校內成果展、跨院系聯展或與產業單位合作辦理公開發表，強化學生動機與應用實感。
實作階段中，部分學生缺乏系統性的專題規劃能力，導致專題進度落後或成效不均。	課程中增加「專題企劃書」撰寫與簡報訓練環節，搭配高年級引導與中期進度檢討，強化專題執行與專案管理能力。
藝術類學生表示對程式語言不熟悉，導致參與技術面時有挫折感，信心受影響。	增加跨領域配對策略與技術支援，設計互補任務與雙向學習機制，讓藝術與電機學生各展專長並相互成長。

陸、活動紀錄表

活動主題	課程演講—打開 AI 視覺之眼
活動日期	_2025_年_4_月_30_日
活動地點	理工二館 B201、遠端 Google Meet
演講者	國立陽明交通大學楊哲旻研究員
參與人數	37(學生)+3(助教)
活動內容	<p>● 活動或講座進行方式與內容</p> <p>本次講座型工作坊採實體與線上同步進行，由國立陽明交通大學博士生楊哲旻研究員主講。楊研究員長期致力於 AI 人工智慧領域，專注於深度學習、大數據分析與影像處理技術，具備豐富的教學與實作經驗，曾多次擔任人工智慧相關工作坊講師。目前其研究方向聚焦於神經科學領域中的腦機介面與 EEG 腦波訊號分析。</p> <p>本場講座以「打開 AI 視覺之眼」為題，從電腦視覺與物件偵測的核心理論切入，深入淺出地說明 YOLO 等主流模型的原理與實務應用方式，並透過實際專案案例的分享，引導參與者理解如何實作並部署物件偵測模型於真實場域中。整場活動歷時三小時，內容安排循序漸進，特別適合零基礎或初學者參與。講師講解清晰、示範具啟發性，與會者反應熱烈，收穫良多。</p>
活動回饋 與 成效	<p>● 意見與回饋</p> <ol style="list-style-type: none"> 1. 多數學生表示講座內容深入淺出，讓他們首次理解「電腦視覺」與「YOLO 演算法」的基本邏輯與應用方式。 2. 學生指出講師以實際應用場景舉例，如交通監控、人臉辨識與自動化工廠，有助於理解物件偵測的實務價值。 3. 有學生建議未來可加入互動實作演練，提升操作參與感。 <p>● 成效</p> <ol style="list-style-type: none"> 1. 增進學生對電腦視覺與 AI 應用的基礎認識，激發學習動機。 2. 建立 AI 視覺領域的初步概念，為後續課程或專題奠定基礎。 3. 強化本課程（智慧科技跨域應用與實作）中學生對 AI 工具的實作理解與跨領域連結。
活動剪影(請檢附二至四張活動照片，並予以簡述)	



打開AI視覺之眼

入門物件偵測，零基礎也能學會！

深入了解電腦視覺與物件偵測的核心技術，從基礎理論到模型實作，掌握YOLO等主流演算法。透過實戰專案，提升應用與部署能力。歡迎各界師生、AI愛好者、開發者與資料科學家參加本次Object Detection Workshop，一起探索AI視覺的無限可能！

課程對象 國立東華大學師生

課程地點 Google Meet 線上課程(同步)
國立東華大學理工二館B201 網路實驗室

內容時間 2025/04/30 18:30 – 21:30

課程講師 楊哲旻 研究員
國立陽明交通大學電機學院
腦科技中心

主辦單位 國立東華大學 教學卓越中心
教學卓越中心 與 電機工程學系 三創教學計畫

協辦單位 東部人工智慧與智慧物聯網產學技術聯盟

聯盟網址 粉絲專頁

該課程的工作坊海報



Google Meet 上課之照片

活動紀錄表-2

活動主題	課程演講—影像分割基礎實戰：精準視覺，像素級 AI 洞察
活動日期	_2025_年_5_月_7_日
活動地點	理工二館 B201、遠端 Google Meet
演講者	國立陽明交通大學楊哲旻研究員
參與人數	37(學生)+4(助教)
活動內容	<p>● 活動或講座進行方式與內容</p> <p>本活動為講座型實作導向課程，採實體與線上同步進行，邀請楊哲旻研究員以「影像分割基礎實戰」為主題，深入淺出介紹電腦視覺中語義分割、實例分割與 U-Net/Mask R-CNN 等深度學習模型應用技巧。</p> <p>講師從影像像素理解出發，引導學員掌握影像分割技術的核心概念與 AI 模型的實作邏輯，並結合應用案例說明技術在醫療影像、智慧製造等場域的潛力與應用價值。</p> <p>整體課程由理論講解搭配應用示例，提供零基礎入門者一套清晰的學習架構，廣受與會學生好評。</p>
活動回饋與成效	<p>● 意見與回饋</p> <ol style="list-style-type: none"> 1. 學生普遍表示講座內容清晰，講師具備高度教學熱忱，能將複雜技術講解得簡單易懂。 2. 多位學生回饋：「以前聽過影像分割但沒理解，這次終於知道為什麼 U-Net 和 Mask R-CNN 在實際應用上那麼重要。」 3. 有學生建議：未來若能搭配程式碼操作或即時 DEMO，將更有助於理解。 <p>● 成效</p> <ol style="list-style-type: none"> 1. 幫助學生理解影像分割在 AI 視覺領域的實際應用場景。 2. 建立 AI 視覺技術從分類到像素級分析的知識脈絡，擴大學生對 AI 應用範疇的視野。 3. 增強學生對深度學習與 AI 演算法背後原理的興趣，促進後續進階課程的學習動力。
活動剪影(請檢附二至四張活動照片，並予以簡述)	



影像分割基礎實戰

精準視覺・像素級AI洞察

本課程將帶您從零開始掌握影像分割的關鍵技術，深入探討語義分割、實例分割等核心概念，並實作如U-Net與Mask R-CNN等深度學習模型。適合AI初學者、開發者、資料科學家與教師參與，透過實作案例強化應用，精準解析每一個像素的價值！

課程對象

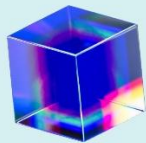
國立東華大學師生

課程地點

Google Meet 線上課程(同步)
國立東華大學理工二館B201 網路實驗室

內容時間

2025/05/07 18:30 – 21:30



課程講師

楊哲旻 研究員

國立陽明交通大學電機學院
腦科技中心

主辦單位



教學卓越中心 與 電機工程學系 三創教學計畫



協辦單位

東部人工智慧與智慧
物聯網產學技術聯盟



聯盟網址



粉絲專頁

該課程的工作坊海報



線上上課之照片

活動紀錄表-3

活動主題	電機&藝設兩系聯合跨領域專題進度審查
活動日期	_2025_年_5_月_16_日
活動地點	藝術學院 A258
演講者	廖慶華院長
參與人數	18(學生)+2(教授)+3(助教)
活動內容	<p>● 活動或講座進行方式與內容</p> <p>本次活動為電機與藝術設計兩系跨域合作專題進度發表與成果分享座談，由學生分組報告目前專題製作之主題概念、應用技術與整合成果。每組學生簡報約 10 分鐘，並開放現場提問與互評討論，實體於藝術學院 A258 教室進行。</p> <p>本次參與學生共 18 人，涵蓋電機系與藝設系跨年級成員，另有授課教師與助教共 5 人到場。</p> <p>在發表過程中，教師與助教亦積極參與講評與提問，針對各組專題的技術可行性、創意表現、實作方法與視覺美感等面向提供具體回饋與優化建議，有助於學生釐清問題、修正方向，並強化後續專題執行的完整度與表現力。</p>
活動回饋與成效	<p>● 意見與回饋</p> <ol style="list-style-type: none"> 1. 參與學生表示，能在課堂中進行階段性成果發表，有助於釐清專題方向與收斂執行策略，提升團隊合作效率。 2. 有學生反映，在實作過程中常遇到技術與美感表現之落差，透過公開交流與教師建議，對調整設計流程與操作方式更有信心。 3. 教師回饋指出，學生能將 AI 與藝術進行具體結合，部分作品已具公開展示潛力，建議持續追蹤並輔導申請創新專利或參與展演機會。 <p>● 成效</p> <ol style="list-style-type: none"> 1. 促進學生對跨領域專題設計之理解與溝通能力，並提升其簡報與表達技巧。 2. 綜整中期專題成果，協助教師進行學習成效評估與後續教學調整。 3. 強化學生在 AI 技術應用與藝術創作整合的實踐能力，為期末專題展示奠定基礎。
活動剪影(請檢附二至四張活動照片，並予以簡述)	

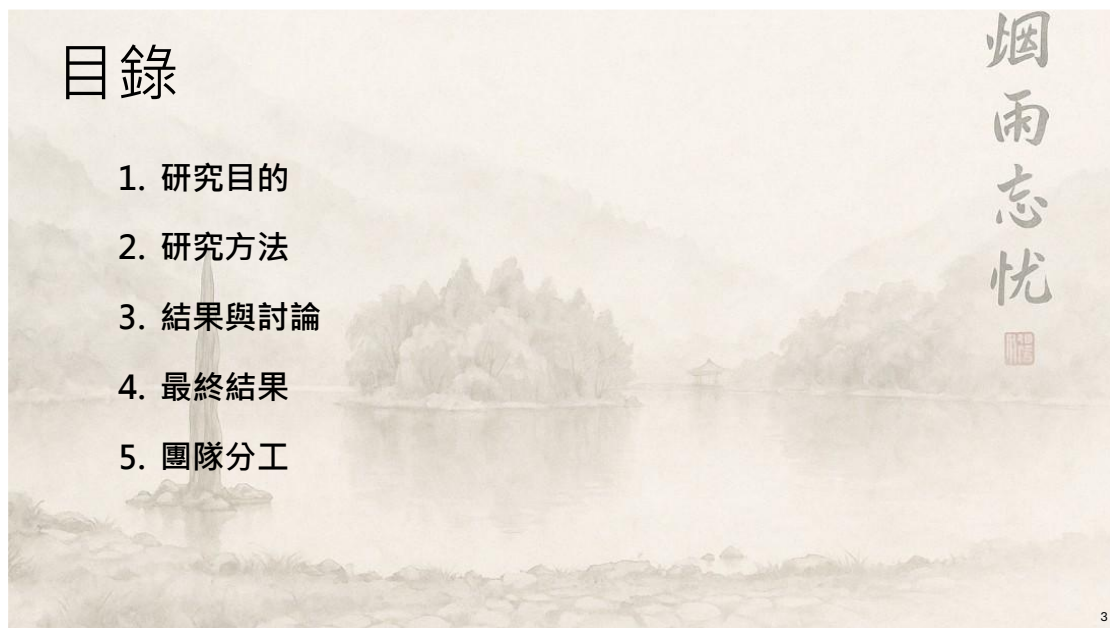


跨領域組別分享專題理念



跨領域組別分享專題理念

專題 1：書法立體化印章



作品理念

- 傳統印章製作方式需要仰賴雕刻並且使用一些極為珍貴的材料製作，客製化門檻高、所需時間長，並且對環境有害。
- 3D列印技術具備快速成型、低成本試作、自由設計造型等優勢，適合應用於客製化印章的製作。
- 本研究希望透過實際操作與案例分析，驗證3D列印技術在印章製作上的可行性與實用性。
- 對於不懂書法的人，可以用拍照的方式直接了解印章中這十二生肖的字屬於是哪種字體。

烟雨忘忧



5

二、研究方法

訓練模型

- 本專題採用ResNet50[1]作為模型在去判斷12生肖字和字體(行書、隸書、楷書)。
- ResNet 50的殘差結構可以解決梯度消失的問題，此外，這樣的設計使每一層只需學習輸入與輸出之間的殘差，大幅降低了模型的學習難度。
- 表一中，最中間的就是ResNet50的結構。

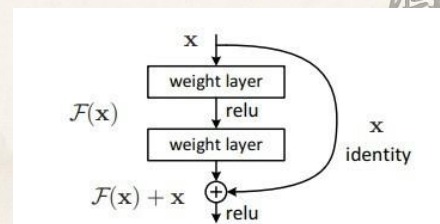


圖1、殘差塊示意圖

表一、ResNet50結構

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112-112	7-7, 64, stride 2				
3-3 max pool, stride 2						
conv2.x	56-56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3.x	28-28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4.x	14-14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5.x	7-7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
1-1		average pool, 10000 fc, softmax				
FLOPs		1.8-10 ³	3.6-10 ³	3.8-10 ³	7.6-10 ³	11.3-10 ³

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

7

環境與訓練參數

- 本專題環境在Visual Studio code上執行，去判斷12生肖字和字體(行書、隸書、楷書)。
- 訓練參數批量大小(Batch Size)為8、學習速率為0.0001、訓練次數為50次。若loss遲遲無法下降，則會將學習速率*0.5
- 使用部分微調：先只使用ResNet50 的前面絕大多數卷積層與池化層，當作特徵提取器；在其後面接上自己的分類層。等到分類層訓練穩定後，再把 ResNet50 最後一小段與分類層一起進行微調訓練。

烟雨忘忧

影像資料集

使用github提供的Chinese Calligraphy Dataset資料集內挑選出十二生肖並分類出字體，並在[集大家書法app](#)中挑選出各生肖的字體圖片。



圖2、圖片蒐集

表二、各字體生肖資料集(總共703張)

	rat	ox	tiger	rabbit	dragon	snake	horse	goat	monkey	rooster	dog	pig
fame(行書)	14	41	30	27	57	28	35	22	15	35	31	18
lishu(隸書)	11	9	23	11	19	12	11	7	7	22	5	6
regular(楷書)	12	25	14	15	31	11	23	10	19	18	13	16

影像前處理與增量

(一)前處理

因為所取得的資料大部分都為黑底白字，所以在資料前處理部分，我們會將書法文字先進行黑白化

(二)增量

透過增加雜訊、縮放，使整體資料擴增至5倍，以提高整體資料集數量且提高模型泛化能力(共4218張)

(三)分割

將增量後的資料以8:2的比例分為訓練集(共3282張)和驗證集(共936張)，以供之後的模型訓練



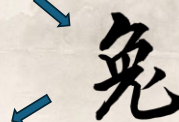
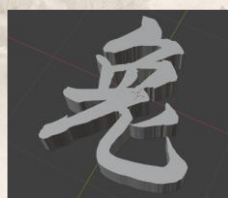
圖3、圖片提取並轉化



圖4、增量後圖片

提取文字

文字：十二生肖字體請
藝設系碩一班
張玉環同學書寫完成



紙張 → 平面轉 → 點陣圖

→ 將點陣圖導入 3 D 軟體

圖5、照片轉3D流程示意圖

11

設計印章圖形

利用chatGPT生成2D圖片



生成一隻兔子的3D模型



再使用meshy AI 將2D圖片轉換成3D模型

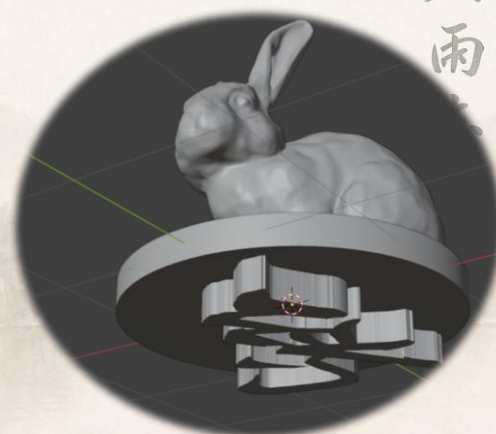
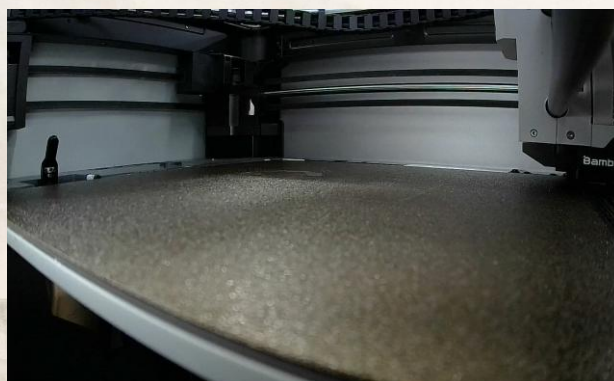


圖6、3D兔子印章建模

最後再使用Blender將生成模型不合理處修飾，再與字合併

12

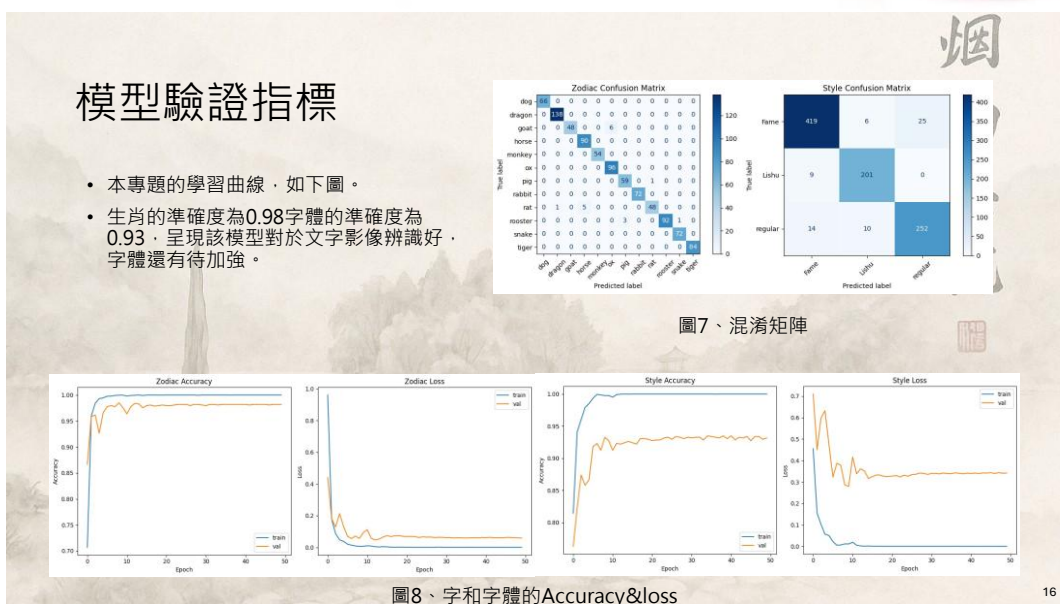
三滴(D)列印



13



三、結果與討論



實際預測

- 使用與藝設系合作同學的書法文字進行預測



圖9、預測結果

17

實際預測

- 使用與藝設系合作同學的書法文字進行預測



圖9、預測結果

17

←圖12、老鼠印章成果



圖13、牛、虎印章成果→



22



23



23

字體字型辨識



25

- 從未接觸過3D的建模軟體。
A：去網路上尋找線上教學的影片，並配合chatGPT操作我所需使用的功能。
- 不知道要如何建模動物。
A：為了配合這堂課的內容，我們找到了AI製圖，使用chatGPT生成2D圖片，再用meshy AI將2D圖檔轉成3D模型。
- 不會操作3D列印的軟體。
A：尋求認識的同屆或是學長的幫助。
- 在拆支撐時，原先設定的參數過於緊密，以至於拆支撐時會導致所需模型斷裂，並且字的洞難以移除。
A：更改支撐模式，並且購買美工刀根據原始檔一刀一刀移除。
- 水性墨水無法附著於塑膠材料。
A：改成買油性的墨水。
- 3D列印的材料有平面不平整的問題，導致墨水無法完整翻印至紙上。
A：我們買了透明硅膠當作底去附著墨水。



26

- 模型的準確度上不去和損失函數下不去。
A：在訓練過程中，將 backbone（主幹網路）凍結可以提升訓練效率，並防止預訓練權重被破壞。在凍結階段，特徵提取部分的參數保持不變，模型僅對後端分類層進行微調。由於不需回傳梯度至 backbone，顯存占用較少，訓練也更加穩定與高效。

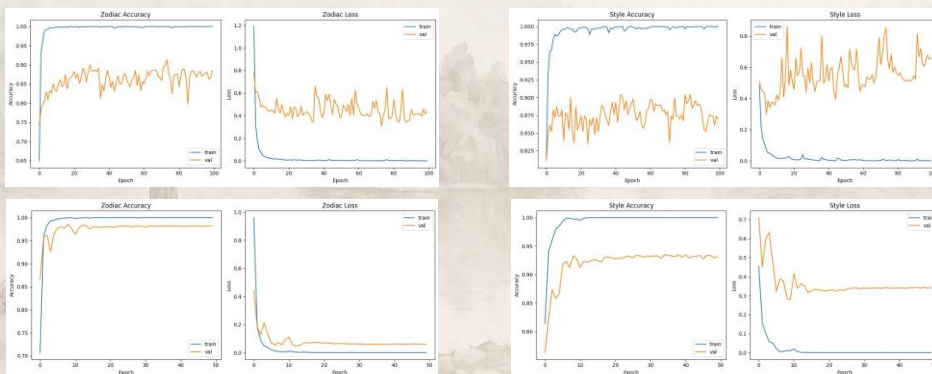


圖19、凍結使用前(上)和使用後(下)的Accuracy&loss

26

8. ResNet50 訓練速度明顯偏慢，觀察後發現模型僅使用 CPU 運算。

A：參照[CUDA cuDNN TensorFlow-GPU 安裝教學](#)，安裝 CUDA 11.2 與 TensorFlow 2.6.0，並確保其他相關套件（如 cuDNN）與 TensorFlow 2.6.0 相容。

烟雨忘忧

27

五、團隊合作和貢獻

29

張玉環	負責書寫十二生肖文字字體、字體辨識資料提供、提供書寫文字影片	20
陳彥兆	蒐集訓練圖片、訓練模型、處理圖片、製作簡報	26
葉裕泉	擷取文字、製作動物模型、購買印章製作所需配件、製作印章底座、尋求學長協助使用3D印表機、製作簡報。	27
朱柏安	蒐集訓練圖片、處理圖片、訓練模型、製作簡報	26

烟雨忘忧

29



專題 2：書法模擬評分遊戲



目錄

- 1 研究問題&目的
- 2 研究方法及流程
- 3 結果展示
- 4 未來展望
- 5 參考資料
- 6 團隊分工

2

研究目的

因為我們覺得臨摹書法是一件很需要信心的事情，所以想要做一個可以判斷使用者的臨摹與範例文字是否相像的小遊戲，並透過給分、類似闖關的方式提高使用者的興趣對於臨摹書法的興趣。

優點

- 書法模擬教學
- 透過AI 給分
- 提高練習書法的興趣

3

選擇字體的含義

《飛合轉承起順望恩盼岸》

副標：遷徙記憶中的情感書寫與回望之途

📌 作者：來自馬來西亞與台灣的兩位創作者

4



5

書法書寫這幾個字的含義：

在於用筆墨承載情感與記憶。
每一筆、每一劃，不只是書寫文字，而是在書寫心路歷程。
這些字是我們遷徙與相遇的縮影，
透過書法，它們變得有重量、有溫度。
用書法來寫，是為了讓這段旅程被「看見」、被「感受」，
也讓每個字，成為一種靜靜說話的情感記號

6

研究方法及流程

點選連結 進入網頁

網頁畫面有藝設系同學寫出的字
和一個可以拿來寫字的框

模擬字體

使用者要模仿藝設系同學寫出來的
字跡，並寫在框內

字體->圖片

將使用者寫出來的字轉成圖片的
格式

判斷相似度

用深度學習判斷使用者寫出來的
字跟藝設系同學的字跡有多少相
似

7

以下是網頁製作的程式

```

<div style="color: red; position: absolute; top: 100px; left: 100px; text-align: center;">請在側邊欄輸入姓名
```

```

const img = new Image();
img.src = URL.createObjectURL(input.files[0]);
img.onload = async () => {
  const canvas = document.createElement('canvas');
  canvas.width = canvas.height = 224;
  const ctx = canvas.getContext('2d');
  ctx.drawImage(img, 0, 0, 224, 224);

  const imgData = ctx.getImageData(0, 0, 224, 224);
  const { data, width, height } = imgData;
  const float32 = new Float32Array(3 * width * height);

  for (let i = 0; i < width * height; i++) {
    float32[i] = data[i * 4 + 0] / 255; // R
    float32[i + 1] = data[i * 4 + 1] / 255; // G
    float32[i + 2] = data[i * 4 + 2] / 255; // B
  }

  const inputTensor = new ort.Tensor('float32', float32, [1, 1, height, width]);

```

```

<div style="text-align: center;">
  <h1>手寫相似度 /h1>
  <input type="file" id="imageInput" accept="image/*" />
  <button id="predictButton">判斷相似度</button>
  <p>相似度: <span id="result"></span></p>
</div>

<script>
  let session = null;
  window.addEventListener('DOMContentLoaded', async () => {
    try {
      session = await ort.InferenceSession.create('he.onnx');
      console.log('ONNX model loaded.');
```

畫布程式碼

```

// PointerEvent Apple Pencil / 滑鼠 / 手指
canvas.addEventListener('pointerdown', (e) => {
  isDrawing = true;
  [lastX, lastY] = [e.offsetX, e.offsetY];
});

// pointer.pressure + canvas.lineTo + stroke 筆壓感應
canvas.addEventListener('pointermove', (e) => {
  if (!isDrawing) return;

  const pressure = e.pressure || 0.5; // 真實筆壓，滑鼠時為 0.5
  const size = parseFloat(sizeSlider.value) * pressure;

  ctx.lineCap = 'round'; // 收尾圓滑
  ctx.lineWidth = 'round'; // 轉折圓滑
  ctx.strokeStyle = 'black';
  ctx.lineWidth = size;

  ctx.beginPath(); // 每筆路徑獨立處理
  ctx.moveTo(lastX, lastY);
  ctx.lineTo(e.offsetX, e.offsetY); // 連接上下一筆的筆畫
  ctx.stroke();

  [lastX, lastY] = [e.offsetX, e.offsetY];
});

```

```

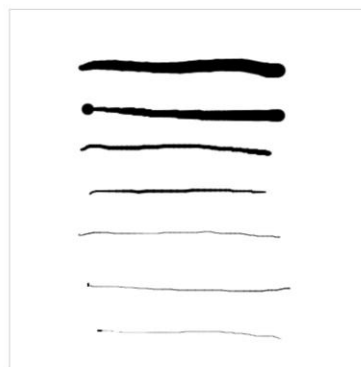
const stopDraw = () => isDrawing = false;
canvas.addEventListener('pointerup', stopDraw);
canvas.addEventListener('pointerout', stopDraw);
canvas.addEventListener('pointercancel', stopDraw);

// 儲存圖片
document.getElementById('btn-save').addEventListener('click', () => {
  const link = document.createElement('a');
  link.download = 'calligraphy.png';
  link.href = canvas.toDataURL();
  link.click();
});

// 清除畫布
document.getElementById('btn-clear').addEventListener('click', () => {
  ctx.clearRect(0, 0, canvas.width, canvas.height);
});

```

毛筆書法模擬（支援筆壓）



畫布中的筆壓感應

筆粗固定時，用不同力道寫出
來的粗細不同

模型訓練程

式

```
anchor_dir = 'an'
other_dir = 'xxx'
model = train(anchor_dir, other_dir, epochs=500)
```

```
Epoch 1/500, Loss: 0.6561
Epoch 2/500, Loss: 0.2632
Epoch 3/500, Loss: 0.2347
Epoch 4/500, Loss: 0.2491
Epoch 5/500, Loss: 0.2500
Epoch 6/500, Loss: 0.2472
Epoch 7/500, Loss: 0.2414
Epoch 8/500, Loss: 0.2548
Epoch 9/500, Loss: 0.2372
Epoch 10/500, Loss: 0.2060
Epoch 11/500, Loss: 0.2438
Epoch 12/500, Loss: 0.2161
Epoch 13/500, Loss: 0.2186
Epoch 14/500, Loss: 0.2192
Epoch 15/500, Loss: 0.2041
Epoch 16/500, Loss: 0.2173
Epoch 17/500, Loss: 0.2321
Epoch 18/500, Loss: 0.2443
Epoch 19/500, Loss: 0.2432
Epoch 20/500, Loss: 0.2273
Epoch 21/500, Loss: 0.2458
Epoch 22/500, Loss: 0.2063
Epoch 23/500, Loss: 0.2266
Epoch 24/500, Loss: 0.2276
Epoch 25/500, Loss: 0.2284
...
Epoch 497/500, Loss: 0.2506
Epoch 498/500, Loss: 0.2502
Epoch 499/500, Loss: 0.2090
Epoch 500/500, Loss: 0.2245
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

```
class ResNetFmEmbedder(nn.Module):
    def __init__(self):
        super(ResNetFmEmbedder, self).__init__()
        resnet = models.resnet50(weights=models.ResNet50_Weights.DEFAULT)
        modules = list(resnet.children())[:-1] # 移除最後分類層
        self.resnet = nn.Sequential(*modules)
        self.fc = nn.Linear(resnet.fc.in_features, 128) # 特徵向量降維

    def forward(self, x):
        x = self.resnet(x)
        x = x.view(x.size(0), -1)
        x = self.fc(x)
        return x
```

```
def train(anchor_dir, other_dir, epochs=10, batch_size=8, lr=1e-4):
    dataset = SiameseDataset(anchor_dir, other_dir, transform)
    loader = DataLoader(dataset, batch_size=batch_size, shuffle=True)

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    model = ResNetFmEmbedder().to(device)
    criterion = ContrastiveLoss(margin=1.0)
    optimizer = torch.optim.Adam(model.parameters(), lr=lr)

    losses = []

    for epoch in range(epochs):
        model.train()
        total_loss = 0.0
        for img1, img2, label in loader:
            img1, img2, label = img1.to(device), img2.to(device), label.to(device)
            out1 = model(img1)
            out2 = model(img2)
            loss = criterion(out1, out2, label)
            optimizer.zero_grad()
            label = label * 2 - 1
            loss.backward()
            optimizer.step()
            total_loss += loss.item()

        print(f"Epoch {epoch+1}/{epochs}, Loss: {total_loss/len(loader):.4f}")

    torch.save(model, 'resnet_siamese.pth')
```

10

```
if __name__ == '__main__':
    model = load_model(model_path=r"C:\Users\yun\Desktop\write\resnet_siamese.pth")
    score = similarity(model,
                       r'an螢幕擷取畫面_2025-05-24_234717-removebg-preview.png',
                       r'testtttt標楷(大圖大字).png')
    print(f"相似度: {score:.2f}%")
```

✓ 0.8s
相似度: 81.31%

恩

609*410
恩
75.3%

609*410
恩
73.82%

613*407
恩
71.63%

6.78%

11

```
preprocess = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
])

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
siamese = ResNetFmEmbedder().to(device)
siamese.load_state_dict(
    torch.load(r"C:\Users\yun\Desktop\write\resnet_siamese.pth",
              map_location=device))
siamese.eval()

anchor_path = r"C:\Users\yun\Desktop\write\2025-05-25_004506-removebg-preview.png"
img = Image.open(anchor_path).convert('RGB')
x = preprocess(img).unsqueeze(0).to(device)
with torch.no_grad():
    anchor_feat = siamese(x).squeeze(0)

class SingleImageSimilarity(nn.Module):
    def __init__(self, backbone: nn.Module, anchor_emb: torch.Tensor):
        super().__init__()
        self.backbone = backbone
        self.register_buffer('anchor', anchor_emb)

    def forward(self, img: torch.Tensor):
        ...
        img: [0, 3, 224, 224]
        回傳 similarity: [0, 1] · 數值範圍約在 [-1,1]
        feat = self.backbone(img)

        feat_norm = feat / feat.norm(p=2, dim=1, keepdim=True)
        anchor_norm = self.anchor / self.anchor.norm(p=2)
        anchor_norm = anchor_norm.unsqueeze(0)
        sim = (feat_norm * anchor_norm).sum(dim=1, keepdim=True)
        return sim

final_model = SingleImageSimilarity(siamese, anchor_feat).to(device)
final_model.eval()
```

```
> torch.onnx.export(
    final_model,
    dummy_in,
    "zuon.onnx",
    input_names = ["input"],
    output_names = ["similarity"],
    dynamic_axes={
        "input": (0: "batch_size"),
        "similarity": (0: "batch_size"),
    },
    opset_version=11,
    do_constant_folding=True,
)
print("成功匯出")
```

匯入ResNet50

訓練500次

判斷相似度

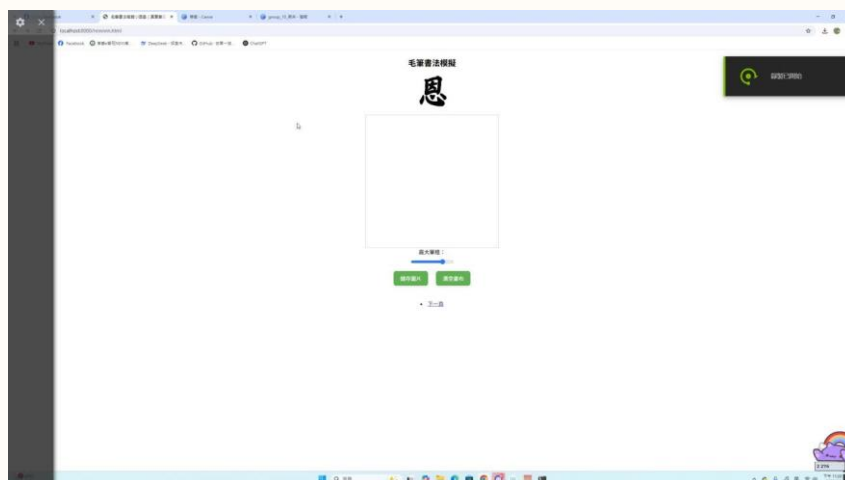
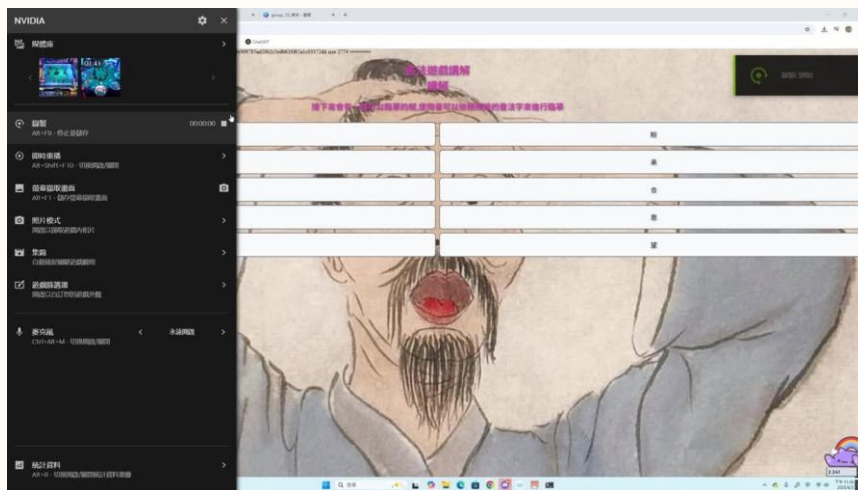
匯出模型

模型轉成.onnx

結果展示

系統可即時判斷使用者所寫字體與範例字的相似度，
達到書法練習與自我評量的功能。

12



26

未來展望

可進一步強化筆壓與筆畫順序的辨識模型，提升真實臨摹互動體驗。

加入語音或語意引導練習。

擴展可比對字樣類型，未來可包含草書、行書等風格字體。

讓系統可以辨識臨摹的字是否與範例相同的功能。

加入使用者練習歷程儲存與評分記錄系統。

13

參考資料

1. https://developer.mozilla.org/zh-TW/docs/Web/API/Canvas_API/Tutorial
2. <https://reurl.cc/rEGj9O>
3. <https://www.remove.bg/zh-tw/upload>
4. <https://medium.com/@rossleecooloh>
5. <https://blog.csdn.net/wuqitong123/article/details/132725824>
6. https://blog.csdn.net/qq_40243750/article/details/117221536

14

舞-3D列印成果

藝設系
611374014林翠珊

電機系
411123063陳銘泰
411123062楊宸緯
411123049楊家宇

研究目的-Ambigram sculpture

在尋找靈感的過程中，我們受到英國藝術家 Lex Wilson 的作品 Real or Fake的啟發，他透過多視角字體設計，讓觀者從不同角度觀看時可看到完全不同的單詞，成功結合了視覺錯視與空間幾何，展現出新型態的文字表達形式。

這種概念讓我們聯想到中文在字體設計上的一大特色：其筆劃結構與風格變化非常豐富，不同書體之間所呈現出的節奏、筆意與情感也不太一樣。

因此，我們最終決定以中文的「行書」與「草書」作為研究對象，選擇其視覺差異較大的字體風格，嘗試透過3D建模與列印技術，設計出一組能夠從不同視角呈現兩種字體的立體文字作品。期待有溫度的文字和冰冷的科技產生新的火花。



研究方法-3D列印建模

1. 評估專案的方法和技術是否有效和可靠:先用在網路上找了貓、狗的實心圖樣，經過inkscape處理成點陣圖匯出成SVG。
2. 在blender中測試，發現若直接匯入SVG檔會破圖，所以採用材質匯入PNG圖像在用錯位的方式就能成功建模。拿到書法的稿子之後，先用影印機掃描成圖檔，用inkscape做對比度、大小比例和去除多餘噪點處理(滴到墨或影印機稿紙上的髒汙)在經過前面測試的方法得到一個完整的模型
3. 是否能夠產生有用的結果及發現:實際用布林做出實體後發現範例的圖是有浮空的，導致建模後的也有浮空。若之後用書法時連成一筆寫，就能解決這個問題。



2

結果與討論-成果展示



3

結果與討論-成果展示



4

結果與討論-成果展示

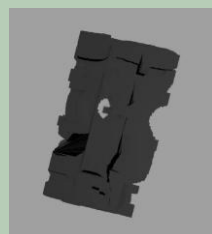
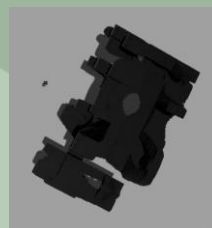


5

未來展望-增加文字占比

目前的模型設計中，我們已經完成了四面視角的文字呈現，透過增加文字在結構中的佔比，讓從指定角度觀看時的辨識度大幅提升。

不過目前仍有兩個側面尚未加入文字設計，未來我們希望能進一步利用這些空白面，加入新的字體或文字訊息，讓整體作品變得更完整，也讓觀者在不同方向移動時有更多層次的發現與互動可能。



6

未來展望-大型藝術裝置

我們已經透過3D列印技術實現從不同角度觀看呈現不同字體的作品。

但是受限於3D列印有著大小的限制，我們希望未來可以將這個概念放大，打造一座大型的公共裝置藝術，讓更多人能夠親身體驗這種視覺錯視的魅力。

例如在城市的某個角落，矗立著一座巨大的雕塑。當你從一側走近時，看到的是行書的「和」字；繞到另一側，則變成了草書的「平」字。

這不僅是一件藝術品，更是一種與觀者互動的體驗，讓人們在日常生活中重新認識文字的美感與多樣性。這樣的裝置藝術不僅能讓文字更有溫度，還能激發人們對文字與藝術的興趣，成為城市中的一處文化地標。



香港裝置藝術作品「Time Loop」

7

參考資料

- Free Chinese Fonts. (n.d.). *Calligraphy Archives*. Retrieved May 29, 2025, from [https://www.freechinesefont.com/category/calligraphy/Free Chinese Fonts](https://www.freechinesefont.com/category/calligraphy/Free%20Chinese%20Fonts)
- Artsy Shark. (2016, August 16). *The Art of 3D Typography: An Interview with Noah Camp*. Retrieved May 29, 2025, from https://www.artsyshark.com/2016/08/16/the-art-of-3d-typography-an-interview-with-noah-camp/Artsy_Shark
- PrintLab. (n.d.). *3D Printed Typography Project*. Retrieved May 29, 2025, from <https://weareprintlab.com/courses/3d-printed-typography/PrintLab>
- All3DP. (2023, April 23). *Blender 3D Printing Tutorial for Beginners*. Retrieved May 29, 2025, from <https://all3dp.com/2/blender-3d-printing-tutorial/All3DP>
- Wired. (2012, April 23). *Designers 3D-print typeface-inspired 'Arkitypo' alphabet*. Retrieved May 29, 2025, from <https://www.wired.com/story/arkitypo-3d-printed/WIRED+2WIRED+2Pinterest+2>
- 3DPrinting.com. (n.d.). *Reviving Tactile Typography with 3D Printing*. Retrieved May 29, 2025, from [https://3dprinting.com/news/reviving-tactile-typography-with-3d-printing/3D Printing](https://3dprinting.com/news/reviving-tactile-typography-with-3d-printing/3D%20Printing)
- Blender Studio. (n.d.). *Blender for 3D Printing*. Retrieved May 29, 2025, from <https://studio.blender.org/training/3d-printing/>

8

團隊分工

組員	工作內容	占比
楊家宇	1. Blender建模 2. 參與進度討論(兩次) 3. 主題發想 4. 期末簡報製作(研究方法) 5. 期末口頭報告(研究方法) 6. 成品修整	50%
楊宸緯	1. 期末簡報製作(全部除了研究方法) 2. 展示影片製作 3. 協調組員溝通 4. 組織開會並參與討論(組織一次參與兩次) 5. 期末口頭報告(全部除了研究方法)	30%
陳銘泰	1. 組織開會並參與討論(組織一次參與兩次) 2. 主題發想 3. 協調組員溝通 4. 成品拍攝 5. 期中簡報製作(全)	20%

9

感謝聆聽

專題 4：字體影像辨識 APP

智慧計算實驗

字體影像辨識
APP
FINISH !!

三創跨領域合作 第十三組

藝設:
411074052陳亭葳
電機:
411223021陳彥青
411223064陳冠銘
411223061廖宇晟

獨特性

Play Store上面目前沒有影像辨識書法應用

創新應用

創新性

相比於字典操作較直覺

書法字典

顏真卿書法字典

王羲之書法字典

米芾書法字典

書法辨識

春

視覺版面

手機應用程式主視覺

適用於節慶限定APP

可以隨手拍一張身邊的字體，
嘗試觸發隱藏的春聯

左上角的典藏紀錄曾經
蒐集過的字體

使用說明

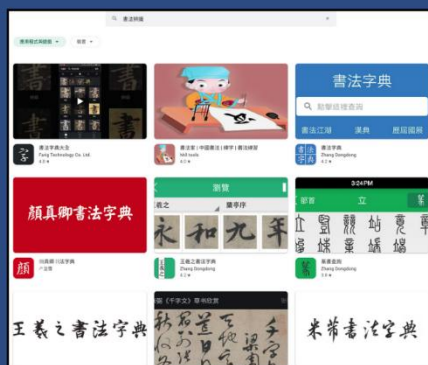
1. 打開APP，可選擇拍照或相簿辨識
2. 字典收藏
3. 收錄教育部70085個成語



創新應用

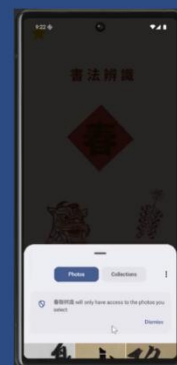
獨特性

Play Store上面目前沒有影像辨識書法應用



創新性

相比於字典操作較直覺



視覺版面

手機應用程式主視覺

適用於節慶限定APP

可以隨手拍一張身邊的字體，
嘗試觸發隱藏的春聯

左上角的典藏紀錄曾經
蒐集過的字體

使用說明

1. 打開APP，可選擇拍照或相簿辨識
2. 字典收藏
3. 收錄教育部70085個成語



視覺版面

手機應用程式主視覺

適用於節慶限定APP

具有影像識別以及相片儲存

儲存功能



← 封閉測試

封閉測試 - Alpha

建立及管理封閉測試版本，讓您管理的測試人員測試應用程式的預先發布版。[瞭解詳情](#)

設定封閉測試群組

未開放購買。暫定應用程式名稱「com.Anderson.calligraphyapp (unreviewed)」[?](#)

已完成 2 項，共 4 項 ^

設定測試群組

✓ 選取國家/地區

✓ 選取測試人員

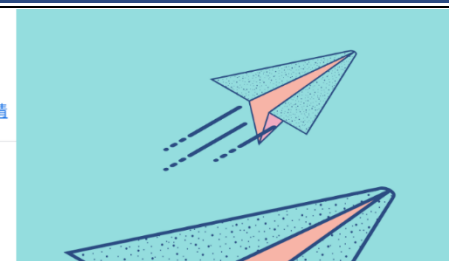
建立及推出版本

○ 建立新版本 >

🔒 預覽並確認版本

🔒 將版本送交 Google 審查

提交Play Store



資訊主頁

準備發布應用程式。[顯示更多](#)

春聯辨識

com.Anderson.calligraphyapp • 應用程式草稿

正式版

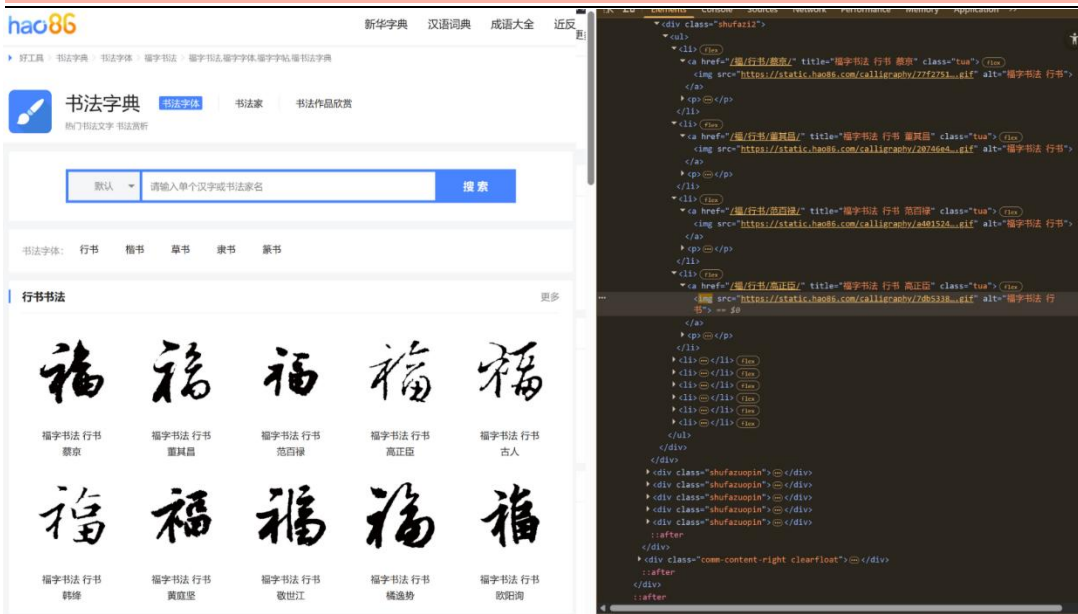
未開放購買

顯示測試群組 ▾

目前
加入測試人員
可以下載到APP
安卓



起點



ConditionalResizeWithTopCrop: 條件式的圖片縮放與頂部裁切。

`transforms.RandomApply([transforms.RandomResizedCrop(...)])`: 以一定機率隨機裁切並縮放圖片。

transforms.RandomHorizontalFlip: 以一定機率水平翻轉圖片。

transforms.RandomAffine: 隨機仿射變換，包括：

旋轉 (degrees)

平移 (translate)

錯切/剪切 (shear)

縮放 (scale)

transforms.RandomPerspective: 隨機透視變換。

`transforms.RandomApply([transforms.ElasticTransform(...)])` (如果 PyTorch 版本支援): 以一定機率進行彈性變換。

RandomBackgroundTransform: 自定義的隨機背景生成，包含：

純色背景 (Solid Color)

隨機點 (Random Dots)

隨機線條 (Random Lines)

隨機色塊 (Random Patches)

使用背景圖片 (Image Background) (如果提供了背景圖片目錄且其中有圖片)

transforms.ColorJitter: 隨機調整圖像的：

亮度 (brightness)

對比度 (contrast)

飽和度 (saturation)

色調 (hue)

`transforms.RandomApply([transforms.GaussianBlur(...)])`: 以一定機率進行高斯模糊。

`transforms.RandomAdjustSharpness` (如果 PyTorch 版本支援): 以一定機率隨機調整圖像銳度。

transforms.RandomErasing: 隨機擦除圖像中的一塊區域。

11種資料増量

觀察到背景影響準確度
調整後大幅增加辨識的準確度

原始模型(TinyVGG 5層)
準度不佳

替換模型:
ResNet18

可以辨識較困難的草書字體



轉檔
.pth to .pt

優點:

體積1/3(Quantization) 、

可以脫離 Python interpreter 、

可移植性(Model Deployment) 、

提升運算速度 (Performance Optimization)

```
checkpoint = torch.load(pth_model_path_str,
                        map_location=torch.device('cpu'))
state_dict = checkpoint['state_dict']

model_instance = LightResNet(
    num_classes=num_classes,
    block_type='basic', # 確保與訓練時一致
    dropout_rate=Config.DROPOUT_RATE, # 確保與訓練時一致
    use_layer1=True,
    use_layer2=True,
    use_layer3=True,
    use_layer4=True
)

# 處理 'module.' 前綴 (如果模型是用 DataParallel 或 DDP 儲存的)
new_state_dict = {}

for k, v in state_dict.items():
    name = k[7:] if k.startswith('module.') else k
    new_state_dict[name] = v

model_instance.load_state_dict(new_state_dict)
model_instance.eval()
model_instance.to(torch.device('cpu'))

scripted_model = torch.jit.script(model_instance)
scripted_model.save(pt_model_path_str)
print(f"JIT Scripted 模型已成功轉換並儲存至 '{pt_model_path_str}'")
```

想辦法訓練完的模型導入

將TorchScript導入
設計版面交互
添增開啟相機、相簿
收藏功能具有記憶性

謝謝大家



專題 1：影像分類模型在居家入侵監測的應用

113年智慧計算實驗專題

影像分類模型在居家入侵
監測的應用

組別： 第一組

成員： 411123025 杜政陽

411123055 楊承燁

目錄

1. 研究目的
2. 研究方法
3. 結果與討論
4. 未來展望
5. 參考資料
6. 團隊分工

研究目的 RESEARCH PURPOSES

在現今社會，監視器幾乎遍布於人們日常活動的各個空間，尤其在人口密集的都會區，如街道路口、公共場所與商業大樓等地點，更是隨處可見。且隨科技發展，有越來越多監視器結合人臉辨識與行為追蹤等智慧影像處理技術，應用場景也日益多元。例如交通系統中的車流監控與違規偵測、智慧工廠中的生產線監控與設備異常警示等，皆反映出影像處理技術在智慧監控領域的重要性與普及性。

然而，在居家環境與小型商業場域中，目前常見的監控應用多以入侵 監測與異常行為 偵測為主，較少具備辨識熟悉與陌生人物的能力。為了提升私人空間的安全性，我們希望將陌生監控技術導入這些場域，特別是在住家門口等敏感區域，透過即時的陌生人物識別與回應機制，提供更主動且智慧的防護措施。

3

研究方法 RESEARCH METHODS

(一)模型來源

我們使用的模型是 insightface 裡的 buffalo_l，其架構是建構在 ResNet50 或類似的深度卷積神經網路變體，包含 50 層卷積層與殘差模組，且專門為人臉辨識任務設計。

這個模型的影像輸入尺寸通常是 RGB 三通道 112x112 像素的圖片，且在特徵提取前，會對臉部進行對齊處理（基於關鍵點如眼睛、鼻子位置），使模型更具泛用性。而輸出則是512 維實數向量，表示輸入臉部的高階特徵。



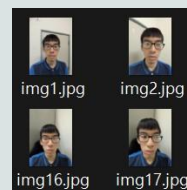
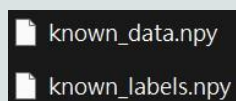
4

研究方法 RESEARCH METHODS

(二)特徵資料集

我們使用的 特徵資料集是蒐集每位 人物的不同臉部角度照片所 製作的，每位人物在預處理資料集中都會有接近 200張的相片。

之後會將所有相片丟給模型進行人臉偵測與特徵向量提取，並將每張照片生成的特徵向量存入資料集之中，並在另一檔案加入對應的人名標籤。



5

研究方法 RESEARCH METHODS

(三)實際運作邏輯

我們程式的實際運作邏輯是將攝影機畫面 (或者影片檔案)輸入模型處理，並將模型輸出的特徵向量與熟悉 資料集中每筆資料 進行餘弦相似度運算 (cosine similarity)，並根據所有運算結果的最大值是否大於我們設定的臨界值來判斷此資料是否為陌生資料集。

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

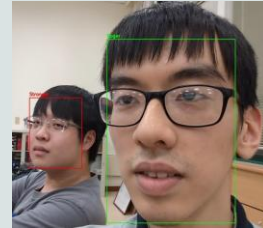
6

結果與討論 RESULTS AND CONCLUSIONS

實驗結果

實驗最終成果有達到我們目的中預想的結果，程式會根據畫面中所有人臉與資料集的運算進行分類，並實際在畫面中進行框選與標註，讓使用者能夠更直觀了解程式的實際輸出。

而在遇到陌生資料集能夠即時擷取的功能，在實作中我們有做出基本架構，程式會根據 擷取建立標註日期與時間的存檔和文字紀錄，使事後人工操作 的時間能夠大幅縮短。



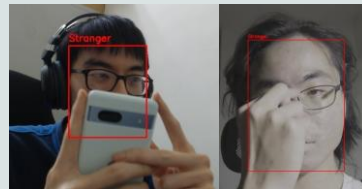
^ #在此感謝微處理機實驗課的同學
願意借出他的尊容當作測試資料集

7

結果與討論 RESULTS AND CONCLUSIONS

討論

在我們實作與測試的過程中，有發現我們的模型會將部分熟悉人物誤判成陌生資料集，這主要會發生在輸入資料品質不佳，尤其 模糊不清、受到遮擋或人物面向鏡頭另一邊，這都會導致模型判斷出錯。



當輸入品質不佳時，所有熟悉資料集
都有可能被模型誤判。

8

結果與討論 RESULTS AND CONCLUSIONS

討論

我們有嘗試引用模型輸出的品質參數 (det_score) 來忽略品質過低的臉部畫面，但我們發現被遮擋的品質參數 (約0.7) 其實比45°側臉 (約0.6) 還高，這就導致忽略遮擋的臉部畫面也同時忽略了側臉的輸入，但側臉輸入的偵測本來其實是能夠正常運作的。

因此最後我們選擇讓模型擷取畫面的同時也會記錄品質參數 (det_score) 與相似度 (similarity)，讓事後人工或機器搜尋能夠根據這兩個數值優先查找可疑對象。

9

未來展望 FUTURE PROSPECTS

1. 擴增熟悉資料集的人物數量，嘗試將模型應用在更大範圍的領域。
2. 增加將陌生資料集根據特徵向量相似度進行分類及建檔的功能，使人工操作時能夠更直接了解同一個陌生資料集的行為。
3. 可以嘗試導入影像前處理，提升模型在低光或者臉部傾斜的穩定性。

10

參考資料 REFERENCES

1. InsightFace official GitHub
<https://github.com/deepinsight/insightface>
2. ONNXRuntime
<https://onnxruntime.ai/>

11

團隊分工 DIVISION OF LABOR

杜政陽	<ol style="list-style-type: none">1. 撰寫主要程式2. 製作期中與期末報告 (協助增加技術資料)3. 負責期中與期末口頭報告	工作佔比: 55%
楊承燁	<ol style="list-style-type: none">1. 製作及整理影像資料集3. 協助拍攝測試集資料3. 製作期中與期末報告 (主要概念及架構統整)	工作佔比: 45%

12

THE END

謝謝大家

13

專題 2：基於 LINE 的多功能聊天機器人

基於 LINE 的多功能 聊天機器人

組員: 鍾瑋哲

沈雍哲

邱顯丞

指導教授: 國立東華大學電機工程學系 陳震宇教授

Outline

- 前言
- 系統架構概覽
- 技術重點
- 成果展示
- 結論與未來展望



前言

前言

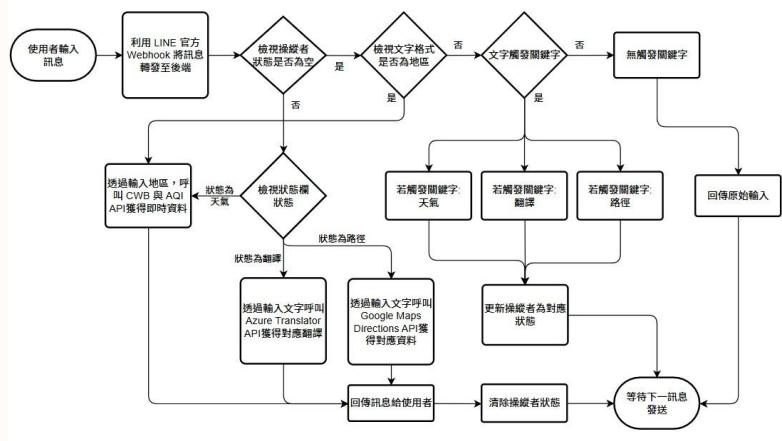
現代使用者常常需要同時查詢翻譯、天氣和路線，卻得在 Google 翻譯、氣象 App、導航 App 等多個程式之間切換，浪費時間又容易中斷操作流程。尤其在手機螢幕有限的情況下，不斷滑動、切換畫面，不僅體驗不佳，也容易錯過重要訊息，例如出門前的天氣預報或通勤路況。

本專題利用 LINE Bot 的主動推播機制，將翻譯、天氣預報和路線規劃整合到同一聊天介面：使用者只要在 LINE 中輸入關鍵字，就能在同一個 App 接收所有回覆；同時，透過定時或特定時機主動推送，讓使用者即使沒主動查詢，也能及時收到今日氣象、空氣品質或主要路段交通狀況，真正做到一站式資訊整合，提升使用者便利性與體驗。



系統架構概覽

系統架構概覽



技術重點

技術重點

一、天氣模組

- 1.查詢即時天氣 (中央氣象局 CWB)
 - 呼叫兩個即時氣象站 API (O-A0001 -001 、 O-A0003 -001)
 - 取得天氣描述 (晴/雨/多雲) 、溫度、濕度
 - 用「縣市 + 鄉鎮」做索引整理結果
- 2.查詢空氣品質 AQI (環保署資料)
 - 呼叫 AQX API · 取得全台 AQI 數據
- 3.組合成完整的文字訊息回傳

技術重點

二、翻譯模組

- 1.使用者輸入文字 (中或英文)
- 2.判斷是否包含中文：
 - 是 → 翻成英文
 - 否 → 翻成繁體中文
- 3.呼叫 Azure 翻譯服務並取得翻譯結果
- 4.回傳翻譯文字給 LINE 使用者

技術重點

三、路徑模組

1. 使用者輸入格式為：「起點 到 目的地」
2. 呼叫函式，向 Google Maps Directions API 發送 GET 請求，預設以 transit（大眾運輸）方式查詢
3. 回傳的 JSON 中包含所有步驟（steps），
每一步驟會標記 travel_mode（如 WALKING、BUS、RAIL 等）
4. 函式依據交通方式分類統計每段的時間，計算出：
total_duration_minutes：總時間
details：各交通方式與所需時間的對應表
5. 程式整合這些資訊組成文字回覆



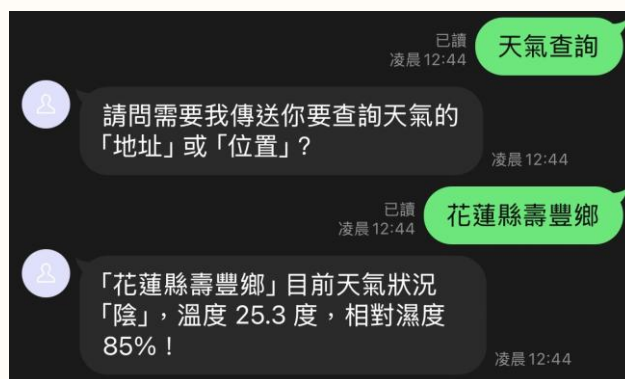
成果展示



成果展示



成果展示



成果展示



結論與未來展望

結論與未來展望

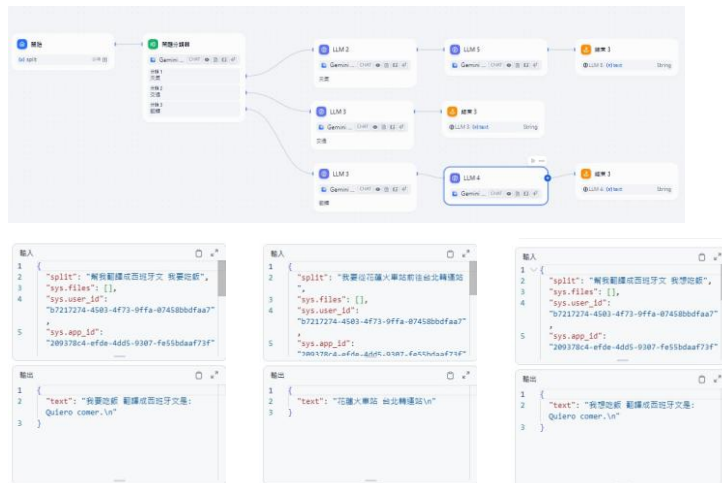
本專題成功整合多項外部 API，透過 LINE Messaging API 提供即時互動式服務。三大核心模組——天氣查詢、翻譯功能、路徑規劃——皆可根據使用者文字或位置輸入，提供有用、直觀的資訊回饋。整體系統具備：

- 清楚的使用流程與狀態機制（user_states）
- 高整合度的 API 串接能力（CWB, Azure, Google Maps）
- 穩定的對話式回應機制（使用 LINE Webhook 與 event handler）

此系統提供了使用者即時查詢生活資訊的便利性，是實用性與技術整合性的良好範例。

由於時間不足，這次系統功能較為基本。未來，希望能朝向與 Dify 工作流（Workflow）平台整合的方向發展，透過大語言模型更精確地理解與分類使用者需求，再使用 Python 調用各類 API，實現模組化、自動化的互動流程，進一步提升系統的智慧性與擴展性。未來可望加入如股票查詢、行事曆提醒、即時新聞推播等應用，打造更全面且智慧的對話式服務平台。

未來展望



坐姿檢測

第四組

組員：

411123052 羅雯慈

411123059 劉菁容

1

坐姿檢測

第四組

組員：

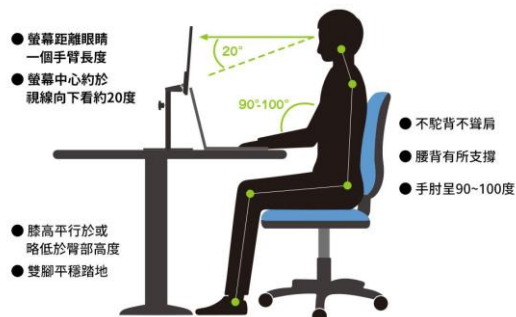
411123052 羅雯慈

411123059 劉菁容

1

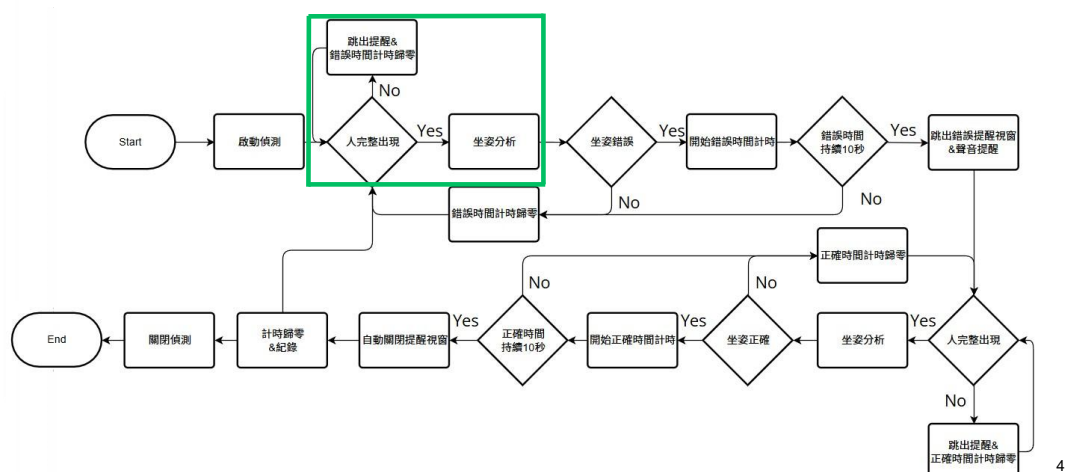
01 研究目的

- 為了改善現代人的姿勢不良帶來的健康問題，且根據近年來的研究顯示，姿勢不良會影響脊椎、肌肉退化、血液循環不良、大腦思考能力退化等問題產生，甚至導致死亡率增加的可能性。
- 本研究主要針對坐姿問題進行改善，藉由提醒人們坐姿是否正確以達到養成良好坐姿的目的。



3

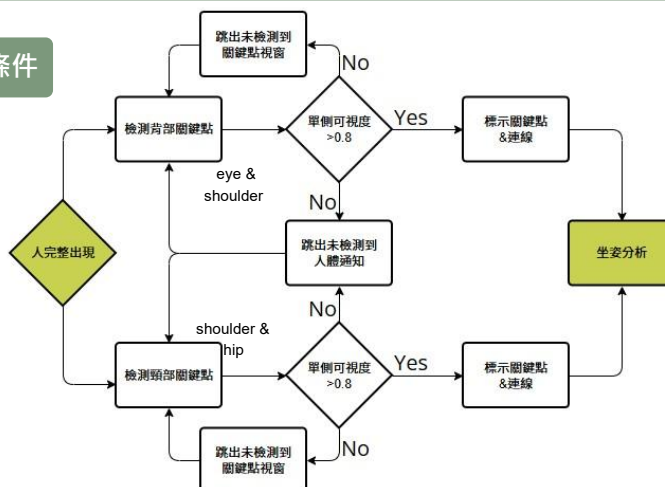
02 研究方法



4

02 研究方法

完整出現判定條件



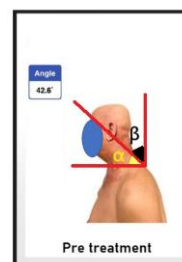
5

02 研究方法

姿勢錯誤判定條件

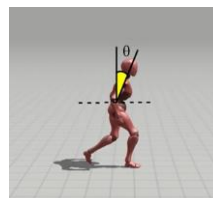
• 頸部：大於40度

當顱頸角 α (CVA)小於50~53度時，會被認定有頭頸前傾的問題。而我們測量的是 β 角，因此取40度作為基準。



• 背部：大於10度

雖然研究證實駝背的背部角度應為20度左右，但在測試中途發現20度不準確，故參考其他專題測量駝背角度的10度作為基準。



6

02 研究方法

坐姿錯誤提醒視窗

- 出現條件：姿勢錯誤時間累計10秒以上 + 畫面上無該部位提醒視窗
- 關閉條件：姿勢正確時間累計10秒以上 + 畫面上有該部位提醒視窗

坐姿正確計時提醒視窗，使其更加顯眼且增加急迫感

- 出現條件：畫面上有錯誤提醒視窗 + 坐姿正確
- 關閉條件：畫面上有錯誤提醒視窗 + 姿勢正確時間累計10秒
- 透過倒數計時機制，引導使用者在坐姿錯誤時主動調整，進而養成正確坐姿的習慣

7

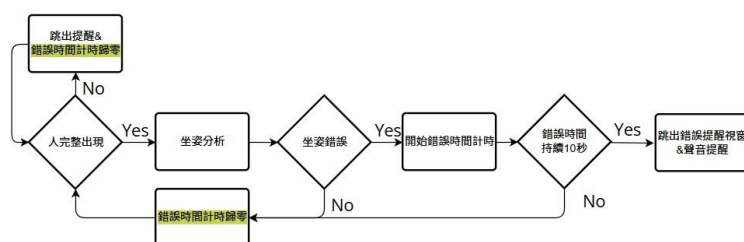
02 研究方法

錯誤時間紀錄

結束錯誤時間計時條件：

- 人無完整出現
- 坐姿正確
- 按下停止檢測

- 一次完整錯誤紀錄：
錯誤時間持續10秒~維持正確坐姿10秒
秒 → 矯正好姿勢才算一次



錯誤提醒視窗顯示時間



8

02 研究方法

資料統整

- 將間隔不到10秒的資料整合成一個數



修正錯誤時間紀錄

開始時間	結束時間	持續時間(秒)	最大角度	平均角度
2025-06-01 0:59:45	2025-06-01 0:59:58	12.57	52.9	51
2025-06-01 3:52:22	2025-06-01 3:52:35	12.99	61.9	54.4
2025-06-01 3:53:09	2025-06-01 3:53:21	11.66	59.2	55.6
2025-06-01 4:16:21	2025-06-01 4:16:32	11.34	75.5	59.9
2025-06-01 4:16:48	2025-06-01 4:16:59	11.68	64.2	59.2
2025-06-01 4:17:05	2025-06-01 4:17:22	16.84	64	60.4

開始時間	結束時間	持續時間(秒)	最大角度	平均角度
2025-06-01 0:59:45	2025-06-01 0:59:58	12.57	52.9	51
2025-06-01 3:52:22	2025-06-01 3:52:35	12.99	61.9	54.4
2025-06-01 3:53:09	2025-06-01 3:53:21	11.66	59.2	55.6
2025-06-01 4:16:21	2025-06-01 4:16:32	11.34	75.5	59.9
2025-06-01 4:16:48	2025-06-01 4:17:22	28.52	64.2	59.8

- 時間：取最前及最後
- 持續時間：相加
- 最大角度：取最大角度
- 平均角度：相加取平均

9

02 研究方法

資料儲存

- 錯誤時間超過十秒便會開始記錄
- 將資料分成頸部及背部兩個檔案儲存
- 資料名稱使用當前日期紀錄 Ex.{neck}_posture_{20250525}
- 同一天的檢測數據會儲存至同一檔案
- 紀錄內容：開始時間 / 結束時間 / 持續時間(s) / 最大角度 / 平均角度

度

開始時間	結束時間	持續時間(秒)	最大角度	平均角度
2025-05-30 0:19:39	2025-05-30 0:19:51	12.09	34.6	28.8
2025-05-30 0:19:51	2025-05-30 0:20:02	10.29	24.8	22.9
2025-05-30 0:20:16	2025-05-30 0:20:23	6.38	25.5	22.4
2025-05-30 0:21:43	2025-05-30 0:22:00	17.32	52.7	32.2
2025-05-30 0:22:39	2025-05-30 0:22:47	7.73	36.8	25.8
2025-05-30 0:23:27	2025-05-30 0:23:44	17.47	41.3	26
2025-05-30 0:24:54	2025-05-30 0:24:59	5.11	25.6	23
2025-05-30 0:25:31	2025-05-30 0:25:54	22.94	65.4	44.7

10

02 研究方法

視窗

利用 Tkinter 設計

主視窗內容，如右圖：

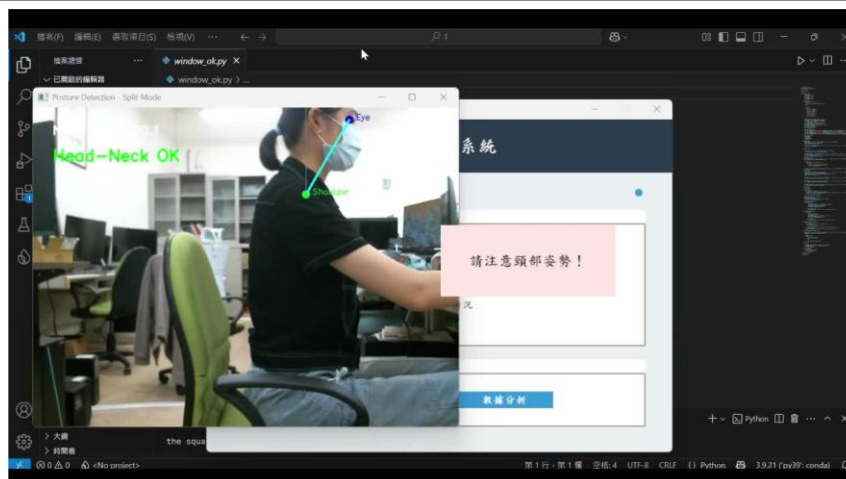
- 系統狀態
→ 顯示目前使用者的操作情形
- 說明
→ 簡略說明系統功能
- 操控按鍵
→ 基本按鍵
(開始/結束/分析)



11

03 結果與討論

影片



13

03 結果與討論



困難:

- 利用耳朵與肩膀來計算頸部前傾的角度，角度不夠精確
——→ 改抓取眼睛與肩膀來計算角度，有較好的改善
- 起初設計視窗較為樸素，不知如何美化
——→ 利用 AI 工具美化視窗
- 攝影機架設位置原本為夾在桌上，但距離無法拉遠
——→ 改用地上型攝影機支架

14

04 未來展望



1. 增加正面判別 Ex. 與眼睛與螢幕距離、高低肩
2. 建立資料庫：可多人紀錄及儲存長期檔案
3. 介面設計優化
4. 偵測更多特徵點 Ex. 膝蓋 → 檢測翹二郎腿
5. 區分成人及兒童檢測
6. 個人化訂定檢測數據閾值
7. 增加久坐判斷



05 參考資料

資料來源:

- [正確坐姿示意](#)

參考論文:

- [基於MediaPipe 的人體姿勢差異辨識系統](#)
- [以壓力感測為基礎之坐姿辨識系統](#)
- [A Comparison of Two Forward Head Posture Corrective Approaches in Elderly with Chronic Non -Specific Neck Pain: A Randomized Controlled Study](#)

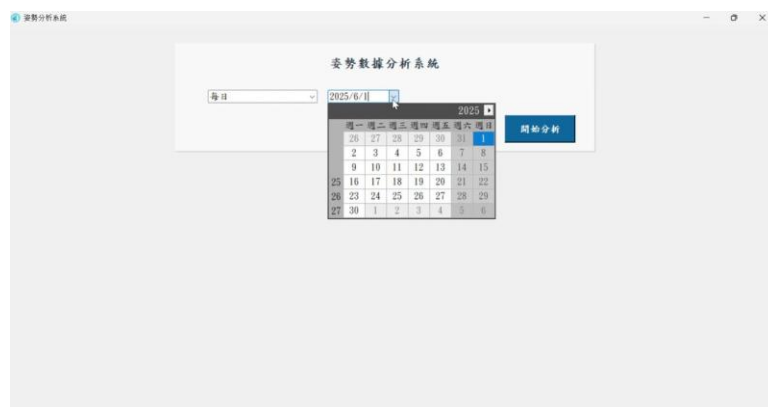
16

02 研究方法

視窗

分析視窗內容:

- 時段: 日/週/月/年
- 圖表: 每日事件數量/
角度分布/時段
分布



12

06 分工



羅雯慈	1. 撰寫程式碼(mediapipe 程式設計、即時提醒設計) 2. 製作簡報(期中、期末)	工作佔比：50 %
劉菁蓉	1. 撰寫程式碼(主視窗、分析視窗設計) 2. 製作簡報(期中、期末) 3. 錄製成果影片	工作佔比：50 %

17

The End

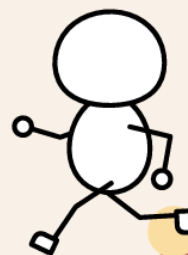
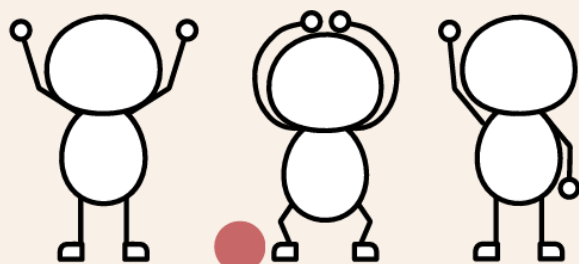


18

智慧計算實驗

姿勢偵測計數器

張芷瑜 林依函



目錄

- 研究目的
- 研究方法
- 結果與討論
- 未來展望
- 參考資料
- 團隊分工



研究目的

- 增加實用性用以協助老師教學
- 自檢運動成果給予情緒價值

P1

研究方法

- 使用PYQT做簡單的模式選擇、目標次數輸入、已完成次數的進度條、達成目標後的GIF
- 使用Mediapipe辨識動作

P2

結果與討論

功能：

1.三種模式

- 開合跳
- 舉手
- 伏地挺身

2.目標次數設定

3.進度條

4.完成動畫

P3

結果與討論

模式一：開合跳



P4

結果與討論

模式二：舉手



P5

結果與討論

模式三：伏地挺身



P6

結果與討論

在上述三個成果影片中，以PYQT和Mediapipe 可以達成大部分研究結果，能做到我們想要的運動計數並且給予一定程度上的互動。

P7

未來展望

功能：

- 增加時間限制
- 達到目標後的鼓勵畫面增加音效
- 增加虛擬人物
- 增加填寫個人資料
- 增加錄影、存檔

P8

未來展望

補充：

- 調整偵測人體時，因不同角度，偵測不到、不夠精確的問題
- 增加動作細節設定：動作角度、使用肌群

P9

參考資料

STEAM學習網

<https://steam.oxxostudio.tw/>

Python PyQt5 學習筆記

<https://hackmd.io/@kaneyxx/HJdX8DXCr>

Qt Designer's Widget Editing Mode

https://ftp.nmr.mgh.harvard.edu/pub/dist/freesurfer/tutorial_versions/freesurfer/lib/qt/qt_doc/html/designer-widget-mode.html

Qt Designer's Widget Editing Mode

<https://neo.dmc.s.pl/po/qt-tutorial/docs/qt/designer-editing-mode.html>

P10

團隊分工

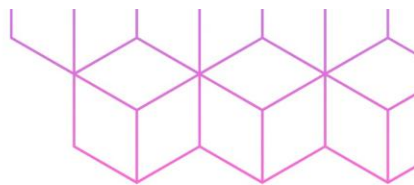
張芷瑜	<ol style="list-style-type: none"> 1.撰寫程式 2.開會並進行討論(參與3次) 3.製作期中報告(負責:目的、流程圖、研究目標、甘特圖) 4.製作期末報告(負責:研究目的、結果與討論影片錄製、demo影片上字幕) 5.參與期中口頭報告 6.參與期末口頭報告 	50%
林依函	<ol style="list-style-type: none"> 1.撰寫程式(和組員說明功能) 2.開會並進行討論(參與3次) 3.製作期中報告(負責:目錄、目的、相關研究、研究目標、材料清單) 4.製作期末報告(負責:研究方法、結果與討論、參考資料、未來展望、封面、目錄、錄製demo影片) 5.參與期中口頭報告 	50%

P11

THANK
YOU



llll



車距偵測以及警示標記

組員：

411223022謝家昇

411223028賴柏諭

411223030陳威縉

研究目的

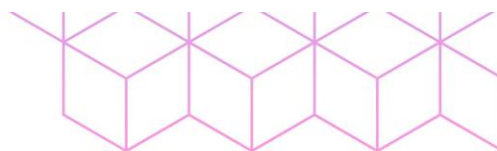
研究方法

結果與討論

未來展望

參考資料

團隊分工表



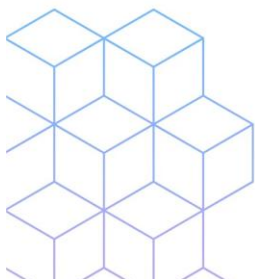
目錄

研究目的

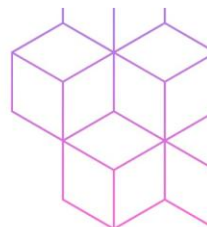
台灣每天都能看見車禍的新聞，有輕微的，也有嚴重的。我們就在想，若能設計一個系統，能去偵測周圍車輛的距離並提出警示的效果，是否能让駕駛人更容易察覺周遭的危險，並適時地做出相對應的反應(例如煞車、迴避等...)

這次研究設計是利用YOLOv12及MiDas模型去預測出影片中車輛間的距離及相對深度值為多少。

透過距離偵測，達到當駕駛的車輛和被偵測的車輛之間駕駛人的安全車距不足時，能夠發出提醒的警報聲

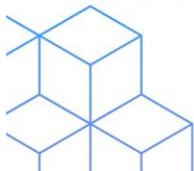


研究方法



Pixel估值推算

- GUI頁面設計
- 按鈕連結功能及計算結果導入
- 連結YOLO+MiDas



YOLO

- 車輛資料集建構
- YOLOv12模型導入及訓練
- 保取訓練最佳weights.pt

MiDas

- 導入MiDas_small.pt
- 計算深度值
- 標示深度值於車輛偵測框

研究方法

Pixel推算-GUI頁面設計

點選影片選擇(.mp4)

查詢所需之畫面所在的fps

在畫面中點擊兩點，得到距離對應的pixel

輸入兩點實際對應的線段距離(例如道路線為15cm)

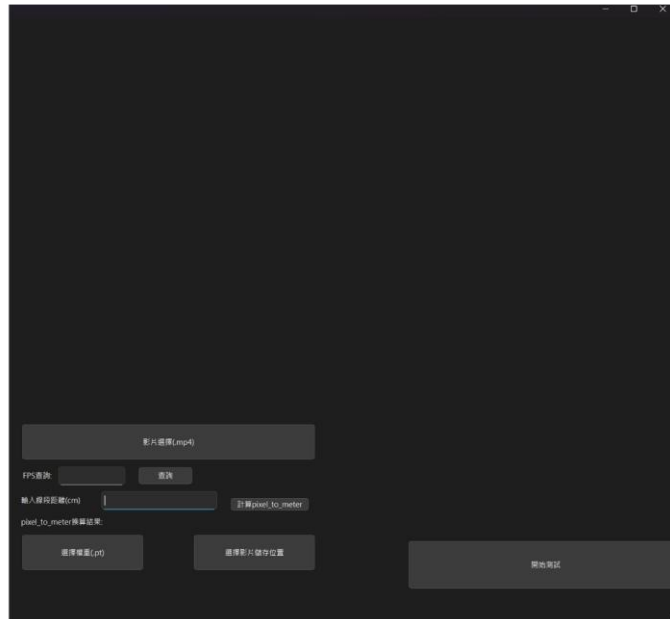
計算並獲取pixel對應的距離

選擇YOLO權重(.pt)

選擇偵測完後影片儲存位置

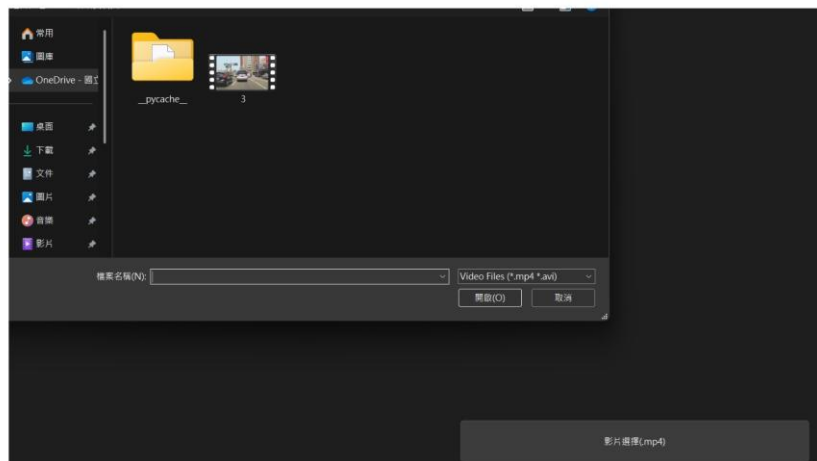
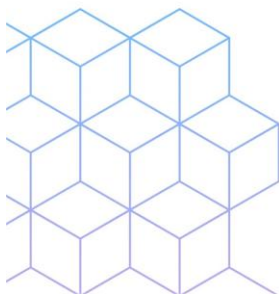
點擊開始測試

關閉GUI畫面並開始偵測



研究方法

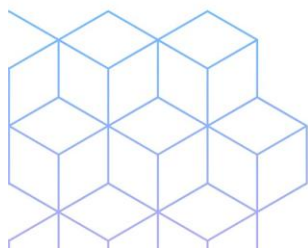
選擇要偵測的影片↓



研究方法

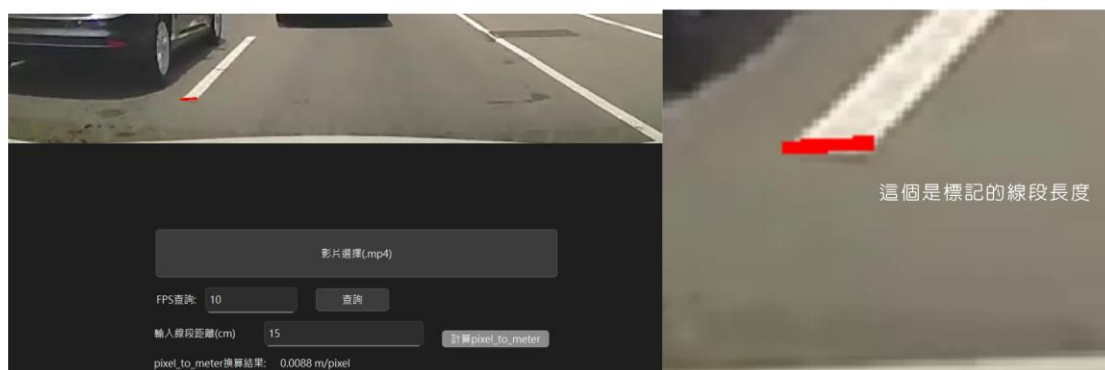
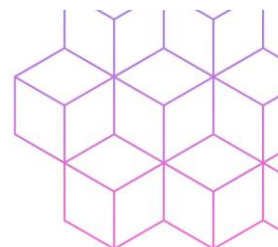
這裡會顯示查詢的fps畫面↓

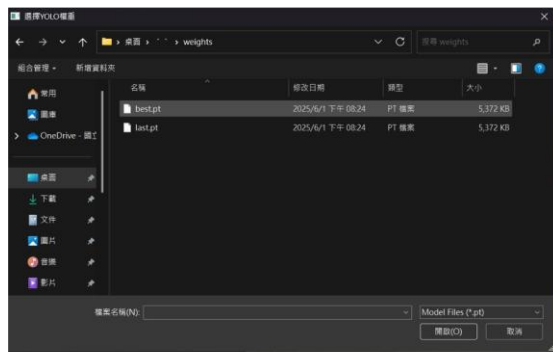
查詢含有對照物的畫面
(前方車型的寬度、地上的
標線等...)



研究方法

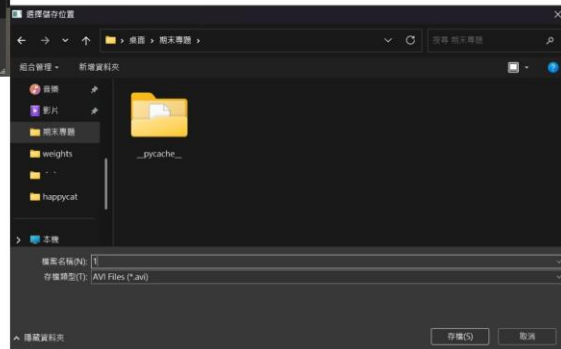
輸入線段距離及計算





← 選取權重

選取影片儲存位置→



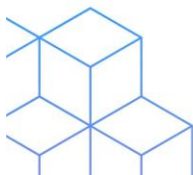
研究方法

警示觸發時間為10秒一次(預設)

設定ROI(限制對車輛的偵測、避免邊緣車輛被偵測)

影片、YOLO模型架構、權重載入

、Midas模型線上載入(需網路)



```
def start_prediction(self):
    #關閉GUI並開始Prediction
    self.close()
    import torch, cv2, numpy as np, time, winsound
    from ultralytics import YOLO
    import os
    os.environ["KMP_DUPLICATE_LIB_OK"] = "TRUE"

    #警示偵測時間
    last_alarm_time = 0
    alarm_interval = 10
    meter_per_pixel = self.meter_per_pixel

    #得到影片的解析度(x,y值) 和total_fps
    cap = cv2.VideoCapture(self.video_path)
    width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = cap.get(cv2.CAP_PROP_FPS)

    #影片輸出設置
    fourcc = cv2.VideoWriter_fourcc(*"XVID")
    out = cv2.VideoWriter(self.save_path, fourcc, fps, (width * 2, height))
    #ROI設定(避免偵測到邊緣的物體)
    x_min = width * 0.1
    x_max = width * 0.9
    #導入YOLO模型和MIDAS模型
    model = YOLO(self.model_path)

    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    midas = torch.hub.load("intel-lisl/MiDaS", "MiDaS_small").to(device).eval()
    transform = torch.hub.load("intel-lisl/MiDaS", "transforms").small_transform
    import torchvision.transforms as T

    transform = T.Compose([
        T.ToPILImage(),
        T.ToTensor(),
        T.Normalize(mean=[0.5], std=[0.5])
    ])

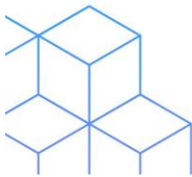
```

研究方法

深度值平均計算(取偵測框內的每點數值並求得平均)
並將標籤(深度值)顯示於偵測框上

YOLO標籤設定為預測距離
[y2(前車框最底y軸)-y1(整個畫面的最底y軸)]

設定警示條件(影片車速為25km/h)
(這邊預設距離>2m且深度值>300時觸發)



```
#開始讀取影片並對每一幀進行處理
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    img_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    input_batch = transform(img_rgb).to(device)

    #深度值計算
    with torch.no_grad():
        prediction = model(input_batch)
        depth = prediction.squeeze().cpu().numpy()
        depth_resized = cv2.resize(depth, (width, height))
        depth_colored = cv2.applyColorMap(
            cv2.normalize(depth_resized, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8),
            cv2.COLORMAP_JET)

    #YOLO的偵測結果顯示
    results = model.track(source=frame, persist=True, conf=0.7, verbose=False)
    boxes = results[0].boxes.xyxy if results[0].boxes is not None else []

    for box in boxes:
        x1, y1, x2, y2 = map(int, box[:4])
        cx = (x1 + x2) / 2
        if x_min <= cx <= x_max:
            pixel_dist = height - y2
            real_dist = pixel_dist * meter_per_pixel
            box_depth = depth_resized[y1:y2, x1:x2]
            mean_depth = np.mean(box_depth) if box_depth.size > 0 else 0
            color = (0, 255, 0)
            #這裡為警示設定，如果物體距離小於2公尺且平均深度大於300，則發出警示並將車框顏色變為紅色
            if real_dist < 2 and mean_depth > 300:
                color = (0, 0, 255)
                now = time.time()
                if now - last_alarm_time > alarm_interval:
                    winsound.Beep(1000, 500)
                    last_alarm_time = now
            cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
            cv2.putText(frame, f"{real_dist:.1f} m", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)
            cv2.rectangle(depth_colored, (x1, y1), (x2, y2), color, 2)
            cv2.putText(depth_colored, f"{mean_depth:.1f}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)

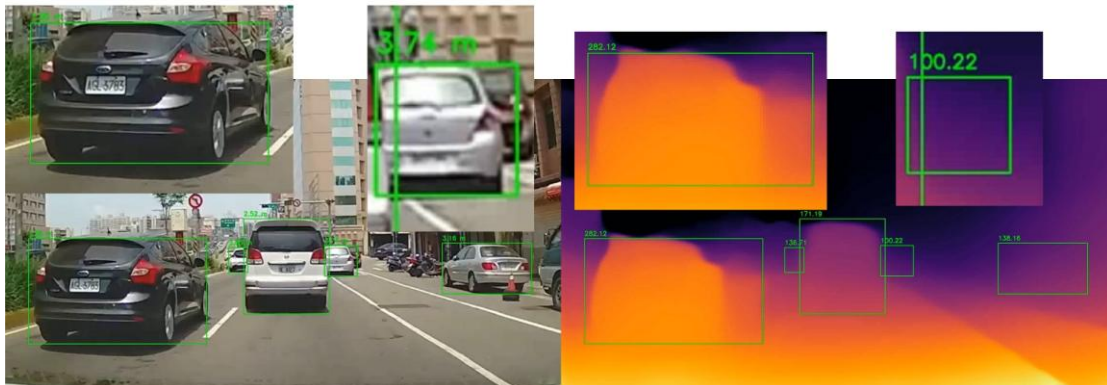
    #將YOLO與Midass的結果合併顯示
    combined = np.hstack((frame, depth_colored))
    out.write(combined)
    cv2.imshow("YOLO + Midas 預測", combined)
    #按下鍵盤Esc，中斷影片播放
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
out.release()
cv2.destroyAllWindows()
```

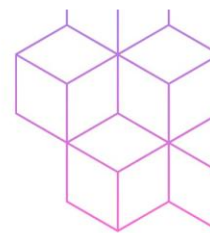
結果與討論

這是輸出影片畫面的其中一部份截圖，可以看到近距離時車輛距離偵測是還算不錯的(例如左邊黑色的車子)。但當車輛走遠時，距離計算的部分就會出現問題(例如右邊那輛銀色車子)。這是因為距離計算的公式所導致的問題($y2-y1$)，這樣會導致車輛走遠了，偵測框雖然變小，但 $y2$ 並沒有明顯的變化，導致距離計算與實際距離出現越來越大的偏差。

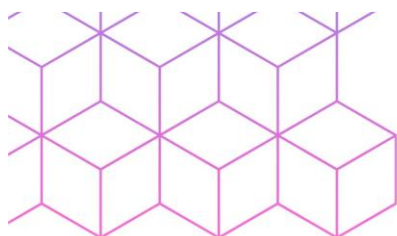
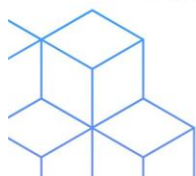
所以搭配了另一個模型MiDas，生成深度圖(右下)。並對其框框內的顏色數值取得平均顯示在框框上，可以看到黑車的深度圖還算明顯(距離近)，而銀色車相對黑車已經沒有明顯的顯示了(距離遠)



未來展望



- 搭配GPS獲得車速，並根據車速修改警示數值
- 加強YOLO模型的訓練(.pt)，讓偵測結果更加正確
- 加入偵測的挑戰(黑夜、雨天等..)，讓結果不易失真
- 加強運算速度，並實際應用到硬體上(現階段還只能使用影片)
- 加上雷達偵測，讓距離更加精準
- 自動獲取相機內部數值(焦距、畫質等...)，替換掉pixel距離估計法

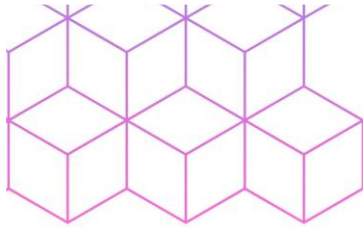


ModelHub-MiDaS - PyTorch
https://pytorch.org/hub/intelisl_midas_v2/

取得影像資訊。STEAM教育學習網
<https://reurl.cc/M35R3v>

測試影片:S550 行車記錄器 實拍影片。Youtube
<https://reurl.cc/o820mD>

參考文獻



分工表

411223022 謝家昇 工作佔比:35%
設計題目實作方法
主要程式碼編寫(GUI設計、模型預測、影片輸出)
期中期末報告製作(研究方法、結果與討論)
測試程式碼運行結果

411223028 賴柏諭 工作佔比:35%
擷取YOLO訓練所需的車輛資料
標記YOLO資料集的標籤
訓練模型獲取weight.pt檔
程式碼編寫(YOLOv12訓練)
期中期末報告製作(研究目的、未來展望)

411223030 廖威權 工作佔比:30%
期中報告製作(提供想法和最終要呈現的結果)
期中期末報告製作(未來展望、提供參考資料)
期中期末口頭報告
查找MIDas及cv2應用方式

專題 6：食物便是卡路里

智慧計算實驗期末專題報告
第八組

食物辨識卡路里
影像辨識/深度學習/網路爬蟲/卡路里估計

1

小組成員



411223077 陳禹丞

411223082 劉震緯

411223090 張新振

2

目錄

- | | |
|---------|--------|
| ① 前言 | ⑤ 未來展望 |
| ② 研究目的 | ⑥ 團隊分工 |
| ③ 研究方法 | ⑦ 參考資料 |
| ④ 結果與討論 | |

3

① 前言

4

前言

科技日新月異，現代生活的節奏也日益加快，使人們常常忽略了每日的飲食與營養攝取。攝取過量的熱量、納、糖分.....也導致患有肥胖、糖尿病、高血壓等現代「文明病」的患者層出不窮。

雖然影像辨識技術已有顯著進展，但市面上的相關 APP 大部分仍須使用者手動輸入食物名稱以查詢資料，不僅使用門檻偏高，且對於特殊族群也不夠直覺與友善。

為解決上述的問題，我們開發一套上傳拍照圖片即可辨識食物給予營養資訊並提供健康提醒的小系統，使得使用者可以輕鬆掌握自身的飲食情況，為未來的個人化健康建議奠下基礎。

5

2 研究目的

6

研究目的

做出一套結合影像辨識與網路爬蟲的食物辨識系統，透過上傳食物圖片，即可從公開來源獲得預估的熱量、納含量與糖分等資訊，並根據使用者的身份模式提供客製化的警語！

提高便利性，降低使用者進行飲食紀錄的門檻，隨時掌握自己的飲食狀況!!!



7

3 研究方法

8

研究方法

影像資料集

- 本研究所使用的食物資料集共包含 17 個類別，其中「白飯」類別取自開放資料集 UECFOOD100 [1]，其餘則由組員從 Google 手動蒐集。每一個類別包含 500 張影像。
- 本研究所使用的硬幣資料集，由組員實際拍攝，共有 100 張影像。

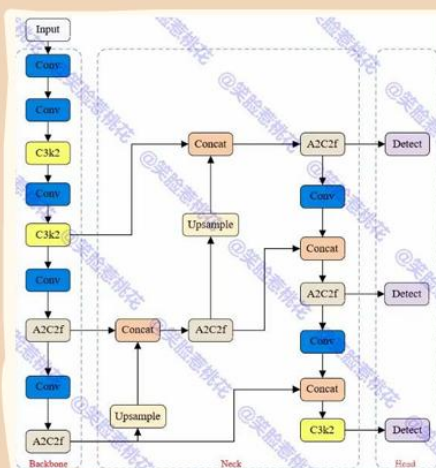
資料集類別

- 白飯, 荷包蛋, 炒高麗菜, 菠菜, 洋葱炒蛋, 麻婆豆腐, 炸蝦, 番茄炒蛋, 滷雞腿, 蒸蛋, 筍絲, 鯛魚, 豆干, 茄子, 冬粉, 香腸, 酸辣湯

[1] Kawano, Y., & Yanai, K. (2014). Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. European Conference on Computer Vision (ECCV), 3-17.

9

研究方法



選用模型：YOLOv12

選用原因：

最新、YOLOv12 提升邊緣運算能力

模型架構：

- Backbone → 特徵提取層
- Neck → 特徵融合層
- Head → 偵測輸出

11

研究方法—超參數定義

食物辨識

```
from ultralytics import YOLO

model = YOLO('yolov12s.pt')

model.train(
    data='C:/Users/USER/Desktop/experiment/chen/all_final/dataset.yaml',
    epochs=100,
    imgsz=640,
    project='C:/Users/USER/Desktop/experiment/chen/all_final/YOLO_run_data',
    name='first_test',
    verbose=True,
    workers=0
)
```

硬幣-參考物

```
from ultralytics import YOLO

model = YOLO('yolov12s.pt')

model.train(
    data='C:/Users/USER/Desktop/experiment/chen/tenset.yaml',
    epochs=50,
    imgsz=640,
    project='C:/Users/USER/Desktop/experiment/chen/all_final/coin_run_data',
    name='first_test',
    verbose=True,
    workers=0
)
```

訓練設定

1. epoch → 訓練次數
2. imgsz → 圖片大小
3. verbose → 顯示詳細的訓練資訊
4. workers = 0 → 不開啟多執行緒，單線程載入資料 (穩定跑完，不會中途崩潰!!!)

評估指標

1. precision (box)
2. recall
3. mAP50
4. mAP50-95

12

研究方法

體積估算

- 利用影像辨識模型偵測出影像中硬幣與食物的邊界框，並將硬幣邊界框的長度、寬度作為參考基準。
- 根據硬幣的實際直徑（本研究採用新台幣 10 元硬幣，直徑為 2.6 公分），可推算影像中每個像素所代表的實際距離，進而換算出食物邊界框的實際長寬，估算其所佔的面積（ cm^2 ）。
- 本研究假設各個類別食物有固定的平均深度，在推算出面積後，將其乘以對應類別的深度值，即可得到近似的體積（ cm^3 ）。

重量估算

- 本研究使用密度公式對食物體積進行重量的換算，我們將估算出來的食物體積乘上對應的食物密度，即得到食物的重量。

公式如下：

$$\text{Weight} = \text{Volume} * \text{Density}$$

13

研究方法

網路爬蟲

- 將模型預測的食物標籤對應至自建的對照表，取得對應的食物 ID
- 透過固定格式的網址拼接該 ID，進行網路爬蟲擷取營養資訊
- 擷取頁面中的熱量、鈉、鉀、磷等數值，解析後呈現在圖形化介面（GUI）中，提供使用者查閱。
- 營養資料來源為衛生福利部食品藥物管理署「食品營養成分資料庫（新版）」

14

研究方法

警語設計

- 本系統針對三類使用者族群設計個別化健康警語，分別為：一般使用者、高血壓患者、腎臟病患者
- 系統會依據每日建議攝取量，針對下列四項營養指標進行分析：熱量 (Calories)、鈉總含量 (Sodium)、磷總含量 (Phosphorus)、鉀總含量 (Potassium)
- 當營養素之預估攝取量高於或低於該使用者對應的每日建議攝取量時，系統即顯示相應之健康警語，提醒使用者注意飲食攝取狀況。
- 各項建議攝取量參考衛福部公布之《國人膳食營養素參考攝取量第八版 (DRIs)》，及相關疾病指引。

15

4 結果與討論

16

結果與討論—驗證指標

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95):
all	1699	1942	0.928	0.937	0.954	0.849
sausage	100	207	0.624	0.585	0.648	0.452
hot and sour soup	100	102	0.96	0.99	0.994	0.984
cabbage	100	105	0.984	0.943	0.983	0.958
spinach	100	100	0.994	1	0.995	0.988
fried eggs with onions	100	100	0.965	1	0.994	0.974
Mapo Tofu	100	100	0.961	0.994	0.985	0.971
fried shrimp	100	106	0.968	0.991	0.993	0.921
tomato scrambled eggs	100	103	1	0.996	0.995	0.989
braised chicken legs	100	103	0.946	0.971	0.989	0.9
steamed egg	100	109	0.982	0.985	0.995	0.972
shredded bamboo shoots	100	101	0.992	1	0.995	0.97
snapper	100	103	0.933	0.932	0.974	0.903
bean curd	100	197	0.774	0.731	0.81	0.527
eggplant	100	104	0.957	0.962	0.958	0.72
winter noodles	100	103	0.922	0.923	0.968	0.693
rice	99	99	0.871	0.955	0.958	0.7
poached eggs	100	100	0.947	0.97	0.982	0.81
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95):
all	20	20	0.997	1	0.995	0.95

17

結果與討論—驗證指標



18

結果與討論—驗證指標



19

結果與討論—GUI 功能

請選擇使用者身份

請勾選您的身份：

☒ 一般使用者

☐ 高血壓患者

☐ 腎臟病患者

確定

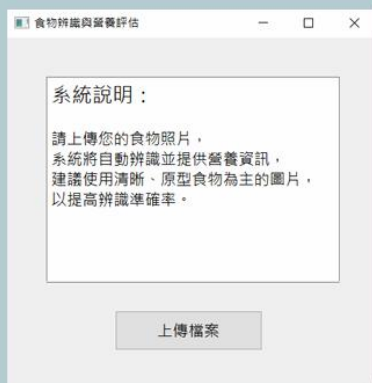
使用者身份選擇：

系統啟動後，首先會開啟使用者身份的選擇介面。預設身份為「一般使用者」，使用者亦可選擇「高血壓患者」或「腎臟病患者」兩類特殊族群。身份選擇完成後，點選「確定」按鈕即可進入下一階段操作。

本系統將根據不同身份所對應的建議攝取量與限制條件，給予個人化的營養提醒與警語。

20

結果與討論—GUI 功能



影像上傳與系統說明：

進入第二個介面後，使用者可上傳欲進行分析的食物影像。系統支援的檔案格式包括：.jpg、.png、.jpeg。

此介面也提供簡要的操作說明，協助使用者順利完成上傳流程。當使用者選擇影像並上傳後，系統將自動啟動食物辨識模型，並透過網路爬蟲，擷取營養資訊，並進行分析。

21

結果與討論—GUI 功能



分析結果呈現與警語提示：

分析完成後，系統將進入第三個視窗，即結果呈現介面。本介面分為左右兩區：

- 左側區塊：呈現食物之營養資訊，包括每個辨識出食物的個別營養成分，以及整體餐點的總營養統計（如總熱量、總鈉含量等）。根據使用者身份（一般、高血壓、腎臟病）比對對應的建議攝取量，提供專屬的營養攝取警語。

- 右側區塊：展示食物辨識的結果影像，含有 YOLO 模型所標註的邊框，提供使用者辨識準確性的參考。

22

結果與討論—時間測試

次數	套件載入	模型導入	模型推論	爬蟲與資料處理	整體處理
1	1.076	0.062	2.535	1.085	7.218
2	0.827	0.054	2.38	1.056	6.58
3	0.795	0.059	2.443	1.07	6.423
4	0.803	0.059	2.375	1.07	5.611
5	1.077	0.059	2.459	1.042	6.601
6	0.805	0.055	2.481	1.041	5.812
7	1.013	0.056	2.43	1.028	5.744
8	0.798	0.06	2.429	1.079	5.985
9	1.022	0.056	2.559	0.978	5.787
10	1.072	0.057	2.427	1.043	5.902
average	0.9288	0.0577	2.4518	1.0492	6.1663

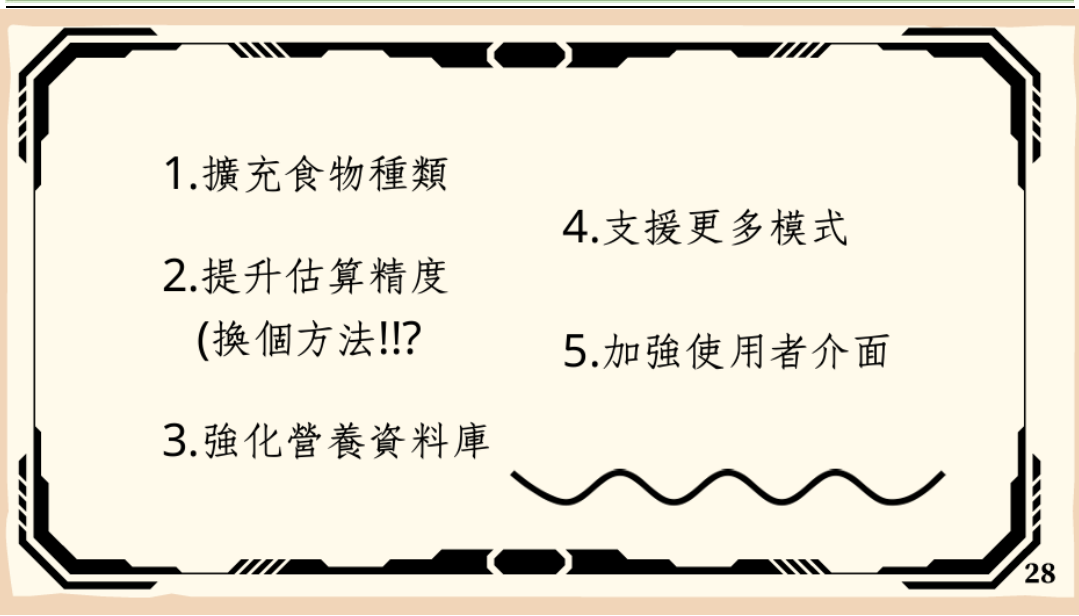
單位：秒

23



⑤ 未來展望

27

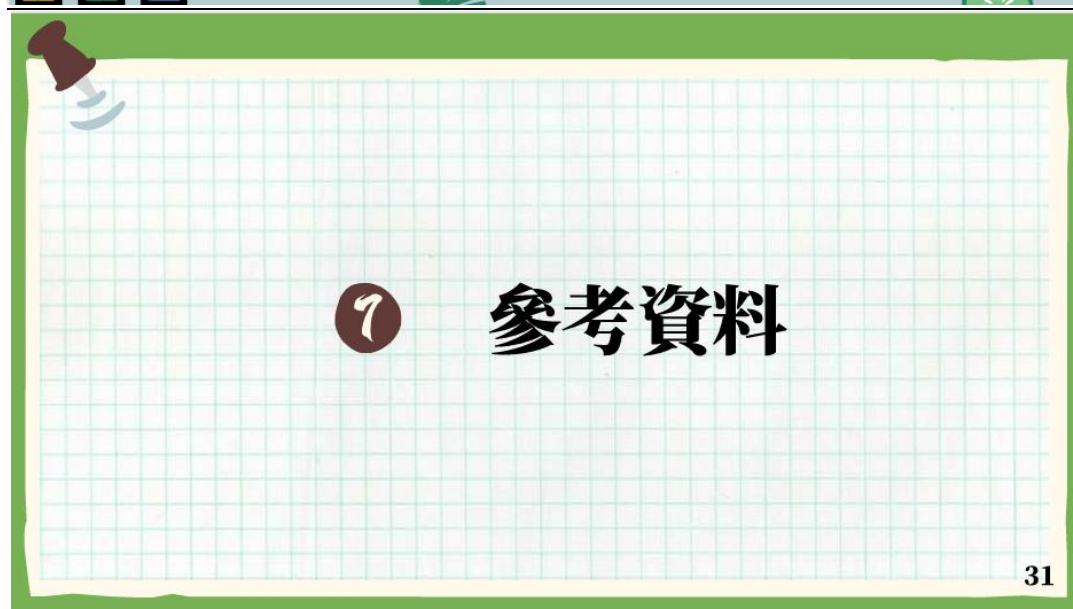
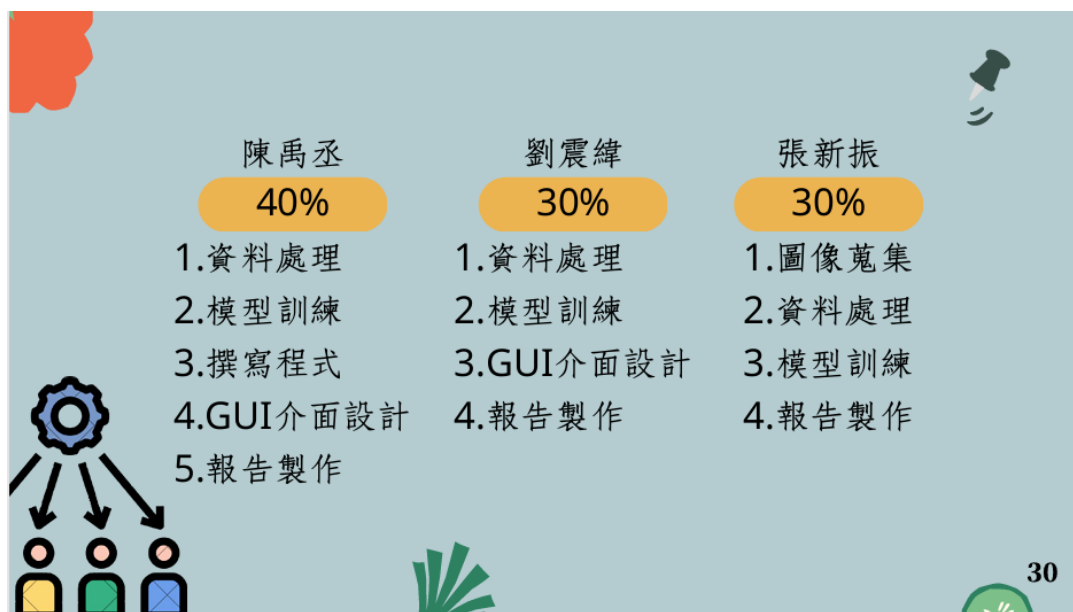
- 
1. 擴充食物種類
 2. 提升估算精度
(換個方法!!?)
 3. 強化營養資料庫
 4. 支援更多模式
 5. 加強使用者介面

28



⑥ 團隊分工

29



參考資料

- [1] Kawano, Y., & Yanai, K. (2014). Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. European Conference on Computer Vision (ECCV), 3-17.
- [2] Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. International Journal of Computer Vision, 77(1-3), 157-173.
- [3] 劉士賢. (2014). 應用影像處理技術於食物攝取辨識之研究 [碩士論文, 高雄應用科技大學資訊工程系]. 全國碩博士論文資訊網。
<https://ndltd.ncl.edu.tw/cgi-bin/gs32/gsweb.cgi/ccd=KVyie6/search?s=id=%22102KUAS0392008%22.&openfull=1&setcurrent=0#XXX>
- [4] 深度學習的滴滴. (2023年8月23日). YOLOv8 教學：從訓練到推理全流程實作 [影片]. 哩哩哩 Bilibili.
<https://www.bilibili.com/opus/1046383324723937287>
- [5] 衛生福利部食品藥物管理署. (2023). 食品營養成分資料庫 (新版).
<https://consumer.fda.gov.tw/FoodNutrition/Index.aspx>
- [6] 衛生福利部國民健康署. (2020). 國人膳食營養素參考攝取量第八版 (DRIs).
<https://www.hpa.gov.tw/Pages/Detail.aspx?nodeid=542&pid=13084>
- [7] Wang, C.-H. (2022). 基於深度學習的食物辨識和卡路里估計雲端系統 (碩士論文, 台灣科技大學). 國家圖書館臺灣碩博士論文系統。
<https://reurl.cc/paj0GZ>
- [8] 笑臉惹桃花. (n.d.). YOLOv5--YOLOv12 各版本網路結構圖. 哩哩哩 Bilibili.
<https://reurl.cc/5KyWjv>
- [9] 陳冠瑩. (2021). 結合深度學習與機器學習之多分類美食辨識研究 [碩士論文, 台北科技大學資訊工程系]. 全國碩博士論文資訊網。
<https://ndltd.ncl.edu.tw/cgi-bin/gs32/gsweb.cgi/ccd=GV2jcl/search?s=id=%22110TIT00146051%22.&openfull=1&setcurrent=0#XXX>

114年智慧計算實驗

智慧傘架：雨天提醒系統

網頁爬蟲 / 螢幕顯示 / 定時更新天氣狀況

國立東華大學 電機工程學系

411123021 鐘以涵

411123024 劉靜謙

Outline

- 研究目的
- 研究方法
- 結果與討論
- 未來展望

研究目的

- 研究動機

我們出門常常忘記帶傘，有時候突然下雨，就會造成不便，如果有這個裝置，就可以提醒我們當天的天氣狀況可能會下雨，這樣便可大大降低天氣不好卻又忘記帶傘的機率。

- 研究目的

開發一個以 Python 撰寫的「天氣預報提醒系統」，在CD螢幕上顯示當天降雨機率，讓使用者判斷今日是否需要攜帶雨具。

工具網頁介紹

爬取天氣預報 <https://opendata.cwa.gov.tw/index>

- 中央氣象署的「氣象資料開放平臺」是臺灣官方提供的氣象資料共享平台，旨在促進政府資料的開放與應用
- 提供資料格式多樣的天氣資訊檔案供研究使用，例如我們所用到的JSON檔



研究方法

天氣預報JSON檔

```
{
  "cwaopendata": {
    "id": "urn:cwa:gov:tw:cwacommon:6.1",
    "identifier": "2a091ae-619-509-800-ec5cf0a00a",
    "sender": "weather.gov.tw",
    "sent": "2025-05-26T16:57:03+08:00",
    "status": "Actual",
    "magtype": "Issue",
    "source": "WGC",
    "dataset": "CWA61-001",
    "scope": "Public",
    "dataset": {
      "datasetInfo": {
        "datasetDescription": "36-hour Weather Forecast",
        "issueTime": "2025-05-26T17:00:00+08:00",
        "update": "2025-05-26T16:57:03+08:00"
      },
      "location": {
        "locationName": "Huailien County",
        "weatherElement": [
          {
            "elementName": "Wx",
            "time": [
              {
                "startTime": "2025-05-26T18:00:00+08:00",
                "endTime": "2025-05-27T06:00:00+08:00",
                "parameter": {
                  "parameterName": "Cloudy",
                  "parameterValue": "7"
                }
              },
              {
                "startTime": "2025-05-27T06:00:00+08:00",
                "endTime": "2025-05-27T18:00:00+08:00",
                "parameter": {
                  "parameterName": "Pop",
                  "parameterUnit": "percent"
                }
              }
            ]
          }
        ]
      }
    }
  }
}
```

```
"elementName": "Pop",
"time": [
  {
    "startTime": "2025-05-26T18:00:00+08:00",
    "endTime": "2025-05-27T06:00:00+08:00",
    "parameter": {
      "parameterName": "20",
      "parameterUnit": "percent"
    }
  },
  {
    "startTime": "2025-05-27T06:00:00+08:00",
    "endTime": "2025-05-27T18:00:00+08:00",
    "parameter": {
      "parameterName": "30",
      "parameterUnit": "percent"
    }
  },
  {
    "startTime": "2025-05-27T18:00:00+08:00",
    "endTime": "2025-05-28T06:00:00+08:00",
    "parameter": {
      "parameterName": "30",
      "parameterUnit": "percent"
    }
  }
]
```

爬取全台未來36小時天氣預報內容並稍加修改，保留花蓮縣天氣資訊

3

Python程式碼

```
import json
import schedule
import time
import serial
from datetime import datetime

# Arduino 序列埠設定 (請根據實際 COM port 調整)
ser = serial.Serial('COM10', 9600) # ← 這裡要確認 Arduino 的序列埠是 COM3

def update_weather():
    try:
        with open('C:\\Users\\jq20\\Downloads\\w.json', 'r', encoding='utf-8') as f:
            data = json.load(f)

        locations = data['cwaopendata']['dataset']['location']

        for location in locations:
            city = location['locationName']
            weather_elements = {
                e['elementName']: e['time'][0]['parameter']['parameterName']
                for e in location['weatherElement']
            }

            wx = weather_elements.get('Wx', 'N/A')
            pop = weather_elements.get('Pop', 'N/A')
```

連接Arduino

讀取天氣資訊的json資料

天氣狀況
降雨機率

4

Python程式碼

```
line1 = f"{city[:16]}"
line2 = f"{wx[:10]} {pop}%"

# 傳送到 LCD
ser.write((line1 + '\n').encode('utf-8'))
time.sleep(0.1)
ser.write((line2 + '\n').encode('utf-8'))
print(f"[{datetime.now()}] Sent to LCD: {line1} | {line2}")
break

except Exception as e:
    print("更新天氣資料失敗:", e)

schedule.every().day.at("08:00").do(update_weather)

print("☀️ Weather updater started and waiting for 08:00...")

while True:
    schedule.run_pending()
    time.sleep(1)
```

設定時間：早上8點顯示

5

Arduino程式碼

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    Serial.begin(9600);
    lcd.init();
    lcd.backlight();
    lcd.clear();
    lcd.print("準備中...");
}

void loop() {
    if (Serial.available()) {
        String line1 = Serial.readStringUntil('\n');
        String line2 = Serial.readStringUntil('\n');

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(line1);
        lcd.setCursor(0, 1);
        lcd.print(line2);
    }
}
```

6

設定時間自動顯示

在命令提示字元輸入:

where python

找出系統中的 Python 執行路徑

```
C:\Users\ljq20\anaconda3\envs\py39\python.exe
C:\Users\ljq20\anaconda3\python.exe
C:\Users\ljq20\AppData\Local\Microsoft\WindowsApps\python.exe
```

Python 執行檔路徑

7

設定時間自動顯示

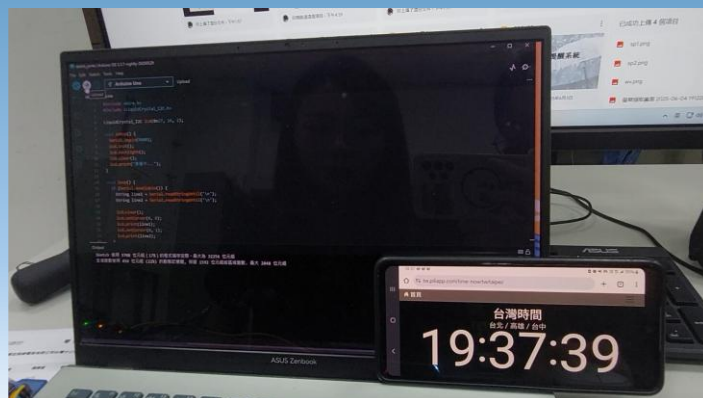
在電腦的工作排成器中建立工作，並在「程式或指令碼」欄位，貼上 Python 執行檔路徑、在「新增引數」欄位，輸入Python 腳本完整路徑



8

結果與討論

成果展示



討論

透過網頁爬蟲讀取氣象資訊，再透過python程式、電腦工作排成器，和Arduino將資訊連接到LCD螢幕上，將LCD屏幕連接到傘架，便可以在每日出門前提醒使用者今日天氣狀況，以判斷是否需要帶傘。

10

未來展望

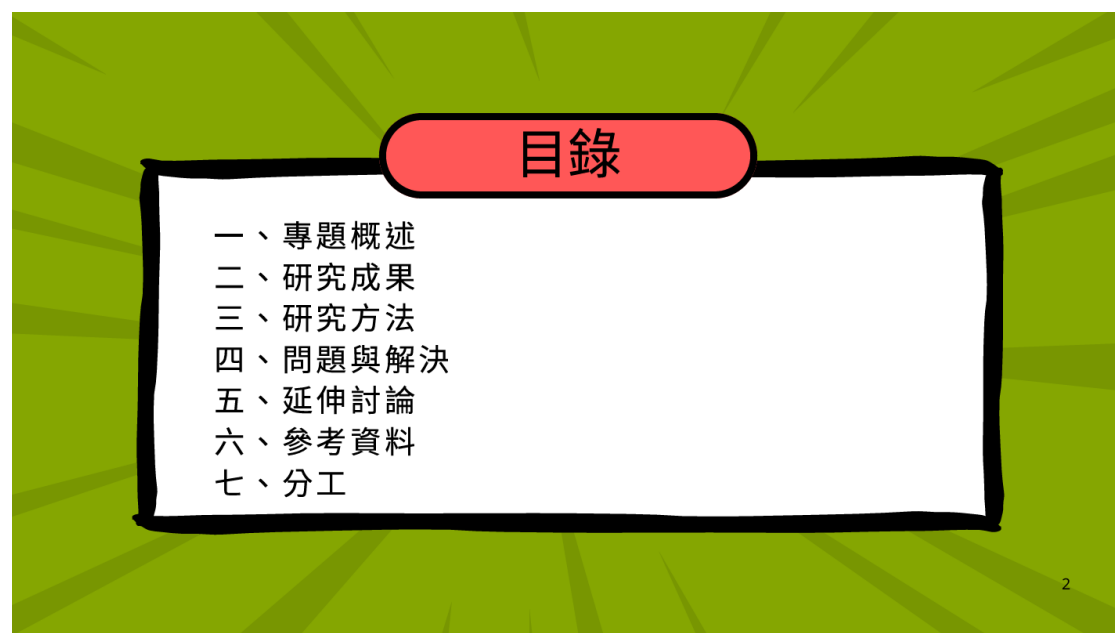
1. 可以在傘架上安裝提醒系統(如蜂鳴器、LED燈)，若天氣預測有雨，傘架上的提醒系統就會發出提醒。
2. 在傘架上安裝感測器，偵測傘有沒有放回傘架，如果傘不在，就亮起LED燈發出提醒，以避免將傘隨手亂放

11

分工表

411123021 鐘以涵	購買專題材料 拍攝影片 製作期中報告 口頭報告	50%
411123024 劉靜謙	撰寫程式 搜尋資料 製作期末報告 口頭報告	50%

18



專題概述

3

一、專題概述

此專題結合電腦視覺與手勢辨識技術的互動式停車遊戲，透過 **MediaPipe** 偵測玩家的手部動作，搭配 **Pygame** 介面實現即時控制，控制車輛完成停車任務。玩家無需實體控制器，而是以直覺的手勢完成停車任務，提升互動體驗與技術應用多樣性。

透過攝影機對手勢進行辨識，即可達成上下左右移動、暫停車輛以及重置遊戲等操作。本遊戲設計具備倒數計時機制、隨機生成停車格與得分統計，增添了許多挑戰與趣味。

4

一、專題概述

創新與獨特

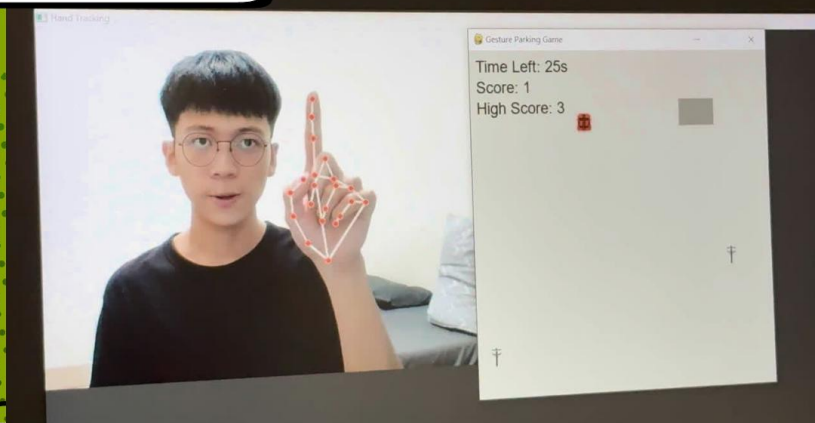
- 創新地將手勢辨識技術 (**MediaPipe**) 與遊戲引擎 (**Pygame**) 整合，可用手勢直接控制車輛移動的互動遊戲，跳脫傳統鍵盤或滑鼠操作模式也無需額外硬體設備。
- 自定義手勢控制邏輯與遊戲機制邏輯
 - a. 食指方向決定車輛行進方向（上、下、左、右）、手張開重新開始遊戲、大拇指暫停等
 - b. 結合停車成功判定、碰撞懲罰、分數紀錄與切換關卡等機制，打造完整遊戲邏輯與挑戰性流程。

5

研究成果

6

二、研究成果



7

研究方法

8

三、研究方法

主要技術

Python
開發語言

MediaPipe
進行手部
辨識

OpenCV
處理攝影機
影像

Pygame
呈現畫面
遊戲邏輯

9

三、研究方法

MediaPipe (手勢辨識模組)

- 是 **Google** 提供的機器學習架構，專門用來進行即時影像處理與追蹤 (**Hands**、**Pose**、**Face Mesh** 等)
- 使用 **MediaPipe Hands** 模型可辨識單手最多 **21** 個節點座標 (**landmarks**)
- 透過 **hands.process(rgb)** 對畫面進行手勢辨識，回傳手部節點資訊

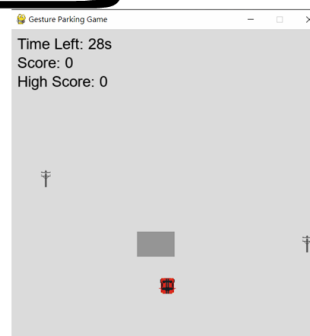


10

三、研究方法

Pygame (遊戲引擎模組)

- 是基於 **Python** 的 **2D** 遊戲開發套件
- 用於建立畫面主視窗、渲染車輛圖像與停車格、繪製分數與計時器等介面元素
- 使用 **pygame.Rect()** 定義遊戲中物件的範圍與碰撞邏輯
- 使用 **pygame.display.update()** 更新畫面、**pygame.transform.rotate()** 處理車輛旋轉角度



11

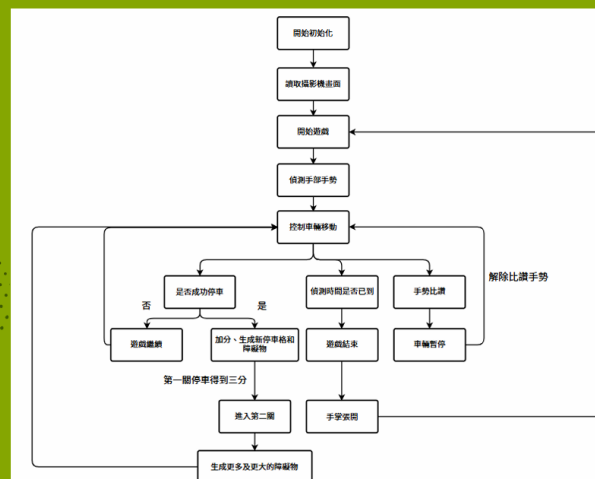
三、研究方法

OpenCV :

<code>cv2.VideoCapture(0)</code>	開啟筆電內建攝影機，擷取即時影像影格
<code>cv2.flip(frame, 1)</code>	水平翻轉畫面，使手勢方向與畫面一致
<code>cv2.imshow()</code>	顯示處理後的畫面（含手部追蹤結果）
<code>cv2.waitKey(1) + ord('q')</code>	每幀檢查是否有按下 q 鍵，按下則離開程式

12

流程圖



13

遊戲機制

三、研究方法

功能說明	設計說明	函式
計時挑戰	30 秒內完成任務，否則顯示 TIME'S UP	<code>time_limit</code>
判定得分	停車格與車身重疊超過一半即得 1 分	<code>pygame.Rect.clip()</code>
隨機生成停車格座標	每次得分或遊戲重置後，重新隨機產生停車格	<code>get_random_parking_spot()</code>
每幀更新畫面顯示	不斷更新 Pygame 畫面、分數、計時器、車圖之類的	<code>pygame.display.update()</code>
根據方向旋轉車圖	根據移動方向，將車圖轉為上/下/左/右方向	<code>pygame.transform.rotate()</code>
車輛邊界限制	避免車輛超出畫面範圍 (X/Y 軸邊界限制)	<code>car_rect.x car_rect.y</code>
暫停與重置控制	判斷手勢狀態，決定是否進入煞車或重置邏輯	<code>brake, pause_start_time, is_all_fingers_up()</code>
碰撞障礙物麻痺	車輛碰撞到障礙物，麻痺兩秒	<code>stunned_obstacles.append((obs, now))</code>

14

三、研究方法

手勢功能說明

手勢	功能說明	設計	函式
食指上下左右	控制車輛方向 (前後左右)	比較 x/y 差值大小 → 判左右還是上下 方向的正負值 → 判斷是左/右或上/下	無獨立函式，直接在主程式中偵測 (以食指尖index 8與基部index 6 之間的向量)
大拇指豎起	煞車 (暫停車輛) brake = True	暫停後不會移動車子，直到放開手勢	if thumb_tip.y < thumb_ip.y - 0.05 指尖比關節高過0.05觸發煞車
五指張開	重置遊戲	持續 0.5 秒以上 → 重置分數與位置	is_all_fingers_up()

三、研究方法

關卡功能說明

為了提升遊戲的挑戰性與層次，我們設計了關卡機制作為研究的一部分。透過設定第一關需要成功停車三次才可過關，並在過關時顯示提示文字「Next Level 2」，引導玩家進入第二階段挑戰。在第二關中，我們將障礙物數量由兩個提升為四個，並放大障礙物尺寸，增加閃避難度，同時維持每關 30 秒的時間限制，確保遊戲節奏不拖沓。

程式實作上，我們以變數 `current_level` 控制遊戲階段，透過條件判斷 `if score >= 3` 進入關卡轉換狀態，並使用暫停邏輯避免轉場時立即受到碰撞懲罰。障礙物則根據關卡階段調整數量與尺寸，並搭配 `pygame.Rect` 進行位置與碰撞管理，使每一關的挑戰度明顯區隔。

問題與解決

四、問題與解決

- 車輛圖片沒有跟方向旋轉：
使用 `pygame.transform.rotate()` 並搭配 `car_direction`，依上下左右自動旋轉車圖，讓方向一致。
- 車輛碰撞停車格時判斷不準確：
原本是用 `colliderect()` 判斷，只要車子碰到就算得分
改用 `car_rect.clip(parking_spot)` 計算重疊面積，只有當重疊面積大於一半才算停進去。
- 手勢偵測不靈敏：
偵測食指方向時，手勢偵測不靈敏反應遲鈍，因此降低 `dx/dy` 的靈敏度門檻（設定大於 ± 0.01 就觸發方向）

18

延伸討論

19

五、延伸討論

- 分析個人分數：利用視覺化分析做圖表，學習數據處理，顯示每局得分走勢、反應速度、停車準確度與失敗時間。
- 使用 `.exe` 直接遊玩：改成用 `python` 寫
- 其他應用：本專題展示手勢控制在人機互動與遊戲設計中的應用潛力，未來可拓展為駕駛模擬教學、手部復健訓練等。
- 播放音效：在遊戲主迴圈中寫入播放音效函式

20

參考資料

21

六、參考資料

- [pygame](#) 官網教學及開源庫
- [MediaPipe](#) 教學

22

分工

23





大綱

研究目的

研究方法

結果與討論

未來展望

參考資料

團隊分工

**1**

研究目的

本專題是希望開發一套自動化車流量偵測系統，利用YOLO深度學習物件偵測模型，配合國道攝影機即時影像，準確統計道路上車輛數量。希望可以以此取代不必要的人工成本，提升交通監控的效率與準確性，並可作為智慧交通管理的基礎工具。

2

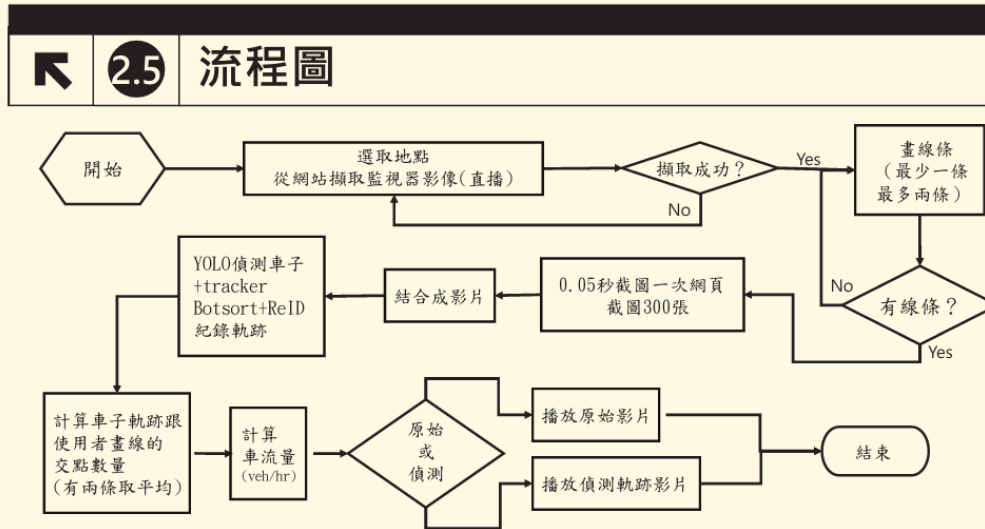
**2**

研究方法

- 系統架構：利用國道網站，核心演算法為YOLO物件偵測模型，負責辨識影像中的車輛。
- 圖形化介面：以PyQt6設計操作介面，採用一幀一幀擷取後轉成影片，計算出車流量。
- 流程步驟：
 - 載入YOLO模型，讀取攝影機影像或影片。
 - 透過YOLO對每一幀進行車輛偵測，統計目標數量。
 - 結果即時顯示於PyQt介面，並可將統計資料存檔。
- 技術重點：
 - 物件偵測演算法（YOLO）
 - OpenCV影像處理
 - PyQt6圖形化互動設計



3



4

2.5 YOLO 訓練過程

- 使用 roboflow 上公開的資料集
- 專案"j"
 - 監視器視角
 - 9592張標註好的相片
 - 五個分類

```

nc: 5
names: ['bus', 'construction', 'cyclist', 'pedestrian', 'vehicle']
  
```

Dataset Split

TRAIN SET	VALID SET	TEST SET
78%	10%	12%
6715 Images	1919 Images	958 Images

5

2.5 YOLO 訓練過程

- 使用colab 訓練

```
from ultralytics import YOLO

# 選擇模型大小:
model = YOLO('yolov8m.pt')

# 開始訓練
model.train(
    data='/content/data.yaml',
    epochs=100,
    imgsz=640,
    batch=16,
    name='yolov8_custom'
)
```

6

2.5 YOLO 訓練過程

```
# 載入訓練完成的模型
from ultralytics import YOLO
model =
YOLO('/content/drive/MyDrive/car_detect_output3.0/yolov8_custom/
weights/best.pt')

# 在 test 資料上驗證
metrics = model.val()
```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)	100%		125/125	[00:18:00:00, 6.72it/s]
all	1999	24833	0.81	0.766	0.829	0.565				
bus	217	281	0.845	0.822	0.888	0.719				
construction	534	2637	0.798	0.714	0.776	0.436				
cyclist	137	221	0.695	0.679	0.734	0.513				
pedestrian	760	3304	0.82	0.743	0.814	0.479				
vehicle	1868	18390	0.891	0.873	0.934	0.676				

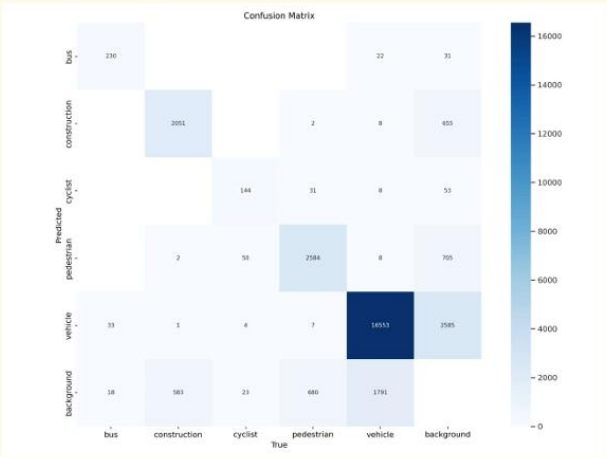
Speed: 0.2ms preprocess, 2.4ms inference, 0.0ms loss, 1.9ms postprocess per image

8

↶

2.5

YOLO 訓練過程

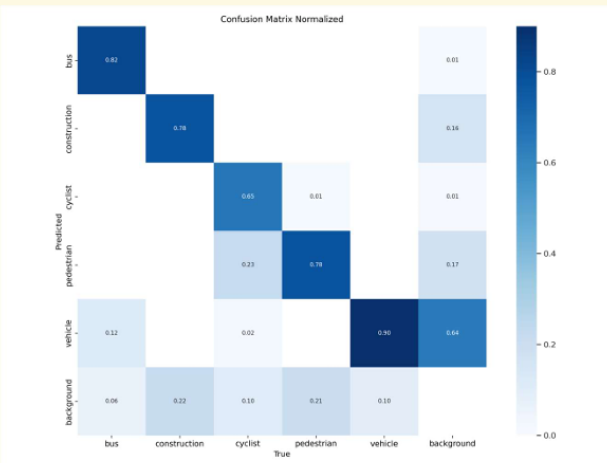


10

↶

2.5

YOLO 訓練過程



11

2.5 YOLO 訓練過程

你要多有信心時，才撿東西，才最厲害？

F1 Confidence Curve

模型在驗證集上進行推論，
並針對每一張圖像給出預測框和信心分數。

將信心門檻從 0 遞增到 1（例如每次增加 0.01）。

每個門檻下，計算：

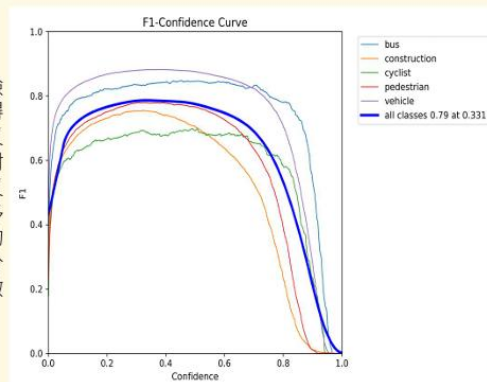
有哪些框被保留（confidence ≥ threshold）
對應的 Precision、Recall、F1-score

畫出不同門檻下的 F1-score 曲線。

$$F1\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

曲線的最高點對應於模型在某一個特定
門檻值下達到的最佳平衡點
（最佳 Precision 與 Recall 的折衷）

撿得又對又多的分數

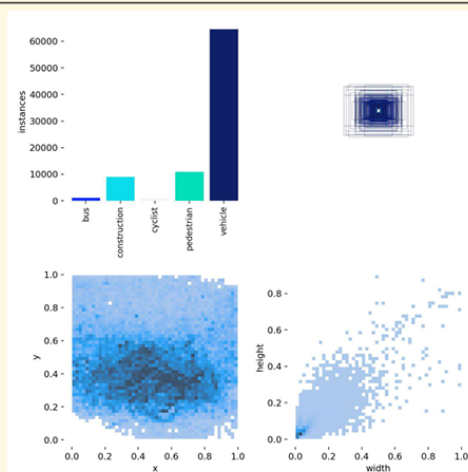


要多有信心才願意撿？
（例如：0 是什麼都撿，1 是超級確定才撿）

12

2.5 YOLO 訓練過程

每個類別的
樣本數量統計圖



整體標註的框長什麼樣子

熱圖：出現目標位置

標註框的寬度與高度的分佈圖

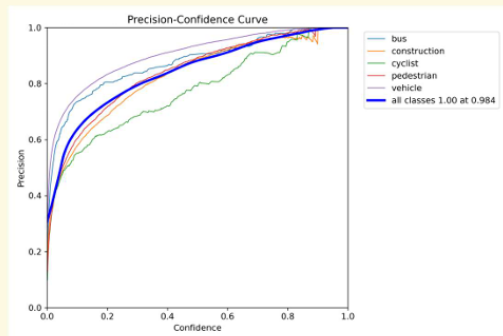
13



2.5

YOLO 訓練過程

說對的機率



多有把握這是對的

決定要不要
「只採用最有信心的預測」？

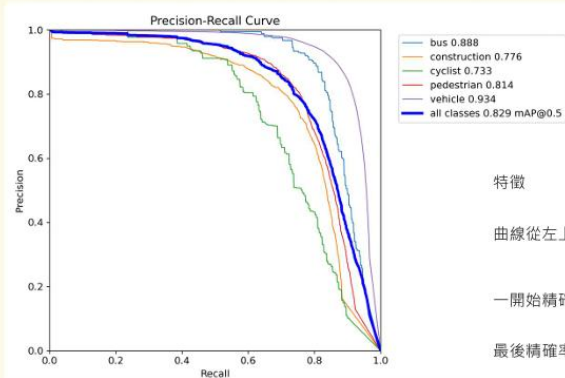
14



2.5

YOLO 訓練過程

你找出來的車裡，有幾個是真的車？



照片裡本來有幾台車，你找到了幾台？

要在一堆照片中找出「車」

特徵

曲線從左上延伸到右下

一開始精確率高、召回低

最後精確率降低、召回高

曲線下方面積 (AUC) 越大越好

說明

代表：越多召回，精確率通常會降低 (因為抓得多，錯的也多)

模型非常保守，只預測最有把握的樣本

模型預測很多樣本，有更多錯誤出現

表示 precision 和 recall 同時表現得好

15

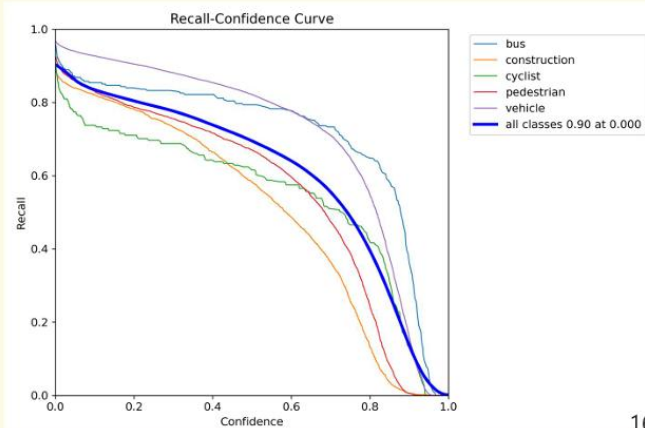


2.5

YOLO 訓練過程

recall-confidence curve 呈「從左上到右下且凸向右上」

- 模型在低 confidence 時能抓到較多正樣本 (高 recall)
- 但信心值逐漸提高時，因預測變保守 → recall 自然下降
- 凸形曲線則反映信心分布或模型 calibration 的特性



16



2.5

YOLO 訓練過程

現象

所有 loss 曲線都下降並平穩

可能原因

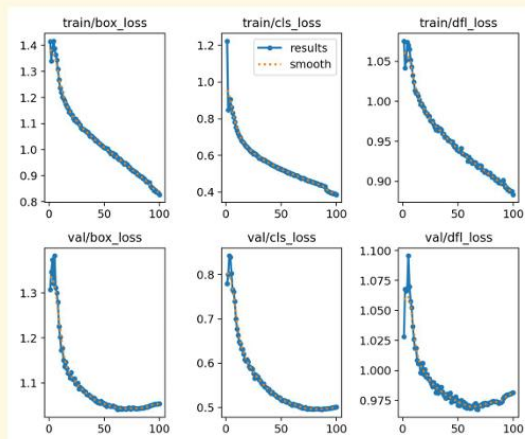
正常訓練狀況

train loss 降低，但 val 升高

過擬合。改進：減少 epoch、增加資料、用 dropout。

loss 突然跳高或震盪

學習率過高、數據亂、模型不穩定。

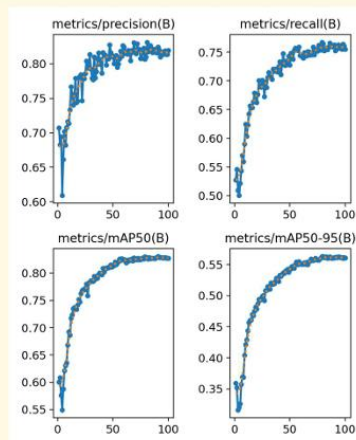


17

**2.5**

YOLO 訓練過程

圖名	中文名稱	代表什麼意思？
metrics/precision	精確率 Precision	預測為正的裡面，有多少是真的。
metrics/recall	召回率 Recall	所有應該抓到的目標中，有多少真的被抓到了。
metrics/mAP_0.5	mAP@0.5	越高越好。常作為主要衡量指標。
metrics/mAP_0.5:0.95	mAP@0.5:0.95	更嚴格的平均精度。越高代表模型越精準且穩定。

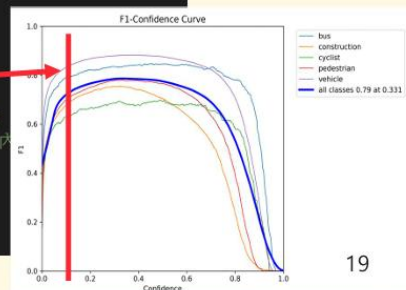


18

**2.5**

程式使用數值調整

```
# 載入模型
current_dir = os.path.dirname(os.path.abspath(__file__))
model_path = os.path.join(current_dir, "best.pt")
print(f"Looking for model at: {model_path}")
# 設置模型權重檔案位置
current_dir = os.path.dirname(os.path.abspath(__file__))
tracker_path = os.path.join(current_dir, "botsort.yaml")
# 執行追蹤
results = self.model.track(
    source=img,
    persist=True,
    conf=0.1,
    iou=0.7, #兩個框重疊程度大於0.7，刪掉信心小的框
    tracker=tracker_path, # 使用與主程式同一資料夾內
    device=0 # 使用GPU
)
```



19

**2.5**

程式使用數值調整

! botsort.yaml X

```
tracker_type: botsort
# 使用的追蹤演算法類型，可選 'botsort' 或 'bytetrack'
track_high_thresh: 0.4
# 高信心值匹配的門檻值，只有 confidence > 0.4 的偵測框會參與第一次匹配（通常是主匹配階段）
track_low_thresh: 0.1
# 低信心值匹配門檻，第二階段補充匹配用（如果第一階段沒成功則用低信心框試著匹配）
new_track_thresh: 0.5
# 啟動新追蹤 ID 的最小信心值，若未匹配成功但 confidence > 0.5 就會新增一個追蹤 ID
track_buffer: 10
# 若目標在這段時間內（以幀為單位）沒有被匹配到，就會移除該追蹤 ID，數字越大越不容易消失
match_thresh: 0.9
# 判定兩個框是否為同一目標的 IOU 門檻，若 IOU > 0.9 才算匹配成功
fuse_score: False
# 是否將 confidence score 加入 IOU 計算，用於匹配時加權判斷；False 表示不融合，只看 IOU
min_box_area: 8
# 最小框面積的限制，用來過濾過小的框（目前此參數未實際啟用）
```

20

**2.5**

程式使用數值調整

! botsort.yaml X

```
# BoT-SORT 設定
gmc_method: None
# Global Motion Compensation 方法，用於補償攝影機移動，目前未啟用可設 None

# ReID 模型相關設定（BoT-SORT 特有）
proximity_thresh: 0.1
# 啟用 ReID 判斷前的最小 IOU 門檻，IOU < 0.1 則不使用 ReID 匹配
# 只有當「目前追蹤中的物件」與「新的偵測框」之間的 IoU（交集比聯集）大於 0.1 時，
# 才會進一步使用 ReID（外觀特徵）來做匹配。
appearance_thresh: 0.1
# ReID 外觀特徵相似度門檻，值越小越寬鬆（容易誤配），建議依模型精度調整
with_reid: True
# 是否啟用 ReID 模型來輔助辨識相同物件的 ID，一般建議設為 True
model: c:\car_detect\best.pt # 自訂的 ReID 模型路徑，不是 YOLO 模型，而是分類/特徵模型，
# 用來比對外觀是否相似
```

21



3

未來改進

對於模型的選擇與訓練方面：

1. 測試不同版本與模型大小的結果。使用更加合適的模型。
2. 使用更多的數據集，增加不同車種的數量。
3. 使用資料擴增。
4. 更加詳細研究訓練模型當中的各個參數。多加利用，調整。
5. 改進在不同天氣，時間下偵測的不準確。

影像使用方面：

1. 使用更清晰順暢的影片來源。
2. 使用更正當的影片。

程式設計方面：

1. 介面在更優化一些
2. 電腦資源的利用優化
3. 流程精簡

22

4 未來展望

1. 增強夜間及惡劣天氣下的辨識能力
2. 多地點同步分析、車種細分類
3. 發展嵌入式即時運算與雲端數據匯整
4. 與智慧交通系統整合，實現自動化管理與預警功能

23

5 團隊分工

黃登樺

程式碼修改+套件使用
更改+更改偵測位置



蔡佳勳

程式設計+資料收集
+簡報製作



白惟中

實驗測試、結果分
析與簡報製作



24

6 參考資料

- Ultralytics YOLOv8 官方文件
- BoTSORT: Robust Association Multi-Object Tracking with Box-Level Tracking and Appearance Embedding
- Roboflow 公開資料集
- 相關智慧交通與影像處理論文

25