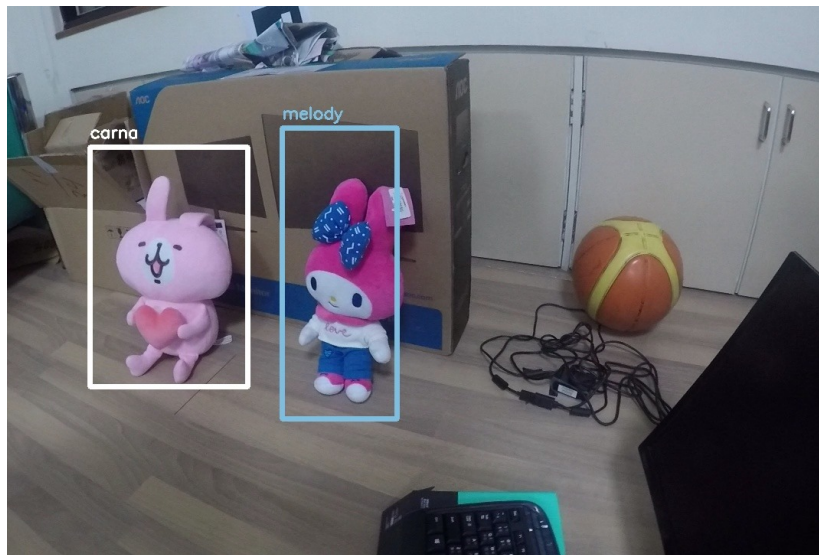


Lab09

<https://drive.google.com/drive/folders/1flmg337v25uA82k801BBdujrsPmHh4PT?usp=sharing>

步驟

1. 安裝tensorflow1.X版本
2. 安裝darkflow
3. 安裝Labellmg
4. 用Labellmg標註照片
5. 用darkflow訓練model
6. 觀看結果



1. 安裝tensorflow1.X版本

- 安裝python 3.X
- 安裝tensorflow1.X版本(CPU or GPU)
- 安裝Microsoft Visual Studio 201X
- X看你們自己要裝哪種版本, 下一頁有搭配

1. 安裝tensorflow1.X版本

pip install tensorflow == 1.15.1
~~download~~

https://www.tensorflow.org/install/source_windows?fbclid=IwAR3qbCp5IfPG6svtetfIF0FpG2DvpdpfCLfksYCUhf8yX8f-67M_2_a66A

CPU

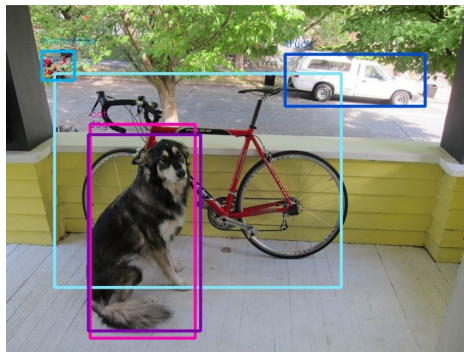
Version	Python version	Compiler	Build tools
tensorflow-2.2.0	3.5-3.8	MSVC 2019	Bazel 2.0.0
tensorflow-2.1.0	3.5-3.7	MSVC 2019	Bazel 0.27.1-0.29.1
tensorflow-2.0.0	3.5-3.7	MSVC 2017	Bazel 0.26.1
tensorflow-1.14.0	3.5-3.7	MSVC 2017	Bazel 0.24.1-0.25.2
tensorflow-1.13.0	3.5-3.7	MSVC 2015 update 3	Bazel 0.19.0-0.21.0
tensorflow-1.12.0	3.5-3.6	MSVC 2015 update 3	Bazel 0.15.0
tensorflow-1.11.0	3.5-3.6	MSVC 2015 update 3	Bazel 0.15.0
tensorflow-1.10.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.9.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.8.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.7.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.6.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.5.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.4.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.3.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.2.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.1.0	3.5	MSVC 2015 update 3	Cmake v3.6.3
tensorflow-1.0.0	3.5	MSVC 2015 update 3	Cmake v3.6.3

GPU

Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow_gpu-2.2.0	3.5-3.8	MSVC 2019	Bazel 2.0.0	7.4	10.1
tensorflow_gpu-2.1.0	3.5-3.7	MSVC 2019	Bazel 0.27.1-0.29.1	7.4	10.1
tensorflow_gpu-2.0.0	3.5-3.7	MSVC 2017	Bazel 0.26.1	7.4	10
tensorflow_gpu-1.14.0	3.5-3.7	MSVC 2017	Bazel 0.24.1-0.25.2	7.4	10
tensorflow_gpu-1.13.0	3.5-3.7	MSVC 2015 update 3	Bazel 0.19.0-0.21.0	7.4	10
tensorflow_gpu-1.12.0	3.5-3.6	MSVC 2015 update 3	Bazel 0.15.0	7	9
tensorflow_gpu-1.11.0	3.5-3.6	MSVC 2015 update 3	Bazel 0.15.0	7	9
tensorflow_gpu-1.10.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.9.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.8.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.7.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.6.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.5.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.4.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	6	8
tensorflow_gpu-1.3.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	6	8
tensorflow_gpu-1.2.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	5.1	8
tensorflow_gpu-1.1.0	3.5	MSVC 2015 update 3	Cmake v3.6.3	5.1	8
tensorflow_gpu-1.0.0	3.5	MSVC 2015 update 3	Cmake v3.6.3	5.1	8

2. 安裝darkflow

- 下載 <https://github.com/thtrieu/darkflow>
- 設定darkflow, 執行指令(三種方法選一個執行)
 - `python3 setup.py build_ext --inplace` *py -3.7 setup.py build-ext --inplace,*
 - `pip install -e .` *py -3.7 -m pip install -e,*
 - `pip install .`
- 下載 <https://pjreddie.com/media/files/yolov2.weights> 並丟到bin資料夾裡
- 試試看是否安裝完成
 - `python flow --model cfg/yolo.cfg --load bin/yolov2.weights --imgdir sample_img/`
 - 結果圖會存到sample_img/out資料夾裡



3. 安裝LabelImg

下載 <https://github.com/tzutalin/labelImg> 建議用python 3 + Qt5

pip install pyqt5

macOS

Python 2 + Qt4

```
brew install qt qt4
brew install libxml2
make qt4py2
python labelImg.py
python labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

Python 3 + Qt5 (Recommended)

```
brew install qt # Install qt-5.x.x by Homebrew
brew install libxml2

or using pip

pip3 install pyqt5 lxml # Install qt and lxml by pip

make qt5py3
python3 labelImg.py
python3 labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

Windows

Install Python, PyQt5 and install lxml.

→ 上一步 3.9
windows
amd

Open cmd and go to the labelimg directory

```
pyrcc5 -o lib/resources.py resources.qrc
For pyqt5, pyrcc5 -o libs/resources.py resources.qrc

python labelImg.py
python labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

Windows + Anaconda

Download and install Anaconda (Python 3+)

Open the Anaconda Prompt and go to the labelimg directory

```
conda install pyqt=5
pyrcc5 -o libs/resources.py resources.qrc
python labelImg.py
python labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```

4. 用LabelImg標註照片



4. 用LabelImg標註照片

Hotkeys

Ctrl + u	Load all of the images from a directory
Ctrl + r	Change the default annotation target dir
Ctrl + s	Save 注意:每張照片都要ctrl + s
Ctrl + d	Copy the current label and rect box
Space	Flag the current image as verified
w	Create a rect box
d	Next image
a	Previous image
del	Delete the selected rect box
Ctrl++	Zoom in
Ctrl--	Zoom out
↑→↓←	Keyboard arrows to move selected rect box

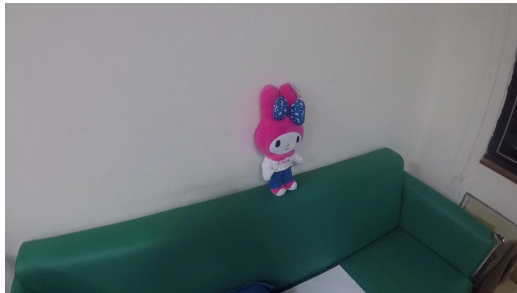
Dataset有三個資料夾

1. 只有卡那荷拉:469張
2. 只有美樂第: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

卡那荷拉?

美樂第?



5. 用darkflow訓練model

一種用CNN做出來的 object detection,

是計畫做物件偵測應用的人

可使用的開源技術

載下的 pretrained weights 要載進這裡面

- cfg/ #裡面放的.cfg檔案為YOLO的架構
- bin/ #別人用YOLO的架構train出來的weight -> 要去雲端下載
- label.txt #分類的種類名稱
- ckpt/ #訓練時每125個step存一次的weight會放這裡
- built_graph/ #.pb及.meta存的地方 (存完check point後要重新取, pb和, meta檔)
- darkflow/defaults.py內的setDefault #訓練時可以修改參數的檔案

```
def setDefaults(self):
    self.define('imgdir', './sample_img/', 'path to testing directory with images')
    self.define('binary', './bin/', 'path to .weights directory')
    self.define('config', './cfg/', 'path to .cfg directory')
    self.define('dataset', './pascal/VOCdevkit/IMG/', 'path to dataset directory')
    self.define('labels', 'labels.txt', 'path to labels file')
    self.define('backup', './ckpt/', 'path to backup folder')
    self.define('summary', '', 'path to TensorBoard summaries directory')
    self.define('annotation', './pascal/VOCdevkit/ANN/', 'path to annotation directory')
    self.define('threshold', -0.1, 'detection threshold')
    self.define('model', '', 'configuration of choice')
    self.define('trainer', 'rmsprop', 'training algorithm')
    self.define('momentum', 0.0, 'applicable for rmsprop and momentum optimizers')
    self.define('verbalise', True, 'say out loud while building graph')
    self.define('train', False, 'train the whole net')
    self.define('load', '', 'how to initialize the net? Either from .weights or a checkpoint, or')
    self.define('savepb', False, 'save net and weight to a .pb file')
    self.define('gpu', 0.0, 'how much gpu (from 0.0 to 1.0)')
    self.define('gpuName', '/gpu:0', 'GPU device name')
    self.define('lr', 1e-5, 'learning rate')
    self.define('keep', 20, 'Number of most recent training results to save')
    self.define('batch', 16, 'batch size')
    self.define('epoch', 1000, 'number of epoch')
```

5. 用darkflow訓練model

- 複製一個你要用的架構並改變名稱
 - ex. 複製一個tiny-yolo.cfg 把名稱改成 tiny-yolo-lab9.cfg
- 修改tiny-yolo-lab9.cfg內的網路架構(不可以改到原 tiny-yolo.cfg)
 - 最後一層convolution及classes數量
- 修改label.txt成你的label種類
 - ex.

carna
melody

100 #####
101
102 [convolutional]
103 batch_normalize=1
104 size=3
105 stride=1
106 pad=1
107 filters=1024
108 activation=leaky
109
110 [convolutional]
111 size=1
112 stride=1
113 pad=1
114 filters=35
115 activation=linear
116
117 [region]
118 anchors = 0.738768,0.874946, 2.42204,2.65704, 4.30971,7.04493, 10.246,4.59428, 12.6868,11.8741
119 bias_match=1
120 classes=2
121 coords=4
122 num=5
123 softmax=1
124 jitter=.2
125 rescore=1
126

這裡的filter要改成：
 $5 * (\text{幾種label} + 5)$
Ex：我有2種label
 $5 * (2 + 5) = 35$

有幾種label要標

卷积层滤波器数量 = 2类

5. 用darkflow訓練model - 開始訓練

- 參考darkflow/defaults.py內的setDefault給指令
- 從別人訓練好的tiny_yolo.weight(pretrained weight)訓練
 - python flow --model `cfg/tiny-yolo-lab9.cfg` `--load bin/tiny_yolo.weights`
--train --annotation **label檔的路徑** --dataset **圖檔的路徑**
- 每125個step存一次checkpoint, 先停止目前訓練 更改learning rate後, 接續上一次的繼續train
 - python flow --model `cfg/tiny-yolo-lab9.cfg` --load `-1` --train --annotation **label檔的路徑** --dataset **圖檔的路徑** --lr 1e-5
- 利用最近一次checkpoint的結果存成.pb檔案及.meta檔案
 - python flow --model `cfg/tiny-yolo-lab9.cfg` --load -1 --savepb

⇒ build-graph 是新增 .pb 和 .meta 檔案

5. 用darkflow訓練model - 開始訓練

- 利用存好的.pb檔及.meta檔預測結果
 - `flow --pbLoad built_graph/tiny-yolo-lab9.pb --metaLoad built_graph/tiny-yolo-lab9.meta --imgdir 待預測的所有圖 --threshold 信心指數多高才算該物體(0~1)`

5. 用darkflow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，假如美樂第效果較差，加強訓練美樂第的資料集
- 訓練時的learning rate
 - loss高時設大 ex. $1e-3$
 - loss接近1時設小 ex. $1e-5 \sim 1e-7$
- 一定要用pretrained weight, 不然效果很差
- 嘗試不同的.cfg檔及.weights檔
 - yolov2-tiny-voc.cfg + yolov2-tiny-voc.weights, 效果不錯

} 都

} 訓練到一半改lr

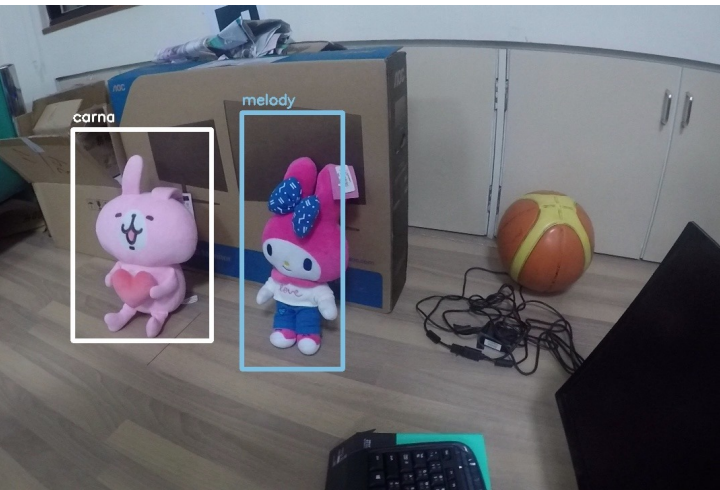
6. 觀看結果

也在雲端裡

- 下載 eval_yolo.py, 放到資料夾內
- 修改讀入的檔案

↓
修改檔案

```
threshold = 0.02
options = {"pbLoad": "result/0521_both_carna_melody_body_0.41/yolov2-tiny-voc-carna.pb",
           "threshold": threshold,
           "metaLoad": "result/0521_both_carna_melody_body_0.41/yolov2-tiny-voc-carna.meta"
          }
```



demo

- 修改eval_yolo.py並讀入demo.wmv
- 顯示預測結果

