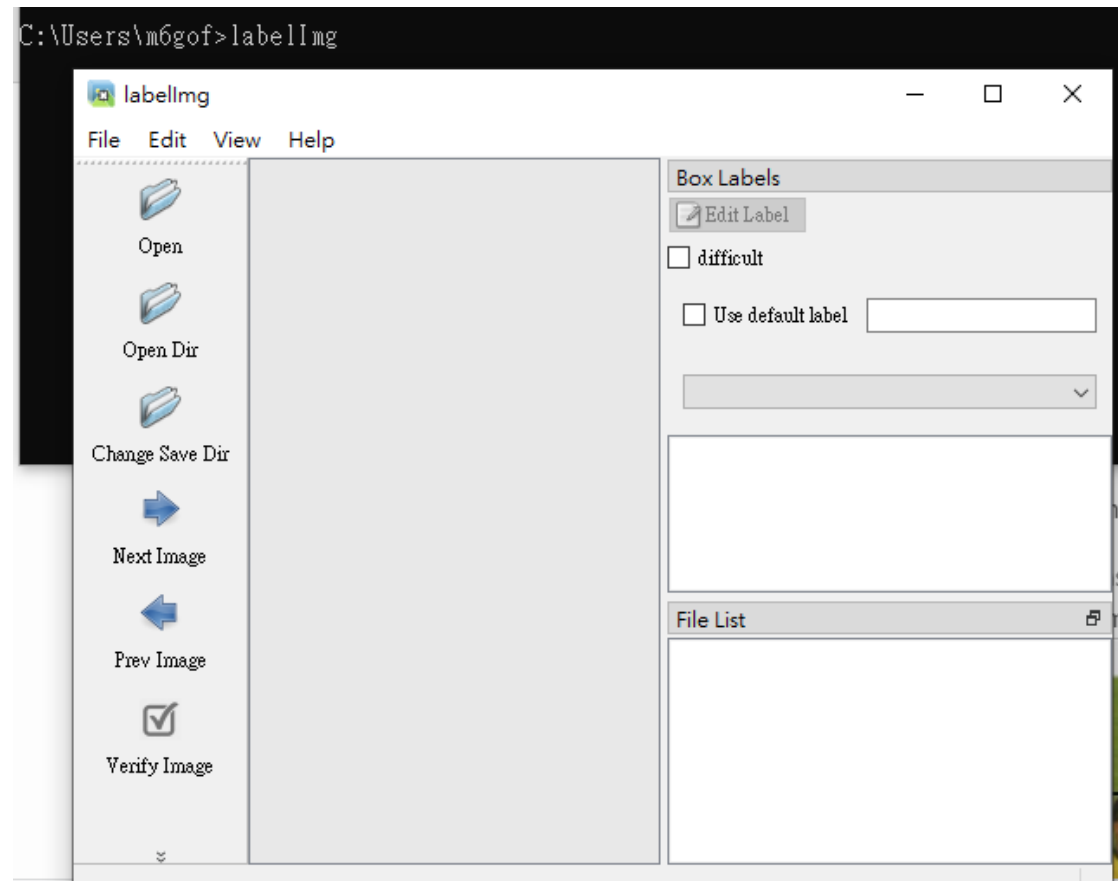


先安裝標記工具 labellmg

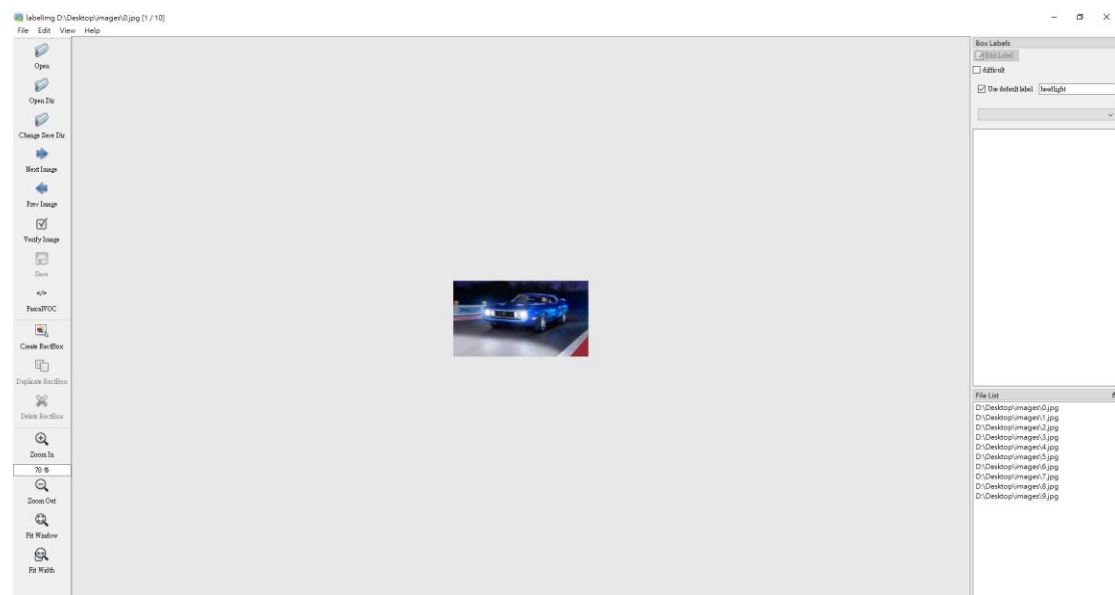
pip install labellmg

labellmg

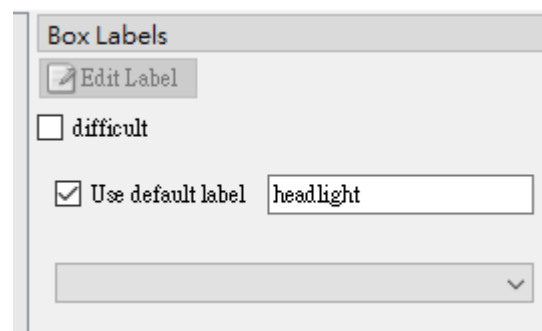
```
C:\Users\m6gof>pip install labellmg
Collecting labellmg
  Downloading labellmg-1.8.6.tar.gz (247 kB)
    ----- 247.7/247.7 kB 1.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting PyQt5
  Downloading PyQt5-5.15.7-cp37-abi3-win_amd64.whl (6.8 MB)
    ----- 6.8/6.8 MB 10.3 MB/s eta 0:00:00
Requirement already satisfied: lxml in c:\users\m6gof\appdata\local\programs\python\python39\lib\site-packages (from labellmg) (4.9.1)
Collecting PyQt5-sip<13,>=12.11
  Downloading PyQt5_sip-12.11.0-cp39-cp39-win_amd64.whl (78 kB)
    ----- 78.3/78.3 kB 2.2 MB/s eta 0:00:00
Collecting PyQt5-Qt5>=5.15.0
  Downloading PyQt5_Qt5-5.15.2-py3-none-win_amd64.whl (50.1 MB)
    ----- 50.1/50.1 MB 5.5 MB/s eta 0:00:00
Building wheels for collected packages: labellmg
  Building wheel for labellmg (setup.py) ... done
  Created wheel for labellmg: filename=labellmg-1.8.6-py2.py3-none-any.whl size=261529 sha256=3166118a367e4986a83f652d4de3b1bc20faf5730d65adc5fb64e8d7c656b5ea
  Stored in directory: c:\users\m6gof\appdata\local\pip\cache\wheels\df\39\12\bf80c441ad46616d5b271867a6ece21e8a083af10903b8e53a
Successfully built labellmg
Installing collected packages: PyQt5-Qt5, PyQt5-sip, PyQt5, labellmg
Successfully installed PyQt5-Qt5-5.15.2 PyQt5-sip-12.11.0 labellmg-1.8.6 PyQt5-5.15.7
C:\Users\m6gof>
```



## Ctrl+U 選擇訓練圖片資料夾



右上角先填入 label 名稱，叫什麼都可以，我這裡叫 headlight



案 W 進行標記，框出目標位置即可(Ctrl+S 保存 XML 檔案，這邊注意圖片是放在 images 資料夾底下，XML 檔案室放在 labels 資料夾底下，如下圖所示)



headlight				
	名稱	修改日期	類型	大小
	cascade_training	2022/11/8 下午 12:44	檔案資料夾	
	images	2022/11/8 下午 12:01	檔案資料夾	
	labels	2022/11/8 下午 12:01	檔案資料夾	
	1_labels_to_pos_neg_imgs	2022/11/8 下午 12:04	Python File	6 KB
	2_generate-negatives	2022/11/8 下午 12:06	Python File	4 KB
	3_augmentation	2022/11/8 下午 12:07	Python File	3 KB
	4_add_aug_positives_to_list	2022/11/8 下午 12:11	Python File	1 KB

標記完全部圖片後，依序執行

**1\_labels\_to\_pos\_neg\_imgs.py:** 此程式目的是將所有相片中的標記框取出另外存到一個資料夾 (cascade\_training) 下，這些圖片稱為 **positives**，圖片中不含標記框的其它區域則存到 **neg\_bg** 資料夾中，這些與貓臉無關的圖片稱為 **negatives**。

**2\_generate-negatives.py:** 本程式功能為使用 **sliding window** 方式，把 **neg\_bg** 資料夾下的相片切裁為指定大小的 **negatives** 圖片，並產生一個 **negatives.info** 檔案。








**3\_augmentation.py:** 本程式使用 **augmentation** 強化資料方式產生更多正向圖片，新增加的圖片放置於 **aug\_positives** 資料夾下。

**4\_add\_aug\_positives\_to\_list.py:** 將前一步所產生的圖片放到 **positives.info** 檔案。

已經幫你放到相對位置，直接執行就好(如果有相關 **library** 沒安裝報錯，再安裝就好)

接下來要產生訓練用的 **VEC** 檔 **Cascade Classifier** 不能直接讀取圖片檔，我們必須轉為 **VEC** 檔案才能開始訓練。

進到 **cascade\_training/**資料夾底下

 cascade_training	2022/11/8 下午 12:44	檔案資料夾	
 images	2022/11/8 下午 12:01	檔案資料夾	
 labels	2022/11/8 下午 12:01	檔案資料夾	
 1_labels_to_pos_neg_imgs	2022/11/8 下午 12:04	Python File	6 KB
 2_generate-negatives	2022/11/8 下午 12:06	Python File	4 KB
 3_augmentation	2022/11/8 下午 12:07	Python File	3 KB
 4_add_aug_positives_to_list	2022/11/8 下午 12:11	Python File	1 KB

在終端機輸入(我是用 **windows** 的 **cmd**)

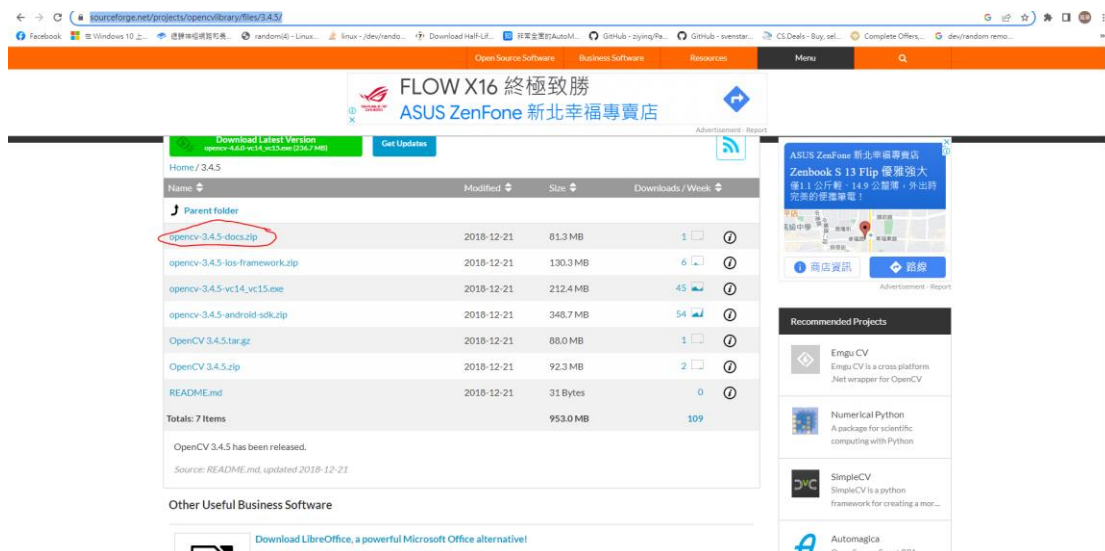
**opencv\_createsamples -info positives.info -vec samples.vec -w 64 -h 64 -num 91**

(其中 **-num** 數量依照 **positives.info** 內的檔案數決定 )

如果是 **opencv\_createsamples** 有問題，再去

<https://sourceforge.net/projects/opencvlibrary/files/3.4.5/>

下載對應作業系統的資料，我是 **windows** 所以我下載的是這個，這取決於你是什麼作業系統



下載完去到 **opencv** 底下找到所需的 **opencv\_createsamples** 和 **opencv\_traincascade** 檔案

本機 > 文件 > Downloads > opencv > build > x64 > vc15 > bin				
名稱	修改日期	類型	大小	
opencv_annotation	2018/12/21 下午 11:31	應用程式	47 KB	
opencv_createsamples	2018/12/21 下午 11:31	應用程式	54 KB	
opencv_ffmpeg345_64.dll	2018/12/21 下午 11:18	應用程式擴充	17,635 KB	
opencv_interactive-calibration	2018/12/21 下午 11:31	應用程式	131 KB	
opencv_traincascade	2018/12/21 下午 11:31	應用程式	314 KB	
opencv_version	2018/12/21 下午 11:31	應用程式	34 KB	
opencv_version_win32	2018/12/21 下午 11:31	應用程式	32 KB	
opencv_visualisation	2018/12/21 下午 11:31	應用程式	59 KB	
opencv_world345.dll	2018/12/21 下午 11:31	應用程式擴充	75,191 KB	
opencv_world345	2018/12/21 下午 11:31	Program Debug ...	28,844 KB	
opencv_world345d.dll	2018/12/21 下午 11:25	應用程式擴充	117,636 KB	
opencv_world345d	2018/12/21 下午 11:25	Program Debug ...	123,908 KB	

最好像我這樣直接複製到訓練路徑下，可以省下很多打字時間

headlight > cascade_training				
名稱	修改日期	類型	大小	
aug_positives	2022/11/8 下午 12:08	檔案資料夾		
neg_bg	2022/11/8 下午 12:04	檔案資料夾		
negatives	2022/11/8 下午 12:06	檔案資料夾		
output	2022/11/8 下午 02:39	檔案資料夾		
positives	2022/11/8 下午 12:11	檔案資料夾		
negatives.info	2022/11/8 下午 12:06	INFO 檔案	7 KB	
opencv_createsamples	2018/12/21 下午 11:31	應用程式	54 KB	
opencv_traincascade	2018/12/21 下午 11:31	應用程式	314 KB	
opencv_world345.dll	2018/12/21 下午 11:31	應用程式擴充	75,191 KB	
positives.info	2022/11/8 下午 12:40	INFO 檔案	4 KB	
samples.vec	2022/11/8 下午 12:40	VEC 檔案	729 KB	

最後打這行就可以進行模型訓練了(如果要開始新的一輪訓練，要記得刪除 **output** 資料夾內的所有東西，不然無法訓練)

**opencv\_traincascade -data cascade\_training -vec samples.vec -bg negatives.info -numPos 42 -numNeg 58 -numStages 8 -minHitRate 0.995 -maxFalseAlarmRate 0.3 -w 64 -h 64 -featureType LBP**

(-numPos 42 這個值要去看 **positives.info** 這個檔案有多少行來決定)

(- numNeg 42 這個值要去看 **negatives.info** 這個檔案有多少行來決定)

(-numStages 8 最大訓練次數)

(-minHitRate 0.995 分類器每個階段的最小期望命中率，調越高表示你希望模型表現更好但時間也越久)

(-maxFalseAlarmRate 0.3 分類器每個階段的最大期望誤報率，調越低表示你希望模型表現更好但時間也越久)

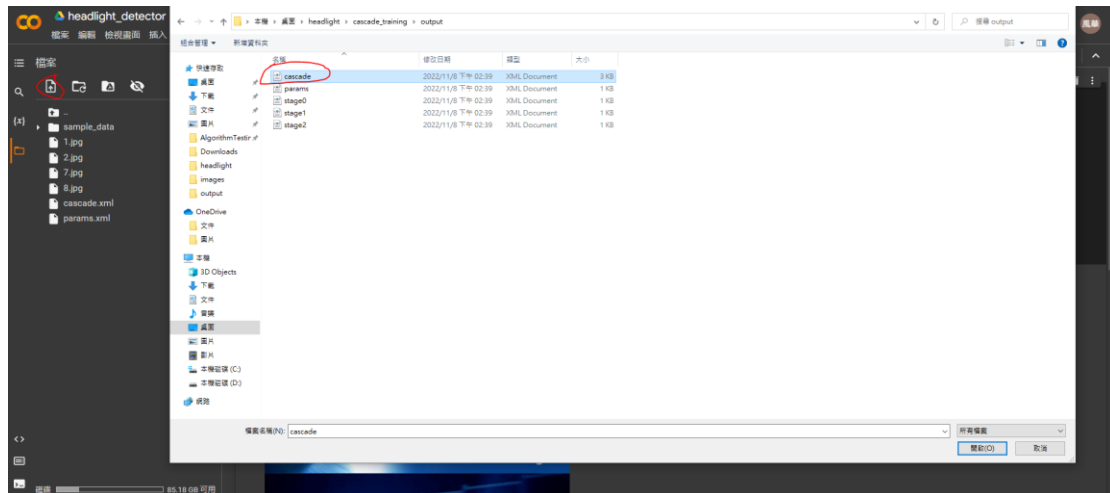
aug_positives	2022/11/8 下午 12:08	檔案資料夾	
neg_bg	2022/11/8 下午 12:04	檔案資料夾	
negatives	2022/11/8 下午 12:06	檔案資料夾	
output	2022/11/8 下午 02:39	檔案資料夾	
positives	2022/11/8 下午 12:11	檔案資料夾	
negatives.info	2022/11/8 下午 12:06	INFO 檔案	7 KB
opencv_createsamples	2018/12/21 下午 11:31	應用程式	54 KB
opencv_traincascade	2018/12/21 下午 11:31	應用程式	314 KB
opencv_world345.dll	2018/12/21 下午 11:31	應用程式擴充	75,191 KB
positives.info	2022/11/8 下午 12:40	INFO 檔案	4 KB
samples.vec	2022/11/8 下午 12:40	VEC 檔案	729 KB

最後訓練完，我們到 **output** 資料夾底下看看

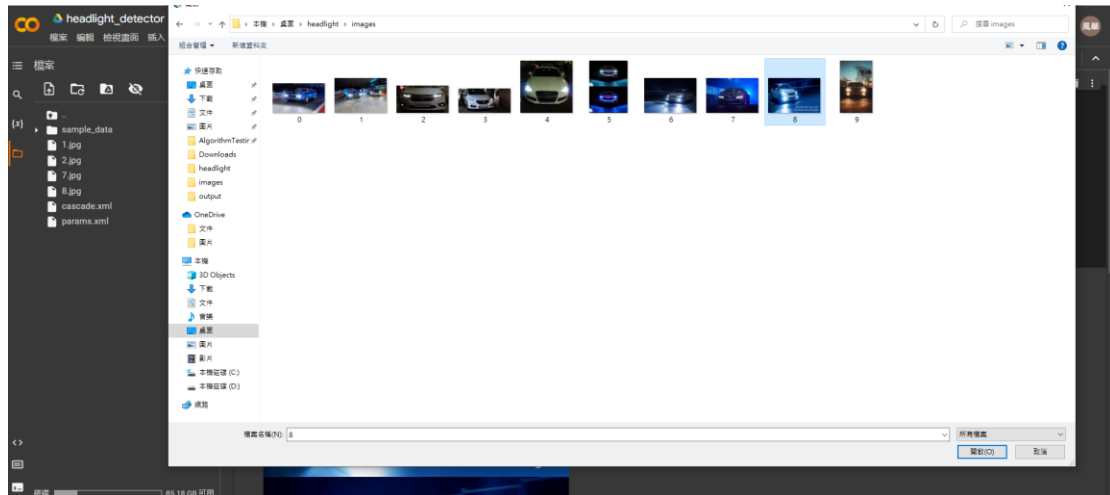
cascade	2022/11/8 下午 02:39	XML Document	3 KB
params	2022/11/8 下午 02:39	XML Document	1 KB
stage0	2022/11/8 下午 02:39	XML Document	1 KB
stage1	2022/11/8 下午 02:39	XML Document	1 KB
stage2	2022/11/8 下午 02:39	XML Document	1 KB

會有一個叫 **cascade.xml** 的分類器，它能幫我們框出目標物所在

最後在 colab 執行 `headlight_detector.ipynb` 檔案看輸出結果



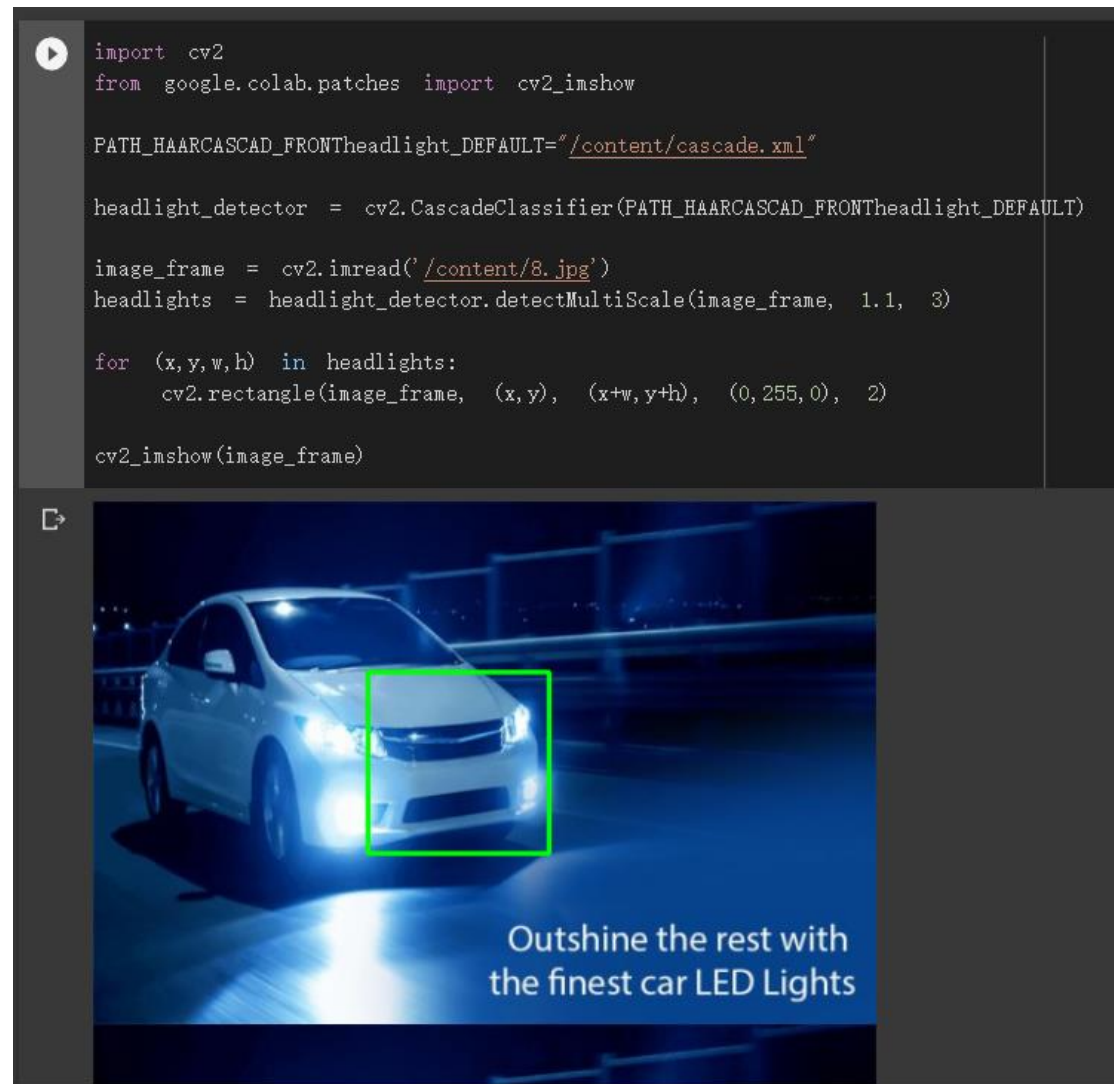
點左邊上傳圖標，選擇 `output` 資料夾下的 `cascade.xml`



在上傳想要預測的圖片

然後這邊輸入想要預測的圖片名稱(`content` 是 colab 的預設路徑不用管它)

```
image_frame = cv2.imread('/content/8.jpg')
```



然後 run 程式碼，就可以看到輸出結果了  
效果好不好取決於你的訓練集大小和你的訓練難度