

【智慧企業整合】

Project 2

Turn 口罩?!

Group 3

110034560 王勇盛

110034563 許宇韶

108005522 包中南

110034546 胡瑜茹

指導教授：邱銘傳 教授

中華民國 110 年 12 月 17 日

目錄

一、	背景說明.....	4
	(一)、現況描述	4
	(二)、問題描述 5W1H	4
二、	資料介紹、處理	4
	(一)、資料內容	4
	(二)、資料處理	6
三、	模型介紹.....	6
	(一)、Haar Cascade	6
	(二)、Resnet50	7
	(三)、VGG19	8
四、	訓練過程.....	10
	(一)、Resnet50	10
	(二)、VGG19	12
五、	結論與展望	17
	(一)、結論	17
	(二)、展望	18
六、	參考資料.....	19

圖目錄

圖 1. 辨識是否符合社交距離	錯誤! 尚未定義書籤。
圖 2. 辨識是否符合社交距離	5
圖 3. Data augmentation.....	6
圖 4. Resnet50 架構圖	7
圖 5. VGG 架構	9
圖 6. Resnet50 模型訓練結果	11
圖 7. Resnet50 迭代 20 次結果	11
圖 8. Resnet50 迭代 100 次結果	12
圖 9. Resnet50 測試集結果.....	12
圖 10. VGG19 模型架構.....	13
圖 11. VGG19 模型訓練.....	13
圖 12. VGG19 可視化損失函數(左)、可視化準確率(右)	14
圖 13. VGG19 測試資料.....	16
圖 14. VGG19 測試結果.....	16
圖 15. 你保持社交距離了嗎.....	17
圖 16. 你戴口罩了嗎.....	18

表目錄

表 1. 5W1H.....	<u>4</u>
表 2. Resnet50 參數調整.....	<u>10</u>
表 3. VGG19 參數調整.....	<u>14</u>

一、 背景說明

(一)、現況描述

2020 年初全球爆發 COVID-19 疫情，造成重大傷亡與損失，也改變了人們生活的習慣，為避免疫情的傳播，同時也保護自我健康，口罩的使用已融入我們生活之中。近期 Omicron 出現使指揮中心再次呼籲，民眾應落實佩戴口罩等個人防護措施。

(二)、問題描述 5W1H

我們的目標是建立一個深度學習模型，該模型可以識別該人是否戴口罩，還可以檢測人們是否違反了社交距離規範。社交距離，是指與其他非家人關係的人保持安全空間。要進行社交距離必須在室內和室外空間與非家人關係的人保持至少 1.5 公尺（約 2 臂長）的距離。

表 1. 5W1H

What	你戴口罩了嗎?你保持社交距離了嗎?
When	每一秒
Who	每一個人
Where	每一個地方
Why	落實防疫政策
How	深度學習、影像處理

二、 資料介紹、處理

(一)、資料內容

由 Kaggle 公開數據集中取得 [Face Mask Detection](#) 的圖像資料集，資料集共分成 3 種類別：

- 帶口罩
- 不戴口罩
- 口罩佩戴不正確

利用此圖像資料集作為辨別該圖是否符合社交距離。

圖 1. 辨識是否符合社交距離



由 Kaggle 公開數據集中取得 Face Mask Detection ~12K Images Dataset 的圖像資料集，分成 3 個資料集分別為：

- 訓練集資料
- 驗證集資料
- 測試集資料

每一資料集共分成 2 種類別：

- 帶口罩
- 無戴口罩

利用此圖像資料集作為判斷該圖是否戴口罩。

圖 2. 辨識是否符合社交距離



(二)、資料處理

為了避免 Overfitting 的情況將訓練集資料，透過 Data augmentation 的方式增加訓練及驗證張數，其生成的張數會依照不同 batch size 的大小在每一個 epoch 而有變化其生成張數的設定公式為原有訓練的張數乘上所設定的倍數除以 batch size 的商數為 Data augmentation 所生成的張數。

Keras 透過 ImageDataGenerator class 提供 Data augmentation 相關的功能如：

- 資料的正規化：我們的圖像在 RGB 通道都是 0~255 的整數，這樣的操作可能使圖像的值過高或過低，所以我們將這個值定為 0~1 之間的數。
- 進行隨機水平翻轉：隨機的對圖片進行水平翻轉
- 偏移：剪切強度（逆時針方向的剪切變換角度）。是用來進行剪切變換的程度。
- 放大縮小：浮點數或形如 [lower, upper] 的列表，隨機縮放的幅度，若為浮點數，則相當於 [lower, upper] = [1 - zoom_range, 1+zoom_range]。

圖 3. Data augmentation

```
train_datagen = ImageDataGenerator(rescale=1.0/255,
                                   horizontal_flip=True,
                                   zoom_range=0.2,
                                   shear_range=0.2)
train_generator = train_datagen.flow_from_directory(directory=train_dir,
                                                    target_size=(128,128),
                                                    class_mode='categorical',
                                                    batch_size=32)
```

三、模型介紹

(一)、Haar Cascade

Haar Cascade 是「Haar Feature-based Cascade Classifier」的縮寫，使用 Haar Cascade 作為分類器的物件偵測是 Paul Viola 和 Michael Jones 在他們 2001 年的論文“Rapid Object Detection using a Boosted Cascade of Simple Features”中提出的一種有效的物件偵測方法。它是一種基於機器學習的方法，其中級聯函數是從大量

正樣本(與真實標籤一致)和負樣本(與真實標籤不一致)圖像中訓練出來的。然後使用它來檢測其他圖像中的對象。我們將使用經過訓練的 Haar Cascade 模型來檢測人臉，以獲得圖像中人臉的邊界框坐標。

(二)、Resnet50

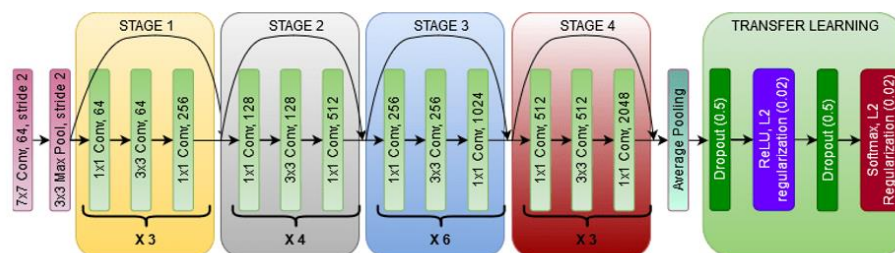
ResNet 全名為 Deep Residual Neural Network，中文名為「殘差網路」，是 2015 年的 ILSVRC 比賽冠軍。ResNet 的優點是它在 VGG 的基礎上，新增直連通路(skip connection)，不會新增任何參數，但是可以解決梯度消失的問題，但其缺點需要較久的訓練時間。Resnet50 架構如下：

1. conv1：7x7 卷積核(共 1 層)
2. conv2_x：由三種卷積核(1x1, 3x3, 1x1)組成一個 block，疊 3 次。(共 9 層)
3. conv3_x：由三種卷積核(1x1, 3x3, 1x1)組成一個 block，疊 4 次。(共 12 層)
4. conv4_x：由三種卷積核(1x1, 3x3, 1x1)組成一個 block，疊 6 次。(共 18 層)
5. conv5_x：由三種卷積核(1x1, 3x3, 1x1)組成一個 block，疊 3 次。(共 9 層)
6. output：由 pooling+全連接層+softmax 組成。(共 1 層)

上方這六個部分加起來共 50 層。

模型加深可以提高準確率，但是加深會有許多困難要克服，例如：1. 參數量變大、2. 推論時間變久、3. 梯度消失嚴重、4. 硬體設備不足。這些困難導致模型無法訓練成功。而殘差網路解決了 1. 和 3.，但它也只是模型加深的帶來的困難，而非完全解決。

圖 4. Resnet50 架構圖



(三)、VGG19

VGG 有兩種結構，分別是 VGG16 和 VGG19，兩者並沒有本質上的區別，只是網絡深度不一樣。

VGG 原理：

VGG16 是採用連續的幾個 3×3 的卷積核代替較大卷積核 (11×11 , 7×7 , 5×5)。採用堆積的小卷積核是優於採用大的卷積核，因為多層非線性層可以增加網絡深度來保證學習更復雜的模式，而且代價還比較小（參數更少）。

簡單來說，在 VGG 中，使用了 3 個 3×3 卷積核來代替 7×7 卷積核，使用了 2 個 3×3 卷積核來代替 5×5 卷積核，這樣做的主要目的是在保證具有相同的條件下，提升了網絡的深度，在一定程度上提升了神經網絡的效果。

比如，3 個步長為 1 的 3×3 卷積核的一層層疊加作用可看成一個大小為 7 的感受野（其實就表示 3 個 3×3 連續卷積相當於一個 7×7 卷積），其參數總量為 $3 \times (9 \times C^2)$ ，如果直接使用 7×7 卷積核，其參數總量為 $49 \times C^2$ ，這裏 C 指的是輸入和輸出的通道數。很明顯， $27 \times C^2$ 小於 $49 \times C^2$ ，即減少了參數；而且 3×3 卷積核有利於更好地保持圖像性質。這裏解釋一下為什麼使用 2 個 3×3 卷積核可以來代替 5×5 卷積核： 5×5 卷積看做一個小的全連接網絡在 5×5 區域滑動，我們可以先用一個 3×3 的卷積濾波器卷積，然後再用一個全連接層連接這個 3×3 卷積輸出，這個全連接層我們也可以看做一個 3×3 卷積層。這樣我們就可以用兩個 3×3 卷積級聯（疊加）起來代替一個 5×5 卷積。

VGG 網絡結構：

圖 5. VGG 架構

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

含義：

1. conv 表示卷積層
2. FC 表示全連接層
3. conv3 表示卷積層使用 3x3 filters
4. conv3-64 表示深度 64
5. maxpool 表示最大池化

VGG19 包含了 19 個隱藏層（16 個卷積層和 3 個全連接層），如上圖中的 E 列所示。

VGG 優缺點：

優點：

- VGGNet 的結構非常簡潔，整個網絡都使用了同樣大小的卷積核尺寸（3x3）和最大池化尺寸（2x2）。
- 幾個小濾波器（3x3）卷積層的組合比一個大濾波器（5x5 或 7x7）卷積層好：驗證了通過不斷加深網絡結構可以提升性能。

缺點：

- VGG 耗費更多計算資源，並且使用了更多的參數，導致更多的內存佔用。

四、 訓練過程

(一)、Resnet50

資料集分為訓練、驗證、測試三種，經過 Data augmentation 後分別為 10000 張、800 張、800 張，首先以訓練和驗證資料對模型做訓練。

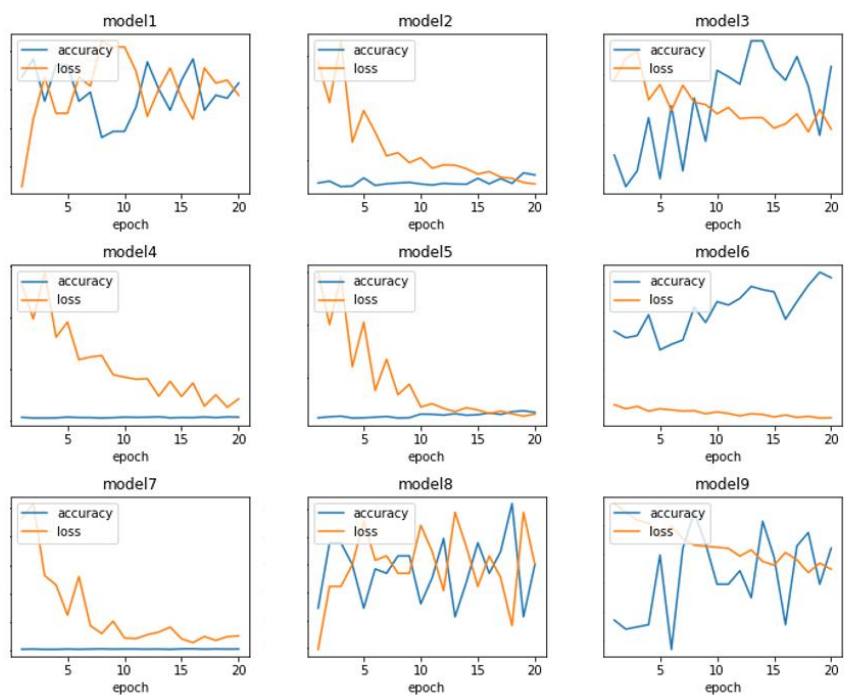
將優化器、學習率、損失函數訂定為三種水準，組成 9 種不同的實驗模型，固定 epochs 為 20 次，觀察那一組跑的參數最佳，再挑選其模型繼續訓練。

表 2. Resnet50 參數調整

Model	Optimizers	Learning Rate	Loss
1	SGD	0.1	mean_squared_error
2		0.01	binary_crossentropy
3		0.001	categorical_crossentropy
4	Adagrad	0.1	binary_crossentropy
5		0.01	categorical_crossentropy
6		0.001	mean_squared_error
7	Adam	0.1	categorical_crossentropy
8		0.01	mean_squared_error
9		0.001	binary_crossentropy

由下方圖形可發現，model3、model6、model9 準確率和損失率都有上升和下降，由這 3 個模型可知，學習率設定在 0.001 是可以有效訓練，其中 model6 表現最佳。

圖 6. Resnet50 模型訓練結果



針對 model6 再另外進行 epoch 為 100 次的訓練，模型的準確率從原先的 72.2% 提升到 82.4%。

圖 7. Resnet50 迭代 20 次結果

```
k = np.array(historys[6].history['accuracy'])  
k  
  
array([[0.53125 , 0.5078125 , 0.515625 , 0.58984375, 0.46484375,  
       0.484375 , 0.5 , 0.6171875 , 0.5625 , 0.63671875,  
       0.625 , 0.6484375 , 0.69140625, 0.6796875 , 0.671875 ,  
       0.57421875, 0.63671875, 0.6953125 , 0.7421875 , 0.72265625])
```

圖 8. Resnet50 迭代 100 次結果

```
k = np.array(historys[6].history['accuracy'])
k
array([0.4765625 , 0.51171875, 0.53515625, 0.55859375, 0.50390625,
       0.5703125 , 0.51171875, 0.72265625, 0.6171875 , 0.6875   ,
       0.6796875 , 0.6953125 , 0.62109375, 0.62890625, 0.59375   ,
       0.6484375 , 0.625    , 0.59375   , 0.73046875, 0.72265625,
       0.59765625, 0.7578125 , 0.7578125 , 0.73828125, 0.62109375,
       0.765625  , 0.765625  , 0.75     , 0.65234375, 0.73046875,
       0.609375  , 0.72265625, 0.64453125, 0.74305558, 0.73046875,
       0.77734375, 0.62890625, 0.72265625, 0.76953125, 0.81640625,
       0.8203125 , 0.83203125, 0.79296875, 0.796875  , 0.796875  ,
       0.80859375, 0.828125  , 0.8671875 , 0.81640625, 0.66796875,
       0.84765625, 0.8046875 , 0.83984375, 0.80859375, 0.78125   ,
       0.80078125, 0.78125   , 0.796875  , 0.80859375, 0.828125  ,
       0.7421875 , 0.796875  , 0.83984375, 0.7890625 , 0.78515625,
       0.828125  , 0.82421875, 0.84375   , 0.88671875, 0.8125    ,
       0.80078125, 0.796875  , 0.875     , 0.86328125, 0.78515625,
       0.890625  , 0.8671875 , 0.84375   , 0.78515625, 0.796875  ,
       0.83984375, 0.82421875, 0.84765625, 0.85546875, 0.8515625  ,
       0.78515625, 0.84375   , 0.79296875, 0.84765625, 0.81640625,
       0.859375  , 0.875     , 0.859375  , 0.80078125, 0.87109375,
       0.86328125, 0.875     , 0.83203125, 0.83203125, 0.82421875])
```

圖 9. Resnet50 測試集結果

```
model.evaluate(test_generator)
7/7 [=====] - 24s 3s/step - loss: 0.1464 - accuracy: 0.8438
[0.14643320441246033, 0.84375]
```

為了能確認模型的泛化性，在測試資料上模型準確率為 84.3%。

(二)、VGG19

由於所使用的 Resnet50 模型在調整多次超參數後，都無法超過 90%的準確率，所以試著利用不同模型，這裡使用官方已經打包好的 VGG19 的模型來檢視是否能有所改善。

模型層級與層級說明：

- 模型初始化
- 層級說明 Keras model 使用 Keras 提供之 CNN 模型(VGG19)
- 全連接層 Flatten
- 輸出層 Dense(2, activation=" sigmoid")
- model.summary()如下圖：

圖 10. VGG19 模型架構

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 4, 4, 512)	20024384
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 2)	16386

```
Total params: 20,040,770  
Trainable params: 16,386  
Non-trainable params: 20,024,384
```

為了有利與 Resnet50 進行客觀比較，因此同樣固定 epochs 為 20 次，利用 VGG19 模型訓練過程如下圖所示：

圖 11. VGG19 模型訓練

```
Epoch 1/20  
9/9 [=====] - 7s 656ms/step - loss: 0.5481 - accuracy: 0.7292  
Epoch 2/20  
9/9 [=====] - 5s 543ms/step - loss: 0.2207 - accuracy: 0.9228  
Epoch 3/20  
9/9 [=====] - 6s 601ms/step - loss: 0.1427 - accuracy: 0.9549  
Epoch 4/20  
9/9 [=====] - 7s 735ms/step - loss: 0.1521 - accuracy: 0.9479  
Epoch 5/20  
9/9 [=====] - 6s 592ms/step - loss: 0.1245 - accuracy: 0.9549  
Epoch 6/20  
9/9 [=====] - 6s 608ms/step - loss: 0.0824 - accuracy: 0.9861  
Epoch 7/20  
9/9 [=====] - 6s 609ms/step - loss: 0.0886 - accuracy: 0.9653  
Epoch 8/20  
9/9 [=====] - 7s 726ms/step - loss: 0.0991 - accuracy: 0.9688  
Epoch 9/20  
9/9 [=====] - 6s 694ms/step - loss: 0.1035 - accuracy: 0.9618  
Epoch 10/20  
9/9 [=====] - 7s 763ms/step - loss: 0.0842 - accuracy: 0.9792  
Epoch 11/20  
9/9 [=====] - 6s 641ms/step - loss: 0.0831 - accuracy: 0.9757  
Epoch 12/20  
9/9 [=====] - 7s 622ms/step - loss: 0.0620 - accuracy: 0.9792  
Epoch 13/20  
9/9 [=====] - 6s 629ms/step - loss: 0.0644 - accuracy: 0.9861  
Epoch 14/20  
9/9 [=====] - 7s 717ms/step - loss: 0.0493 - accuracy: 0.9861  
Epoch 15/20  
9/9 [=====] - 6s 651ms/step - loss: 0.0570 - accuracy: 0.9826  
Epoch 16/20  
9/9 [=====] - 6s 662ms/step - loss: 0.0408 - accuracy: 0.9896  
Epoch 17/20  
9/9 [=====] - 6s 649ms/step - loss: 0.0573 - accuracy: 0.9861  
Epoch 18/20  
9/9 [=====] - 6s 672ms/step - loss: 0.0455 - accuracy: 0.9965  
Epoch 19/20  
9/9 [=====] - 6s 677ms/step - loss: 0.0488 - accuracy: 0.9896  
Epoch 20/20  
9/9 [=====] - 7s 709ms/step - loss: 0.0700 - accuracy: 0.9688
```

將利用 VGG19 模型訓練過程透過可視化表示迭代次數與損失函數和迭代次數與準確率關係如下圖所示，我們可以看到在 VGG19 模型訓練過程隨著迭代次數增加，我們能夠有效降低損失函數同時提升準確率。

圖 12. VGG19 可視化損失函數(左)、可視化準確率(右)

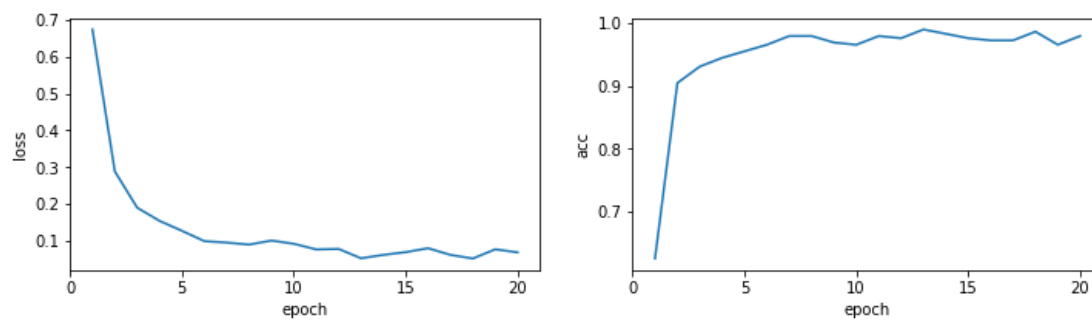


表 3. VGG19 參數調整

Model	Optimizers	Learning Rate	Loss	Accuracy
1	SGD	0.1	mean_squared_error	98.6%
		0.01		98.2%
		0.001		98.1%
2	SGD	0.1	binary_crossentropy	98.5%
		0.01		97.7%
		0.001		98%
3	SGD	0.1	categorical_crossentropy	96.8%
		0.01		98.3%
		0.001		98.3%
4	Adagrad	0.1	mean_squared_error	98.8%
		0.01		99.2%
		0.001		98.2%
5	Adagrad	0.1	binary_crossentropy	97.8%
		0.01		98.7%
		0.001		98.3%
6	Adagrad	0.1	categorical_crossentropy	98.8%
		0.01		99%
		0.001		98.7%
7	Adam	0.1	mean_squared_error	50%
		0.01		
		0.001		
8	Adam	0.1	binary_crossentropy	98.5%
		0.01		98.8%
		0.001		99.3%
9	Adam	0.1	categorical_crossentropy	97.2%
		0.01		98.7%
		0.001		99.7%

利用測試資料集測試訓練過後的模型，最終結果辨識率可達到 99.7% 準確率，損失函數為 0.02%，表示我們可以利用訓練好的模型進行口罩辨識。

圖 13. VGG19 測試資料

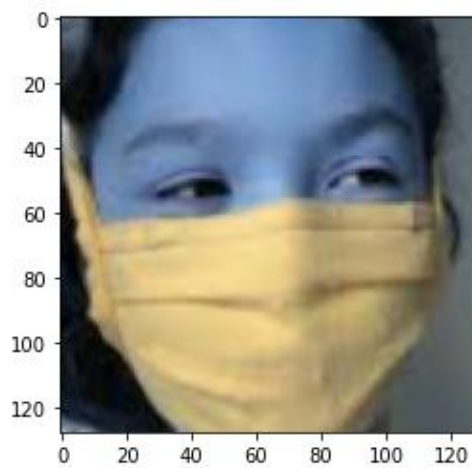


圖 14. VGG19 測試結果

```
array([[0.9972728 , 0.00292024]], dtype=float32)
```

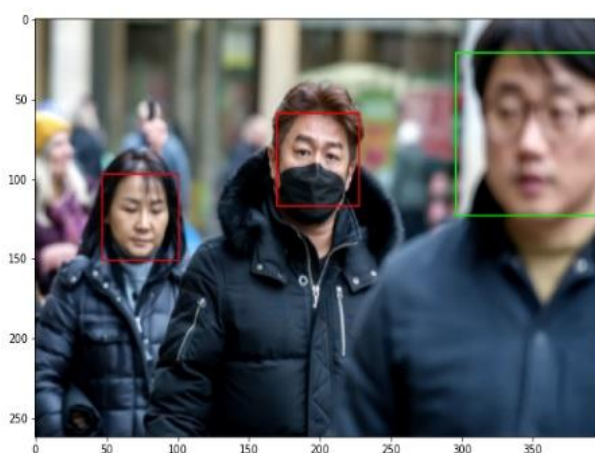
五、 結論與展望

(一)、 結論

美國和歐洲的許多城市現在都在謹慎地重新開放。人們被要求在外出時保持安全距離。但是人們照著做嗎？城市對人們的安全距離是否符合規則進行評估並採取相應的行動是很重要的。如果大多數人都遵守疫情期間的命令，那麼就可以安全地開放更多的公共場合。

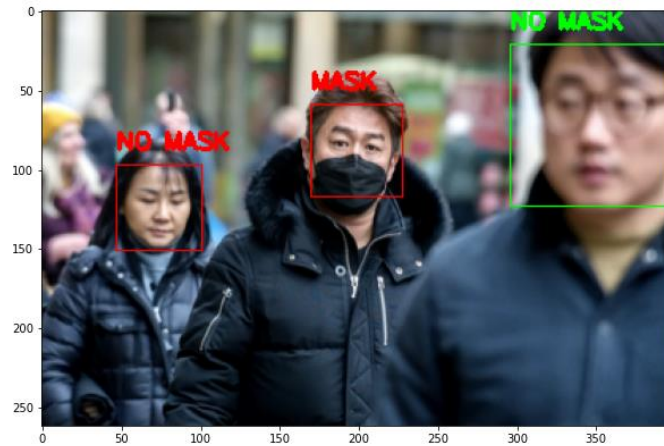
我們如何使用人工智慧和機器學習來檢測人們是否遵循社交距離規則？大多數城市已經在公共場所安裝了攝像頭，這些攝像頭正好可以用於此目的。

圖 15. 你保持社交距離了嗎



由模型實作成果圖可以看到，有戴口罩者即被綠色框框住，並標註「MASK」；而無戴口罩者則被紅色框框住，並標註「NO MASK」。一旦原本有戴口罩被綠色框住的對象脫下口罩，系統即時自動將綠色框轉為紅色框；反之，當沒有戴口罩的對象戴上口罩，紅色框會自動即時將紅色框轉為綠色框，VGG19 的在測試集的準確率與 Resnet50 在測試集的準確率相比起來準確率高 15%，因此可見 VGG19 針對本資料在辨識口罩配戴上有較佳效果。

圖 16. 你戴口罩了嗎



(二)、展望

在全球疫情還未正式趨緩前，員工健康管理仍為企業復工後的首要任務，面對每日人潮眾多的辦公大樓，如果額外安排人力量測進出人員的體溫、手動進行酒精消毒，不僅會在上班尖峰時間造成排隊耗時的狀況，更會徒增人事成本以及集體感染的風險。

此外，台灣企業運行多年的上下班打卡鐘以及門禁卡仍存在許多缺點，包含同事冒名代刷卡、員工忘記帶卡而需事後申請補打卡，以及出差勤打卡時間異常而造成薪資計算錯誤等突發事件。縱使後續有了指紋門禁考勤的新機制，仍免不了有心人士自製指紋模打卡，且在疫情期間，指紋辨識為接觸性生物辨識技術，更是感染病毒的一大疑慮。

人臉辨識結合門禁打卡系統，有效管理員工出勤、安全與健康

近幾年，人臉辨識的應用範圍越來越廣，為了有效防範新冠疫情的蔓延，許多企業開始逐步導入人臉辨識門禁管理系統來取代原先傳統的門禁刷卡機，例如指紋按壓、輸入密碼或刷卡等方式，透過人臉辨識門禁機，企業可以大大降低接觸風險，且有效率地管理員工出缺勤狀況。

在單不離口的新常態底下，透過最新的人臉辨識技術，整合門禁管理系統與硬體，打造一站式的人臉辨識考勤機與智慧健康偵測站，不但能有效掌握人員進出，更可以監測每個人的口罩配戴狀況，並測量體溫，同時達到人事考勤、公衛安全，以及人流管理的目的。

六、 參考資料

- https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
- <https://www.kaggle.com/nageshsingh/mask-and-social-distancing-detection-using-vgg19>
- <https://keras.io/zh/preprocessing/image/>
- <https://ithelp.ithome.com.tw/articles/10192162>