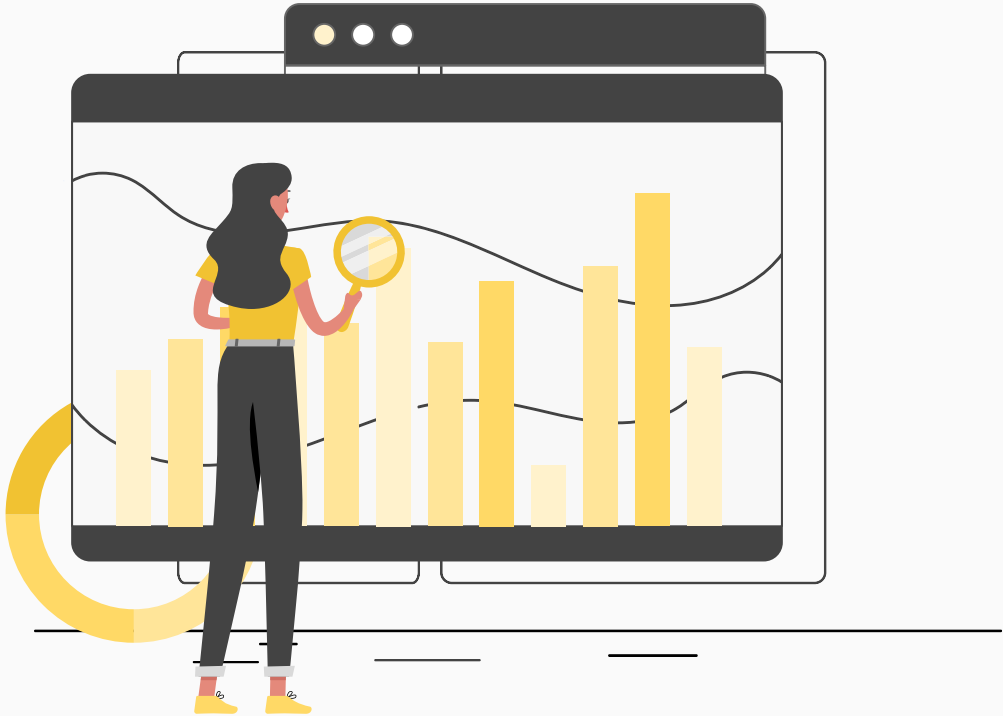
辨識錢幣紙鈔

使用YOLOv4辨識錢幣紙鈔

109034544姜柏宏

目錄

01.
問題定義



5WH

WHY

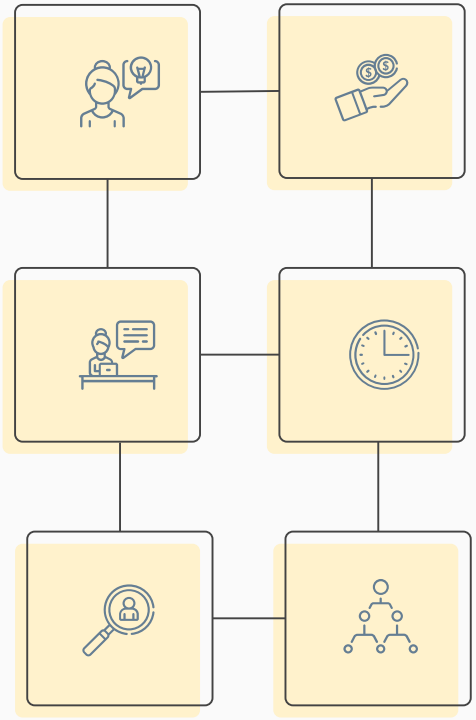
錢幣是每個人日常皆會用到的物品，而常常也有算錯錢找錯錢的情況，希望透過辨識來避免錯誤。

WHERE

所有需要用到錢的場合，店鋪、銀行、家中。

wHo

店員、業主等所有會使用到錢的人。



WHAT

快速算出紙鈔及硬幣數量，降低數鈔時間和避免找錢錯誤。

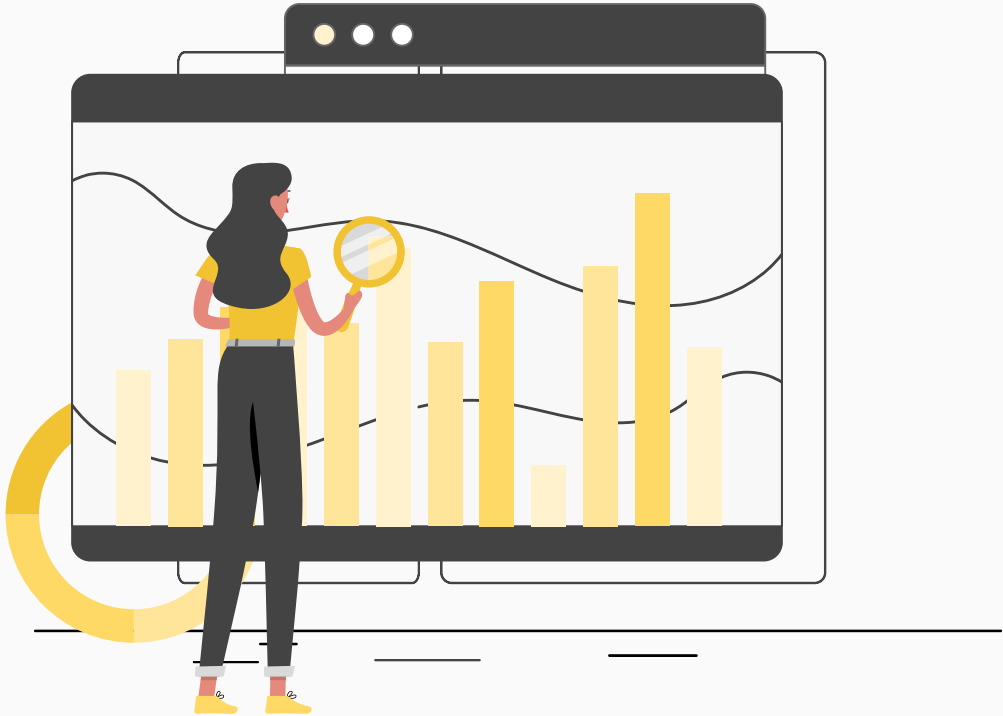
WHEN

買東西、給錢、清點營收等。

HOW

利用卷積神經網路的方法快速辨識螢幕內錢幣數量。

02.
YOLOv4介紹



You only look once

YOLO在從影像輸入到輸出預測結果僅靠一個CNN來實現利用，CNN來同時預測多個bounding-box並且針對每一個box來計算物體的機率，而在訓練的時候也是直接拿整張圖丟到NN中來訓練，這樣end-to-end的算法可以避免傳統object detection的必須分開訓練的缺點，並且大幅加快運算速度。

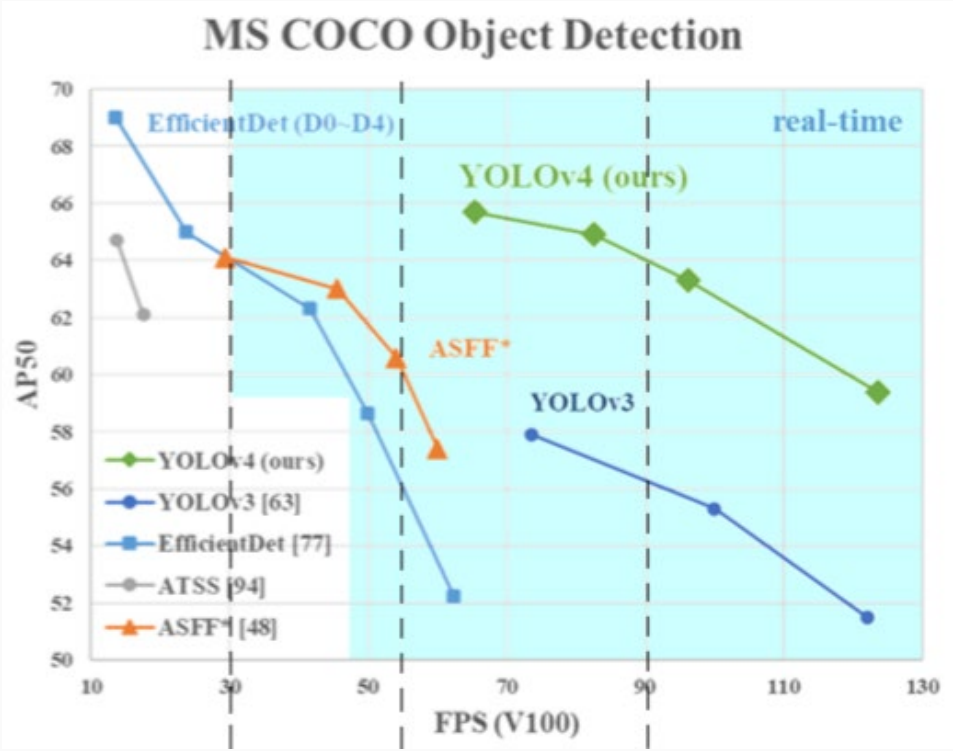
YOLO演進

2020年4月，YOLO v4發佈。其在MS COCO數據集上的精度達到了43.5% AP，速度達到65FPS，與YOLO v3相比分別提高了 10% 和 12%。

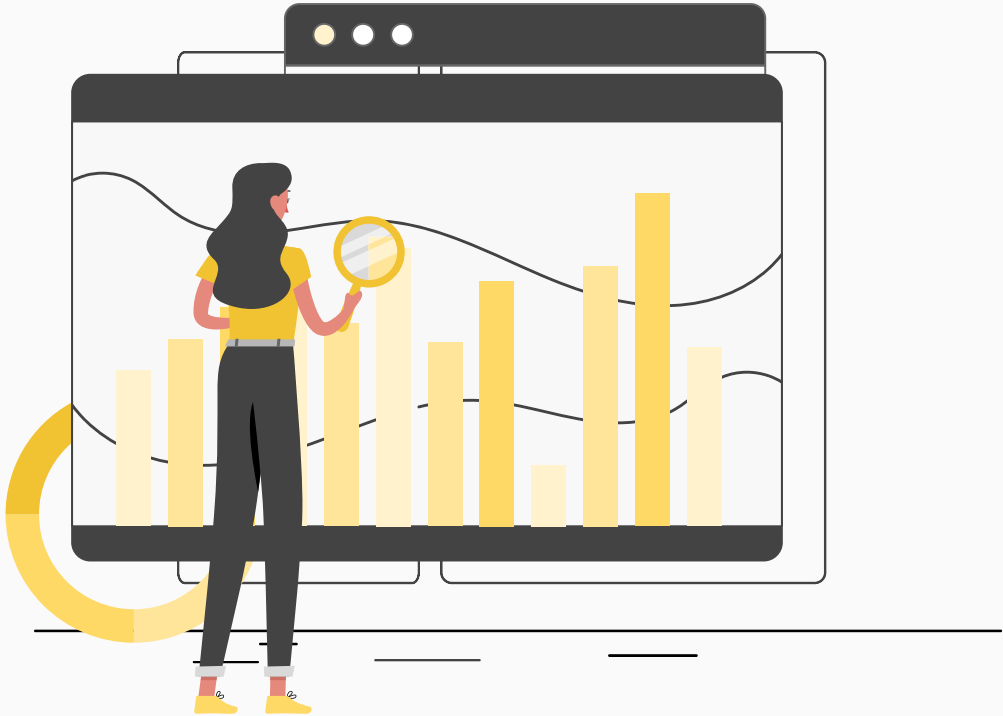
而相較於其他CNN網路模型，其賣點就是超高的FPS，但些微犧牲準確度。

主要貢獻如下：

- 建立了一個高效強大的目標檢測模型，並且使用 1080Ti 或 2080Ti 的 GPU 就可以進行訓練。
- 驗證了SOTA (State of the Art) 的Bag-of-Freebies 和 Bag-of-Specials 目標檢測方法在檢測器訓練過程中的影響。
- 改進了一些 tricks 、SOTA 的方法，包括CBN、PAN、SAM等，使之更加高效，並能夠在單 GPU 上訓練。



03.
研究架構



研究架構

01
資料

準備好欲訓練的資料，各個種類多個角度以及多張照片。

02
標記

利用標記軟體在欲訓練的資料上框出偵測目標。

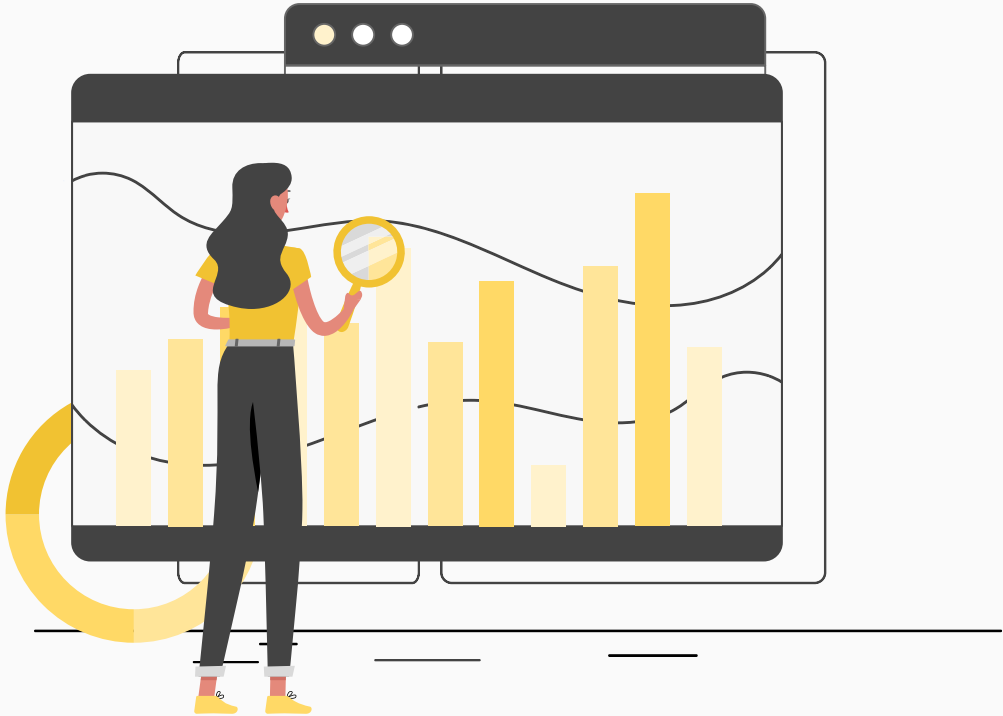
03
訓練

將標記好資料與標記檔送入神經網路中訓練。

04
推論

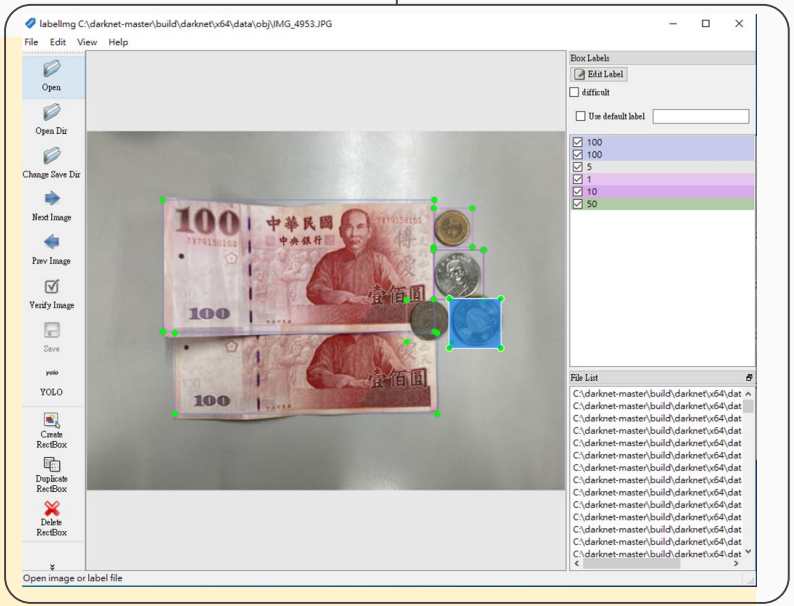
訓練完權重後，即可用來做後續推論新資料和影片檔案。

04.
資料處理



資料預處理

2.Label img



1轉畫質

```python
import glob
import os
from PIL import Image

img_path = glob.glob("C:/darknet-master/build/darknet/x64/data/obj/*.jpg")
path_save = "C:/darknet-master/build/darknet/x64/data/obj/*.jpg"
for file in img_path:
    name = os.path.join(path_save, file)
    im = Image.open(file)
    im.thumbnail((640,640))
    print(im.format, im.size, im.mode)
    im.save(name,'JPEG')
```

3.TXT檔案

IMG_4951 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
0 0.439844 0.383333 0.557813 0.341667
0 0.450000 0.676042 0.553125 0.202083
5 0.709375 0.525000 0.087500 0.104167
4 0.753906 0.269792 0.073438 0.102083
6 0.767969 0.401042 0.085938 0.127083
7 0.807813 0.519792 0.103125 0.143750
```

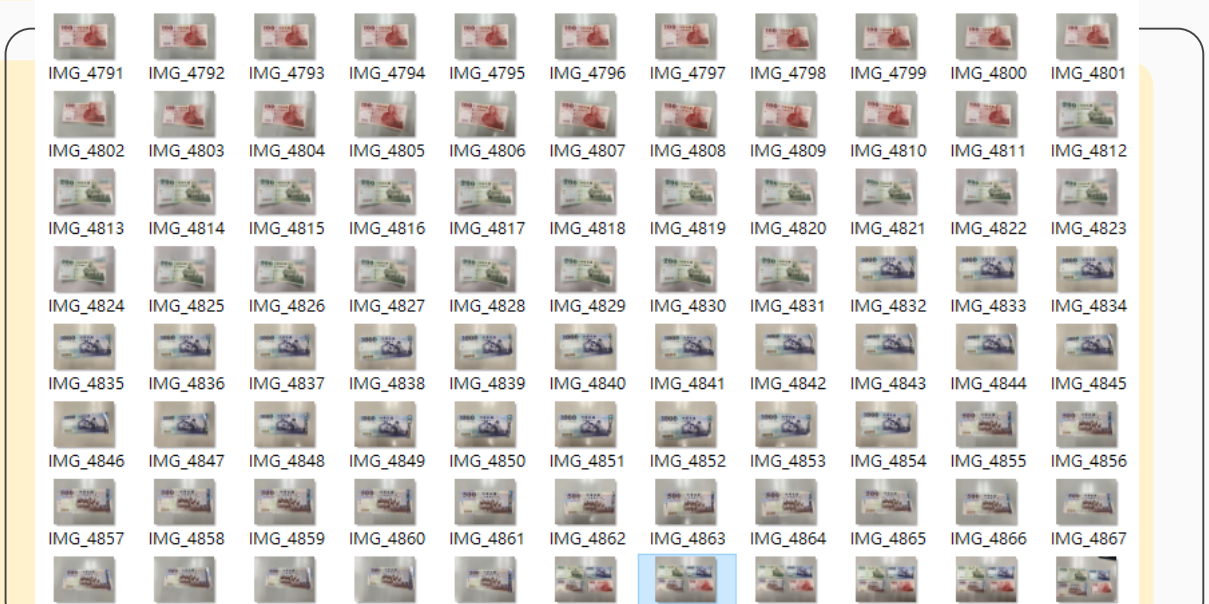
資料預處理

1分類

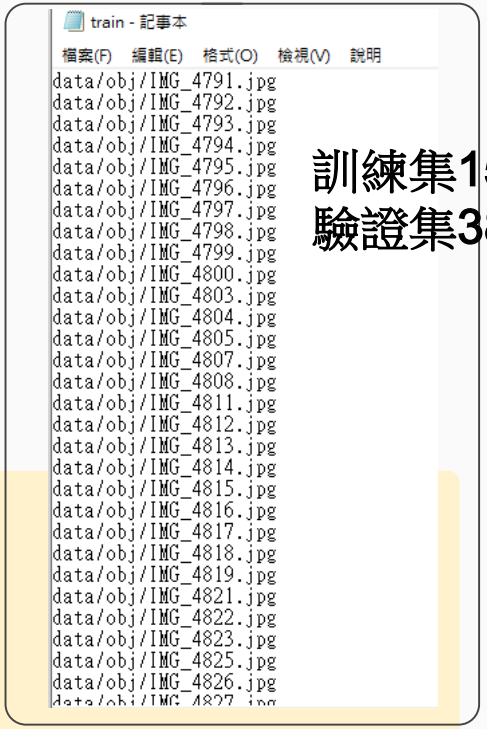
```
num=len(total_xml)
list=range(num)
tv=int(num*trainval_percent)
tr=int(tv*train_percent)
trainval= random.sample(list,tv)
train=random.sample(trainval,tr)

ftrainval = open('VOC2007/ImageSets/Main/trainval.txt', 'w') #指定訓練加驗證集清單檔
ftest = open('VOC2007/ImageSets/Main/test.txt', 'w') #指定測試資料集清單
ftrain = open('VOC2007/ImageSets/Main/train.txt', 'w') #指定訓練資料集清單
fval = open('VOC2007/ImageSets/Main/val.txt', 'w') #指定驗證資料集清單

for i  in list:
    name=total_xml[i][:-4]+'\n'
    if i in trainval:
        ftrainval.write(name)
        if i in train:
            ftrain.write(name)
        else:
            fval.write(name)
    else:
        ftest.write(name)
```

```
trainval_percent = 1 #訓練加驗證集佔全部資料集比例
train_percent = 0.8 # 訓練集佔訓練加驗證集比例
xmlfilepath = 'VOC2007\Annotations' #標註檔路徑
txtsavepath = 'VOC2007\ImageSets\Main' #訓練、驗證、測試清單路徑
total_xml = os.listdir(xmlfilepath)
```

2.欲訓練資料

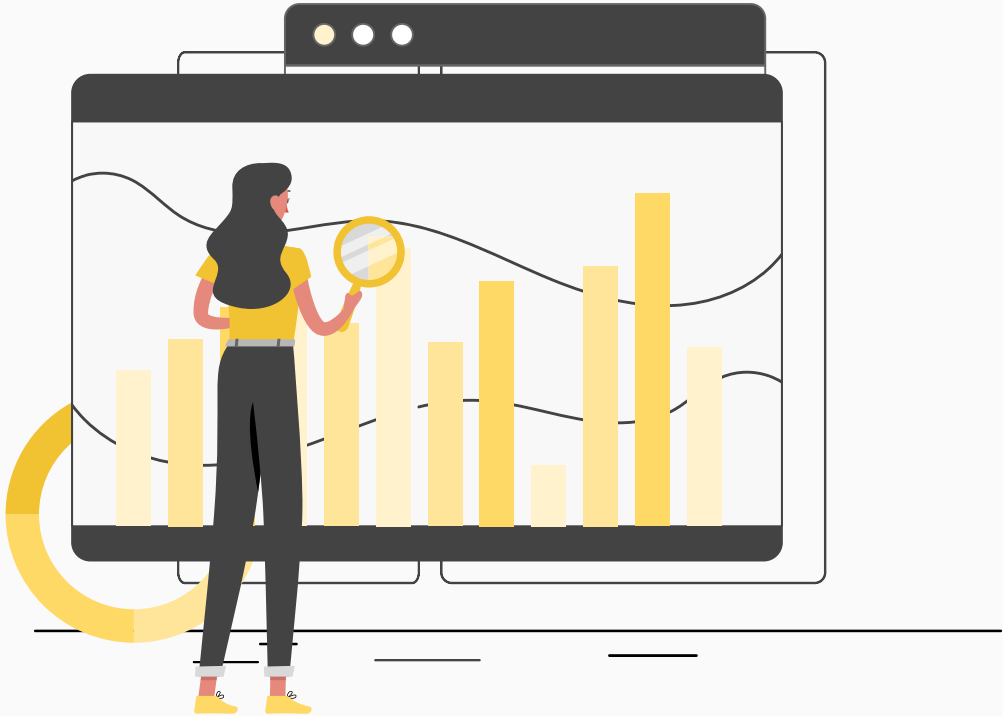


3.TXT檔案



訓練集151張
驗證集38張

05.
模型架構

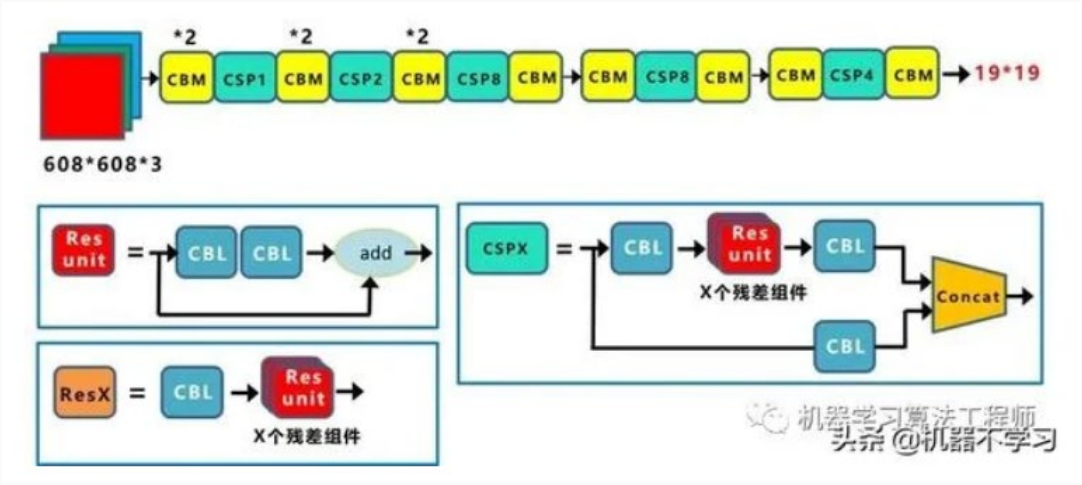


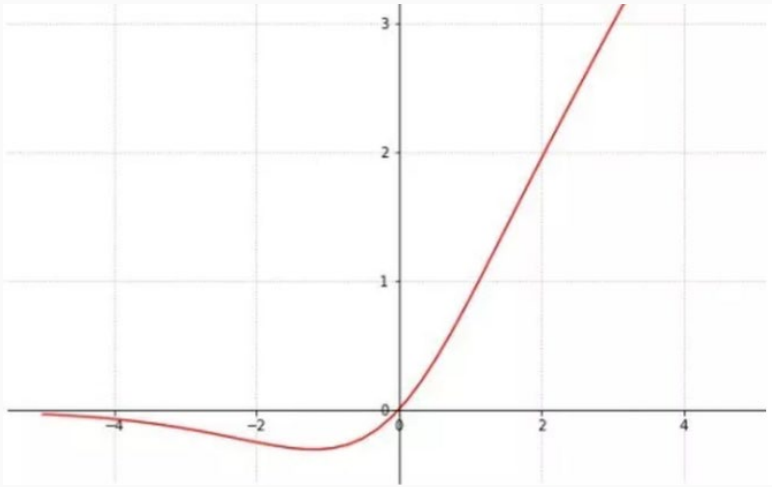
CSPDarknet53 BackBone

Yolov4在主幹網絡Backbone採用CSPDarknet53網絡結構，主要有三個方面的優點：

- 優點一：增強CNN的學習能力，使得在輕量化的同時保持準確性。
- 優點二：降低計算瓶頸
- 優點三：降低內存成本

Yolov4的Backbone中都使用了Mish激活函數，而後面的網絡則還是使用leaky_relu函數。



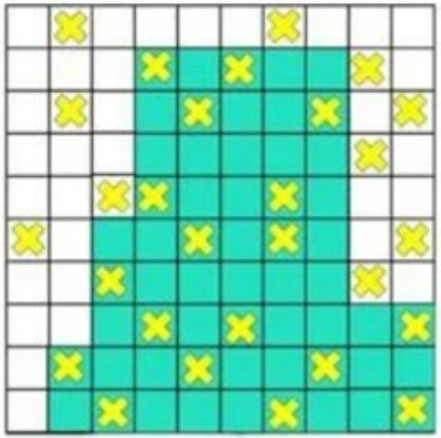


CSPDarknet53 + Dropblock

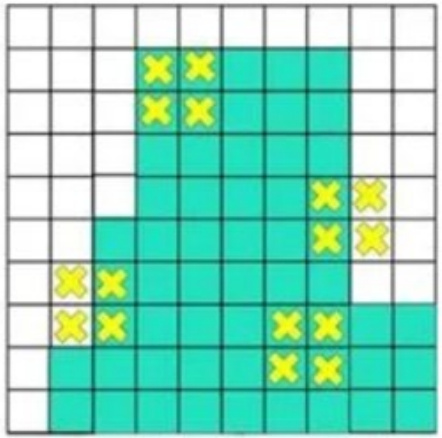
Yolov4中使用的Dropblock，其實和常見網絡中的Dropout功能類似，也是緩解過擬合的一種正則化方式。

因為卷積層通常是三層連用：卷積+激活+池化層，池化層本身就是對相鄰單元起作用。而且即使隨機丟棄，卷積層仍然可以從相鄰的激活單元學習到相同的信息，因此，在全連接層上效果很好的Dropout在卷積層上效果並不好。
所以從右圖Dropblock的研究者則乾脆整個局部區域進行刪減丟棄。

原始圖片

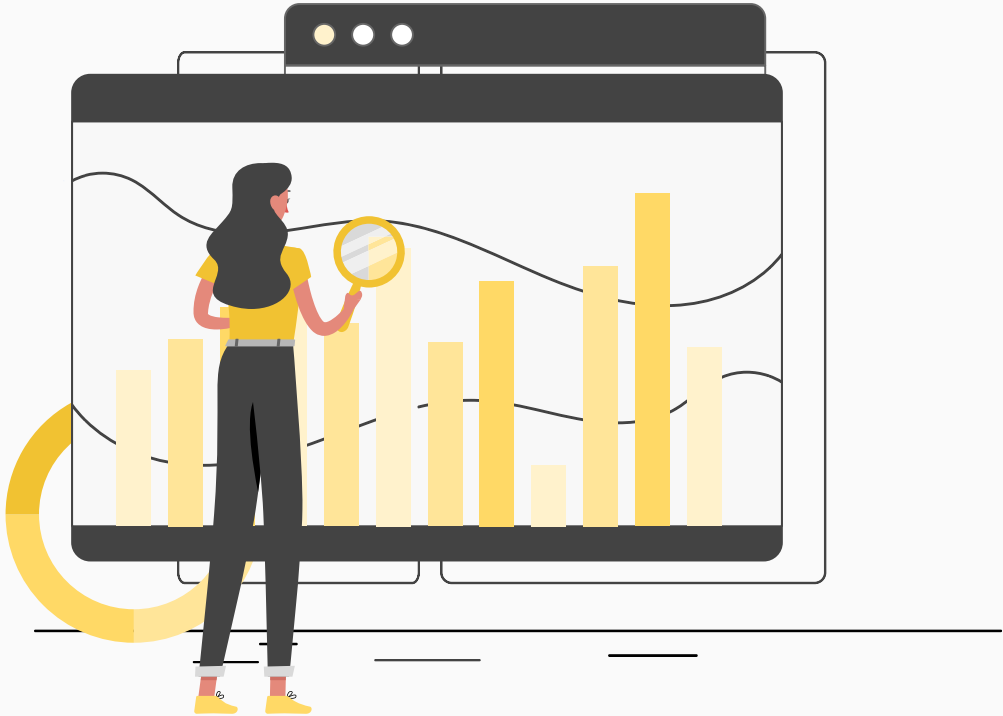


dropout



dropblock

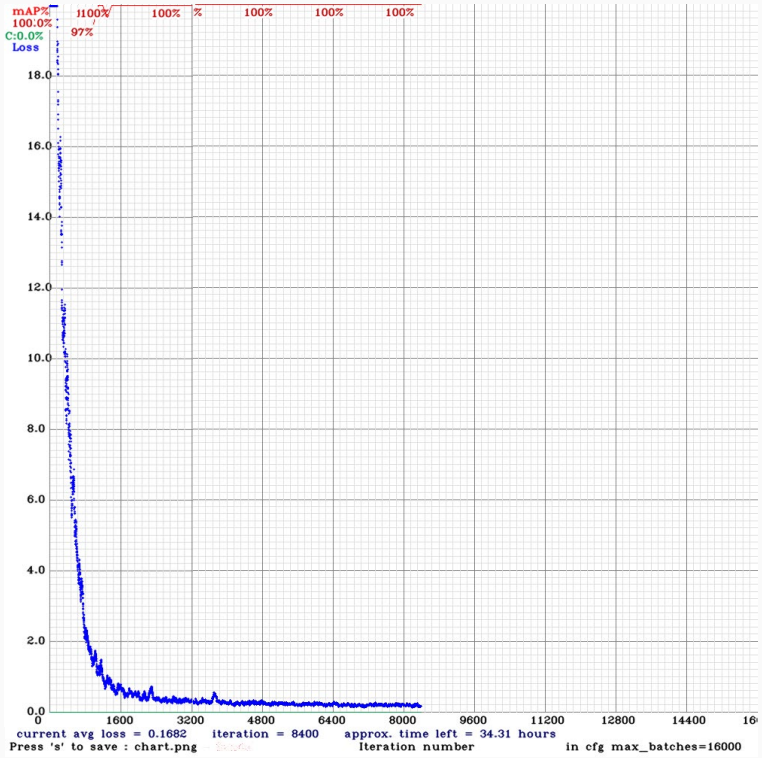
06.
結果呈現



訓練結果

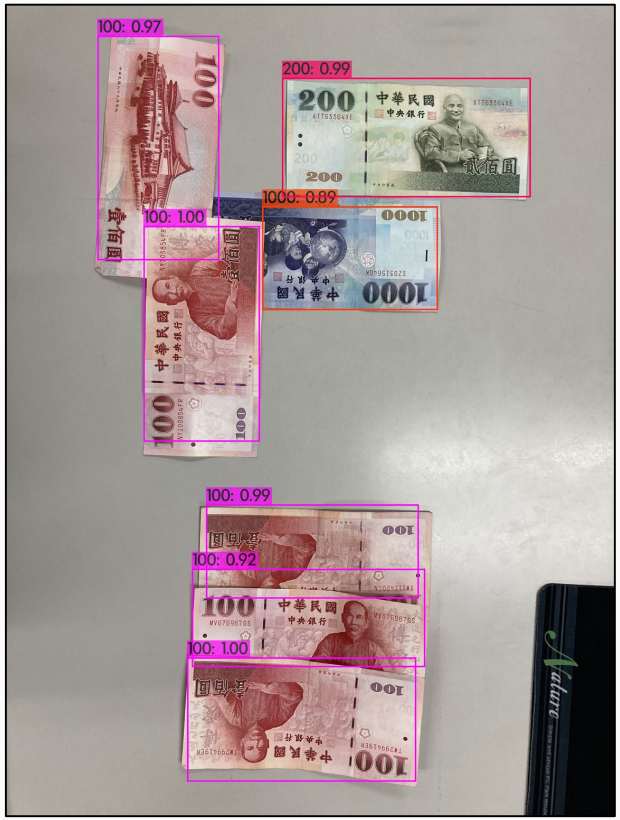
- 訓練8400個iteration
- 平均loss為0.1682
- 驗證集mAP為100%
- 共耗時40小時左右訓練

```
Total params: 64,020,145
Trainable params: 63,953,841
Non-trainable params: 66,304
```



辨識優良範本

成果呈現


100: 0.97
200: 0.99
1000: 0.89
100: 1.00
100: 0.99
100: 0.92
100: 1.00

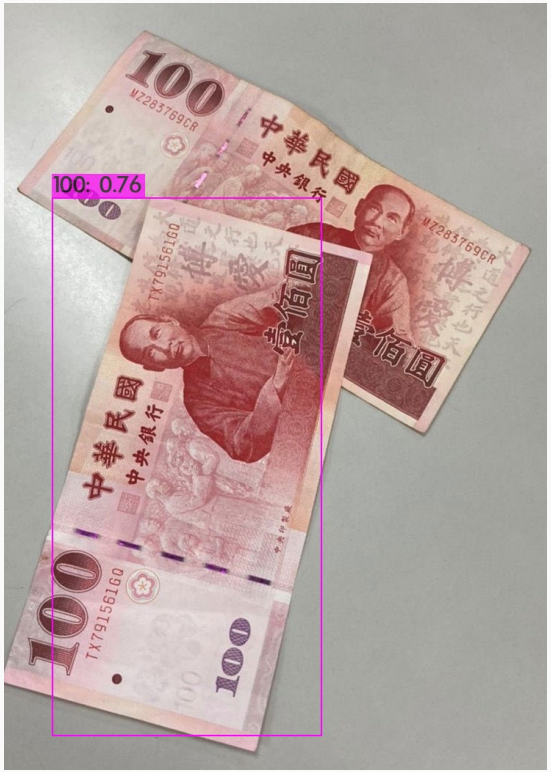

100: 0.98
100: 0.85
100: 0.99


1: 1.00
50: 1.00
10: 1.00
5: 1.00

成果呈現

辨識缺點:

1放置斜的重疊鈔票
2.髒污硬幣
3.硬幣在畫面中過小



結論

研究限制

- **訓練數不足**，本次訓練共用了151張訓練集、38張驗證集，但較完善之模型訓練張數應為上千張。
- **設備限制**，因顯卡內存不夠大，訓練速度慢，故每一次嘗試模型需花上半天測試。
- **樣本變異**，因錢幣紙鈔新舊、髒污程度不同造成訓練誤差。

未來展望

- **提高訓練數**，增加更多角度更多款式之訓練數，讓模型可適應各種錢幣。
- **即時辨識**，修改程式使其可串接手機或穿戴式裝置即時辨識，提供給店家，甚至是視力不佳需要輔助工具等人士。
- **即時運算**，透過辨識框加總所有辨識框之標籤，可即時算出該圖片內含有多少錢。

referance

- YOLOv4作者神AlexeyAB https://github.com/AlexeyAB/darknet
- 深入淺出Yolov3和Yolov4https://kknews.cc/zh-tw/tech/3yzxlza.html
- YOLOv4建置流程https://wings890109.pixnet.net/blog/post/68926387-yolov4%e5%bb%ba%e7%bd%ae%e6%b5%81%e7%a8%8b

THANKS

Does anyone have any questions?