

## 期末專題書面報告

### Final Project: Face Mask Detector with Deep Learning

60947089S 李佩穎/60947100S 宋立晴 國立臺灣師範大學資訊工程學系

#### 壹、摘要

類神經網路期末專題，把之前老師教導給我們的課程運用在此次的專題上，題目可以選擇老師推薦的比賽或是自己發想一個主題，而我們這組選擇自己發想一個主題—Face Mask Detector with Deep Learning。

#### 貳、介紹

專題發想是因為最近疫情日趨嚴重，戴口罩可以保護自己也可以保護別人，如果可以自動偵測有無戴口罩可以省去監督的麻煩也可以提醒沒戴口罩的人戴口罩，使用由 Prajna Bhandary 創建的 Dataset 進行影像訓練 [1] 如圖 1 所示，運用 openCV 結合 Deep Learning 進行實作。

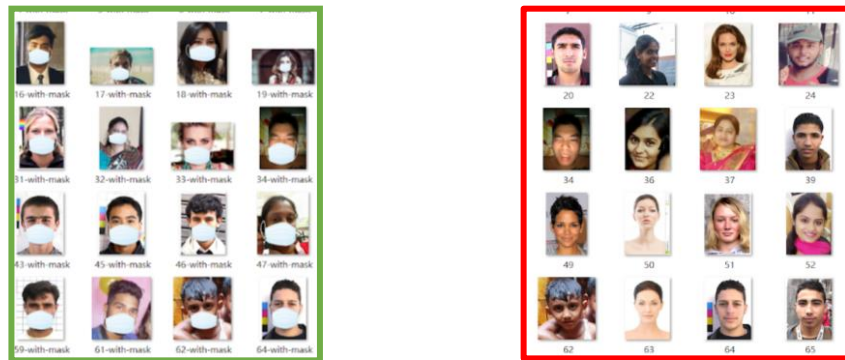


圖 1 左圖為手工加上口罩 dataset 共 690 張影像，右圖為臉部 dataset 共 686 張影像

#### 參、方法與實驗

方法是參考 [2] 的內容製作的流程如圖 2 所示，以下會介紹每個階段的實際做法內容。

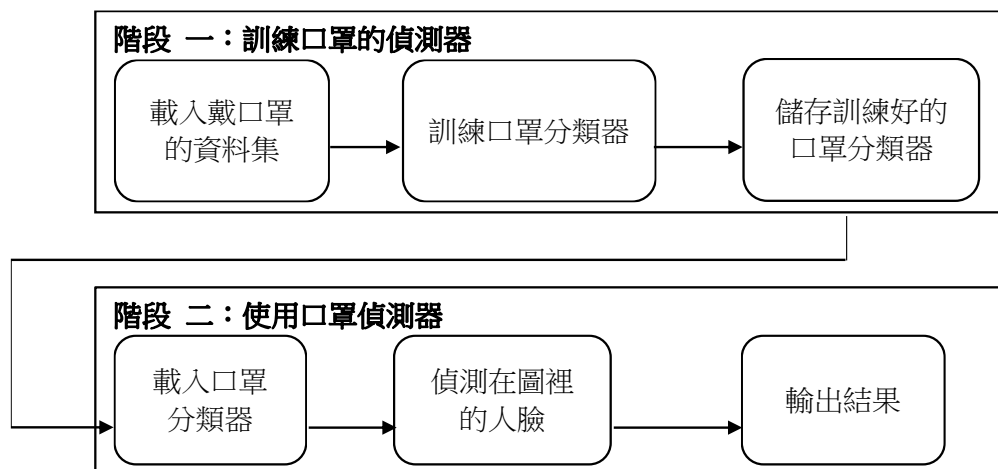


圖 2 方法流程

此流程其實類似於臉部辨識的方法，但是差別在於要偵測的是人的臉部是否有戴口罩。

階段一：

首先是要進行影像前處理主要是要提升準確度，使用 Data Augmentation 可以達成此效果，如圖 3 所示，另外還做了包括調整大小為 224x224 像素，轉換為陣列格式，以及將輸入影像中的像素亮度縮放到範圍  $[-1, 1]$ ，再來是對 label 進行 one-hot encoding，目的是要劃分 dataset，最後把這些 dataset 用來訓練。

One-hot encoding：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，為每個類別新增一個欄位，用 0/1 表示是否，當某一系列的資料存在的該行的類別則顯示 1，反則顯示 0。[3]

```
# 數據增強
aug = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest")
```

圖 3 Data Augmentation 包含隨機旋轉、縮放、裁切、位移和翻轉

再來進行訓練分類器，使用 Google 發表的 MobileNetV2 進行訓練 [4]，並且進行微調，使用此類神經網路的好處在於它是適合用於圖像分類、可以在有限的硬體設備下使用，載入 MobileNetV2 分類器，使用預訓練的 ImageNet[5] 權重對該模型進行微調。

ImageNet：是一個按照 WordNet 層次結構（目前只有名詞）組織的圖像數據庫，其中層次結構的每個節點都由成百上千的圖像來描繪。

WordNet：是一個大型英語詞彙數據庫。

修改原本的 fully-connected 層 (圖 4)，最後使用此類神經網路進行訓練，而為了要將基礎層的權重在 backpropagation 過程中不會更新，而頭層的權重將被調整，所以凍結 network 的基礎層 (圖 5)。

```
# 修改原本的fully-connected層
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
```

圖 4

```
for layer in baseModel.layers:
    layer.trainable = False
```

圖 5

微調完成後再來進行訓練我們的口罩偵測器網路，首先 compile 我們的模型與 Adam optimizer、learning rate decay schedule 和 binary cross-entropy，再來就是訓練模型了。

訓練好我們的模型將在測試集上評估我們生成的模型，對測試集進行 prediction，獲取機率最高的類別索引標籤並進行檢查，若機率值不錯則把模型儲存到.h5 的文件中，訓練結果可以知道此模型有 99%的準確率。

階段二：

載入訓練好的人臉偵測跟口罩偵測模型，且讓輸入的圖像大小為 300\*300 並且執行 mean subtraction，當偵測到影像當中的臉部時進行定位，讓我們偵測臉部是否有戴口罩。

判斷有無口罩，使用循環 detections 和擷取 confidence 來衡量 confidence 的 threshold，然後計算 Box 的邊界特定人臉的值並確保框有落在圖像的邊界內，再來針對人臉提取 ROI 值[6]，而 ROI 值是用來判斷是否有戴口罩的值，分別會在有口罩跟沒口罩的值，當有口罩的值大於沒口罩的值則為戴口罩，框住臉的 Box 會呈現綠色否則為紅色，而 Box 的頂端會顯示機率值(%)。

#### 肆、實驗結果

訓練結果如圖：

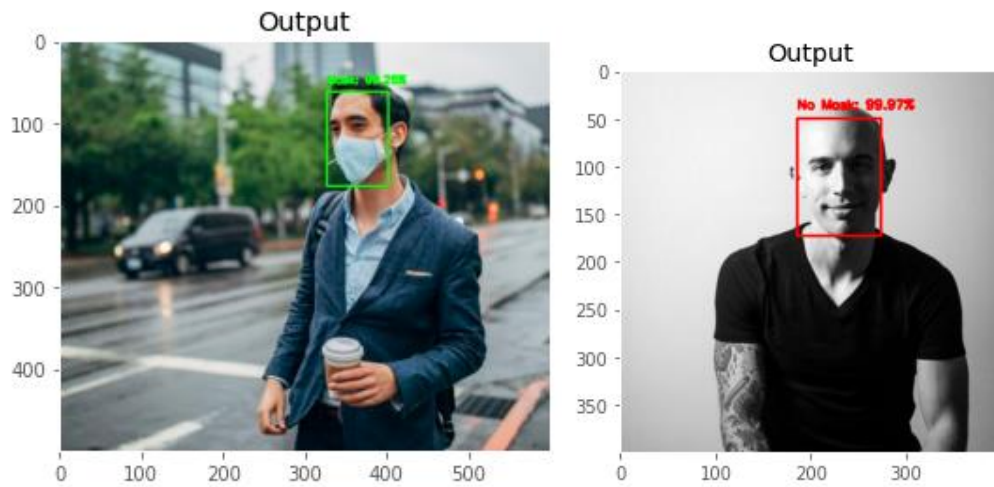
	precision	recall	f1-score	support
with_mask	0.99	0.99	0.99	138
without_mask	0.99	0.99	0.99	138
accuracy			0.99	276
macro avg	0.99	0.99	0.99	276
weighted avg	0.99	0.99	0.99	276

[INFO] saving mask detector model...

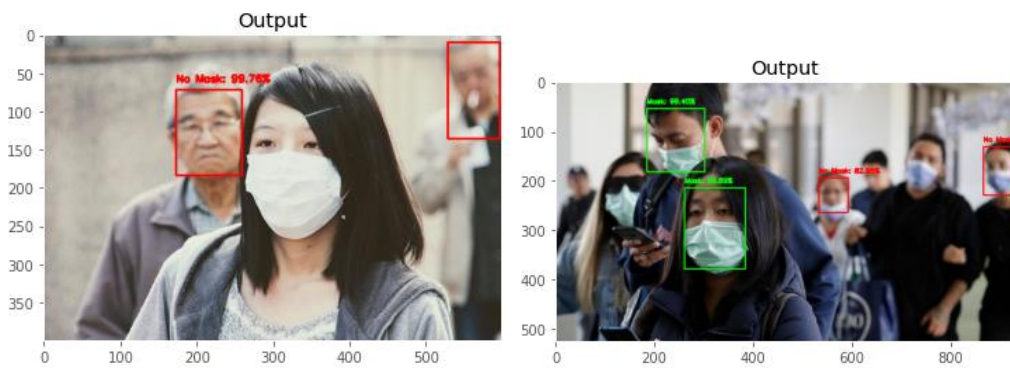


準確度高達 99%且沒有 overfitting 的問題

輸出影像結果  
成功的結果：



失敗的結果：



失敗可能的原因為：

1. 訓練資料有限 2. 人工生成的戴口罩資料集 3. 口罩遮蓋區域過大

## 伍、結論

李佩穎：

這次的期末專題主題其實是我之前就一直很想做的題目了，想說可以藉由這次的機會來實現自己想要做的專題，參考了網站上所說的方法去製作，讓我學習到了類神經網路的應用，老師有說一開始的資料集可以針對戴口罩就可以辨識人臉的話就更好了，但是我想了想我是針對原本的人有沒有戴口罩這點，所以針對戴口罩可以偵測到那個人是誰的話，可能是另一種主題了，所以我還是維持原本的資料集去呈現我的結果，之後我會實際去實現當戴口罩還可以辨識人臉的部分，感謝老師這學期的教導，讓我學習到很多，雖然沒有其他同學厲害，但我會把我所學的在未來的研究上去應用。

宋立晴：

人臉辨識是近幾年最熱門的人工智慧應用之一，也已廣泛應用到許多產品。近一兩年因疫情所需，我們處處可以見到口罩偵測、體溫量測等功

能，甚至是支援戴口罩人士進行身份識別供門禁、打卡使用，非接觸性門禁系統不但可以在非常時期提高社交距離，實際上也可降低人力負擔，使打卡系統更容易使用。之前在其他課堂時我已有作過人臉辨識功能，因此希望藉這次期末報告的機會，實作更符合時事的主題，這些都是我選擇在期末專題做口罩偵測器的原因。

我認為這次期末報告最特別的是我們選用的資料集，此資料集在疫情爆發不久就上線供大家使用，讓我驚訝的是這資料集只是一般沒有戴口罩的人臉資料，經過作者處理過後才被加上口罩的，可想而知疫情初爆發時，要搜集一份廣大的口罩資料照片是不容易的，但作者透過這樣的變通方式，在第一時間做出了應對方法且達到很好的效果，這點讓我覺得很值得學習，也期許自己未來在面對困難時，有山不轉路轉，路不轉人轉的機智。

陸、程式碼

[https://colab.research.google.com/drive/1pkGk-F5ul4\\_8aB40JydCP35lrydpmrg4?usp=sharing](https://colab.research.google.com/drive/1pkGk-F5ul4_8aB40JydCP35lrydpmrg4?usp=sharing)

柒、參考

- [1] <https://github.com/prajnasb/observations>
- [2] <https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/>
- [3] <https://medium.com/@PatHuang/%E5%88%9D%E5%AD%B8python%E6%89%8B%E8%A8%98-3-%E8%B3%87%E6%96%99%E5%89%8D%E8%99%95%E7%90%86-label-encoding-one-hot-encoding-85c983d63f87>
- [4] <https://paperswithcode.com/paper/mobilenetv2-inverted-residuals-and-linear>
- [5] <https://image-net.org/>
- [6] <https://www.zangcq.com/2019/05/07/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E6%A6%82%E5%BF%B5%E7%90%86%E8%A7%A3-roi/>