Digital Image Processing Final Project

P76104231 蔣有為

Problem :

Problem A : Detection the location and Classification of Scaphoid of hand from X-ray Image.

Problem B : Classify if the Scaphoid of Image was fracture or not and Detection the location of the Scaphoid.

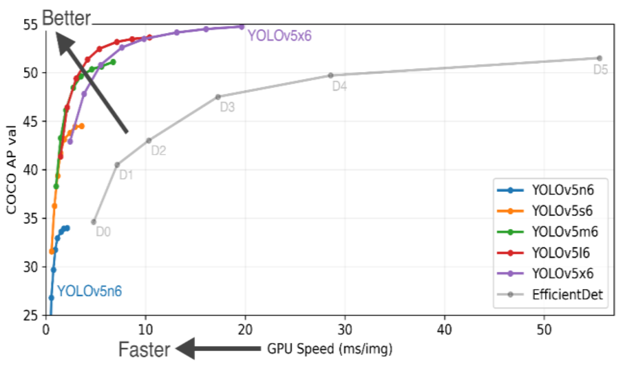
Method :

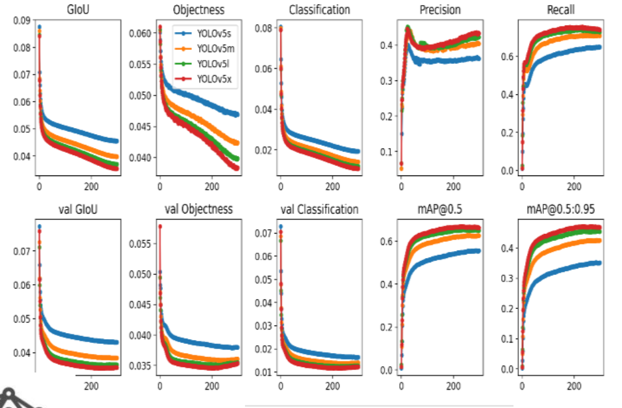
Environment requirement : Python >= 3.7.9, including PyTorch>=1.9

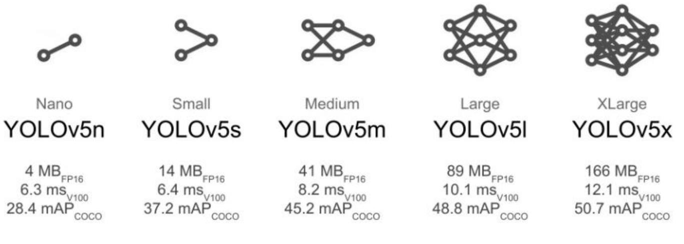
使用了yolov5進行模型訓練。Shortly after the release of YOLOv4 Glenn Jocher introduced **YOLOv5** using the Pytorch framework.

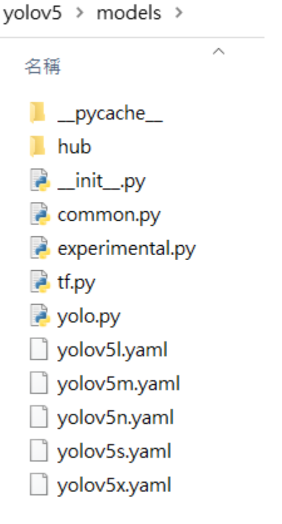
The open source code is available on [GitHub]

Model Selection









Dataset

normal

fracture

Evaluation : 3 fold cross-validation

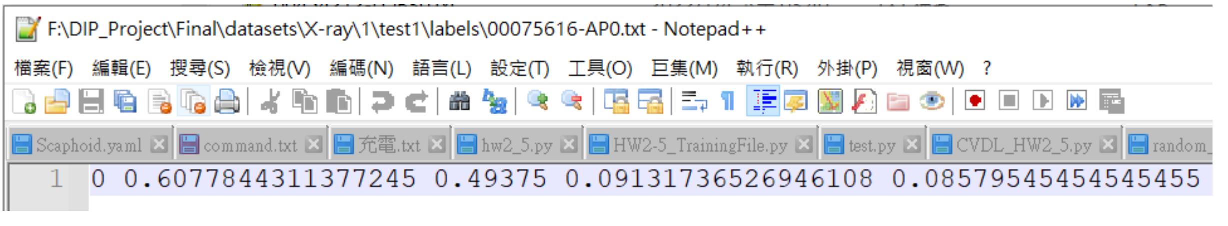
Dataset :

```
Source X-ray images
    | Normal   ---- 120 images
    | Fracture ---- 120 images
After split (8:2) |
|--- 1 set (0~33%)
    | Train (66%)
              | Normal   --- 80 images
             | Fracture --- 80 images
    | Test  (33%)
              | Normal   --- 40 images (00169382 L 51F AP0 (1)~ 08838278-OB30 (40))
             | Fracture --- 40 images (00075616-AP0(1) ~ 08092913-SC0(40))
| --- 2 set (33~66%)
        | Train (66%)
| Test  (33%)
        | Normal   --- 40 images (08838278-SC0 (41) ~ 18059482-AP0 (80))
        | Fracture --- 40 images (08187768-SC20 (41) ~ 18345938-SC20 (80))
    | --- 3 set (33~66%)
        | Train (66%)
        | Test  (33%)
              | Normal   --- 40 images (18060182-OB0 (81) ~ 182504260 L 28F AP0 (120))
            | Fracture --- 40 images (18389767-OB0 (81) ~ 20153305-AP0 (120))
```

Data Labels

每一張圖片對應到的舟骨位置資料轉換成yoloV5的label input形式

Class, x_center, y_center, width, height



Data classes

```yaml
# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to,
path: ../datasets/X-ray/1  # dataset root dir
train: train1  # train images (relative to 'path') 128 images
val: test1  # val images (relative to 'path') 128 images
test:  # test images (optional)
# Classes
nc: 1  # number of classes
names: ['Scaphoid']  # class names
```

Model Evaluation :

Data Augmentation 資料增強

Data Image

0075616-AP0.bmp

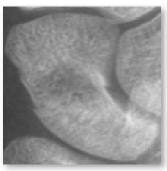
00169382 L 51F AP0.bmp

00230304 R 50F AP0.bmp

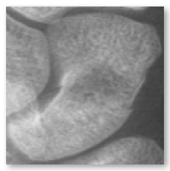
00293717 L 77M AP0.bmp

00454212-LOB0.b
p

加入Data Augmentation功能增加資料集的數量，提昇其準確率

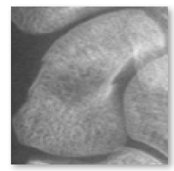


00454212-LOB3 0.bmp

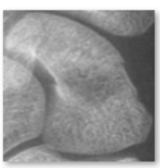


00454212-LOB3 0_h.bmp

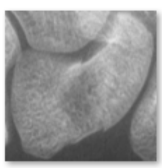
00454212-LOB3 0_h_r90.bmp



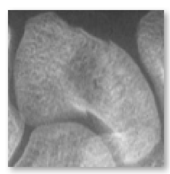
00454212-LOB3 0_h_r180.bmp



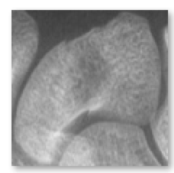
00454212-LOB3 0_r180.bmp



00454212-LOB3 0_r270.bmp



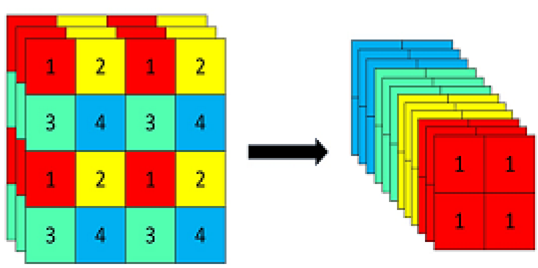
00454212-LOB3 0_h_r270.bmp



00454212-LOB3 0_r90.bmp

Yolov5 Focus:

分別為 l、m、s、x，主要差別在於depth_multiple 和 width_multiple，分別代表控制網路的深度和寬度的因子，前者深度主要是控制 BottleneckCSP 的數量，後者寬度則是控制卷積核數量。

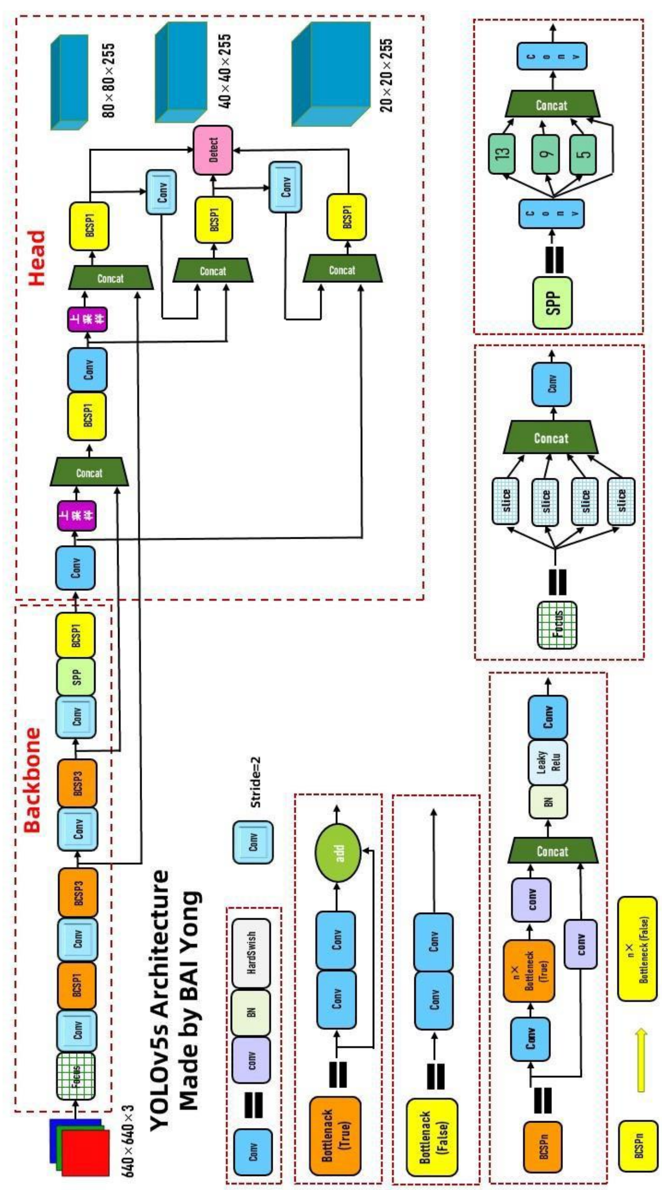


從上面架構圖看到，640 x 640 x 3 輸入，在 YOLOv5 中是圖片進入 Backbone 前，對圖片進行切片操作，將圖像相鄰的四個位置進行堆疊，類似於鄰近下採樣，拿到四張圖片，再將四張圖片互補和將 W 和 H 集中到了通道，輸入成原本的 4 倍，即拼接起來的圖片相對於原先的 RGB 三通道模式變成了12 個通道，最後將得到的新圖片再經過卷積操作，最終得到了沒有資料丟失情況下的二倍下採樣特徵圖。

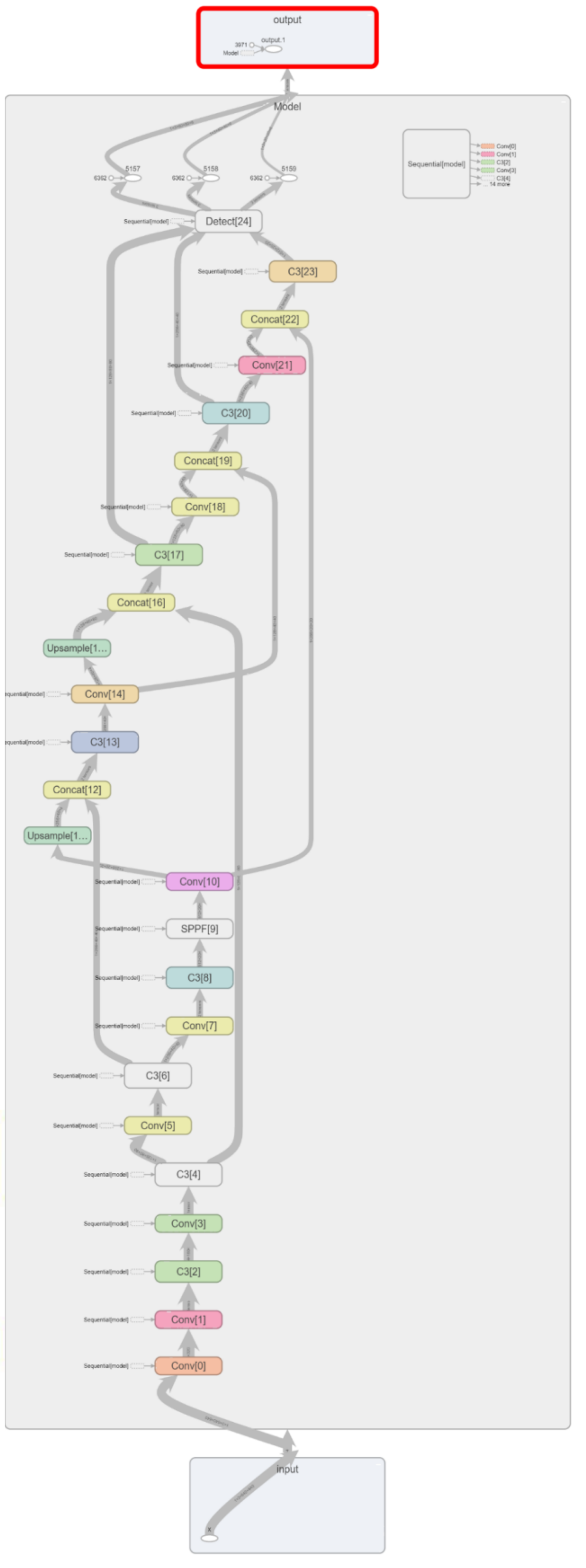
YOLOv5 架構圖

Yolov5 優點

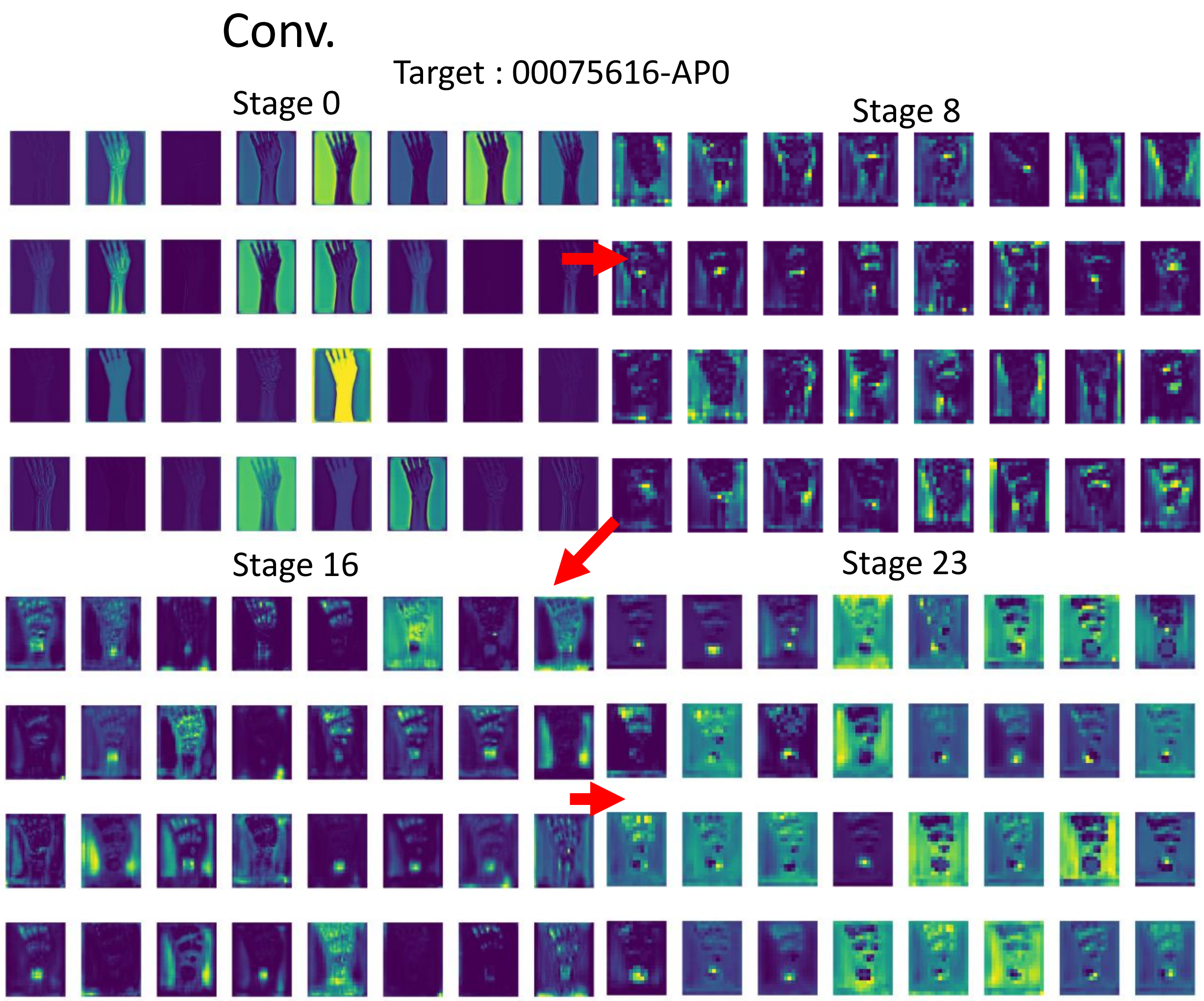
（1）Mosaic資料增強
（2）自適應錨框計算
（3）自適應圖片縮放



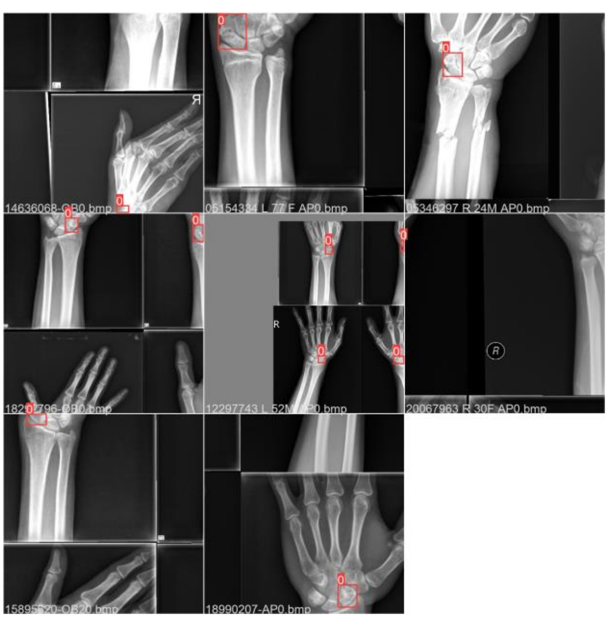
Model Graphs



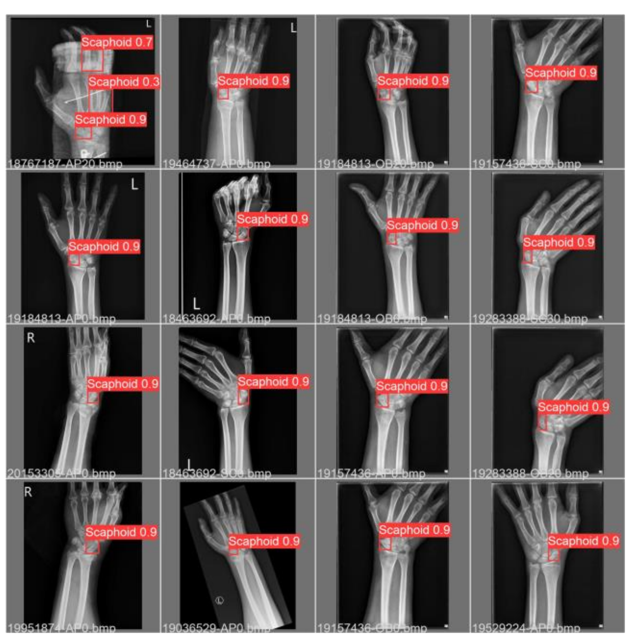
Detection Scaphoid of X-ray Image:



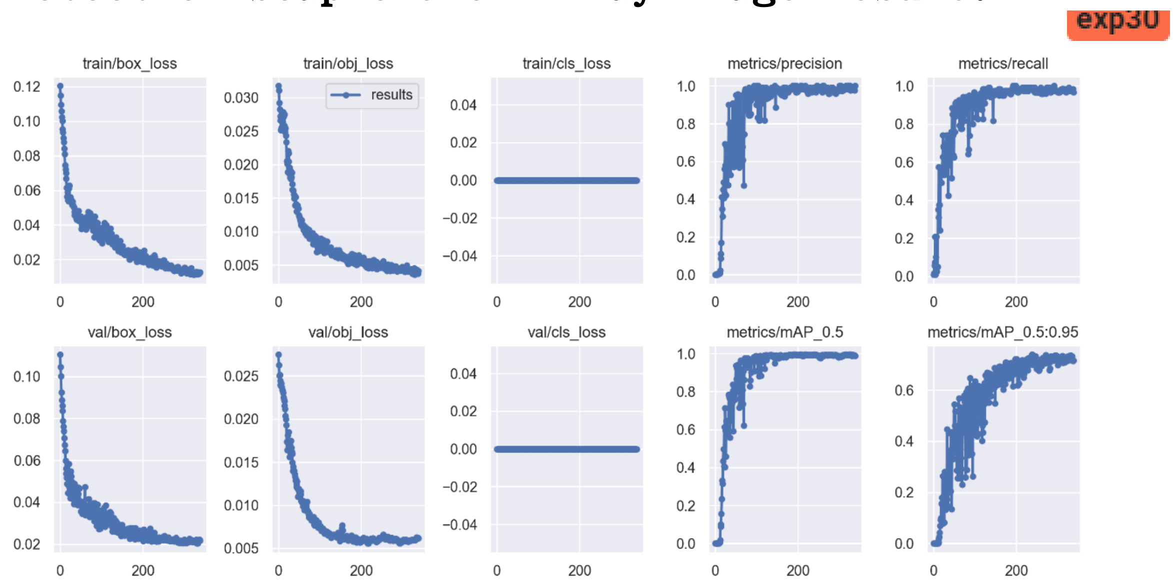
train_batch0

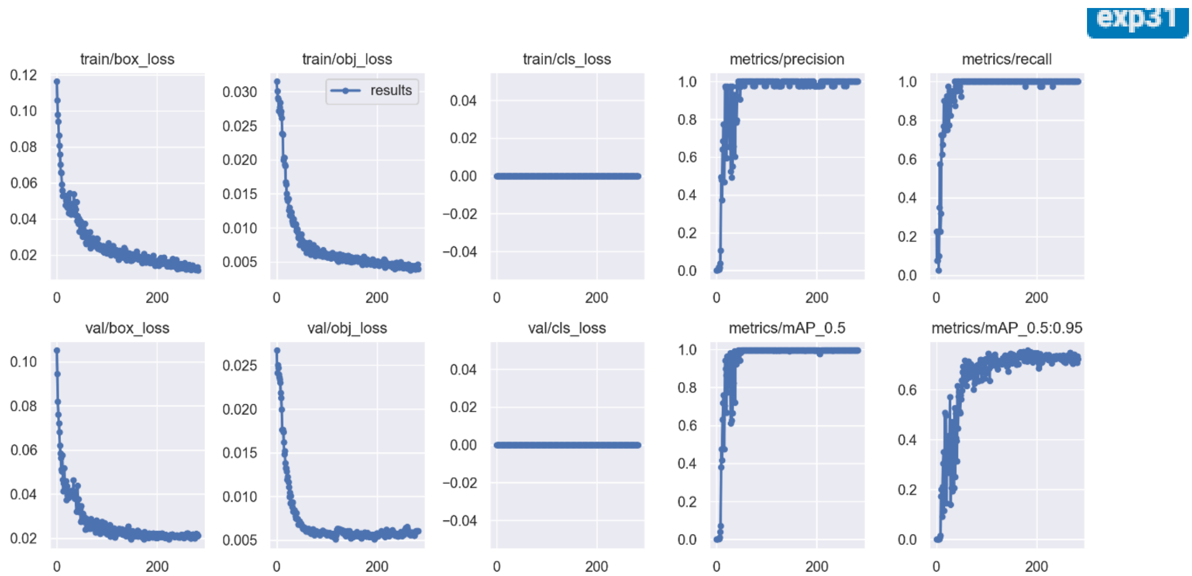


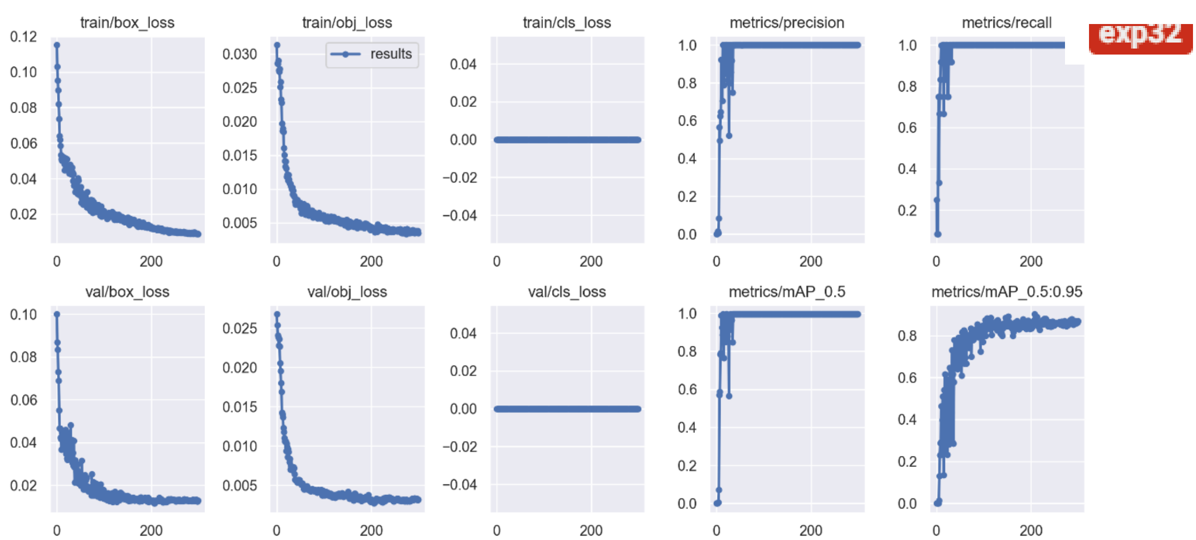
val_batch0



Detection Scaphoid of X-ray Image Result:



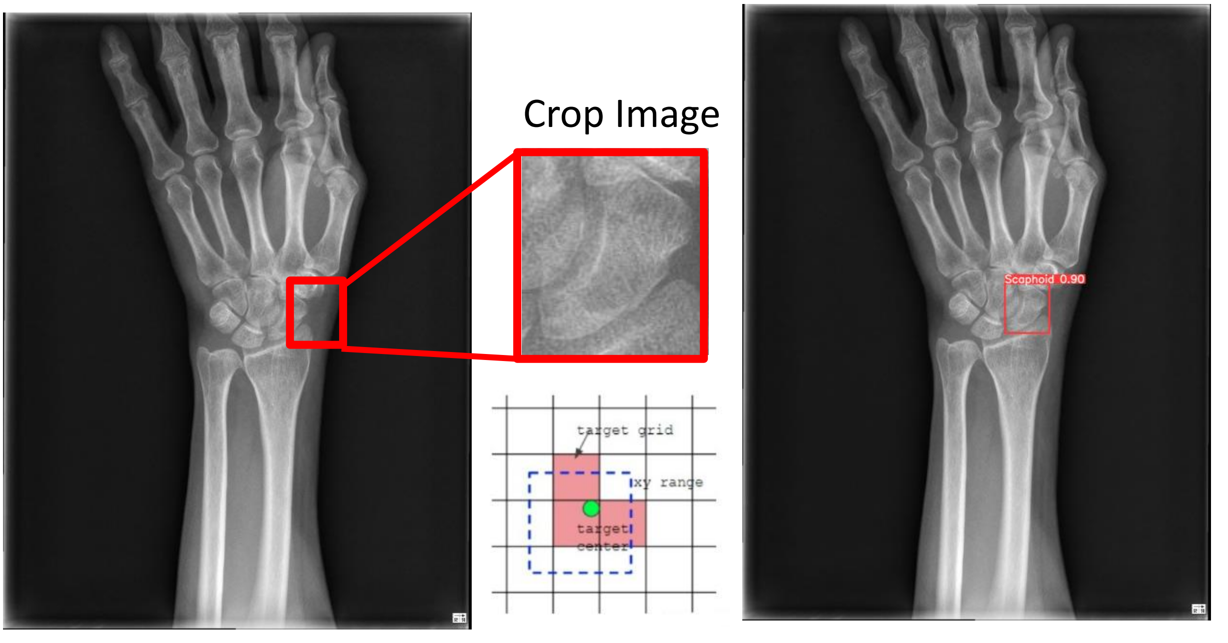




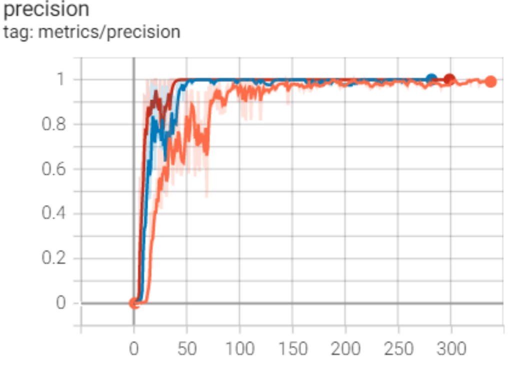
Detection

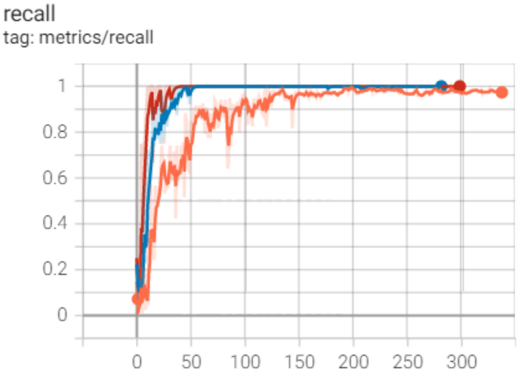
Source Target : 00075616-AP0

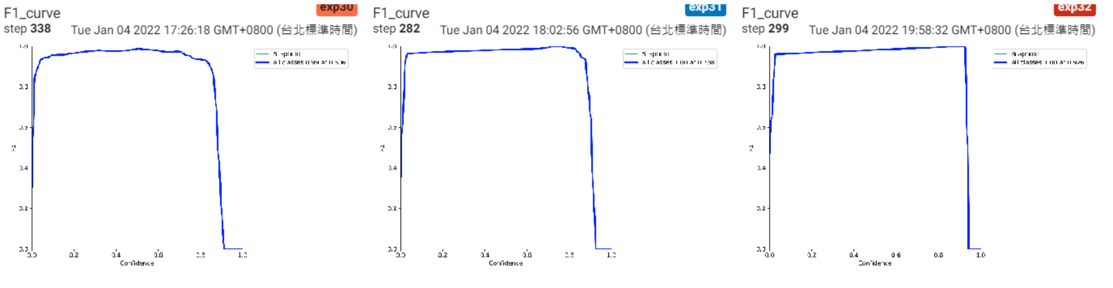
Predict Result : 00075616-AP0



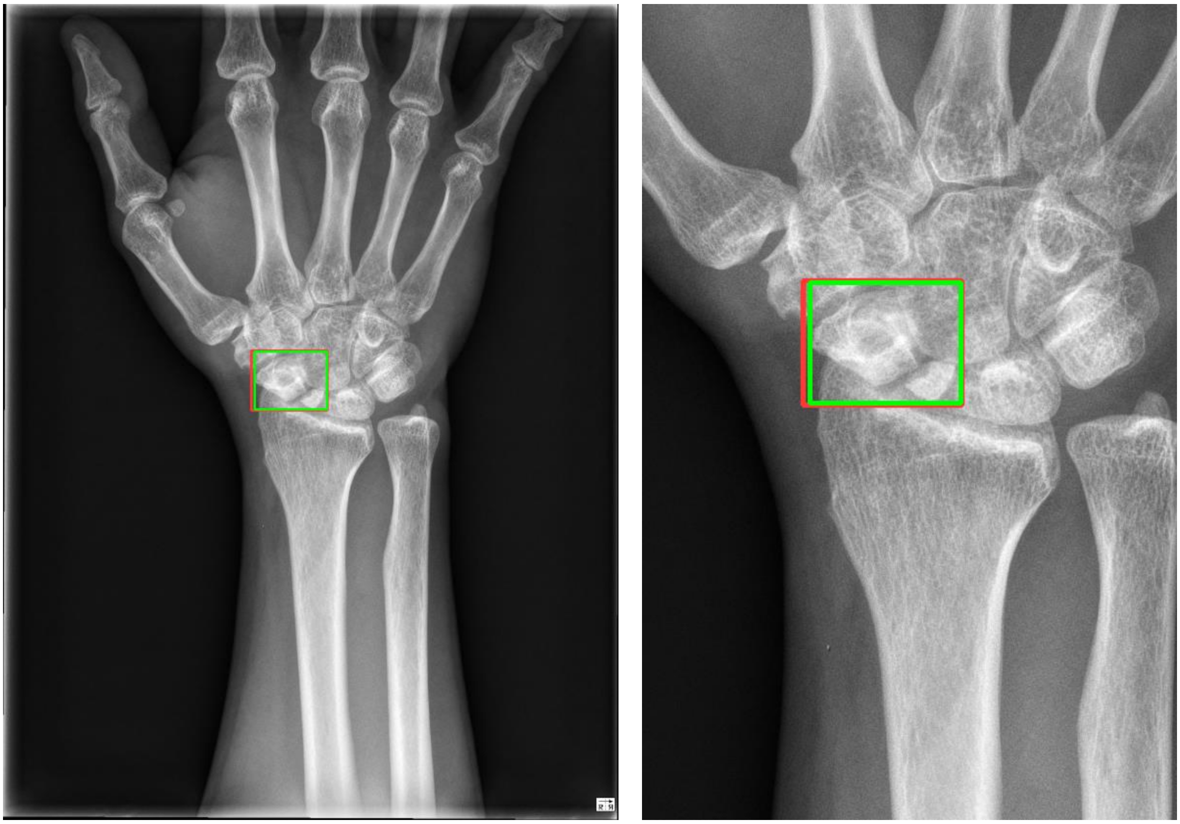
Detection Scaphoid of X-ray Image:







Detect with Ground Truth



Classification

Form : https://github.com/ultralytics/yolov5/tree/classifier

一開始使用ResNet-32、ResNet-50、VGG-16等模型，但效果都差強人意，準確度一直無法好好上升，最後使用yolov5的改良模型作為分類的模型，但是效果沒有到很好。針對物件檢測上，可以利用Yolov5其特性，以物件是否有偵測到做為分類的依據，準確率能夠提升到9成。

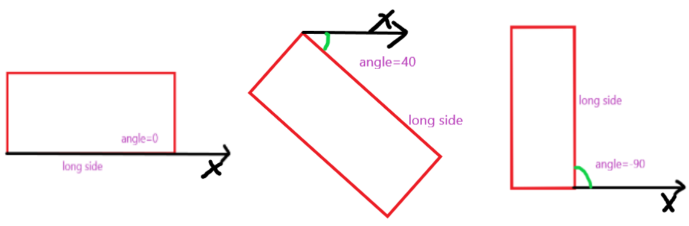
Optimizer:
使用sgd, Adam, AdamW, 等

效果較差的主要原因是：
（1）小目標尺寸
以輸入的圖片大小(150,150)，在Yolov5中下取樣都使用了5次，因此最後的特徵圖大小是19×19，38×38，76×76。
（2）顯示卡計算能力
當如果想要提高準確度，提高batchsize後，GPU的計算負擔會提升，甚至好幾次無法正常執行。
轉換想法:
（1） 增加計算量
比如原本240張的影像，可以經過旋轉產生不同的圖片。

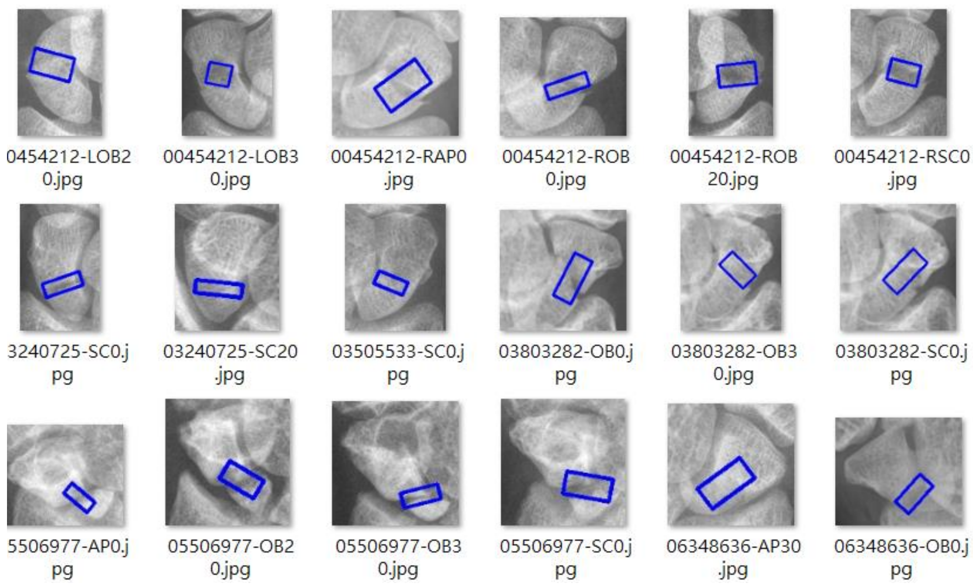
Detection fracture of Scaphoid was cropped :

採用的模型為rotation-yolov5，一開始使用fast-rcnn，但結果並不好。Base on yolov5 format, it define the box label is (cls, c_x, c_y, Longest side，short side, angle)。

(Tip: Range for angle is [-90，90), so wo should add 90 in angle while make your dataset label and then your label's Range should be [0,179))

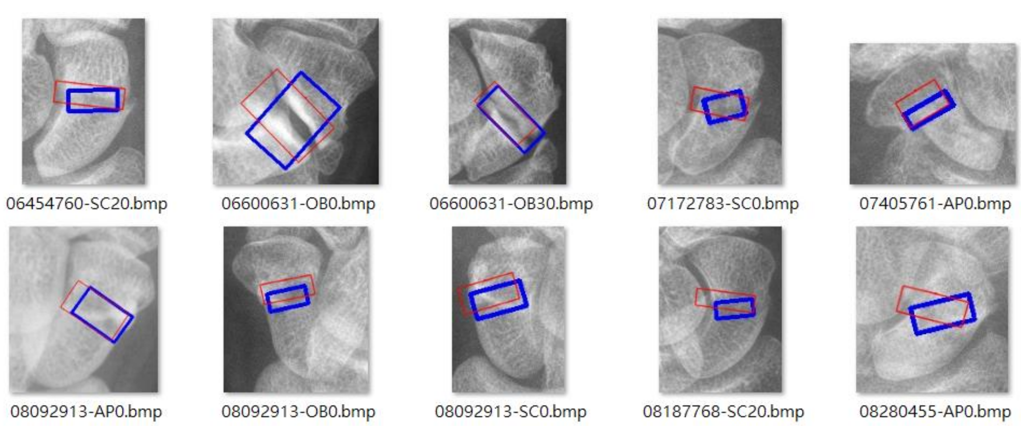


Detection fracture of Scaphoid was cropped :

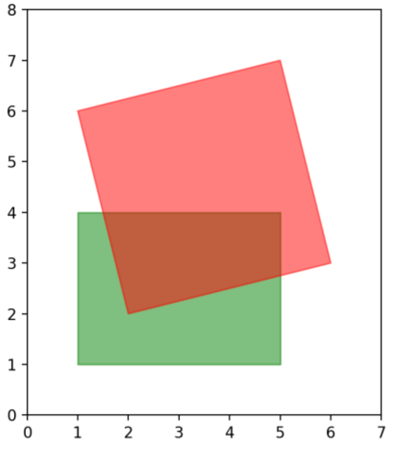


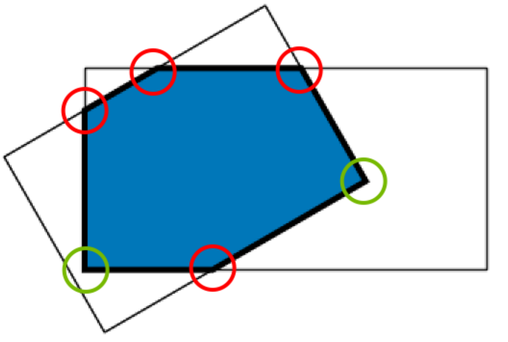
Fracture of Scaphoid with Ground Truth:

Method : 提高conf_thres 信心度，只標記可能性高的地方，避免偵測重疊。



Calculating IoU between rotated boxes





Intersect over Union Threshold，IOU值：

預測框大小∩真實框大小 預測框大小∪真實框大小。預測框與真實框的交集與並集的取值。

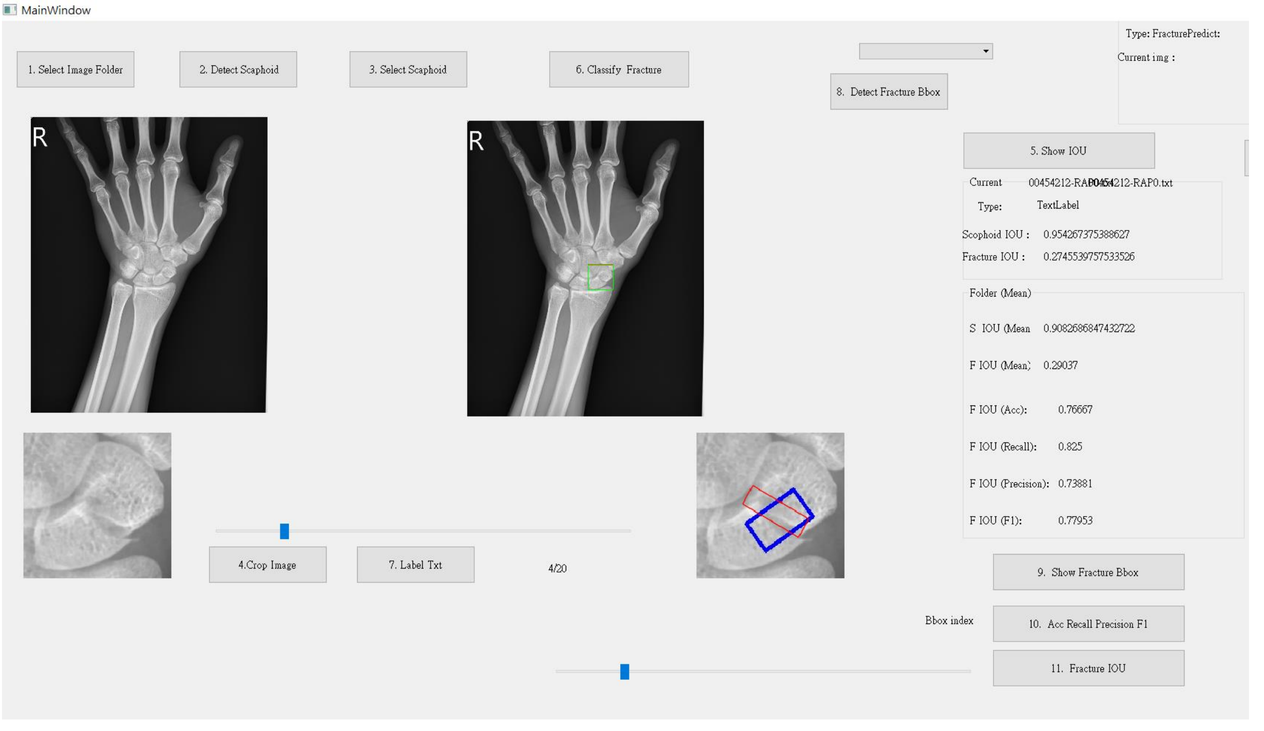
這部分也是較難的部分，首先要得到Ground Truth和偵測後的Bounding Box 的 中心座標，和長度、寬度，還有角度。

計算 IoU 的部分是利用兩個矩形經過角度的轉換後，在計算各自重疊的區域，從先前檢測座標和矩形的數據可以與Ground Truth 進行計算，得到旋轉後的矩形重疊比例(IOU)。

<u>Intersection over Union for rotated rectangles</u>

```
fracture_predict_box  (62, 103, 81, 42, -6.965478008538062)
fracture_Gt_box  (60, 97, 81, 40, 2)
iou  0.339240646360935
0.339240646360935
['txt', '0', '90', '96', '54', '38', '0.610865', '0.653167\n']
fracture_predict_box  (90, 96, 54, 38, 31.419297601872824)
fracture_Gt_box  (85, 85, 87, 28, 23)
iou  0.4364860471050962
0.4364860471050962
iou_mean :  0.39128
Fracture_IOU Done
```

GUI :



Result :

Scaphoid Accuracy :	1.0	Scaphoid precision :	1.0	Scaphoid recall :	1.0	Scaphoid f1 :	1.0
Fracture Accuracy :	0.76667	Fracture precision :	0.73881	Fracture recall :	0.825	Fracture f1 :	0.77953

Conclusion :

因為這個作業要使用到Object Detection和Classification的Model，首先一開始使用ResNet與Faster RCNN，但是在於靈活性和準確性上說，我選擇Yolov5來做模型的訓練和預測。這次的作業有一定的難度，所以要進一步的去了解訓練模型的各種細節，而且判斷骨折和骨折位置的偵測又比單純的舟骨位置偵測還要困難很多，所以在這個作業學習到了很多。

由於這次選用的是Yolov5，經過改良的架構，使得其速度與精度都得到了極大的性能提升，並且使用起來更方便靈活，透過不同參數設定可以得到不同複雜度的模型。

Reference

1. Python 執行另一個 Python 指令碼
2. 深度學習中的batch的大小對學習效果有何影響
3. https://www.zhihu.com/question/32673260
4. https://stackoverflow.com/questions/18207181/opencv-python-draw-minarearect-rotatedrect-not-implemented
5. https://github.com/BossZard/rotation-yolov5
6. https://github.com/ultralytics/yolov5