Lab09

Object Detection with yolo-v3

https://drive.google.com/drive/folders/1IC2_KnEkPKI48cdRuYuSiaZANzc2Ps57?usp=sharing

步驟(以win10為例)

1. 安裝Microsoft Visual Studio及Cmake GUI
2. 安裝darknet
3. 安裝yolo_mark
4. 用yolo_mark標註照片
5. 用darknet訓練model
6. 觀看結果

安裝Microsoft Visual Studio及Cmake GUI

- MSVC 2019或2017，助教測試用的是2017
- Cmake GUI
  https://cmake.org/download/

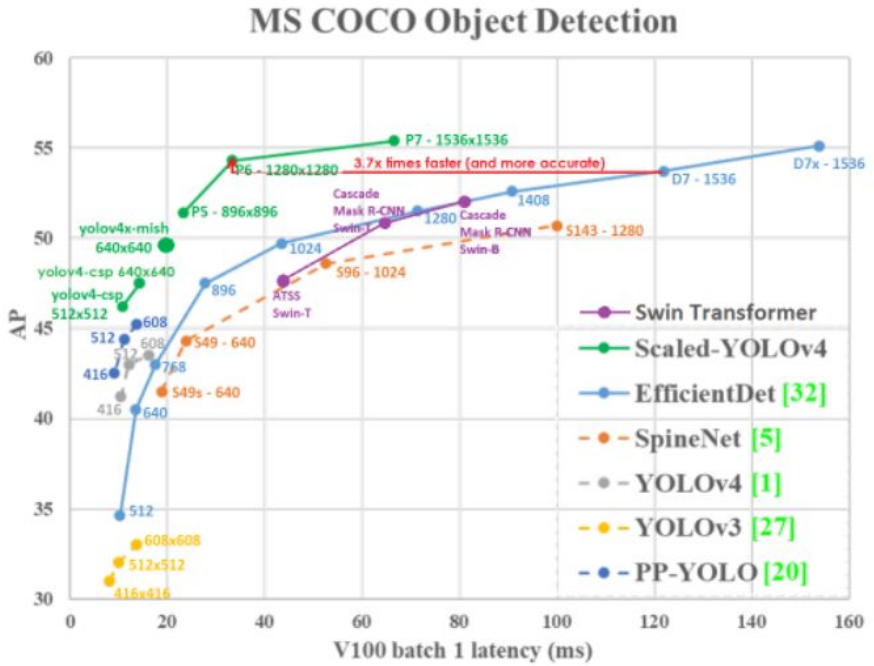
Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.23.1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.23.1.zip

Darknet

https://github.com/AlexeyAB/darknet

https://pjreddie.com/darknet/yolo/



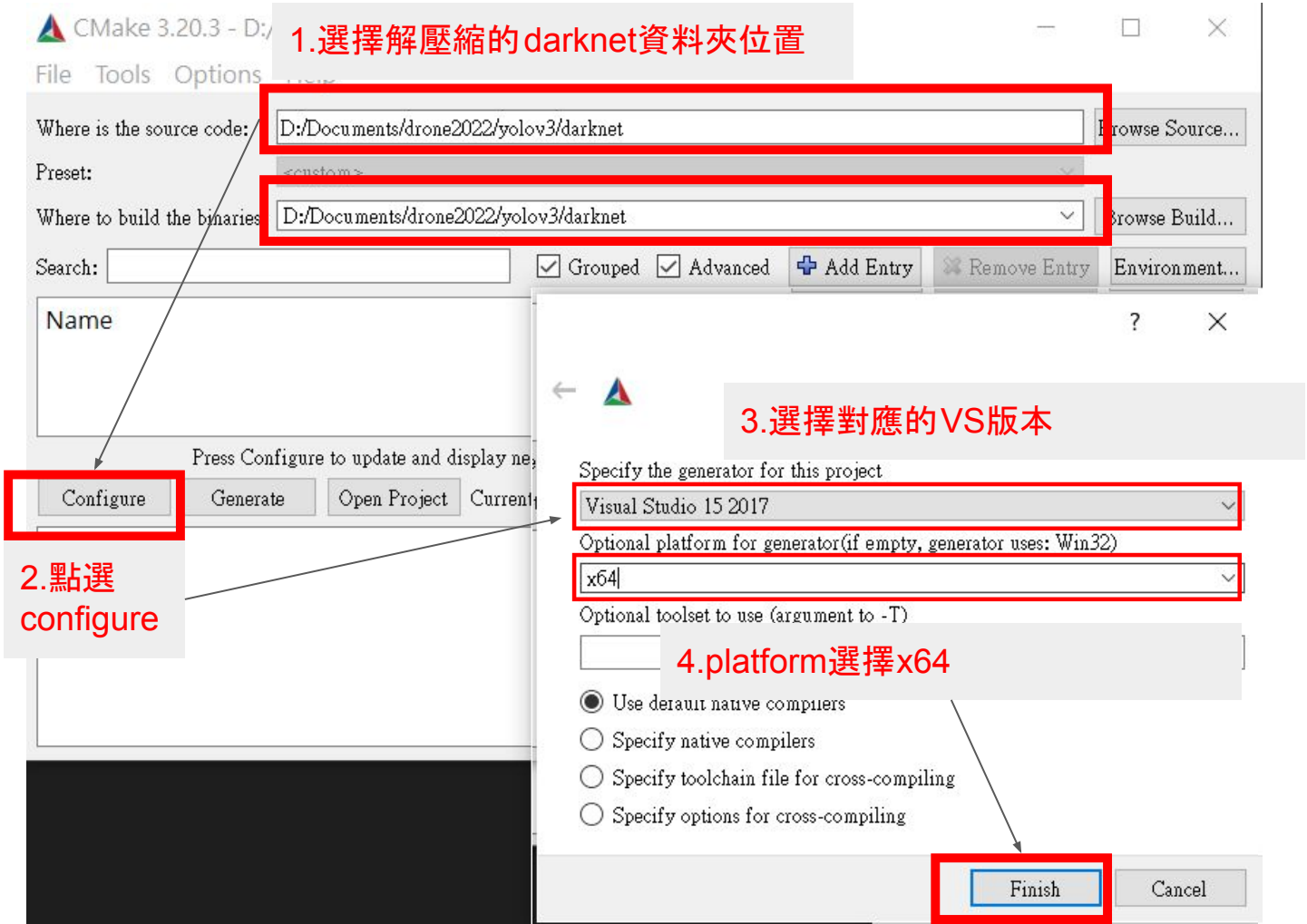
Requirements for Windows, Linux and macOS

- **CMake >= 3.18**: https://cmake.org/download/
- **Powershell** (already installed on windows): https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell
- **CUDA >= 10.2**: https://developer.nvidia.com/cuda-toolkit-archive (on Linux do Post-installation Actions)
- **OpenCV >= 2.4**: use your preferred package manager (brew, apt), build from source using vcpkg or download from OpenCV official site (on Windows set system variable `OpenCV_DIR` = `C:\opencv\build` - where are the `include` and `x64` folders image)
- **cuDNN >= 8.0.2** https://developer.nvidia.com/rdp/cudnn-archive (on **Linux** follow steps described here https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#installlinux-tar , on **Windows** follow steps described here https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#installwindows)
- **GPU with CC >= 3.0**: https://en.wikipedia.org/wiki/CUDA#GPUs_supported

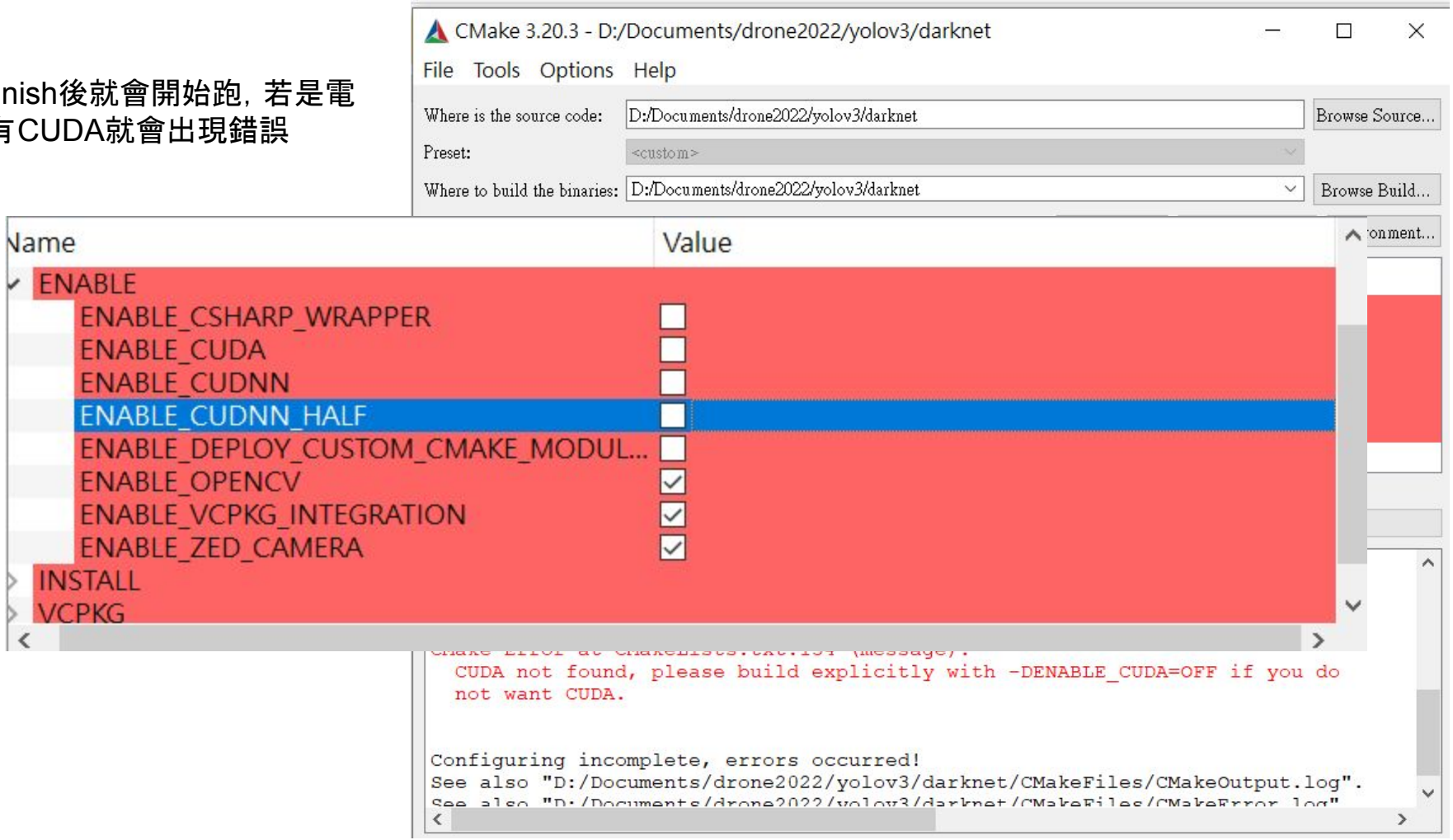
使用Cmake在Windows上安裝

下載darknet (https://github.com/AlexeyAB/darknet) 並解壓縮

開啟CmakeGUI

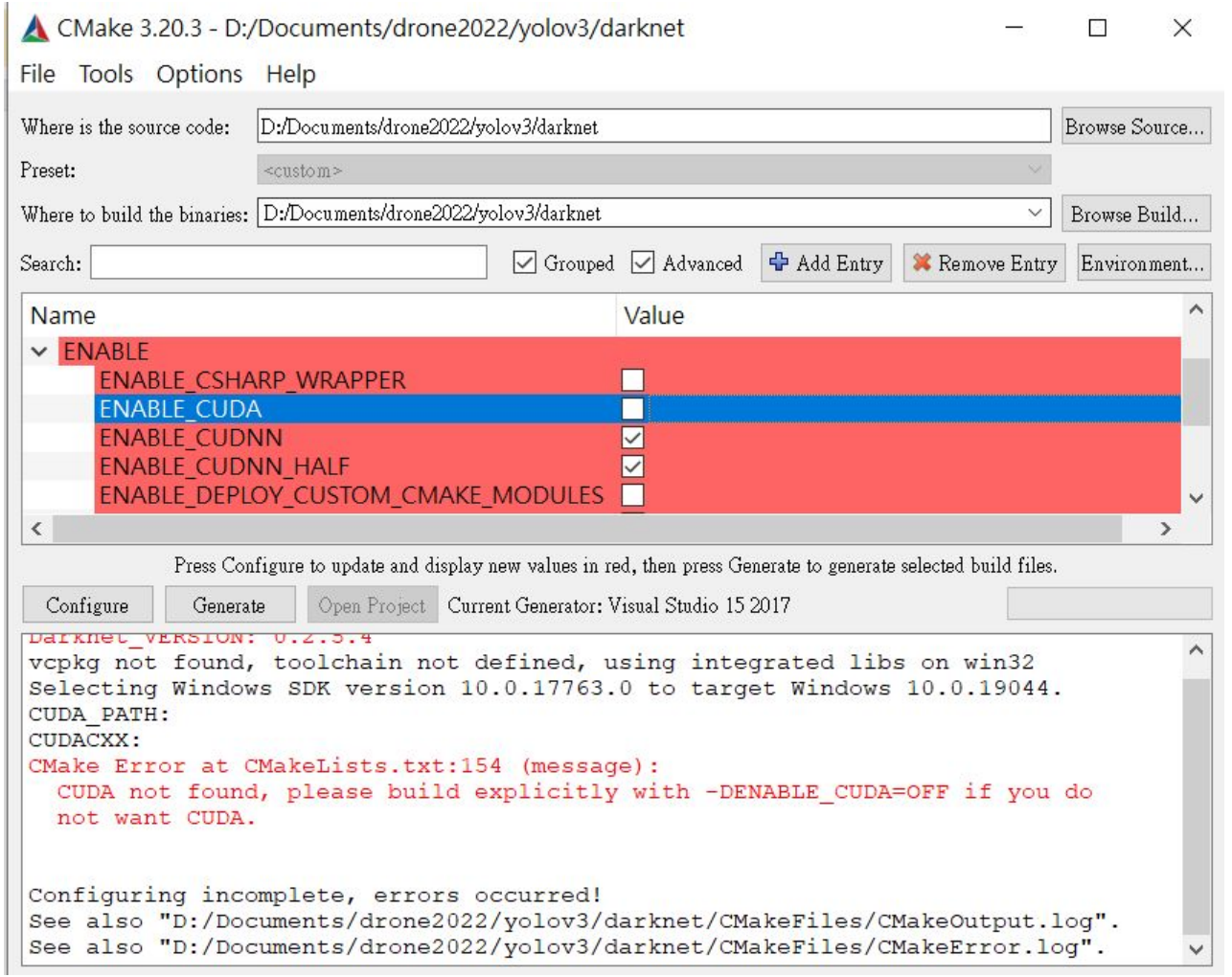

CMake 3.20.3 - D:/
File Tools Options
Where is the source code:
D:/Documents/drone2022/yolov3/darknet
Browse Source...
Preset:
Where to build the binaries
D:/Documents/drone2022/yolov3/darknet
Browse Build...
Search:
Grouped
Advanced
Add Entry
Remove Entry
Environment...
Name
Press Configure to update and display ne
Configure
Generate
Open Project
Current
1.選擇解壓縮的darknet資料夾位置
2.點選 configure
3.選擇對應的VS版本
Specify the generator for this project
Visual Studio 15 2017
Optional platform for generator(if empty, generator uses: Win32)
x64
Optional toolset to use (argument to -T)
4.platform選擇x64
Use default native compilers
Specify native compilers
Specify toolchain file for cross-compiling
Specify options for cross-compiling
Finish
Cancel

按下finish後就會開始跑，若是電腦沒有CUDA就會出現錯誤



ENABLE->ENABLE_CUDA
取消勾選

再次 Configure

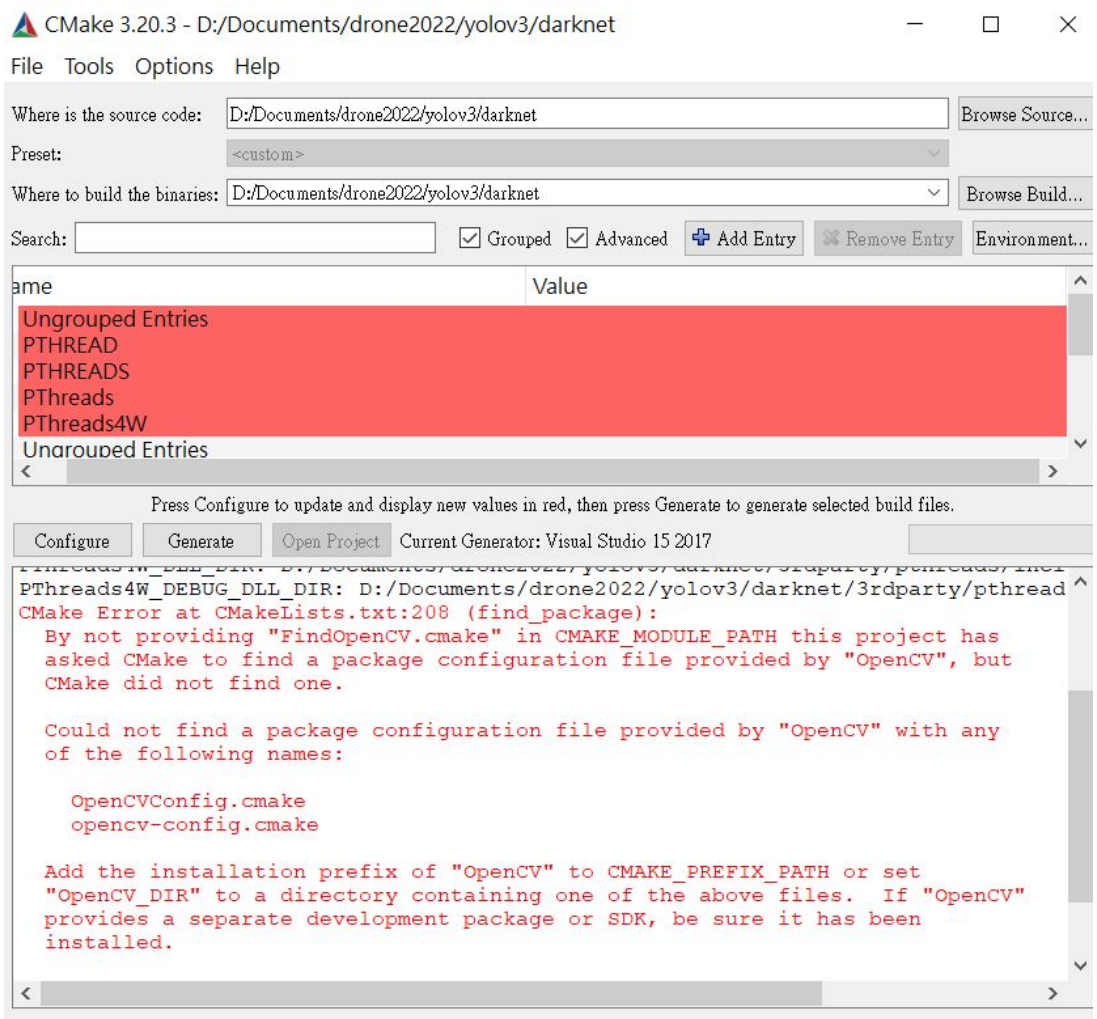


找不到opencv.cmake

打開darknet/CmakeList.txt，並
在上方加入Opencv的路徑

重新 configure

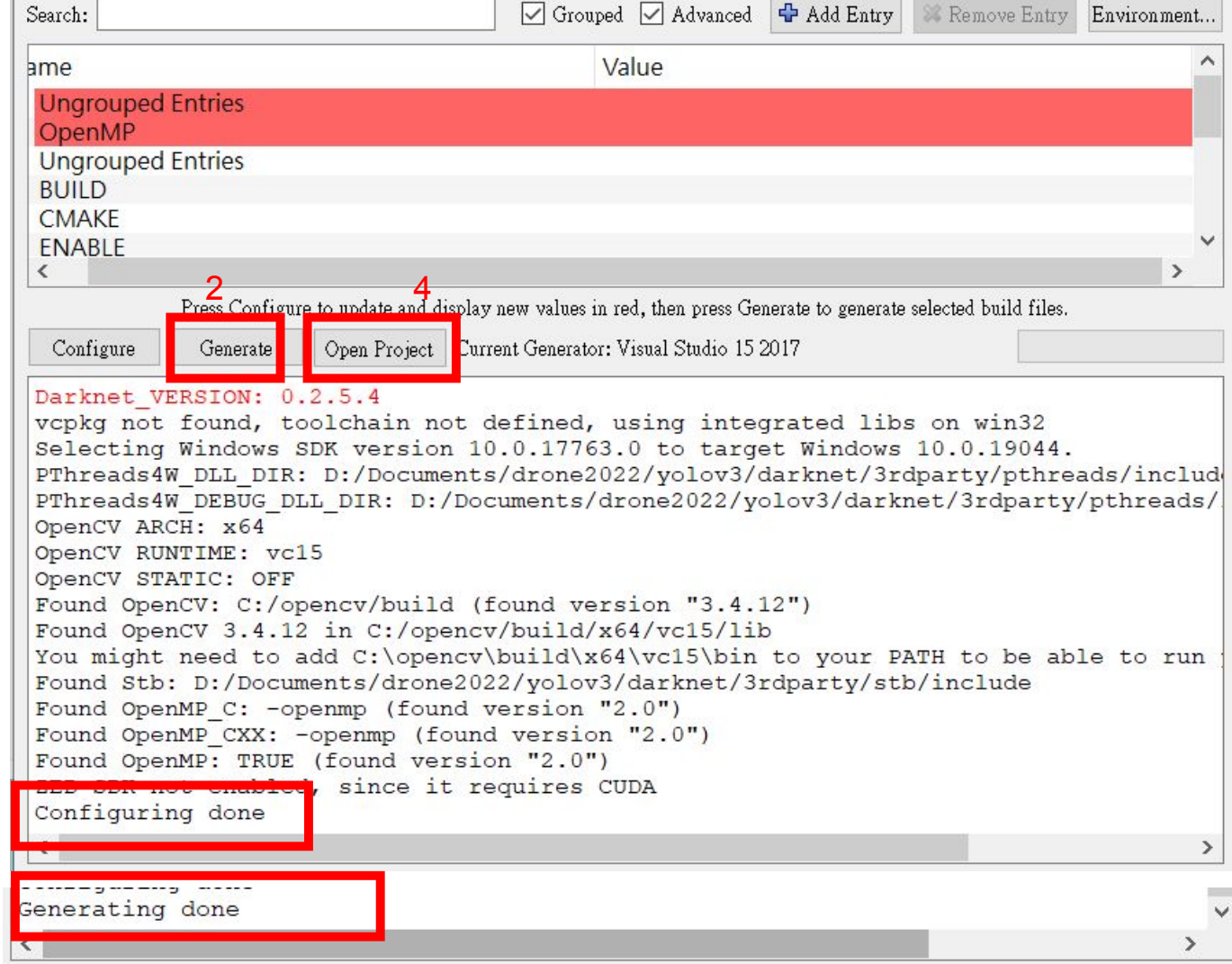
```
1  cmake_minimum_required(VERSION 3.18)
2  include(CMakeDependentOption)
3
4  set(OpenCV_DIR C:/opencv/build)
5  set(Darknet_MAJOR_VERSION 0)
6  set(Darknet_MINOR_VERSION 2)
7  set(Darknet_PATCH_VERSION 5)
```



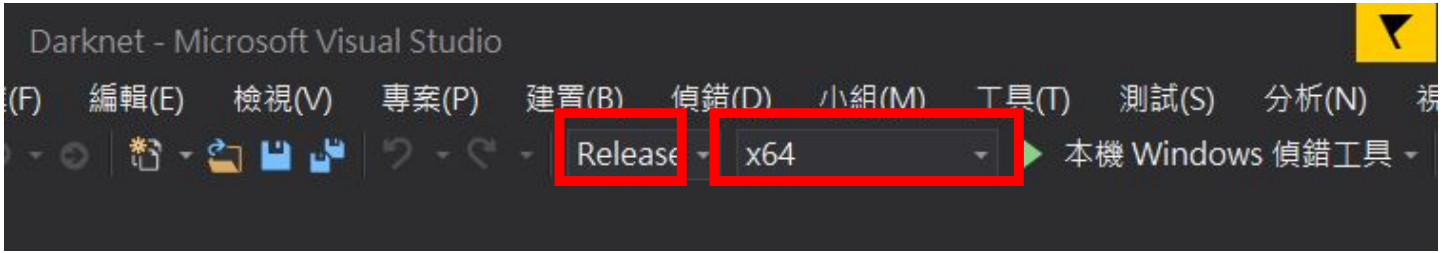
下方出現 Configuring done
表示成功

再點選Generate後會出現
Generating Done

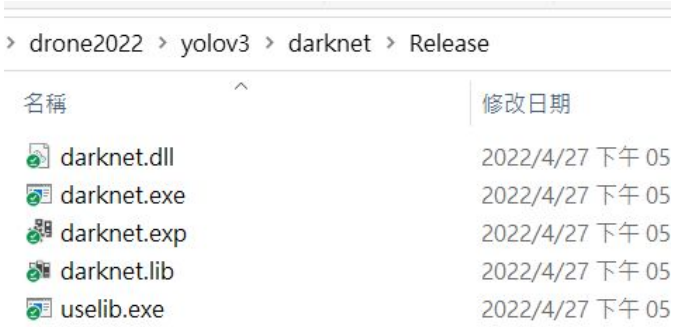
點選Open Project打開MSVS

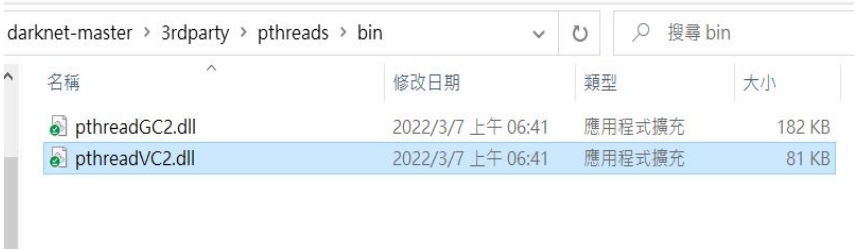


選擇 Release以及x64建置專案

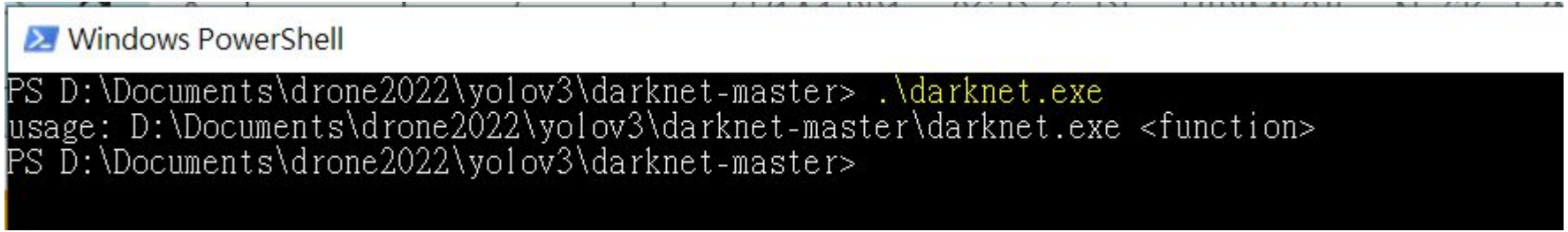


建置成功在darknet/Release/下會出現 darknet.exe
將Release 內的檔案複製貼到上一層的 /darknet 資料夾方便使用,
另外將 /3rdparty/pthreads/bin/ 當中的pthreadVC2.dll也貼到同一個位置



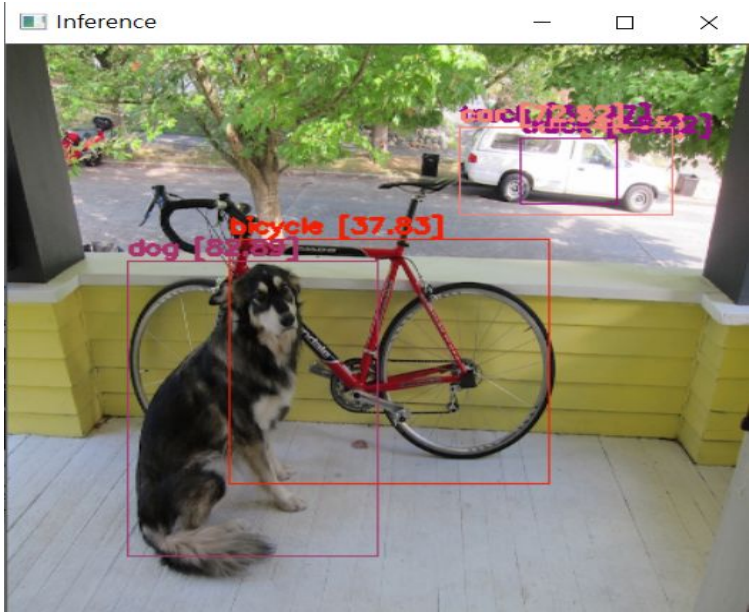


安裝成功後執行 ./darknet.exe會輸出function提示



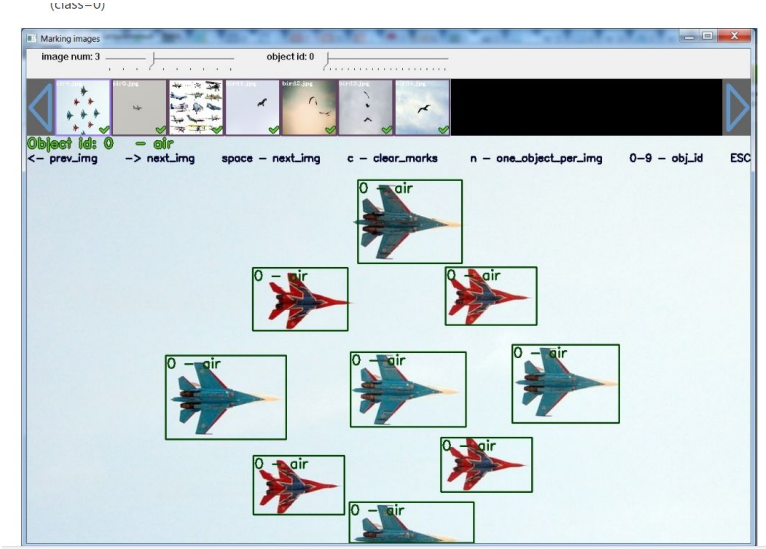
可以下載 [yolov3-tiny.weight] 嘗試執行

> python .\darknet_images.py --weights ./yolov3-tiny.weights --config_file ./cfg/yolov3-tiny.cfg --input ./data/dog.jpg



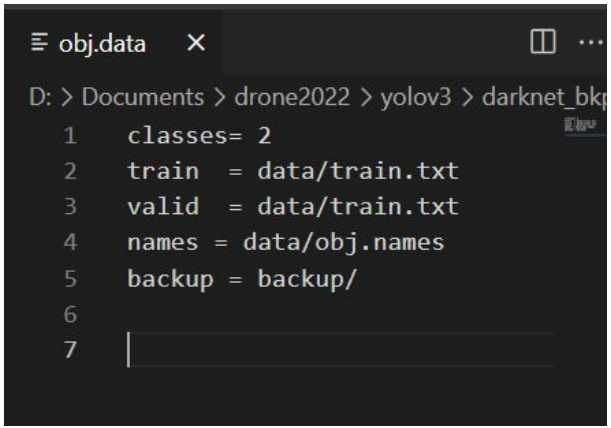
安裝 Yolo_mark

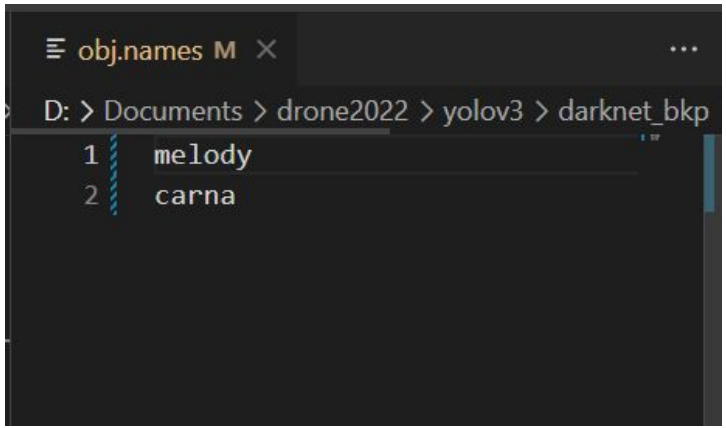
- Windows & Linux GUI for marking bounded boxes of objects in images for training Yolo v3 and v2 (v4也可以用)
- 依照github https://github.com/AlexeyAB/Yolo_mark 指引安裝



標註照片

1. 將下載的照片放進/data/img 下
2. 在/data建立兩個檔案obj.data obj.names
3. 執行yolo_mark.cmd



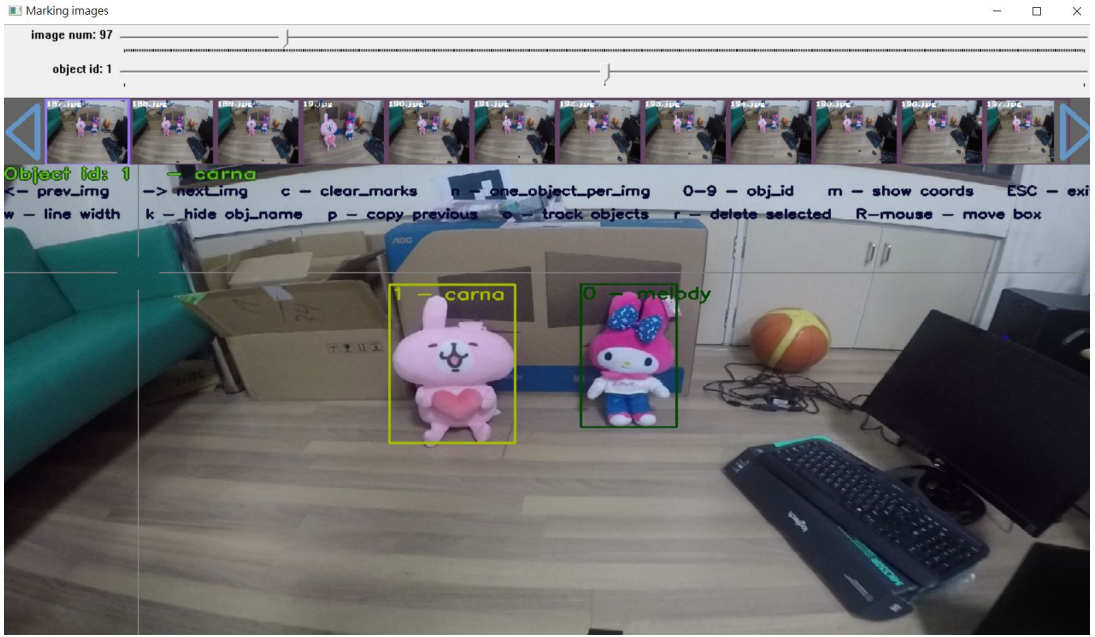


標註照片

基本操作

Keyboard Shortcuts

Shortcut	Description
→	Next image
←	Previous image
r	Delete selected box (mouse hovered)
c	Clear all marks on the current image
p	Copy previous mark
o	Track objects
ESC	Close application
n	One object per image
0-9	Object id
m	Show coords
w	Line width
k	Hide object name
h	Help



標註照片

Dataset共有三個資料夾

1. 只有卡娜赫拉:469張
2. 只有美樂蒂:481張
3. 兩個同時出現:589張

- 不需要全部都標
- 兩個種類的數量要差不多
- 怎麼標要講好(ex.框臉or全身)

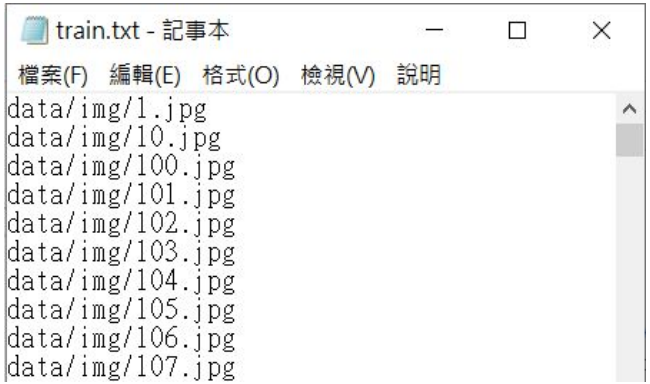
標註照片

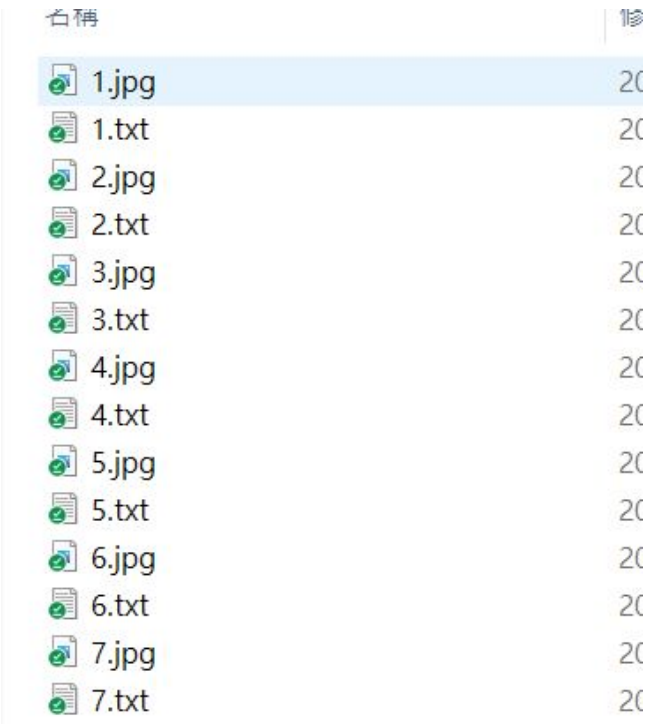
標註完成後每張照片會產生一個對應的.txt

內容為一行一個物件，格式如下

```
<object-class> <x_center> <y_center> <width> <height>
```

另外會產生一個train.txt存了從darknet到所有訓練照片的相對路徑





訓練model

/cfg　　放.cfg檔案，為YOLO的架構
/weights 別人訓練好的權重，需下載
/data　　放照片以及obj.names, obj.data, train.txt
/backup　訓練好的weights會存在這邊

cfg檔案內一些可以設定的參數：

- learning rate
- max_batches = classes*2000，且不小於6000或是訓練照片數量
- steps=0.8/0.9*max_batches

```
17
18  learning_rate=0.001
19  burn_in=1000
20  max_batches = 6000
21  policy=steps
22  steps=4800,5400
23  scales=.1,.1
24
```

訓練model

- 複製一個你要用的架構並改變名稱
  - ex. 複製一個yolov3-tiny.cfg 把名稱改成 yolov3-tiny-lab9.cfg
- 修改yolov3-tiny-lab9.cfg內的網路架構(不可以改到原本的)
  - classes=2
  - 每一層yolo前的filters=(classes + 5)x3

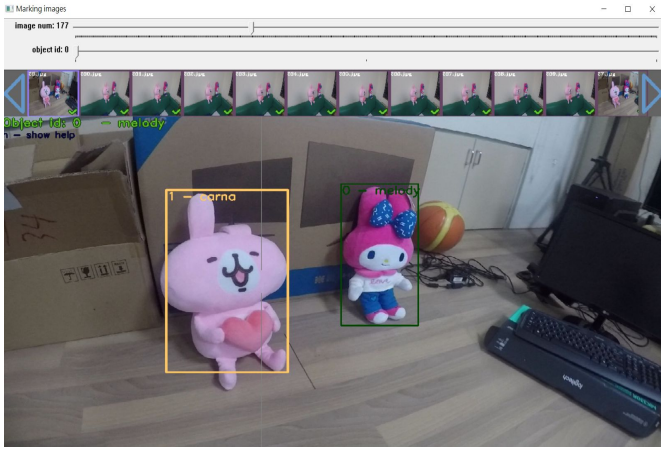
```
123  [convolutional]
124  size=1
125  stride=1
126  pad=1
127  filters=21
128  activation=linear
```

```
132  [yolo]
133  mask = 3,4,5
134  anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
135  classes=2
136  num=6
137  jitter=.3
```

```
167  [convolutional]
168  size=1
169  stride=1
170  pad=1
171  filters=21
172  activation=linear
```

```
174  [yolo]
175  mask = 0,1,2
176  anchors = 10,14,  23,27,  37,58,  81,82,  135,169,  344,319
177  classes=2
```

訓練model - 調整訓練



- 框的時候盡量不要框進太多背景
- 選擇框臉或頭或身體
  - 框臉跟框身體的不能一起訓練 , 要統一
- 訓練時
  - 先學兩隻一起出現的 , 再訓練單一隻出現的
  - 全部放在一起訓練
  - 先訓練單一隻出現的 , 再訓練兩隻出現的
  - 觀看結果 , 假如美樂蒂效果較差 , 加強訓練美樂蒂的資料集
- 訓練時的learning rate
  - loss高時設大　ex. 1e-3
  - loss接近1時設小　ex. 1e-5 ~ 1e-7
- 一定要用pretrained weight, 不然效果很差
- 嘗試不同的.cfg檔及.weights檔

觀看結果

資料夾內有 darknet_video.py可讀入影像並顯示結果

> python ./darknet_video.py --config_file <path to .cfg> --weights <path to weights> --data_file <path to .data> --input <path to test video>