

HW3 Report

姓名:陳鎧勳

學號:0716314

0. GitHub link of my code:

https://github.com/KaivinC/CV_HW3

1. Reference:

MaskRCNN:

<https://arxiv.org/abs/1703.06870#:~:text=The%20method%2C%20called%20Mask%20RCNN%2C%20running%20at%205%20fps.>

Detectron2:

<https://github.com/facebookresearch/detectron2>

2. Brief introduction:

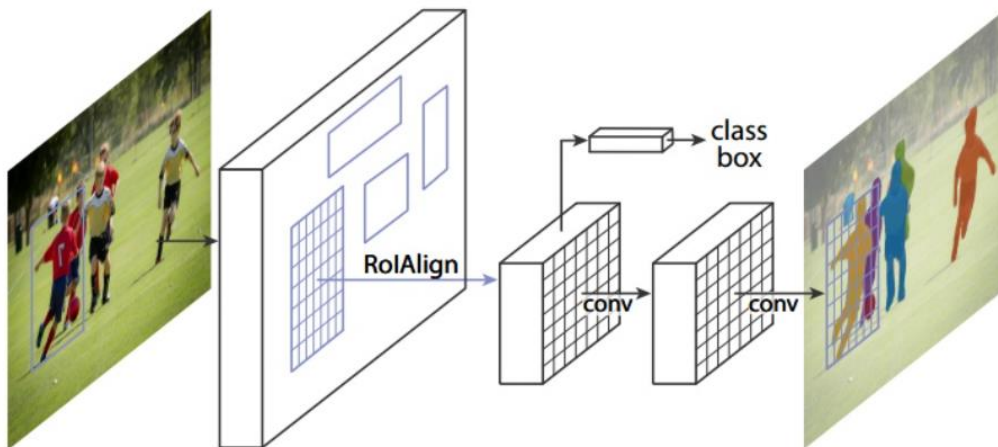
這個作業我所使用的技術是參考了 MaskRCNN 這篇論文裡的方法，裡面使用的 FPN(feature pyramid network)讓大物體或小物體皆可以偵測的到，最後實作是用 Detectron2 這個開源的 API。

3. Methodology:

1. Data pre-process:

首先利用助教給的 pycocotools 去讀檔案，然後我會用自己寫的"Cal.py"算出整個 Dataset 的長寬比，然後決定要怎麼 resize 整個 Dataset 的图片。

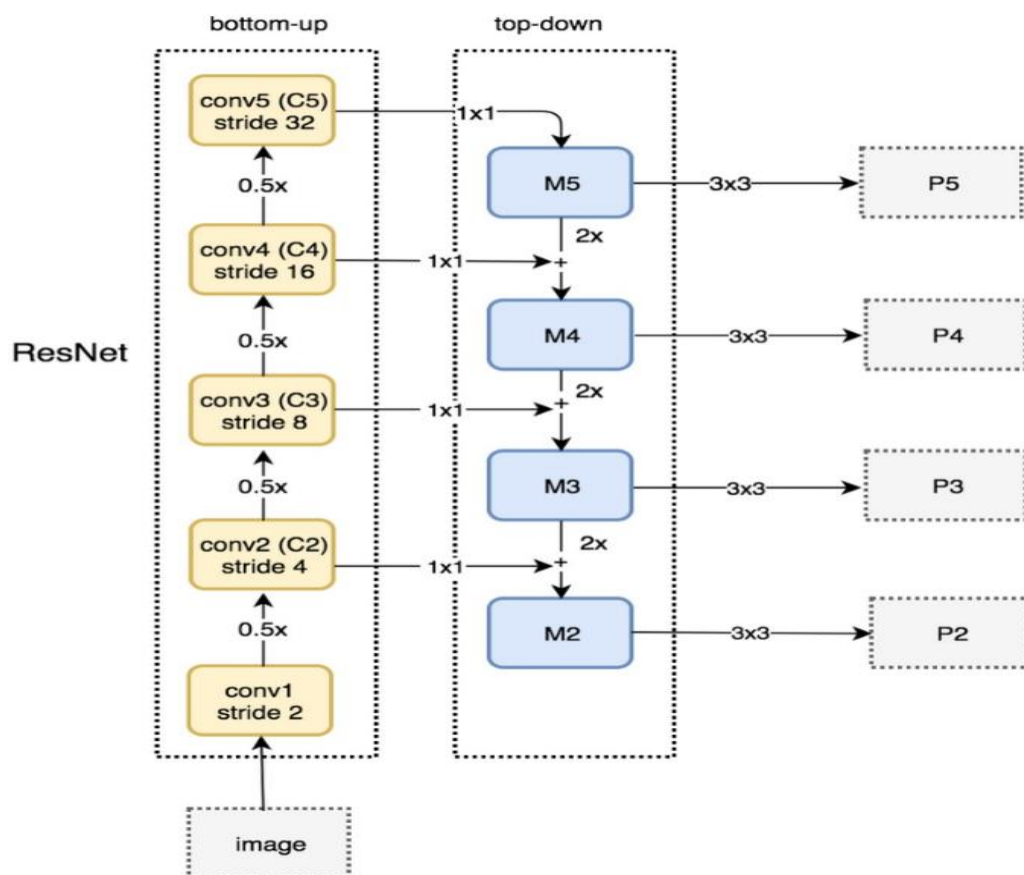
2. Model architecture:



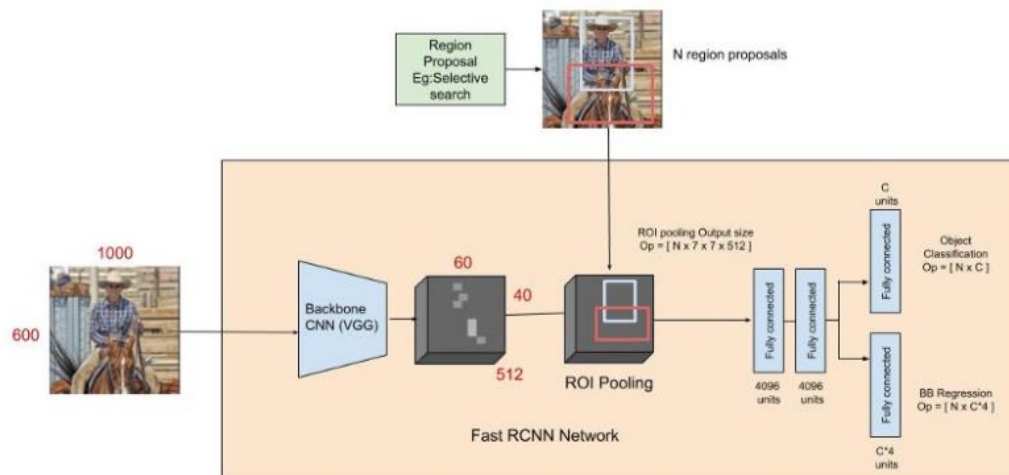
我這次使用的是 maskRCNN，maskRCNN 實際上是由三個概念組成，分別是 ResNet-FPN、Fast RCNN 和 mask。

這邊先說明 FPN + Fast RCNN 的概念

首先是 FPN 的部分，FPN 全名是 feature pyramid network，而它是一個多尺度檢測的網路，結構主要由“由上至下”、“由下至上”和“橫向連結”組成的(如下圖)，這麼做的目的可以讓我們的任何大小的物件都偵測得到，而且在各層級的特徵 fuse 更可以使其有豐富的語意和空間信息。



而 Faster RCNN 則是兩階段的目標檢測法，分別為階段一的 Region proposal 和階段二的 bounding box。Faster RCNN 利用 Resxnet 提取圖像的特徵，然後使用 RPN 去提取出 ROI，接著使用 ROI pooling 將這些 ROI 全部變成固定大小，最後再給接下來的全連接層進行 Bounding box 回歸和分類預測



最後壓軸就是把上面提到的兩個概念再加上 mask 就變成 mask RCNN 了，而 mask RCNN 其實就是把 faster RCNN 裡的 ROI pooling 改成 ROI align 以及最後加上個 mask 分支。

為甚麼要把 ROI pooling 改成 ROI align 呢?因為我們的候選框和最開始回歸出來的位置有一定的偏差(因為兩次整數化的原因)，因此 mask RCNN 就想出了用雙線性插植的方法，重新設計 ROI pooling，讓最後的結果是浮點數，這樣就不會因為整數化而捨去其小數點後的位數導致候選框偏差了。

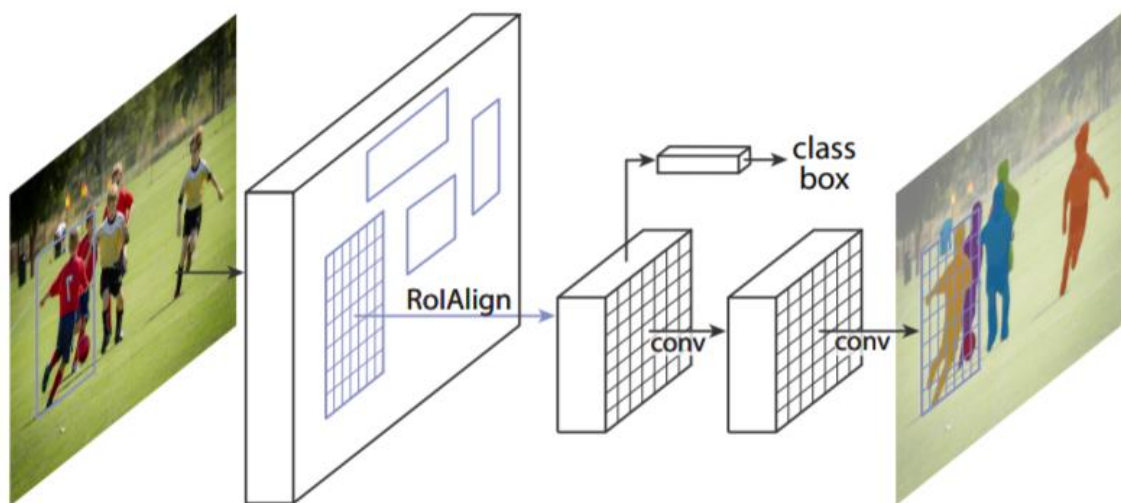


Figure 1. The **Mask R-CNN** framework for instance segmentation.

3. Hyperparameter:

本次作業我的 optimizer 是使用 SGD，weight decay 是設定成 10^{-2} ，然後 learning rate 則是利用 cosine annealing 做一個 $1e-3 \sim 1e-10$ 的周期性切換 epoch 總數為 $(100000/8)$ 。

Data augmentation 則是把圖都 resize 成 800×1000 接著 randomly random crop 成 $size \times 0.7$ 的大小，還有隨機調整圖片的飽和度、對比度、以及亮暗還有水平翻轉。

4. Summary:

這次的作業最難的地方就是在於資料太少，導致模型很容易就 overfitting，本來沒有 augmentation 時大概到一半就 overfitting 了，後來增加了 augmentation 有撐了比較久才 overfitting，現今機器學習最大的阻礙就莫過於 data 數量的不足了，其實如果有時間我還蠻想用用看 unsupervised 的 model 來 train 看看。