



工業技術研究院

Industrial Technology
Research Institute

限閱
RESTRICTED

DNN 車牌辨識技術與應用

林宏軒

2019.11.19

從車牌辨識看 影像處理與DNN的差異

車牌辨識

- 傳統影像處理的缺點：
 - 需要撰寫一連串的影響處理技術手法
 - 當車牌傾斜、模糊、光影變化等發生時準確率極低
 - 演算法無法自動學習，無法自動進步
- AI深度學習演算法的優點：
 - 僅需提供大量資料，不需撰寫演算法
 - 可自動學習處理車牌傾斜、模糊、光影變化等情況
 - 系統部署後，隨著收集資料的增多，準確度可一直學習提升

傳統影像處理手法：

- 灰階
- 二質化
- 找直線
- 將線條膨脹/侵蝕
- 找邊緣線條
- 找方框
- SVM學習篩選結果
- OCR數字辨識

AI深度學習處理手法：

- 決定AI模型
- 調整參數
- 提供資料自動訓練

DNN 車牌辨識

- 找出車牌(License plate detection)



YOLO



- 車牌轉正(License plate alignment)



Perspective
transform



- 車號辨識(License plate recognition)



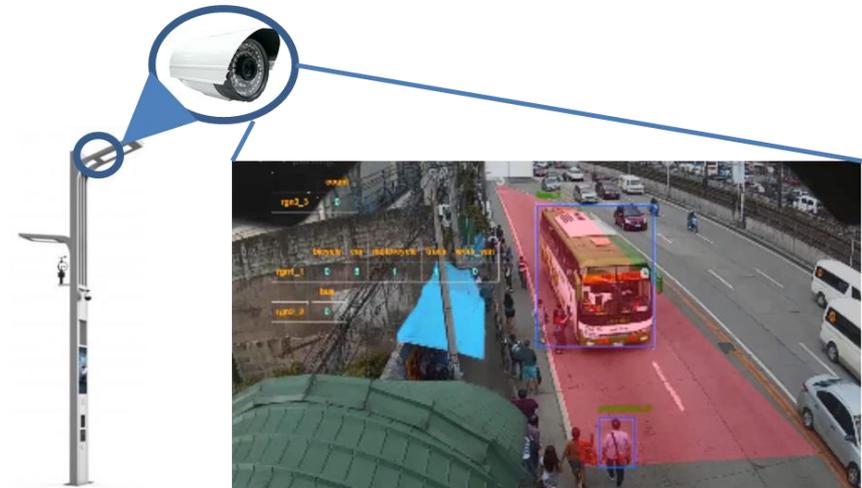
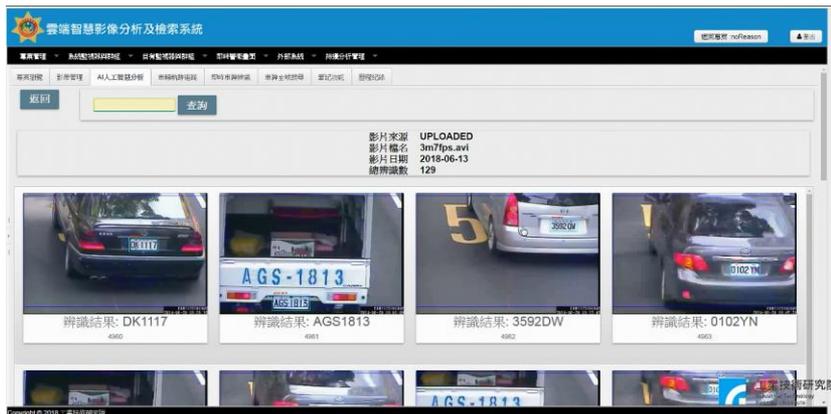
OCR

ARH-5335

車牌辨識應用

License Plate Recognition

- 車牌辨識技術可應用於：
 - 贓車自動偵測
 - 路口監視器、警員身上穿戴式裝置
 - 自動收費系統
 - 路邊停車格、定點式停車場、高速公路ETC
 - 科技執法
 - 路口監視器、路燈、公車行車記錄器
 - 出入口管制
 - 公司/政府單位重要出入口



License Plate Recognition

- 目前結果

- AI模型可以處理**360度**各種旋轉的車牌影像
- 斜角車牌辨識單張影像準確度達**95%**
- 若為多張照片車牌投票，準確度達**99%**以上



AAC-8326



Large View Angle Car Plate Recognition

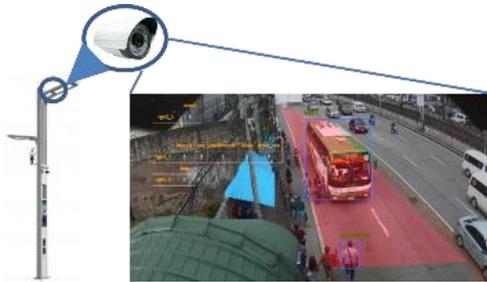
Made by: ITRI



Traffic Law Enforcement

- 以智慧路燈實現科技執法為例

- 偵測違規事項包括：紅線違停、公車格違停、跨越雙黃/白線、違規轉彎、闖紅燈等
- 困難點：距離遠、斜向視角、使用量大、耗電量需小



Intel NCS2

- Record 4K video
- Resize from 4K to 480p
- Object detection
- 11~13 fps



Intel GPU

- Judge illegal-event car
- Crop the illegal-event car region from 4K frame
- Find the license plate, alignment, and recognize
- 6~7 fps

Smart Highway

• 高速公路車牌辨識

- 目前結果：約30,000張影像訓練出 **98%** 準確度模型
- 困難點：準確度越精準越好，希望超過99%
- 目標：**99.2%** 準確度 (目前全世界最佳準確度)
 - ITS World Congress 2020 (ITS America)



ITS  AMERICA



Oct 4~8, 2020

從車牌辨識技術看AI訓練

車牌辨識流程

- 找出車牌 (License plate detection)



- 車牌轉正 (License plate alignment)



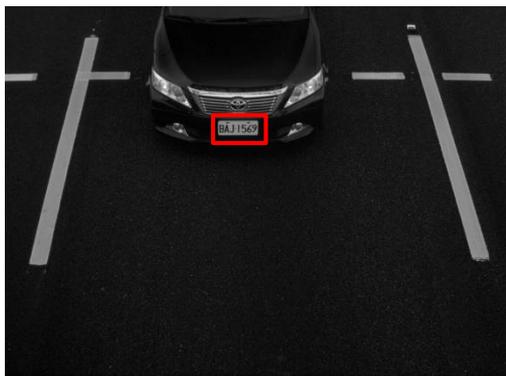
- 車號辨識 (License plate recognition)



5517PS

車牌辨識流程

- 找出車牌 (License plate detection)



- ~~車牌轉正 (License plate alignment)~~



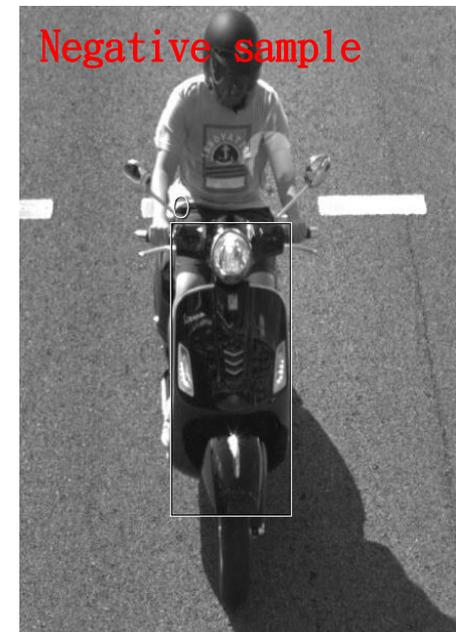
- 車號辨識 (License plate recognition)



BAJ1569

License Plate Detection

- 訓練資料量:
 - # of positive samples = 1,021
 - # of negative samples = 574
 - # of total samples = 1,595
- Model: yolov3
- Image size: 608x608
- Threshold: 0.2

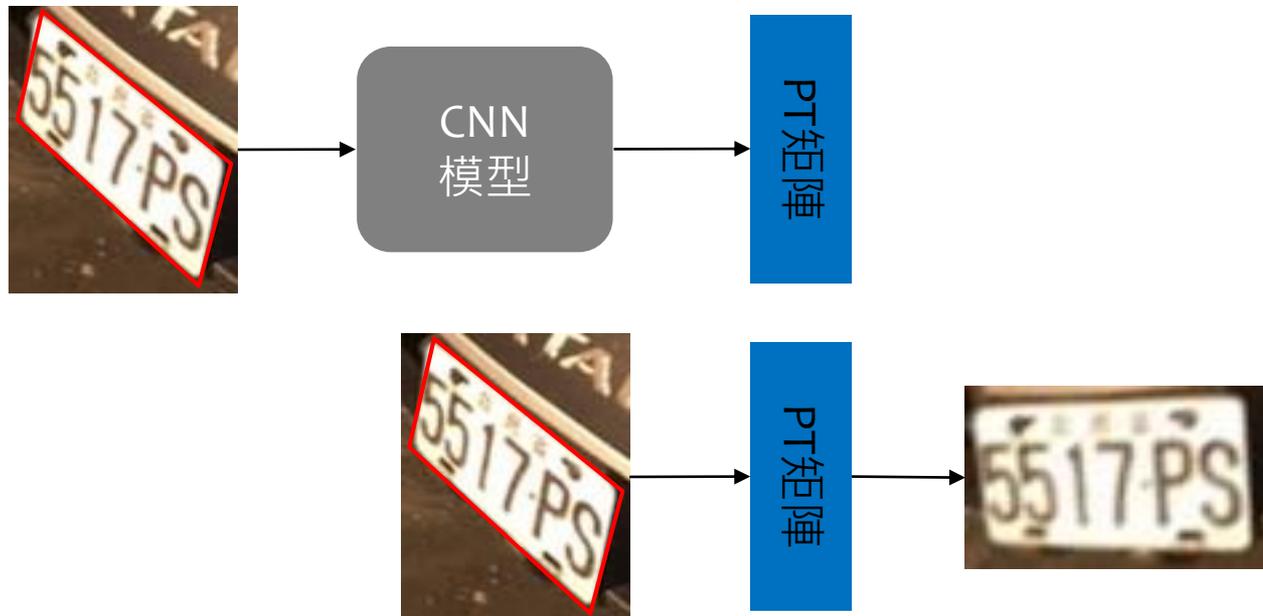


License Plate Detection

- 測試資料量:
 - # of testing data = 30,399
- 測試結果:
 - TP = 29,869
 - FP = 87
 - FN = 144
 - Recall = $TP/(TP+FN) = 29,869/(29,869+144) = 99.52\%$
 - Precision = $TP/(TP+FP) = 29,869/(29,869+87) = 99.71\%$

License Plate Alignment

- 訓練資料量：1600張
- 測試資料量：10萬張
- Framework：Keras
- 模型訓練：
 - 訓練CNN模型使其可預測校正車牌的Perspective Transformation (PT)矩陣



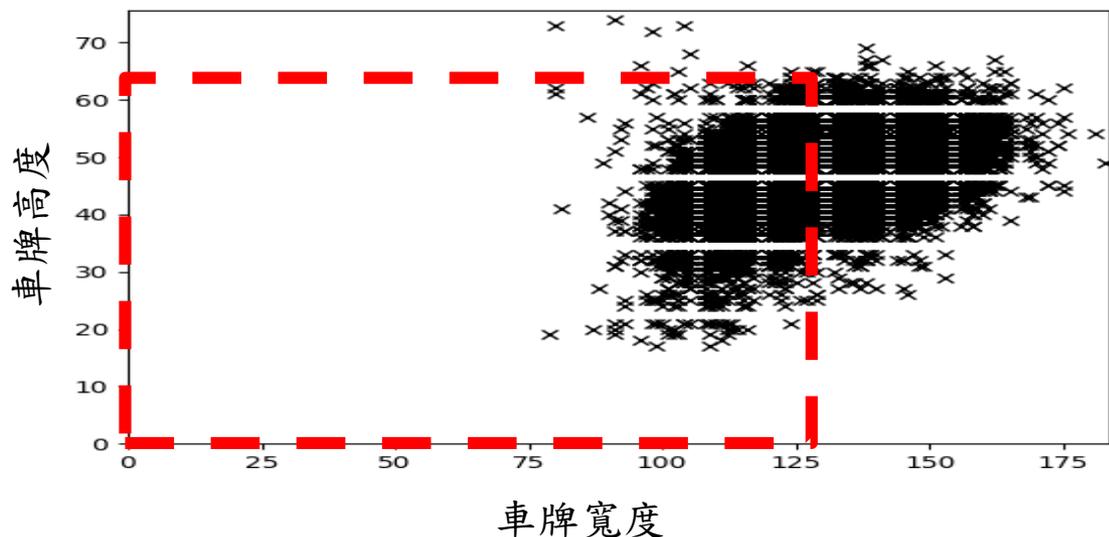
License Plate Alignment

- 測試資料量：全部10萬張
 - 轉正準確度：約98%以上



License Plate Recognition

- 訓練資料量: ~24,000
- 測試資料量: ~6,000
- Model: customized CNN
- 辨識結果包含[a-z][0-9][-][空白]，不包含中文
- 框出的車牌外圍各多取10%範圍，數量統計於下圖
- 車牌統一resize為高度64 pixels、寬度128 pixels(下圖紅框)

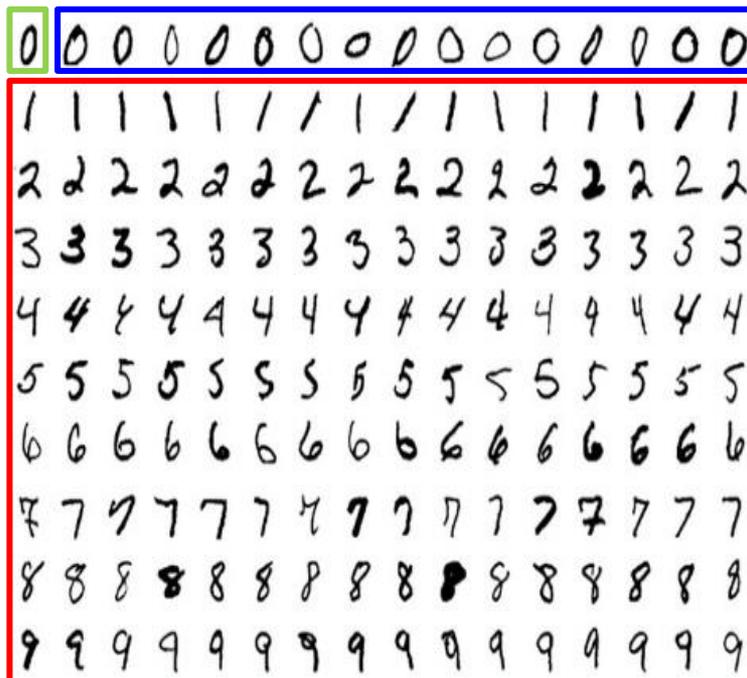


License Plate Recognition

- 測試資料量: ~6000張
 - 以車牌為單位的辨識率: 97.26 %
 - 以字元為單位的辨識率: 99.35 %
- 串接車牌辨識模型的整體結果：
 - 辨識準確度
 - 偵測車牌加車牌全對測試數據: ~95.0 %
 - 辨識效能
 - 車牌位置辨識 (608x608): 32ms
 - 車牌號碼辨識: 4ms

Single Layer Perceptron Learning

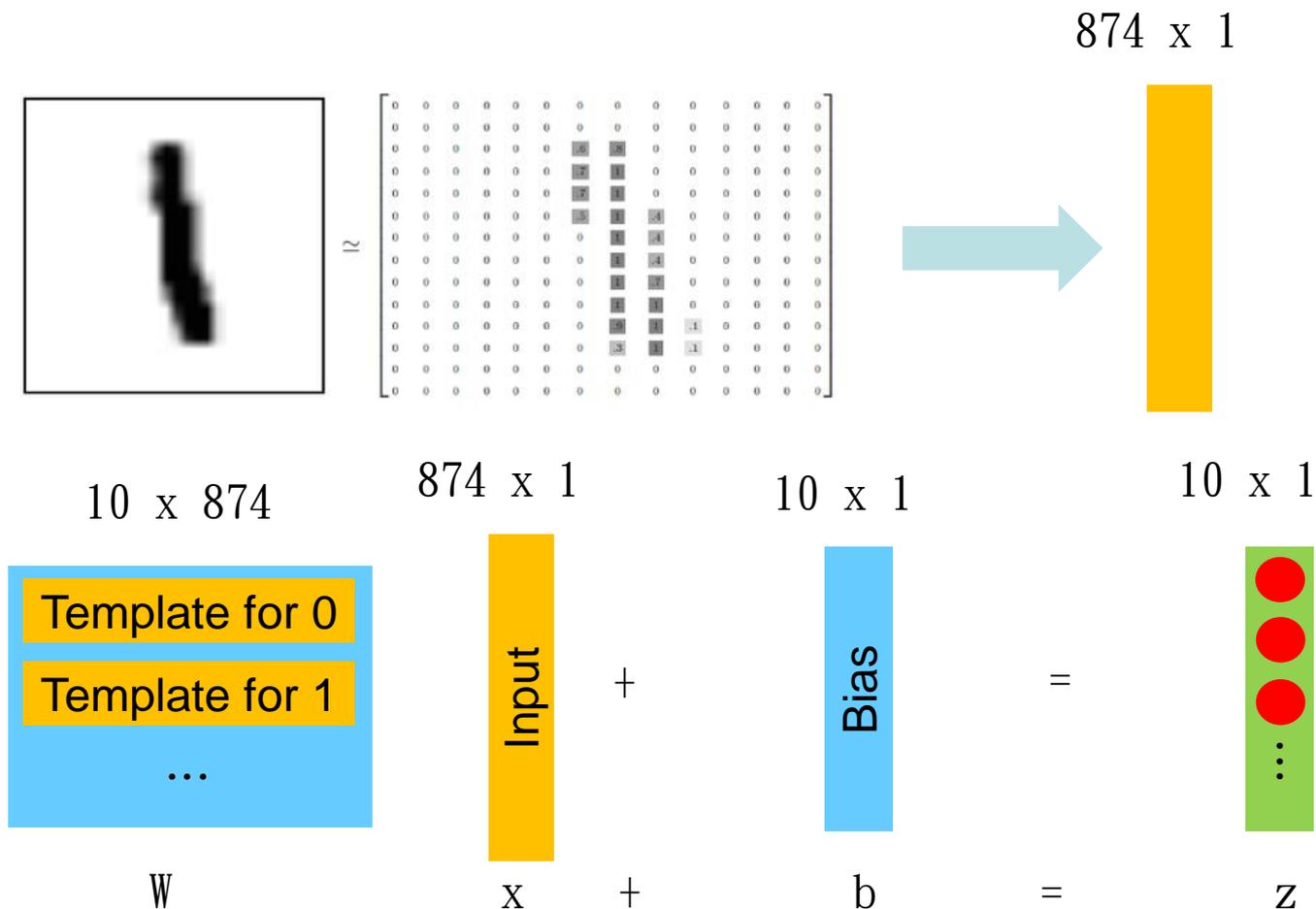
- 手寫數字資料集 MNIST dataset



- 抓兩個數字疊在一起：(template matching)
 - 若是兩個一樣數字，重疊部分多
 - 若是兩個不同數字，重疊部分少

Single Layer Perceptron Learning

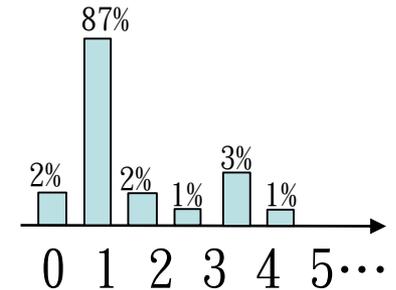
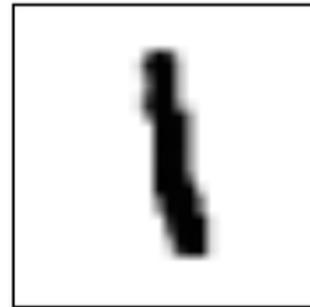
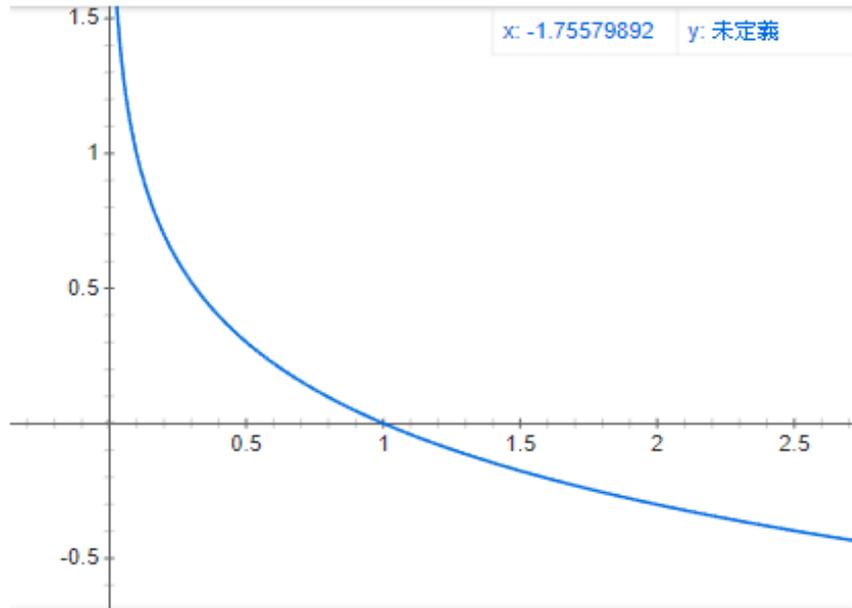
- 手寫數字資料集 MNIST dataset



Single Layer Perceptron Learning

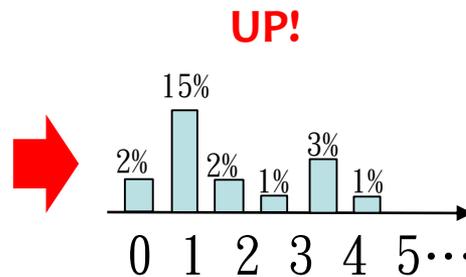
- 手寫數字資料集 MNIST dataset
- 如果數字是1，P1機率應該越高越好
- $\text{loss} = -\log(P1)$

$-\log(x)$ 的圖表

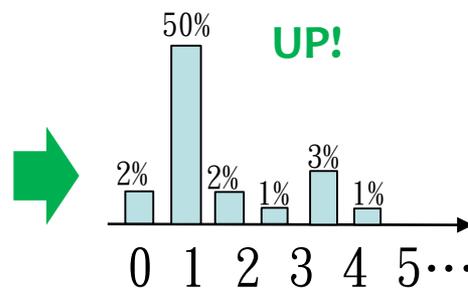
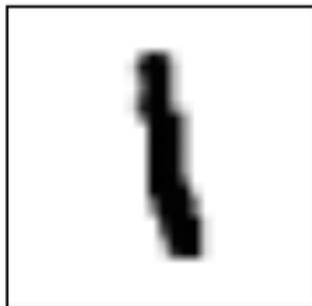


Single Layer Perceptron Learning

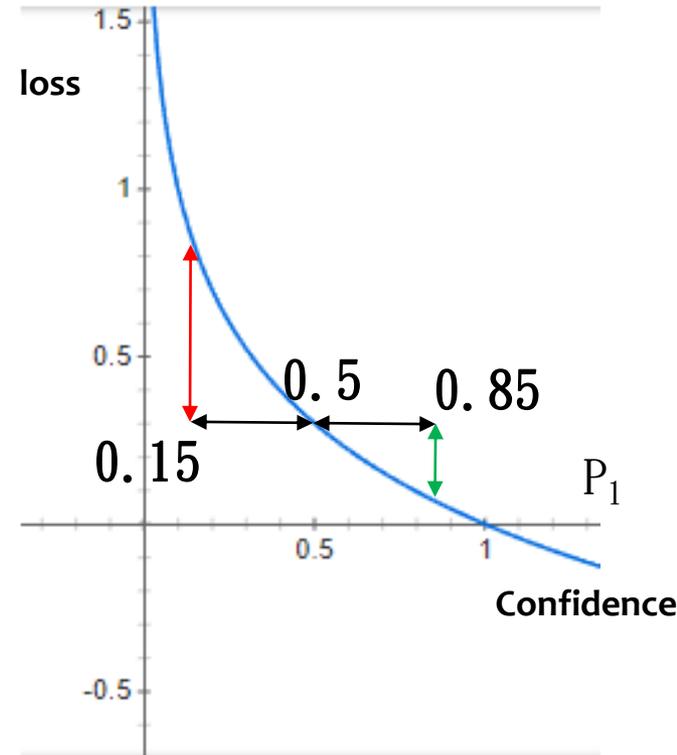
- 為什麼 loss function 使用 log 形式?
 - Loss function 的功能為將任務未完成的程度量化
 - 先求有再求好
- CASE 1: 從 15%(低機率)改善 35% **重要**



- CASE 2: 從 50%(高機率)改善 35%



$-\log(x)$ 的圖表

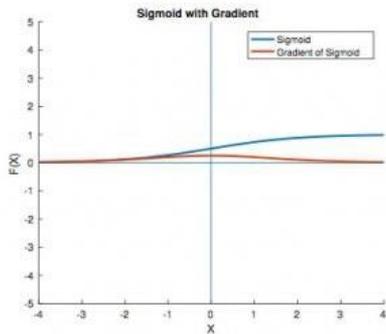


Activation Function

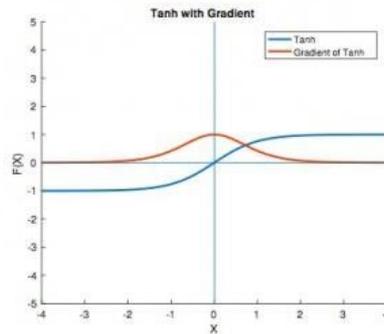
- 考慮一個多層感知學習模型，忽略bias
 - $Y = W_3 (W_2 (W_1 X)) = (W_3 W_2 W_1) X = W_{321} X$ (矩陣相乘之結合律)
 - 堆疊三層的model效果與只有一層相同，無”深度”學習
- 考慮一個多層感知學習模型，忽略bias，但加上activation function
 - $Y = W_3 f(W_2 (f(W_1 X))) \neq W_{123} X$ if $f(x) \neq x$ (包含non-linearity)
- 多層堆疊model可以比只有一層的擁有更強的表達能力，能學得更多

Activation Function

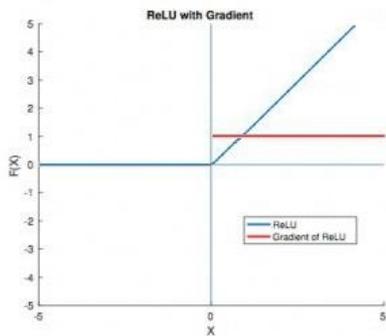
- 有多種Activation Function，如Sigmoid、Tanh、ReLU等
- 近期比較常用的為Leakly ReLU，為ReLU的變型



(a) Sigmoid



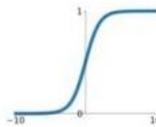
(b) Tanh



(c) ReLU

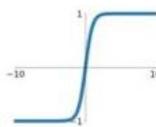
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



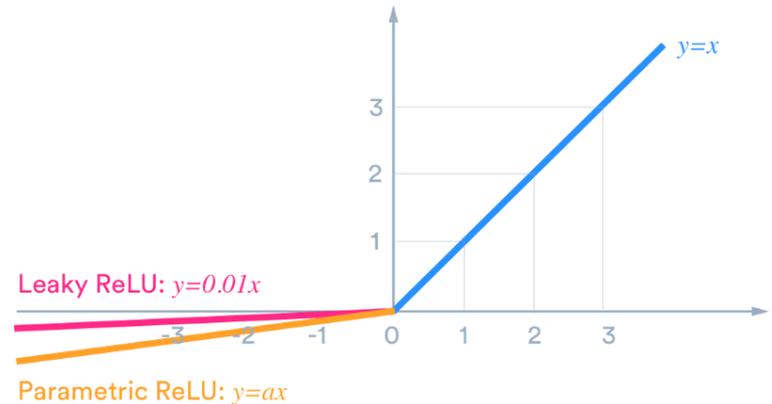
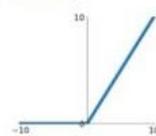
tanh

$$\tanh(x)$$



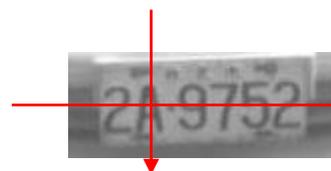
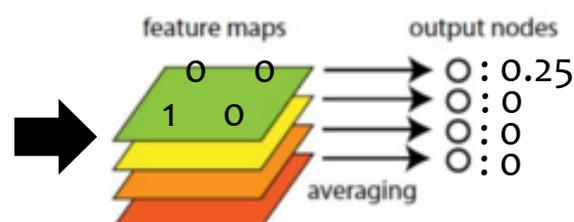
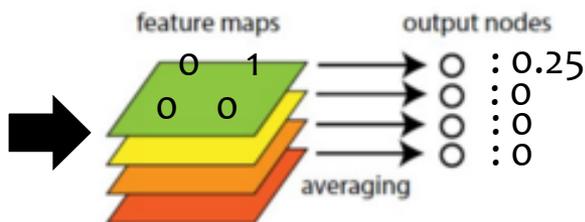
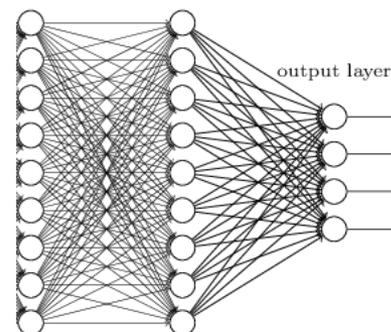
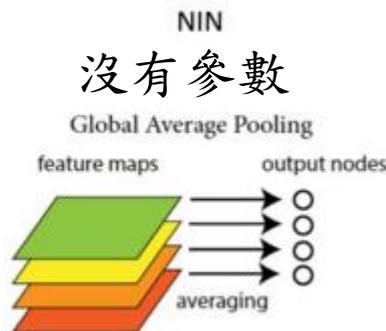
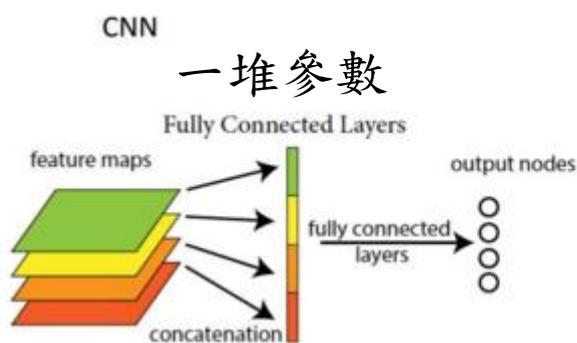
ReLU

$$\max(0, x)$$



Global-Average-Pooling

- 在不需明確知道物件位置時，可取代全連接層
 - 大幅減少參數



X方向：多個字元，不適用GAP

Y方向：單個字元A，適用GAP

國際車牌辨識

國際車牌

奈及利亞
單排、背景圖



越南車牌
有單排、雙排



來往香港/大陸兩地
一台車兩張車牌



上方香港、下方大陸

馬來西亞
有單排、雙排



大陸車牌
極大部份單排



國際車牌

美國車牌



干擾



直排文字



間隔

馬來西亞車牌



單排車牌



雙排車牌



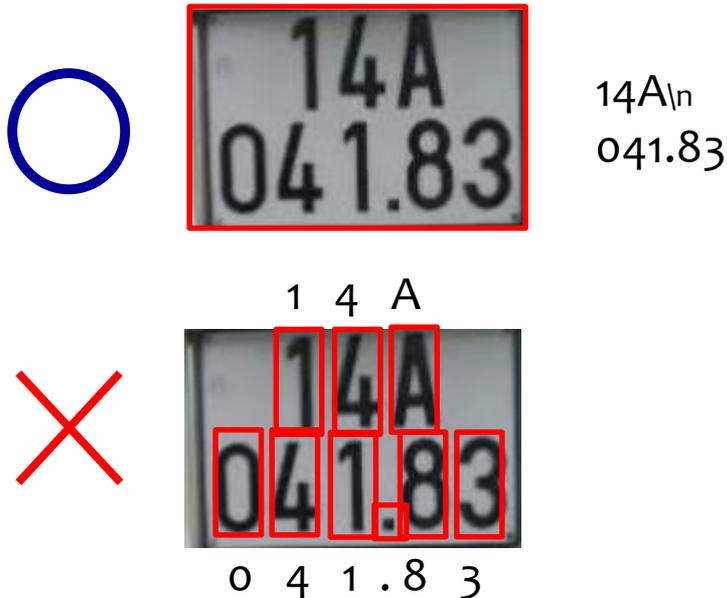
特殊車牌 RIMAU開頭



特殊車牌 地名PUTRAJAYA開頭

目標1：整張車牌的標記

- 痛點：使用者習慣的標記為整張車牌
- 以整張車牌的車號標記做為模型訓練資料集
 - 不需車牌號碼逐字標記，為人常用的車號輸入方式
- 與台灣停管處路邊停車車號記錄方式相同
 - 若有管道取得國外相關資料，則可直接訓練



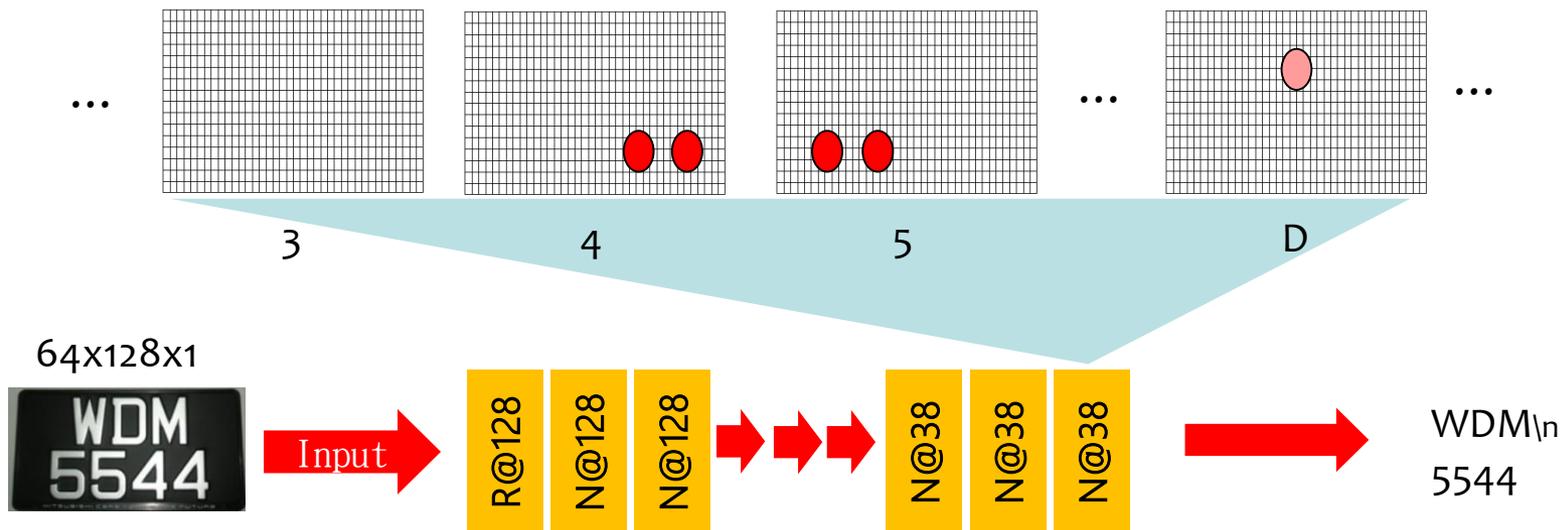
路邊停車收費員拍照開單

目標2：適用於各種格式

- 痛點：車牌多樣化
- 透過模型可視化技術，找出車牌字元位置
- 預計建構國際車牌模型
 - 利用模型中間層的特徵值
 - 判斷各種可能字元位置+字元種類
 - 利用每種字元的classification，產出loss function以校正模型



模型可視化範例



目標3：資料增量

- 痛點：國外車牌資料難取得
- 研發適用於車牌的資料增量、資料合成技術
 - 先以簡單的合成處理產生合成資料初版
底圖白色、文字黑色



- 使用傳統影像處理，包含抽換顏色、透視轉換等增加難度
(傳統影像處理可以做得不錯的部分，就先處理)
處理手法會留下一些參數(文字與底圖顏色RGB、旋轉角度的範圍等)

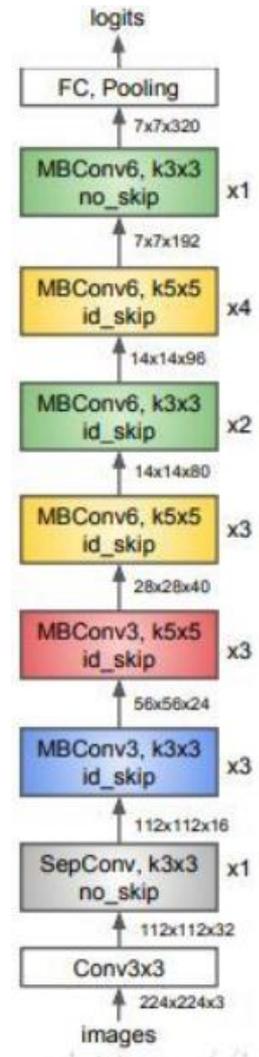


- 最後套用CycleGAN技術，由模型自動學習補上真實場域的Noise



目標4：整合NAS結果

- 痛點：手工設計模型常常不夠優化
- 基於Network Architecture Search (NAS)模型自動搜尋的非制式車牌模型架構
 - 模型參數空間不浪費，同等參數量下的準確率大幅優化
- Google的MnasNet
 - 支援在Mobile上的小型模型架構最佳化
- Google的EfficientNet
 - 支援在Server端的7種大小模型架構最佳化
- 基於上述NAS的子架構結果，堆疊建構工研院專屬車牌模型
 - 以EfficientNet建構適用於國際車牌的辨識模型



MnasNet

國際車牌小結

- 建構適用於國際車牌的DNN辨識模型
 - 辨識國際車牌的各種可能樣貌
 - 單排文字、雙排文字、單排混雙排文字
 - 非車牌文字但在車牌上(干擾文字)
 - 僅需提供整張車牌的標記，即可自動辨識字元及其位置/順序
 - 套用Google的模型架構自動搜尋結果，優化模型
 - 導入適用於車牌的自動增量技術
 - 降低所需資料量



結論

- **DNN車牌辨識**
 - 優點為具有極高的成長性，缺點是需要大量場域資料
 - 適用於需要極高精準度的場合，或用於各種極端情況如斜角、光影變化等情境
- **傳統影像辨識車牌**
 - 適合系統架設初期使用
 - 若後期發現準確度不足，再套用DNN解決方案
- **工研院DNN車牌辨識**
 - 以找出車牌、車牌轉正、車號辨識三個模型串連
 - 正面視角車牌辨識準確度達98%以上
 - 斜向視角車牌辨識準確度達95%以上
 - 即使肉眼難以辨識的車牌影像亦有機會辨識出來