Keras_yolov3實作教學

2019/12/19

TA-侑學

Outline

- Introduction
- 實作
- 訓練自己的model

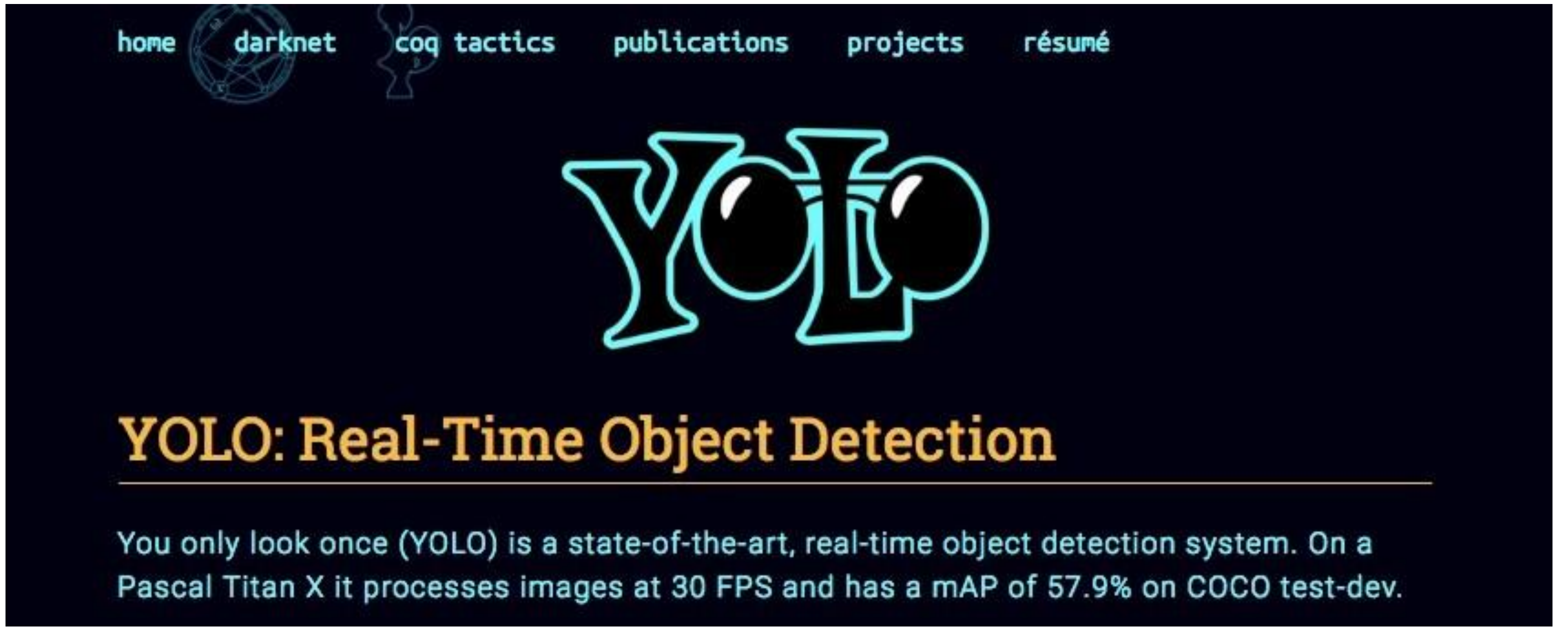
檔案連結-請先下載

- https://drive.google.com/file/d/1gViUGFXEVYmyQm6s5aiWp7VVhNGzl4tQ/view?usp=sharing

Introduction

- YOLO全名 - You only look once，顧名思義就是只要學習過一一次即可，代表學習的速度很快。

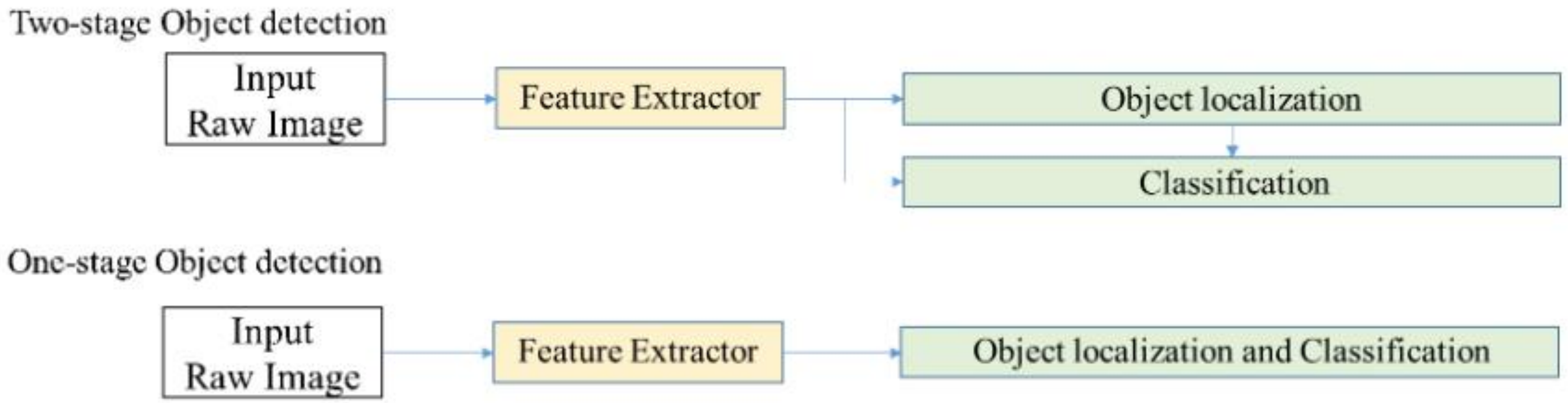
- YOLO是one stage的物件偵測方法，也就是只需要對圖片作一次CNN架構便能夠判斷圖形內的物體位置與類別。



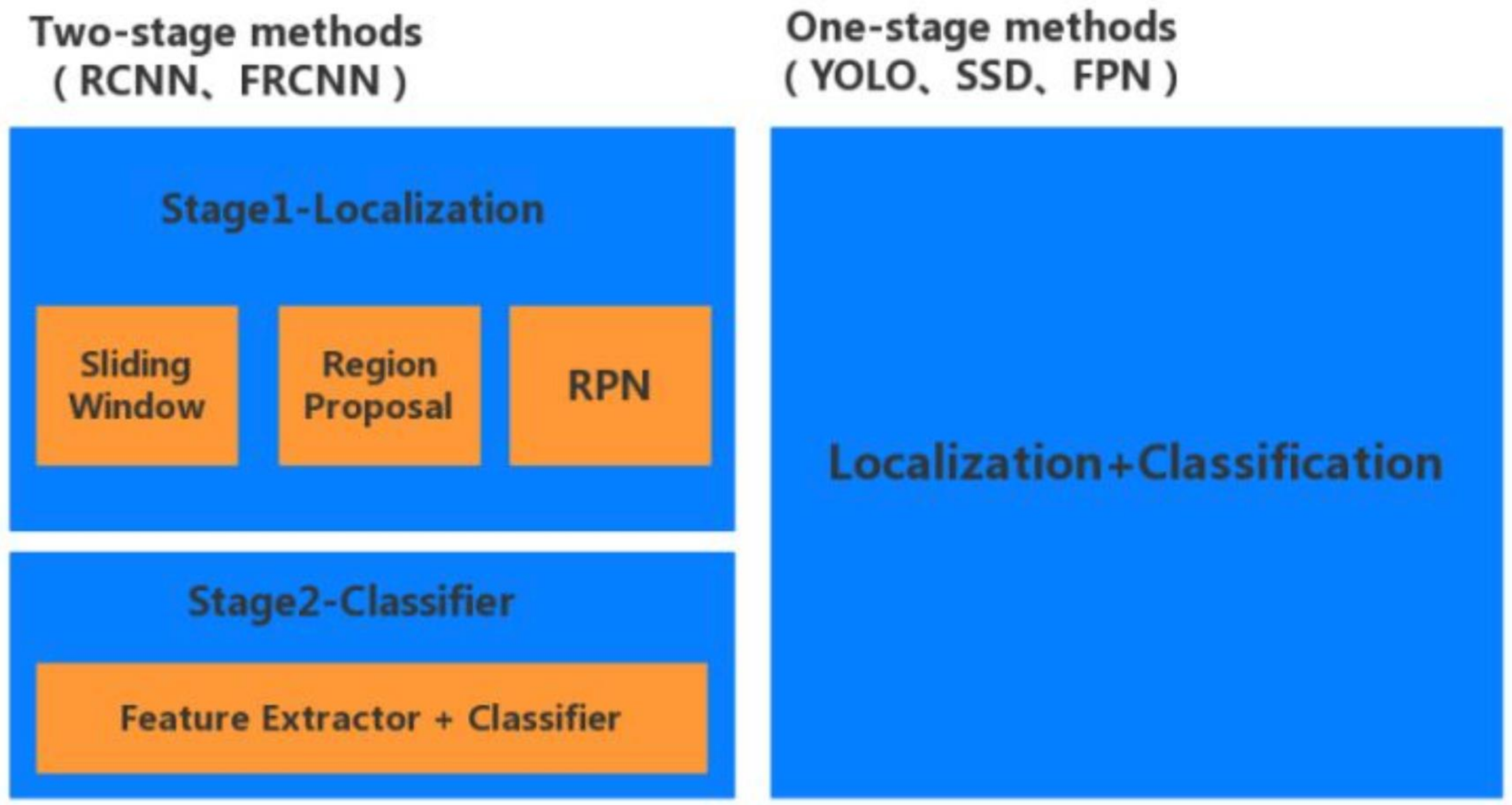
YOLO: Real-Time Object Detection

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev.

現今兩種Object detection方法



現今兩種Object detection方法



Yolo發展- YoloV1

- 1.從 RCNN、fast RCNN、faster RCNN、Yolo 的思路一路發展上來，Yolo 最大的特色是直接 end-to-end 做物件偵測，利用整張圖片作為神經網路的輸入，直接預測 bounding box 坐標位置、bounding box 含物體的 confidence 和物體所屬的類別。

- 2. YoloV1 計算快速，能夠達到 real-time 速度需求，缺點是對位置的預測不夠精確，且小物體預測效果較差。

Yolo發展 - YoloV2

- 1. YoloV2 針對 YoloV1 的缺點做了一些改進：
- 2. 引入 Faster RCNN 中的 anchor box，不再直接 mapping bounding box 的座標，而是預測相對於 anchor box 的參數，並使用 K-Means 求 anchor box 比例。
- 3. 去掉 fully connected layer，改成全部皆為 conv layer。
- 4. 每層加上 batch normalization，去掉 dropout。
- 5. 增加解析度：增加 ImageNet pre-train 的解析度，從 224×224 提升至 448×448。

YoloV3-邊界框預測

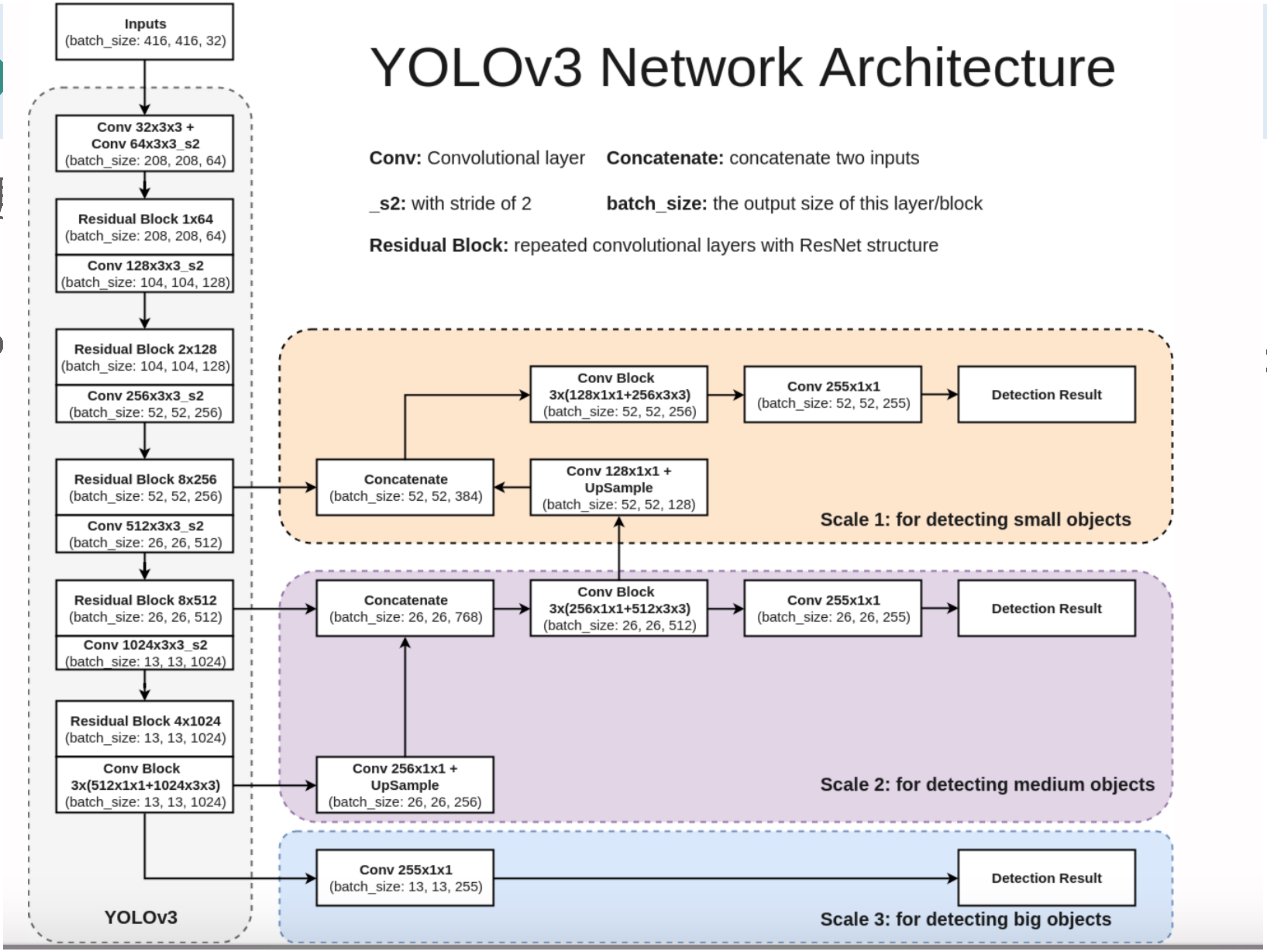
- 邊界框預測 - YOLOv3在邊界框的預測上，改成以logistic regression來預測邊界框包含物體的分數。YOLOv3只把與ground-truth重疊率最高的那個邊界框認作滿分1，不是被認作最好的那些邊界框，就不會對分類或是邊界框座標的損失有做貢獻。

- 分類預測 - YOLOv3改用independent logistic classifiers，並搭配binary cross-entropy來衡量分類上的損失。

Yo

- 1.
- 2. P

s)



YOLOv3 Network Architecture

Conv: Convolutional layer **Concatenate:** concatenate two inputs

_s2: with stride of 2 **batch_size:** the output size of this layer/block

Residual Block: repeated convolutional layers with ResNet structure

Inputs (batch_size: 416, 416, 32)
Conv 32x3x3 + Conv 64x3x3_s2 (batch_size: 208, 208, 64)
Residual Block 1x64 (batch_size: 208, 208, 64)
Conv 128x3x3_s2 (batch_size: 104, 104, 128)
Residual Block 2x128 (batch_size: 104, 104, 128)
Conv 256x3x3_s2 (batch_size: 52, 52, 256)
Residual Block 8x256 (batch_size: 52, 52, 256)
Conv 512x3x3_s2 (batch_size: 26, 26, 512)
Residual Block 8x512 (batch_size: 26, 26, 512)
Conv 1024x3x3_s2 (batch_size: 13, 13, 1024)
Residual Block 4x1024 (batch_size: 13, 13, 1024)
Conv Block 3x(512x1x1+1024x3x3) (batch_size: 13, 13, 1024)
YOLOv3
Conv Block 3x(128x1x1+256x3x3) (batch_size: 52, 52, 256)
Conv 255x1x1 (batch_size: 52, 52, 255)
Detection Result
Concatenate (batch_size: 52, 52, 384)
Conv 128x1x1 + UpSample (batch_size: 52, 52, 128)
Scale 1: for detecting small objects
Concatenate (batch_size: 26, 26, 768)
Conv Block 3x(256x1x1+512x3x3) (batch_size: 26, 26, 512)
Conv 255x1x1 (batch_size: 26, 26, 255)
Detection Result
Conv 256x1x1 + UpSample (batch_size: 26, 26, 256)
Scale 2: for detecting medium objects
Conv 255x1x1 (batch_size: 13, 13, 255)
Detection Result
Scale 3: for detecting big objects

Why use YOLO?

- 簡單的安裝步驟

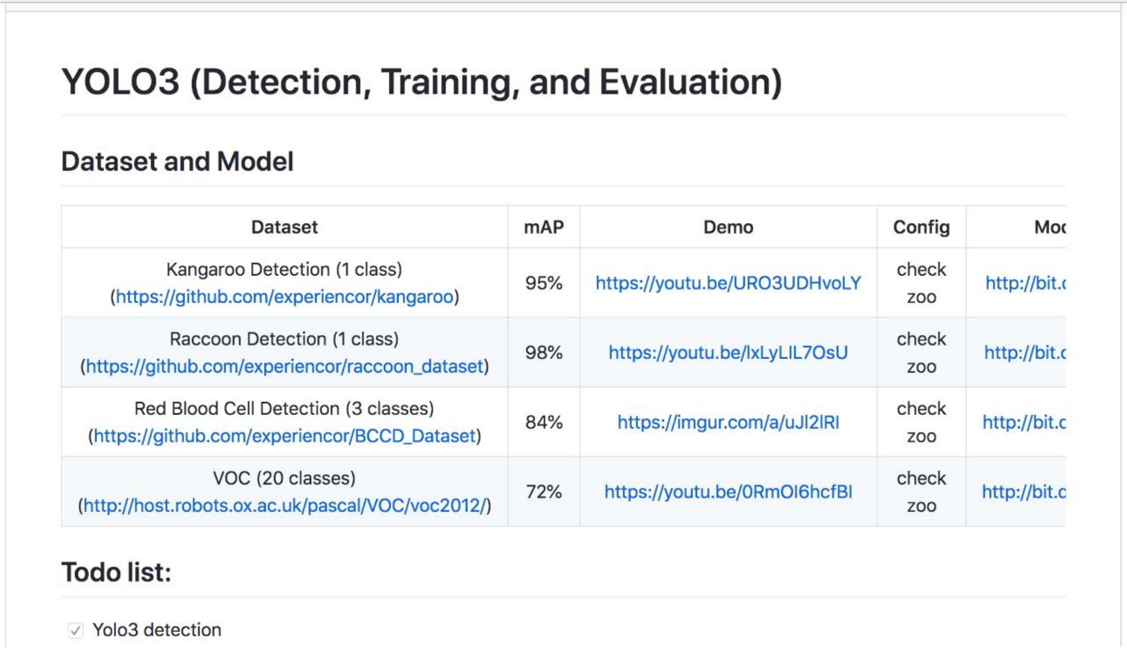
- 不需要調動太多的參數，即可有良好的訓練效果

實作

Keras-yolov3

- Google drive 連結：https://drive.google.com/file/d/1QYu-pUQ8DwIx7AKdW1k4-rifz8xouhOd/view?usp=sharing
- 此次實作的keras-yolov3版本為:https://github.com/experiencor/keras-yolo3

- 詳細運作可看此篇的說明



YOLO3 (Detection, Training, and Evaluation)

Dataset and Model

Dataset	mAP	Demo	Config	Moc
Kangaroo Detection (1 class) (https://github.com/experiencor/kangaroo)	95%	https://youtu.be/URO3UDHvoLY	check zoo	http://bit.c
Raccoon Detection (1 class) (https://github.com/experiencor/raccoon_dataset)	98%	https://youtu.be/lxLyLIL7OsU	check zoo	http://bit.c
Red Blood Cell Detection (3 classes) (https://github.com/experiencor/BCCD_Dataset)	84%	https://imgur.com/a/uJl2lRl	check zoo	http://bit.c
VOC (20 classes) (http://host.robots.ox.ac.uk/pascal/VOC/voc2012/)	72%	https://youtu.be/0RmOI6hcfBI	check zoo	http://bit.c

Todo list:

- [x] Yolo3 detection

資料夾內部

- 主要使用train.py, predict.py , train.ipynb , predict.ipynb
- 當訓練的時候如果怕會斷線可使用.py的方式執行，在hub上可使用.ipynb執行即可
- 畫框的部分則是在utils中的bbox.py檔案裡面

資料夾內部

- 可使用train.ipynb檔案進行操作，如果使用ipynb檔就不需要再另外執行cmd指令

- Train.ipynb裡包含了predict的程式(最後一段)，訓練完model後可直接執行，或者另外拉出來，就不用每次都重新執行程式。

- 主要會用到的train.py 以及predict.py ，畫框的部分則是在utils中的bbox.py檔案裡面

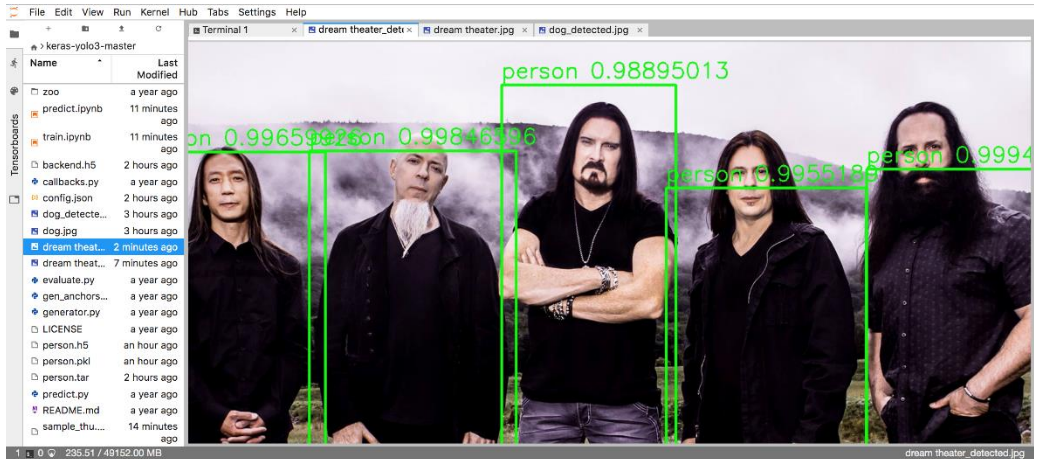
測試

- 可執行這行指令產出結果

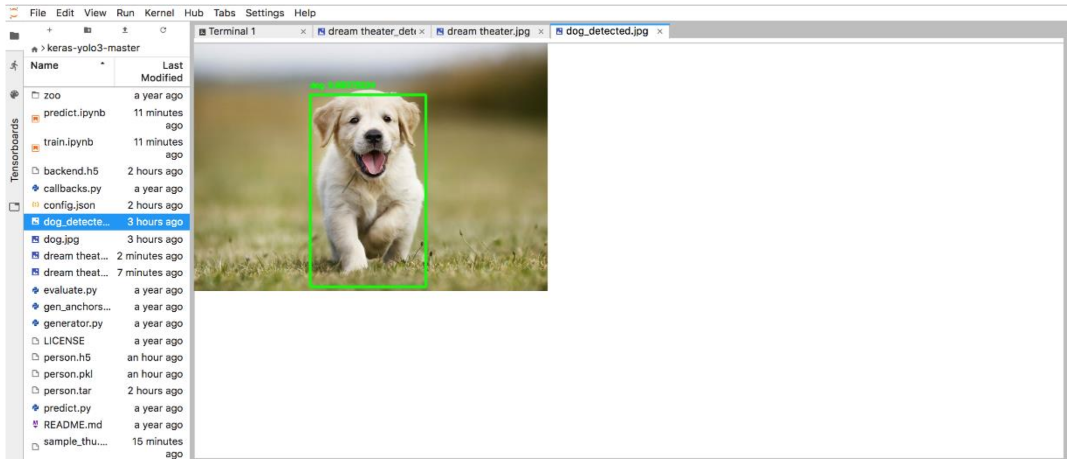
- python3 yolo3_one_file_to_detect_them_all.py -w yolo3.weights -i dog.jpg

- -w 代表載入權重的位置， -i 代表載入擋案的位置

- 執行完後會呈現detect 結果，可自行測試



Name	Last Modified
zoo	a year ago
predict.ipynb	11 minutes ago
train.ipynb	11 minutes ago
backend.h5	2 hours ago
callbacks.py	a year ago
config.json	2 hours ago
dog_detecte...	3 hours ago
dog.jpg	3 hours ago
dream theat...	2 minutes ago
dream theat...	7 minutes ago
evaluate.py	a year ago
gen_anchors...	a year ago
generator.py	a year ago
LICENSE	a year ago
person.h5	an hour ago
person.pkl	an hour ago
person.tar	2 hours ago
predict.py	a year ago
README.md	a year ago
sample_thu...	14 minutes ago

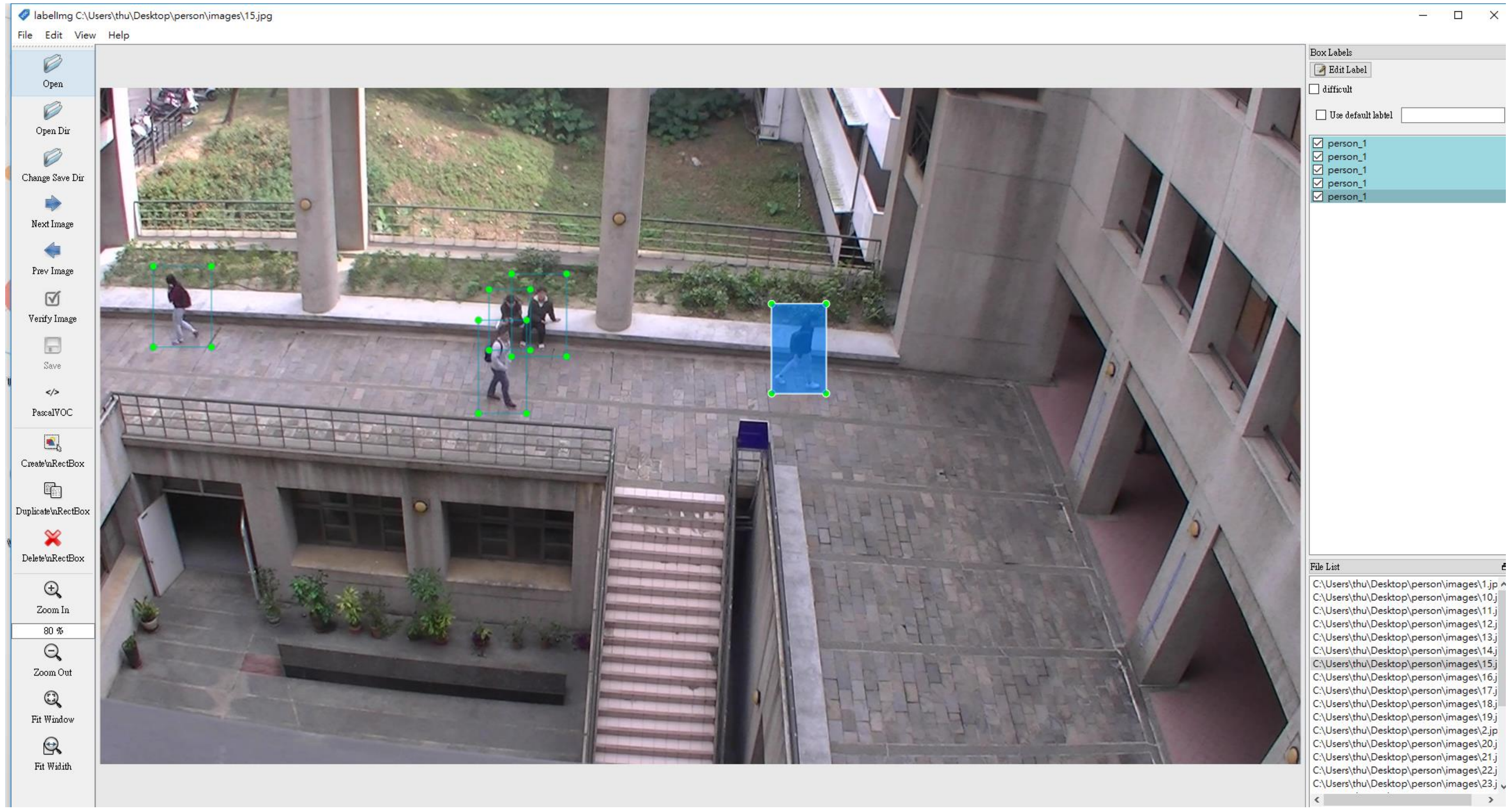


Name	Last Modified
zoo	a year ago
predict.ipynb	11 minutes ago
train.ipynb	11 minutes ago
backend.h5	2 hours ago
callbacks.py	a year ago
config.json	2 hours ago
dog_detecte...	3 hours ago
dog.jpg	3 hours ago
dream theat...	2 minutes ago
dream theat...	7 minutes ago
evaluate.py	a year ago
gen_anchors...	a year ago
generator.py	a year ago
LICENSE	a year ago
person.h5	an hour ago
person.pkl	an hour ago
person.tar	2 hours ago
predict.py	a year ago
README.md	a year ago
sample_thu...	15 minutes ago

訓練自己的model

Keras-yolov3

- 要進行訓練之前需要準備 image 跟 label ，也就是上上禮拜學的標記。
- 將label儲存成Pascal VOC格式之後再存儲的時候比較不會有問題
- 如果label還不熟悉得請參考之前的資料標記講義


labelImg C:\Users\thu\Desktop\person\images\15.jpg
File
Edit
View
Help
Open
Open Dir
Change Save Dir
Next Image
Prev Image
Verify Image
Save
PascalVOC
Create\nRectBox
Duplicate\nRectBox
Delete\nRectBox
Zoom In
80 %
Zoom Out
Fit Window
Fit Width
Box Labels
Edit Label
difficult
Use default label
person_1
person_1
person_1
person_1
person_1
File List
C:\Users\thu\Desktop\person\images\1.jp
C:\Users\thu\Desktop\person\images\10.j
C:\Users\thu\Desktop\person\images\11.j
C:\Users\thu\Desktop\person\images\12.j
C:\Users\thu\Desktop\person\images\13.j
C:\Users\thu\Desktop\person\images\14.j
C:\Users\thu\Desktop\person\images\15.j
C:\Users\thu\Desktop\person\images\16.j
C:\Users\thu\Desktop\person\images\17.j
C:\Users\thu\Desktop\person\images\18.j
C:\Users\thu\Desktop\person\images\19.j
C:\Users\thu\Desktop\person\images\2.jp
C:\Users\thu\Desktop\person\images\20.j
C:\Users\thu\Desktop\person\images\21.j
C:\Users\thu\Desktop\person\images\22.j
C:\Users\thu\Desktop\person\images\23.j

Keras-yolov3

- 在做資料標記時，要特別掌握每個框的大小，不能差距過大，否則很容易誤判。
- 要注意再丟進去訓練之前一定要檢查每一張相片都要有label到，否則則會噴錯。
- 至於要標記多少張則是需要case by case
- 在做訓練的時候，執行train的時候就不能執行其他程式了，會造成OOM(Out Of Memory)。

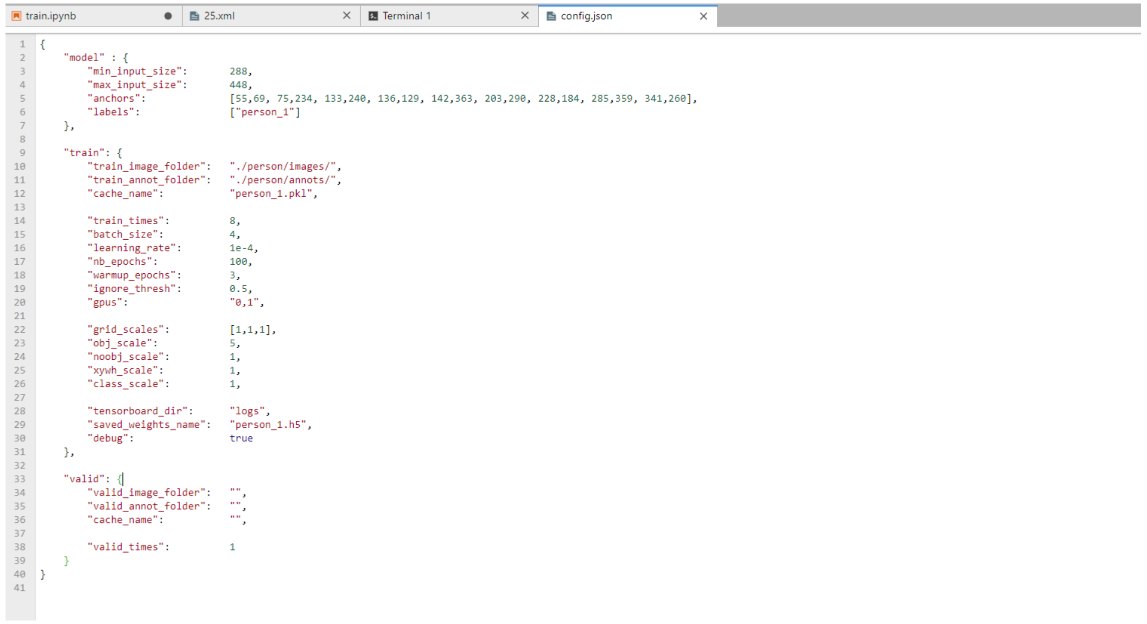
Keras-yolov3

- 常用需要更動的參數

Names	參數說明
labels	照片 裡有label到的名子
anchors	錨點，可用預設或gen_anchor.py
train_image_folder	存放訓練資料照片 的路徑
train_annot_folder	存放訓練資料labels的路徑
cache_name	pickle 檔，會儲存訓練的紀錄(資料更動後需要刪除，否則則會讀取前一次的訓練資料)
saved_weights_name	你所要存放的權重名稱
nb_epochs	訓練的圈數

Keras-yolov3

- 再準備好資料之後可新增一資料夾(ex:person)，然後打開config.json去修改你要讀取的資料。
- 容易OOM的話可將batch size調小
- 如果有複數Labels可用,隔開(ex:["person_1”，” apple”])

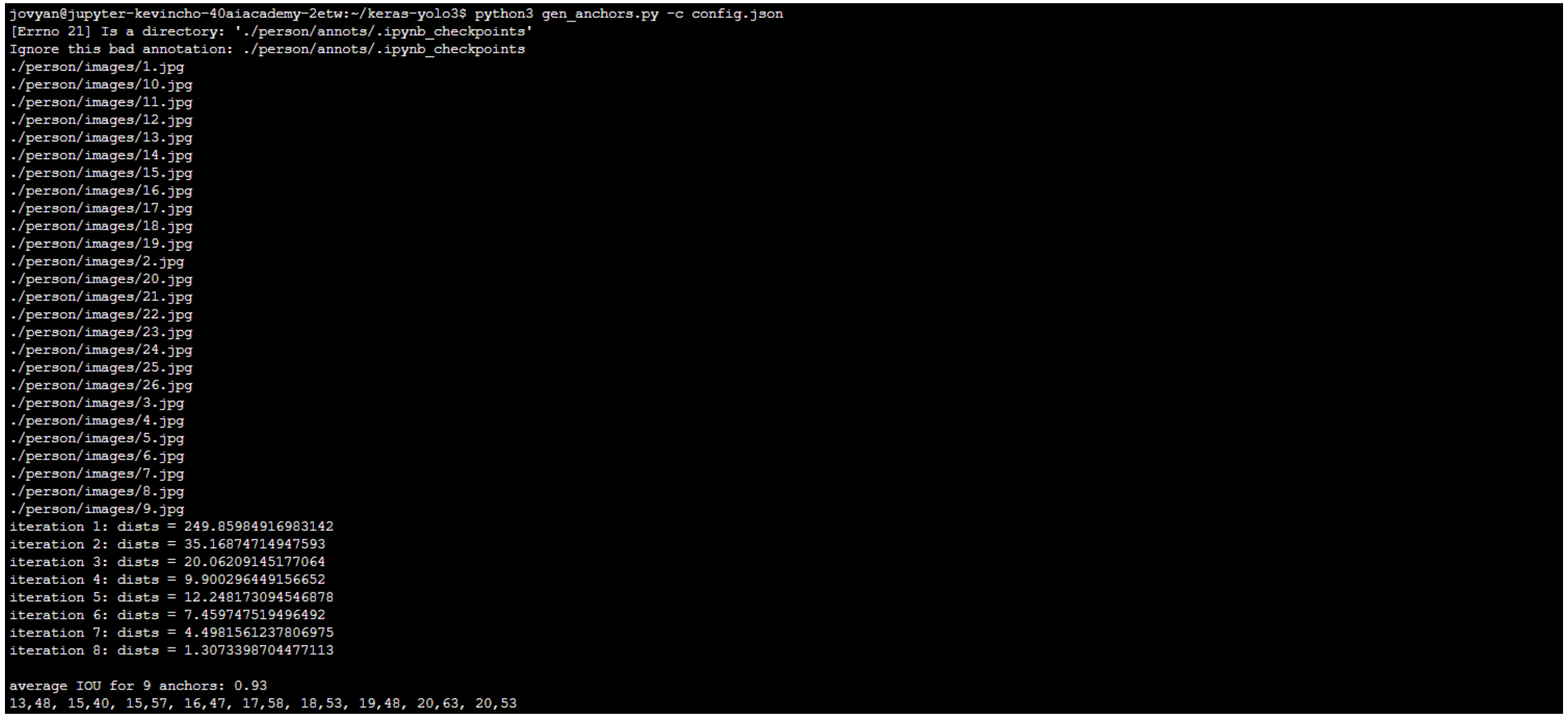


```
{
    "model" : {
        "min_input_size":       288,
        "max_input_size":       448,
        "anchors":              [55,69, 75,234, 133,240, 136,129, 142,363, 203,290, 228,184, 285,359, 341,260],
        "labels":               ["person_1"]
    },

    "train": {
        "train_image_folder":   "./person/images/",
        "train_annot_folder":   "./person/annots/",
        "cache_name":           "person_1.pkl",

        "train_times":          8,
        "batch_size":           4,
        "learning_rate":        1e-4,
        "nb_epochs":            100,
        "warmup_epochs":        3,
        "ignore_thresh":        0.5,
        "gpus":                 "0,1",

        "grid_scales":          [1,1,1],
        "obj_scale":            5,
        "noobj_scale":          1,
        "xywh_scale":           1,
        "class_scale":          1,

        "tensorboard_dir":      "logs",
        "saved_weights_name":   "person_1.h5",
        "debug":                true
    },

    "valid": {
        "valid_image_folder":   "",
        "valid_annot_folder":   "",
        "cache_name":           "",

        "valid_times":          1
    }
}
```

Python檔執行

Keras-yolov3

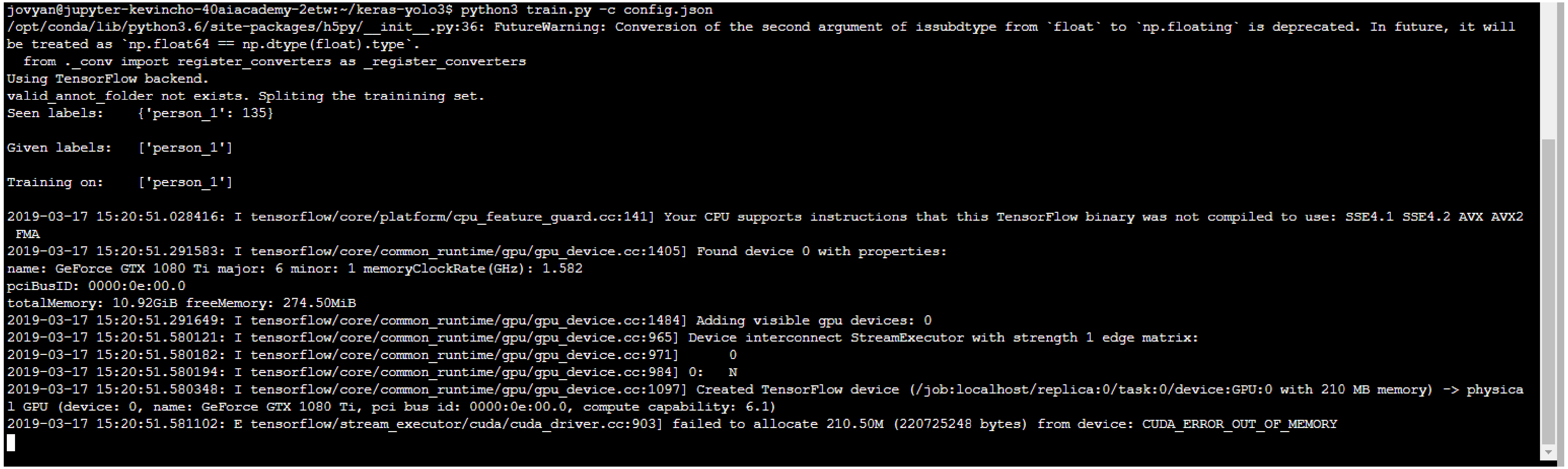
- 如果不知自己的anchors要設多少可使用(也可用預設即可)
- pyhton3 gen_anchors.py –c config.json 獲取推薦值



```
jovyan@jupyter-kevincho-40aiacademy-2etw:~/keras-yolo3$ python3 gen_anchors.py -c config.json
[Errno 21] Is a directory: './person/annots/.ipynb_checkpoints'
Ignore this bad annotation: ./person/annots/.ipynb_checkpoints
./person/images/1.jpg
./person/images/10.jpg
./person/images/11.jpg
./person/images/12.jpg
./person/images/13.jpg
./person/images/14.jpg
./person/images/15.jpg
./person/images/16.jpg
./person/images/17.jpg
./person/images/18.jpg
./person/images/19.jpg
./person/images/2.jpg
./person/images/20.jpg
./person/images/21.jpg
./person/images/22.jpg
./person/images/23.jpg
./person/images/24.jpg
./person/images/25.jpg
./person/images/26.jpg
./person/images/3.jpg
./person/images/4.jpg
./person/images/5.jpg
./person/images/6.jpg
./person/images/7.jpg
./person/images/8.jpg
./person/images/9.jpg
iteration 1: dists = 249.85984916983142
iteration 2: dists = 35.16874714947593
iteration 3: dists = 20.06209145177064
iteration 4: dists = 9.900296449156652
iteration 5: dists = 12.248173094546878
iteration 6: dists = 7.459747519496492
iteration 7: dists = 4.4981561237806975
iteration 8: dists = 1.3073398704477113

average IOU for 9 anchors: 0.93
13,48, 15,40, 15,57, 16,47, 17,58, 18,53, 19,48, 20,63, 20,53
```

Keras-yolov3

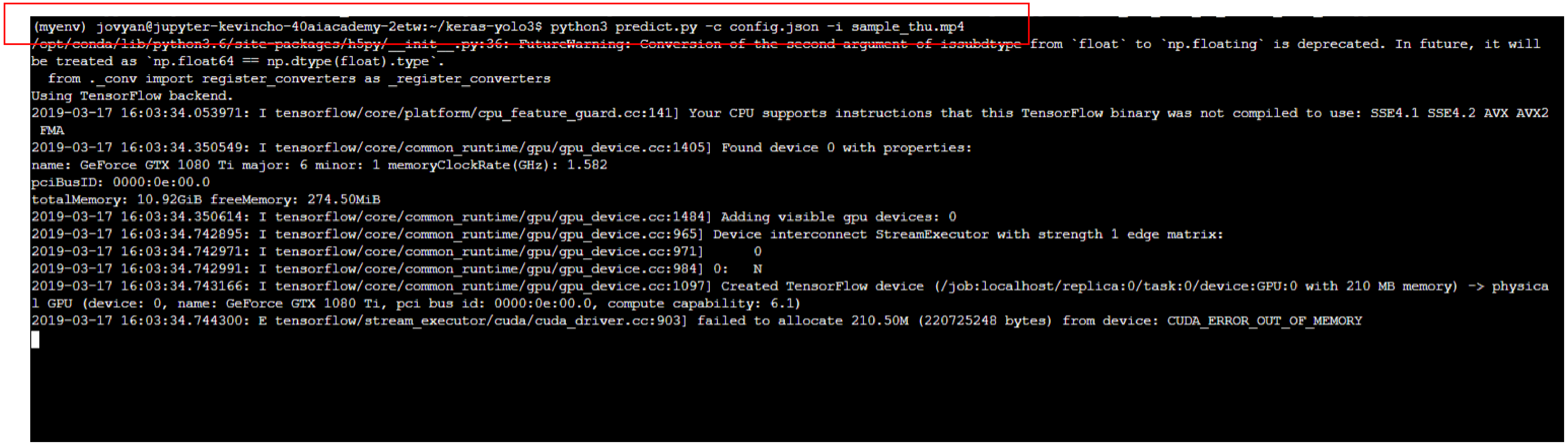
- 將以上參數設定完畢後便可開始訓練
- pyhton3 train.py –c config.json



```
jovyan@jupyter-kevincho-40aiacademy-2etw:~/keras-yolo3$ python3 train.py -c config.json
/opt/conda/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will
be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
valid_annot_folder not exists. Spliting the trainining set.
Seen labels:    {'person_1': 135}

Given labels:    ['person_1']

Training on:     ['person_1']

2019-03-17 15:20:51.028416: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AVX AVX2
 FMA
2019-03-17 15:20:51.291583: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1405] Found device 0 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.582
pciBusID: 0000:0e:00.0
totalMemory: 10.92GiB freeMemory: 274.50MiB
2019-03-17 15:20:51.291649: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1484] Adding visible gpu devices: 0
2019-03-17 15:20:51.580121: I tensorflow/core/common_runtime/gpu/gpu_device.cc:965] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-03-17 15:20:51.580182: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971]      0
2019-03-17 15:20:51.580194: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] 0:   N
2019-03-17 15:20:51.580348: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1097] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 210 MB memory) -> physica
l GPU (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:0e:00.0, compute capability: 6.1)
2019-03-17 15:20:51.581102: E tensorflow/stream_executor/cuda/cuda_driver.cc:903] failed to allocate 210.50M (220725248 bytes) from device: CUDA_ERROR_OUT_OF_MEMORY
```

Keras-yolov3

- Train 完之後會跑出.pkl以及.h5檔
- 這時候可用python3 predict.py –c config.json –i (檔案)來查看預測
- 檔案預設存放在output資料夾內

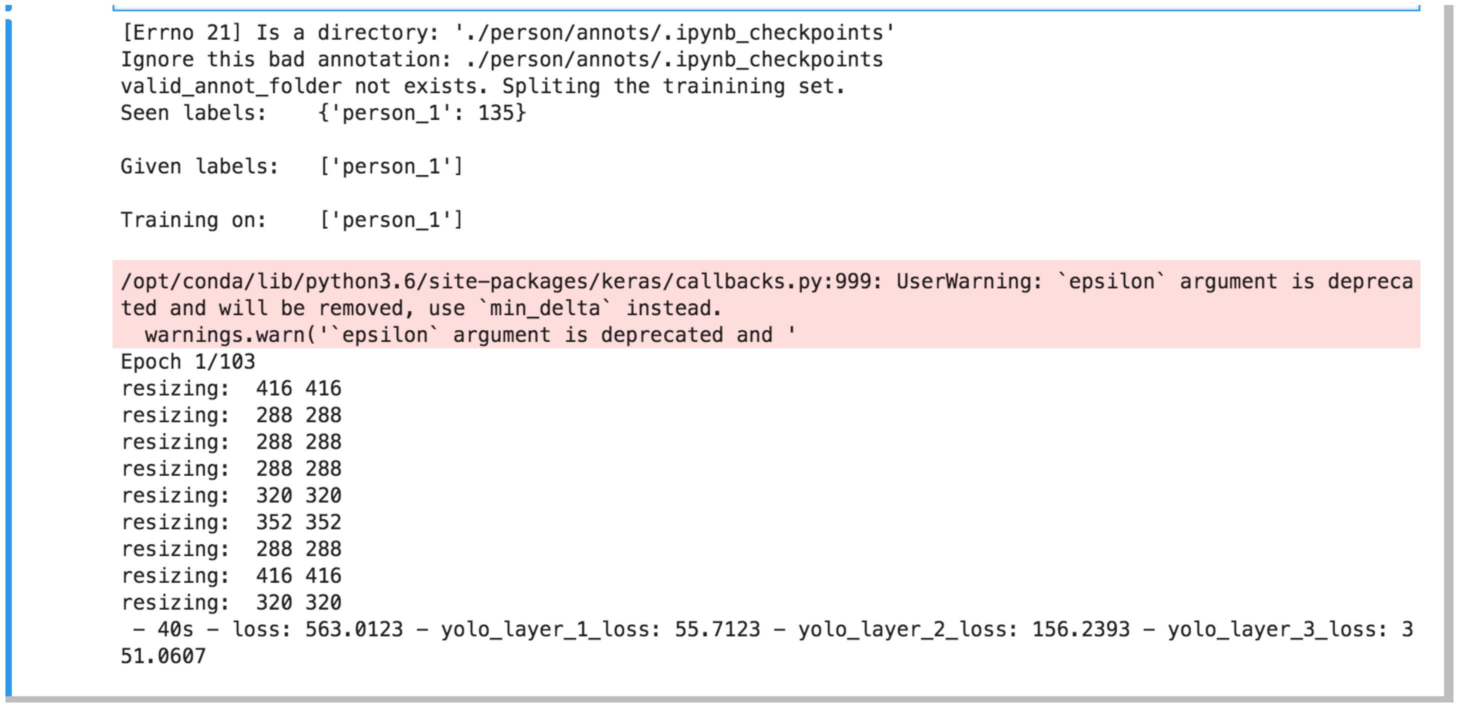


```
(myenv) jovyan@jupyter-kevincho-40aiacademy-2etw:~/keras-yolo3$ python3 predict.py -c config.json -i sample_thu.mp4
/opt/conda/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will
be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
2019-03-17 16:03:34.053971: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AVX AVX2
 FMA
2019-03-17 16:03:34.350549: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1405] Found device 0 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.582
pciBusID: 0000:0e:00.0
totalMemory: 10.92GiB freeMemory: 274.50MiB
2019-03-17 16:03:34.350614: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1484] Adding visible gpu devices: 0
2019-03-17 16:03:34.742895: I tensorflow/core/common_runtime/gpu/gpu_device.cc:965] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-03-17 16:03:34.742971: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971]      0
2019-03-17 16:03:34.742991: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] 0:   N
2019-03-17 16:03:34.743166: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1097] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 210 MB memory) -> physica
l GPU (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:0e:00.0, compute capability: 6.1)
2019-03-17 16:03:34.744300: E tensorflow/stream_executor/cuda/cuda_driver.cc:903] failed to allocate 210.50M (220725248 bytes) from device: CUDA_ERROR_OUT_OF_MEMORY
```

Jupyter notebook

Train

- 改動需要用到的參數, train_annot_folder, train_image_folder, labels, saved_weights_name, nb_epochs 等
- 如果有複數labels的時候，Seen labels 與Given labels需要對齊，否則output 的labels會出錯
- 當有出現Epochs的訓練時代表已開始訓練



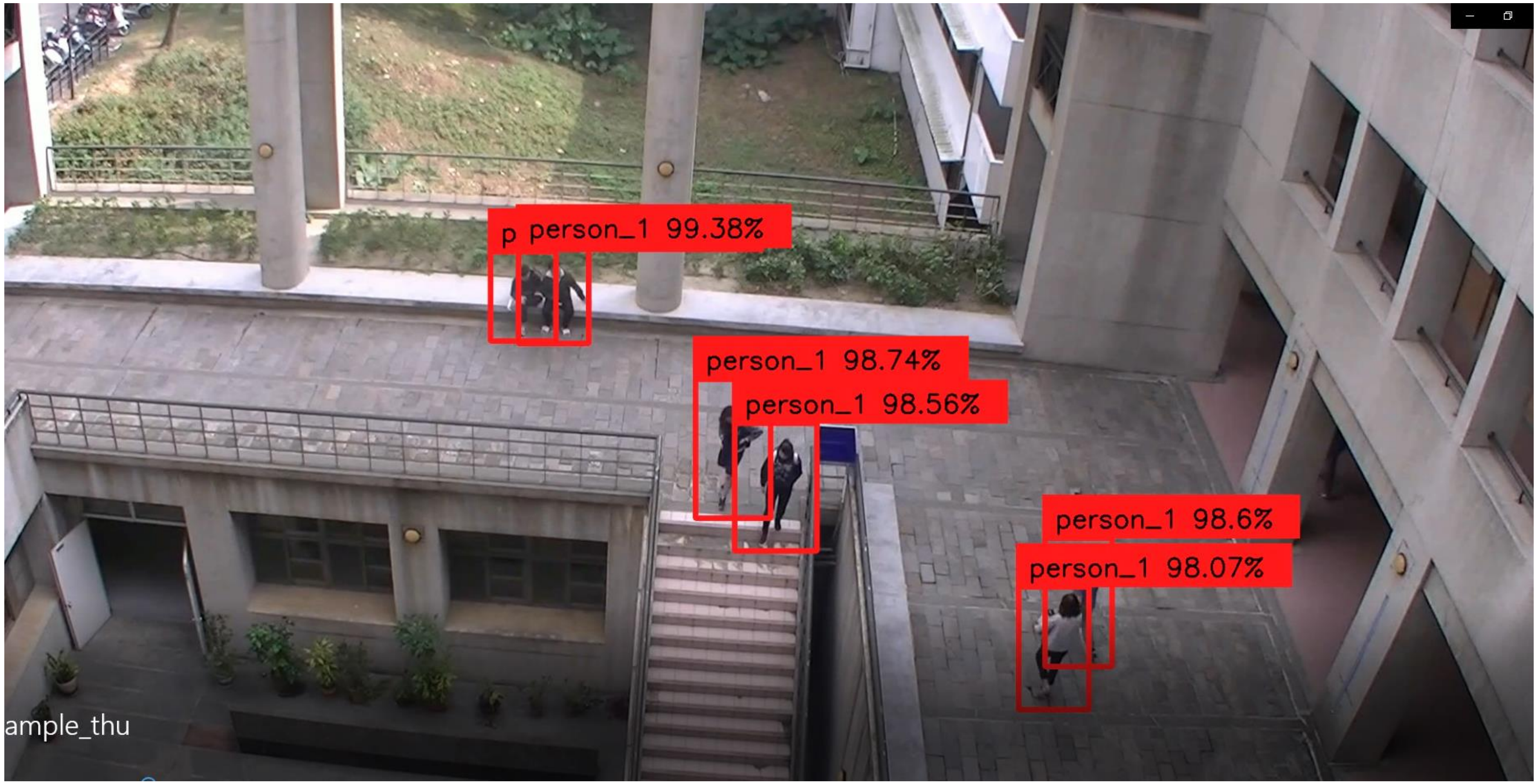
```
[Errno 21] Is a directory: './person/annots/.ipynb_checkpoints'
Ignore this bad annotation: ./person/annots/.ipynb_checkpoints
valid_annot_folder not exists. Spliting the trainining set.
Seen labels:    {'person_1': 135}

Given labels:    ['person_1']

Training on:    ['person_1']

/opt/conda/lib/python3.6/site-packages/keras/callbacks.py:999: UserWarning: `epsilon` argument is depreca
ted and will be removed, use `min_delta` instead.
  warnings.warn('`epsilon` argument is deprecated and '
Epoch 1/103
resizing:  416 416
resizing:  288 288
resizing:  288 288
resizing:  288 288
resizing:  320 320
resizing:  352 352
resizing:  288 288
resizing:  416 416
resizing:  320 320
 - 40s - loss: 563.0123 - yolo_layer_1_loss: 55.7123 - yolo_layer_2_loss: 156.2393 - yolo_layer_3_loss: 3
51.0607
```

Predict

- 訓練結束後可執行Predict.ipynb，並更改input_path,labels,與load_model裡的名字
- 在input_path中可放入.mp4或圖片檔皆可predict
- Predict完的結果可在output資料夾裡查看

Keras-yolov3

- 察看結果



實作參考

- Yolo實作影片
- https://www.youtube.com/watch?v=DhkuUVzK2Vw&t=4s

- **看Yolo3代碼的筆記（寫得也就自己能到懂）**
- https://www.twblogs.net/a/5bf34200bd9eee040518bf66

AND

After Detection

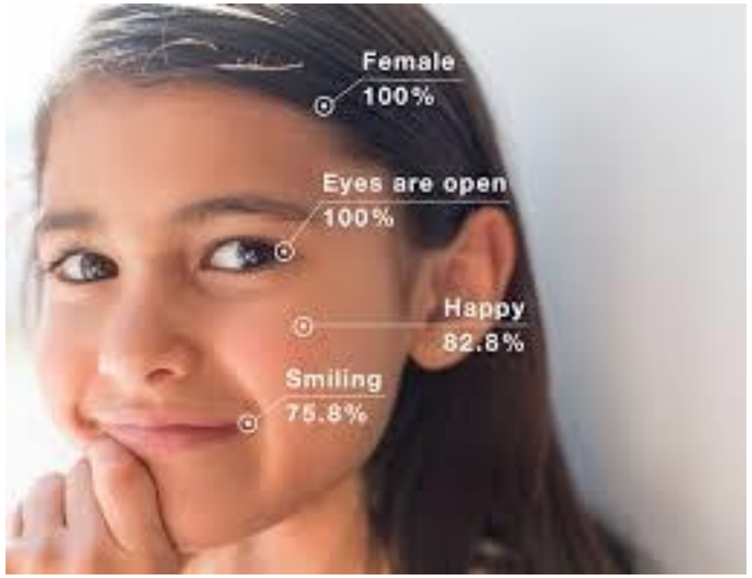
What else you can do?

Face recognition

Face recognition

DMobi

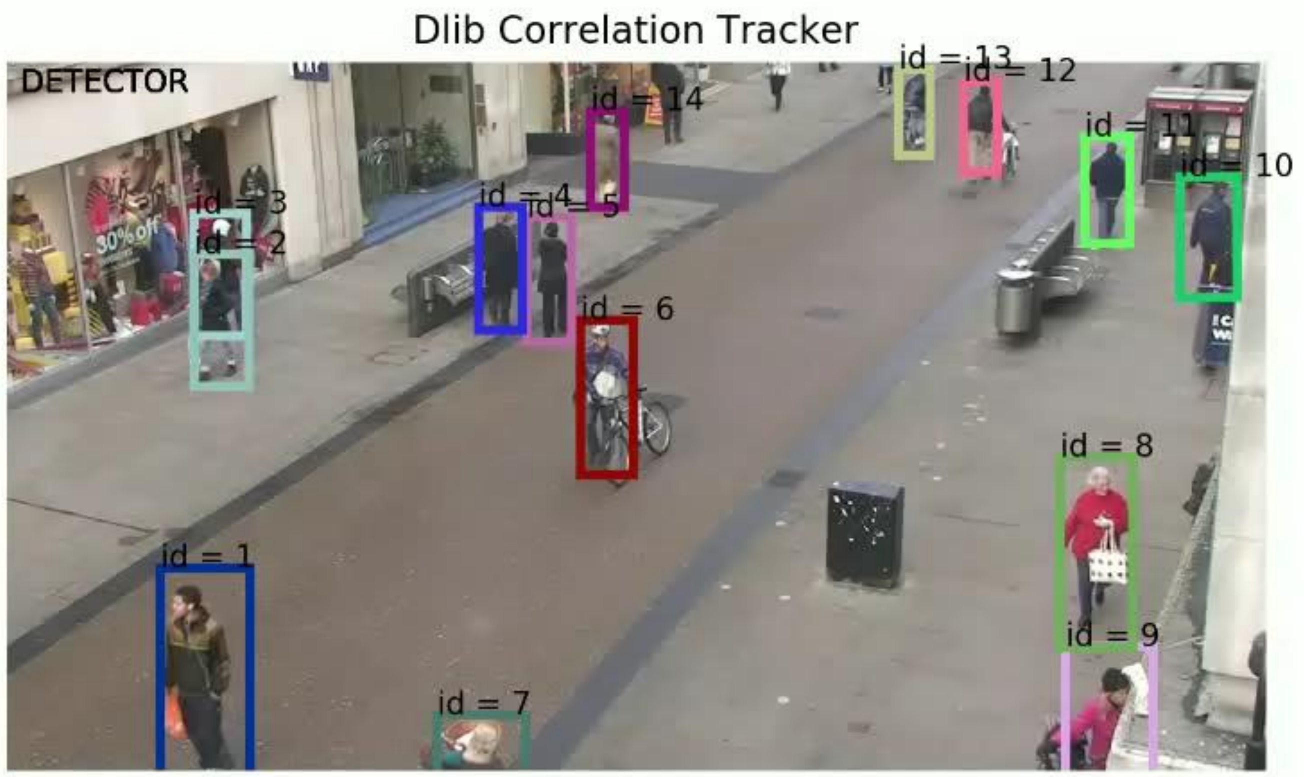
原理



- 人臉識別是由一系列的幾個相關問題組成的：
- 1.找到一張圖片中的所有人臉。
- 2.對於每一張臉來說，無論光線明暗或面朝別處，它依舊能夠識別出是同一個人的臉。
- 3.能夠在每一張臉上找出可用於與他人區分的獨特之處，比如說眼睛有多大，臉有多長等等。
- 4.將這張臉的特點與已知的所有人臉進行比較，以確定這個人的姓名。

Tracking

Tracking


Dlib Correlation Tracker
DETECTOR
id = 1
id = 2
id = 3
id = 4
id = 5
id = 6
id = 7
id = 8
id = 9
id = 10
id = 11
id = 12
id = 13
id = 14

Pose Estimation

Pose

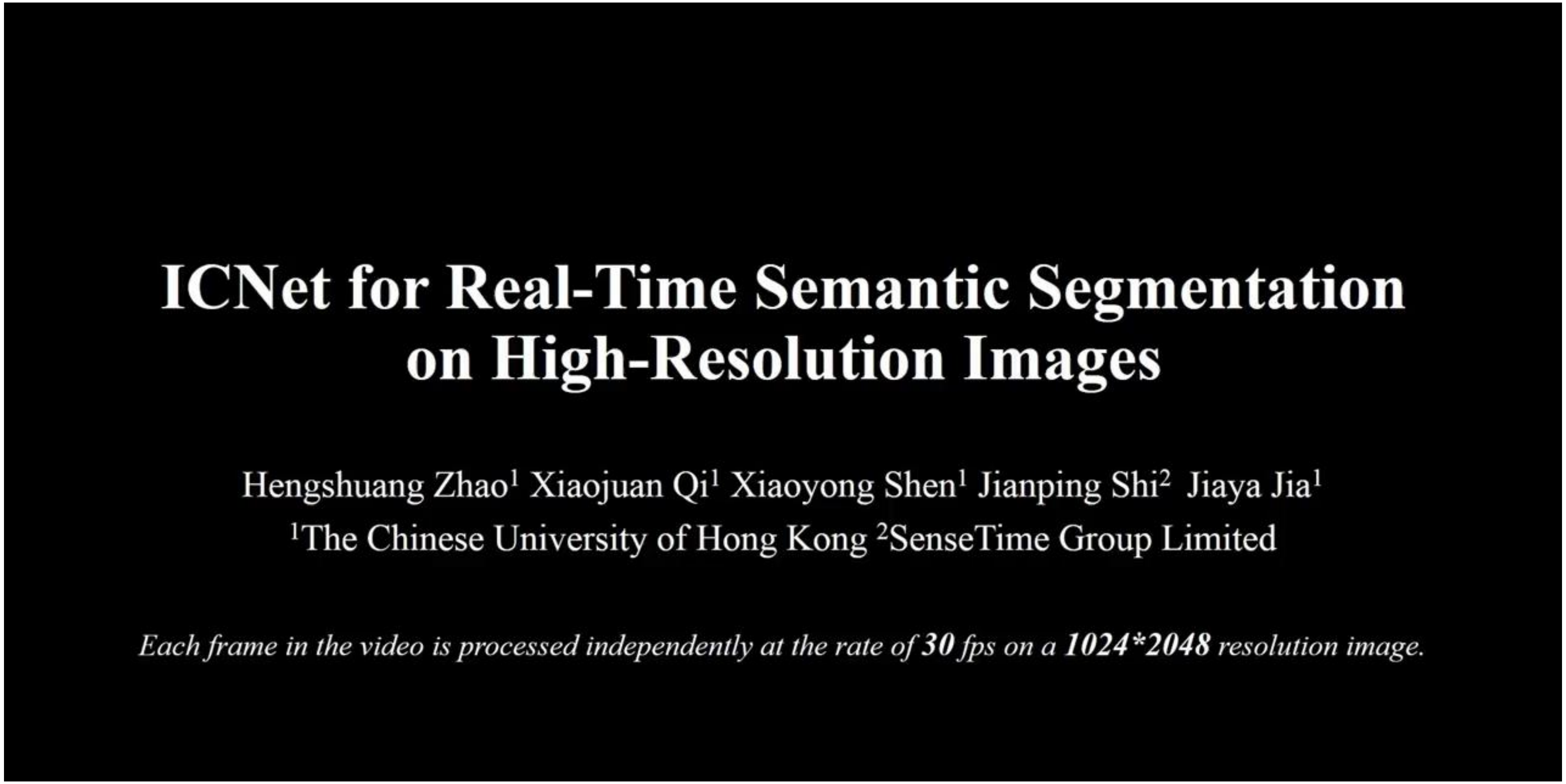
Real-time Multi-Person 2D Pose Estimation Using Part Affinity Fields

Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh

Carnegie Mellon University

Semantic Segmentation

Semantic Segmentation



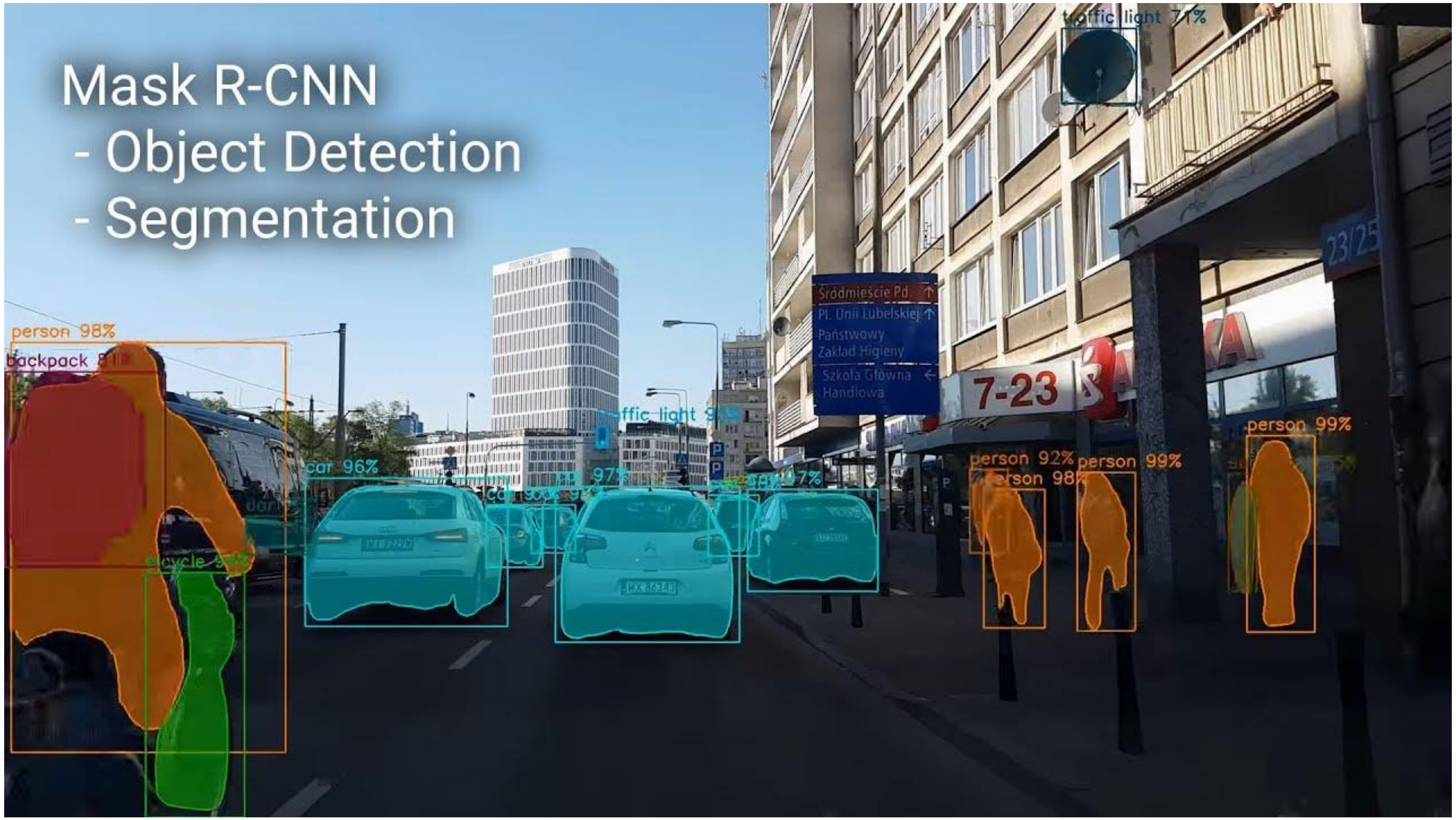
ICNet for Real-Time Semantic Segmentation on High-Resolution Images

Hengshuang Zhao[1] Xiaojuan Qi[1] Xiaoyong Shen[1] Jianping Shi[2] Jiaya Jia[1]

[1]The Chinese University of Hong Kong [2]SenseTime Group Limited

Each frame in the video is processed independently at the rate of ***30*** *fps on a* ***1024*2048*** *resolution image.*

Semantic Segmentation



Person re-identificaiton

Person re-identificaiton

Cam1

Cam2

Thanks