智慧化企業整合

深度學習應用 -

汽車駕駛安全輔助系統

指導教授： 邱銘傳 教授

學生：109034403 伍仰輝

中 華 民 國 １１０ 年 一 月 七 日

目錄

圖目錄

表目錄

1. 研究背景

在資訊時代下，手機與網路的功能已越來越發達，使用這些資訊裝置的時間也越來越平凡，甚至是在開車的時候都會分心使用資訊裝置，不只是駕駛中使用資訊裝置，在駕駛中聊天，尋找後座的東西等分心駕駛室非常危險的。根據 Federal Communications Commission 的官方統計，美國國家公路交通安全管理局的資料顯示，過去 7 年，美國超過 9%的致命車禍都是由分心駕駛造成的。除此之外，最新報告的資料顯示，2018 年，有 2800 多人因分心駕駛而喪命。在美國 2018 年，估計有 40 萬人在分心駕駛造成的車禍中受傷。

這些駕駛分心的根本解決的方法其實說到底就是駕駛時要專心，但是往往這種簡單的要求卻是難以服從的，其中的原因可能是駕駛時很無聊，心情很煩悶，朋友之間的打擾等。近幾年來，AI 的發展讓許多汽車品牌如：Tesla，BMW，Mercedes，Toyota 等都有輔助駕駛系統。雖然輔助駕駛系統可以減低分心駕駛導致車禍的機率，但是現有的輔助駕駛系統需要手動啟動，並沒有辦法完全自動啟動。因此本次研究希望提出一個自動啟動輔助駕駛的系統的流程，解決駕駛員因分心駕駛到底車禍的問題。

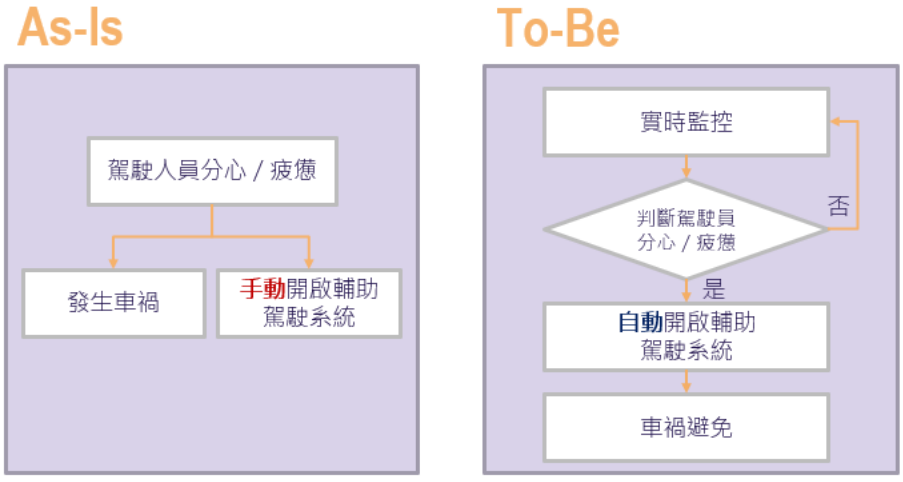


圖 1 As-Is vs. To-Be 意識圖

2. 研究目的

本次研究以將針對分心駕駛提出一套自動啟動輔助駕駛系統的流程，並且結合深度學習的方法作為啟動的判斷，其中兩種深度學習的模型分別為駕駛員行為辨識模型以及駕駛員疲勞狀態辨識模型。此外也希望透過這兩組模型搭配實時監控演算法可以成功辨識駕駛員在駕駛中的行為是否為分心，作為啟動輔助駕駛的判斷。

駕駛員行為辨識模型是以 State Farm 提供的公開資料集建立駕駛員行為的辨識，其中會運用卷積神經網路 (CNN) 架構實作，冀以此模型之實作練習影像處理與 CNN 演算法之模型建立。駕駛員疲勞狀態辨識模型則會以 MRL Eye Dataset 進行眼睛開關的分類從而進行辨識駕駛員的疲勞程度。

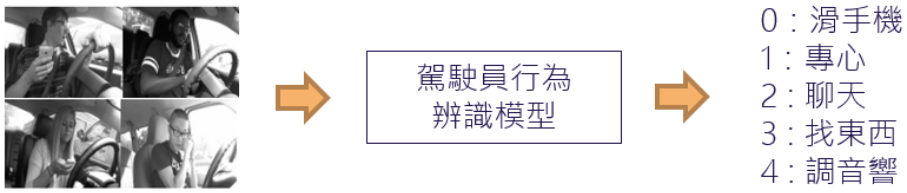


圖 2 駕駛員行為辨識模型流程示意圖

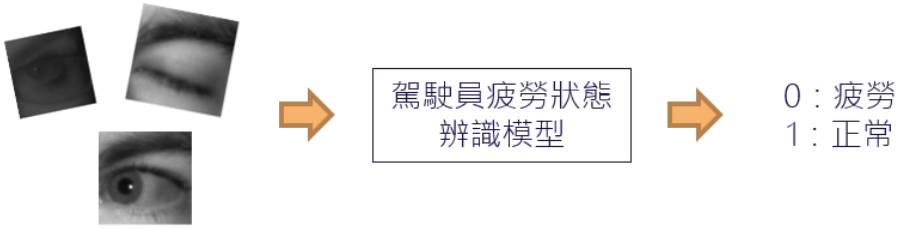


圖 3 駕駛員疲勞狀辨識模型流程示意圖

圖 4 5W1H 研究目的示意圖

3. 公開資料集介紹

本實驗採取 State Farm 資料集以及 MRL Eye 資料集進行駕駛員行為辨識模型以及駕駛員疲勞狀態辨識模型。State Farm 資料集裡有 9 總分類，分別為 {c0-c9: Safe Driving, Texting-right, Talking on the phone-right, Texting-left, Talking on the phone-left, Operating the radio, Drinking, Reaching behind, Hair and make-up, Talking to the passenger}，資料集總共有 22,424 筆資料集，訓練資料集共 17,943 筆，而測試資料集共 4,481 筆，每張照片皆為灰階圖，並以解析度 480 x 640 像素值呈現，如圖 4 所示。

MRL Eye 資料集裡有 2 總分類，分別為{open:開眼睛，close:關眼睛}，資料集總共有 77,283 筆資料，訓練資料集共 61,826 筆，而測試資料集共 15,457 筆，每張照片皆為灰階圖，並以解析度 86 x 86 像素值呈現，如圖 5 所示。

圖 5　State-Farm 訓練與測試資料視覺化示意圖

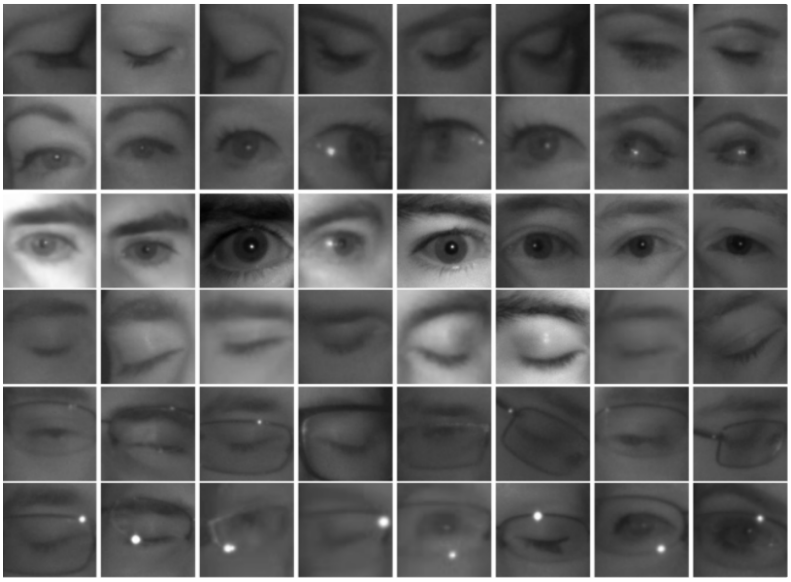


圖 6　MRL Eye Dataset 訓練與測試資料視覺化示意圖

4. 研究方法

圖 6 為本次實驗的研究架構圖，首先會將資料及進行視覺化的分析，接著進行資料分割、One-hot Encoding、資料標準化與資料增強等前處理作業。清理後的資料，藉由 CNN 模型框架完成行為辨識模型以及駕駛員疲勞狀態辨識模型。

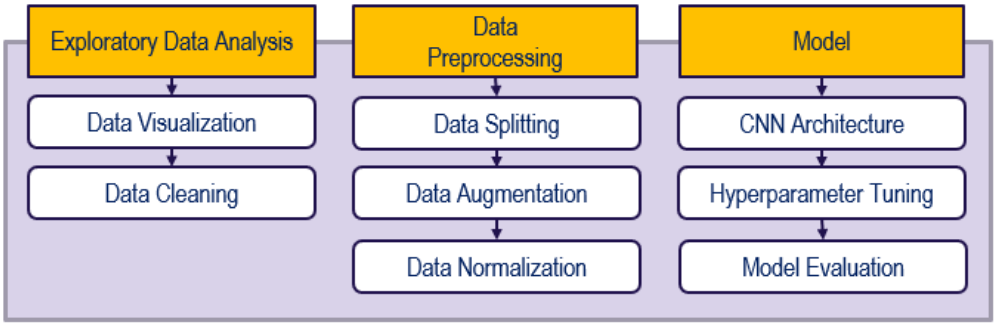


圖 6 研究架構圖

4.1 One-hot Encoding

One Hot encoding 的編碼邏輯為將類別拆成多個行 (column)，每個列中的數值由 1、0 替代，當某一列的資料存在的該行的類別則顯示 1，反則顯示 0。目的是為了將類別 (categorical) 或是文字 (text) 的資料轉換成數字，而讓程式能夠更好的去理解及運算。

4.2 資料標準化

由於此資料集以像素值呈現，而色碼定義之像素值皆為 0~255，因此將一張照片中的每個像素除上 255 以利縮小數據的跨度。

4.3 資料增強 (Image Augmentation)

透過 Image Augmentation，我們藉由旋轉、裁切、增加噪點、白化等技術，將原本的圖片做加強，如此一來，我們就硬生生地增加了許多的資料,彌補資料不足。在訓練一個分類器時，很容易遇到 Overfitting 的狀況，也就是對 Training Data 過於完美的擬合，此時，透過適當的圖像增強，也能降低 Overfitting 的可能性。本次運用 Keras 之 ImageDataGenerator 套件進行資料增強的作業。

4.4 CNN 原理

深度學習中的 CNN 較傳統的 DNN 多了 Convolutional (卷積) 及池化 (Pooling) 兩層 Layer，用以維持形狀資訊並且避免參數大幅增加。在加入此

兩層後，我們所看到的架構就如下圖分別有兩層的卷積和池化層，以及一個全連結層（即傳統的 DNN），最後再使用 Softmax activation function 來輸出分類結果。

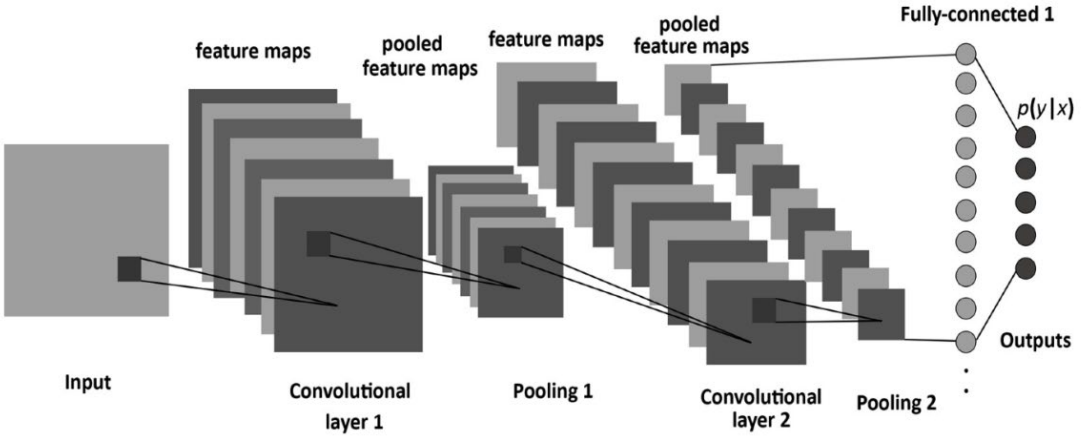


圖 7 CNN 演算法流程示意圖

4.5 遷移學習 (Transfer Learning) 原理

遷移學習的工作原理是先刪除所謂的「損失輸出結果」層，這是用於預測的最後一層，換成用於預測馬匹的新損失輸出結果層。這個損失輸出結果層是一個微調節點，用於判斷訓練如何處分有標籤的資料與預測輸出結果間的偏差。接著投入少量馬匹影像資料組，在整個五十層神經網路上，或是在最後幾層上，或是光是在損失層上進行訓練。運用這些遷移學習技術，在新的卷積神經網路上最終輸出結果便能辨識馬匹圖案。本次研究將採用 VGG-16, MobileNet 架構模型後再與建立的 CNN 架構比較進行模型評估。

5. 研究流程

5.1 駕駛員行為辨識模型

5.1.1 資料探索 (Exploratory Data Analysis)

從資料集可以看出資料集有以下的分佈，總共 10 總組合，分別為 Safe Driving, Talking to passenger, hair and makeup, reaching behind, dinking, operating the radio, talking on the phone- left, talking on the phone – right,

texting – left, texting – right, talking on the phone – left。從圖 10 右邊的圓餅圖可以看出最小的資料 talking to passenger 種類佔 8.5%，最大的資料種類 safe driving 佔 11.1%。由此可見，資料集並沒有純在不平衡資料，因此在建立模型時不必擔心結果又偏差或分類錯誤的問題。

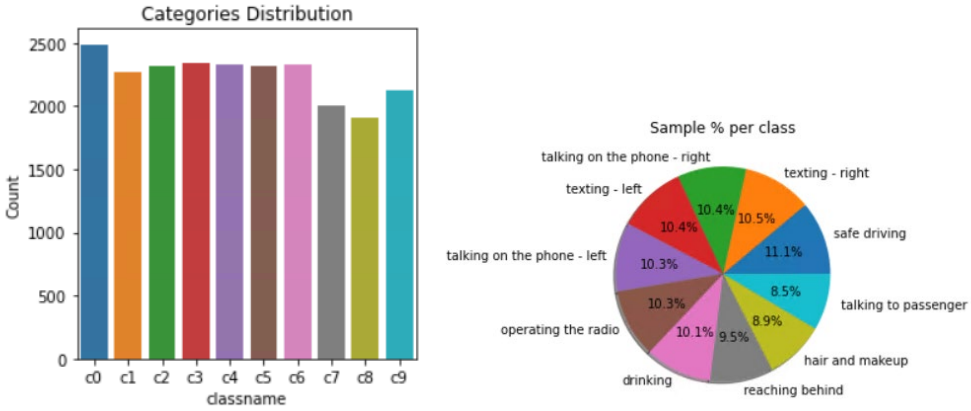


圖 8　資料視覺化示意圖

接著進行資料分割的作業，將資料分成訓練資料 80%與測試資料集 20%。資料分割程式碼與各資料集筆數示意圖如圖 11。

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=
```

```
print(y_train.shape,y_test.shape)
```

```
(17939, 1) (4485, 1)
```

圖 9　資料分割程式碼與各資料集筆數示意圖

5.1.2 資料前處理 (Data Preprocessing)

由於原先目標值為 0~24 之整數類別資料，因此本實驗將進行 One-hot Encoding 程序將目標值轉換成 0,1，以利電腦作業。圖 12 為 One-hot Encoding 程式碼示意圖，圖 13 為 One-hot Encoding 結果示意圖。

```
from keras.utils import np_utils

Y_train = np_utils.to_categorical(y_train,num_classes=10)
Y_test = np_utils.to_categorical(y_test,num_classes=10)
```

```
Y_train
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 1., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.]], dtype=float32)
```

圖 10　One-hot Encoding 程式碼示意圖

為了避免訓練資料與測試資料過擬合，以 Keras 之 ImageDataGenerator 套件對於訓練資料集進行資料增強的作業。調整的範圍有照片旋轉的角度、隨機鏡像調整、隨機垂直與水平移動圖像等設定。圖 17 為資料增強設定程式碼示意圖。

```
train_datagen = ImageDataGenerator(rescale = 1.0/255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True,
                                   validation_split = 0.2)
```

圖 11　資料增強設定程式碼示意圖

5.1.3 模型架構 (Model Construction)

一開始以 Trial and Error 嘗試 CNN 的架構設計，經過無數次的嘗試後得出以下的架構，架構如圖 10。並且 loss function 以 categorical_crossentropy，optimizer 使用適合用於圖形辨識的 adam，模型參考指標也以準確率為主。

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_16 (Conv2D)           (None, 238, 238, 128)     3584
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 119, 119, 128)     0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 117, 117, 64)      73792
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 58, 58, 64)        0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 56, 56, 32)        18464
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 28, 28, 32)        0
_________________________________________________________________
flatten_4 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_11 (Dense)             (None, 1024)              25691136
_________________________________________________________________
dense_12 (Dense)             (None, 256)               262400
_________________________________________________________________
dense_13 (Dense)             (None, 10)                2570
=================================================================
Total params: 26,051,946
Trainable params: 26,051,946
Non-trainable params: 0
_________________________________________________________________
```

圖 12　本研究使用的 CNN 模型架構

一开始調整 Batch Size，本實驗比較 5 種 Batch Size，如 32, 64, 128, 256, 560。可以得知 Batch Size 為 560 的時候模型表現最好，因此保留其參數設定。表 1 為 Batch Size 評估比較表。

表 1　Batch Size 評估比較表

Batch Size	Training	Testing
32	70.08%	69.63%
64	86.52%	85.46%
128	88.13%	87.31%
256	90.67%	90.12%
560	92.54%	91.72%

調整 Epoch 的次數，本實驗比較三種 Epoch 的次數，10、20、30。可以得知 Epoch 為 30 的時候模型表現最好,但是我认为 30 Epoch 并没有显著的比较好，表示當 Epoch 為 10 時已經收斂，因此考虑时间成本下决定选择 Epoch 为 10 的结果，保留其參數設定。表 2 為 Epoch Times 評估比較表。

表 2　Epoch Times 評估比較表

Epoch Times	Training	Testing
10	92.54%	91.34%
20	92.78%	91.54%
30	92.80%	91.72%

接著進行超參數的調整，首先謹調整輸出層 Activation Function，本實驗比較三種 Activation Function，如 softmax、ReLU、Sigmoid。可以得知 Activation Function 為 Sigmoid 的時候模型表現最好，因此保留其參數設定。表 3 為 Activation Function 評估比較表。

表 3　Activation Function 評估比較表

Activation Function	Training	Testing
softmax	97.17%	94.79%
ReLU	80.56%	80.15%
Sigmoid	92.54%	91.34%

從超參數調整的過程中，可以發現模型的訓練資料與測試資料準確度已經從原始設定的 92.54%，91.72% 提升至 97.1%，94.79%，並且從以下圖 x 也可看出此模型並沒有 Overfitting 的問題，因此接受本次模型為最佳模型。

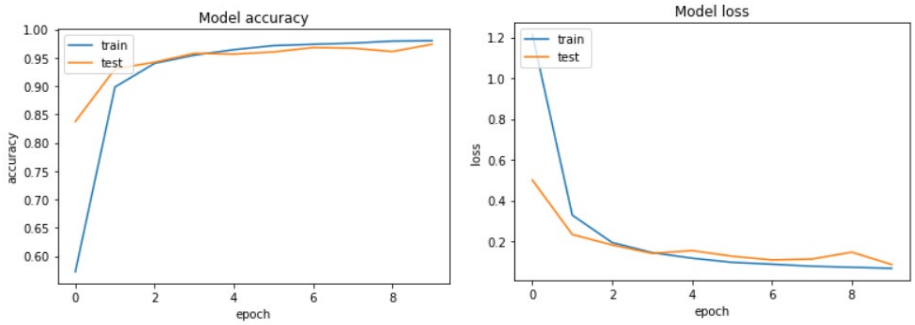


圖 13　本研究模型 Training vs. Testing Acc & Loss

5.1.4 遷移模型 (Transfer Learning)

Vgg16 是牛津大學視覺幾何組 (Oxford Visual Geometry Group) 2014 年提出的一個模型。Vgg 模型也得名於此。2014 年，vgg16 拿了 Imagenet Large Scale Visual Recognition Challenge 2014 (ILSVRC2014) 比賽的冠軍。因此運用 VGG16 重複跑出以上的資料集，並檢視此模型結果是否有比本研究提出的模型來得好。

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_2 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d_1 (MaxPooling2	(None, 112, 112, 64)	0
conv2d_3 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_4 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_2 (MaxPooling2	(None, 56, 56, 128)	0
conv2d_5 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_7 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_3 (MaxPooling2	(None, 28, 28, 256)	0
conv2d_8 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_10 (Conv2D)	(None, 28, 28, 512)	2359808
max_pooling2d_4 (MaxPooling2	(None, 14, 14, 512)	0
conv2d_11 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_12 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_13 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_5 (MaxPooling2	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_1 (Dense)	(None, 4096)	102764544
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4096)	16781312
dropout_2 (Dropout)	(None, 4096)	0
dense_3 (Dense)	(None, 2)	8194

Total params: 134,268,738
Trainable params: 134,268,738
Non-trainable params: 0

圖 14 VGG16 模型意識圖

本模型同樣進行超參數優化，並且以 Epoch 為 25，Batch Size 為 560 以及 Softmax 作為 VGG16 的參數設定。

表 4 VGG-16 Epoch Time 比較表

Epoch Time	Training	Testing
20	84.81%	83.79%
25	86.54%	85.34%

30	86.84%	85.47%

表 5　VGG-16 Batch Size 比較表

Batch Size	Training	Testing
140	82.14%	82.15%
280	85.41%	84.11%
560	86.54%	85.34%

表 6　VGG-16 Activation Function 比較表

Activation Function	Training	Testing
softmax	91.70%	91.75%
ReLU	74.56%	74.15%
Sigmoid	86.54%	85.34%

從超參數調整的過程中，可以發現模型的訓練資料與測試資料準確度已經從原始設定的 96.54%，85.34% 提升至 91.70%，91.75%，並且從以下圖 x 也可看出此模型並沒有 Overfitting 的問題，因此接受本次模型為最佳模型。

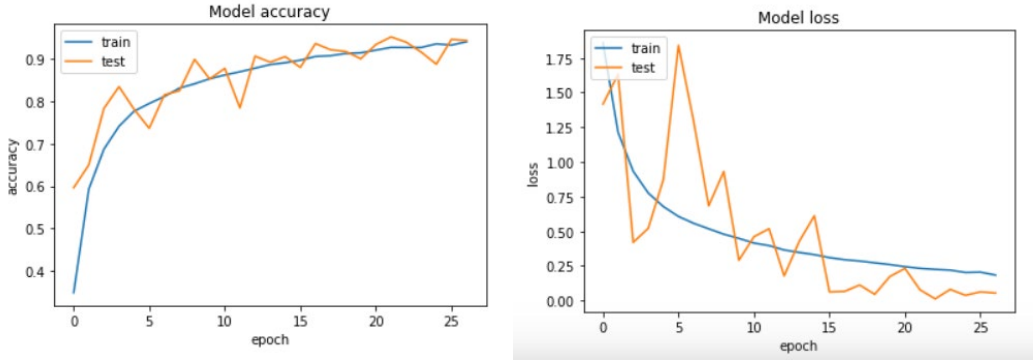


圖 15　VGG-16 Training vs. Testing Acc & Loss

5.2 駕駛員疲勞狀態辨識模型

5.2.1 資料探索 (Exploratory Data Analysis)

本次資料集運用 MRL EYE Dataset,資料集共有 77283 筆資料，並且分為開眼睛以及關眼睛。過後以 80%切割資料成 Training Data, 20% 作為 Testing Data。並且從圖 16 也可以看出資料分佈均勻，沒有不平衡的問題，因此沒有再去做解釋。

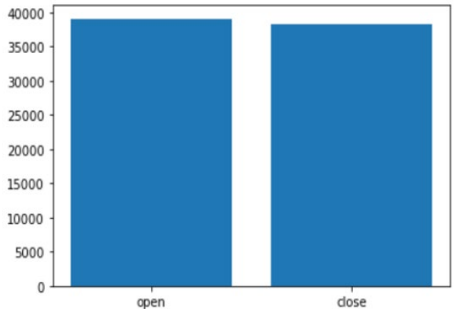


圖 16　資料分佈示意圖

5.2.2 資料前處理 (Data Preprocessing)

本次資料前處理運用了 backtorgb 將圖片轉換成 RGB 的陣列，並且重組大從 86x86 至 244x244。過後也對訓練集進行了 Normalization。

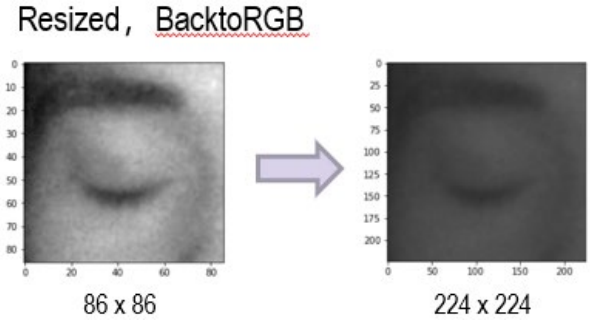
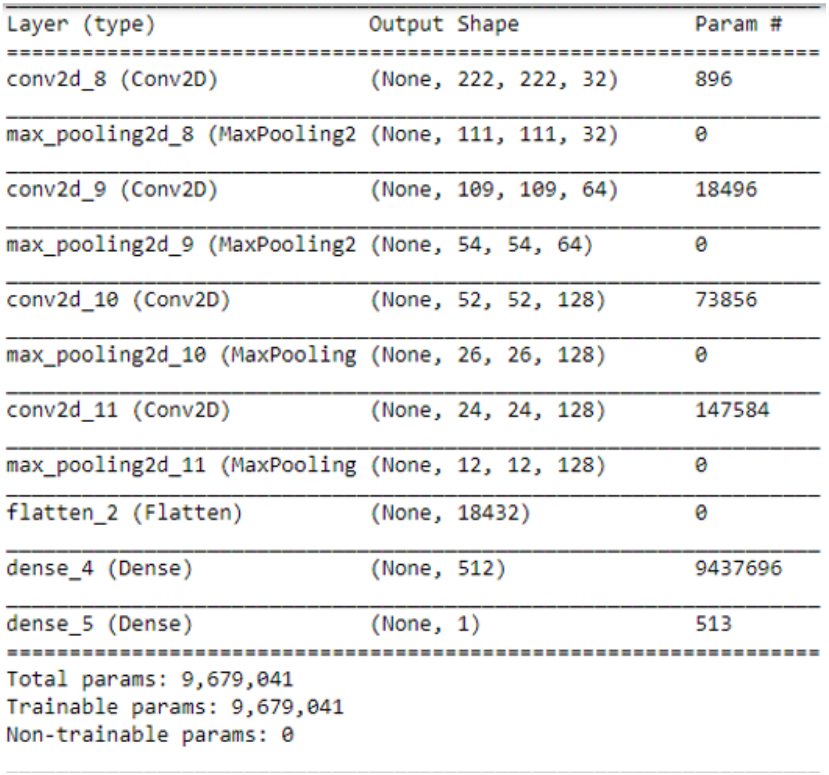


圖 17　資料前處理示意圖

5.2.3 模型架構 (Model Construction)

一開始以 Trial and Error 建立 CNN 的架構設計，經過無數次的嘗試後得出以下的架構。輸出層以 224x224x3 設計，輸出層以 1 的 Fully Connected Layer 設計。並且 loss function 以 binary_crossentropy，optimizer 使用適合用於圖形辨識的 adam，模型參考指標也以準確率為主。



Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_8 (MaxPooling2	(None, 111, 111, 32)	0
conv2d_9 (Conv2D)	(None, 109, 109, 64)	18496
max_pooling2d_9 (MaxPooling2	(None, 54, 54, 64)	0
conv2d_10 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_10 (MaxPooling	(None, 26, 26, 128)	0
conv2d_11 (Conv2D)	(None, 24, 24, 128)	147584
max_pooling2d_11 (MaxPooling	(None, 12, 12, 128)	0
flatten_2 (Flatten)	(None, 18432)	0
dense_4 (Dense)	(None, 512)	9437696
dense_5 (Dense)	(None, 1)	513

Total params: 9,679,041
Trainable params: 9,679,041
Non-trainable params: 0

圖 18 本研究 CNN 模型架構

以下進行超參數優化，並且選擇效果較好的 Testing Accuracy 作為選擇的指標，因此選擇 sigmoid, optimizer 選擇 RMSprop，Epoch Time 選擇 10。

表 7 Activation Function 比較表

Activation Function	Training	Testing
softmax	50.54%	53.45%
Sigmoid	96.86%	96.39%

表 8 Optimizer 比較表

Optimizer	Training	Testing
adam	96.86%	96.39%
RMSprop	96.81%	97.37%

表 9 Epoch 比較表

Epoch Time	Training	Testing
5	96.86%	97.39%
10	97.85%	97.41%
20	98.29%	97.50%

從超參數調整的過程中，可以發現模型的訓練資料與測試資料準確度已經從原始設定的 96.86%，96.39% 提升至 98.29%，97.50%，並且從以下圖

也可看出此模型並沒有 Overfitting 的問題，因此接受本次模型為最佳模型。

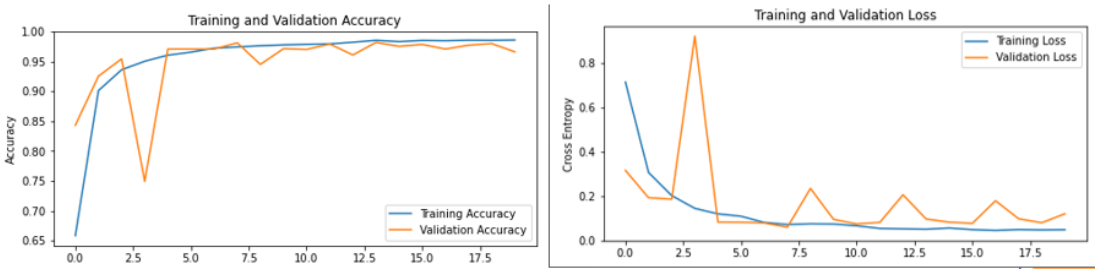


圖 19 本研究 CNN Training vs. Testing Accuracy & Loss

5.2.4 遷移模型 (Transfer Learning)

Mobilenet 是基於一種流線型的架構，它使用縱向的可分離卷積來構建輕量級的深層神經網絡。我們引入了兩個簡單的全局超參數，有效地在延遲和精度之間進行權衡。這些超參數允許模型生成器根據問題的約束爲其應用程序選擇合適大小的模型。我們在資源和精度權衡方面進行了廣泛的實驗，並在圖像網分類方面與其他流行的模型相比表現出很強的性能。然後， MobileNet 在廣泛應用和使用案例中的有效性，包括目標檢測、細粒度分類、人臉屬性和大規模地理定位。

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

圖 20　MobileNet 架構示意圖

運用架構改類似如圖 18 改良 Input Layer 與 Output Layer 的 MobileNet，設 Epoch 為 5, Optimizer 為 adam, Loss 為 binary_crossentroypy，可得 98.79% 的 Training Accuracy, 98.95 % 的 Testing Accuracy，結果可以說是相當的高，這也許是因為此分類問題為二元分類問題，並不是非常複雜的問題，因此在用 pretrain 的 model 可以得到相當高的 accuracy。並且透過圖 19 也可以看出模型並沒有 overfitting 的問題，結果雖然感覺還未收斂，但是結果已經相當的好，因此接受此模型，並且也沒再進行超參數優化。

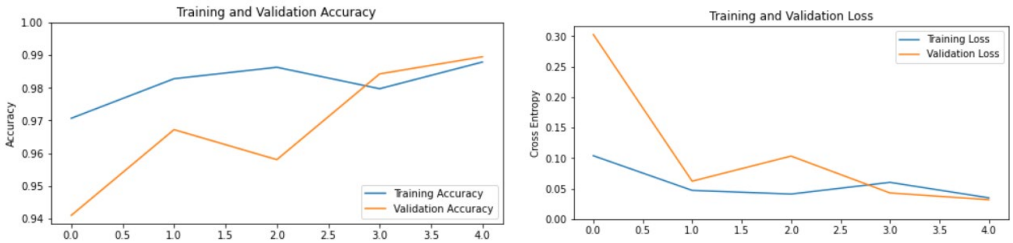


圖 21　MobileNet Training vs. Testing Accuracy & Loss

6. 結論

6.1 模型研究結果

6.1.1 駕駛員行為識模型

從超參數優化可以看出，超參數對模型的影響非常的大，並且透過模型比較選出最好的模型。

表 10 駕駛員行為識模型比較表

	本研究提出模型	VGG-16
Training Accuracy	97.17%	91.70%
Testing Accuracy	94.79%	91.75%
Training Loss	0.1492	0.2710
Testing Loss	0.1405	0.2521

從本次提出模型比較 VGG-16 可以看出，本次研究所提出的模型準確度比 VGG-16 來得好，考慮到不同的資料集的因素，這也許只是針對此資料集模型才特別有效。

6.1.2 駕駛員疲勞辨識模型

表 11 駕駛員疲勞辨識模型比較表

	本研究提出模型	MobileNet
Training Accuracy	98.29%	98.79%
Testing Accuracy	97.50%	98.95%
Training Loss	0.0558	0.0348
Testing Loss	0.0823	0.0317
Computational Time	1617.4 秒	3856.8 秒

從本次提出模型比較 MobileNet 雖然 MobileNet 的準確度比較高，但是如果考慮到模型執行的速度，運用本研究提出的模型架構依然可以達到高達　　的準確率。

本次研究再結合 OPENCV 實時判斷駕駛員是否關眼睛大於 1.6 秒皆可以知道駕駛員是否打瞌睡並且是否需要啟動輔助駕駛系統，其架構圖如下。

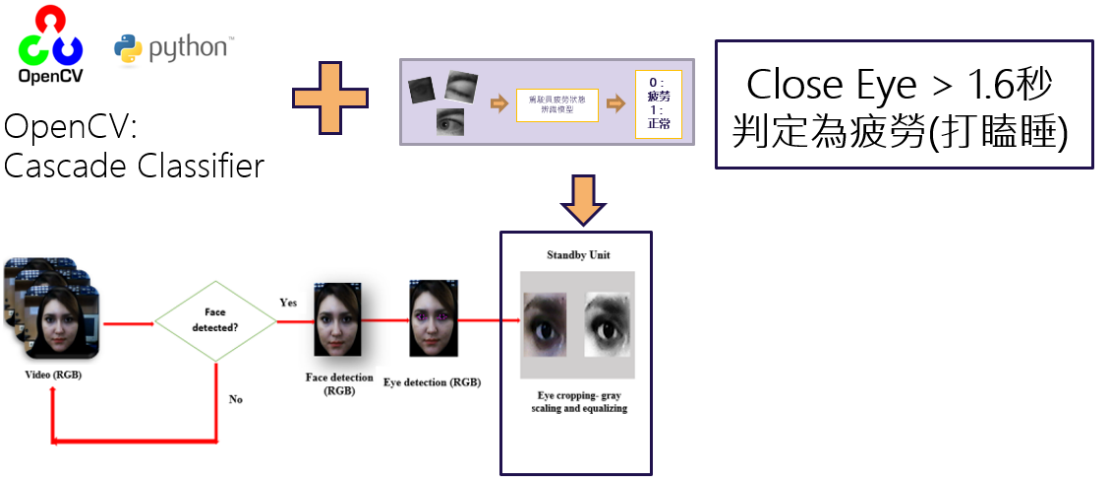


圖 22 駕駛員疲勞辨識模型啟動輔助駕駛系統架構示意圖

6.2 模型應用

6.2.1 自動啟動輔助駕駛系統

結合了以上模型後，在搭配實時監控的演算法 Open CV，可以得出自動啟動輔助駕駛系統的契機，下圖為其架構圖。

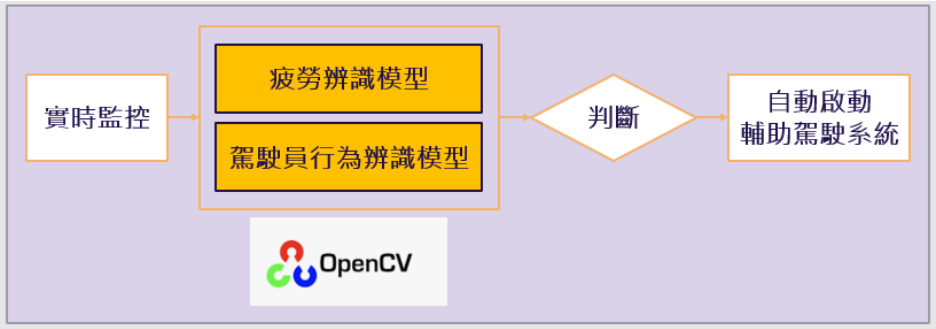


圖 23 自動啟動輔助駕駛系統架構示意圖

6.2.2 機車安全帽產品設計

除了結合成自動啟動輔助駕駛系統，更可以利用疲勞辨識模型應用在安全帽的設計上，記得有一次課上演講有請到以為學長分享到他有一次太累騎車所以摔車，因此透過此模型可以在駕駛員要睡著時發出警報身提高警惕，或是放出小小的電刺激使用者等。

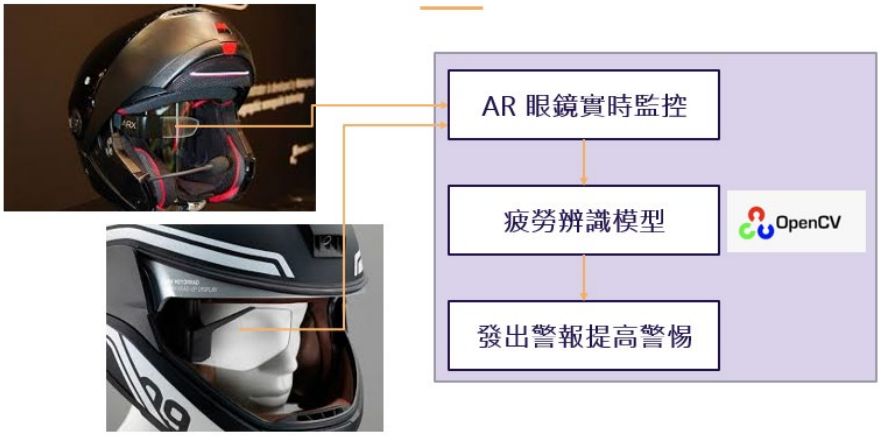


圖 24 創新產品設計 - 安全帽示意圖

6.2.3 創新研發與數據分析

透過實時監控系統，在使用者的同意下可以截取使用者在時間下的行為，這些行為是非常可貴的消費者的聲音，並且透過大數據分析而分析目前的汽車在內部設計上有什麼不足的地方。

6.3 未來研究方向

本次研究的雖然準確度高，但是作為以眼睛開關的狀態從而判斷疲勞程度依然有待加強，未來可以嘗試利用生理數據或是眼睛裡的血絲進行疲勞程度辨識，或是利用多幾組不同的資料集進行訓練，讓模型在不同背景有人物可以有更好的準確度。

References

1. https://towardsdatascience.com/drowsiness-detection-system-in-real-time-using-opencv-and-flask-in-python-b57f4f1fcb9e
2. https://data-flair.training/blogs/python-project-driver-drowsiness-detection-system/
3. EyeNet: An Improved Eye States Classification System using Convolutional Neural Network, IEEE 2020 22nd International Conference on Advanced Communication Technology (ICACT)
4. https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69
5. https://medium.com/datadriveninvestor/review-on-mobilenet-v1-abec7888f438

Dataset

- https://www.kaggle.com/c/state-farm-distracted-driver-detectionSmiley delivery man in car with mobile
- http://mrl.cs.vsb.cz/eyedataset