

2025 圖形識別期末報告

題目：簡易型交通標誌辨識系統

繳交日期： 2025 年 6 月 5 日中午 12 時以前。

系所：電機系 級別：大學-四年級

姓名：張佑禕 學號：110511185

摘要：

本專題聚焦於自動駕駛系統中交通標誌之即時辨識應用，鑑於自駕車需即時判讀周遭環境中之交通標誌以輔助行車決策，特以台灣常見具幾何形狀的標誌（如正方形、正三角形、圓形與正八角形）作為辨識對象。配合公開資料集 TTSRB (Taiwan Traffic Sign Recognition Benchmark)，可有效進行模型訓練與驗證，降低資料蒐集成本。本專題旨在設計一輕量化之卷積神經網路 (CNN)，能部署於運算資源有限之車載嵌入式系統上，實現交通標誌於完整視野中的定位與分類，期望提升智慧駕駛系統之實用性與行車安全性。

一、主題描述：

（請同學從日常生活中，找一個可以將所學「圖形識別」技術應用上去的例子，並詳述挑選這個應用例的動機與想法。（20%）

在圖形識別技術的眾多應用場域中，自動駕駛系統可說是最具代表性且發展成熟的領域之一。自駕車必須能即時且準確地辨識周遭車輛的動向、交通號誌以及各類交通標誌，才能做出正確的行駛判斷。以台灣為例，常見的交通標誌形式包括正方形、正三角形、圓形與正八角形，這些具有明確幾何形狀的標誌便於進行邊界圈選與特徵擷取，進而提升辨識的準確性與效率。此外，目前已有公開資料集如 TTSRB (Taiwan Traffic Sign Recognition Benchmark)，可供進行模型訓練與驗證，降低資料蒐集的門檻。

本專題的發想動機即源自此實務需求，目標是設計一種輕量化且具即時運算能力的卷積神經網路 (Convolutional Neural Network, CNN)，可應用於運算資源有限的車載嵌入式裝置上。預期本系統能有效對行車視野中的交通標誌進行初步定位與辨識，作為智慧駕駛系統的一環，提升道路行車的安全性與自動化程度。

二、方法描述：

（請針對第一部分，清楚地寫下你所設計的識別系統、電腦模擬、及得到相對應之結論。（40%）

程式檔案連結：

<https://github.com/RogerChang424/RogerChang424-Taiwanese-Traffic-Signs-Recognition>

dataset 連結：

<https://drive.google.com/file/d/1EHZHgGK1aTW9ierIM1EoanMgxZUF1U3K/view?usp=sharing>

I. 系統設計

1. 資料集處理

本專題採用 TTSRB (Taiwan Traffic Sign Recognition Benchmark) 資料集作為模型訓練與驗證基礎。該資料集涵蓋 27 種台灣常見交通標誌類別，原始擴充訓練集 (train_plus 目錄) 中，各類別圖像數量約為 287 至 1200 張，分布不均。為減少模型將無標誌區域判定為有標誌，故隨機從 google Maps 台灣街景中，提取 287 張無標誌的圖像，創建第 28 個類別-無標誌，完整類別表如下：

0	Max. 40 (最高速限 40)	14	aware of traffic light(當心號誌)
1	Max. 50 (最高速限 50)	15	Min. 60(最低速限 60)
2	Max. 60 (最高速限 60)	16	aware of speed traps(當心測速)
3	Max. 70 (最高速限 70)	17	no right turns(禁右轉)
4	Max. 80 (最高速限 80)	18	no right/left turns(禁左、右轉)
5	Max. 90 (最高速限 90)	19	no left turns(禁左轉)
6	Max. 100 (最高速限 100)	20	no U-turns(禁迴轉)
7	Max. 110 (最高速限 110)	21	no parking(禁停車)
8	road separation(分道)	22	no entry(禁入)
9	right confluence(右匯入)	23	no temp. Parking(禁臨停)
10	left confluence(右匯入)	24	slow(慢)
11	car/motorcycle only(汽車與重機專用)	25	turn lights on(開大燈)
12	car only(汽車專用)	26	drive on the right(右側通行)
13	2-step left turn(兩段左轉)	27	none(無標誌)

為提升模型訓練的穩定性與類別學習的均衡性，本專題採用降採樣 (downsampling) 方式進行資料平衡：僅取用 train_plus 資料集的前 280 張，作訓練與一般測試用。將前 280 張(索引值介於 0-279)，依尾數區分，尾數為 0、1 複製進 balanced_test 目錄，供一般測試，餘下複製至 balanced_train 目錄，供訓練用。

最終驗證資料集，起初採用原資料包的 test 目錄，然而其不均衡的特性，不僅無法準確反映行車時，各標誌的常見程度，亦造成驗證結果的偏差。故改為採用 train_plus 資料集的第 281-287 張(索引值為 280-286)，以達成更高的均衡度。

2. 圖像前處理

為統一輸入尺寸並降低計算複雜度，所有輸入圖像在進入 CNN 前皆進行前處理。處理流程如下：

- 1) 尺寸縮放：將圖像縮放至 40x40 像素，提供固定尺寸給 CNN。
- 2) 隨機旋轉：只對訓練集旋轉，增加資料變化，以減少 over-fitting
- 3) 伸縮平移：進行縮放，將每像素值域由[0, 255]轉為[-1, 1]

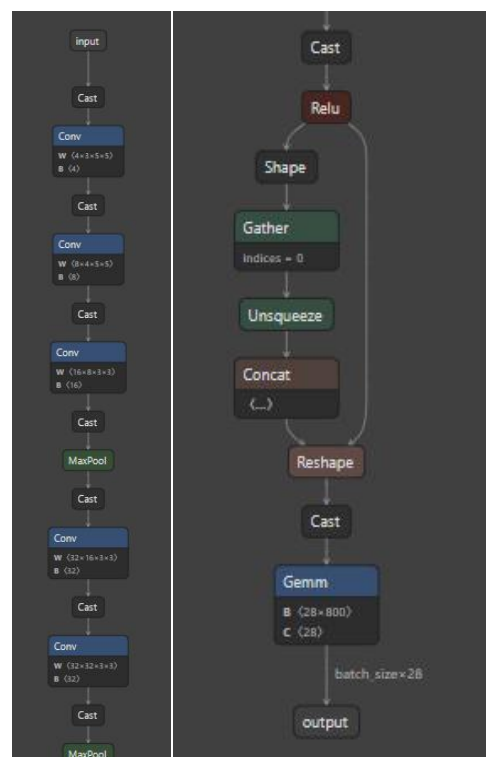
上述處理完畢的影像將轉換為 torch.tensor 格式，以供基於 PyTorch 架構設計之 CNN 模型進行訓練與推論。

3. CNN 模型設計

輸入層為一卷積層，有 3 個輸入通道，以接受處理後的 RGB 影像。為提取大特徵，kernel 尺寸為 5x5。

隱藏層包含兩組卷積層，每組有兩層，並以 2x2 maxpooling 接續，以減少运算複雜度與 over-fitting。其中第一層卷積層的 kernel 尺寸也設為 5x5，以提取較大特徵。

輸出層為全連接層，將 maxpooling 後的結果，經過 relu 運算、並改變張量形狀後，由此層輸出 28 個類別對應的相對信心值(未經 softmax 運算)。模型架構以 netron 視覺化如下：



4. 訓練策略

以下為本專案的訓練目標，與對應的嘗試方法：

1) 減少 over-fitting：

於 optimizer 中，加入 L2 regulation weight decay，強度為 $10^{-1.5}$ ，並將 learning rate 設為較高的值(0.001)。

2) 減少將無標誌判定為有標誌：

設定 cross-entropy loss 的類別權重，調低「無標誌」類別的對應權重至 0.2，使模型傾向輸出此結果。

3) 減少將較低的最高速限標誌，判定為較高的

設定 cross-entropy loss 的類別權重如下，以使模型接收到外觀相似的速限標誌，能傾向輸出較低的速限結果。

Max. 40	Max. 50	Max. 60	Max. 70	Max. 80	Max. 90	Max. 100	Max. 110
1	1.01	1.02	1.03	1.04	1.05	1.06	1.07

5. 應用端設計

在實際應用場景中，交通標誌往往僅佔整體視野中極小的區域。因此，如何將訓練完成之卷積神經網路（CNN）模型應用於完整輸入影像中，達成快速且有效的標誌定位與即時辨識，便成為本專題設計的核心目標。

考量目前多數車載攝影裝置採用的影像比例為 4:3，本系統選擇使用解析度為 360×480 的輸入圖像作為模擬基礎。在實作流程中，系統在第一階段，採用固定大小為 40×40 像素的視窗作為滑動區塊（sliding window），並以 20 像素作為步長（stride）進行掃描，以覆蓋整張輸入影像。在第二階段，滑動區塊尺寸改為 80×80，步長增為 40，以掃描更大面積的圖形，且在輸入進 CNN 模型前，會以 cv2.resize() 將尺寸縮放為 40×40。

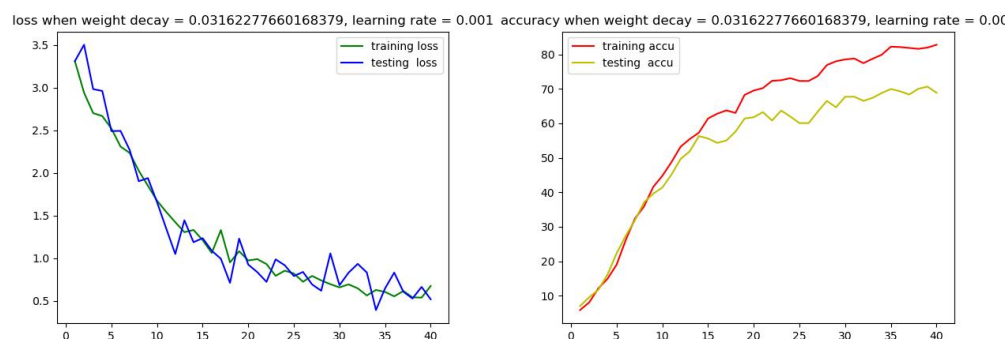
每次滑動視窗所擷取的區塊，將送入 CNN 模型進行分類推論。若該區塊分類結果中的最高信心值（confidence score）超過預設門檻，且分類結果不為「標誌」，該區域即被判定為「候選標誌區域」，並依照模型預測類別進行框選與標註。此設計可於即時影像中動態地定位與辨識潛在交通標誌，有效支援智慧駕駛系統之需求。

為強化系統的部署彈性與運行效率，CNN 模型將轉換為 ONNX（Open Neural Network Exchange）格式，使其能透過 OpenCV 的 DNN 模組（cv2.dnn）進行推論，取代原始 PyTorch 執行環境。此轉換不僅大幅減輕執行負擔，也提升模型於輕量化嵌入式裝置中的應用可行性。

II. 測試結果

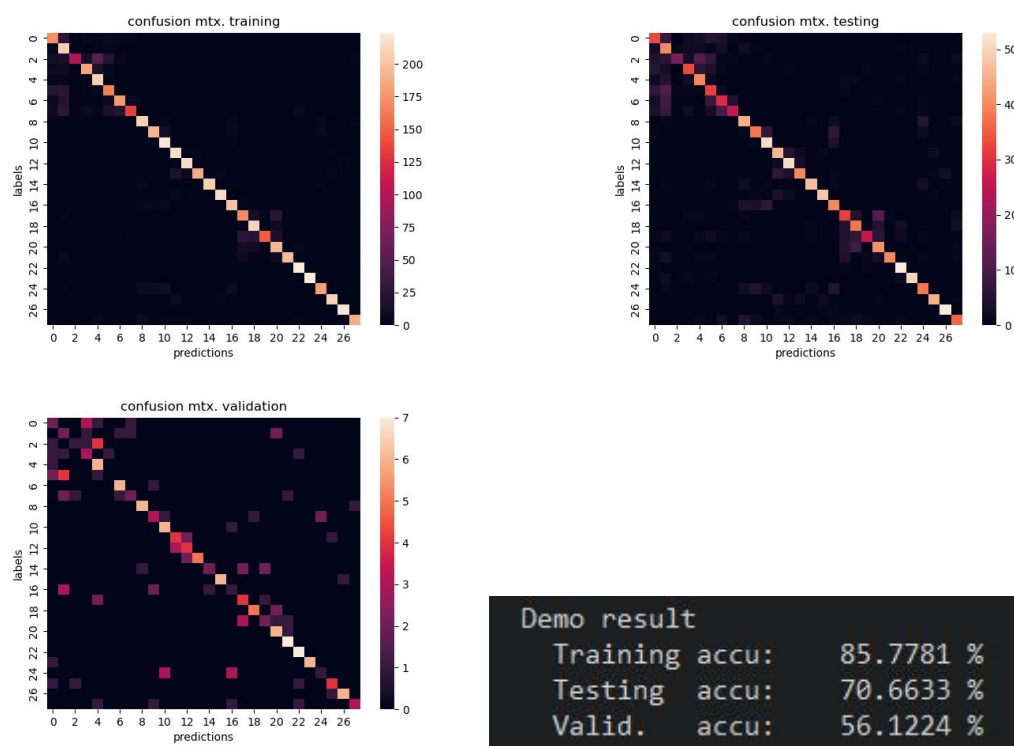
1. 模型測試數據

訓練時的 loss/accuracy 折線圖如下：



可觀察到 testing loss 未有增長情況，代表 over-fitting 並不明顯，且測試與訓練 accuracy 的差未達 15%。

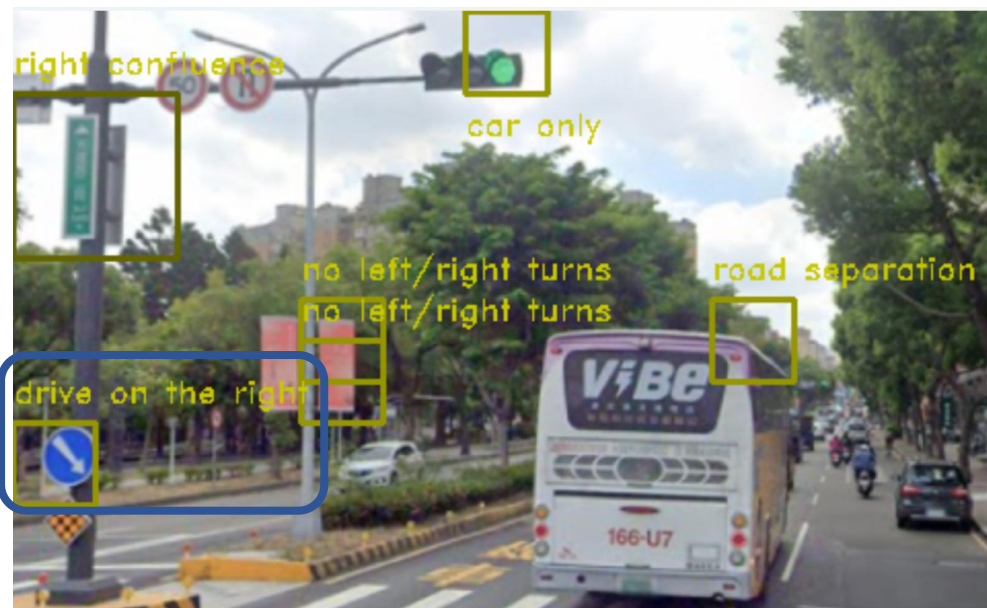
分別以訓練、測試與最終驗證集測試，得出以下 confusion matrix heatmap 與數據：



對於訓練時未見過的資料，模型仍有一定的泛化力。以 28 個類別而言，約 56% 的 accuracy，相對於 3.57% (隨機猜測的 accuracy) 已高不少，足以做出有意義的判斷。

2. 實際應用記錄

輸入兩張自 google map 擷取的街景圖。雖有將無號誌區判定為有號誌的狀況，但仍有正確識別出的標誌，如以下藍框圈選處。



三、 結論與自評：

(請針對第二部分所得到的結論，做個自評與未來期許(例如：對結果作評論，或列出更多可以考慮的因素或是瓶頸，讓所得之結論更加可靠且實用等等)。(30%))

1. Sliding window 的劣勢：

Sliding window 是獲取區塊邊界，最直觀的方式，易於程式的設計。然而，固定的區塊尺寸與步長，可能無法有效複雜的路況、器材解析度迥異的情況，此時可考慮 Region-Proposal CNN，將自動擷取區塊的功能整合進模型。

2. 誤報改善：

改進誤報，最直接的方法，即為拓展資料集。多數背景區塊，被誤判為標誌的主因，乃是「無標誌」類別的資料不足，致使模型無法應對複雜而多變的背景。同理，若有標誌無法被探測，也說明該類別的資料量不足，令模型未能掌握標誌特徵。

若僅能獲取「無標誌」類別的資料圖像，也能採用雙模型方案。先蒐集「無標誌」類別的圖像，至數量與其他類別總和相同。此時，可以創建兩個不同的資料集：「有無號誌」與「27 種號誌種類」，藉此訓練兩支模型。應用時，先將區塊內圖像輸入前者，若判定有標誌，則輸出予後者判定種類。

四、 附錄：

（請列出第一至第三部分中，所參考的相關文獻。如有其他資料無法在上面寫作中呈現，可以附錄形式列出。（10%）

[1] shivank856, "GTSRB - CNN (98% Test Accuracy)," *Kaggle.com*, Jan. 21, 2021.

<https://www.kaggle.com/code/shivank856/gtsrb-cnn-98-test-accuracy>

(accessed Jun. 02, 2025).

[2] exodustw, "GitHub - exodustw/Taiwan-Traffic-Sign-Recognition-Benchmark," *GitHub*, Jun.

22, 2022. <https://github.com/exodustw/Taiwan-Traffic-Sign-Recognition-Benchmark>

(accessed Jun. 02, 2025).