

2021 Synopsys ARC 盃 AIoT 設計應用競賽 決賽作品

駕駛風格回饋系統

Driver Feedback System for Steering Style

報告人- 蕭又瑜、彭恩宇

7/25/2021



Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

作品概述

• 目標

- 將人工智慧部署在終端進行推論
- 分析實時的駕駛數據
- 協助駕駛人了解自身駕車風格
- 改善駕車技術、培養安全駕駛行為
- 提升駕車效率（如節省油耗）



作品概述

• 使用情境

- 將裝置安裝於汽車前擋風玻璃或機車龍頭
- 系統會在行駛過程針對各項駕駛行為評分
- 旅程結束後顯示全程駕駛各項評分

• 評分項目

- 起步
- 加速
- 煞停
- 巡航
- 轉彎
- 以五星等級評分以上項目



Agenda

- 作品概述
- **難點與創新**
- 設計與實現
- 作品進度
- 測試結果
- 總結展望

難點與創新

• 難點

- 基於車種、排氣量、懸吊裝置等差異，會使不同車輛在同樣的行駛狀態時產生不同的數據，不容易用同一個模型應用在不同車種
- 有些分類項目之間的差異較小，模型分類的準確率和信心較低
- 有些分類的數據遠遠不足另一類，如機車需要兩段左轉，導致左轉數據非常少
- 收集訓練模型的數據集時，資料本身不一定乾淨（如收集車輛起步的資料時，可能也會記錄到怠速與巡航時的數據），需要再手動去除混雜的數據
- 當開發板擺放角度歪斜時，可能影響模型的判斷結果
- 搜集資料與整理資料需要大量時間
- 需要找到封閉場地測試
- 調試時間漫長，必需真的騎車出門才知道效果如何

難點與創新

• 創新

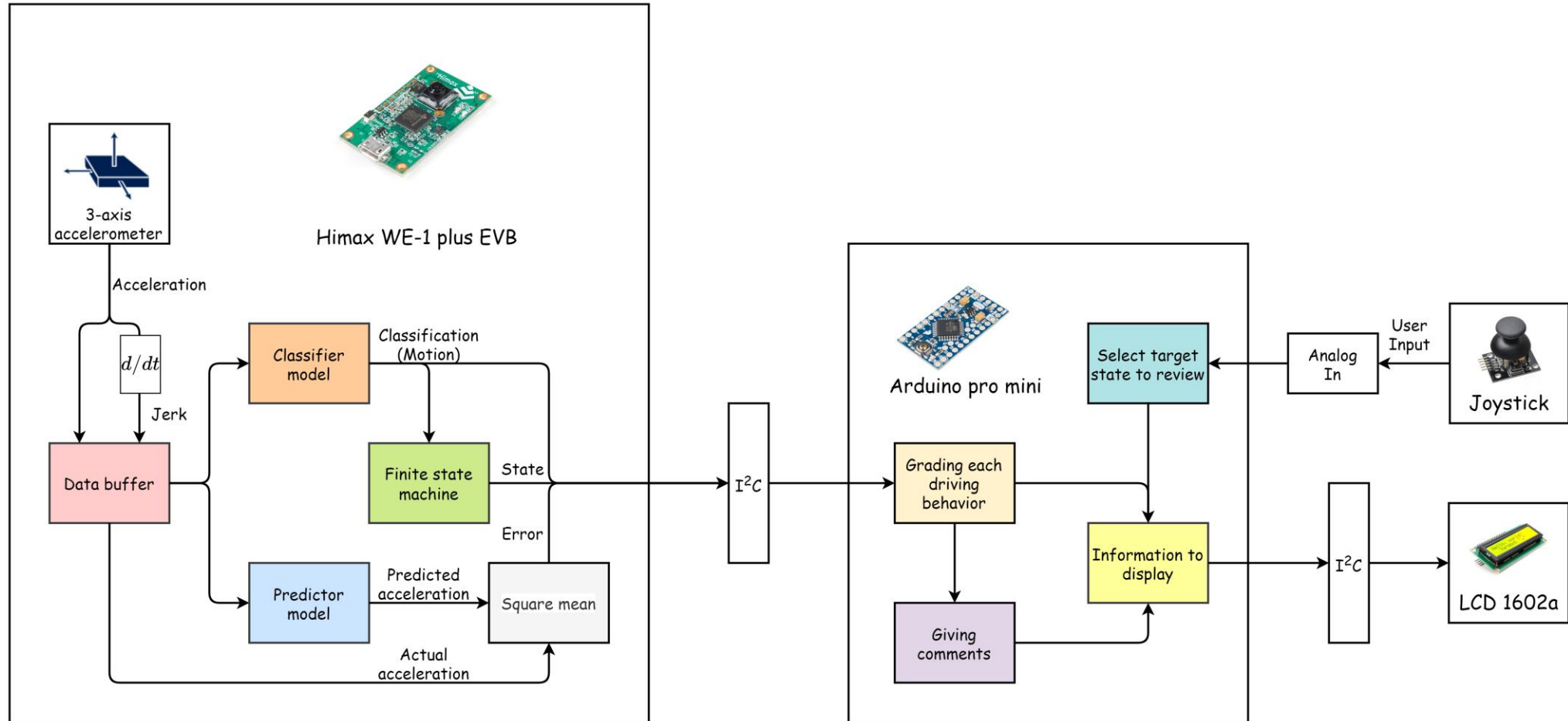
- 即裝即用的可拆卸式裝置
- 智慧提醒駕駛人操駕習慣，進而達到高效率低油耗的駕駛表現
- 用五星等級評量駕駛人操駕行為，並提示可改進的駕車習慣
- 更客觀的評價操駕風格
- 協助駕駛新手更快提升駕駛技巧的裝置
- 讓人工智慧來提醒駕駛，乘客不必開口，不會傷感情

Agenda

- 作品概述
- 難點與創新
- **設計與實現**
- 作品進度
- 測試結果
- 總結展望

設計與實現

• 主要架構



設計與實現

• 硬體

- 以Himax WE-I Plus開發板進行開發
 - 使用開發板的加速度計紀錄三軸加速度
- 以I²C連接Arduino作為子系統，處理評分、評語並建立使用時的人機操作介面，處理使用者的輸入與輸出
- 以搖桿的操作作為使用者的輸入介面
 - 直覺、易操作
- Arduino以Digital Pin模擬I²C Master連接LCD作為顯示的輸出介面
 - 由於Arduino接收Himax開發板之資料，已作為I²C Slave
- 使用Micro USB為系統供電（Arduino與Himax WE-I Plus）
 - 考量到現代車輛上通常有USB的插孔
- 整合所有裝置到電路板上，美觀且方便攜帶與使用

設計與實現

• 軟體 (Cont'd)

- 利用Tensorflow Functional API設計兩個CNN (Convolutional Neural Network) 模型，分別為Classifier與Predictor
 - Classifier以連續時間內紀錄的資料來推論、分辨車輛的駕駛行為
 - Predictor則以相同的輸入推論、預測下一刻正常操駕下車輛行駛的可能數據
- 將Tensorflow模型轉換成Tensorflow Lite的模型並量化，再將Tensorflow Lite模型轉為.cc檔
 - 量化後模型縮小，且推論速度更快
- 將兩個模型佈署到Himax WE-I Plus開發板，以Pipeline的方式逐一進行推論
- 撰寫C/C++程式做模型輸出值的後處理與整體駕駛狀況分析，主要有三個子程式，分別為：
 - MotionDetector
 - StateMachine
 - ErrorEvaluation

設計與實現

• 軟體 (Cont'd)

- MotionDetector是將Classifier模型各類信心輸出，記錄一小段時間（約0.36秒），取其平均，並找出最大的類別、檢查是否高於設定之信心閾值（設為75%）
 - 信心足夠則回傳類別（Classification或稱Motion）
 - 不足則回傳Unknown
- StateMachine是將MotionDetection之回傳值作為狀態機轉換狀態 (Transition) 的依據，當同一個Motion連續出現 7 次後則觸發Transition
 - 共16種狀態：「怠速 (Idle)」、「起步 (Start-off)」、「巡航 (Cruise)」、「煞車 (Brake)」、「左轉 (Left)」、「右轉 (Right)」與「加速 (Accel)」7種各有Strong與Weak兩個狀態，再加上「初始狀態 (Initial)」與「等待 (Pending)」
 - 當駕駛行為不穩定時，狀態機會在Weak的狀態間跳轉，當一個Motion連續觸發兩次Transition後則進入Strong的狀態，若又遇到不穩定時，則會先跳回同狀態下的Weak狀態，再等下次
 - 狀態前送 (State Forwarding)：設計Transition時，考量到駕駛行為的連續性，如怠速後通常為起步、或是煞車後常接怠速，這種情況下則不會再跳回原狀態的Weak State，而是直接跳下一狀態的Weak State

設計與實現

• 軟體 (Cont'd)

- ErrorEvaluation會記錄每一個時間點下，實際偵測到的加速度與上一個時間點的預測值的差值為「誤差」，當狀態機進入某一類的Strong state後，則開始計算此狀態下誤差之平方和，當跳離此State後，計算其平均並回傳 (Mean Square)
 - 由於都是以正常操駕之數據訓練Predictor，故可以將其理解為「這個狀態下的數據與正常操駕之數據的誤差」或是「這個狀態有多像正常之操駕？」
 - 「起步」、「巡航」與「加速」計算前後與上下方向的 Mean Square Error的和
 - 「左轉」與「右轉」計算前後與左右的 Mean Square Error的和
 - 「巡航」計算三軸 Mean Square Error的和
- 撰寫Arduino程式與Himax WE-I Plus開發板進行通訊
 - 以I²C接收Mean Square Error、Motion與State三個資訊，再作下一步的處理
- 以Arduino程式實作評分機制並處理人機介面
 - 如顯示狀態資訊、偵測到的Motion與實時評分
 - 偵測使用者輸入的，給予操作回饋

設計與實現

• 軟體 (Cont'd)

– Arduino評分系統之評分機制

- 以Himax開發板傳來之Mean Square Error作為評分依據
- 先上路搜集一般行駛時各狀態Mean Square Error的數據，計算出各狀態的平均 (μ_{state}) 與標準差 (σ_{state})
- 假設Mean Square Error為常態分佈，評分時以下式正歸化給定輸入

$$\overline{MSE_{state}} = (MSE_{state} - \mu_{state}) / \sigma_{state}$$

- 再將此正歸化的值取負號（愈小愈高分）傳入Sigmoid函數，設計當其與平均值差超過2個標準差後分數接近0或1（當Sigmoid函數x值大於5或小於-5後的y漸進值）

$$y = Sigmoid\left(-\frac{\overline{MSE_{state}}}{2} \times 5\right)$$

- 將Sigmoid函數的回傳值 (y) 乘以5即獲得5星等級

$$Star = 5y$$

設計與實現

- 模型(分為**Classifier**與**Predictor**)

- Classifier

- 根據過去30筆的資料（加速度與加速度的微分）判斷目前駕駛行為

- 分類項目：

1. 閒置
2. 巡航
3. 加速
4. 煞車
5. 左轉
6. 右轉

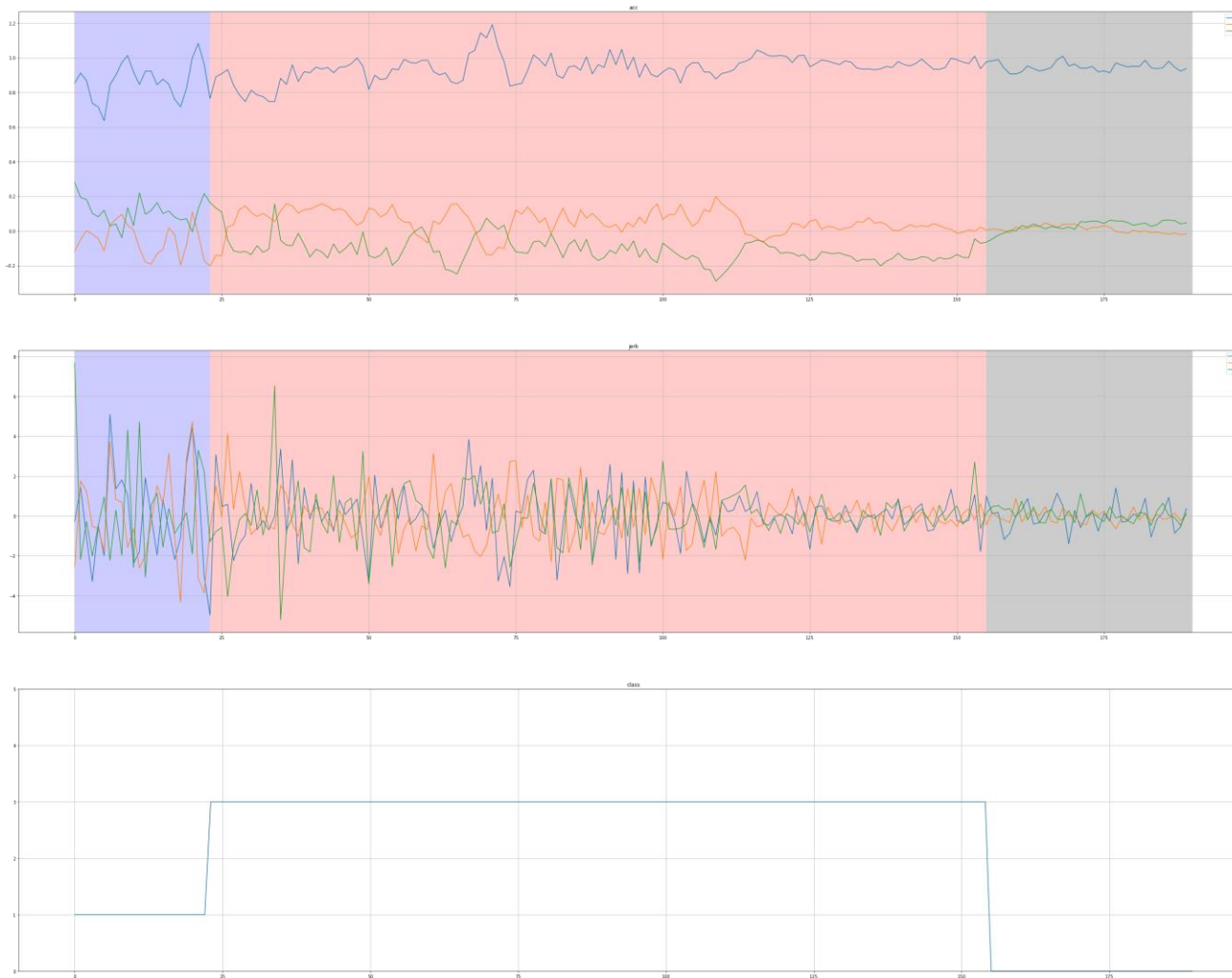
- Predictor

- 根據過去30筆的資料（加速度與加速度的微分）預測接下來車輛的三軸加速度數值

設計與實現

• 資料處理

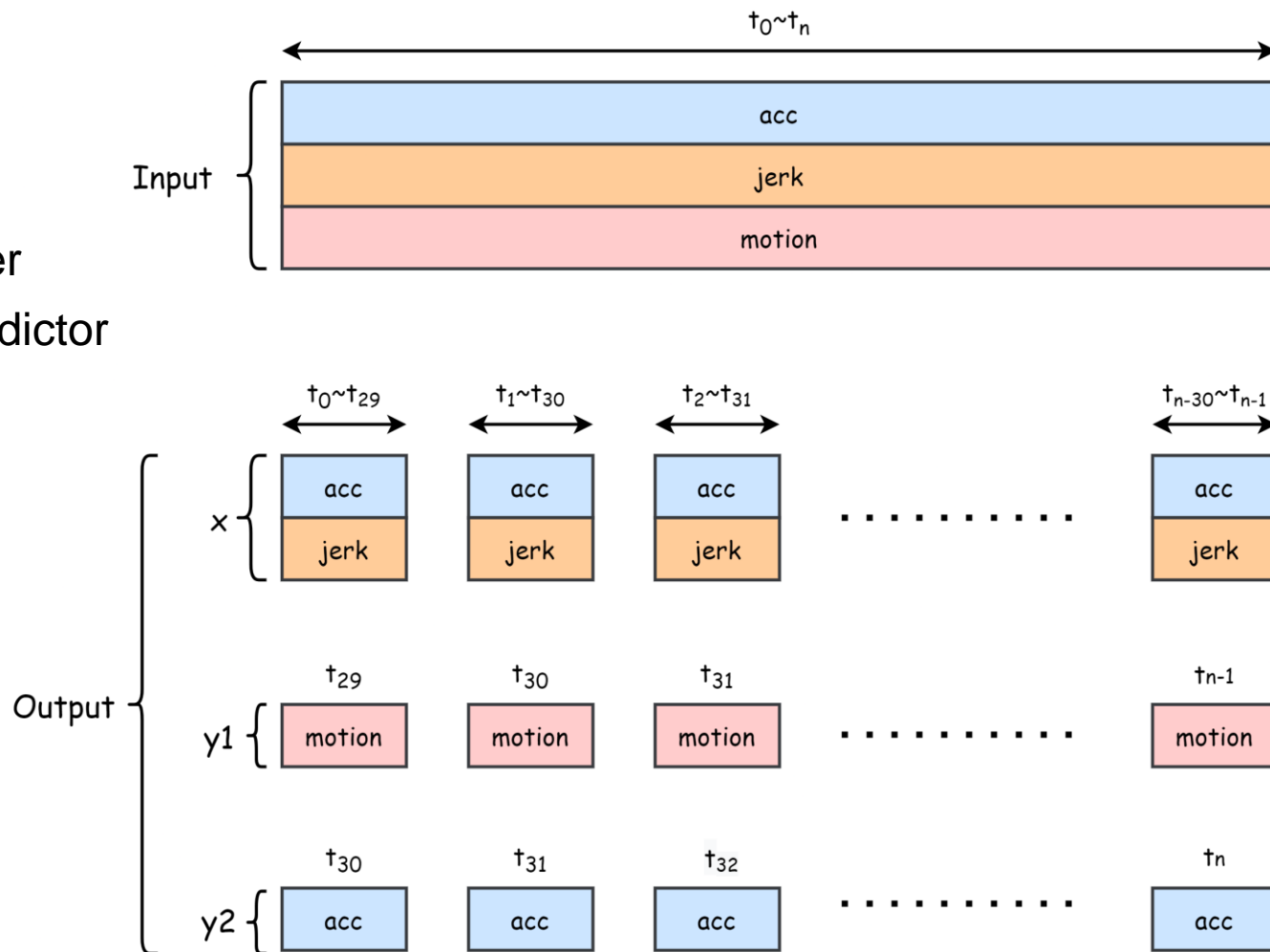
- 使用資料(每40毫秒紀錄一次)
 - 加速度
 - 加速度的微分(jerk)
 - 手動標記六種分類項目
- 前處理
 - 分別訓練集和驗證集
 - 依據六類項目區分資料
 - 正規化資料
 - 將資料乘上旋轉矩陣進行擴增



設計與實現

• 資料處理 (cont'd)

- 產生訓練資料
 - 產生長度為30的時間序列x
 - x時間內的分類為y1，用來訓練Classifier
 - x之後的下一筆資料為y2，用來訓練Predictor

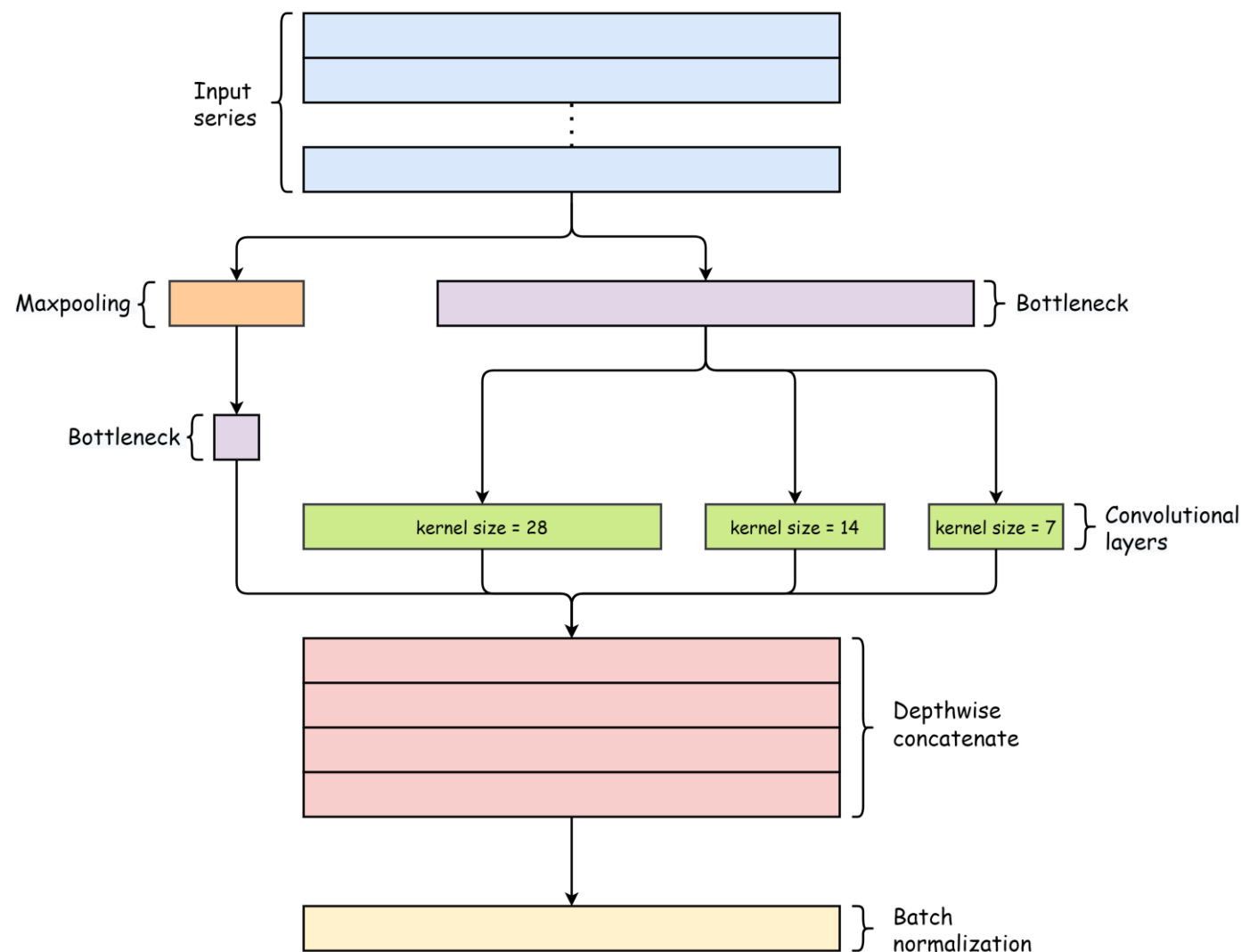


設計與實現

• 模型架構

– Classifier - Inception cell

- Bottleneck：用Conv1D對輸入資料降維
- Conv1D：用三種長度從長到短的kernel，個別著重於長、中、短期的特徵
- Depthwise concatenate：將4個Conv1D的輸出按照深度連接起來作為輸出

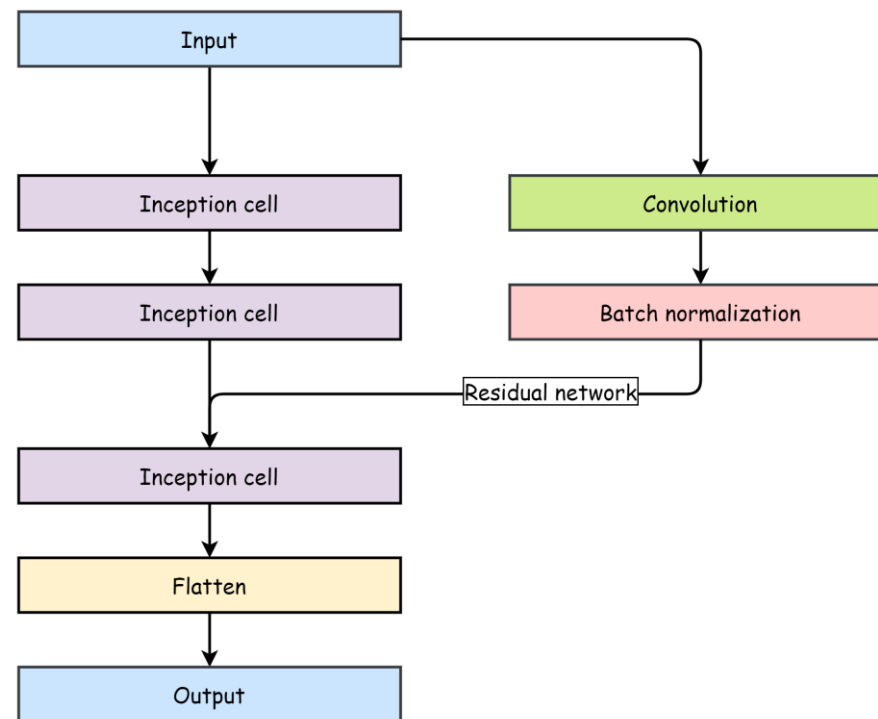


設計與實現

• 模型架構 (cont'd)

– Classifier

- 完整的模型
 - 經過3個inception cell
 - 1個residual network從input直接接到第三個inception cell
 - 用flatten展開inception cell的輸出
 - 用one-hot格式輸出六種分類的信心程度(0~1)

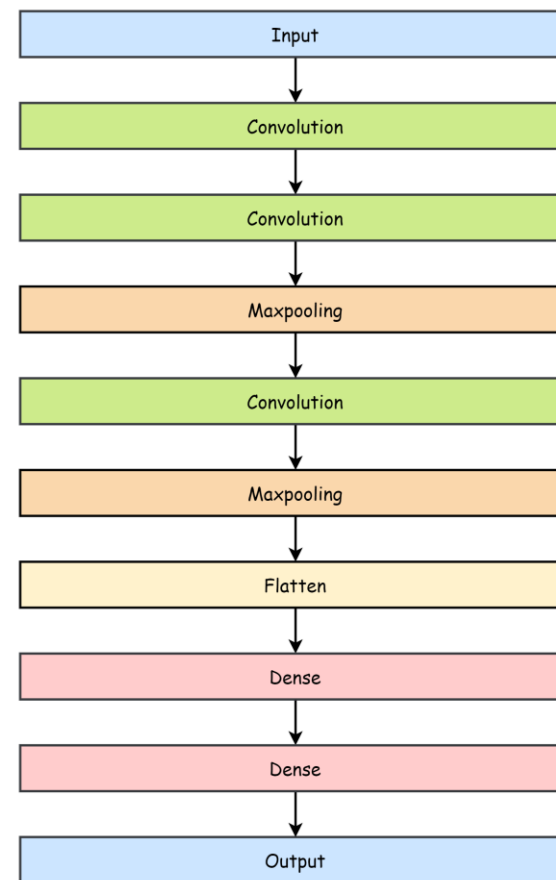


設計與實現

• 模型架構 (cont'd)

– Predictor

- 前段使用連續的Conv1D和Maxpooling，後段使用全連接層
- 輸出三軸加速度預期的數值
- 參考來源
 - <https://machinelearningmastery.com/how-to-develop-convolutional-neural-network-models-for-time-series-forecasting/>



設計與實現

• 量化模型

- Quantization Aware Training (QAT)：由於尚不支援模型使用的Conv1D layer，所以訓練時並未使用QAT
- 量化Int8 模型：
 - 優點：輕量化、運算快
 - 缺點：損失精確度
- 步驟
 1. 使用TFLiteConverter
 2. 挑選一部分資料(representative data)用來校正模型
 3. 將tensorflow模型轉換成tflite模型
 4. 使用TFLite Interpreter，並調整輸入輸出的維度
 5. 取得輸入和輸出的scale和zero point
- 使用模型
 - 利用得到的scale和zero point，將實際輸入轉換成int8的值輸入給模型，再將模型的int8輸出轉換回實際值，公式為： $\text{RealValue} = (\text{Int8Value} - \text{ZeroPoint}) * \text{Scale}$

Agenda

- 作品概述
- 難點與創新
- 設計與實現
- **作品進度**
- 測試結果
- 總結展望

作品進度

- 已完成之待辦事項（初賽時所提及的）

- ~~結合鏡頭~~

- 由於影像傳輸耗時，資料蒐集不易，且處理影像需要更大的模型，考量時間成本與效益不成比例，故捨棄之

- 加入加速度之微分 (Jerk) 作為偵測依據

- $J_x(t) = \frac{A_x(t) - A_x(t-1)}{dt}$

- 增加可偵測的車輛行駛狀態種類

- 加入左右轉，並加入有限狀態機來更精確的分辨車輛之行為

- 如怠速後偵測到加速為「起步 (Start-off)」

- 若是巡航或是左右轉後偵測到加速，則辨認為「加速 (Accel)」

作品進度

- 已完成之待辦事項（初賽時所提及的）

- 使用低通濾波來減少加速度波形的震盪
 - 使用C語言實作數位 Simple infinite impulse response filter
 - 以LPF參數 $\beta = 0.25$ 來消除部份路面之影響
- 同步標記頻率與推論頻率
 - 使用Non-blocking delay的程式設計手法 (初賽時為Blocking delay)
 - 以 $dt = 0.04$ (s) 來搜集資料與推論
 - 提升模型訓練與推論時的一致性，獲得更大的準確率
- 增廣資料集
 - 將旋轉矩陣套用到訓練集，進而產生角度稍有不同的增廣資料
 - 藉此平衡各類別的資料量
 - 提升模型對於數據的兼容性

作品進度

- 已完成之待辦事項（初賽時所提及的）

- 加入評分系統

- 目前可以對於「起步」、「巡航」、「煞車」、「左轉」、「右轉」與「加速」等6種狀態給予5星等級之評比

- 「起步」、「巡航」與「加速」著重加減速之穩定性（前後之加速度變化）

- 「左轉」與「右轉」則著重側向速度之穩定性（左右之加速度變化）

- 「巡航」則是總合評比（各方向之穩定性）

- 行駛結束後可以查看評分紀錄，並獲得評語

- 連接顯示裝置

- 以LCD 1602A作為顯示裝置

- 白天或是夜晚都可以看得很清楚

作品進度

- 已完成之待辦事項（初賽時所提及的）
 - 改用Arduino Pro Mini來縮小產品體積
 - 完成
 - 模組化供電裝置
 - 使用Micro USB介面之CP2102模組作為供電模塊
 - 只需一條線即可同時給予Arduino Pro Mini 5V電壓、Himax開發板3.3V電壓
 - 設計夾具以便安裝於車輛上
 - 搭配可快拆之夾具，方便拆卸或安裝到其他車種上

作品進度

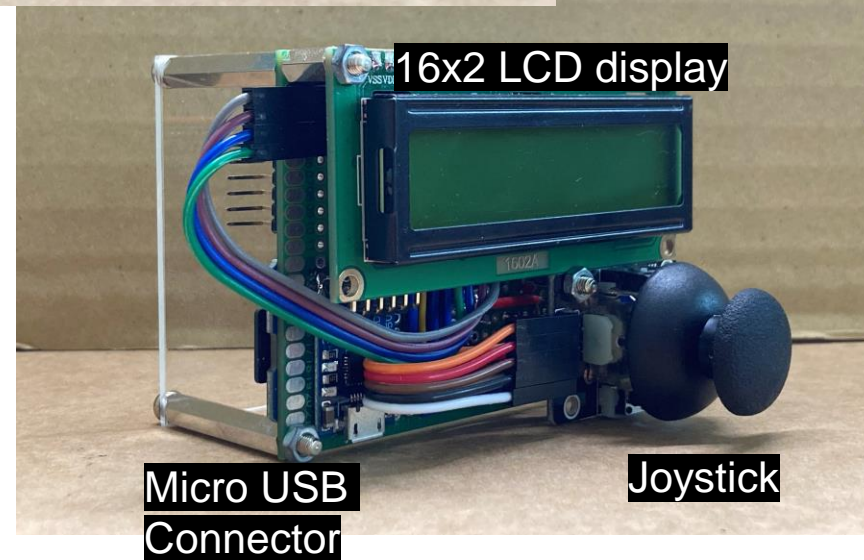
• 硬體

- 將元件焊接到6cmX8cm的電路板上
- 長寬高約為60mmX80mmX75mm
- LCD與搖桿安裝到電路板背面方便使用者操作
- 將Micro USB的充電線插入即可使用
- 預計使用SD卡模組儲存評分紀錄
- 硬體部分已完成

正面



背面



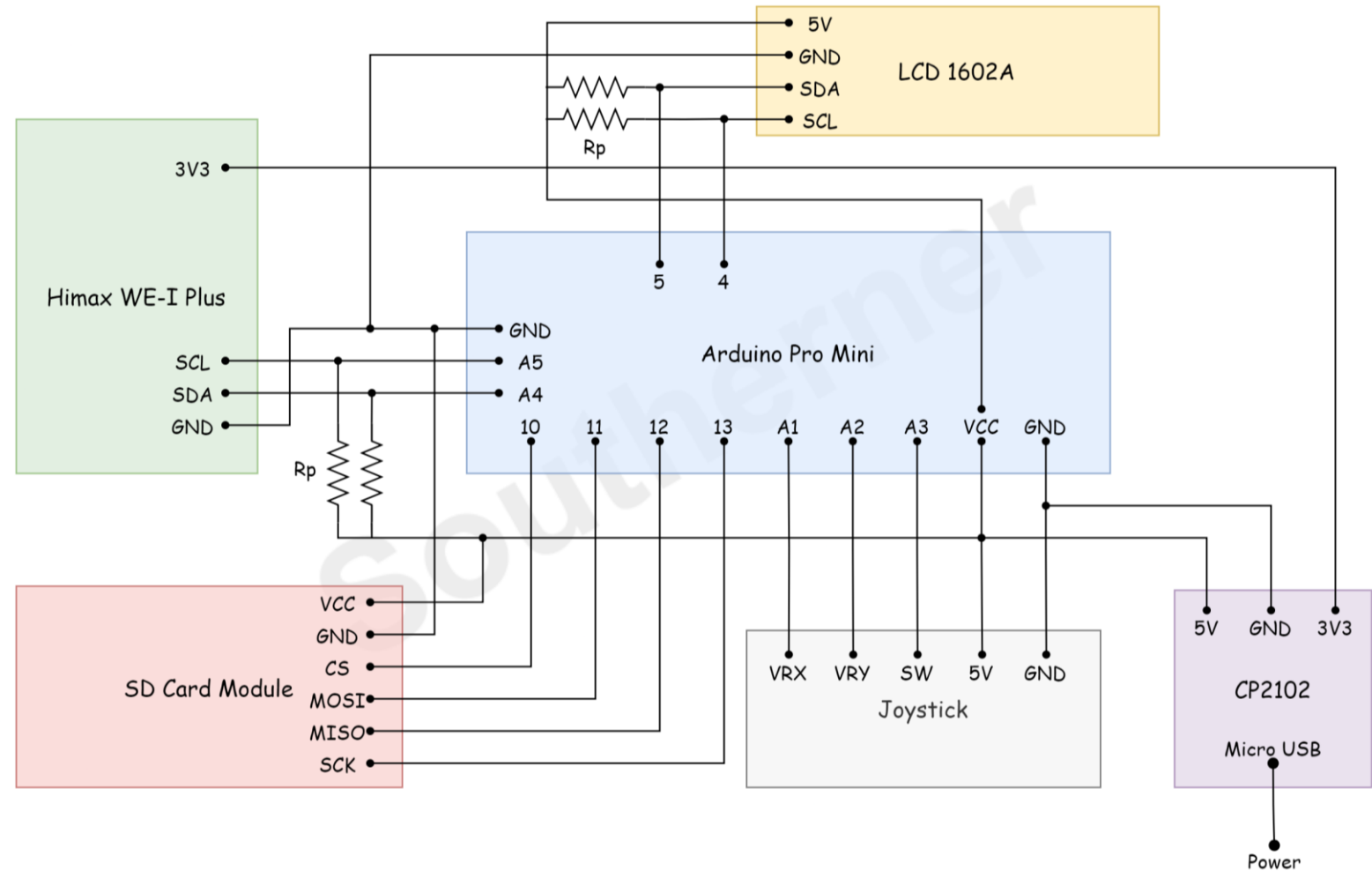
作品進度

- 硬體 (Cont'd)
 - 實際使用之情境



作品進度

• 硬體架構圖



作品進度

• 模型

- 已確定模型架構
- 成功部署兩個Int8模型到Himax WE-I Plus上
- Classifier
 - 準確率: 85.4%
 - Loss (Categorical Cross Entropy): 0.425
- Predictor
 - Loss (MSE): 0.36

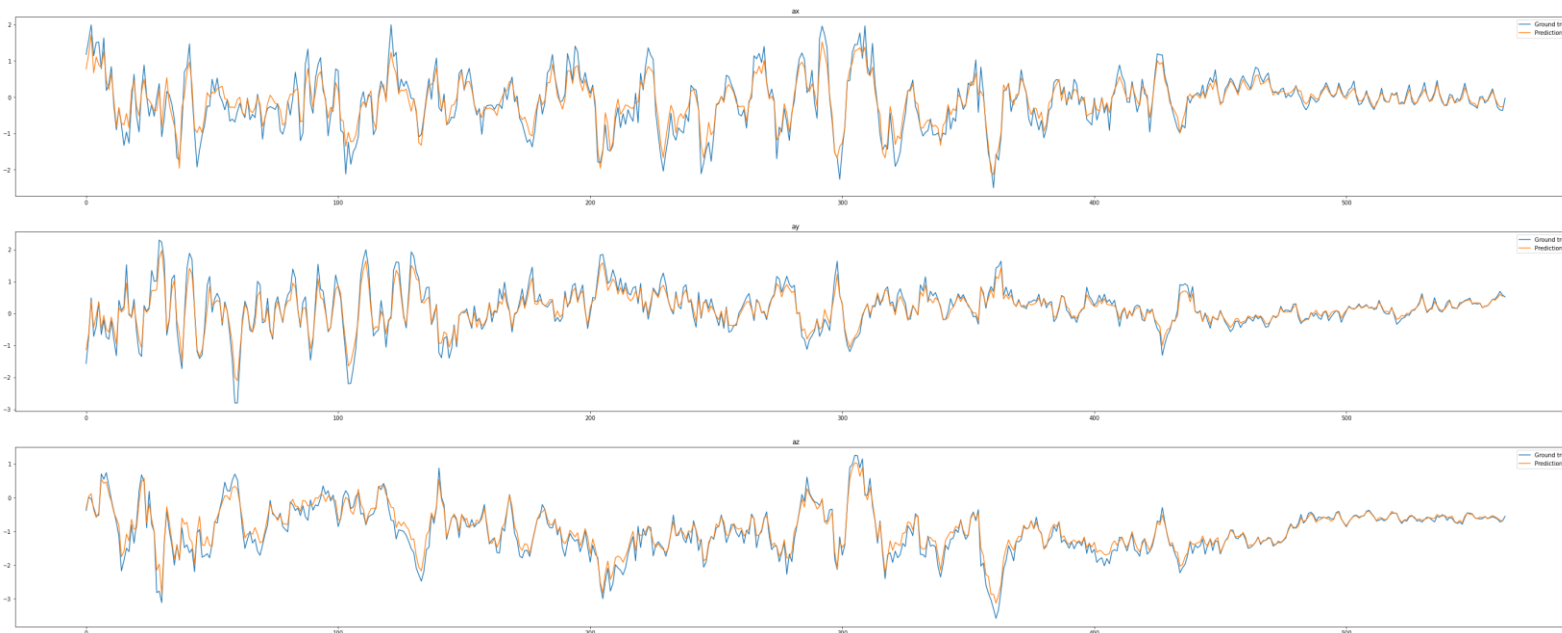
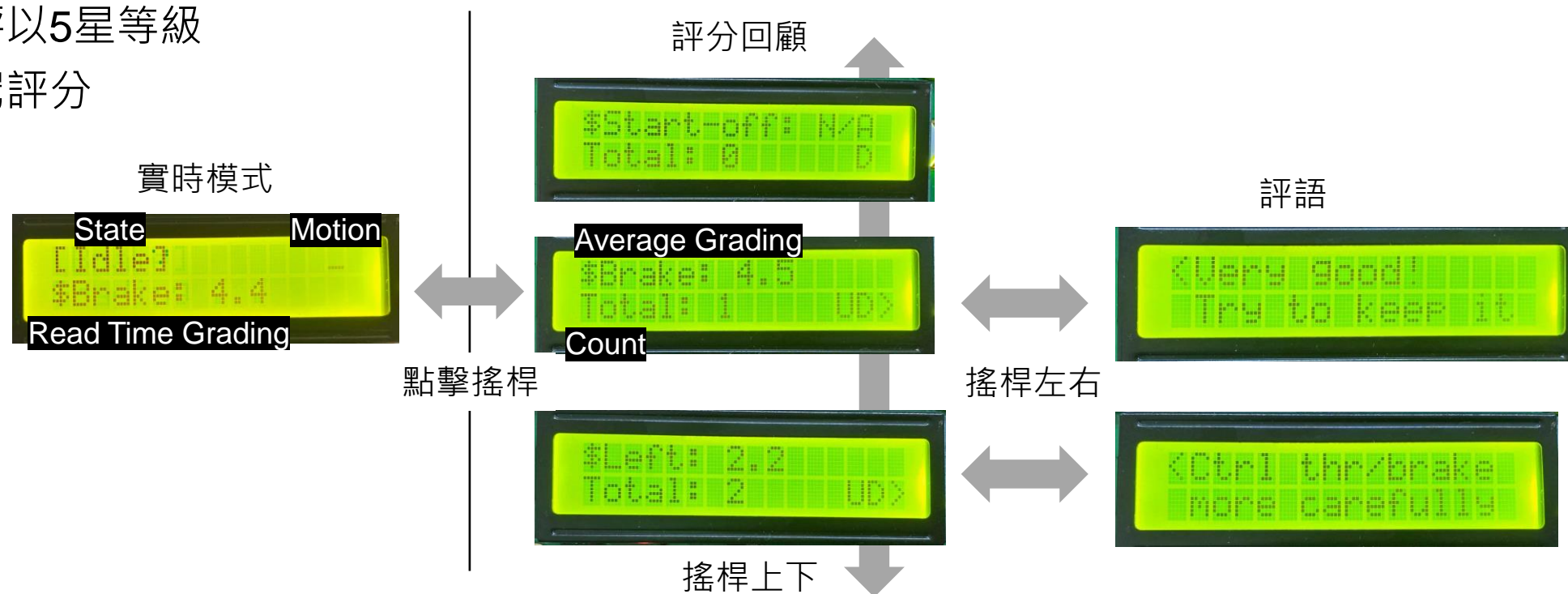


Fig. Predictor fitting validation data
(Blue: Ground Truth / Orange: Predictor output)

作品進度

• 軟體

- 實時模式下LCD顯示當前狀態與偵測到的操駕行為
 - 以Idle為例，[Idle]表示Strong State、Idle?表示Weak State
 - 偵測到的操駕行為以符號：_ = + - < > . 分別表示偵測到：閒置、巡航、加速、煞車、左轉與右轉
- 可在操駕行為發生後進行實時評分，並顯示於LCD上
- 可對操駕行為評以5星等級
- 可回顧各項操駕評分
 - 顯示平均
 - 評分次數
 - 評語



Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- **測試結果**
- 總結展望

測試結果

• 實際上路

- 模型大致可正確分辨出現在是什麼操駕行為
- 在一個操駕行為結束後，會實時給出該行為的評分
- 有時候還是會出現誤判的狀況
 - 如巡航時出現左右轉
 - 過慢的起步不容易被Model察覺
 - 擺放方式也會影響準確率
- 錯誤的行為判斷會導致評分失準
 - 出現極低的分數



起步評分為4.7顆星

測試結果

- 實際上路



巡航評分為4.4顆星



煞停評分為3.3顆星

測試結果

• 評分回顧

- 使用者可以藉由搖桿操作系統，選擇想回顧的評分項目
- 可以顯示項目的平均分數與發生次數
- 不同的評分會出現不同的評語
- 操作手感良好、流暢



起步評分為4.7顆星

測試結果

• 模型推論速度與效能

- 由於資料是以25Hz的頻率搜集的，故推論時也是以25Hz進行
- 推論時以25Hz做了以下
 - 資料前處理，如：LPF、Normalization與Quantization
 - 兩個模型管線化的推論
 - 模型輸出的後處理，如：Dequantization、Motion Detection、跑Finite State Machine與Mean Square Error的計算
 - 由I²C傳輸資料到Arduino
- 由於有做Quantization，Model都是以Int8進行運算，故推論效能非常好
- 為了確保判斷的正確性，程式會持續觀察模型輸出一段時間才給出判斷（約1秒）
 - Motion Detection設定之0.36秒與兩個State Transition：7*0.04*2 秒共約0.92秒
- Arduino接收25Hz的資料同時也處理使用者輸入與評分
 - 使用者輸入之回饋非常即時

Agenda

- 作品概述
- 難點與創新
- 設計與實現
- 作品進度
- 測試結果
- **總結展望**

總結展望

• 作品完成度

- 約90%
- 硬體架構已完成
- 模型架構與訓練流程已完成

• 待辦事項

- 搜集更多訓練資料
- 訓練更強大的**Classifier**模型以兼容其他車種與數據
- 完善評分機制，加入其他評分考量，如轉彎前是否有先煞車或是記錄扣分原因等
- 跟據扣分原因給予不同評語
- 使用**SD**卡來記錄評分歷史
- 加強**UI/UX**以增進操作性
- 增加硬體防水性，以對抗室外環境、預防下雨損壞硬體
- 想出決賽時如何**Demo**
- 祈求不要下雨

總結展望

• 未來展望

- 設計漂亮的外殼
- 讓更多人使用，來改善大台南地區的交通安全
- 結合手機App，連結線上競賽平台，讓使用者相互較量操駕風格，並設計獎勵機制
- 作為考照時路考的評分依據
- 作為職業駕駛的駕駛舒適度標準
- 藉此作品來減少路上三寶的數量

Thank You

