人工智慧

口罩辨識系統製作

吳智鴻 教授

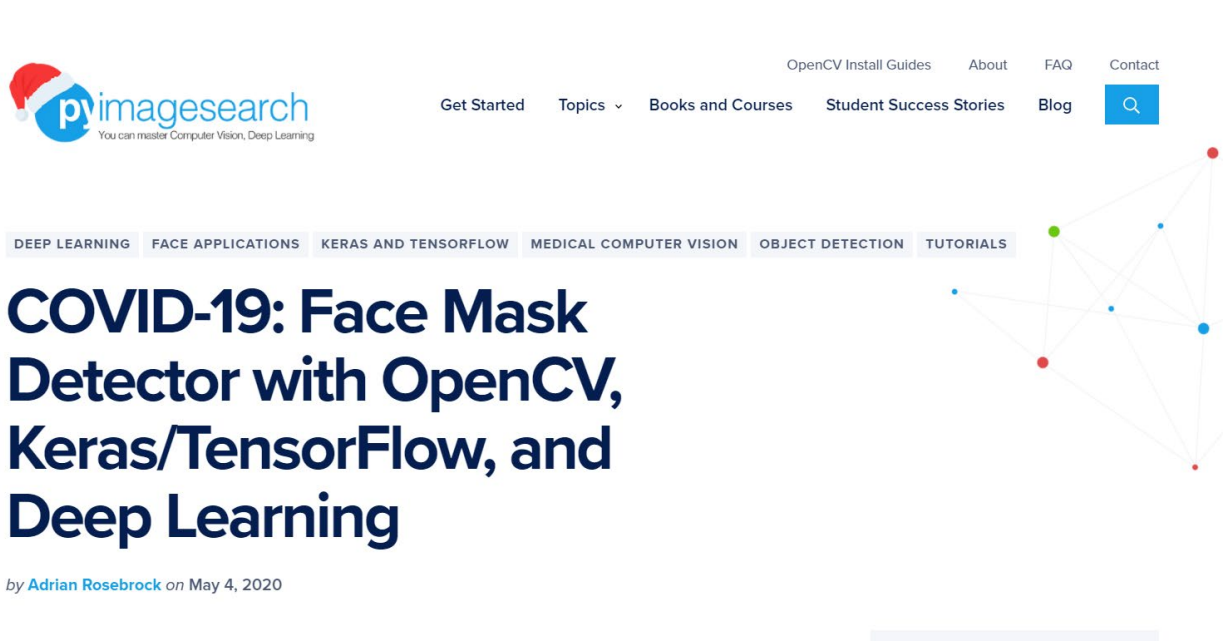
國立臺中教育大學 數位內容科技學系

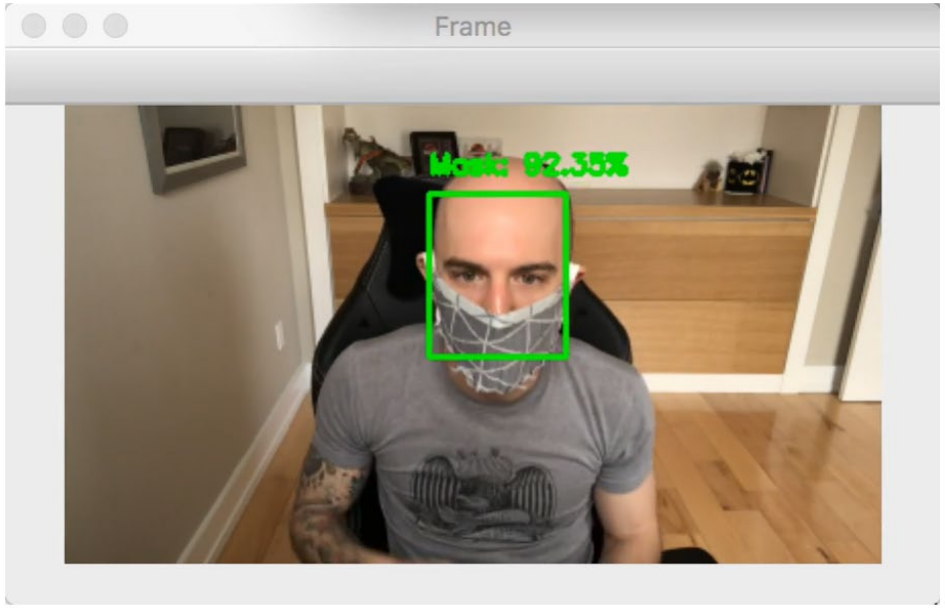
資料來源

參考網站：

https://github.com/chandrikadeb7/Face-Mask-Detection

Good website for learning deep learning techniques



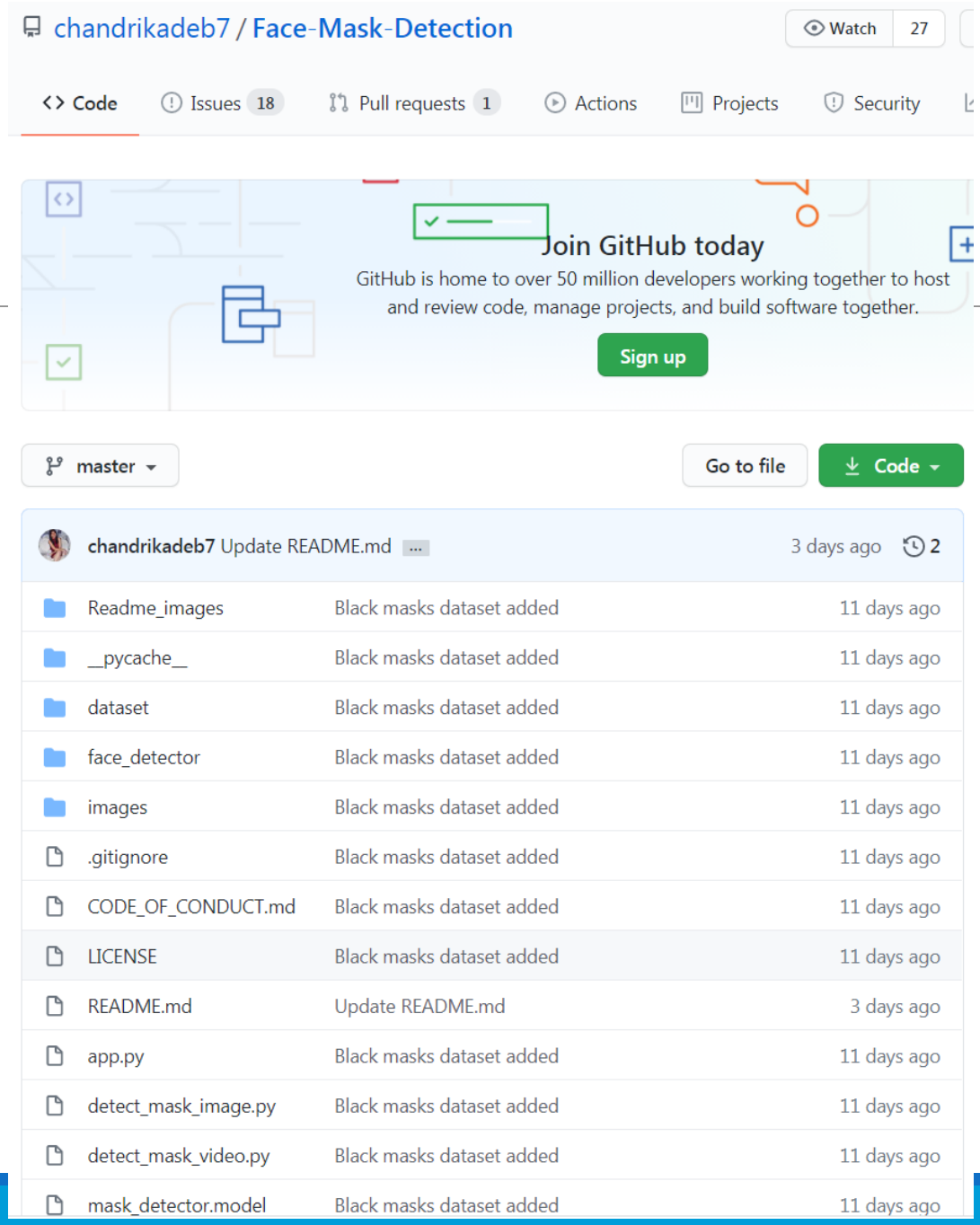


In this tutorial, you will learn how to train a COVID-19 face mask detector with OpenCV, Keras/TensorFlow, and Deep Learning.

Last month, I authored a blog post on detecting COVID-19 in X-ray images using deep learning.

網站連結

系統架構



chandrikadeb7 / Face-Mask-Detection

Watch 27

Code | Issues 18 | Pull requests 1 | Actions | Projects | Security

Join GitHub today

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

Sign up

master | Go to file | Code

chandrikadeb7 Update README.md ... 3 days ago 2

Readme_images	Black masks dataset added	11 days ago
__pycache__	Black masks dataset added	11 days ago
dataset	Black masks dataset added	11 days ago
face_detector	Black masks dataset added	11 days ago
images	Black masks dataset added	11 days ago
.gitignore	Black masks dataset added	11 days ago
CODE_OF_CONDUCT.md	Black masks dataset added	11 days ago
LICENSE	Black masks dataset added	11 days ago
README.md	Update README.md	3 days ago
app.py	Black masks dataset added	11 days ago
detect_mask_image.py	Black masks dataset added	11 days ago
detect_mask_video.py	Black masks dataset added	11 days ago
mask_detector.model	Black masks dataset added	11 days ago

使用到的架構

⚠️ TechStack/framework used

- OpenCV
- Caffe-based face detector
- Keras
- TensorFlow
- MobileNetV2

MobileNet v1

谷歌 2017 年推出了 MobileNetV1，它是一種為移動設備設計的通用計算機視覺神經網絡，因此它也能支持圖像分類和檢測等。一般在個人移動設備上運行深度網絡能提升用戶體驗、提高訪問的靈活性，以及在安全、隱私和能耗上獲得額外的優勢。

此外，隨著新應用的出現，用戶可以與真實世界進行實時交互，因此我們對更高效的神經網絡有著很大的需求。

原文網址：https://kknews.cc/tech/nekyn68.html

MobileNet V2

下一代移動視覺應用 MobileNetV2 已經發布。

MobileNetV2 在 MobileNetV1 的基礎上獲得了顯著的提升，並推動了移動視覺識別技術的有效發展，包括分類、目標檢測和語義分割。

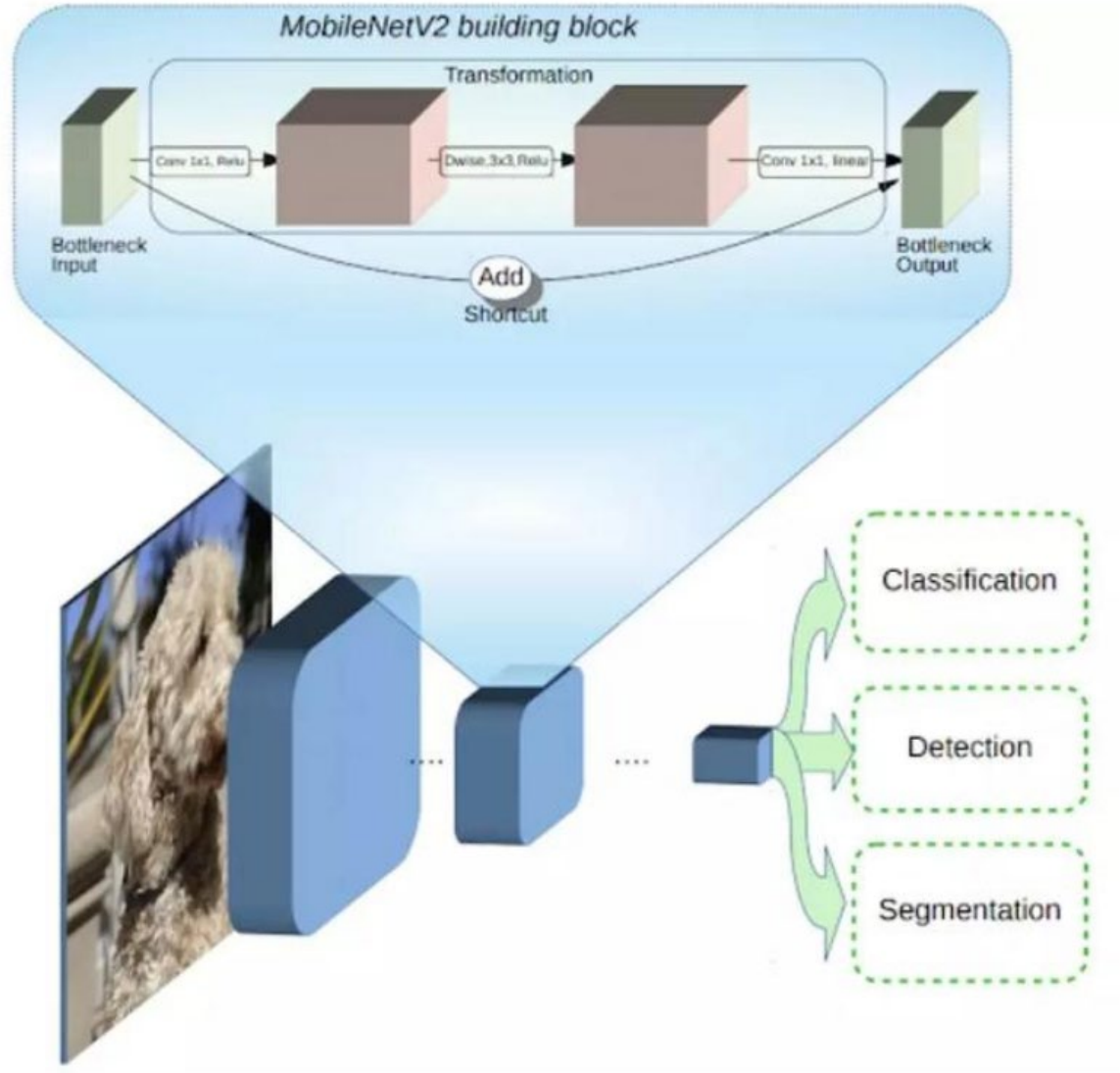
MobileNetV2 作為 TensorFlow-Slim 圖像分類庫的一部分而推出，讀者也可以在 Colaboratory 中立即探索 MobileNetV2。

原文網址：https://kknews.cc/tech/nekyn68.html

MobileNetv2架構

可以直觀理解為，瓶頸層對模型的中間輸入與輸出進行編碼，而內層封裝了模型從像素等低級概念到圖像類別等高級概念的轉換能力。最後，與傳統的殘差連接一樣，捷徑能快速訓練並獲得更優精確度。

原文網址：
https://kknews.cc/tech/nekyn68.html



MobileNetV2 的架構概覽，藍色塊如上所示為複合卷積構建塊。

MobileNet v2的優點

V2 與第一代的 MobileNet 相比有什麼區別？

總體而言，MobileNetV2 模型在整體延遲範圍內上實現相同的準確度要更快。

特別是，目前新模型減少了兩倍 operations 的數量，且只需要原來 70% 的參數，在 Google Pixel 手機上的測試表明 V2 要比 MobileNetV1 快 30% 到 40%，同時還能實現更高的準確度。

原文網址：https://kknews.cc/tech/nekyn68.html

⭐ Features

Our face mask detector didn't use any morphed masked images dataset. The model is accurate, and since we used the MobileNetV2 architecture, it's also computationally efficient and thus making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.).

This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed.

採用MobileNetV2架構。 可部屬在 Raspberry Pi, Google Coral等嵌入式系統上

資料集來源

📁 Dataset

The dataset used can be downloaded here - Click to Download

This dataset consists of **3835 images** belonging to two classes:

- **with_mask: 1916 images**
- **without_mask: 1919 images**

The images used were real images of faces wearing masks. The images were collected from the following sources:

- **Bing Search API** (See Python script)
- **Kaggle datasets**
- **RMFD dataset** (See here)

🚀 Installation

1. Clone the repo

```
$ git clone https://github.com/chandrikadeb7/Face-Mask-Detection.git
```

2. Change your directory to the cloned repo and create a Python virtual environment named 'test'

```
$ mkvirtualenv test
```

3. Now, run the following command in your Terminal/Command Prompt to install the libraries required

```
$ pip3 install -r requirements.txt
```

安裝方式

1. git hub下載原始檔

   https://github.com/chandrikadeb7/Face-Mask-Detection

   解開後放在 c:\python\mask目錄下

2. 打開anaconda prompt

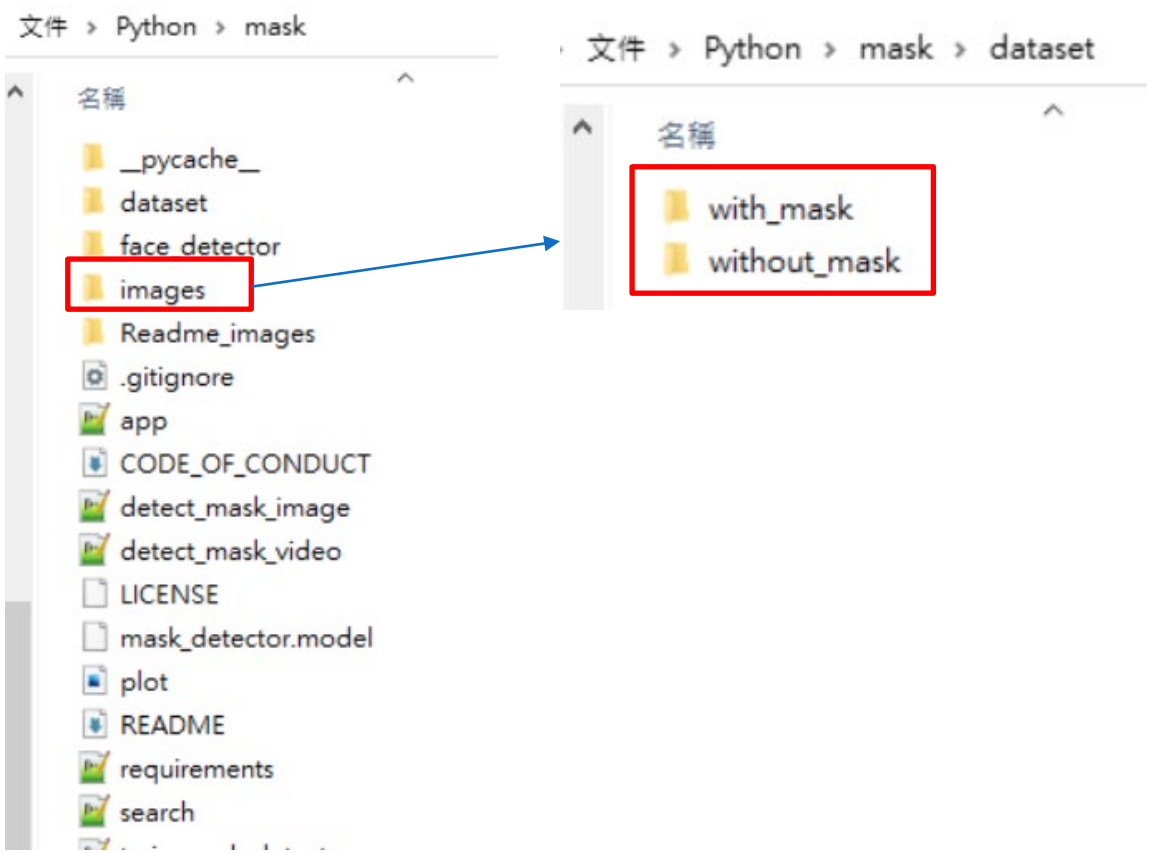
   python.exe -m pip install —upgrade pip (更新 pip指令)
   conda create -n mask (創建一個虛擬環境 mask)

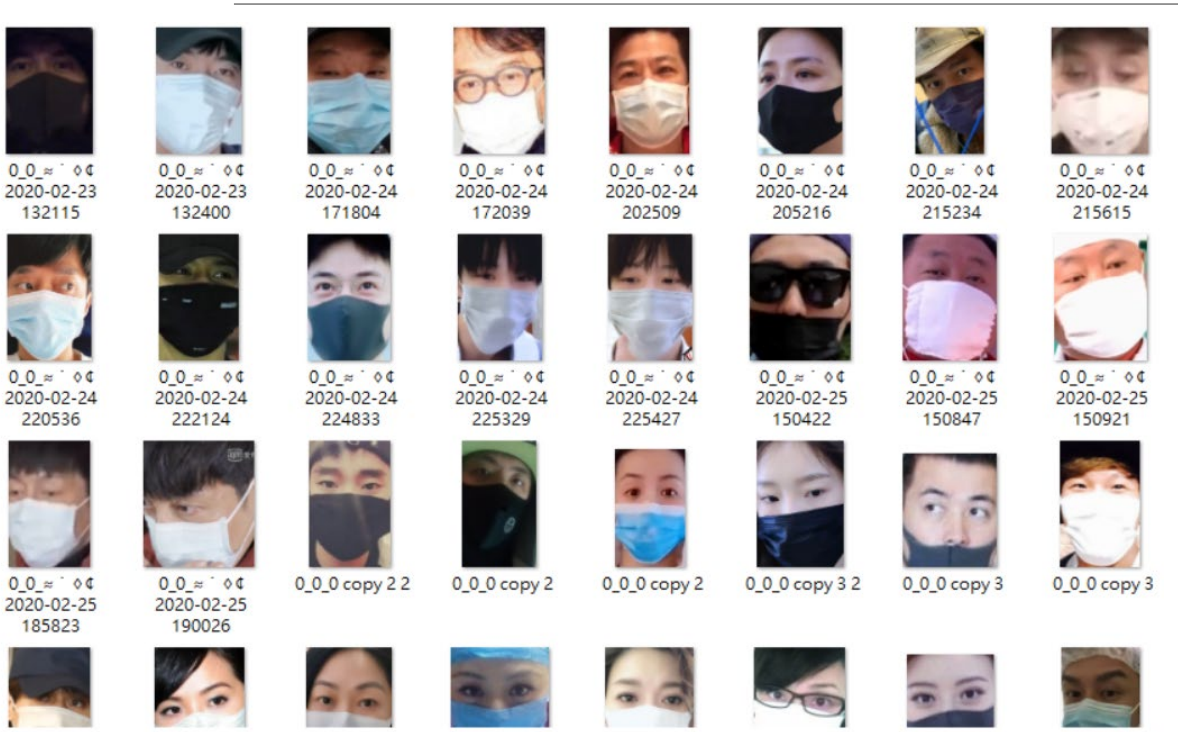
   conda activate mask (啟動虛擬環境)

   cd c:\python\mask\

   pip3 install -r requirements.txt (安裝本project所需套件)

目錄結構


文件 › Python › mask
名稱
__pycache__
dataset
face_detector
images
Readme_images
.gitignore
app
CODE_OF_CONDUCT
detect_mask_image
detect_mask_video
LICENSE
mask_detector.model
plot
README
requirements
search
文件 › Python › mask › dataset
名稱
with_mask
without_mask


0_0_≈ ˙ ◊¢ 2020-02-23 132115
0_0_≈ ˙ ◊¢ 2020-02-23 132400
0_0_≈ ˙ ◊¢ 2020-02-24 171804
0_0_≈ ˙ ◊¢ 2020-02-24 172039
0_0_≈ ˙ ◊¢ 2020-02-24 202509
0_0_≈ ˙ ◊¢ 2020-02-24 205216
0_0_≈ ˙ ◊¢ 2020-02-24 215234
0_0_≈ ˙ ◊¢ 2020-02-24 215615
0_0_≈ ˙ ◊¢ 2020-02-24 220536
0_0_≈ ˙ ◊¢ 2020-02-24 222124
0_0_≈ ˙ ◊¢ 2020-02-24 224833
0_0_≈ ˙ ◊¢ 2020-02-24 225329
0_0_≈ ˙ ◊¢ 2020-02-24 225427
0_0_≈ ˙ ◊¢ 2020-02-25 150422
0_0_≈ ˙ ◊¢ 2020-02-25 150847
0_0_≈ ˙ ◊¢ 2020-02-25 150921
0_0_≈ ˙ ◊¢ 2020-02-25 185823
0_0_≈ ˙ ◊¢ 2020-02-25 190026
0_0_0 copy 2 2
0_0_0 copy 2
0_0_0 copy 2
0_0_0 copy 3 2
0_0_0 copy 3
0_0_0 copy 3


0
0_0_aidai_0014
0_0_aidai_0029
0_0_aidai_0043
0_0_aidai_0074
0_0_aidai_0084
0_0_aidai_0136
0_0_anhu_0004
0_0_anhu_0020
0_0_anhu_0025
0_0_anhu_0027
0_0_anhu_0056
0_0_anhu_0057
0_0_anhu_0062
0_0_anhu_0063
0_0_anhu_0098
0_0_anhu_0103
0_0_anhu_0155
0_0_anhu_0157
0_0_anhu_0189
0_0_anhu_0201
0_0_anhu_0205
0_0_anhu_0209
0_0_anhu_0211
0_0_anhu_0214
0_0_anhu_0216
0_0_baibaihe_0077
0_0_baibaihe_0085
0_0_baibaihe_0093
0_0_baibaihe_0204
0_0_baibaihe_0216
0_0_baibaihe_0236

requirements內容
本專案所需套件&版本

```
1  tensorflow>=1.15.2
2  keras==2.3.1
3  imutils==0.5.3
4  numpy==1.18.2
5  opencv-python==4.2.0.*
6  matplotlib==3.2.1
7  argparse==1.1
8  scipy==1.4.1
9  scikit-learn==0.23.1
10 pillow==7.2.0
11 streamlit==0.65.2
```

💡 Working

1. Open terminal. Go into the cloned project directory and type the following command:

```
$ python3 train_mask_detector.py --dataset dataset
```

2. To detect face masks in an image type the following command:

```
$ python3 detect_mask_image.py --image images/pic1.jpeg
```

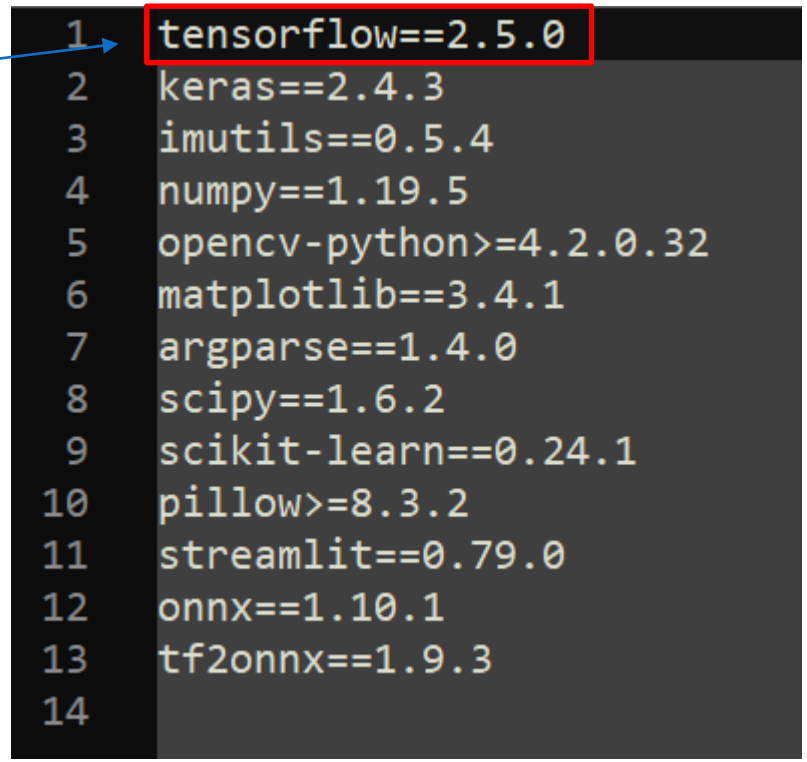
3. To detect face masks in real-time video streams type the following command:

```
$ python3 detect_mask_video.py
```

修改#1

修改Tensorflow版本(避免下載過久)

改成 tensorflow==2.5.0



修改#2

epochs次數修改
20->2 到5
節省訓練時間

打開train_mask_detector.py

train_mask_detector.py

```
16  from tensorflow.keras.preprocessing.image import load_img
17  from tensorflow.keras.utils import to_categorical
18  from sklearn.preprocessing import LabelBinarizer
19  from sklearn.model_selection import train_test_split
20  from sklearn.metrics import classification_report
21  from imutils import paths
22  import matplotlib.pyplot as plt
23  import numpy as np
24  import argparse
25  import os
26
27  # construct the argument parser and parse the arguments
28  ap = argparse.ArgumentParser()
29  ap.add_argument("-d", "--dataset", required=True,
30      help="path to input dataset")
31  ap.add_argument("-p", "--plot", type=str, default="plot.png",
32      help="path to output loss/accuracy plot")
33  ap.add_argument("-m", "--model", type=str,
34      default="mask_detector.model",
35      help="path to output face mask detector model")
36  args = vars(ap.parse_args())
37
38  # initialize the initial learning rate, number of epochs to train for,
39  # and batch size
40  INIT_LR = 1e-4
41  EPOCHS = 20
42  BS = 32
43
44  # grab the list of images in our dataset directory, then initialize
45  # the list of data (i.e., images) and class images
46  print("[INFO] loading images...")
47  imagePaths = list(paths.list_images(args["dataset"]))
48  data = []
49  labels = []
50
51  # loop over the image paths
52  for imagePath in imagePaths:
53      # extract the class label from the filename
54      label = imagePath.split(os.path.sep)[-2]
55
56      # load the input image (224x224) and preprocess it
57      image = load_img(imagePath, target_size=(224, 224))
58      image = img_to_array(image)
59      image = preprocess_input(image)
60
61      # update the data and labels lists, respectively
```

🔑 Results

Our model gave 93% accuracy for Face Mask Detection after training via `tensorflow-gpu==2.0.0`

```
[INFO] evaluating network...
              precision    recall  f1-score   support

   with_mask       0.99      0.86      0.92       383
without_mask       0.88      0.99      0.93       384

    accuracy                           0.93       767
   macro avg       0.93      0.93      0.93       767
weighted avg       0.93      0.93      0.93       767

[INFO] saving mask detector model...
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

We got the following accuracy/loss training curve plot

觀察哪個地方
正確率平穩?

觀察哪個地方
誤差最低

程式報錯的修正語法

```
143  # plot the training loss and accuracy
144  N = EPOCHS
145  plt.style.use("ggplot")
146  plt.figure()
147  plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
148  plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
149  #舊語法 history["acc"] -> history["accuracy"]
150  plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
151  #舊語法 history["val_acc"] -> history["val_accuracy"]
152  #plt.plot(np.arange(0, N), H.history["val_acc"], label="val_acc")
153  plt.plot(np.arange(0, N), H.history["val_acc"], label="val_acc")
154  plt.title("Training Loss and Accuracy")
155  plt.xlabel("Epoch #")
156  plt.ylabel("Loss/Accuracy")
157  plt.legend(loc="lower left")
158  plt.savefig(args["plot"])
159
```

修改語法

實際跑出的正確率

	precision	recall	f1-score	support
with_mask	0.97	0.97	0.97	387
without_mask	0.97	0.97	0.97	386
accuracy			0.97	773
macro avg	0.97	0.97	0.97	773
weighted avg	0.97	0.97	0.97	773

Exercise#1 完成一張口罩圖片的辨識

指令:

準備一張自己的口罩照片，放在images目錄下

python detect_mask_image.py —image images/xxx.jpeg

將結果截圖上傳FB社團

準備一張照片
images底下

命令列執行：

python detect_mask_image.py –image images/pic4.png

辨識結果

Exercise#2

修改訓練的次數，從20次-> 5次 觀察訓練成效acc & loss

修改 train_mask_detector.py

```
38  # initialize the initial learning rate, number of epochs to train for,
39  # and batch size
40  INIT_LR = 1e-4
41  EPOCHS = 20
42  BS = 32
```

觀察哪個地方
要修改?

python train_mask_detector.py --dataset dataset

Exercise#3 增加自己的訓練圖片

增加幾張自己的口罩圖片。放在dataset底下

重新訓練模型

Exercise#3 增加自己的訓練圖片 (做法)

增加幾張自己的口罩圖片。放在dataset底下

重新訓練模型

Python train_mask_detector.py —dataset dataset

Exercise#4 啟用webcam自動辨識口罩並修改顯示內容

要求 :

啟動webcam辨識口罩程式

修改顯示內容，

修改 Mask & No mask的顯示文字

修改 Mask & No mask的顯示外框顏色

顯示後結果截圖上傳FB社團

Exercise#4 (解答)
啟用webcam自動辨識口罩並修改顯示內容

要求：

啟動webcam辨識口罩程式

python detect_mask_video.py

修改顯示內容，顯示後結果截圖上傳

```
118     for (box, pred) in zip(locs, preds):
119         # unpack the bounding box and predictions
120         (startX, startY, endX, endY) = box
121         (mask, withoutMask) = pred
122
123         # determine the class label and color we'll use to draw
124         # the bounding box and text
125         label = "Mask" if mask > withoutMask else "No Mask"
126         color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
127
128         # include the probability in the label
129         label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
130
131         # display the label and bounding box rectangle on the output
132         # frame
133         cv2.putText(frame, label, (startX, startY - 10),
134             cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
135         cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
```

資料來源

1. https://github.com/chandrikadeb7/Face-Mask-Detection

2. Mobilenet v2介紹

https://medium.com/ai-academy-taiwan/efficient-cnn-%E4%BB%8B%E7%B4%B9-%E4%BA%8C-mobilenetv2-7809721f0bc8

3.谷歌發布MobileNetV2：可做語義分割下一代移動端計算機視覺架構

原文網址：https://kknews.cc/tech/nekyn68.html

Congratulations.

You can design your real-time Mask Detection now ！