

國立高雄科技大學(第一校區)

資訊管理系

109 學年度

實務專題期末報告

題目

人工智慧之影像辨識

-口罩監控系統

組員：蕭仁豪 C107118111、蕭煜宸 C107118137、
黃仲偉 C107118151、洪嘉聰 C107118250

指導老師：莊文勝 博士

中 華 民 國 110 年 6 月 4 日

國立高雄科技大學(第一校區)資訊管理系

大學部實務專題期末報告確認單

專 題 題 目	人工智慧之影像辨識 -口罩監控系統
組 員	蕭仁豪、蕭煜宸、黃仲偉、洪嘉聰
指 導 老 師	莊文勝
<p>* 專題內容如無須修正，煩請指導老師簽名，謝謝。</p> <p>指導老師簽名：莊文勝</p>	

一、 摘要

近年來人工智慧 (Artificial Intelligence, AI) 發展非常迅速，給人們帶來許多意外驚喜。從 2016 年開始 AI 相關的技術廣泛應用於智慧控制、機器人、自動化控制、棋類遊戲、醫學領域以及語言圖像辨識領域，尤其在影像辨識、語言分析等方面，在許多的研究成果中，在單方面的能力展現已達到超越人類的水準。

本研究重要的主題在於利用 YOLO 物件偵測來達到口罩檢測，並使用樹莓派做為監控平台端點的建置，以確保是否正確配戴口罩。

關鍵詞：人工智慧、樹莓派、影像辨識、口罩檢測

目錄

一、	摘要.....	3
二、	緒論.....	6
	2.1 研究的動機.....	6
	2.2 研究的問題.....	6
	2.3 研究目的.....	6
三、	文獻探討.....	7
	3.1 YOLO:基於深度學習的物件偵測	7
	3.2 YOLO 如何偵測物件.....	7
	3.3 YOLO 如何訓練.....	8
	3.4 Loss function.....	9
	3.5 Activation function.....	9
	3.6 OpenCV	10
	3.7 機器人作業系統[4](Robot Operating System,ROS).....	10
	3.8 機器學習.....	10
	3.9 深度學習.....	10
	3.10 樹莓派(Raspberry Pi)	10
四、	研究方法與步驟.....	12
	4.1 研究流程架構.....	12
	4.2 系統功能內容.....	12
	4.3 操作及功能.....	13
	4.4 人臉辨識之資安漏洞探討.....	14
	4.4.1.照片攻擊	14
	4.4.2.影片攻擊	14
	4.4.3.面具攻擊	14
	4.5 步驟流程.....	15
五、	目前進度.....	16
	5.1 研究介紹.....	16
	5.2 情況一.....	16
	5.3 情況二.....	17
	5.4 情況三.....	18
	5.5 後台畫面及通知.....	19
	5.6 寄信.....	20
	5.7 其他.....	20

圖目錄

圖 1：YOLO 的總體框架	7
圖 2：bounding box 之計算方式	7
圖 3：YOLO 之偵測模型	8
圖 4：YOLO multi-part loss function 之定義	9
圖 5：線性整流函數.....	9
圖 6：研究流程架構.....	12
圖 7：YOLO 物件檢測	13
圖 8：labelimg 頁面.....	14
圖 9：面具攻擊示意圖.....	15
圖 10：CPM 圖	15
圖 11：口罩樣本 A.....	16
圖 12：口罩樣本 A 檢測結果 with_mask	17
圖 13：口罩樣本 B.....	17
圖 14：口罩樣本 B 檢測結果 without_mask	18
圖 15：口罩樣本 C.....	18
圖 16：口罩樣本 C 檢測結果 mask_wearred_incorrect.....	19
圖 17：樹莓派傳送郵件和警示成功.....	19
圖 18：傳送 Line 通知	20
圖 19：個人信箱收到郵件和辨識結果的影片檔.....	20
圖 20：影片偵測檔.....	21
圖 21：google colaboratory 訓練檔	22

二、 緒論

(一)研究的動機

近年來人工智慧 (AI) 已成為備受全球各大產業矚目的熱門議題，預估至 2030 年為止，人工智慧將為全球市場帶來 15 兆美金的產值。

而影像辨識一直是人們最希望用人工智慧和機器學習來幫忙處理的問題。從基本的手寫文字辨識、物件識別到人臉辨識，還有無人駕駛車，都是深度學習和影像辨識整合後的應用。以此為動機配合現在 covid-19(新冠病毒)疫情，全球陷入危機，各國出入公共場所均須配戴口罩。而台灣最近發布全國進入第三級警戒，出門在外都須配戴口罩，便希望能透過影像辨識技術結合攝影鏡頭來製作口罩辨識的系統，並用樹莓派當作端點，架設在所需的場所，用以落實是否佩戴口罩，降低公共場所染疫風險。

(二)研究的問題

透過在各個場所架設 webcam 檢查出入的行人是否有正確配戴口罩，各場域端點架設的問題我們透過樹莓派來當作端點，不需像一般電腦一樣價格昂貴且占空間，能簡單實現。而使用 webcam 辨識口罩的同時，也會拍攝到人臉，這部分可能涉及隱私權的問題，是我們需要再討論的。

(三) 研究目的

本研究在樹莓派上安裝 yolo v5 做為監控平台的開發架構，監控平台需要透過前方的攝像鏡頭即時分析圖像進而辨識人臉是否佩戴口罩，；例如:公共場所人群眾多，出入的人若未配戴口罩或是沒有正確配戴口罩，透過攝影鏡頭拍攝到，經由樹莓派辨識確認後便會發出警示訊息。

本研究之目的為解決及探討以下相關問題:

(1) 即時分析圖像:

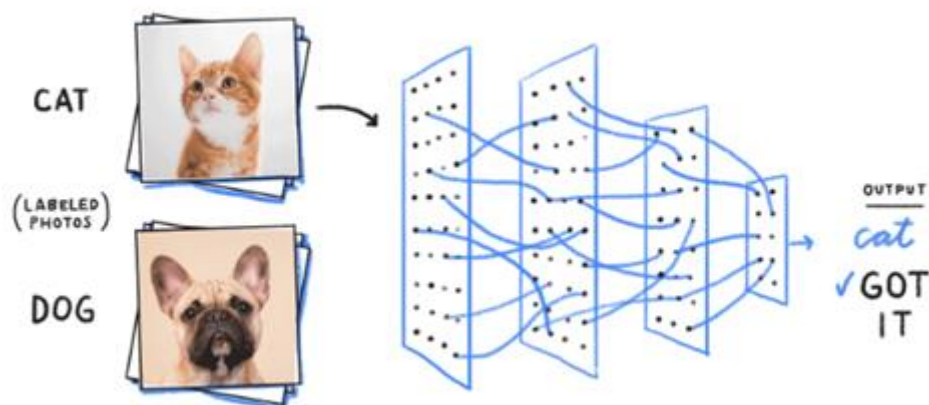
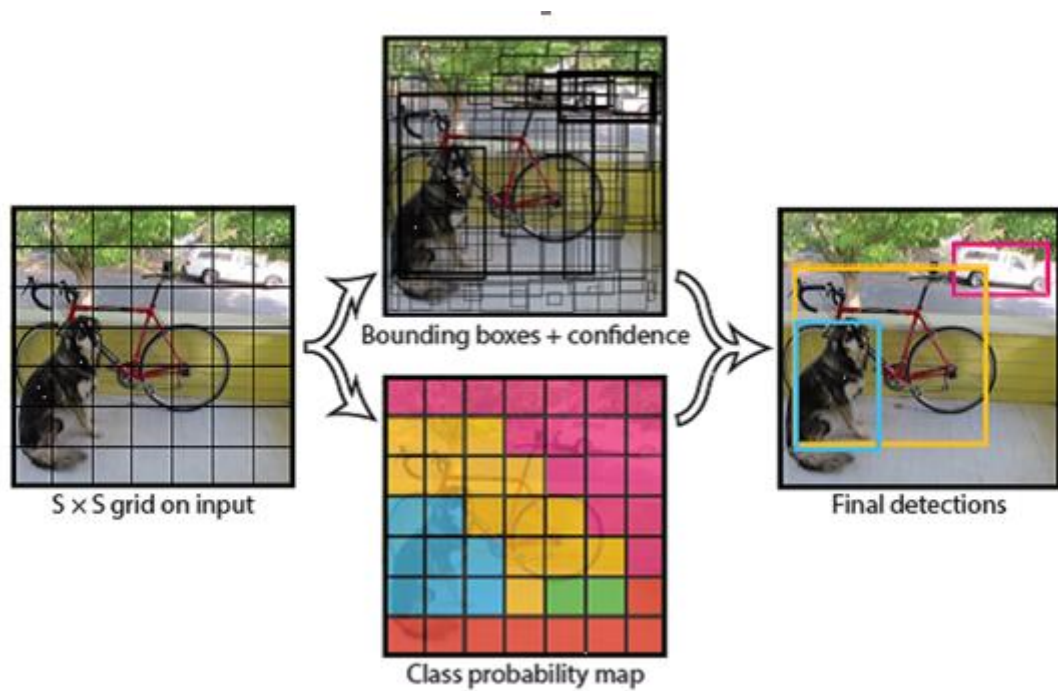
讓 yolo v5 辨識即時辨識影像，且不會因場域不同就無法偵測。

(2) 將訓練完成的模型結合樹莓派:

讓訓練完成的模型結合樹莓派，使訓練完成的深度學習偵測模型能架設在居家或公共場所中，以便監測是否配戴口罩。

(3) 強化資訊安全:

利用資訊共享、自我監督以及漏洞報告與揭露等來維護其資訊安全。



(圖 3 : YOLO 之偵測模型)

3.3 YOLO 如何訓練

YOLO 作者用 ImageNet 1000-class competition dataset 來預先訓練模型，但只訓練前 20 卷積層後面接上一個 average-pooling layer 和一個全連結層。

Bounding Box 正規化

YOLO 的最後一層預測的是類別機率和 bounding box 等在 YOLO 訓練前會先依據輸入圖的長寬，正規化 (normalize) bounding box 的長寬，因此 bounding box 的長寬會介於 0~1 之間。

Bounding box 的中心座標(x,y)是在特定 grid cell 的偏移(offset)，所以座標也會介於 0~1 之間。

3.4 Loss function

一般都用平方誤差和當作 loss function，原因是容易最佳化。作者認為此方法不能完美校正去最大化目標的平均精度，主要原因是每項目的 error(比如 bounding box 的定位誤差和分類的誤差)都佔有一樣的比重，所以結果不太好，而且物件偵測多數情況，大多數的 grid cell 內是沒有物件的(在梯度求解的時候容易將有物件的 cell 壓過去)，所以容易導致 confidence 幾乎趨近於 0，也因為如此容易造成模型不穩定。為了解決這個問題，作者增加了在 bounding box 座標預測的 loss 權重 ($\lambda_{coord}=5$)及減低那些不包含物件的 Box，confidence 預測時的權重($\lambda_{noobj}=0.5$)。

YOLO multi-part loss function(如圖 4)定義如下:

$$\begin{aligned} \text{loss}_{YOLO} = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

1_i^{obj} : 是物件有出現在 grid cell i 。

1_{ij}^{obj} : 是在第 i 個 grid cell 的第 j 個 Bounding Box 負責做預測。

(圖 4: YOLO multi-part loss function 之定義)

如此一來此 loss function 只會針對物件如果有出現在某個 grid cell 下進行懲罰分類錯誤。並且只會針對有責任於偵測 Ground truth box 的預測者進行懲罰 bounding box 錯誤，也就是 grid cell 中有最高 IOU 的預測者。

3.5 Activation function

整個 YOLO 架構除了最後一層用線性輸出為，每一層都會搭配線性整流函數(如圖 5)。

$$\text{leaky ReLU: } \phi(x) = \begin{cases} x & \text{if } x > 0 \\ 0.1x & \text{O.W.} \end{cases}$$

(圖 5：線性整流函數)

3.6 OpenCV

是一個跨平台的電腦視覺庫，由英特爾公司發起並參與開發，以 BSD 授權條款授權發行，可以在商業和研究領域中免費使用。OpenCV 可用於開發即時的圖像處理、電腦視覺以及圖型識別程式，該程式庫也可以使用英特爾公司的 IPP 進行加速處理，目前運用的領域有物體辨識、臉部辨識及圖像分割等。

3.7 機器人作業系統[4](Robot Operating System,ROS)

專為機器人軟體開發所設計出來的一套電腦作業系統架構，它是一個開源的元級作業系統(後作業系統)，提供類似於作業系統的服務，包括硬體抽象描述、底層驅動程序管理、共用功能的執行、程序間消息傳遞、程序發行包管理，同時它也提供一些工具和庫用於獲取、建立、編寫和執行多機融合的程序。

ROS 的運行架構是一種使用 ROS 通信模塊實現模塊間 P2P 的鬆耦合的網絡連接的處理架構，它執行若干種類型的通訊，包括：(1)基於服務的同步 RPC(遠程過程調用) 通訊；(2)基於 Topic 的異步數據流通訊，還有參數伺服器上的數據儲存。

3.8 機器學習

機器學習是人工智慧的核心，屬於人工智慧的一個分支，是一個大的領域，是讓計算機擁有像人一樣的學習能力，模擬和實現人的學習行為和能力，可以像人一樣具有識別和判斷的能力，可以看作是仿生學。機器學習的核心就是資料，演算法(模型)，算力(計算機運算能力)，不過最近幾年網路發展和大資料的積累，使得人工智慧能夠在資料和高運算能力下發揮它的作用，機器學習應用領域十分廣泛，例如：資料探勘、資料分類、計算機視覺、自然語言處理(NLP)、生物特徵識別、搜尋引擎、醫學診斷、檢測信用卡欺詐、證券市場分析、DNA 序列測序、語音和手寫識別、戰略遊戲和機器人運用等。

3.9 深度學習

深度學習是機器學習中一種基於對資料進行表徵學習的演算法。觀測值可以使用多種方式來表示，例如每個像素強度值的向量，或者更抽象地表示成一系列邊、特定形狀的區域等，而使用某些特定的表示方法更容易從實例中學習任務例如，臉部辨識或面部表情辨識，深度學習的好處是用非監督式或半監督式的特徵學習和分層特徵，提取高效演算法來替代手工取得特徵。

3.10 樹莓派(Raspberry Pi)

樹莓派是一款基於 Linux 系統的單晶片電腦，它由英國的樹莓派基金會所開發，目的是以低價硬體及自由軟體刺激在學校的基本電腦科學教育。

樹莓派配備一枚博通(Broadcom) 出產的 ARM 架構 700MHz BCM2835 處理器，256MB 記憶體(B 型已升級到 512MB 記憶體)，使用 SD 卡當作儲存媒體，且擁有一個 Ethernet、兩個 USB 介面、以及 HDMI(支援聲音輸出) 和 RCA 端子輸出支援，樹莓派只有一張信用卡大小，體積大概是一個

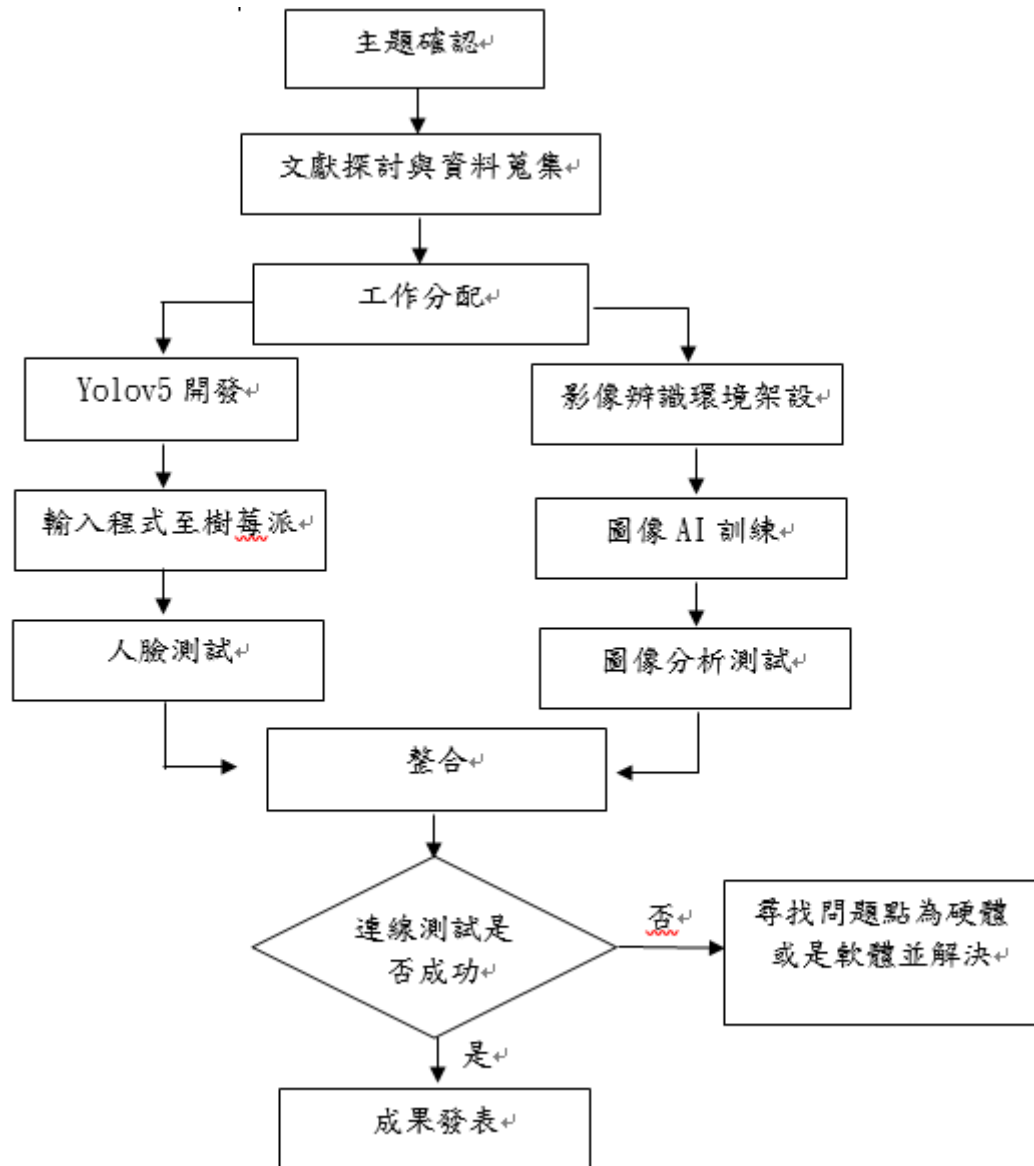
火柴盒大小，卻可以執行遊戲和進行 1080p 影片的播放。操作系統採用開源的 Linux 系統如 Debian、ArchLinux，自帶的 Iceweasel、KOffice 等軟體，能夠滿足基本的網路瀏覽、文字處理以及電腦學習的需要。

作業系統

主要使用以 Linux 核心的作業系統，目前 Raspbian 是樹莓派所有型號的官方作業系統，樹莓派 1 搭載了基於 ARMv6 架構的 ARM11 晶片，目前的幾個流行的 Linux 版本，包括 Ubuntu 在內，均不能在 ARM11 上執行，Windows 無法在未改裝的樹莓派上執行；樹莓派 2 目前只支援 Ubuntu Snappy Core、Raspbian、OpenELEC 和 RISC OS，樹莓派 2 和 3 已經可以執行"Windows 10 物聯網核心版；樹莓派 3 已經可以執行 Windows 10 ARM 版本，2019 年樹莓派 4 的一個重要的新功能是預設使用開源 OpenGL 影片驅動程式，且這是使用最新版本的 Debian 來開發的。

四、 研究方法與步驟

4.1 研究流程架構



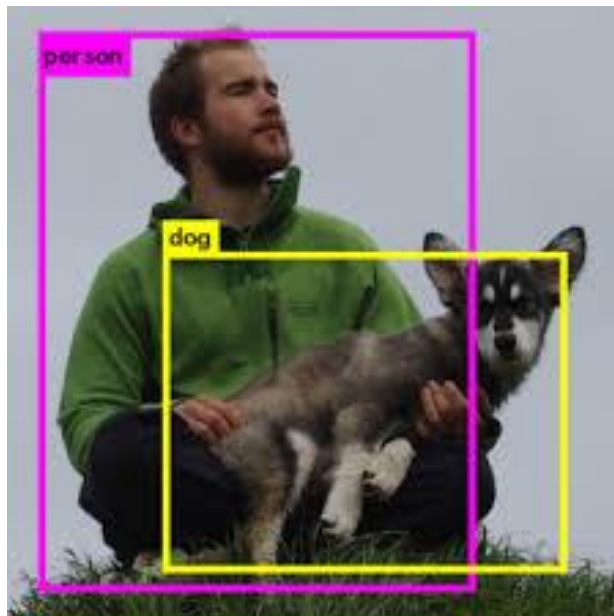
(圖 6：研究流程架構)

4.2 系統功能內容

本研究利用攝像鏡頭擷取圖像做即時分析，會先將圖像傳至 YOLO5，再從圖檔之資料庫當中找出相似之物件，並判斷其物件為哪個身分 ID，進而由樹莓派進行分析及比對，當資料庫無法獲得該身分時，蜂鳴器會響起，從而排除防盜程式之錯誤亦或是鏡頭本身之問題。

4.3 操作及功能

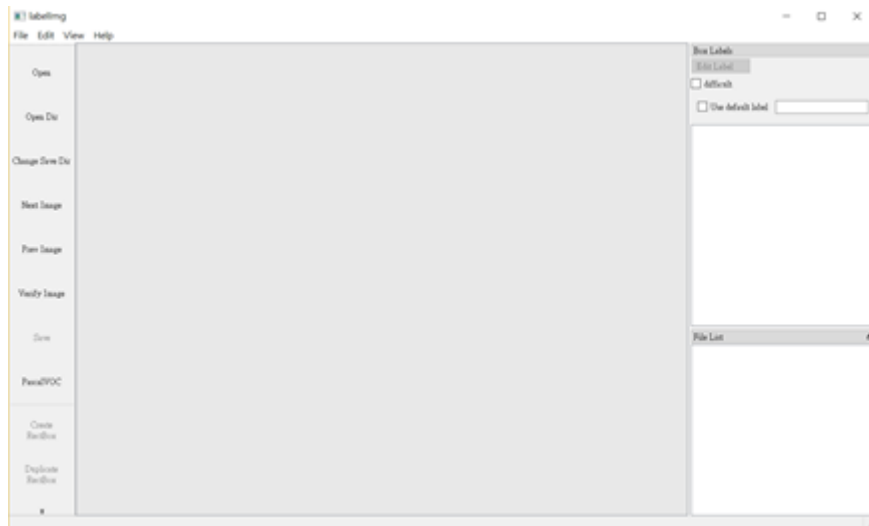
本研究使用 google colab 來進行訓練，訓練完成後，會產生訓練檔，這時 YOLO 環境架設便完成(如圖 7)。



(圖 7：YOLO 物件檢測)

安裝 labelimg(如圖 8)，過後便能開始建立 Bounding box，把存下來準備訓練 xml 檔放進對應的路徑上，爾後更改 darkflow 裡面有個 labels.txt，把裡面的標籤改成道路、行人、車子等的名稱，設定完成後開始進行模型的訓練。

防盜硬體的部分，主要以樹莓派、WebCam、蜂鳴器組裝而成，再將機器人作業系統安裝至樹莓派上當作開發架構，之後將圖像傳送至本機端辨識，圖像從訓練模型當中找出相似特徵之標籤名稱後，再回傳至作業系統，進而分析比對資料庫內與即時偵測產生的內容，如果及時偵測的物件無法在資料庫比對中找出相對應的物件，則蜂鳴器會響起。



(圖 8 : labeling 頁面)

4.4 人臉辨識之資安漏洞探討

4.4.1.照片攻擊

針對人臉辨識系統的偽冒攻擊成本最低且最容易執行手法即為照片攻擊 (Photo Attack)。攻擊者可利用偷拍的方式或從網路社群媒體下載合法用戶的臉部照片，並利用高解析的印表機印出照片或者直接使用手機螢幕，在人臉辨識系統的感測器前顯示合法用戶的臉部影像，即可實施照片攻擊。

4.4.2.影片攻擊

惡意攻擊者不使用靜態的人臉照片，而是使用數位設備（例如手機、平板或筆記型電腦）播放合法用戶的臉部動態影片。影片攻擊屬於進階式攻擊手法，在感測器前顯示合法用戶的各種動態動作，例如頭部擺動、臉部表情、眨眼等等，因此更加難以防範。

4.4.3.面具攻擊

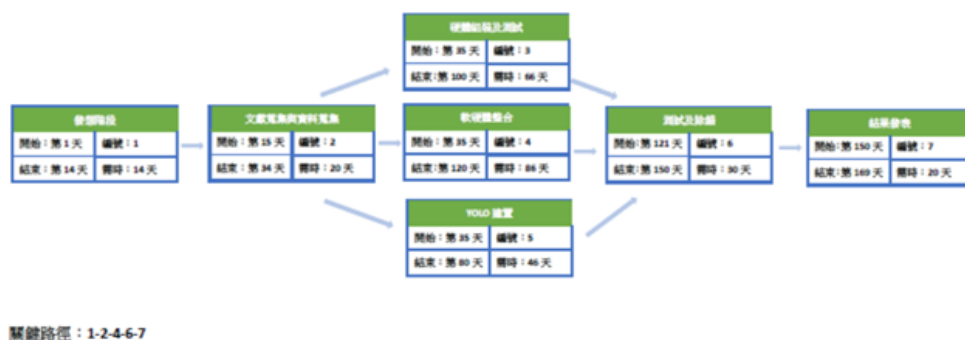
使用者透過配戴合法使用者的 3D 臉部矽膠面具對人臉辨識系統實施偽冒攻擊，如下圖所示。由於矽膠面具仿造了合法用戶臉部的完整 3D 結構，因此這種特殊威脅十分難以偵測。但 3D 面具製作不易，須具備特殊設備及高成本，較不容易實現；加以 3D 面具是僵硬的，缺乏具體的生理資訊（如表情與溫度變化），要有效達到攻擊目的亦有其困難之處，所以前述的照片和影片攻擊是對人臉辨識系統最常見的偽冒攻擊手段。



(圖 9：面具攻擊示意圖 資料來源：Akhtar Z., Micheloni C. and Foresti G. L., “Biometric Liveness Detection: Challenges and Research Opportunities”, IEEE Security & Privacy, vol. 13, no. 5, pp. 63-72, 2015.)

4.5 步驟流程

研究流程以 CPM 圖(如圖 10)的方式表示，預計花費兩週時間架構完整的計畫，擬定使用什麼工具語言 Coding、使用何種物件偵測去辨識等，之後進行文獻探討與資料蒐集，由於此成果為技術層面，需要許多的資料及教學影片多方參考來得到最佳方法，以便達到最佳成果。YOLO 環境建置與防盜系統的組裝須並行開發，以便於之後軟體與硬體連線測試；硬體與軟體整合時間預計 12 週，剩餘時間以 YOLO 進行訓練提高其準確性，最終放在各個場域進行模擬測試，並於最後四週測試完整性及運作的順暢程度。



(圖 10：CPM 圖)

五、 目前進度

5.1 研究介紹

本研究包含使用物件偵測及深度學習，將圖像即時分析傳送至電腦或樹莓派並利用 WebCam 監控場域的即時狀況後回報場所狀況至手機或電腦，是深度學習的重要應用之一。

以下是我們的人工智慧之影像辨識-口罩監控系統展示的進行流程在各個公共場所或所需地點架設 webcam 和樹莓派作為端點進行監控。下圖(如圖 11)透過 Webcam 拍到到進出場域的人員，便會開始進行辨識。



(圖 11：口罩樣本 A)

5.2 情況一

有配戴口罩，便會顯示 with_mask (如圖 12 所示)因為正確配戴口罩所以沒有發出警示。



(圖 12：口罩樣本 A 檢測結果 with_mask)

下圖(如圖 13)是繼續透過 Webcam 拍到進出場域的人員，便會開始進行辨識。



(圖 13：口罩樣本 B)

5.3 情況二

沒有配戴口罩，便會顯示 without_mask (如圖 14 所示)配戴口罩不正確便會發出警示和寄信(配合情況三在以下一起進行說明)



(圖 14：口罩樣本 B 檢測結果 without_mask)

下圖(如圖 15)是繼續透過 Webcam 拍到進出場域的人員，便會開始進行辨識。



(圖 15：口罩樣本 C)

5.4 情況三

雖然有配戴口罩，但配戴不正確，沒有遮住口鼻，便會顯示 mask_wearred_incorrect(如圖 16 所示)如果出現情況二和情況三，配戴口罩不正確或沒有佩戴口罩便會發出警示和寄信



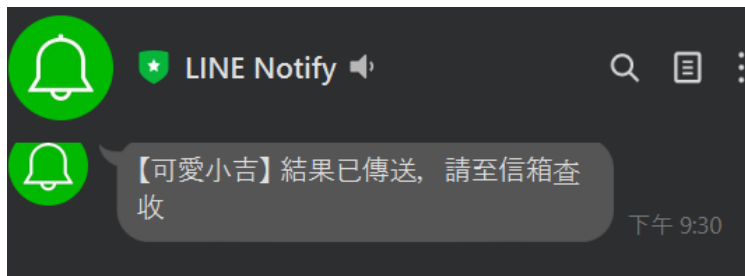
(圖 16：口罩樣本 C 檢測結果 mask_worn_incorrect)

5.5 後台畫面及通知

發生情況二和情況三，因為沒配戴口罩和口罩配戴不正確，後台便會寄信至 E-mail (如圖 17 所示)

```
C:\WINDOWS\system32\cmd.exe
video 1/1 (365/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 1 without_mask, Done. (0.119s)
video 1/1 (366/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 1 without_mask, Done. (0.119s)
video 1/1 (367/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.120s)
video 1/1 (368/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.124s)
video 1/1 (369/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.120s)
video 1/1 (370/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.118s)
video 1/1 (371/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.120s)
video 1/1 (372/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.120s)
video 1/1 (373/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.121s)
video 1/1 (374/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.119s)
video 1/1 (375/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.122s)
video 1/1 (376/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.117s)
video 1/1 (377/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.117s)
video 1/1 (378/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.119s)
video 1/1 (379/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.119s)
video 1/1 (380/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.119s)
video 1/1 (381/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.121s)
video 1/1 (382/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.120s)
video 1/1 (383/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.123s)
video 1/1 (384/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.120s)
video 1/1 (385/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.118s)
video 1/1 (386/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.117s)
video 1/1 (387/387) F:\Topic\yolov5\runs\video\result5.mp4: 256x416 Done. (0.118s)
Results saved to runs\detect\exp32
Done. (52.001s)
請輸入save_dir/檔名
runs\detect\exp32\result5.mp4
郵件傳送成功！
(dp1) F:\Topic\yolov5>
```

(圖 17：樹莓派傳送郵件和警示成功)



(圖 18：傳送 Line 通知)

接收 Line 訊息提醒結果已寄至信箱 (如圖 18 所示)

5.6 寄信

在信箱中收到檢測結果的影片檔，並進行警示 (如圖 19 所示)



(圖 19：個人信箱收到郵件和辨識結果的影片檔)

5.7 其他

以下是我們的影片偵測檔和影像辨識的訓練檔(圖 20 和圖 21)

```
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.4, device='', exist_ok=False, img_size=416, iou_thres=0.45, name='exp', nosave=False, project='runs/detect', save_conf=False, save_txt=False, source='runs/video/result2.mp4', update=False, view_img=False, weights=['runs/model/last.pt'])
YOLOv5 v4.0-109-gfd96810 torch 1.8.0+cpu CPU

Fusing layers...
Model Summary: 224 layers, 7059304 parameters, 0 gradients, 16.3 GFLOPS
video 1/1 (1/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.261s)
video 1/1 (2/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.131s)
video 1/1 (3/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.120s)
video 1/1 (4/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.121s)
video 1/1 (5/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.120s)
video 1/1 (6/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.120s)
video 1/1 (7/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.122s)
video 1/1 (8/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.120s)
video 1/1 (9/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.121s)
video 1/1 (10/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.124s)
video 1/1 (11/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.119s)
video 1/1 (12/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.119s)
video 1/1 (13/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.119s)
video 1/1 (14/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.128s)
video 1/1 (15/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.128s)
video 1/1 (16/334) F:\Topic\yolov5\runs\video\result2.mp4: 256x416 Done. (0.122s)
```

(圖 20：影片偵測檔)

	from	n	params	module	arguments
0		-1	3520	models.common.Focus	[3, 32, 3]
1		-1	18560	models.common.Conv	[32, 64, 3, 2]
2		-1	18816	models.common.C3	[64, 64, 1]
3		-1	73984	models.common.Conv	[64, 128, 3, 2]
4		-1	156928	models.common.C3	[128, 128, 3]
5		-1	295424	models.common.Conv	[128, 256, 3, 2]
6		-1	625152	models.common.C3	[256, 256, 3]
7		-1	1180672	models.common.Conv	[256, 512, 3, 2]
8		-1	656696	models.common.SPP	[512, 512, [5, 9, 13]]
9		-1	1182720	models.common.C3	[512, 512, 1, False]
10		-1	131584	models.common.Conv	[512, 256, 1, 1]
11		-1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13		-1	361984	models.common.C3	[512, 256, 1, False]
14		-1	33024	models.common.Conv	[256, 128, 1, 1]
15		-1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17		-1	90880	models.common.C3	[256, 128, 1, False]
18		-1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20		-1	296448	models.common.C3	[256, 256, 1, False]
21		-1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23		-1	1182720	models.common.C3	[512, 512, 1, False]
24	[17, 20, 23]	1	21576	models.yolo.Detect	[3, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]
Model Summary: 283 layers, 7068936 parameters, 7068936 gradients, 16.4 GFLOPS					

(圖 21：google colaboratory 訓練檔)



國立高雄科技大學(第一校區)資訊管理系
一〇九年度 實務專題 期末報告

人工智慧之影像辨識—口罩監控系統

組員：蕭仁豪、蕭煜宸、
黃仲偉、洪嘉聰

指導教授：莊文勝