

成 功 大 學
系 統 及 船 舶 機 電 工 程 學 系

數位影像處理

Project 3

授課老師：江珮如

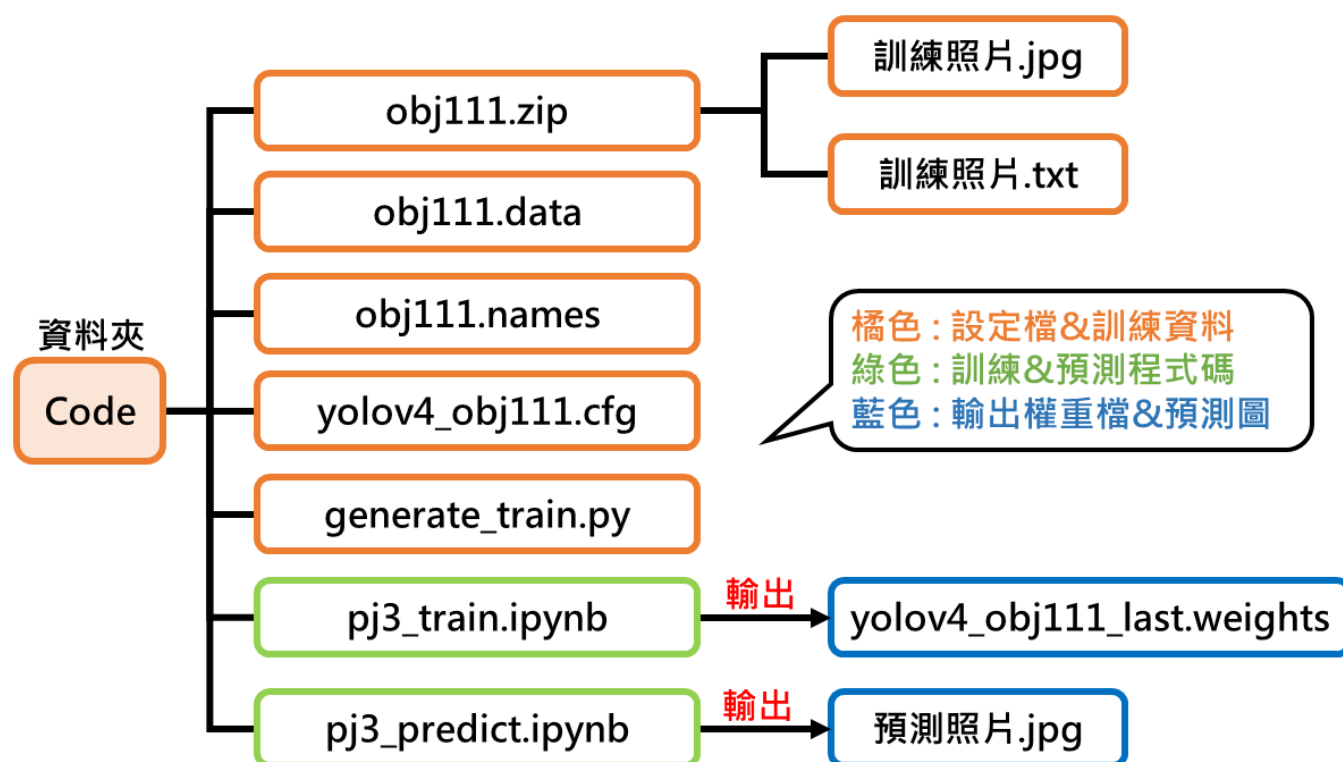
系 級：系統 112

學 生：周呈陽

學 號：F14081046

中 華 民 國 一 百 一 十 一 年 六 月 十 四 日

流程圖



流程說明

這次我使用了 labelIMG 和 Google Colab Pro 來進行 project 3，這次我使用到了 yolov4 物件偵測的技術，因此在訓練圖檔前須先準備 yolov4 所需資料，雖然 project 3 有分兩個部分，但在程式編譯上並無差異，兩者僅在 yolov4_obj111.cfg 的設定檔內有參數上的不同，我也會在下方一一介紹不同檔案的用處，介紹如下：

1. Code 資料夾

這次因為使用了 yolov4 的技術進行影像處理，因此

須建立一個資料夾來存放 yolov4 所需要的訓練資料和設定檔。

2. obj111.zip

此壓縮檔為所需要訓練的檔案，可以看到壓縮檔裡有 jpg 和 txt 兩個檔案類別，這次我使用了 labelIMG 的軟體進行圖片的標註，jpg 為訓練圖檔，txt 是圖檔標註後的位置資料。

3. obj111.names

此檔內容為訓練圖檔的 label 列表，本次 project 的 label 為 nail，yolov4 在訓練與預測時皆需要讀取此檔。

4. obj111.data

定義 label 數目、不同設定檔、和權重檔 weights 目錄的路徑，yolov4 在訓練與預測時皆需要讀取此檔。

5. yolov4_obj111.cfg

yolo 模型的設定檔，此模型的原模型為 yolov4，本次的 project 有兩個部分，因此其中的參數我有進行個別的調整，呈現如下：

- i. **Part 1**：因為自己拍攝的 100 張指甲照片中，指甲大小變化不大，所以我將 batch 設為 16、subdivisions 設

為 8。

- ii. **Part 2**：老師提供的指甲照片，因為指甲大小變化很大，故我將 batch 設為 64、subdivisions 設為 32，希望能透過增加每 batch 學習的樣本資料和 mini-batch 的數量，來提升模型的準確度。

6. generate_train.py

將訓練圖檔分割成 train.txt 和 test.txt。

7. pj3_train.ipynb

此檔為進行 google 雲端的環境建置和訓練檔，因為這次的 project 我是在 colab 上進行編譯，故需先在 colab 上建置 darknet 的相關環境，再讀取訓練資料和設定檔，便能進行模型的訓練。

8. yolov4_obj111_last.weights

此 weights 權重檔為訓練後的結果，在 yolov4 進行預測時會使用到。

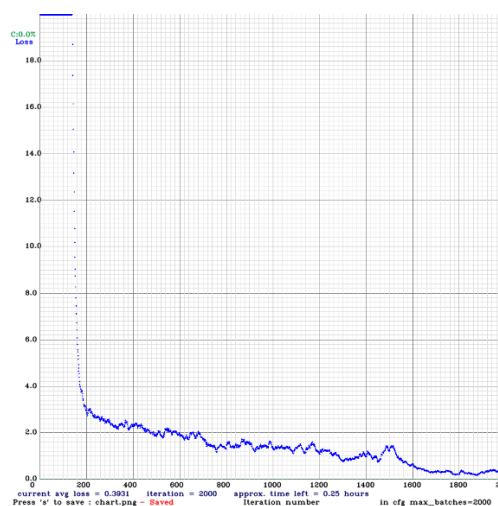
9. pj3_predict.ipynb

讀取 yolov4 模型(yolov4_obj111.cfg)、訓練完的權重檔(yolov4_obj111_last.weight)和待預測的圖檔，即可進行預測，並輸出預測圖檔。

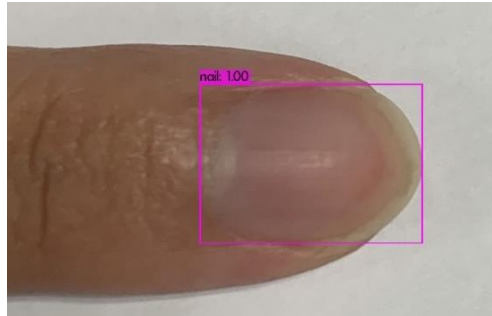
結果分析

1. Part 1 結果分析

Yolov4 的結果比預期來的好，從圖(一)可以看到最後的 loss 收斂到 0.3931，因為之前只有使用機器學習的相關經驗，因此一開始單就 loss 上來判斷，會覺得預期照片成效不會太好，但跑完 100 張圖後發現結果圖的準度都高達 0.99，同時也能準確地將指甲的位置框起來，如圖(二)所呈現，其最主要的因為這 100 張圖片內的內容都差不多，例如指甲大小、白底、環境光等等，對於 yolov4 的訓練也相對單純，故結果呈現也比預期來的好很多。



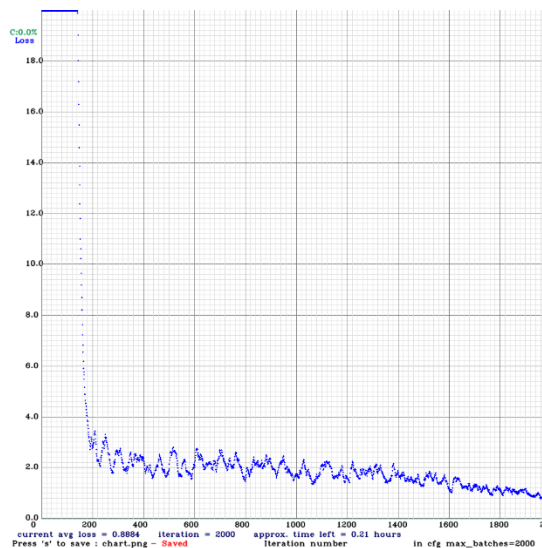
圖(一) Part 1 的 loss 收斂圖



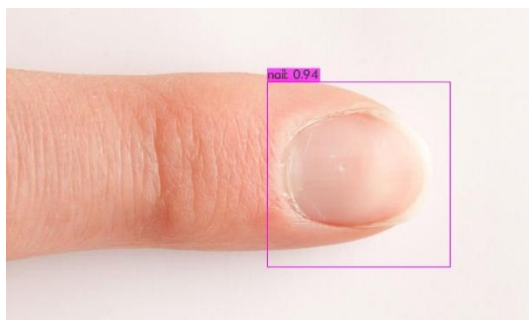
圖(二) Part 1 的結果圖

2. Part 2 結果分析

相對 part 1 來說，part 2 的訓練圖檔圖片大小不一、環境不同、指甲大小也不一樣，故在訓練的時候我將 batch 和 subdivisions 調大一點，希望能解決這樣的問題，最後可以看到圖(三)的 loss 收斂到 0.8884，結果的呈現也算還行，預測準確度最低也有 0.85，雖然無法很準確地將指甲框出來，但都能框出指甲的位置，如圖(四)所呈現。



圖(三) Part 2 的 loss 收斂圖



圖(四) Part 2 的結果圖

3. Part 2 討論

雖然結果呈現的還不錯，但若要能做到與 part 1 一樣的準確度，我覺得可以透過增加 database、提高 batch 和 subdivision 的數值，首先增加 database 就是將更多不同背景資訊的指甲一起訓練，這樣也能讓模型在訓練時，能將判讀區分更多種類的雜訊；其次，提高 batch 能讓每個 batch 學習採用更多的樣本資料，提升 subdivision 則是產生更多的 mini-batch 進行梯度下降法，以此來提高模型的準確度。

程式碼

pj3_train.ipynb

```
from google.colab import drive
drive.mount('/content/drive')

!git clone https://github.com/AlexeyAB/darknet
%cd '/content/darknet'
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
```

```

!usr/local/cuda/bin/nvcc --version
!make
!ls
!cp /content/drive/Shareddrives/影像處理/yolov4_project3/obj111.zip ../
!unzip ../obj111.zip -d data/
!cp /content/drive/Shareddrives/影像處理/yolov4_project3/yolov4_obj111.cfg ./cfg
!cp /content/drive/Shareddrives/影像處理/yolov4_project3/obj111.names ./data
!cp /content/drive/Shareddrives/影像處理/yolov4_project3/obj111.data ./data
!cp /content/drive/Shareddrives/影像處理/yolov4_project3/generate_train.py ./
!python generate_train.py
!./darknet detector train data/obj111.data cfg/yolov4_obj111.cfg -dont_show

```

pj3_predict.ipynb

```

from google.colab import drive
drive.mount('/content/drive')
!git clone https://github.com/AlexeyAB/darknet
%cd '/content/darknet'
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!usr/local/cuda/bin/nvcc --version
!make
!./darknet detect /content/drive/Shareddrives/影像處理
/chou/yolov4_obj111.cfg /content/drive/Shareddrives/影像處理
/chou/yolov4_obj111_last.weights /content/drive/Shareddrives/影像處理
/chou/1.JPG -dont show

```

generate_train.py

```

import os
image_files = []
os.chdir(os.path.join("data", "obj111"))
for filename in os.listdir(os.getcwd()):
    if filename.endswith(".jpg"):
        image_files.append("data/obj111/" + filename)
    if filename.endswith(".JPG"):
        image_files.append("data/obj111/" + filename)

```



```
os.chdir("..")
with open("train.txt", "w") as outfile:
    for image in image_files:
        outfile.write(image)
        outfile.write("\n")
    outfile.close()
os.chdir("..")
```

心得

這次 project 3 相對前兩次難了很多，之前因為也只接觸過機器學習，原本想從機器學習下手，但發現成效很差，因此蒐集資料後，找到了以物件導向為名的 yolov4，在訓練的過程裡起出因為 batch 和 subdivision 太大，Colab 的 GPU 限制也一度達到上限，後來就直接刷卡升級到 Colab Pro 讓他順利繼續訓練，最後的成果也算還不錯，其實最大的問題在於環境建設和設定檔的修正，花了不少時間和心力。這次的 project 讓我認識了 yolov4，話雖如此，但其理論知識真的很難，充其量只能說我會用而已，希望未來自己能再更深入研究這些訓練模型，感謝老師願意花時間回復我的問題，也謝謝助教花費時間批改我這學期的每一份 project 和作業，謝謝。

(Project3 完整壓縮檔：<https://reurl.cc/6ZxL9k>)