

## 資料探勘與巨量資料分析 期末報告

### 一、選擇之資料來源

- 來源

kaggle

Kaggle 是一個數據建模和數據分析競賽平台。企業和研究者可在其上發布數據，統計學者和數據挖掘專家可在其上進行競賽以產生最好的模型。

- 使用資料集

<https://www.kaggle.com/andrewmvd/face-mask-detection>

### 二、分析的內容及目的

- 內容

2020 至今新冠肺炎肆虐，進出公眾場所「戴口罩」成為保護自己、保護別人必要的動作，但樹大必有枯枝，總是會有人忘記戴上這保命工具，而成為防疫破口。那麼警示這些忘記戴上口罩的人們就變成一項重要的工作，因此我想利用這次的期末專題，訓練一個能夠判斷人們是否有戴上口罩的影像辨識模型。

- 目的

透過訓練完的模型可以辨識影像中的人們是否有戴口罩，希望有機會能在市場上運用並達成以下目標：

1. 公眾場所出入口能多一道防線
2. 隨時警示人們不要脫下口罩
3. 減少因口罩造成人們間的衝突

### 三、分析方法與流程

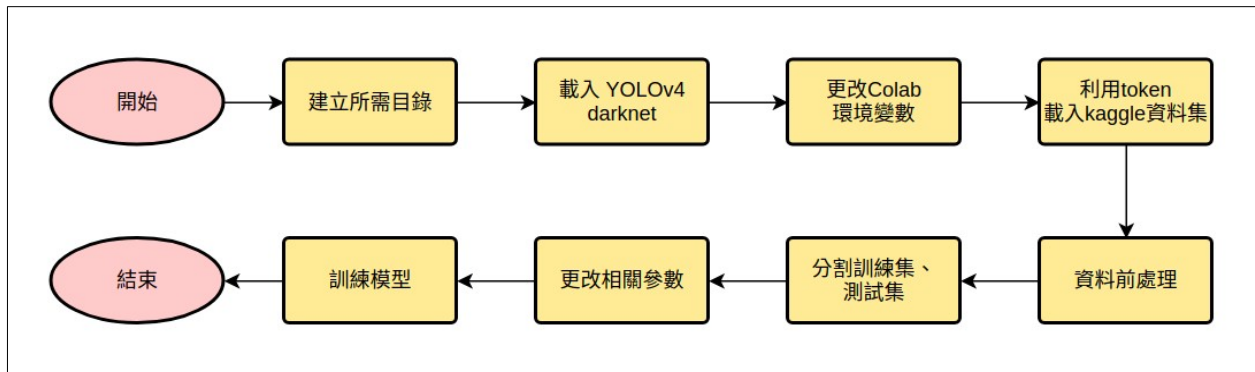
- 開發工具及環境

環境	Google Colab
	Ubuntu 20.04
開發工具	YOLOv4
	Python 3.9

- 分析方法

使用 Google Colab 上提供的 GPU 資源訓練 YOLOv4 模型，並在本機將 YOLOv4 訓練完得權重檔利用 Python 內建之套件，例如：OpenCV 等，串接 Webcam 及影像處理。Yolov4 是以 CNN 為架構的模型。

- 分析流程



- 程式碼解釋

### STEP 1 建立所需目錄

連結Google drive

```
[ ] import os
import pandas as pd
from google.colab import drive
drive.mount('/content/gdrive')
```

建立有的沒的資料夾

```
[ ] import os
if not os.path.exists('/content/gdrive/MyDrive/yolo_space/mask_try'):
    os.makedirs('/content/gdrive/MyDrive/yolo_space/mask_try')
else:
    print('folder exist')

%cd /content/gdrive/MyDrive/yolo_space/mask_try
if not os.path.exists('/content/gdrive/MyDrive/yolo_space/mask_try/yolo_download'):
    os.makedirs('/content/gdrive/MyDrive/yolo_space/mask_try/yolo_download')
else:
    print('folder exist')
%cd /content/gdrive/MyDrive/yolo_space/mask_try/yolo_download
```

### STEP 2 載入 YOLOv4 darknet

```
[ ] #下載Yolov2套件
!git clone https://github.com/hai-h-nguyen/Yolo2Pascal-annotation-conversion.git
#下載darknet
!git clone https://github.com/AlexeyAB/darknet
```

### STEP 3 更變 colab 環境變數

```
▶ %cd /content/gdrive/MyDrive/yolo_space/mask_try/yolo_download
#更變環境變數
!sed -i "s/GPU=0/GPU=1/g" darknet/Makefile
!sed -i "s/CUDNN=0/CUDNN=1/g" darknet/Makefile
!sed -i "s/CUDNN_HALF=0/CUDNN_HALF=1/g" darknet/Makefile
!sed -i "s/OPENCV=0/OPENCV=1/g" darknet/Makefile
#編譯darknet
!cd darknet; make
```

**STEP 4** 訓練資料處理

```
[ ] from PIL import Image
import os
# 轉 jpg
for img in os.listdir("/content/gdrive/MyDrive/yolo_space/dataset/img"):
    if img.endswith(".png"):
        im = Image.open("/content/gdrive/MyDrive/yolo_space/dataset/img/{}".format(img))
        rgb_im = im.convert("RGB")
        img_name = img.split(".")[0]
        rgb_im.save("/content/gdrive/MyDrive/yolo_space/dataset/img/{}.jpg".format(img_name))
```

刪除png檔

```
[ ] from PIL import Image
import os
# 刪除原始 png 資料
for img in os.listdir("/content/gdrive/MyDrive/yolo_space/dataset/images"):
    if img.endswith(".png"):
        os.remove("/content/gdrive/MyDrive/yolo_space/dataset/images/{}".format(img))
```

```
[ ] %cd /content/gdrive/MyDrive/yolo_space/mask_try/yolo_download/Yolo2Pascal-annotation-conversion/pascal2yolo
!python voc2yolo.py /content/gdrive/MyDrive/yolo_space/dataset/to_yolo_format_mine

/content/gdrive/MyDrive/yolo_space/mask_try/yolo_download/Yolo2Pascal-annotation-conversion/pascal2yolo
Convert 5.xml
Image path: /content/gdrive/MyDrive/yolo_space/dataset/to_yolo_format_mine/5.jpg
```

**STEP 5** 訓練資料分割

```
[ ] # split data to train and dev
import os, shutil
import random
# 90%訓練 10%驗證
train_ratio = 0.9
train_num = int(round(853 * train_ratio, 0))
# prepare train folder
images_list = []
for img in os.listdir("/content/gdrive/MyDrive/yolo_space/dataset/to_yolo_format"):
    if img.endswith(".jpg"):
        images_list.append(img.split(".")[0])

random.shuffle(images_list)
yolo_format_folder = os.path.join("/content/gdrive/MyDrive/yolo_space/dataset/to_yolo_format")
train_folder = os.path.join("train")
dev_folder = os.path.join("dev")
if not os.path.isdir("train"):
    os.mkdir(train_folder)
if not os.path.isdir("dev"):
    os.mkdir(dev_folder)
# train data
for train_data in images_list[:train_num]:
    shutil.copyfile(os.path.join(yolo_format_folder, "{}.jpg".format(train_data)),
                    os.path.join(train_folder, "{}.jpg".format(train_data)))
    shutil.copyfile(os.path.join(yolo_format_folder, "{}.txt".format(train_data)),
                    os.path.join(train_folder, "{}.txt".format(train_data)))

# dev data
for test_data in images_list[train_num+1:]:
    shutil.copyfile(os.path.join(yolo_format_folder, "{}.jpg".format(test_data)),
                    os.path.join(dev_folder, "{}.jpg".format(test_data)))
    shutil.copyfile(os.path.join(yolo_format_folder, "{}.txt".format(test_data)),
                    os.path.join(dev_folder, "{}.txt".format(test_data)))

# show total data
print("="*35)
print("number of training set :", len(os.listdir(train_folder)))
print("number of dev set :", len(os.listdir(dev_folder)))
print("="*35)
```



**STEP 6** 訓練模型

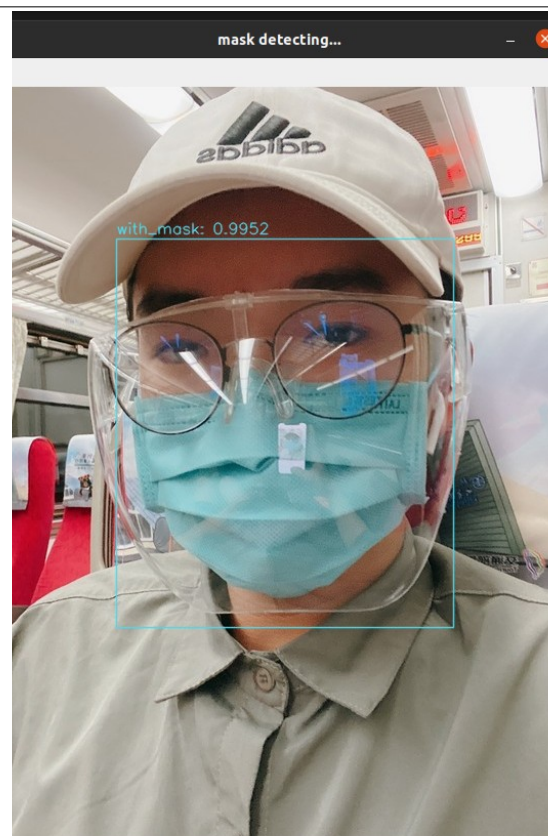
```

! ./darknet detector train
content/gdrive/MyDrive/yolo_space/mask_try/yolo_download/prepare/darknet/data/mask.data
/content/gdrive/MyDrive/yolo_space/mask_try/yolo_download/prepare/darknet/cfg/yolov4-mask.cfg
/content/gdrive/MyDrive/yolo_space/mask_try/yolo_download/prepare/yolov4.conv.137 -dont_show

36 Shortcut Layer: 33, wt = 0, wn = 0, outputs: 52 x 52 x 128 0.000 BF
37 conv 128 1 x 1/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
38 conv 128 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.797 BF
39 Shortcut Layer: 36, wt = 0, wn = 0, outputs: 52 x 52 x 128 0.000 BF
40 conv 128 1 x 1/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
41 conv 128 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.797 BF
42 Shortcut Layer: 39, wt = 0, wn = 0, outputs: 52 x 52 x 128 0.000 BF
43 conv 128 1 x 1/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
44 conv 128 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.797 BF
45 Shortcut Layer: 42, wt = 0, wn = 0, outputs: 52 x 52 x 128 0.000 BF
46 conv 128 1 x 1/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
47 conv 128 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.797 BF
48 Shortcut Layer: 45, wt = 0, wn = 0, outputs: 52 x 52 x 128 0.000 BF
49 conv 128 1 x 1/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
50 conv 128 3 x 3/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.797 BF
51 Shortcut Layer: 48, wt = 0, wn = 0, outputs: 52 x 52 x 128 0.000 BF
52 conv 128 1 x 1/ 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
53 route 52 25 -> 52 x 52 x 256
54 conv 256 1 x 1/ 1 52 x 52 x 256 -> 52 x 52 x 256 0.354 BF
55 conv 512 3 x 3/ 2 52 x 52 x 256 -> 26 x 26 x 512 1.595 BF
56 conv 256 1 x 1/ 1 26 x 26 x 512 -> 26 x 26 x 256 0.177 BF
57 route 55 -> 26 x 26 x 512

```

## • DEMO



配戴口罩，辨識度 0.9952



#### 四、結果說明

這次的作業雖然我覺得好像跟上課有一些不同，但因為正好在研究 YOLOv4 就拿來玩玩看了。專題中最讓我意外的是我訓練的圖片僅有 800 多張（包含未戴口罩、未正確配戴口罩、戴口罩），辨識度卻意外的高。希望老師會喜歡我這次做的作業。

另外因為模型檔案太大，我將檔案放入雲端硬碟內，若老師對程式碼有興趣可以看看：  
[https://drive.google.com/drive/folders/1S2IBbl\\_MBAQdd70bYas0dPpnULmqRqAa?usp=sharing](https://drive.google.com/drive/folders/1S2IBbl_MBAQdd70bYas0dPpnULmqRqAa?usp=sharing)