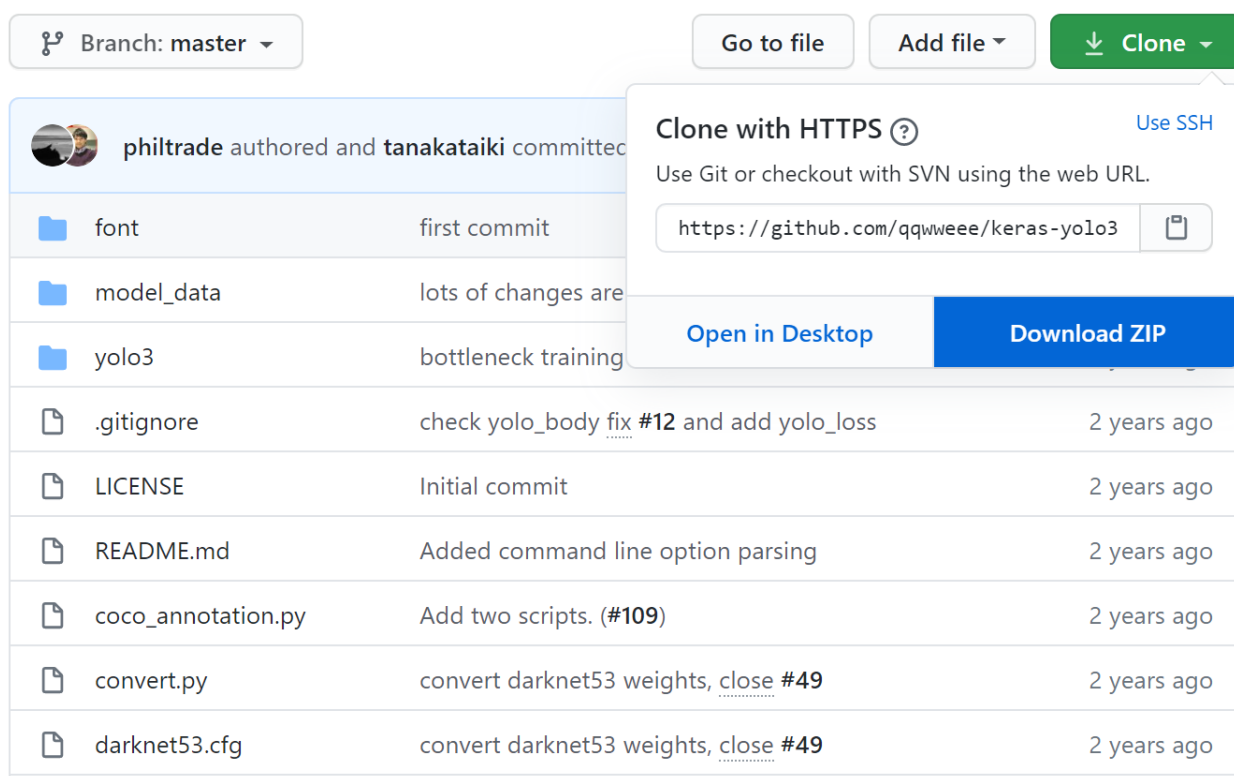


2020.06.30

## 一、安裝與測試

1.先到 github(<https://github.com/qqwweee/keras-yolo3>) 下載 yolo3 程式



2.從 <https://pjreddie.com/media/files/yolov3.weights> 下載 yolov3.weights 檔案

3.下載完後使用將 yolov3.weights 放入置 yolov3 資料夾裡面並開啟 cmd 進行測試，使用：  
`python convert.py yolov3.cfg yolov3.weights model_data/yolo.h5`  
(將 yolov3.weights 放在 keras-yolo3-master 資料夾內 跟 yolo3 同層 開啟 Anaconda Prompt(testAI)的 cmd 測試)

4.將下載好的 yolov3.weights 複製一份轉檔成.h5 檔的，放在 model\_data 裡面，再將影像(dog.jpg)放到資料夾內(需要先將影像丟到資料夾內，才可以測試)

開始進行測試時，輸入：`python yolo_video.py -image`

1) RuntimeError: `get\_session` is not available when using TensorFlow 2.0.

<https://github.com/OlafenwaMoses/ImageAI/issues/367>

2) 使用 TensorFlow 2.0 时'get\_session'不可用'。

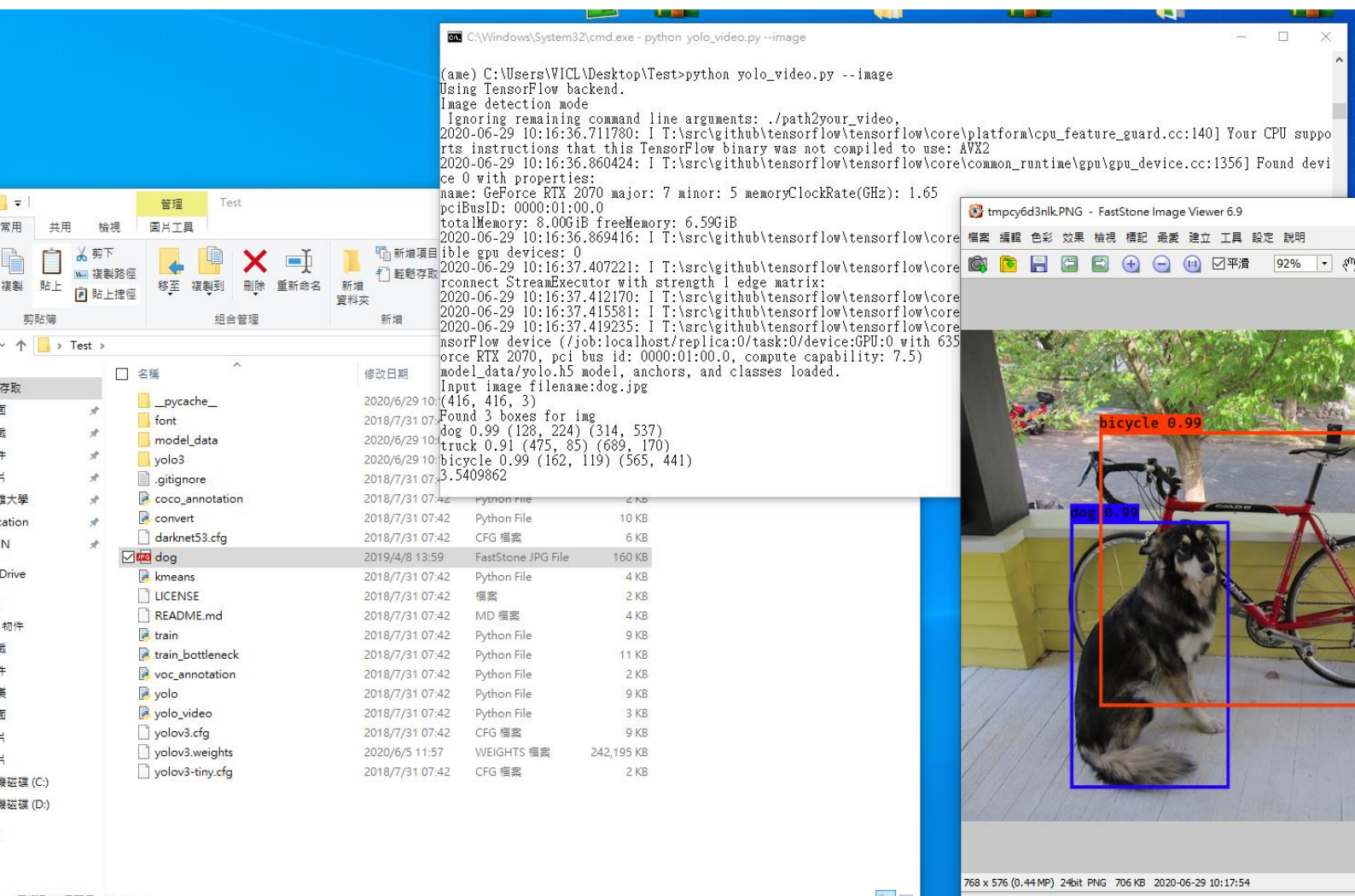
<https://mlog.club/article/3361758>

3) 解決 ImportError: cannot import name 'tf\_utils'

<https://blog.csdn.net/lingcai7071/article/details/103305364>

#### 4) pip 安裝指定 keras 版本

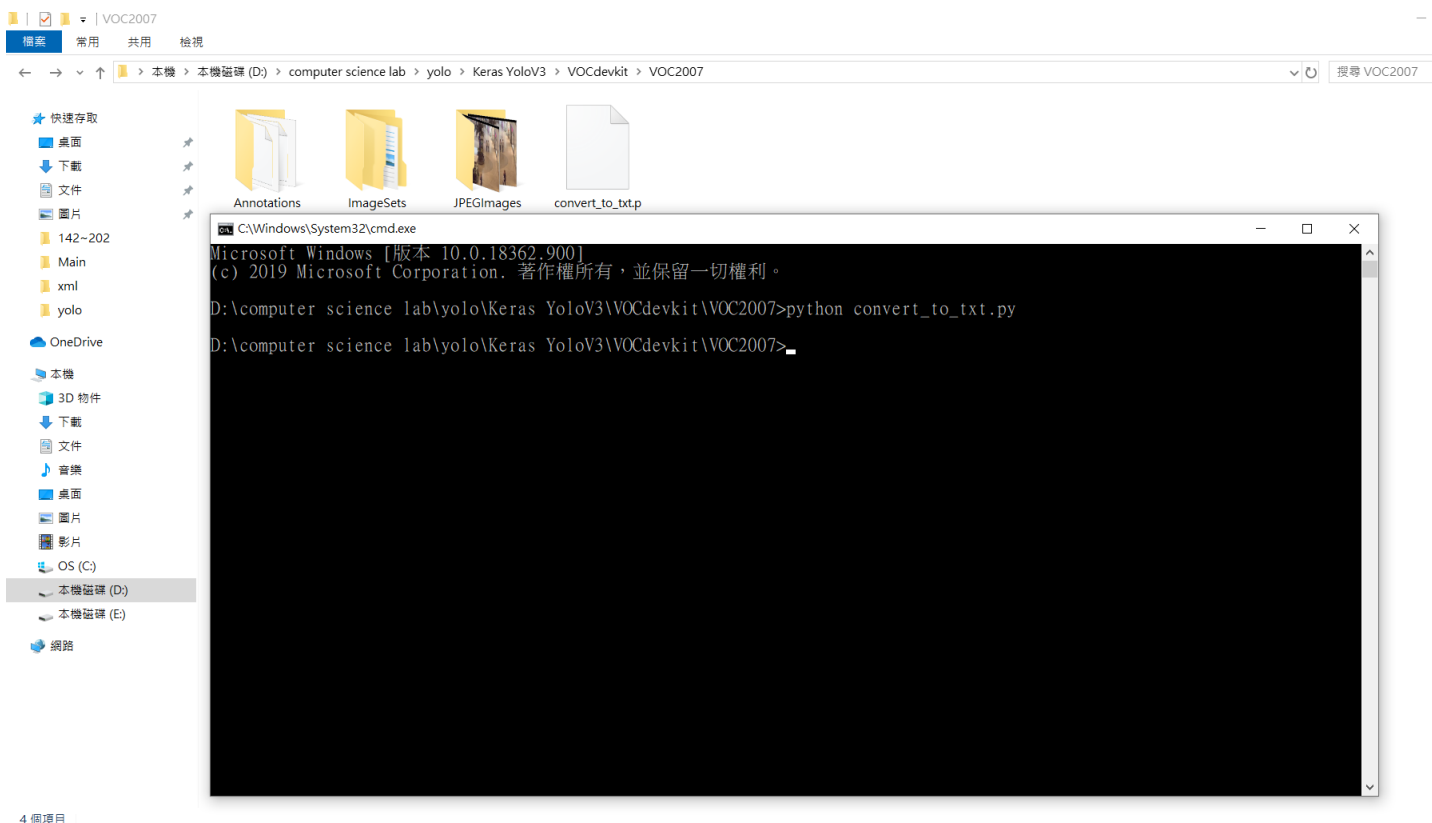
[https://blog.csdn.net/baidu\\_32936911/article/details/79753733](https://blog.csdn.net/baidu_32936911/article/details/79753733)



(成功後代表我們可以使用官方的 weights 檔來訓練我們自己的影像了)

## 二、建立與訓練模組

1. (在目錄)建立資料集(VOCdevkit)裡面需要放(VOC2007)並在(VOC2007)裡面放入 3 個資料夾分別為: Annotations (匯入 XML，底下有 XML 教學)、ImageSets (需要在建立一個 Main 的資料夾裡面放 test.txt、train.txt、val.txt、trainval.txt)和 JPEGImages(訓練影像)



2. 將自己的圖片以及 xml 按照要求放好後，在 VOC2007 的同級目錄下建立 convert\_to\_txt.py 檔案，複製下面程式碼至 convert\_to\_txt.py 並執行。

該程式碼是讀取上面的 xml 檔案中圖片名稱，並儲存在 ImageSets/Main 目錄下的 txt 檔案中。注意:此處 txt 中僅有圖片名稱。

(下為亂數生成程式)

```
import os
import random

trainval_percent = 0.1
train_percent = 0.9

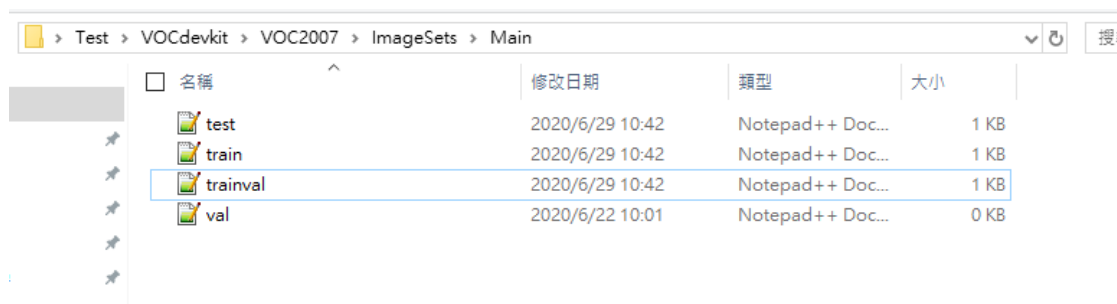
xmlfilepath = 'Annotations'
txtsavepath = 'ImageSets/Main'
total_xml = os.listdir(xmlfilepath)
```

```

num = len(total_xml)
list = range(num)
tv = int(num * trainval_percent)
tr = int(tv * train_percent)
trainval = random.sample(list, tv)
train = random.sample(trainval, tr)

ftrainval = open('ImageSets/Main/trainval.txt', 'w')
ftest = open('ImageSets/Main/test.txt', 'w')
ftrain = open('ImageSets/Main/train.txt', 'w')
fval = open('ImageSets/Main/val.txt', 'w')
for i in list:
    name = total_xml[i][:-4] + '\n'
    if i in trainval:
        ftrainval.write(name)
        if i in train:
            ftest.write(name)
        else:
            fval.write(name)
    else:
        ftrain.write(name)
ftrainval.close()
ftrain.close()
fval.close()
ftest.close()

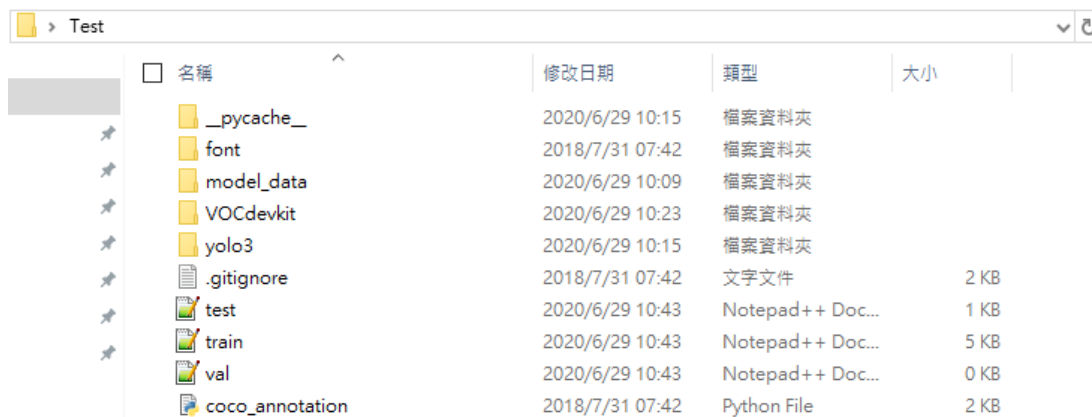
```



名稱	修改日期	類型	大小
test	2020/6/29 10:42	Notepad++ Doc...	1 KB
train	2020/6/29 10:42	Notepad++ Doc...	1 KB
trainval	2020/6/29 10:42	Notepad++ Doc...	1 KB
val	2020/6/22 10:01	Notepad++ Doc...	0 KB

執行完後 記事本裡會有隨機值在記事本裡面。

3.回到原目錄執行 `voc_annotation.py` 此時會出現 3 個檔案  
(分別為 `2007_test`、`2007_train`、`2007_val`) 手動把 `2007_`刪除  
完成後如下圖所示:



名稱	修改日期	類型	大小
_pycache_	2020/6/29 10:15	檔案資料夾	
font	2018/7/31 07:42	檔案資料夾	
model_data	2020/6/29 10:09	檔案資料夾	
VOCdevkit	2020/6/29 10:23	檔案資料夾	
yolo3	2020/6/29 10:15	檔案資料夾	
.gitignore	2018/7/31 07:42	文字文件	2 KB
test	2020/6/29 10:43	Notepad++ Doc...	1 KB
train	2020/6/29 10:43	Notepad++ Doc...	5 KB
val	2020/6/29 10:43	Notepad++ Doc...	0 KB
coco_annotation	2018/7/31 07:42	Python File	2 KB

此外，有可能剛執行檔案時發生以下狀況

```

C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.18362.900]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\WICL\Desktop\Test>activate ame

(ame) C:\Users\WICL\Desktop\Test>voc_annotation.py
Traceback (most recent call last):
  File "C:\Users\WICL\Desktop\Test\coc_annotation.py", line 31, in <module>
    convert_annotation(year, image_id, list_file)
  File "C:\Users\WICL\Desktop\Test\coc_annotation.py", line 11, in convert_annotation
    tree=ET.parse(in_file)
  File "C:\Users\WICL\Anaconda3\lib\xml\etree\ElementTree.py", line 1197, in parse
    tree.parse(source, parser)
  File "C:\Users\WICL\Anaconda3\lib\xml\etree\ElementTree.py", line 598, in parse
    self._root = parser._parse_whole(source)
UnicodeDecodeError: 'cp950' codec can't decode byte 0xe5 in position 142: illegal multibyte sequence

```

解決方法需要在程式裡加入

```

def convert_annotation(year, image_id, list_file):
    in_file = open('VOCdevkit/VOC%s/Annotations/%s.xml'%(year, image_id), encoding='utf-8')
    tree=ET.parse(in_file)
    root = tree.getroot()

    for obj in root.iter('object'):
        difficult = obj.find('difficult').text
        cls = obj.find('name').text
        if cls not in classes or int(difficult)==1:
            continue
        cls_id = classes.index(cls)
        xmlbox = obj.find('bndbox')
        b = (int(xmlbox.find('xmin').text), int(xmlbox.find('ymin').text), int(xmlbox.find('xmax').text), int(xmlbox.find('ymax').text))
        list_file.write(" " + ",".join([str(a) for a in b]) + ',' + str(cls_id))

wd = getcwd()

```

加入完後重新執行即可。

4.在資料夾中建立 `logs/ooo` 存放 `weight.h5` 檔  
`log_dir` 是將 `.h5` 檔放置的路徑。

```
C:\Users\VICL\Desktop\Test\train.py - Notepad++
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?

train.py
1  """
2  Retrain the YOLO model for your own dataset.
3  """
4  import numpy as np
5  import keras.backend as K
6  from keras.layers import Input, Lambda
7  from keras.models import Model
8  from keras.callbacks import TensorBoard, ModelCheckpoint, EarlyStopping
9  from yolo3.model import preprocess_true_boxes, yolo_body, tiny_yolo_body, yolo_loss
10 from yolo3.utils import get_random_data
11
12
13 def _main():
14     annotation_path = 'train.txt'
15     log_dir = 'logs/000/'
16     classes_path = 'model_data/voc_classes.txt'
17     anchors_path = 'model_data/yolo_anchors.txt'
18     class_names = get_classes(classes_path)
19     anchors = get_anchors(anchors_path)
20     input_shape = (512,512) # multiple of 32, hw
21     model = create_model(input_shape, anchors, len(class_names))
22     train(model, annotation_path, input_shape, anchors, len(class_names), log_dir=log_dir)
23
```

設定好路徑後，使用 CMD 輸入 python train.py

## 5.訓練過程

```
Epoch 88/100
4/4 [=====] - 2s 587ms/step - loss: 120.0867 - val_loss: 140.8875
Epoch 89/100
4/4 [=====] - 2s 598ms/step - loss: 123.1705 - val_loss: 132.5142
Epoch 90/100
4/4 [=====] - 2s 592ms/step - loss: 124.0760 - val_loss: 113.8043
Epoch 91/100
4/4 [=====] - 2s 597ms/step - loss: 134.1238 - val_loss: 138.8856
Epoch 92/100
4/4 [=====] - 2s 595ms/step - loss: 123.5458 - val_loss: 140.2670
Epoch 93/100
4/4 [=====] - 2s 600ms/step - loss: 129.7838 - val_loss: 126.4146
Epoch 94/100
4/4 [=====] - 2s 595ms/step - loss: 128.8952 - val_loss: 128.3808
Epoch 95/100
4/4 [=====] - 2s 590ms/step - loss: 121.4716 - val_loss: 157.2316
Epoch 96/100
4/4 [=====] - 2s 585ms/step - loss: 125.1414 - val_loss: 150.5125
Epoch 97/100
4/4 [=====] - 2s 594ms/step - loss: 121.9227 - val_loss: 145.9632
Epoch 98/100
4/4 [=====] - 2s 584ms/step - loss: 147.4830 - val_loss: 154.8668
Epoch 99/100
4/4 [=====] - 2s 592ms/step - loss: 119.8548 - val_loss: 100.8764
Epoch 100/100
4/4 [=====] - 2s 580ms/step - loss: 131.2757 - val_loss: 148.3926
```

## 6.結束後進行測試

在 cmd 中輸入 python yolo\_video.py --image

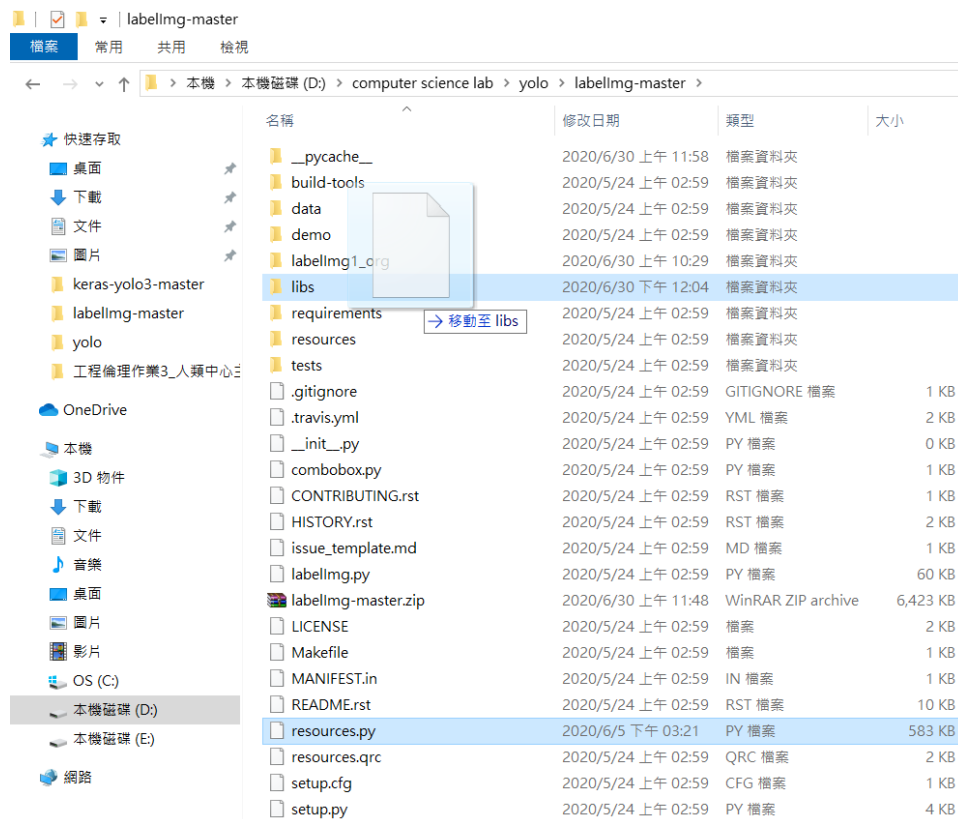




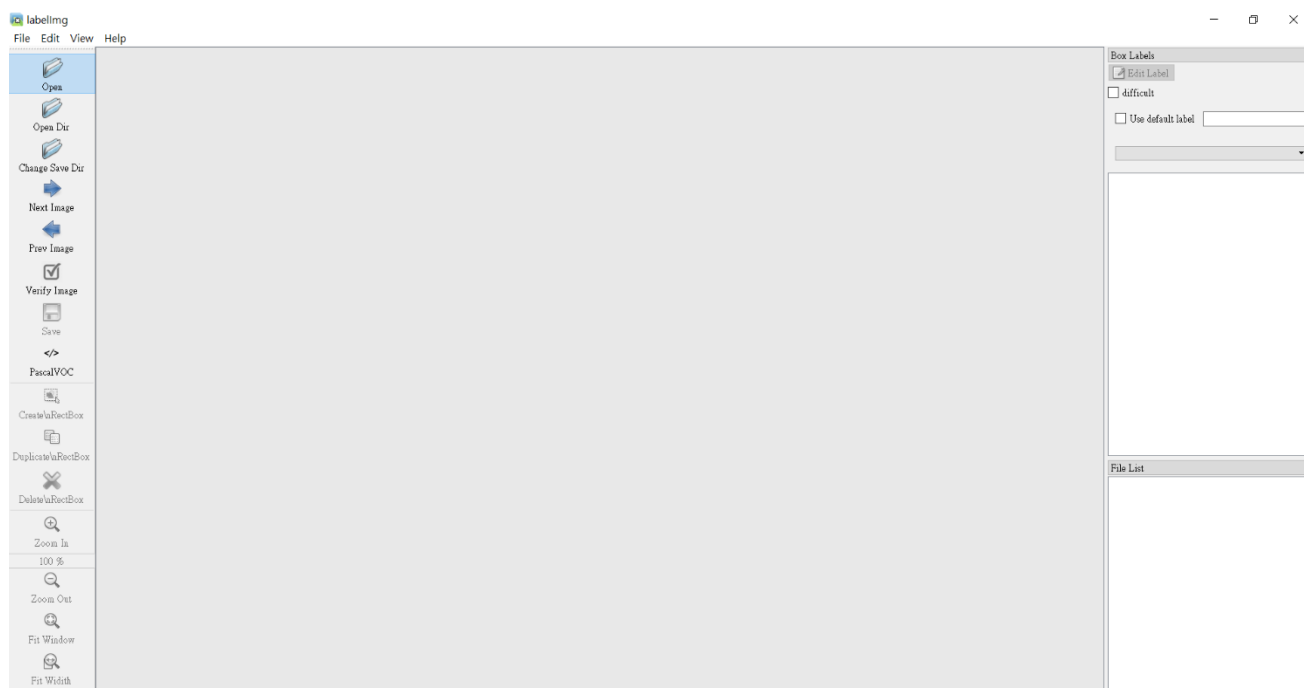
的)放進 libs 資料夾解決 ModuleNotFoundError 問題，就可以正常執行了)

2.從 cmd (Anaconda Prompt(testAI)開啟 labelImg，使用: `python labelImg.py`  
ModuleNotFoundError: No module named 'libs.resources'

[https://blog.csdn.net/mo\\_38027013/article/details/93597145](https://blog.csdn.net/mo_38027013/article/details/93597145)

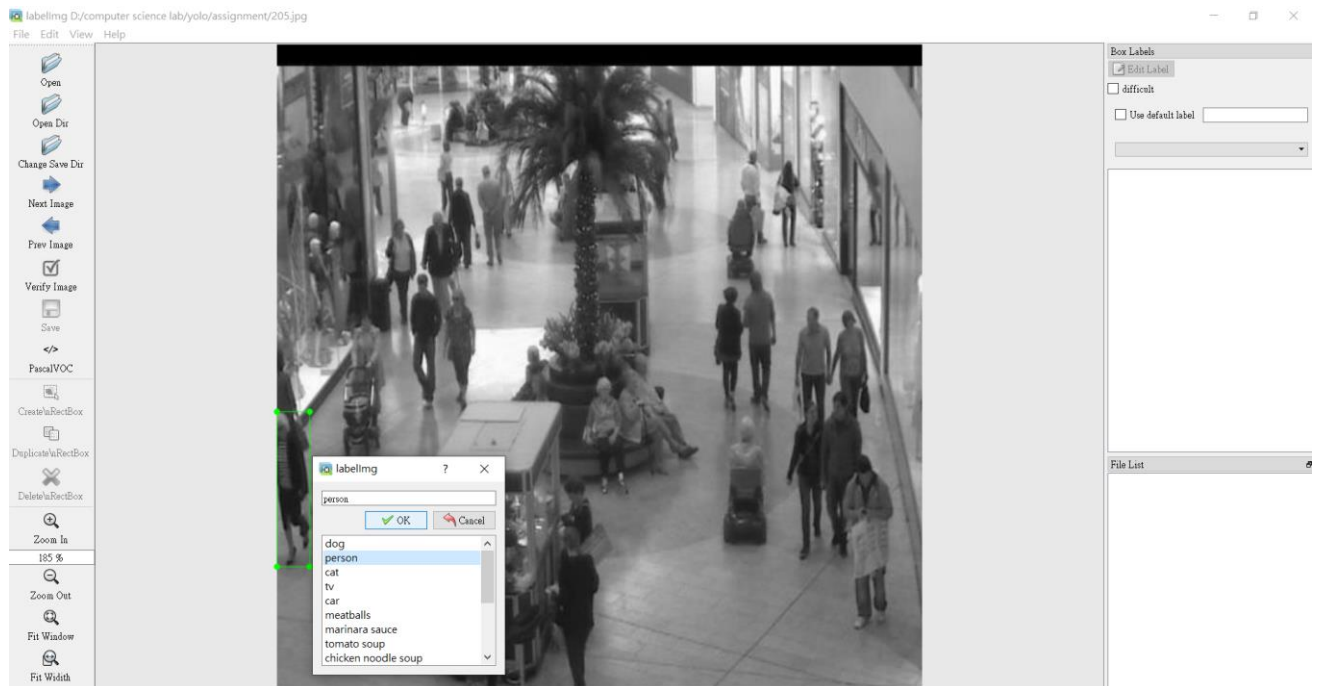


3.開啟 labelImg.py 後點 OPEN 開啟需要編輯的圖片



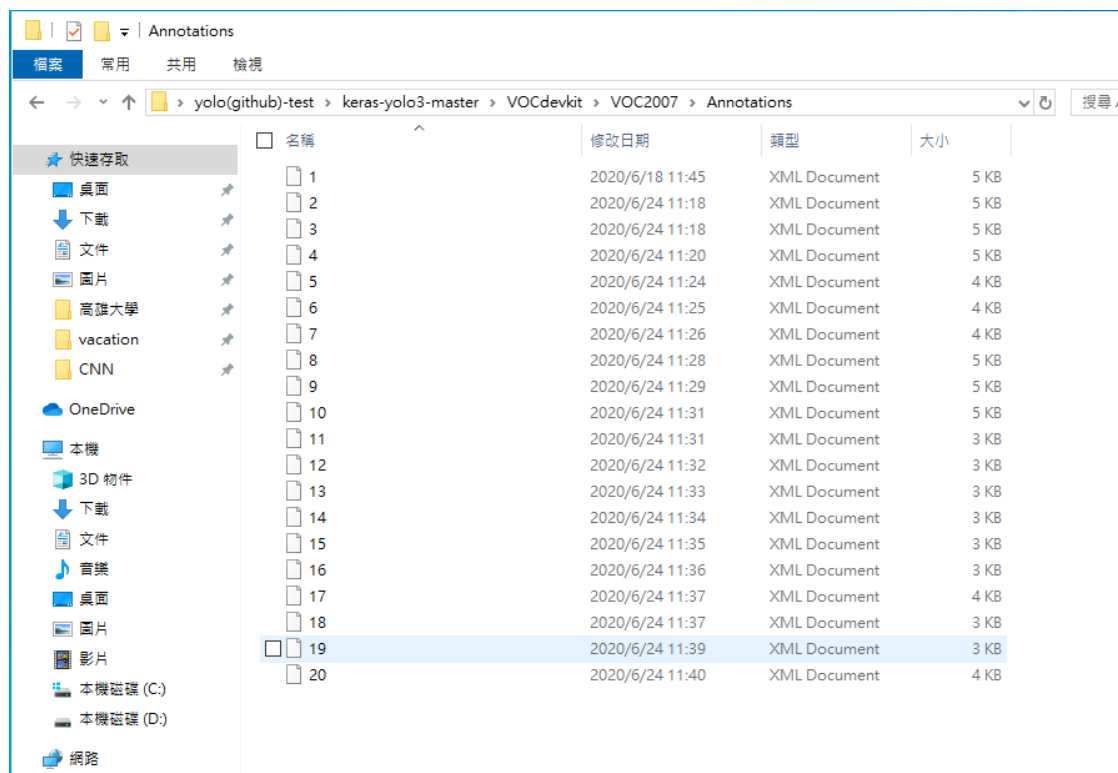


## 1)使用 Create 進行框人的動作



## 2)之後另存新檔到 xml(任意資料夾)中





問題:

1. 尺寸問題(推薦使用 **Mall- 512\*512** 的影像 )
2. 是否可以判斷影像中有多少框框
3. 資料量不足
4. 訓練量不夠，導致有些影像無法成功辨識

參考網址

1. <https://blog.csdn.net/Thomson617/article/details/99972062>
2. <https://www.itread01.com/content/1542118573.html>
3. <https://github.com/qqwweee/keras-yolo3>
4. [https://blog.csdn.net/patrick\\_Lxc/article/details/80615433](https://blog.csdn.net/patrick_Lxc/article/details/80615433)