卷積神經網路 Convolutional Neural Network
&
電腦視覺 Computer Vision
Part7

林彥宇&教研處

「版權聲明頁」

本投影片已經獲得作者授權台灣人工智慧學校得以使用於教學用途，如需取得重製權以及公開傳輸權需要透過台灣人工智慧學校取得著作人同意；如果需要修改本投影片著作，則需要取得改作權；另外，如果有需要以光碟或紙本等實體的方式傳播，則需要取得人工智慧學校散佈權。

–台灣人工智慧學校

課程內容

本日課程

1. YOLO
   - YOLO v1 & v2
   - YOLO v2 implement
2. YOLOv2 手把手

延伸閱讀

1. YOLOv3
2. Advanced CV Applications

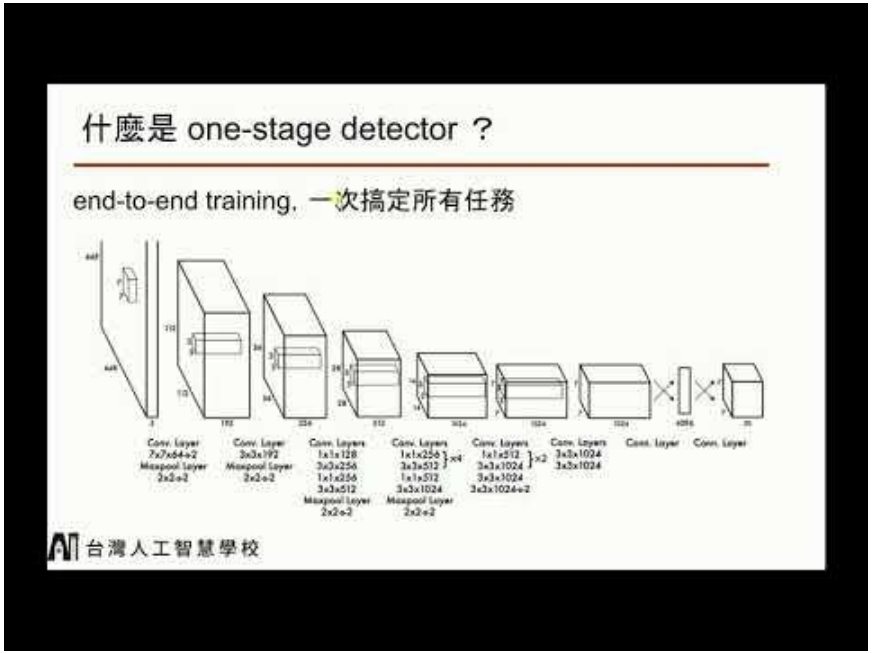
本次課程結束後你 (妳) 應該會什麼?

- **軟實力**
  - 理解One-stage與Two-stage的差別
  - 了解YOLOv1與YOLOv2的架構與程式碼內容
- **硬底子**
  - 完成手把手YOLOv2
  - 了解YOLOv3更新的內容(optional)
  - 了解其他的影像處理方法(optional)

YOLO v1 & v2

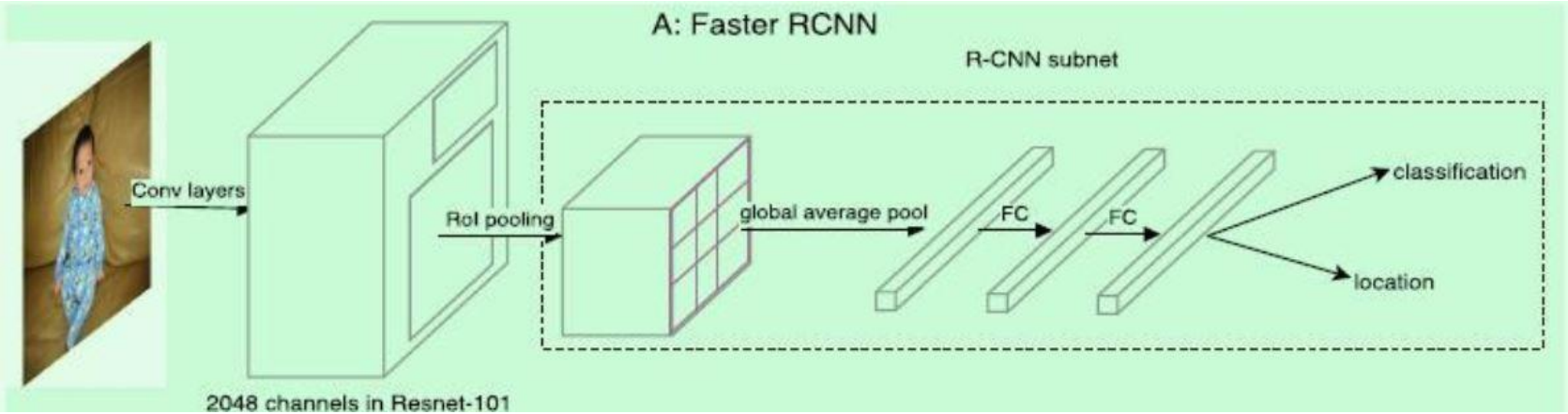
YOLO paper
Project Website

YOLO Introduction



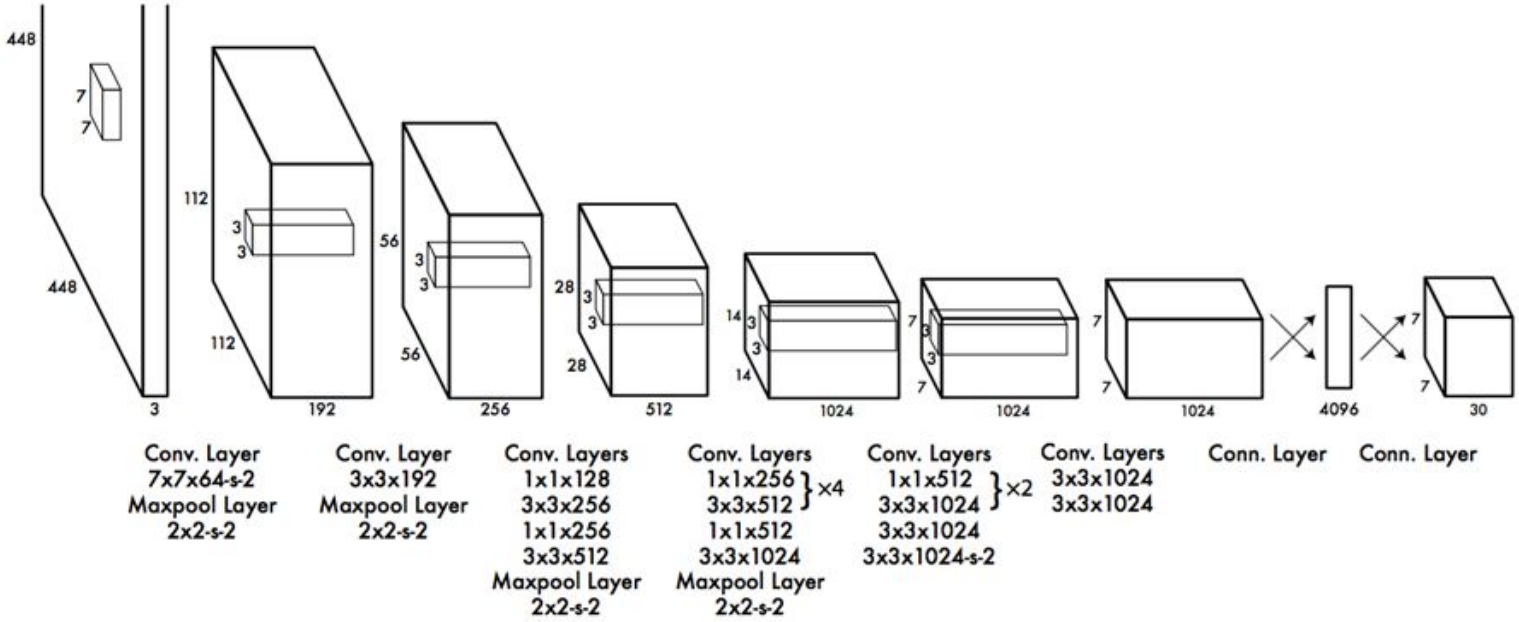
什麼是 two-stage detector ？

將 object detection 分為兩個步驟：localization & classification



什麼是 one-stage detector ？

end-to-end training，一次搞定所有任務



YOLO: Real-Time Object Detection

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Titan X it processes images at 40-90 FPS and has a mAP on VOC 2007 of 78.6% and a mAP of 48.1% on COCO test-dev.

YOLO Structure -1

"We frame object detection as a *regression* problem to spatially separated bounding boxes and associated class probabilities."

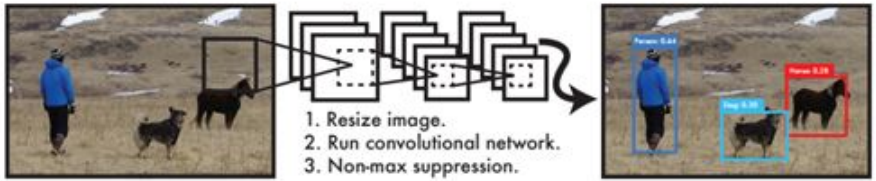
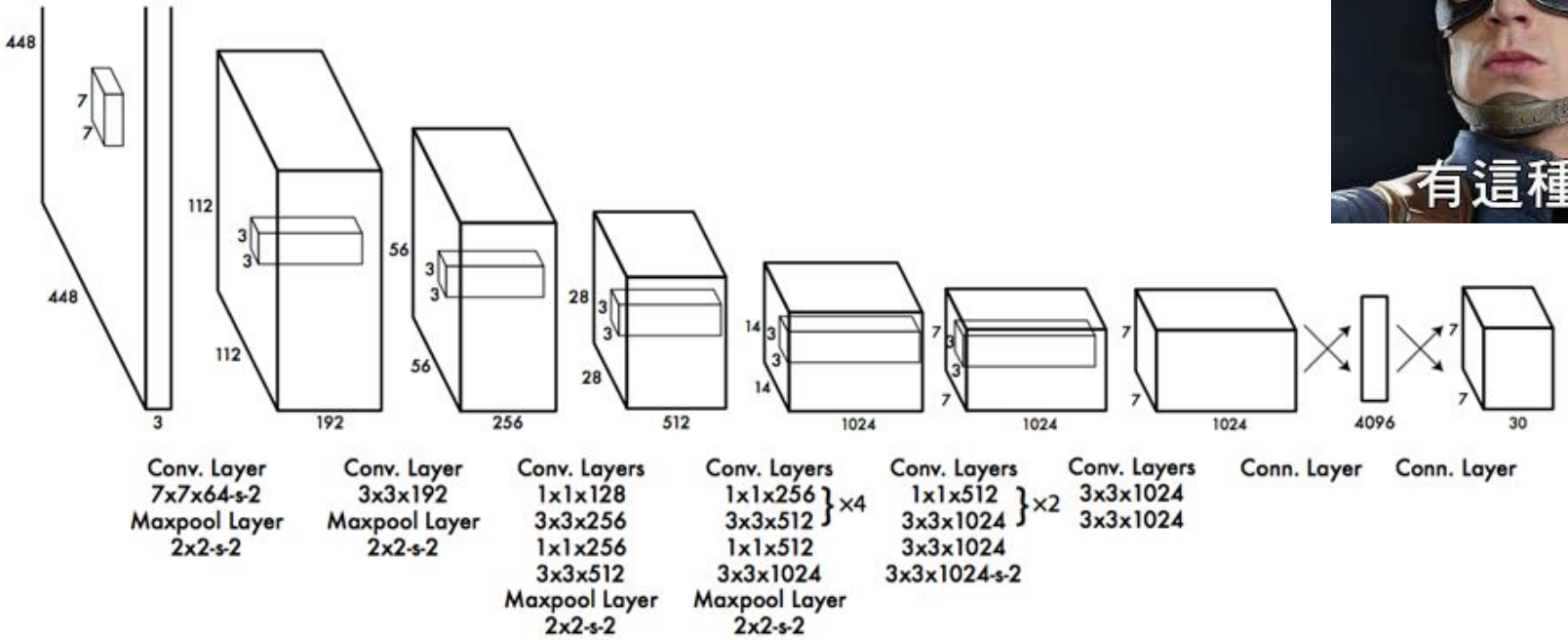


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

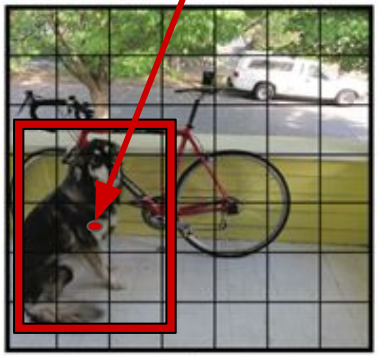
偷喵一下 network 的樣子



劃分成 **S×S** 個grid cell
object box 中心落在某個grid cell內部
則中心 cell 負責 classification

每個grid cell 預測
1. B 個 bbox 的位置 (x, y, w, h)
2. bbox有沒有 object

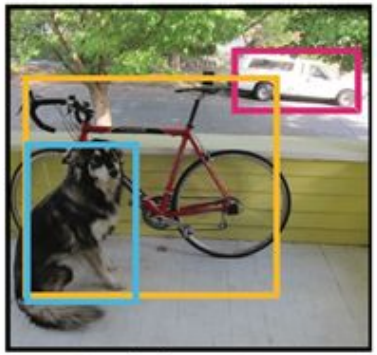
$$\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$



S × S grid on input

Bounding boxes + confidence

Class probability map



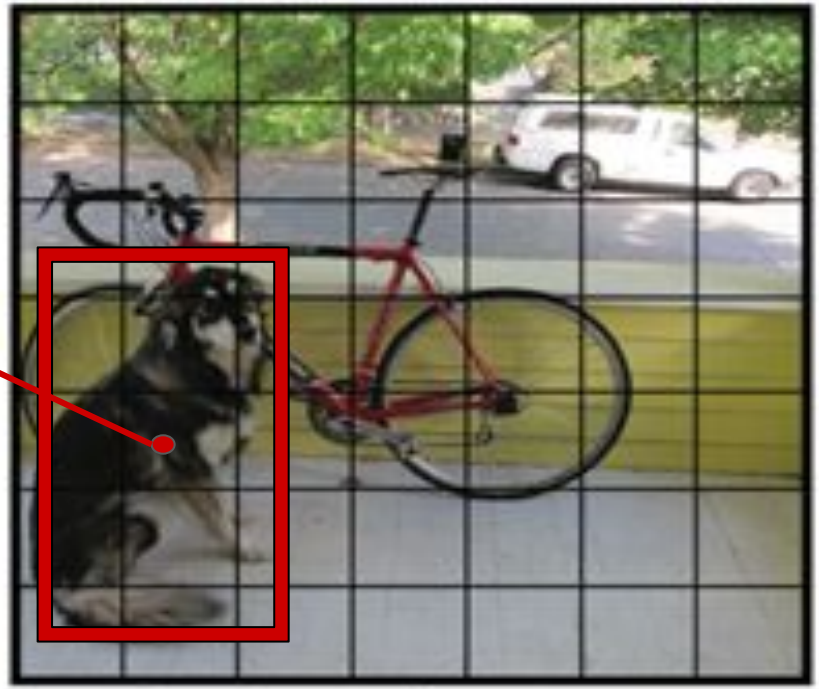
Final detections

Output shape: S x S x (5B + C)
In our case:
S = 7, B = 2, C = 20 (i.e 20 classes)
=> output 是7×7×30 的 tensor

3. (只為) every grid cell 預測 C 個條件機率

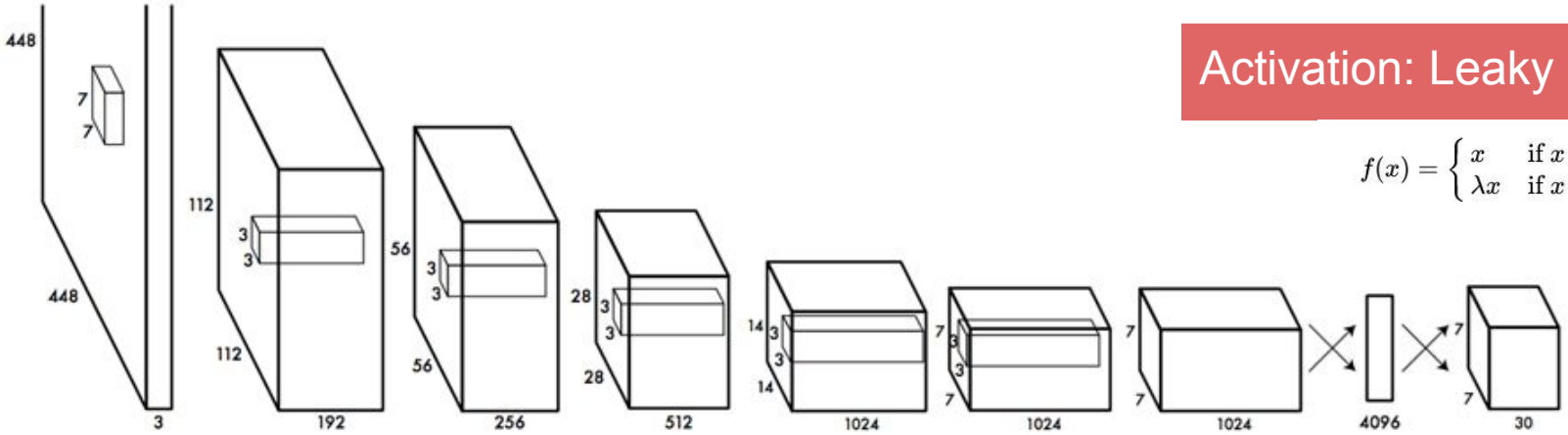
30

x1 = 30
y1 = 400
w1 = 100
h1 = 200
obj_1 = 1
x2 = 0
…
0
0
0
class dog = 1



S × S grid on input

YOLO 網路架構

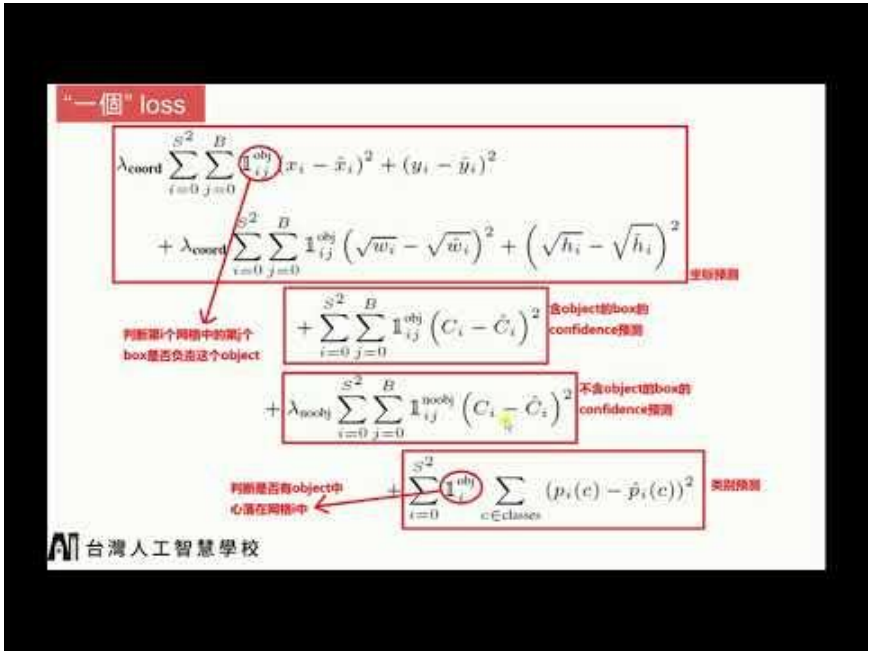


Activation: Leaky ReLU

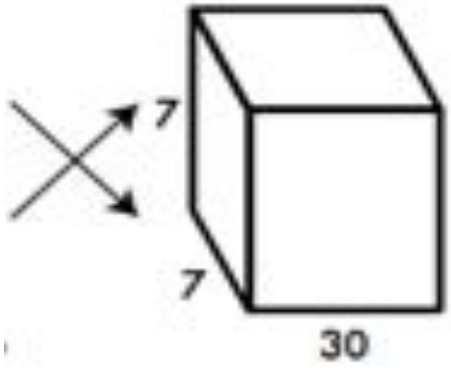
$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \lambda x & \text{if } x \leq 0 \end{cases}$$

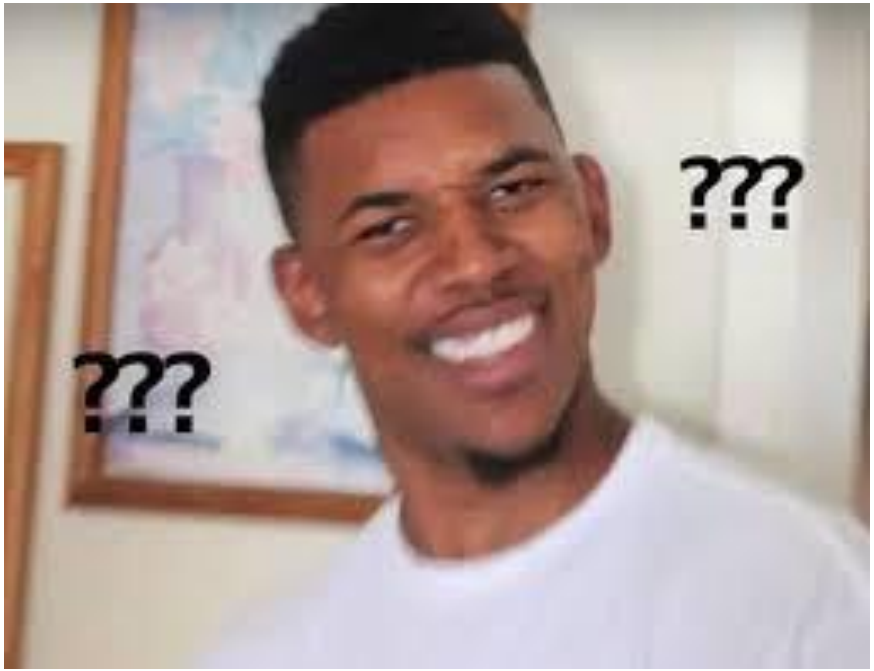
Pretrain on ImageNet
dropout
learning rate schedule
data augmentation

YOLO Structure -2



可以開 train 了！咦，loss function 怎麼算？？？





“一個” loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2$$

坐标预测

判断第i个网格中的第j个box是否负责这个object

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2$$

含object的box的confidence预测

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2$$

不含object的box的confidence预测

判断是否有object中心落在网格i中

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$$

类别预测

“一個” loss 的架構以及問題

- 一個張圖的輸出有 7 × 7 個長度為 30 的 tensor
  - 8: 2 個 bbox 的 x, y, w, h
  - 2: 2 個 bbox 的 confidence (是不是 object)
  - 20: 20 個類別的條件機率

優雅而粗暴地使用 sum-squared error

- 可預見問題
  - 8 個 bbox & 20 個類別的 error 不可相提並論
  - 一個 grid 只會有 2 個 bbox, 一個 class
  - 不同大小的 bbox, 小 bbox 偏一點比大 bbox 偏一點更難接受, 但在 sum-squared error loss 一樣

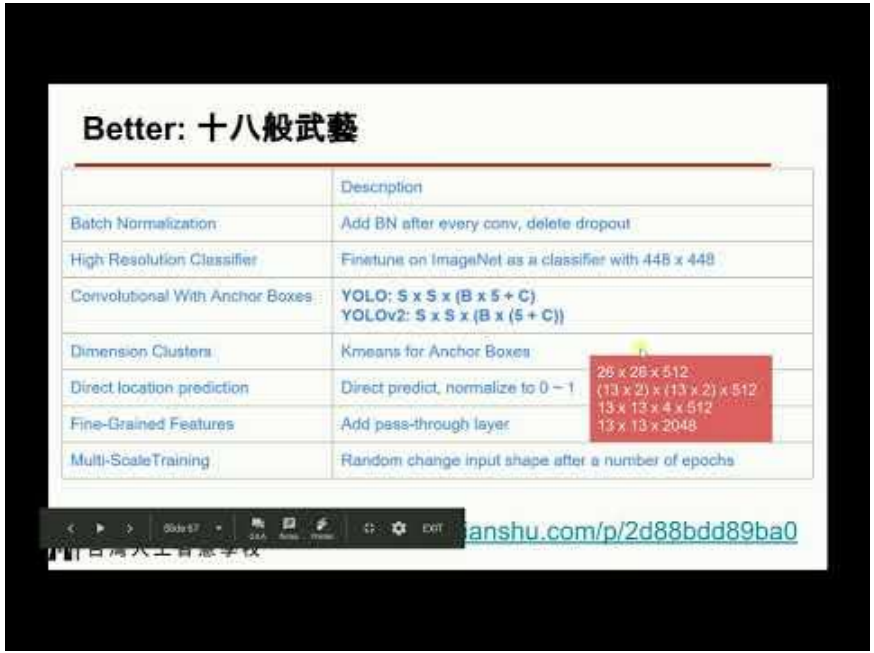
YOLO: Limitation & Generalization of results

好還要更好！

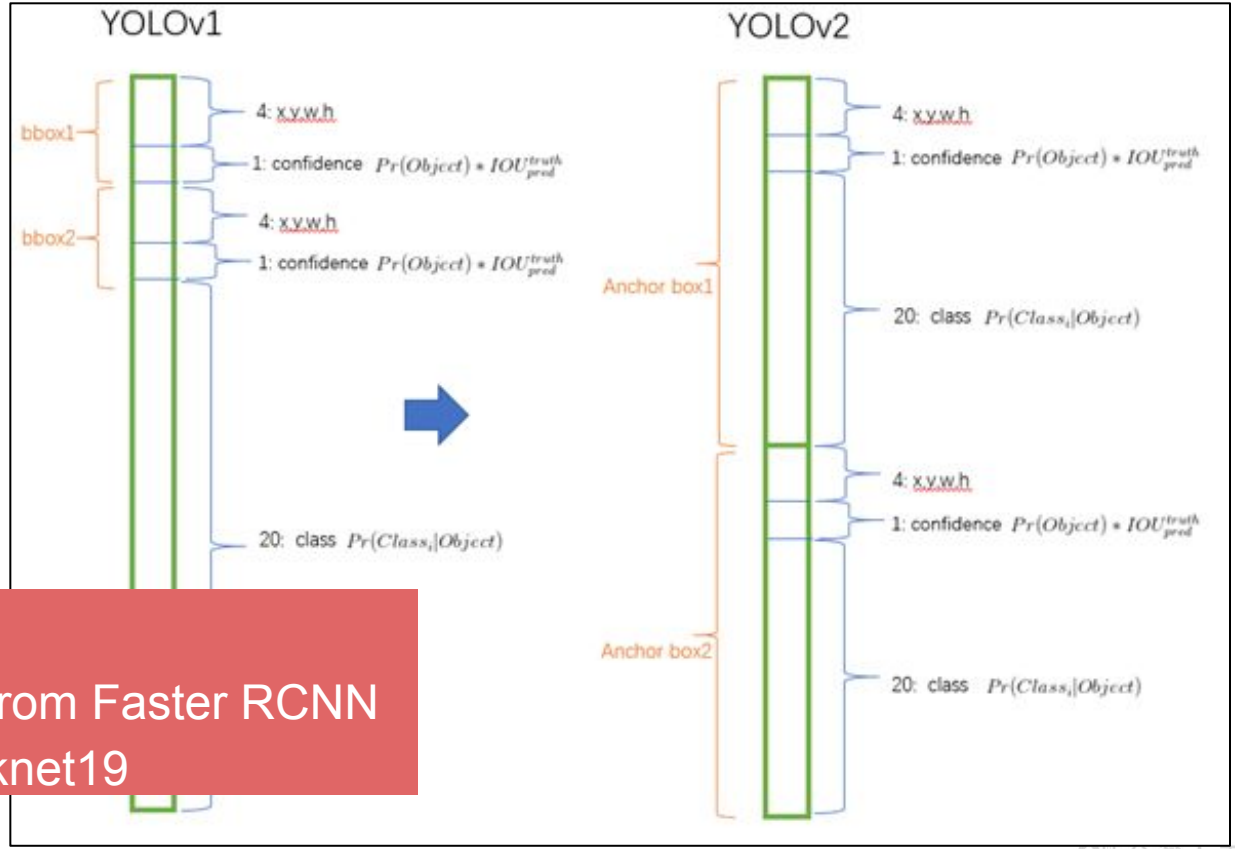
YOLO9000:
Better, Faster, Stronger

Joseph Redmon*†, Ali Farhadi*†
University of Washington*, Allen Institute for AI†
http://pjreddie.com/yolo9000/

YOLO9000



YOLO9000: **Better, Faster**, Stronger



In brief

- 借鑒 anchor 概念 from Faster RCNN
- GoogleNet >> darknet19

Better: 十八般武藝

	Description
Batch Normalization	Add BN after every conv, delete dropout
High Resolution Classifier	Finetune on ImageNet as a classifier with 448 x 448
Convolutional With Anchor Boxes	**YOLO: S x S x (B x 5 + C)** **YOLOv2: S x S x (B x (5 + C))**
Dimension Clusters	Kmeans for Anchor Boxes
Direct location prediction	Direct predict, normalize to 0 ~ 1
Fine-Grained Features	Add pass-through layer
Multi-ScaleTraining	Random change input shape after a number of epochs

26 x 26 x 512
(13 x 2) x (13 x 2) x 512
13 x 13 x 4 x 512
13 x 13 x 2048

https://www.jianshu.com/p/2d88bdd89ba0

Faster: Network 的改善，用 1 x 1 減少參數

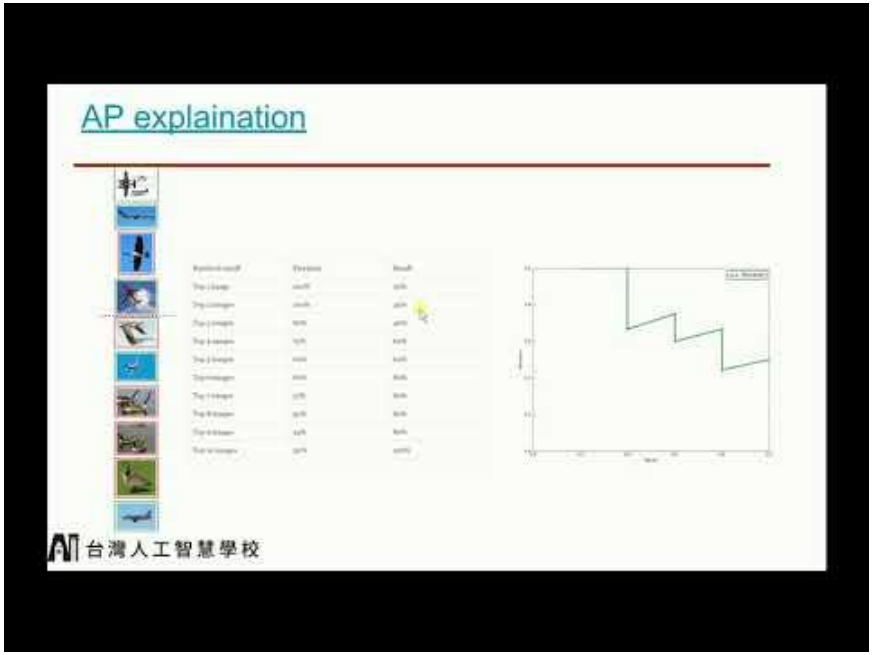
Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Table 6: Darknet-19.

題外話: Why does 1 x 1 conv help?

- https://stats.stackexchange.com/questions/194142/what-does-1x1-convolution-mean-in-a-neural-network
- https://www.zhihu.com/question/56024942

AP Explanation



Compare

	YOLO							YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓
anchor boxes?				✓	✓			
new network?					✓	✓	✓	✓
dimension priors?						✓	✓	✓
location prediction?						✓	✓	✓
passthrough?							✓	✓
multi-scale?							✓	✓
hi-res detector?								✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

AP explaination

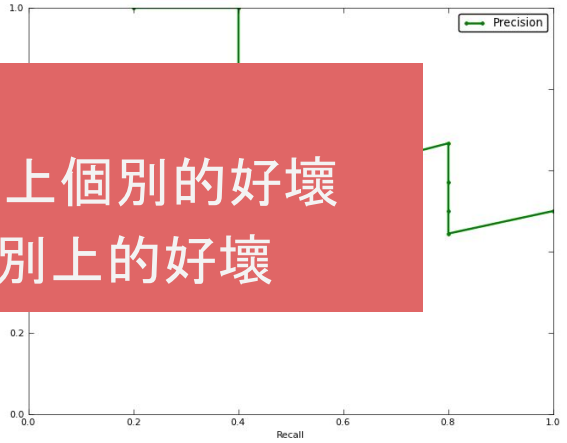
In our example, this is (1 * 0.2) + (1 * 0.2) + (0.66 * 0) + (0.75 * 0.2) + (0.6 * 0) + (0.66 * 0.2) + (0.57 * 0) + (0.5 * 0) + (0.44 * 0) + (0.5 * 0.2) = 0.782.

Retrieval cutoff	Precision	Recall
Top 1 image	100%	20%
Top 8 images	50%	80%
Top 9 images	44%	80%
Top 10 images	50%	100%

一個類別可以算一個 AP

AP 衡量的是學出的模型在每個類別上個別的好壞

mAP 衡量的是學出的模型在所有類別上的好壞

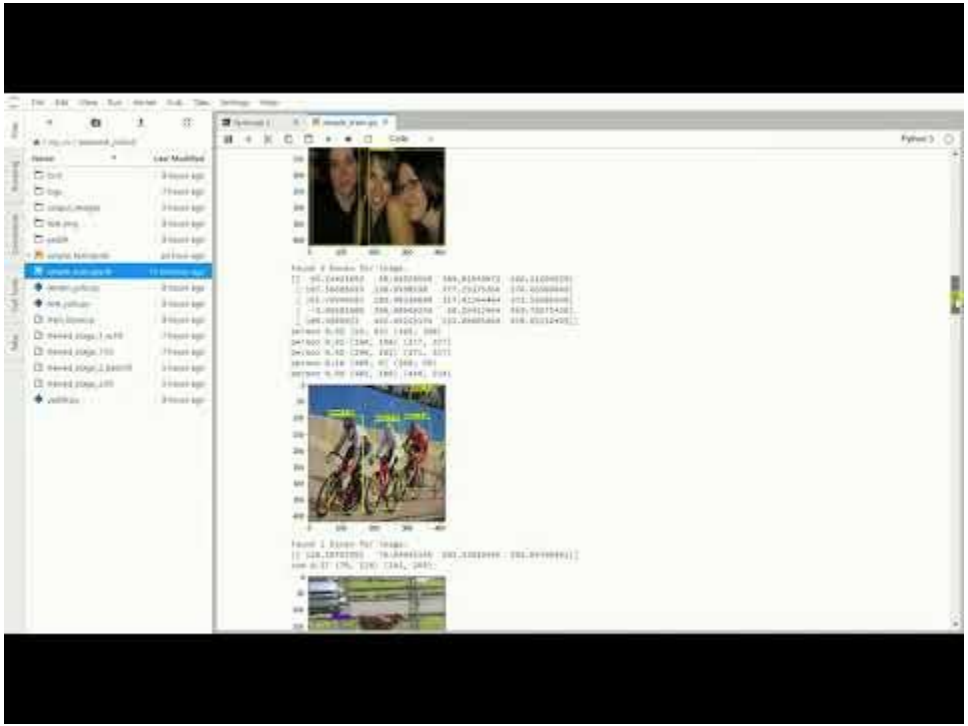


YOLO v2 implement on VOC

https://github.com/allanzelener/YAD2K

YOLO v2 & 9000 paper

YOLOv2 - implementing



研究 Computer Vision 的常用資料集

- Pascal VOC
  - 20 classes
  - http://blog.csdn.net/weixin_35653315/article/details/71028523
- Microsoft COCO
  - 80 classes
  - http://blog.csdn.net/u012905422/article/details/52372755

Pascal VOC

類別

- person
- bird, cat, cow, dog, horse, sheep
- aeroplane, bicycle, boat, bus, car, motorbike, train
- bottle, chair, dining table, potted plant, sofa, tv/monitor

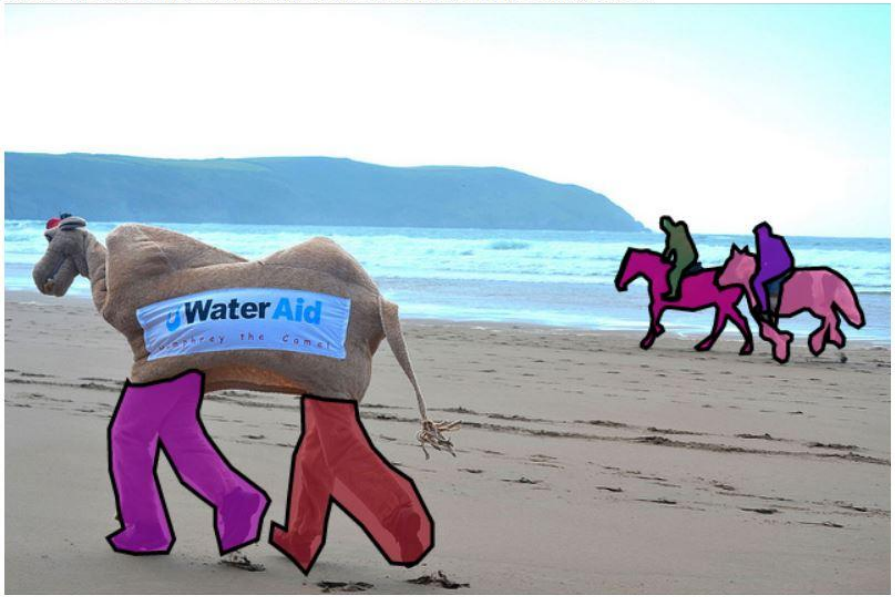
Pascal VOC

```
<annotation>
    <folder>VOC2012</folder>
    <filename>2007_000033.jpg</filename>    //文件名
    <source>                                //文件來源
        <database>The VOC2007 Database</database>
        <annotation>PASCAL VOC2007</annotation>
        <image>flickr</image>
    </source>
    <size>                          //圖片的長，寬，通道數
        <width>500</width>
        <height>366</height>
        <depth>3</depth>
    </size>
    <segmented>1</segmented>        //可否用於語意分割任務，1表示可以，也就是這張圖片在SegmentationClass/Object裡面有
    <object>                        //檢測的目標，如果有多個會有多個<object>標籤
        <name>aeroplane</name>      //目標類別
        <pose>Unspecified</pose>    //拍攝角度
        <truncated>0</truncated>    //是否被截斷，0表示完整
        <difficult>0</difficult>    //目標是否難以辨識，0表示容易識別
        <bndbox>                    //bounding-box，包含左下角和右上角xy座標
            <xmin>9</xmin>
            <ymin>107</ymin>
            <xmax>499</xmax>
            <ymax>263</ymax>
        </bndbox>
    </object>
```

COCO

示意圖

people dressed up in a camel costume and people riding horses along the water
two people dressed in costume advertising while two others ride a horse.
two people on horseback and others in costume on the beach.
two men run on the beach in a camel costume near two horse riders.
two people in a camel suit running on a beach and two people riding horses.



What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints


带上你的武器！

YAD2K

YAD2K: Yet Another Darknet 2 Keras

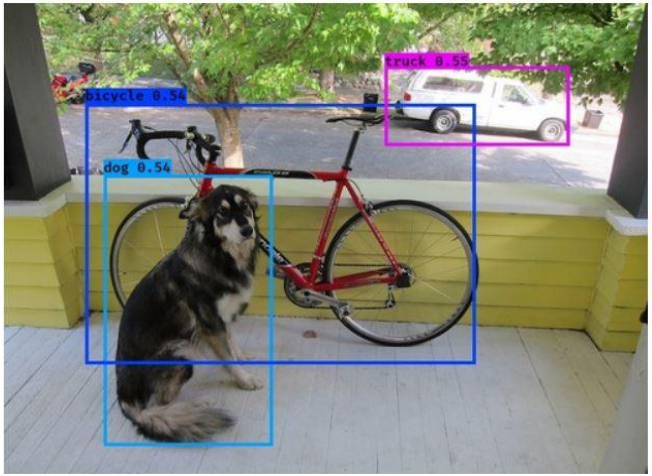
license MIT

Welcome to YAD2K

You only look once, but you reimplement neural nets over and over again.

YAD2K is a 90% Keras/10% Tensorflow implementation of YOLO_v2.

Original paper: YOLO9000: Better, Faster, Stronger by Joseph Redmond and Ali Farhadi.



用 YAD2K 實做的主要目的

- 迅速體驗 training 的過程
- 了解 training 所需要的輸入 input 及 label 格式
- 看懂別人公開的程式碼

練習時間

- 請打開 CNNCV/part7/session4_yolov2/simple_train-comp.ipynb 練習
- 學習視自己的需求更改別人公開的程式碼

Other Resources

- Another one-stage detector: https://arxiv.org/abs/1512.02325
- https://github.com/jbhuang0604/awesome-computer-vision
- Light-Head R-CNN: https://arxiv.org/abs/1711.07264
- A tool for label data: https://github.com/tzutalin/ImageNet_Utils
  - or google "annotation tool for image data"

小天使提醒:
本章全部有下底線的都附有方便的超連結。

Yolo V2 手把手 快速上手

此篇程式碼來源為以下網址，並進行修改
https://github.com/allanzelener/YAD2K

YOLO-V2手把手簡介(一)

YOLO-V2是物件偵測的演算法的一種，輸出屬於Boxes，由於是One Stage的演算法，所以特點是辨識速度較為快速，當然相對的缺點則是辨識效果可能相對於其他Two Stage系列的較為不佳。

YOLO-V2手把手簡介(二)

這邊使用YoloV2需要準備的資料分別有圖片資料集，以及Label完成後**PASCAL VOC**格式的.csv file，差別在於，如果是使用Labelimg(Label工具)中所提供的Yolo格式，本程式是不支援的，請轉成**PASCAL VOC格式。**

在Hub中有提供以下三份.ipynb，方便學員直接上手Yolo-V2

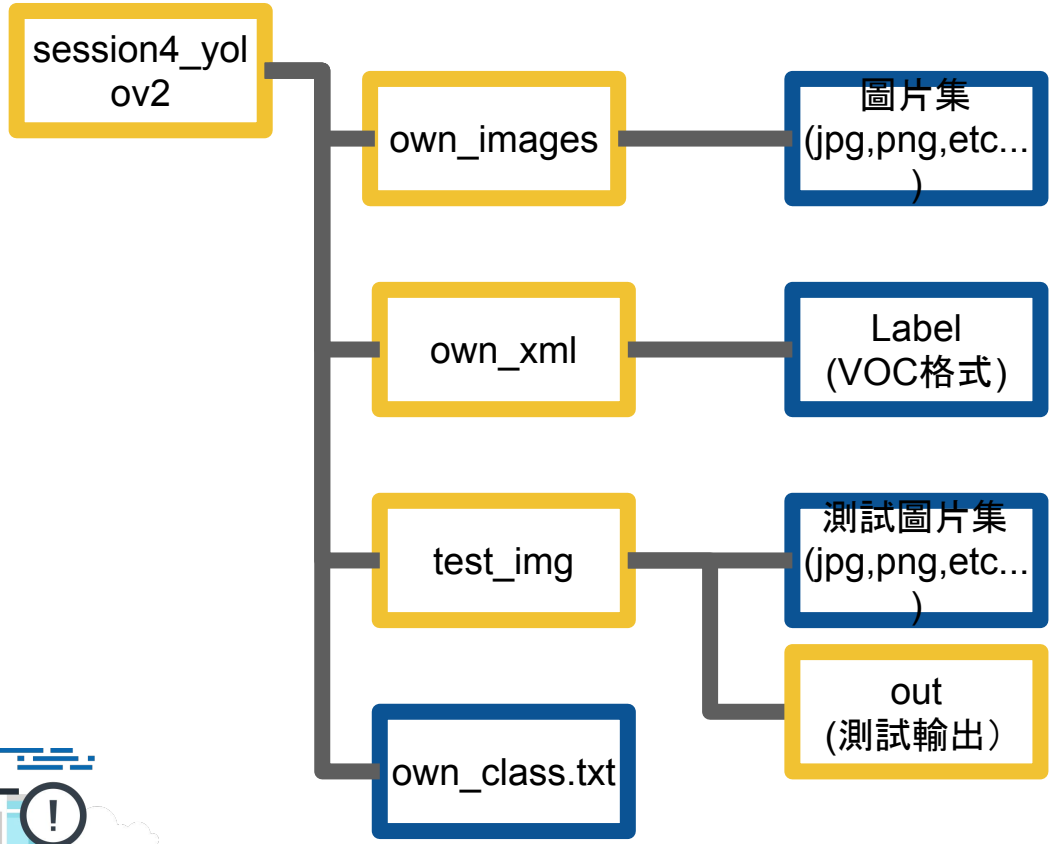
1.simple_train-comp.ipynb(使用Demo資料集訓練)

2.simple_train_owndata.ipynb(使用自己的資料集)

3.simple_test.ipynb(完成訓練後測試辨識)

本篇只說明**simple_train-comp.ipynb**這份程式碼

數據放置位置



資料夾

檔案本體

請將資料集以及Label文件放置對應的資料夾之下。

simple_train-comp.ipynb

```
import os
import cv2
import pickle
#圖片訓練資料集路徑
data_path = '/data/examples/yolo_data/train_img/'

#已經將label的bounding box轉為 model可以讀取的格式
boxes_file = 'train_boxes.p'

# bounding box 的訊息
with open(boxes_file, 'rb') as file:
    boxes_dict = pickle.load(file)

# images 的路徑
file_list = os.listdir(data_path)

print("number of images: {}".format(len(file_list)))
file_list = file_list[:10]
images = [cv2.imread(os.path.join(data_path, f))[:, :, ::-1] for f in file_list]
boxes =  [boxes_dict[f] for f in file_list]
```

小提醒:這邊所使用的label格式與labelimg所輸出的Yolo格式並不相同，若想要訓練自己的資料，Label完後請選擇輸出VOC 格式，在這裡已經有幫各位完成轉換的程式，請在資料夾直接放入VOC檔案即可。

simple_train-comp.ipynb

```
#Label類別與格式
f = open( '/data/examples/yolo_data/model_data/pascal_classes.txt','r')
print(f.read())
```

```
aeroplane
bicycle
bird
boat
bottle
bus
car
cat
.....
```

小提醒:如果訓練自己的資料，物件的類別請以左邊一樣的形式做成Txt檔案，左邊綠底的文字是以Demo的範例，自己的訓練資料並不需要填寫這些類別，請以自己的類別重新建構。

simple_train-comp.ipynb

```
print("images shape: {}".format(images[0].shape))
print("boxes information: {}".format(boxes[0]))

#boxes information 每張圖片 數個物件  每個物件有 5個參數為一個物件, 分別為 [class,x_min, y_min, x_max, y_max]
#若有多個物件則如下方所列出會在同個 List內(下方為例, 共有5個物件)
```

```
images shape: (375, 500, 3)
boxes information: [  8  23 188 193 325  19 149 153 211 211  15 387 170 439 214  15
465 127 491 212  15   2 118  60 185]
```

simple_train-comp.ipynb

```
import retrain_yolo

args = retrain_yolo.argparser.parse_known_args()[0]
args.data_path = data_path
args.boxes_path = boxes_file
args.classes_path = '/data/examples/yolo_data/model_data/pascal_classes.txt'
args.epoch_1 = 10       #此epoch為，第一部分訓練，正常更新全部權重
args.epoch_2 = 30       #此epoch為，凍結前面的權重，只保留更新最後一層含有 bonuding box與class的卷積層
args.own_data = True   #若要使用自己的訓練資料請選 True

t = retrain_yolo._main(args)
```

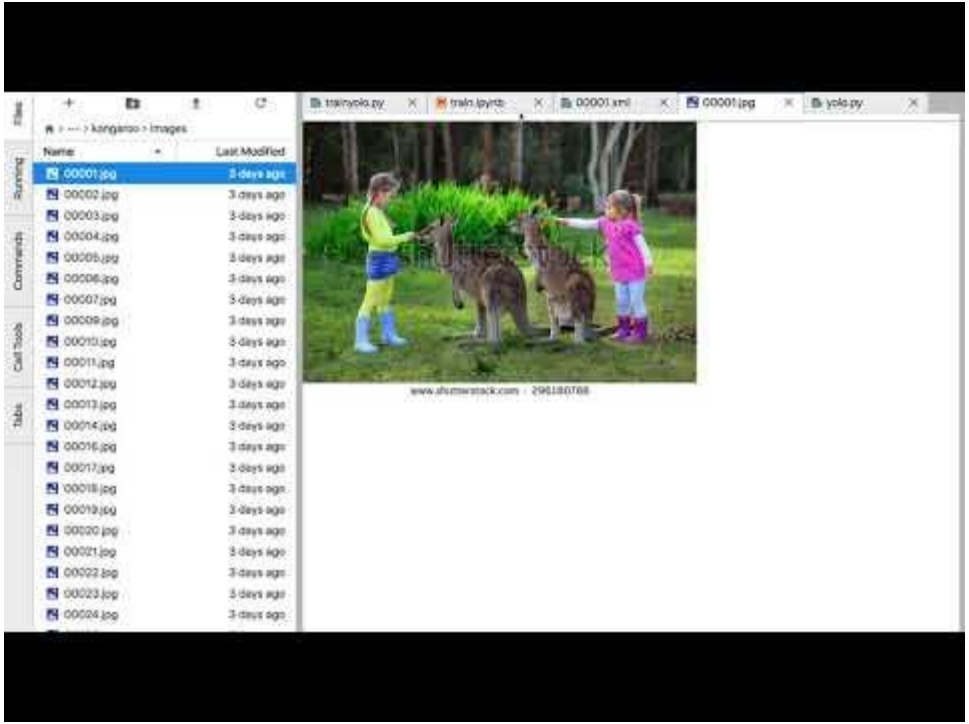
```
Use Demo Data!
Data loaded.
There are 10000 images and boxes
Preprocessing data...
```

延伸閱讀-1

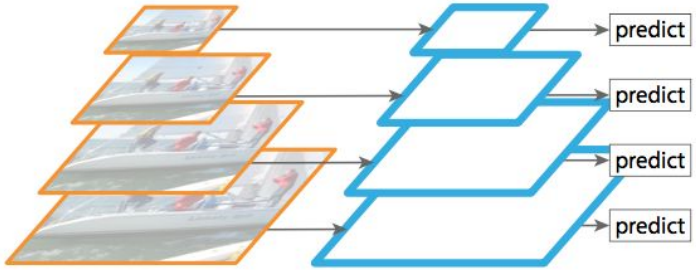
YOLO v3 (Optional)

https://github.com/experiencor/keras-yolo3

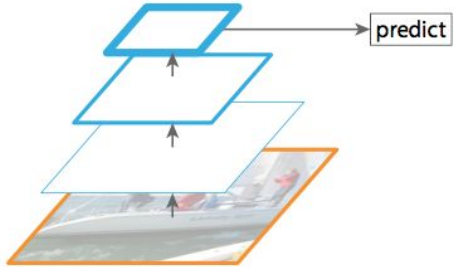
YOLOv3 (computer_vision/keras_yolo3)



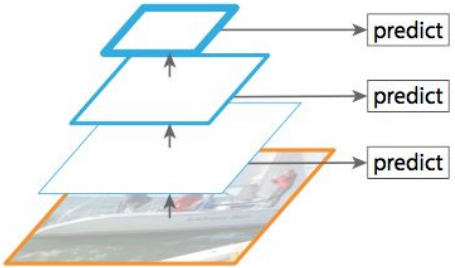
Feature Pyramid Network



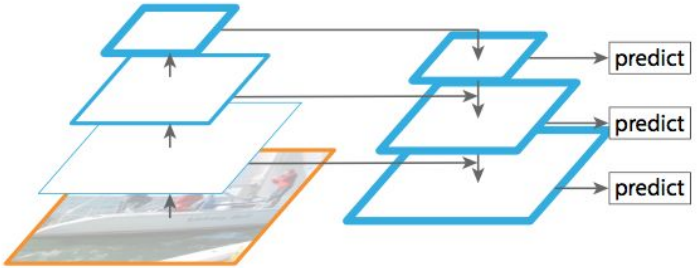
(a) Featurized image pyramid



(b) Single feature map



(c) Pyramidal feature hierarchy



(d) Feature Pyramid Network

Class Prediction

softmax ⇨ logistic classifier + binary cross entropy

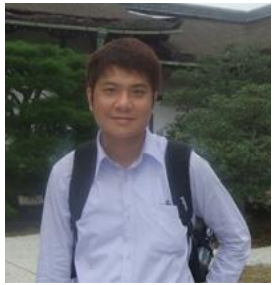
	人	狗
label	1	0
softmax	0.9	0.1
logistic classifier	0.9	0.1

	人	男人
label	1	1
softmax	0.6	0.4
logistic classifier	0.9	0.9

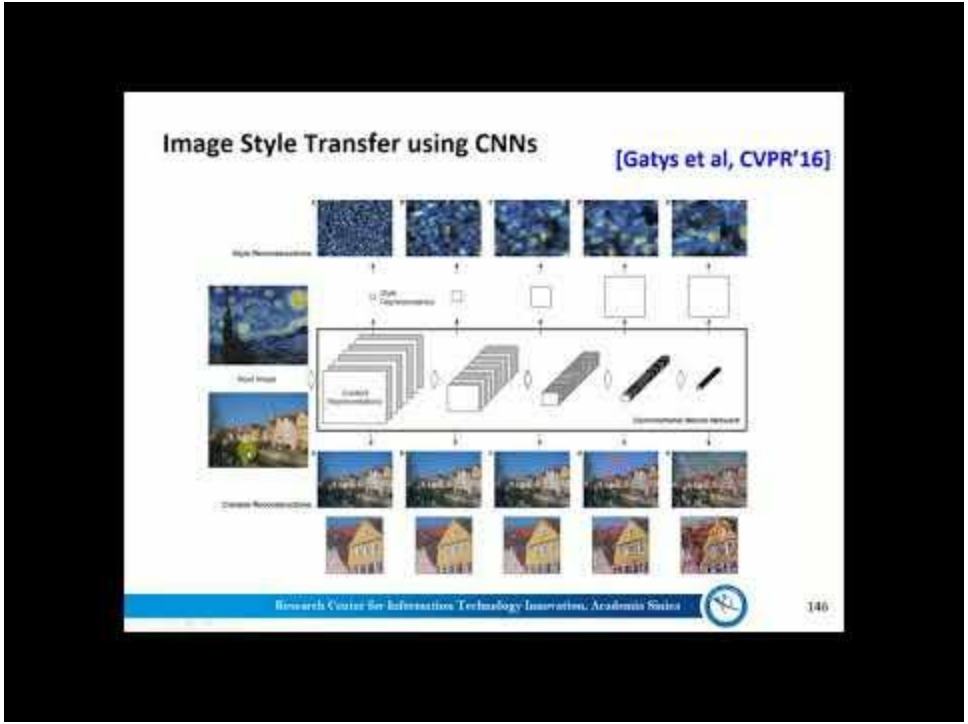
延伸閱讀-2

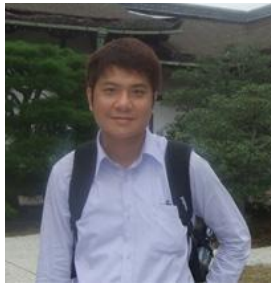
Advanced CV Applications

理論講授 - Super-resolution



理論講授 - Style Transfer





理論講授 - Action Recognition (Optional)