Hw2 Report

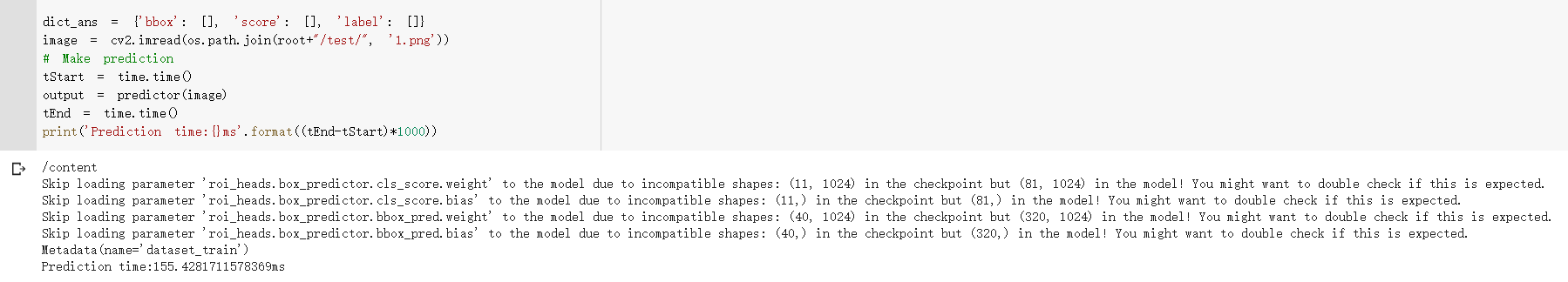
0756545 楊秉澄

Github link: <https://github.com/BingChengYang/cv_hw2>

Reference: https://github.com/facebookresearch/detectron2

Speed Bench Mark:

在Colab上辨識一張圖片的時間為155ms。



Introduction:

這次作業的task為digit上的object detection，所使用的dataset為Street View House Number(SVHN)裡面有33402張training data和13068張testing data，其中在training data裡有70000多個annotations。

這次作業不僅僅要滿足accuracy 0.36898的Baseline還要滿足558ms辨識一張圖片，因此模型的選擇上要可以兼顧上面2項條件，我透過detectron2這個由facebook基於pytorch開發的工具分別嘗試faster rcnn、retinanet 2種網路架構。

Methodology:

Environment:

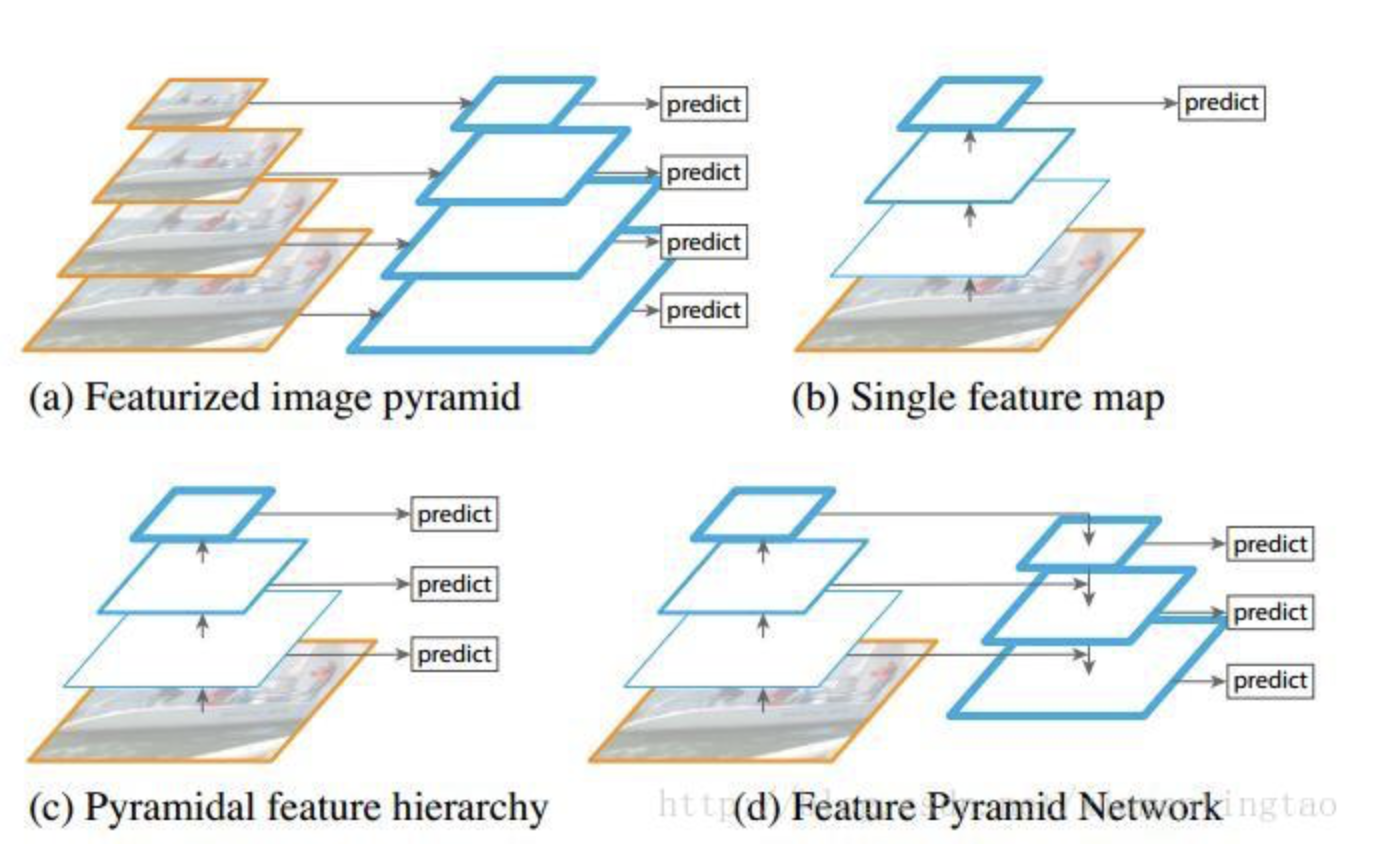
* CPU: AMD R5-2600
* RAM: 16G
* GPU: Nvidia RTX-2070 8G

Data Preprocessing:

由於我這次採用detectron2來實作object detection，為了需要將其annotations轉換成coco dataset的格式，除此之外我沒有特別對data做任何處理。

Model Architecture:

Model我使用detectron2內建的2種模型，分別是fpn faster rcnn、retinanet，最終是選擇fpn faster rcnn當作我的模型，我用resnet50當作我的backbone雖然resnet101的效果不差，但為了有更快的辨識速度，因此我還是選擇resnet50，採用fpn是為了避免小偵測目標在最後一層的時候特徵都已經遺失了，所以採用feature pyramid network可以有效解決多尺度的檢測問題。



Hyperparameters:

參數分別有learning rate(lr)、mini\_batch\_size、Max\_iter需要設置，lr我試過0.0001,0.00015,0.001,0.000001，其中0.001太大很容易讓loss變成NaN，0.0001和0.00015訓練起來效果差不多，但0.00015訓練出來的模型有通過baseline，因此最後我使用0.00015訓練我的模型，0.000001收斂太慢因此不採用。

Mini\_batch\_size嘗試了32、64、128其中3者訓練起來效果差不多，但32訓練出來的mAP稍微好一點點，因此用32來訓練我最終的模型。

Max\_iter設定為5000，跑在1000以下訓練出來的效果非常不好，因此最少需要設定在1000以上。

Summary

這次作業為了設置detectron2在window10上的運行花費了我很多時間，所幸最後還是有訓練出一個可以通過baseline的模型，且在速度上也有不錯的表現，可惜detectron2沒有內建yolo的使用，因此之後我想嘗試yolo並比較他與faster rcnn看誰的效果比較佳。