繁體中文場景文字辨識競賽－

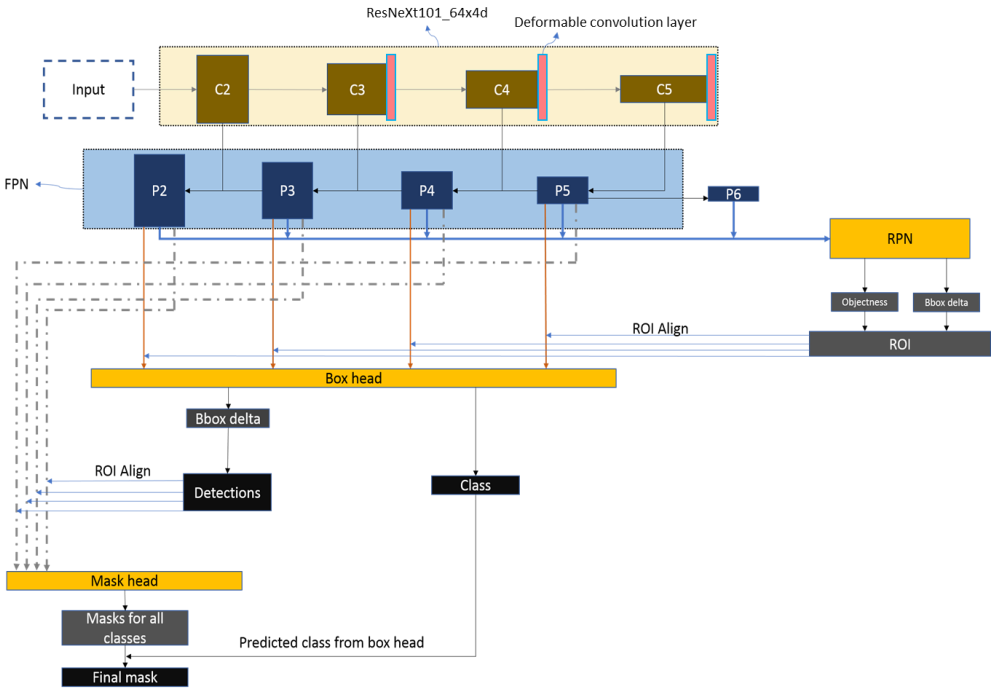
初階：場景文字檢測

壹、 環境

程式運行於 Linux 作業系統，語言使用 Python 3.7.10，全程模型訓練與驗證環境皆透過 Google colab。我們使用 open-mmlab 開發的基於 Pytorch 與 mmdetection 的開源文本檢測與辨識之工具庫 MMOCR(https://github.com/open-mmlab/mmocr)，所需套件包括 torch 1.5.0、torchvision 0.6.0、cudatoolkit 10.1、mmcv-full 1.3.4、mmdet 2.11.0、numpy、scipy、pandas、opencv-python、scikit-image、glob、os、json、matplotlib、warnings。我們使用 Mask R-CNN 當作主要模型，其中特徵提取的 backbone 部份使用 open-mmlab 的預訓練 ResNeXt101_64x4d 模型(來源: https://github.com/open-mmlab/mmcv/blob/master/mmcv/model_zoo/open_mmlab.json)，訓練過程並未使用額外資料集。

貳、 演算方法與模型架構

我們選擇 Mask R-CNN 模型，其架構包括在第 3 到 5 stage 使用了 Deformable convolution 的 ResNeXt101_64x4d、Feature Pyramid Network(FPN)、Region Proposal Network(RPN)、RoI Align、Box Head 以及 Mask Head，如下圖所示:



Mask R-CNN 透過 ResNeXt101_64x4d 與 FPN 提取多尺度影像特徵，5 種不同尺度的特徵(P2~P6)會被輸入至 RPN 網路中獲得許多候選框，由於這些候選框來自於不同尺度特徵的觀察，因此很適合這次街景文字偵測任務，許多街景的文字都是由近至遠，慢慢減小，導致影像中包含了許多大小不一的文字，如何讓模型能同時兼顧不同大小的文字偵測能力會是比賽的重點。有了候選框後，模型會根據候選框的大小選擇適合的特徵圖(P2~P5)，小的候選框就選比較底層的特徵(例如 P2)，因為空間的解析度較高，反之，則選擇較高層的特徵圖。RoI Align 被用來處理候選框大小不一的問題，因為網路並無法處理任意大小的特徵，所以 RoI Align 用了基於插值的方法來將這些大小不一的特徵轉化成相同大小的特徵。最後，特徵被輸入 Box Head 獲得 bounding box 座標與分類信心程度，而 Mask Head 基於 Box Head 的檢測框預測出物體的 Mask。

在實作細節上，ResNeXt101_64x4d 為預訓練權重(注意:Deformable convolution layer 的權重是被隨機初始化的)，並且我們在訓練期間凍結第 1 stage 的參數。由於大部分參數都與原始 Mask R-CNN[1]相同，這邊只指出針對文字偵測任務參數的不同之處，在 RPN 網路中的 anchor scale 被設為 8，aspect ratio 則為[0.5, 1.0, 2.0]。

此外，由於在 two-stage 檢測算法中，RPN 階段會生成大量的候選框，很多時候一張圖片絕大部分候選框都是沒有目標的(negative sample)，為了減少計算就需要進行 sampling，然而單純的 random sampling 選出來的框不一定是容易錯的框(hard negative sample)，因此我們使用 OHEM sampling 根據框的 loss 得到容易錯的框(hard negative sample)，來解決難易樣本以及正負樣本比例問題。

參、 資料處理

我們使用全部的訓練資料，並未刪減或增補，影像中的 RGB 通道被分別歸一化透過減去均值 mean=[123.675, 116.28, 103.53]以及除以方差 std=[58.395, 57.12, 57.375]。

由於 MMOCR 是基於針對 COCO dataset 物件偵測的 mmdetection 開源工具庫，因此資料需要被轉換至 COCO 的格式，具體格式如下圖:

```
{
    "images": [image],
    "annotations": [annotation],
    "categories": [category]
}


image = {
    "id": int,
    "width": int,
    "height": int,
    "file_name": str,
}

annotation = {
    "id": int,
    "image_id": int,
    "category_id": int,
    "segmentation": RLE or [polygon],
    "area": float,
    "bbox": [x,y,width,height],
    "iscrowd": 1,  // don't care
}

categories = [{
    "id": int,
    "name": str,
}]
```

COCO 格式的 Json 檔中會包含三個 Key 值(images, annotations, categories)，images 的 item 值為所有影像資訊組成的 list，包括 id, width, height, file name，而 annotations 的 item 值為所有檢測框資訊所組成的 list，包括檢測框的 id, 對應到的影像 id(image_id), 對應到的種類 id(category_id), mask polygon(segmentation), mask 面積(area), 檢測框的左上 x,y 以及寬高(bbox)，最後則是類別的資訊。

我們排除了所有 Ground truth 中被分類至 Don’t care 類別的檢測框，此外，考量到稀少的訓練資料量，若再將檢測框去做分類，可能會導致各個類別的檢測框數量不足以最佳化訓練網路，因此我們透過忽略檢測框的分類問題來將任務簡化，剩下 Ground truth 中的檢測框都在上述資料格式轉換後被歸類在同一類別。

肆、 訓練方式

考量到訓練資料集數量的限制，我們決定不另外再從中切出驗證資料集，透過提交結果至 public leaderboard 上來調適模型的參數與找出模型大約收斂的 epoch 位置。

訓練過程使用 Data Augmentation 來增加訓練資料的多樣性，首先圖片的長與寬被分別隨機縮放在[640, 2560]之間，接著圖片有 0.5 的機率被水平翻轉，最後隨機從圖形中裁切出長寬為 640*640 的影像，並且影像中會確保至少包含待偵測文字。

模型被訓練 100 epoch，每個 epoch 皆使用所有訓練資料集，由於 GPU 記憶體限制(12GB)batch size 被設置為 2，我們在訓練初期使用 warm up policy，learning rate 從 5*10^-6 在 500 個 iteration 後線性上升至初始的 learning rate=0.005，並且在第 50 與 80 epoch 被除以 10，根據 leaderboard 分數變化，我們選擇第 65 epoch 的訓練結果作為最後提交的模型。我們使用 Stochastic Gradient Descent 優化器，其中參數 momentum 與 weight_decay 分別被設為 0.9 和 0.0001。

模型在推理時，並未使用 Test Time Augmentation，且所有圖片被縮放至 1280*1280。由於模型輸出的檢測座標是根據預測的 mask 去計算最小能包圍 mask 的 bounding box，所以 pixel 的門檻會影響模型效果，我們將 pixel 的門檻訂為的 0.65。此外，模型也會輸出每個檢測框的信心程度，對此我們訂了 0.6 的門檻，所有信心程度小於 0.6 的檢測框都會被忽略。

伍、 分析與結論

我們一開始決定從以 ResNet50 為骨幹的 Mask R-CNN 出發，考量到資料集的大小，我們最初想透過 ICDAR 2019 Robust Reading Challenge on Reading Chinese Text on Signboard 中的簡體中文檢測資料來擴增數據集，但我們馬上就發現兩個數據集存在著 bounding box 評估上的差異，使用這次比賽的訓練集訓練出來的模型可以輕易達到 0.5 以上的分數，但是使用外部資料訓練的模型卻始終只能獲得 0.37~0.45 左右的分數，因此我們放棄使用外部資料集的做法，只專注在如何利用這次比賽中有限的訓練資料獲得更好的預測。

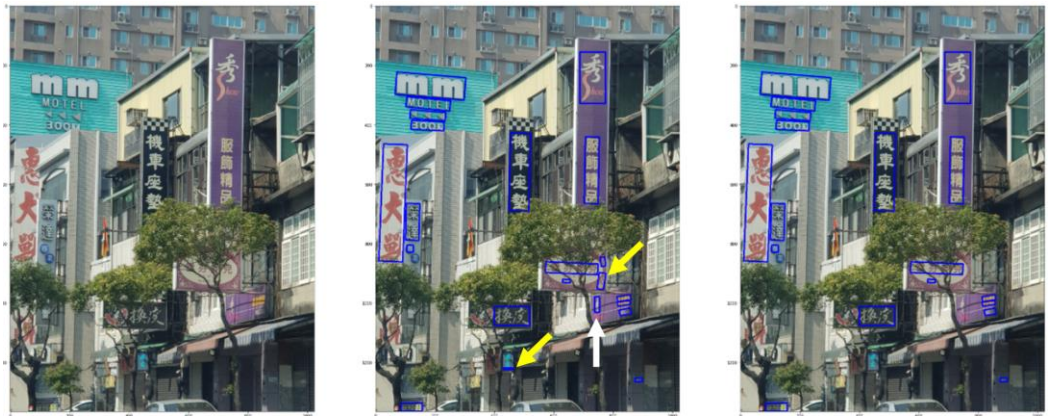
有鑑於過往 Scene Text Detection on ICDAR 2015 參賽者的經驗，檢測模型的訓練 epoch 數都不低，因此我們在一開始對 ResNet50 為骨幹的 Mask R-CNN 訓練了 159 epoch，由於當時還離 Public leaderboard 上最高的分數(約 0.65)，有一大段落差，所以沒有特別紀錄提交的結果是哪個 epoch 的，只知道最好

的結果分數為 0.6160510173。此外，我們也對推理時，模型應該被縮放至哪個一致的寬高作一些探索，最佳的寬與高大小約莫落在 1024~1365 左右，並且過大或過小的縮放皆會嚴重影響模型的效果，因此後續實驗我們將大小固定在 1280*1280。

根據過往經驗，將特徵提取器換至更優秀的模型是提升模型效果最簡單的途徑，因此我們將骨幹換成更好的 ResNeXt101_32x4d，考量到比賽時程與 colab 上 GPU 訓練速度，我們無法在必須使用更小 batch size 的情況如同第一個模型一樣訓練如此多的 epoch 數，對此，雖然正常情況下 initial learning rate 應該隨著 batch size 線性調整，當 batch size 縮小兩倍，learning rate 也應該縮小兩倍，但是在有限的資源與實驗下，為了加快訓練進程，根據第一個模型的經驗約 100 epoch 以後的效果變化就已經不明顯了，因此我們決定不將 learning rate 調小，此外 training scheme 也一併被往前調整。

ResNeXt101_32x4d 為骨幹的 Mask R-CNN 相比第一個模型提升了約 4%的效果，基於這個有利的證據，我們再次將模型的骨幹升級，使用了加入 Deformable convolution layer 的 ResNeXt101_64x4d 作為骨幹，並調整至正常的學習率，batch size 與 learning rate 一起線性縮小 4 倍，因為我們預估這是本次比賽能訓練到的最後一個模型了，最後根據 public leaderboard 分數的探索，我們選擇第 65 epoch 當作最後提交模型，相比第一個模型，分數提升了約 5％。

在比賽尾聲，我們在 Threshold 參數上作探索，包括輸出 mask 的 pixel threshold 與檢測框的信心程度 threshold。Pixel threshold 的上升能幫助模型進一步提升效果，簡單有效，但在調整參數的同時確有 overfitting public test data 的風險，由於此時提交次數只剩一次，因此雖然 Pixel threshold 在 0.65 與 0.7 的時候都有相似的效果，但我們傾向不要給予過高的門檻，所以最後選擇 0.65 的 pixel threshold 加上 0.6 的檢測框的信心程度 threshold 來嘗試，並執行最後一次提交，最終結果在 public leader board 上獲得 0.6890389978，private leader board 獲得 0.688366。下圖為提高檢測框的信心程度 threshold 後，產生的檢測框預測結果，依序為原圖、threshold=0.5 的結果以及 threshold=0.6 的結果，如黃色箭頭所指出，在 threshold=0.5 預測結果包含一些錯誤的檢測框；白色箭頭所指之處雖然為場景文字，但根據我們對 Ground Truth 標記基準的理解，這種不明顯且過於傾斜的文字並不會出現在 Ground Truth 標記中。



由於我們參與比賽的時間並不長，我們能做的實驗並不多，根據經驗最佳的檢測框的信心程度 threshold 應該還能在高一點。此外，一些更新穎的物件偵測網路像是 Cascade Mask R-CNN 與 Hybrid Task Cascade 都有機會能獲得不錯的結果，另外近幾年很常被使用在視覺與文本任務的注意力機制，我認為非常值得花時間投資，幾乎也是提升效果的保證，尤其是一些 Non-local 的注意力機制很適合用於這種街景影像，因為它的背景太過複雜了，基於全局的注意力機制在這項任務很有機會能對模型提供幫助。最後則是 Test Time Augmentation 與不同模型之間的 ensemble，都是未來能在提升效果的手段。

模型	Batch size	Initial learning rate	Training scheme	epoch	Public leaderboard 分數
以 Deformable ResNeXt101_64x4d 為骨幹的 Mask R-CNN	4	0.005	在第 50 與 80 epoch 後的 learning rate 被除以 10	57	0.6560071916
				60	0.6676300247
				65	0.6690983712
				72	0.667863541
				81	0.6671336771
				86	0.6654982906
以 ResNeXt101_32x4d 為骨幹的 Mask R-CNN	8	0.02	在第 50 與 80 epoch 後的 learning rate 被除以 10	64	0.6452265466
				76	0.6438919574
				80	0.6596055929
				84	0.6542822839
				96	0.6494559758

以 ResNet50 為骨幹的 Mask R-CNN	16	0.02	在第 80 與 128 epoch 後的 learning rate 被除以 10	-	0.6160510173

模型	Pixel threshold	Box confidence threshold	Public leaderboard 分數
以 Deformable ResNeXt101_64x4d 為骨幹的 Mask R-CNN	0.5	0.5	0.6690983712
	0.6	0.5	0.6773695366
	0.65	0.5	0.6803355424
	0.7	0.5	0.6805075773
	0.8	0.5	0.6771333067
	0.65	0.6	0.6890389978

陸、 程式碼

程式碼請另外附檔,包含資料處理、訓練流程、預測等相關程式碼。程式碼應附 README.md 檔案交代安裝配置環境，重要模塊輸出/輸入，以讓第三方用戶可以除錯、重新訓練與重現結果。

柒、 使用的外部資源與參考文獻

1. MMOCR Contributors. (2019). MMOCR: A Comprehensive Toolbox for Text Detection, Recognition and Understanding. https://github.com/open-mmlab/mmocr/tree/a86ab115e4fae0066e5a6a4167c27c3e233e7470
2. Machine-Vision Research Group. (2019, April 26). Mask R-CNN Unmasked. https://medium.com/@fractaldle/mask-r-cnn-unmasked-c029aa2f1296

聯絡資料

- **隊伍**

隊伍名稱	Private leaderboard 成績	Private leaderboard 名次
NE6084020	0.688366	3

- **隊員(隊長請填第一位)**

姓名(中英皆需填寫)	學校系所	電話	E-mail
涂登耀 (Deng-Yao Tu)	成功大學人工智慧碩士學位學程	0975150896	gercommand@gmail.com

- **指導教授**

若為「連結課程」的課堂作業或期末專題，請填授課教師，以利依連結課程彙整。

若非「連結課程」，但有教授實際參與指導，請填寫該位教授。

若以上兩者皆非，可不予填寫。

教授姓名	課程	課號	學校系所	E-mail
謝孫源			國立成功大學資訊工程學系	hsiehsy@mail.ncku.edu.tw
周信宏			國立暨南國際大學資訊工程學系	chouhh@mail.ncnu.edu.tw