

2025 年【科學探究競賽-這樣教我就懂】

□國中組 □普高組 ■技高組 成果報告格式

題目名稱：怕你覺得我太燒

一、摘要

隨著都市化的發展，火災的發生頻率愈加頻繁，常常會對人員和財產造成不可挽回的損失。例如，1995 年中港路上的衛爾康西餐廳大火、1977 年台中遠東百貨，以及 2021 年高雄城中城大樓的火災，這些事都凸顯了火災對生活帶來的危害。

二、探究題目與動機

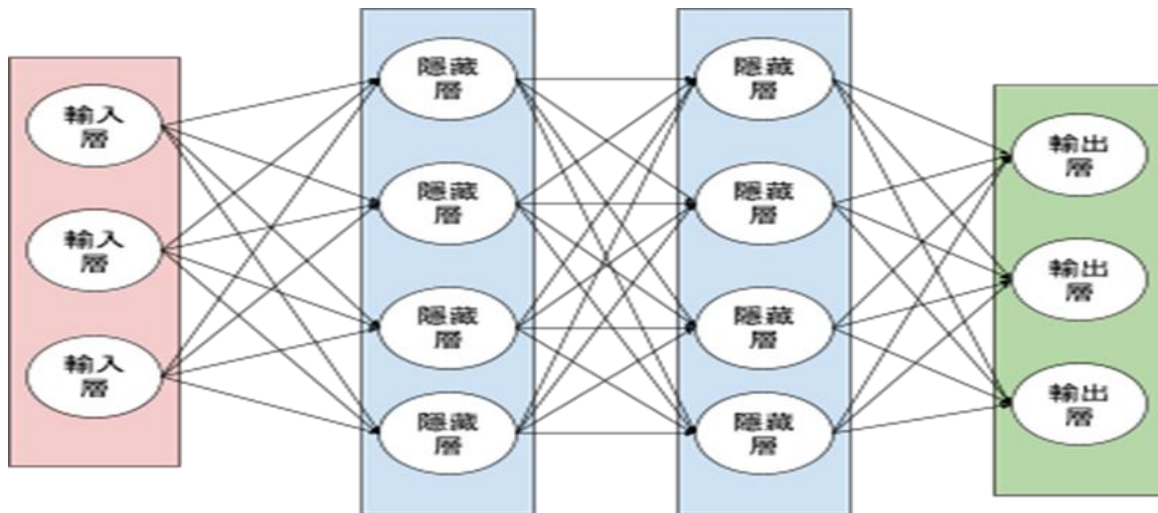
在現今這個便利又進步的時代，人們製作出了許多方便又好用的東西來提供給大眾做使用。普遍家庭都有的瓦斯爐、人手一顆的行動電源。這些方便的東西也帶來了一個不可忽視的風險，那就是火災。像是 2021 年高雄城中城大樓的火災、以及最近 2025 台中大遠百所發生的氣爆。而且在這個科技發達的年代，有許多精密的儀器不適合去接觸水源，如果我們在發生火災等其他緊急情況需要用到水來滅火時，傳統的大範圍灑水器就可能對這些機器造成破壞，所以不適用，於是我們做出了這款"小型安全消防系統"，這可以改善傳統大範圍灑水的缺點，這款消防系統會自動尋找火源，並只對失火處進行滅火的裝置，這樣不但可以靠自動尋找火源的方式來進行滅火，而且針對火源去進行滅火更可以保障其他的儀器不受破壞。

三、探究目的與假設

一、深度學習 (Deep

Learning)「神經網路是人工智慧中的一種方法，指導電腦以受人腦啟發的方式來處理資料。這是一種機器學習程序，稱為深度學習」(AWS, 2025)，深度學習是按照人類大腦設計的神經網路。人腦包含數百萬個互連的生物神經元，這些神經元會協同合作以學習和處理資訊。同樣地，人工神經元是稱為節點的軟體模組，其會使用數學計算來處理資料。深度學習神經網路（或人工神經網路），包含了許多人工神經元層，一同解決複雜的問題。深度學習用以下組合而成

圖一：捲積神經網路示意圖



圖一資料來源：自行繪製

二、卷積神經網路 (Convolution Neural Network, CNN)

卷積神經網路 (Convolution Neural Network, CNN) 一種最常見拿來分析圖片或影片的深度學習模型，通常由以下三個主要層次組成。

(一) 卷積層 (Convolution Layer)：「卷積就是在對圖片去做擷取特徵的動作，找出最好的特徵最後再進行分類」(GGWithRabitLIFE, 2018)。把圖片依照原本得像素去取得每一格的特徵顏色，並把特徵最多的圖片像速取得出來。

(二)池化層 (Pooling Layer)：針對捲積層篩選出來的特徵圖做降維，但依舊保留重要特徵，這樣可以防止機器過度學習

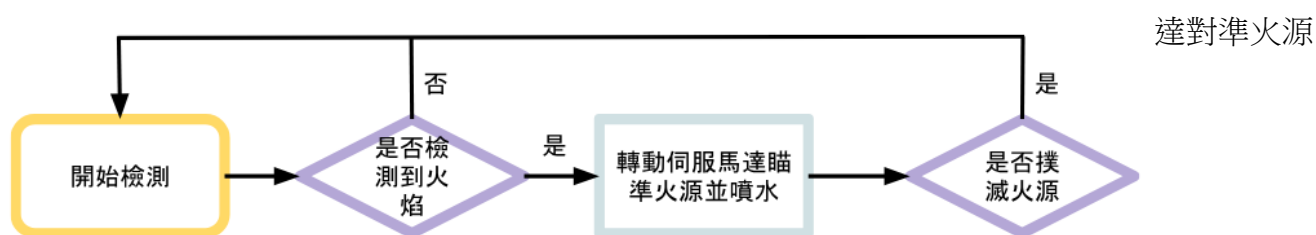
(三)全連接層 (Fully Connected Layer)：「全連接層就是最後的分類器，將前面所擷取出來的特徵，經過權重的計算之後，來辨識出這個所輸入的圖像到底屬於哪一個分類。」(wheam, 2020) 會把上面所精簡到最後的像素列成一排的特徵，如有圖片須檢測，就會拿上面的特徵條一一比對。

「如果要只用一句話不是十分精確地說明什麼是深度學習，可以把深度學習形容成一種比較深的類神經網路」(周秉誼, 2016) 深度學習通常擁有數以千計的參數在一個模型裡面，能夠用在目前頂尖的行業上，尤其是在需要處理大數據分析的領域。例如：

- (一) 視覺方面：醫學影像分析、安防監控和自駕車技術中
- (二) 語言方面：語音助手、機器翻譯和文本生成

四、探究方法與驗證步驟

一、架構



圖二資料來源：自行繪製

二、使用元件

(一) 火源感測器

一般是使用煙霧感測器來觸發警報。另一種類型是偵測火焰，這類型的傳感器用於短距離火災探測（Ceiling Tsai，2019）。火焰感測器利用紅外線對火焰非常敏感的特點，使用特製的紅外線接收管來檢測火焰然後把火焰的亮度轉化為高低變化的電平信號輸入到中央處理器根據信號的變化做出相應程式處理。

(二) esp32 cam

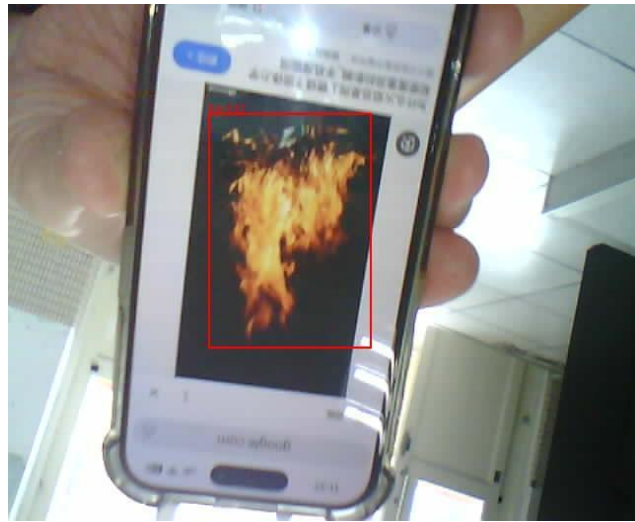
AI-Thinker ESP32-CAM 是一個以 ESP32 作為基礎的開發板，除 ESP32cam 本身內建 WIFI 及藍芽 BLE 之外，再加上 OV2640 兩百萬畫素相機模組、TF 卡（SD Card）、閃光燈模組，所組成的開發版。如圖四所展示、TF 卡（SD Card）、閃光燈模組，所組成的開發版，ESP32-CAM 上可以用鏡頭擷取影像如圖五、用網路傳輸照片、並用藍芽進行遙控，這些都可以在這一個小小的晶片中完成，重點是它也可以在 Arduino IDE 上做開發，而且語法跟 Arduino UNO 是一模一樣的（米羅學院管理員 2024 年 10 月 28 日）。

圖三：esp32 cam

圖四：esp32cam 偵測結果



圖三資料來源：自行拍攝



圖四資料來源：自行拍攝

五、結論與生活應用

一 研究結論

此研究採用了 Python 網路爬蟲技術與 Roboflow 平台來做收集和生成訓練所需要的圖片數據，並使用了 YOLOV5 訓練模型，來達到模型所需要的準確度。除此之外還搭建了一個簡易的伺服器來進行來運行此模型進行圖片的判別並把解果儲存在 Google 雲端，方便之後能有更多的樣本。我們還實現了即時通知和提醒功能，使用 Line Bot 使它能夠在圖片判別出火焰後透過 Line 進行通知。進一步的控制火災的發生。

```
@app.route('/detect', methods=['POST'])
def detect_fire():
    try:
        if 'image' not in request.files:
            return jsonify({'error': '請求中沒有圖片文件.'}), 400

        image_file = request.files['image']
        image = image.open().read().convert('RGB')

        results = model.predict(image)
        predictions = results.pandas().to_dict(orient='records')

        draw = ImageDraw.Draw(image)
        flame_detected = False
        labels_content = []

        for pred in predictions:
            xmin, ymin, xmax, ymax = int(pred['xmin']), int(pred['ymin']), int(pred['xmax']), int(pred['ymax'])
            label = pred['name']
            confidence = float(pred['confidence'])

            if label == 'fire':
                flame_detected = True

            img_width, img_height = image.size
            x_center = (xmin + xmax) / 2 / img_width
            y_center = (ymin + ymax) / 2 / img_height
            width = (xmax - xmin) / img_width
            height = (ymax - ymin) / img_height

            class_id = 0
            labels_content.append([class_id, x_center, y_center, width, height])

            draw.rectangle([xmin, ymin, xmax, ymax], outline='red', width=2)
            draw.text((xmin, ymin - 10), f'{label} {confidence:.2f}', fill='red')

        original_filename = image_file.filename if image_file.filename else 'processed_image.jpg'
        unique_image_filename = get_unique_filename(existing_image_files, original_filename)

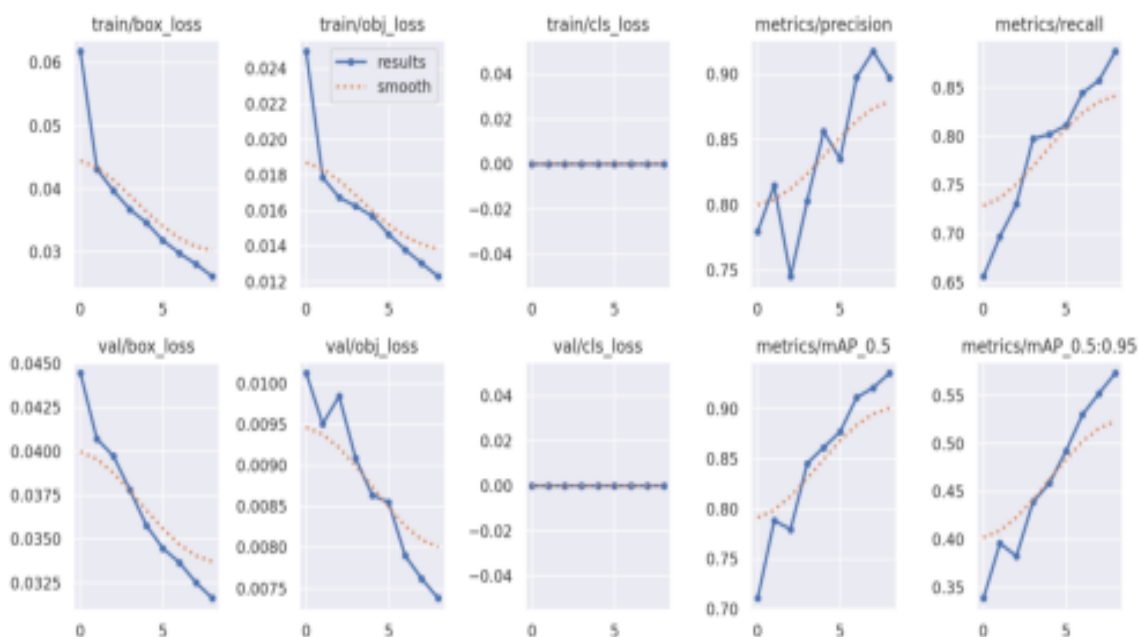
        if flame_detected:
            image.save(os.path.join(image_save_dir, unique_image_filename))
            unique_label_filename = get_unique_filename(existing_label_files, unique_image_filename.replace('.jpg', '.txt'))
```



（二）模型訓練

本研究採用YOLOv5模型，他是基於CNN並結合One-Stage組合而成。他擁有比CNN更加快速的判別，能夠在火災發生的時候更加快速的徵測到。圖七為訓練的線性變化圖，其中train/box_loss（訓練集的邊界框損失），預測的邊界框與標註的火焰框之間的誤差。train/obj_loss（訓練集的物件置信度損失），模型對火焰存在與否的信心錯誤(沒偵測到或是偵測錯誤)。train/cls_loss（訓練集的分類損失），模型只有分辨火焰(只有一種)所以為一直線。metrics/precision（精確率 Precision），偵測出的火焰中，有多少是真的火焰。metrics/recall（召回率 Recall），有火焰的情況下，模型能偵測出來的比例。val/box_loss（驗證集的邊界框損失），顯示與驗證集相比邊界框誤差。val/obj_loss（驗證集的物件置信度損失），與驗證集相比是否能夠區分物件和背景。val/cls_loss（驗證集的分類損失），模型只有分辨火焰(只有一種)所以為一直線。metrics/mAP_0.5（IoU 門檻 0.5 時的均值平均精度）門檻設為 0.5 時，計算模型的平均精確度。metrics/mAP_0.5:0.95（不同 IoU 門檻下的 mAP 平均值），計算 IoU 門檻從 0.5 到 0.95 的平均值

圖七：yolo v5 訓練指標



使用 Flask 建立一個 RESTful API，使 ESP32-CAM 透過 HTTP POST 方法將拍攝的圖片上傳到伺服器。使用 Flask-Ngrok 將 Colab 上的 Flask 伺服器公開到網際網路上。ESP32-CAM 將圖片以 multipart/form-data 的格式上傳到 Flask 伺服器的/detect 路由後，利用 PIL 模組將圖像讀取並轉換為 RGB 格式，使用訓練的模型對圖片進行推論並畫上邊界框，最後把有火焰的圖片儲存在雲端上面並發送警報訊息到 line 上面和 esp32cam。在 esp32cam 的上面，有裝蜂鳴器和一個閃光燈，在接收到火災的訊息後，蜂鳴器和閃光燈會啟動。

二 建議

（一）AI 偵測優化

1. 增加圖片的數量來增加精準度
2. 提高訓練次數減少誤判的機會

（二）伺服器優化

1. 強化 GPU 的效能減少每張圖片偵測的時間
2. 架設在主機上避免Colab每次啟動的時間限制和效能不足

（三）增加人性化操作介面

1. 製作手機App取代掉line bot的限制
2. 能夠實時看火災現場的影像

參考資料

AWS(Amazon Web Services)。什麼是神經網路？深度學習服務 <https://aws.amazon.com/tw/what-is/neural-network/>。

周秉誼（2016）。淺談 Deep Learning 原理及應用。技術期刊，第 0038 期，ISSN 2077-8813，國立臺灣大學資訊工程學系 https://www.cc.ntu.edu.tw/chinese/epaper/0038/20160920_3805.html。

GGWithRabitLIFE（2018 年 12 月 21 號）。[機器學習 ML NOTE] Convolution Neural Network 卷積神經網路 <https://medium.com/雞雞與兔兔的工程世界/機器學習-ml-note-convolution-neural-network-卷積神經網路-bfa8566744e9>。

wheam（2020 年 10 月 29 號）。DAY24 深度學習-卷積神經網路 Fully Connected Layer（全連接層）<https://ithelp.ithome.com.tw/articles/10251253>。

Ceiling Tsai（2019 年 8 月/29 號）。Arduino 筆記（61）：火焰感測模組 KY-23 <https://atceiling.blogspot.com/2019/08/arduino57-ky-23.html>。

米羅學院管理員（2024 年 10 月 28 號）。【ESP32-CAM 應用】第一篇：ESP32-CAM 燒錄程式 打造無線影像串流主機 ESP32cam。 <https://shop.mirotek.com.tw/iot/iot-start-2/>