深度學習應用 –
汽車駕駛安全輔助系統

伍仰輝
109034403

大綱

01 研究目的

- As-Is To-Be
- 5W1H

02 駕駛員動作辨識模型

- Exploratory Data Analysis
- Data Preprocessing
- Model Illustration
- Hyperparameter tuning

03 疲勞辨識模型

- Exploratory Data Analysis
- Data Preprocessing
- Model Illustration
- Hyperparameter tuning

04 結論

- 模型總結
- 自動駕駛優化
- 實時數據回饋

研究背景與動機

Federal Communications Commission | Browse by CATEGORY | Browse by BUREAUS & OFFICES | Search

About the FCC | Proceedings & Actions | Licensing & Databases | Reports & Research | News & Events | For Consumers

Home / Consumer / 消費者指南 /

分心駕駛的危險性

2018年，估計有40萬人在分心駕駛造成的車禍中受傷

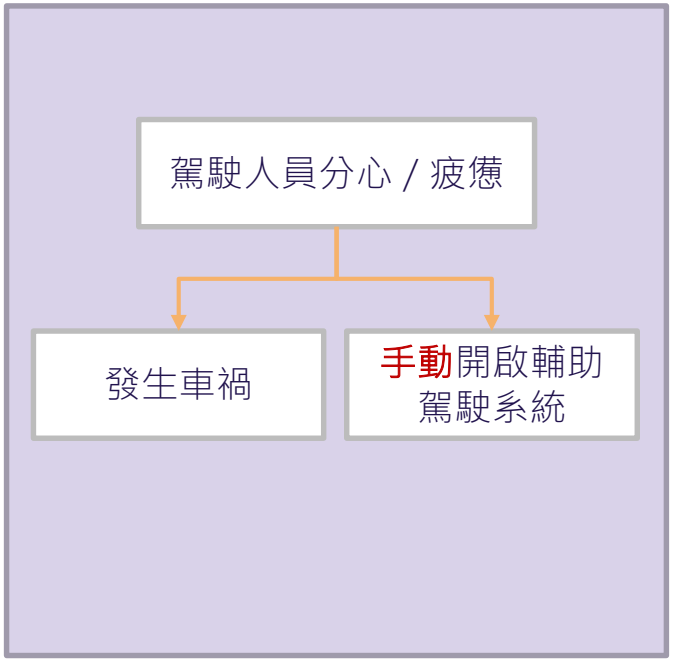
無線裝置的普及帶來了一些意外有時甚至是致命的後果。與駕駛時分心（包括駕駛時使用行動裝置）有關的交通意外事故的數量驚人，造成人身傷害和喪命。全國統計資料發人深省：

- 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。
- 美國國家公路交通安全管理局最新報告的資料顯示，2018年，有2800多人因分心駕駛而喪生。
- 2018年，估計有40萬人在分心駕駛造成的車禍中受傷。
- 《美國全國乘員保護措施使用調查》報告說，16-24歲的司機使用手機的比例依然最高。

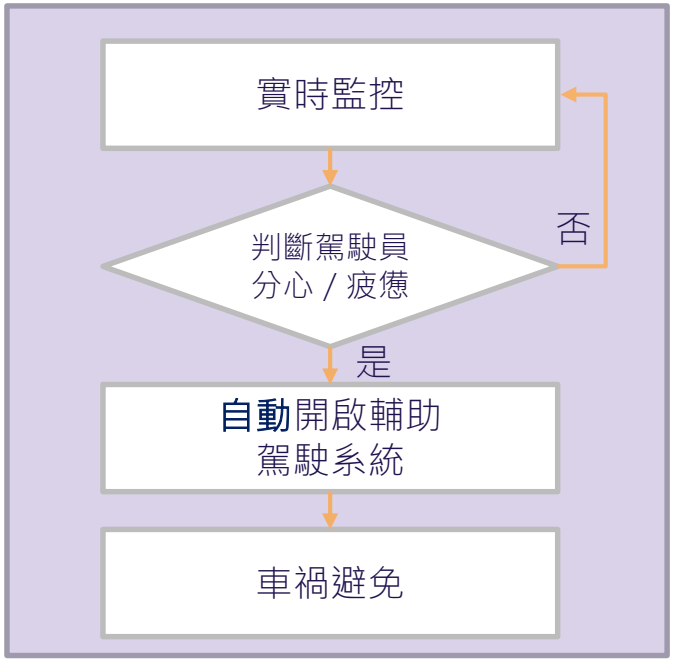
Solution ?

研究目的 (As-Is vs. To-Be)

As-Is



To-Be



研究目的 (5W1H)

Who

- 汽車駕駛員

Where

- 汽車駕駛員在開車的時候往往會分心

When

- 任何開車的時候 (長途駕駛，疲勞駕駛)

Why

- 輔助駕駛系統不會自動啟動，造成駕駛員分心時導致車禍

What

- 實作駕駛員狀況辨識模型 (疲勞狀況 & 行為判斷)
- 在必要時候自動開啟輔助駕駛系統

How

- 利用 CNN & Transfer Learning 辨識駕駛人員行為狀況
- 利用實時監控演算法，預測駕駛員狀況作出對應程序

資料集介紹

駕駛員行為
辨識模型

0：滑手機
1：專心
2：聊天
3：找東西
4：調音響

kaggle

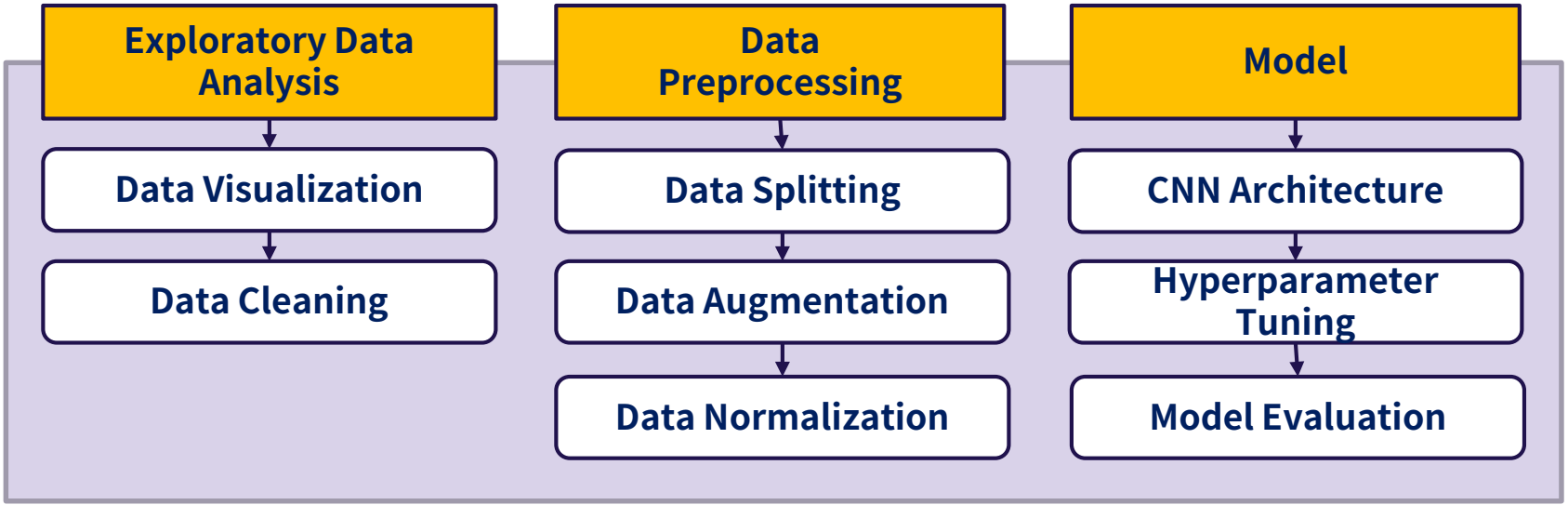
駕駛員疲勞狀態
辨識模型

0：疲勞
1：正常

02

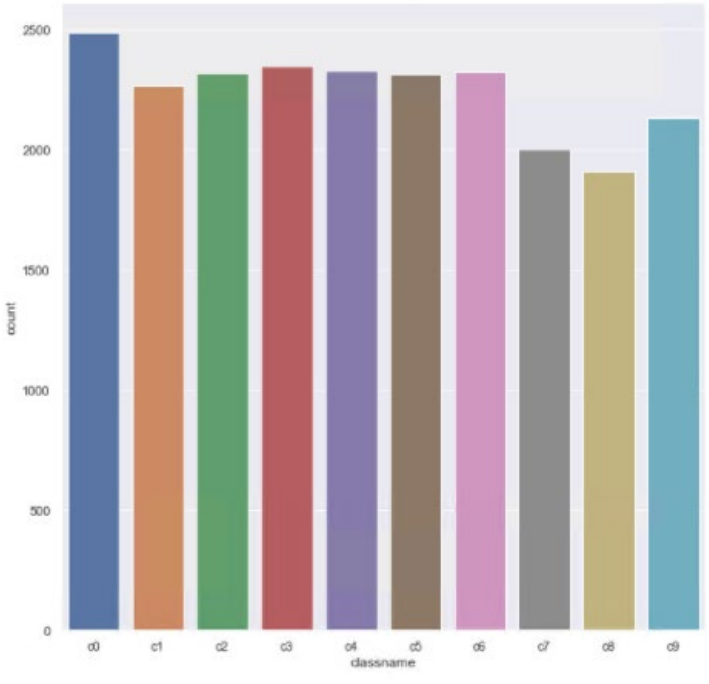
駕駛員行為辨識模型

研究架構


Exploratory Data Analysis
Data Visualization
Data Cleaning
Data Preprocessing
Data Splitting
Data Augmentation
Data Normalization
Model
CNN Architecture
Hyperparameter Tuning
Model Evaluation

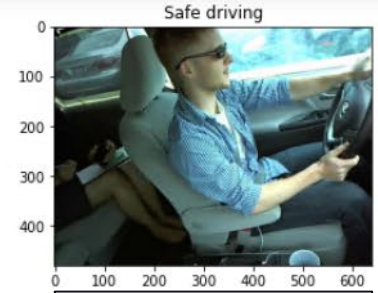
資料探索 (EDA)

- 22424 筆數據
- 17943 Training , 4481 Testing (80% Training , 20% Testing)
- 10 種類別
- 沒有 Imbalanced 問題




```
raw_data = pd.read_csv('C:/Users/chentzuli/Downloads/state-farm-distracted-driver-detection/driver_imgs_list.csv')
plt.figure(figsize = (10,10))
sns.set_style('darkgrid')
sns.countplot(raw_data['classname'])
```

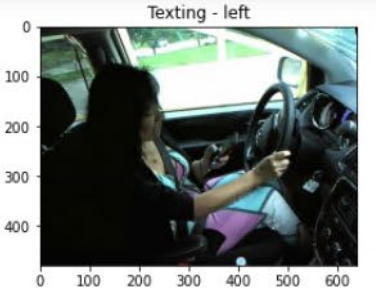
資料探索 (EDA)



Safe Driving

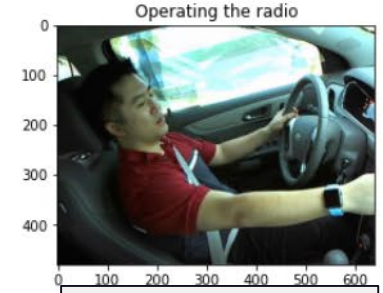
Texting - right

Talking on the phone - right



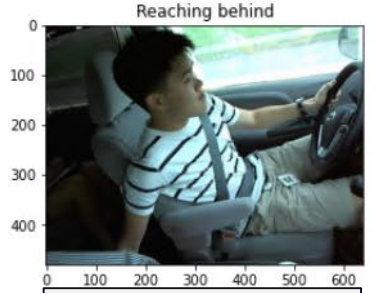
Texting - left

Talking on the phone - left



Operating the radio

Drinking



Reaching behind

Hair and make up

Talking to passenger

資料前處理 (Preprocessing)

- One Hot Encoding
  進行資料類別分類
- Image Augmentation
  增強資料，避免
  Overfitting
- Resized image
  480x640 into 240x240
- Normalization

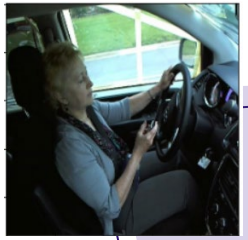
```
from keras.utils import np_utils

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)
```

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



模型 (Model Illustration)

Model: "sequential_15"

Layer (type)	Output Shape	Param #
conv2d_36 (Conv2D)	(None, 238, 238, 128)	3584
max_pooling2d_33 (MaxPooling	(None, 119, 119, 128)	0
flatten_11 (Flatten)	(None, 1812608)	0
dense_31 (Dense)	(None, 10)	18126090

Total params: 18,129,674
Trainable params: 18,129,674
Non-trainable params: 0

Accuracy:
【60.14%, 60.01%】

Low Accuracy

Model: "sequential_18"

Layer (type)	Output Shape	Param #
conv2d_48 (Conv2D)	(None, 238, 238, 128)	3584
max_pooling2d_41 (MaxPooling	(None, 119, 119, 128)	0
conv2d_49 (Conv2D)	(None, 117, 117, 64)	73792
max_pooling2d_42 (MaxPooling	(None, 58, 58, 64)	0
flatten_15 (Flatten)	(None, 215296)	0
dense_38 (Dense)	(None, 256)	55116032
dense_39 (Dense)	(None, 10)	2570

Accuracy:
【47.14%, 52.01%】

Underfitting

Layer (type)	Output Shape	Param #
conv2d_50 (Conv2D)	(None, 238, 238, 128)	3584
max_pooling2d_43 (MaxPooling	(None, 119, 119, 128)	0
conv2d_51 (Conv2D)	(None, 117, 117, 64)	73792
max_pooling2d_44 (MaxPooling	(None, 58, 58, 64)	0
conv2d_52 (Conv2D)	(None, 56, 56, 32)	18464
max_pooling2d_45 (MaxPooling	(None, 28, 28, 32)	0
flatten_16 (Flatten)	(None, 25088)	0
dense_40 (Dense)	(None, 256)	6422784
dense_41 (Dense)	(None, 10)	2570

【72.92%, 70.56%】

模型 (Model Illustration)

```
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
_________________________________________________________________
```

Accuracy:【86.52%, 85.46%】

Optimizer = adam

Loss = categorical_crossentropy

Batch size = 32

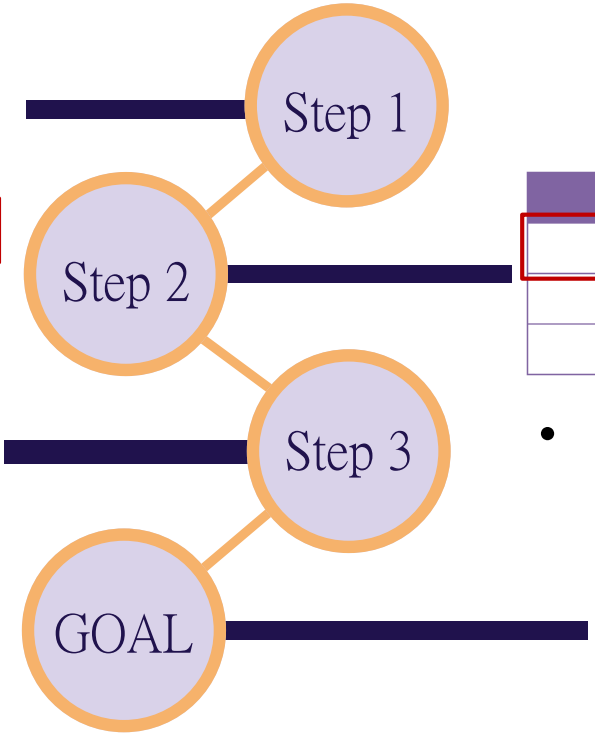
Epoch = 10

Activation Function = Sigmoid

超參數優化 (Hyperparameter Tuning)

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%

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



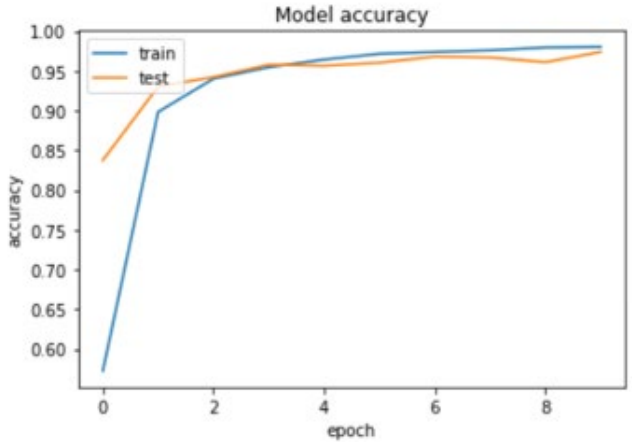
Epoch Times	Training	Testing
10	92.54%	91.34%
20	92.78%	91.54%
30	92.80%	91.72%

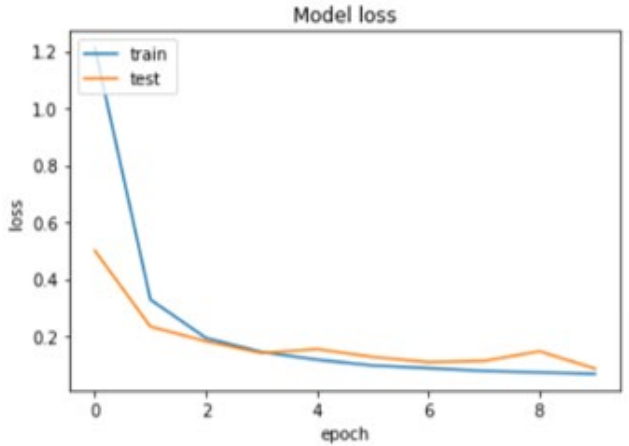
- 考量時間成本

Best Hyperparameter

模型 (Model Illustration)

- 結果有收斂
- 沒有 Overfitting / Underfitting 的現象





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, 10)	8194

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

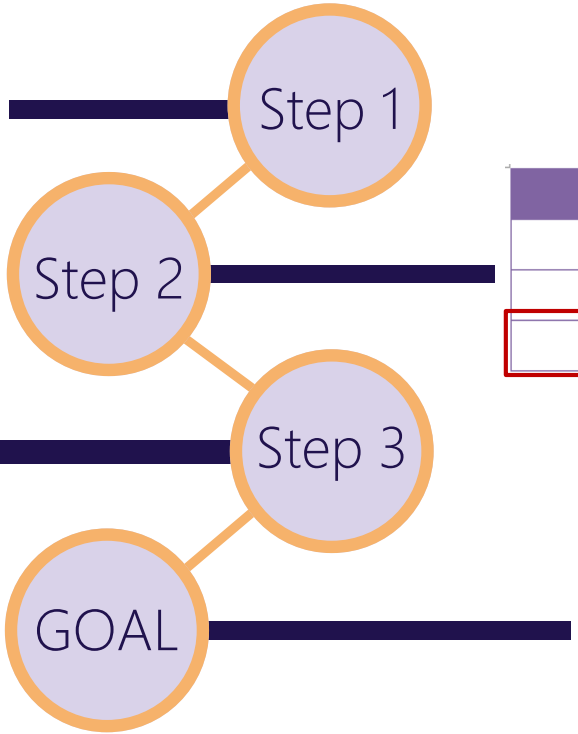
VGG-16 模型

- Accuracy:【86.52%, 85.46%】
- Optimizer = adam
- Loss = categorical_crossentropy
- Batch size = 140
- Epoch = 25
- Activation Function = Sigmoid

超參數優化 (Hyperparameter Tuning)

Epoch Time	Training	Testing
20	84.81%	83.79%
25	86.54%	85.34%
30	86.84%	85.47%

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

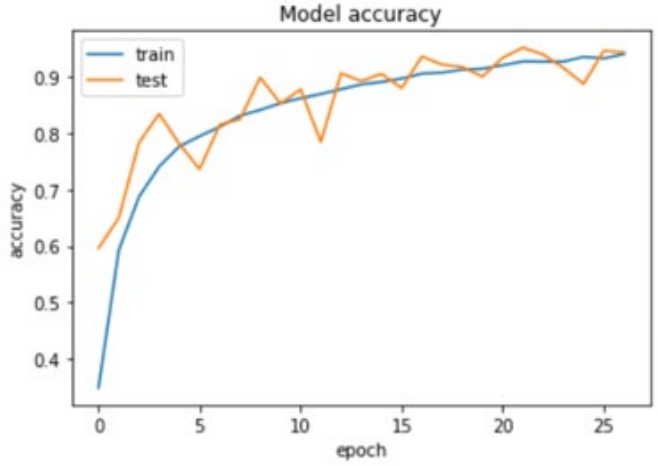


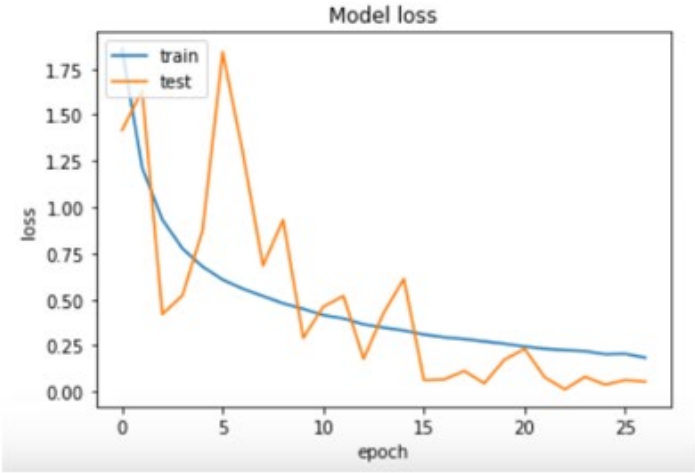
Batch Size	Training	Testing
140	82.14%	82.15%
280	85.41%	84.11%
560	86.54%	85.34%

Best Hyperparameter

VGG -16 模型

- 結果有收斂
- 沒有 Overfitting / Underfitting 的現象



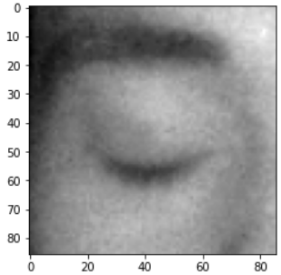


03

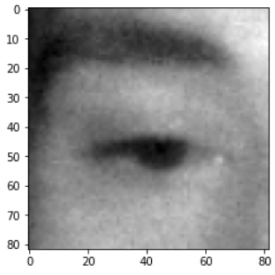
疲勞程度辨識模型

資料探索 (EDA)

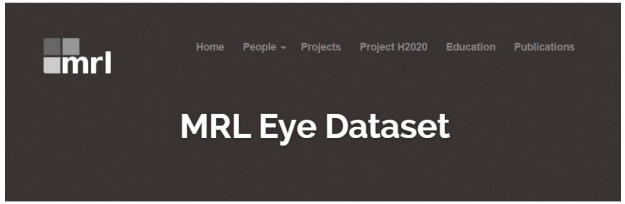
- 77283筆資料集
- 61826 Training, 15457 Test (80% Training , 20% Testing)
- 2 種類別
- 沒有 Imbalanced 問題

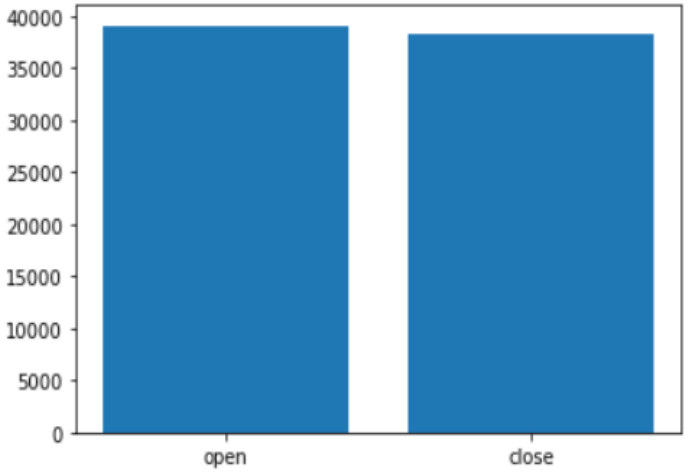


close



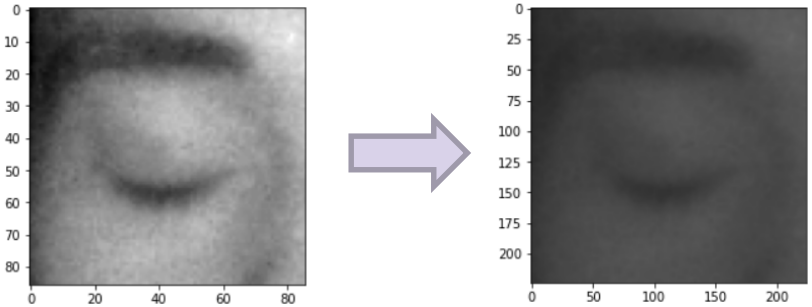
open





資料前處理 (Preprocessing)

Resized，BacktoRGB



86 x 86　　224 x 224

```
Datadirectory = 'C:/Users/Jasper/OneDrive/Desktop/mrlEyes_2018_01/'
Classes = ['close1','open1']
for category in Classes:
    path = os.path.join(Datadirectory,category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img),cv2.IMREAD_GRAYSCALE)
        backtorgb = cv2.cvtColor(img_array,cv2.COLOR_GRAY2RGB)
        plt.imshow(img_array,cmap='gray')
        plt.show()
        break
    break

img_size = 224
new_array = cv2.resize(backtorgb,(img_size,img_size))
plt.imshow(new_array,cmap='gray')
plt.show()
```

模型 (Model Illustration)

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

- Accuracy:【98.79%, 98.95%】
- Computational Time = 1617.4 sec
- Complexity = Medium
- Optimizer = adam
- Loss = binary_crossentropy
- Epoch = 5
- Activation Function = Sigmoid

超參數優化 (Hyperparameter Tuning)

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

Step 1

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

Step 2

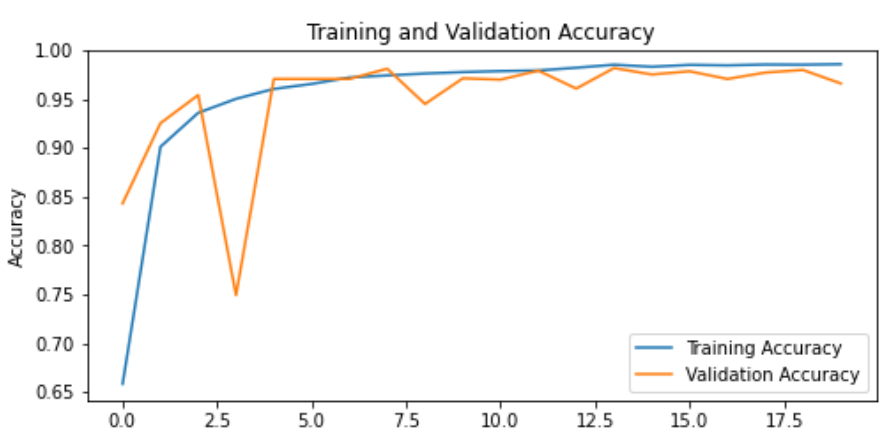
Epoch Time	Training	Testing
5	96.86%	97.39%
10	97.85%	97.41%
20	98.29%	97.50%

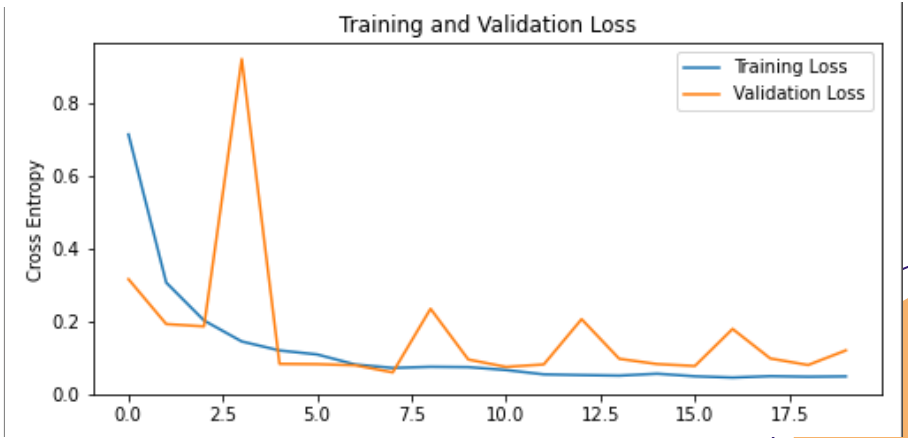
Step 3

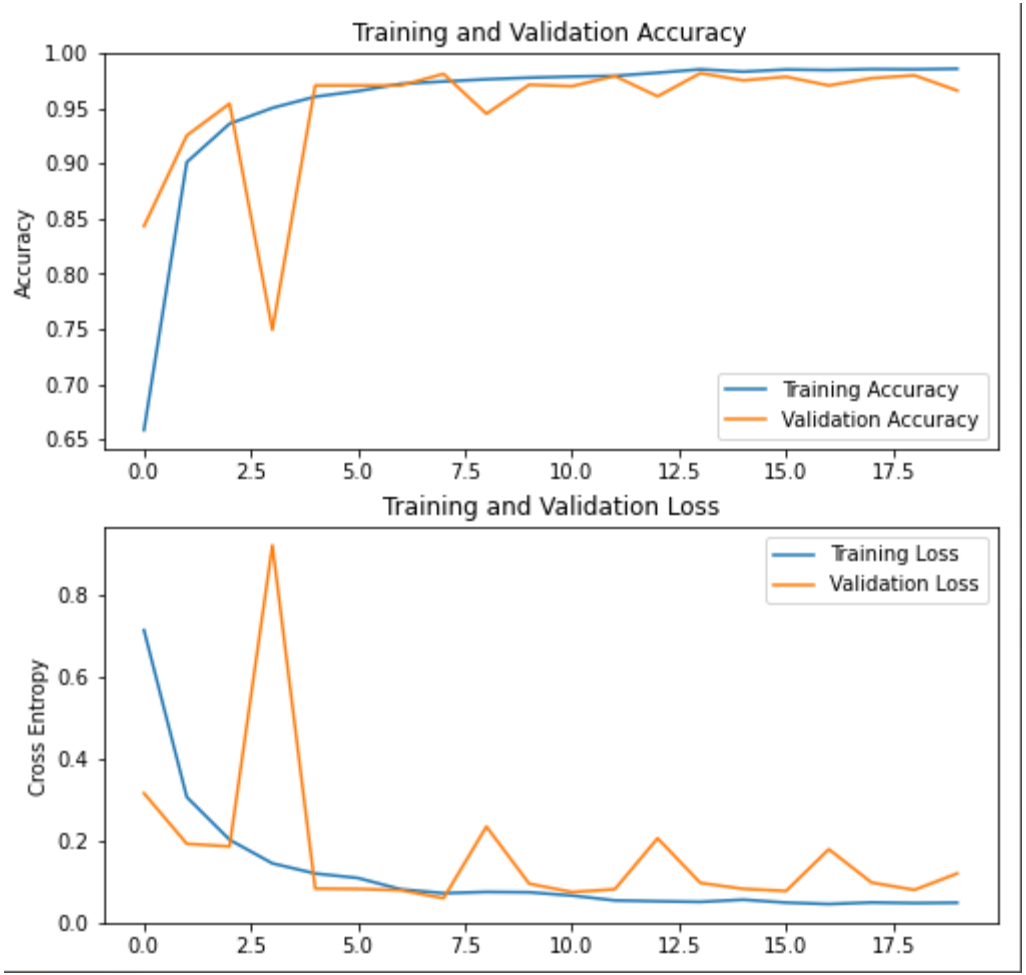
GOAL

Best Hyperparameter

模型 (Model Illustration)






Training and Validation Accuracy
Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Training Accuracy
Validation Accuracy
Training and Validation Loss
Cross Entropy
0.8
0.6
0.4
0.2
0.0
Training Loss
Validation Loss

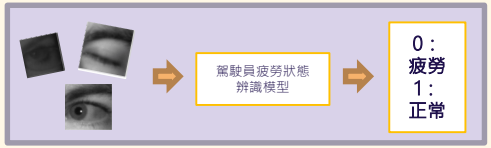
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
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	$14 \times 14 \times 512$
$5\times$ 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	$1 \times 1 \times 1000$

MobileNet 模型

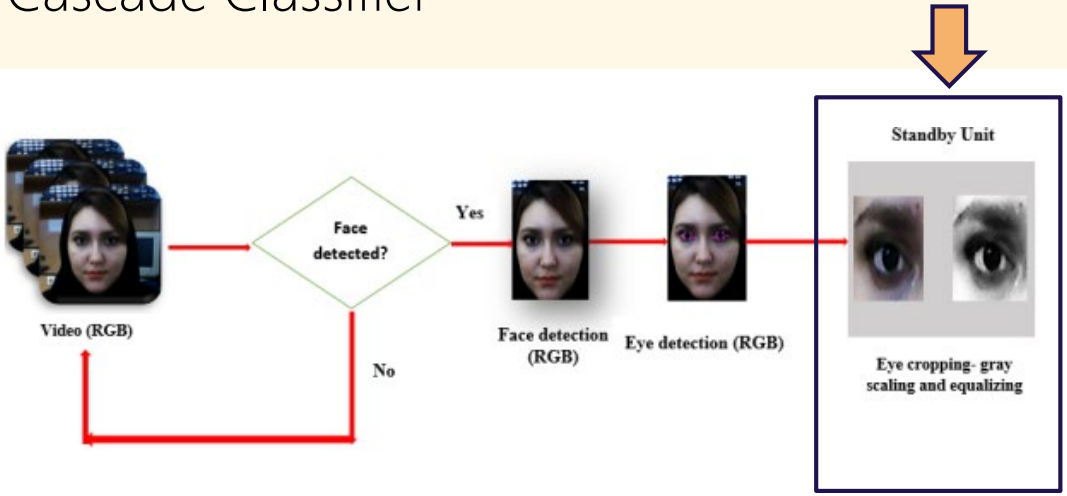
- Accuracy:【98.79%, 98.95%】
- Computational Time = 3856.8 sec
- Complexity = High
- Optimizer = adam
- Loss = binary_crossentropy
- Epoch = 5
- Activation Function = Sigmoid

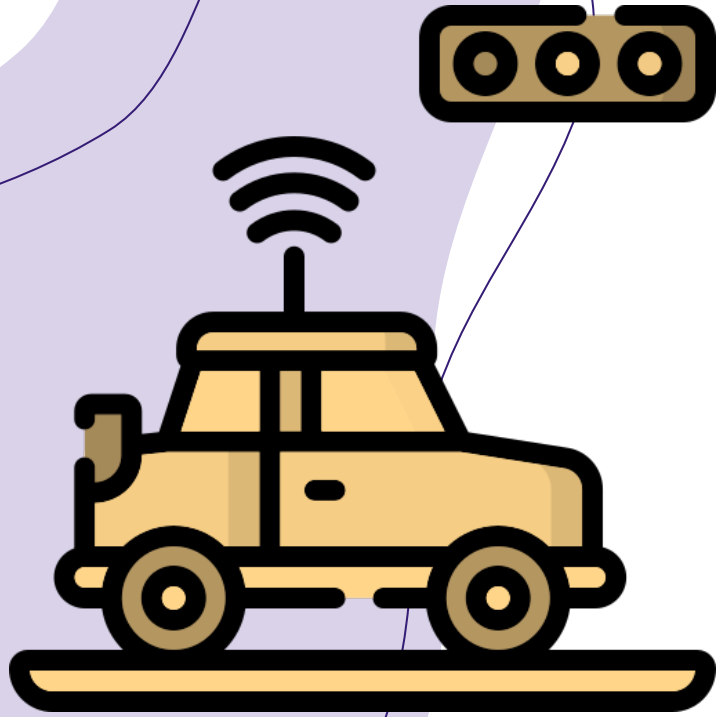
疲勞辨識模型架構

OpenCV: Cascade Classifier



Close Eye > 1.6秒
判定為疲勞(打瞌睡)





結論

模型總結（1/2）

駕駛員行為辨識模型

	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

疲勞辨識模型

	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

總結（2/2）

1. 本研究透過兩種深度學習模型，解決分心導致的車禍問題，並提出一套自動啟動輔助駕駛系統的架構

2. 在駕駛員行為辨識模型，本研究提出的模型架構有更好的準確率

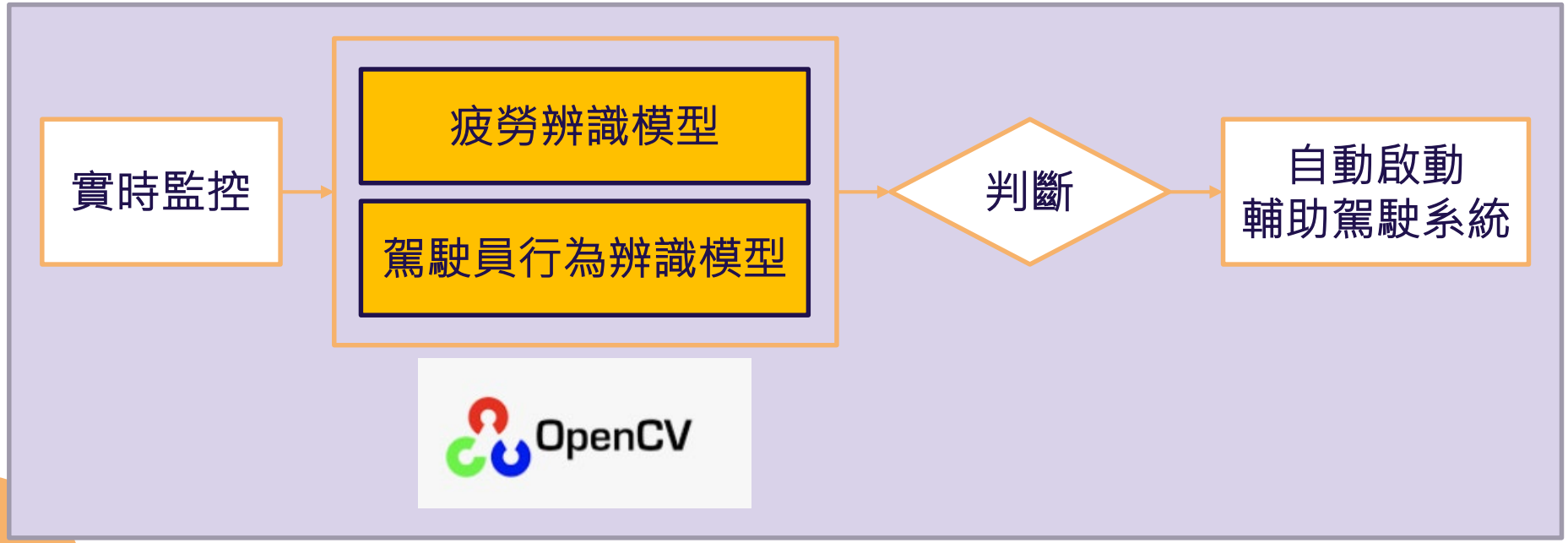
3. 在駕駛員疲勞辨識模型，雖然本研究提出的架構並沒有完全比 MobileNet 的準確率高，但是本研究模型架構的複雜度以及建構時間成本較低也可以得到相近的準確度

4. 未來希望透過不同背景與內裝的照片增加照片的複雜度，訓練更優化的模型

未來應用 (1/3)

實時監控設備

自動啟動輔助駕駛系統架構圖



未來應用 (2/3)

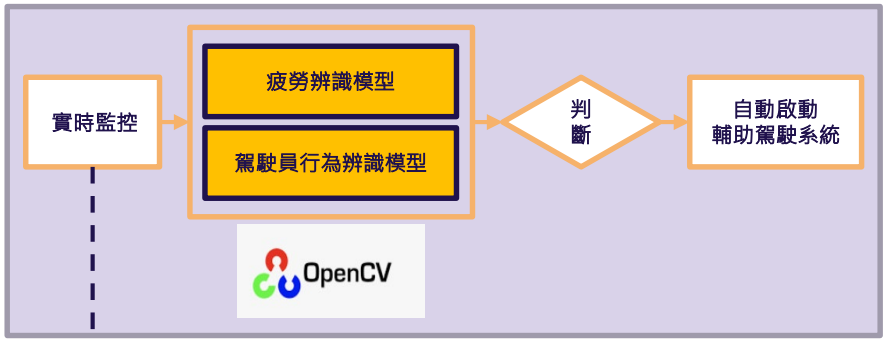
AR 眼鏡實時監控

疲勞辨識模型

OpenCV

發出警報提高警惕

未來應用 (3/3)



Real time User feedback with T&C

開發商
Database

Big Data Analysis

Voice of Customer

- Talking on the phone: left
- Talking on the phone: right
- Texting: left hand
- Texting: right hand
- Adjust the radio
- Drinking
- Reaching behind
- Hair and Make up

REFERRENCE

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

Thanks
Q & A