



視覺辨識課程

謝坤達

JUMBOKH@GMAIL.COM

0953313123

大綱

1. 視覺辨識介紹
2. 影像標示
3. 在雲端上實作深度學習物件辨識模型
4. 簡易 YOLO3 系統辨識
5. 臉部辨識實作

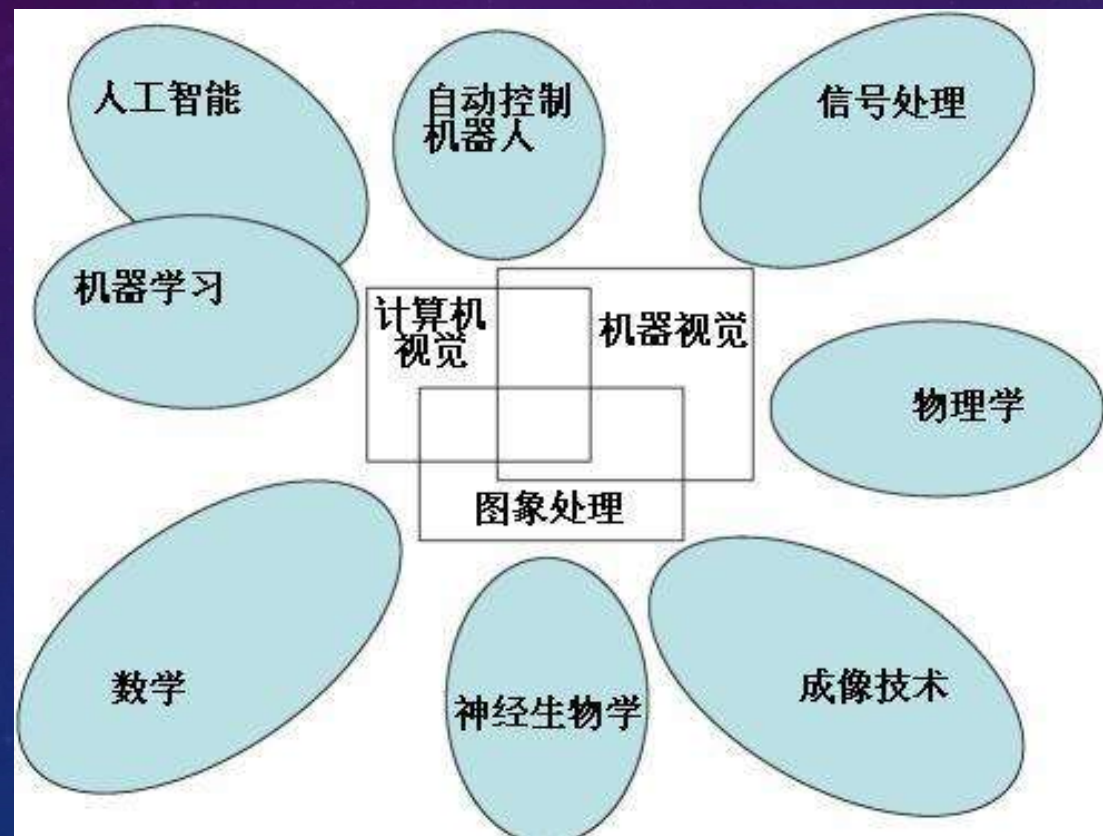
在雲端上實作深度學習物件辨識模型

- 何謂物件辨識
- 物件辨識演算法
- 訓練資料準備
- 實作

電腦視覺（COMPUTER VISION）

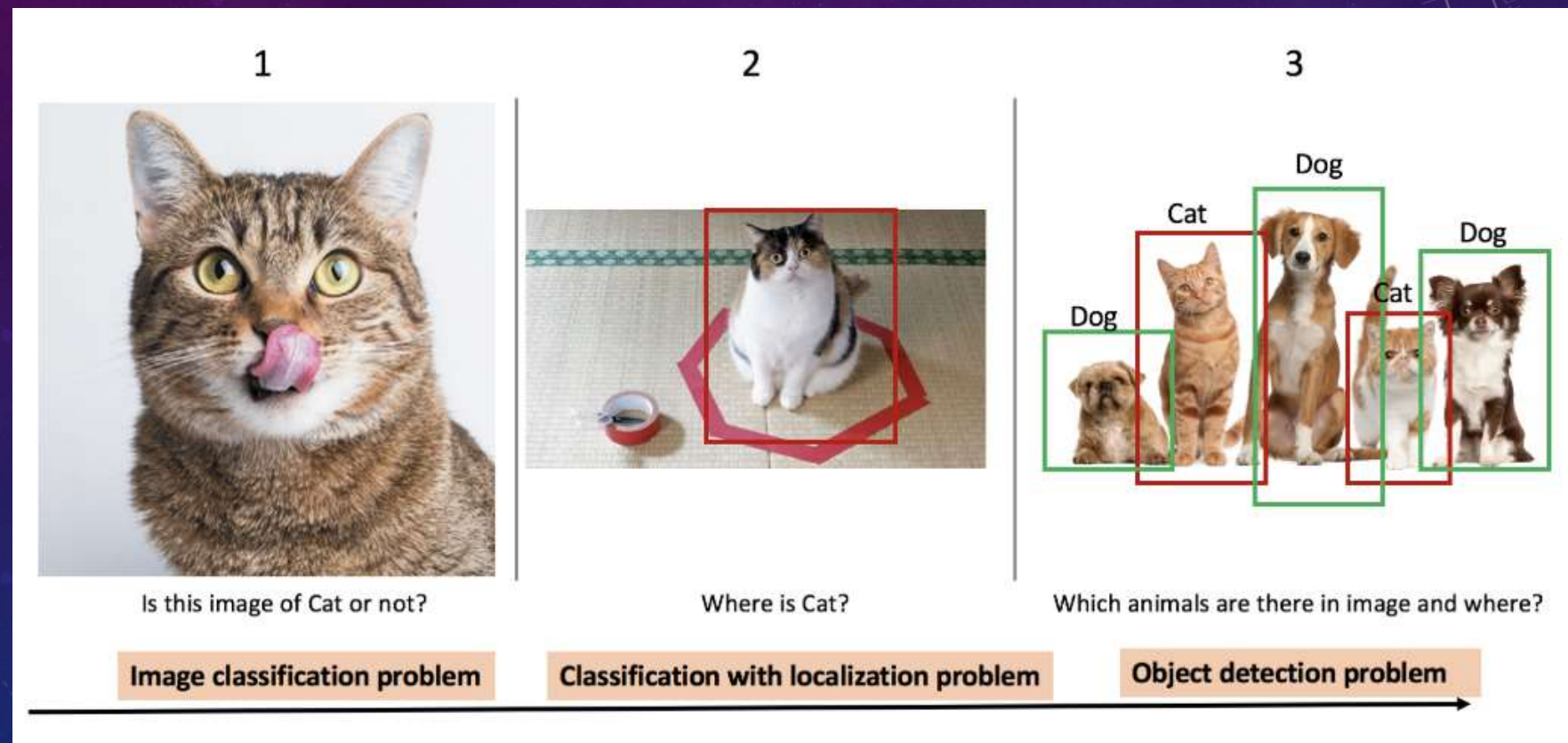
- 電腦視覺（**Computer vision**）是一門研究如何使機器「看」的科學，更進一步的說，就是指用攝影機和電腦代替人眼對目標進行辨識、跟蹤和測量等機器視覺，並進一步做圖像處理，用電腦處理成為更適合人眼觀察或傳送給儀器檢測的圖像

電腦視覺與其他領域的關係(維基百科)



WHAT IS OBJECT DETECTION?

Object detection compares the image classification and localizations



<https://towardsdatascience.com/evolution-of-object-detection-and-localization-algorithms-e241021d8bad>

WHAT IS OBJECT DETECTION?

Other Computer Vision Tasks

Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Classification + Localization



CAT

Single Object

Object Detection



DOG, DOG, CAT

Multiple Object

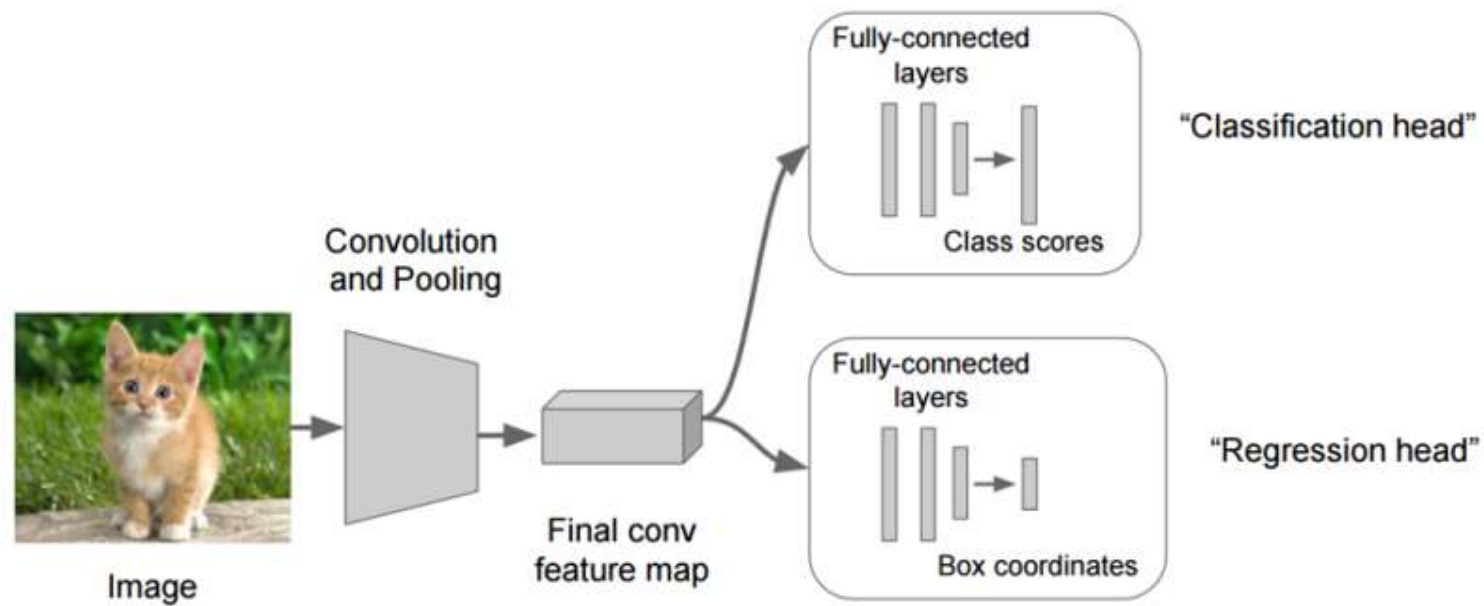
Instance Segmentation



DOG, DOG, CAT

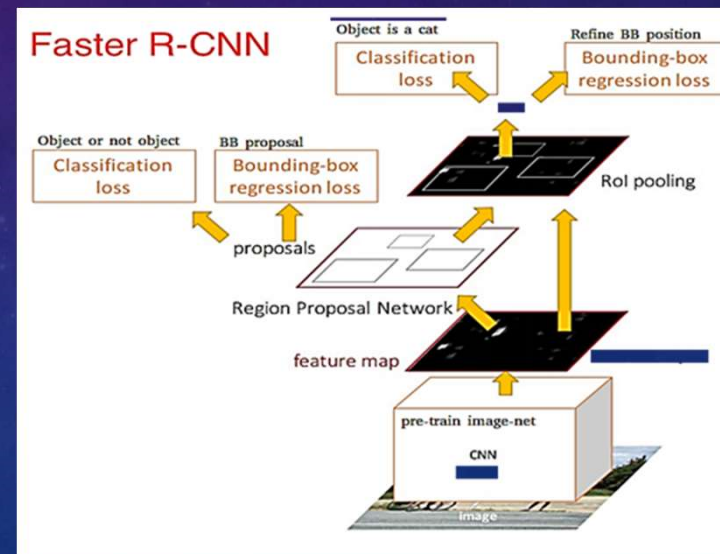
This image is CC0 public domain

OBJECT DETECTION ALGORITHMS



OBJECT DETECTION ALGORITHMS(CONT.)

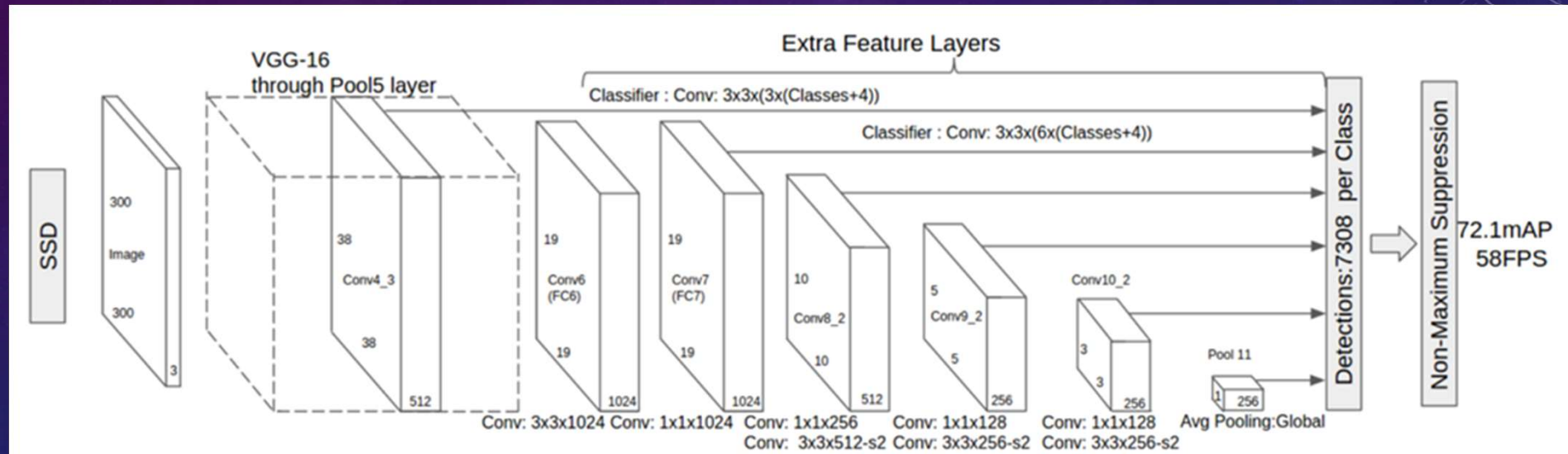
- R-CNN(Region with CNN) (See: <https://arxiv.org/abs/1311.2524>)
- Fast RCNN(See:<https://arxiv.org/abs/1504.08083>)
- Faster RCNN(See: <https://arxiv.org/abs/1506.01497>)
- R-FCN()
- SSD
- YOLOv1 to YOLOv3
- ReinaNet
- FPN



OBJECT DETECTION ALGORITHMS(CONT.)

- (參考: **a. 關於影像辨識，所有你應該知道的深度學習模型**
- **b. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms**)
- Deep Learning for Object Detection: A Comprehensive Review
<https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9>

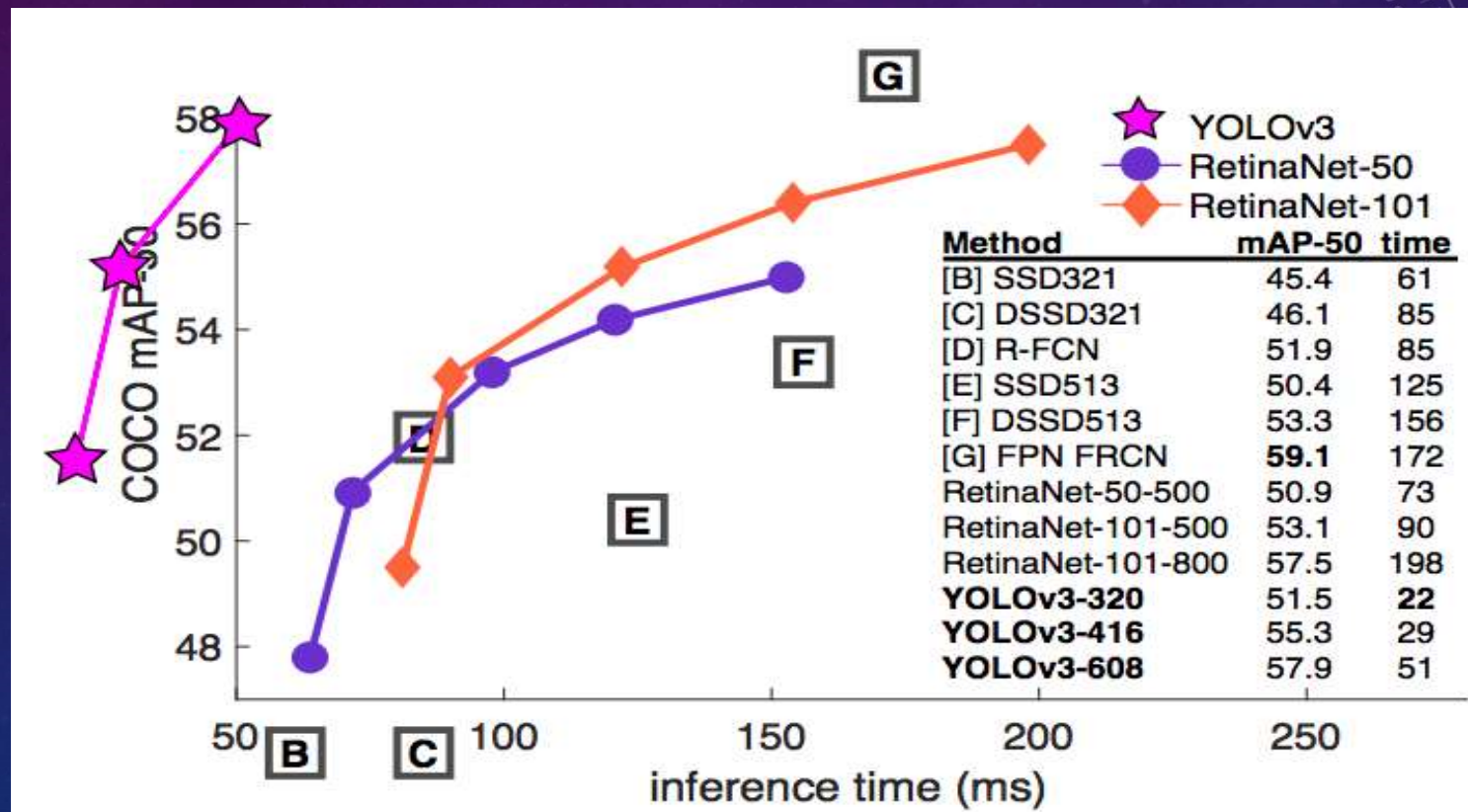
OBJECT DETECTION ALGORITHMS(CONT.)



OBJECT DETECTION ALGORITHMS(CONT.)

- one stage and two stages
 - (深度學習-什麼是one stage，什麼是two stage 物件偵測)
 - Optimizing the Trade-off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction PDF
- Comparison
- YOLOv3 is good as mAP and time
 - ✓What's new in YOLO v3?
- <https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9>
(<https://pjreddie.com/media/files/papers/YOLOv3.pdf>)

YOLO VS RETINANET PERFORMANCE ON COCO 50 BENCHMARK



ACCURACY AND SPEED TRADEOFF ON VOC 2007 (SOURCE: YOLOV2 PAPER)

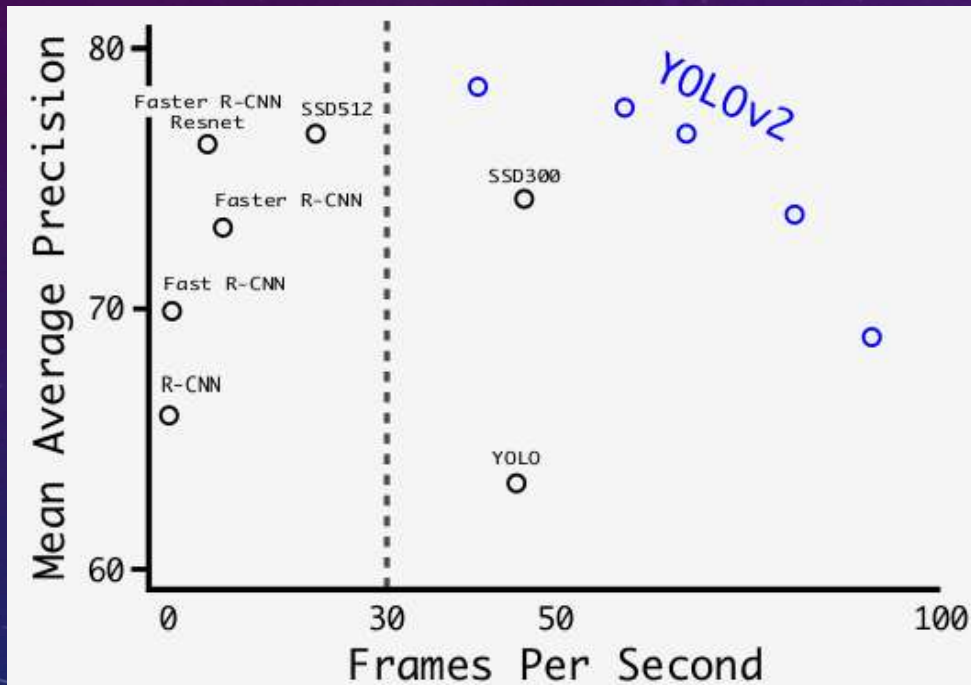


Figure 4: Accuracy and speed on VOC 2007.

✓ YOLO3: A Huge Improvement
<https://mc.ai/yolo3-a-huge-improvement/>

訓練資料準備

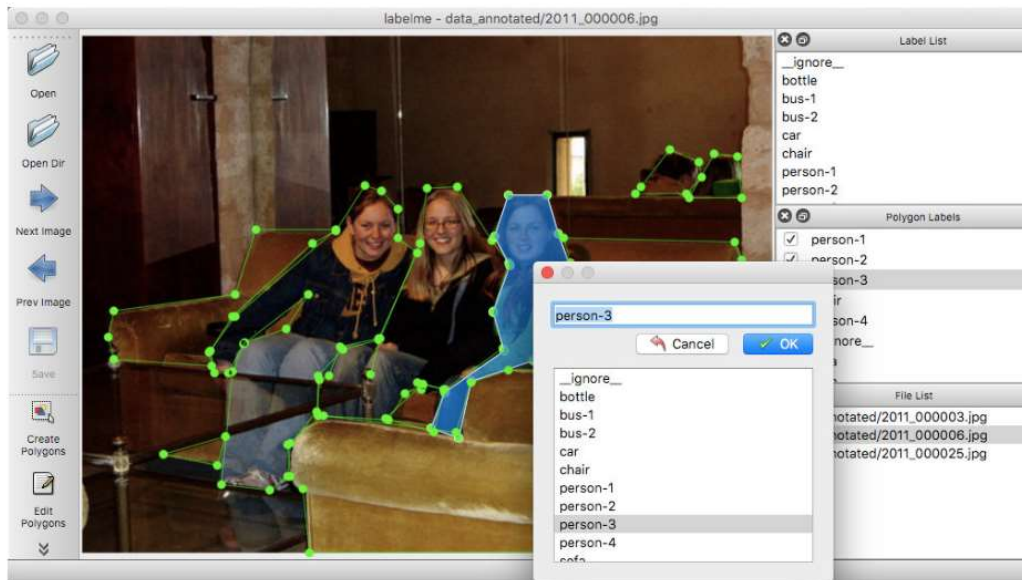
<https://github.com/wkentaro/labelme>

<https://tzutalin.github.io/labellmg/>

labelme

Image Polygonal Annotation with Python

`pypi` `v3.16.7` `python` `2.7` | `3.5` | `3.6` | `3.7` `CI` `passing` `docker build` `passing`



- # python2
- conda create --name=labelme python=2.7
- source activate labelme
- # conda install -c conda-forge pyside2
- conda install pyqt
- pip install labelme
- # if you'd like to use the latest version. run below:
- # pip install git+https://github.com/wkentaro/labelme.git
- # python3
- conda create --name=labelme python=3.6
- source activate labelme
- # conda install -c conda-forge pyside2
- # conda install pyqt
- # pip install pyqt5 # pyqt5 can be installed via pip on python3
- pip install labelme
- # or you can install everything by conda command
- # conda install labelme -c conda-forge

其他安裝文章

- <https://www.itread01.com/content/1544810780.html>

實作：訓練 XML 標示

實作程式碼

- **Colab example**: tensorflow-object-detection-training-colab.ipynb
https://colab.research.google.com/github/Tony607/object_detection_demo/blob/master/tensorflow_object_detection_training_colab.ipynb?hl=en
- Make a copy and test!

IMPLEMENTATIONS: TRAINING

```
from imageai.Detection.Custom import DetectionModelTrainer
trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="hololens")
trainer.setTrainConfig(object_names_array=["hololens"],
batch_size=8, num_experiments=5,
train_from_pretrained_model="pretrained-yolov3.h5")
trainer.setTrainConfig(object_names_array=["hololens"],
batch_size=8, num_experiments=5)
trainer.trainModel()
```


MODEL EVALUATION

IT SHOWS THE MAP OF OUR TRAINED MODELS.

```
from imageai.Detection.Custom import DetectionModelTrainer
```

```
trainer = DetectionModelTrainer()
```

```
trainer.setModelTypeAsYOLOv3()
```

```
trainer.setDataDirectory(data_directory="hololens")
```

```
trainer.evaluateModel(model_path="hololens/models",  
json_path="hololens/json/detection_config.json", iou_threshold=0.5,  
object_threshold=0.3, nms_threshold=0.5)
```

實作

- If I have many items?
- `Object_names_array=["Peanut", "Hazelnut"]`

EXTRAS

- ✓ Run SSD, Faster RCNN and FCN
- ✓ <https://medium.com/swlh/how-to-train-an-object-detection-model-easy-for-free-f388ff3663e>
- ✓ (Or <https://www.dlology.com/blog/how-to-train-an-object-detection-model-easy-for-free/>)

結論

- The impact of the object detection
- YOLO3 is a great framework so far
- Implementation is quite easy now
- Labeling job is quite laborious, we are working on some tricks

REFERENCE

- Reference:
- ImageAI (v2.1.4)
- <https://github.com/OlafenwaMoses/ImageAI>
- **Train Object Detection AI with 6 lines of code (Part_I)** <https://medium.com/deepquestai/train-object-detection-ai-with-6-lines-of-code-6d087063f6ff>
- **Object Detection with 10 lines of code (Part_II)**
- <https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606>
 - **Evolution of Object Detection and Localization Algorithms** <https://towardsdatascience.com/evolution-of-object-detection-and-localization-algorithms-e241021d8bad>
 - **Object Localization and Detection** https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/object_localization_and_detection.html
<https://medium.com/%E8%B3%87%E6%96%99%E9%9A%A8%E7%AD%86/machine-learning-103-d81ef2ad3597>