

繁體中文場景文字辨識競賽一

進階：繁體中文場景文字辨識

報告說明文件

壹、環境

作業系統：Ubuntu 16.04.6 LTS

鏡像：pytorch/pytorch:1.3-cuda10.1-cudnn7-devel

語言：Python 3.6.9

套件：

pandas==1.1.5

numpy==1.17.2

opencv-python==4.5.3.56

opencv-python-headless==4.5.3.56

Shapely==1.7.1

pycocotools==2.0.2

imgaug==0.4.0

p-tqdm==1.3.3

fonttools==4.27.1

torchvision==0.4.1a0+d94043a

Pillow==8.2.0

matplotlib==3.3.4

albumentations==1.1.0

scipy==1.5.4

貳、演算方法與模型架構

字元偵測模型：Mask RCNN (backbone : resnet152)

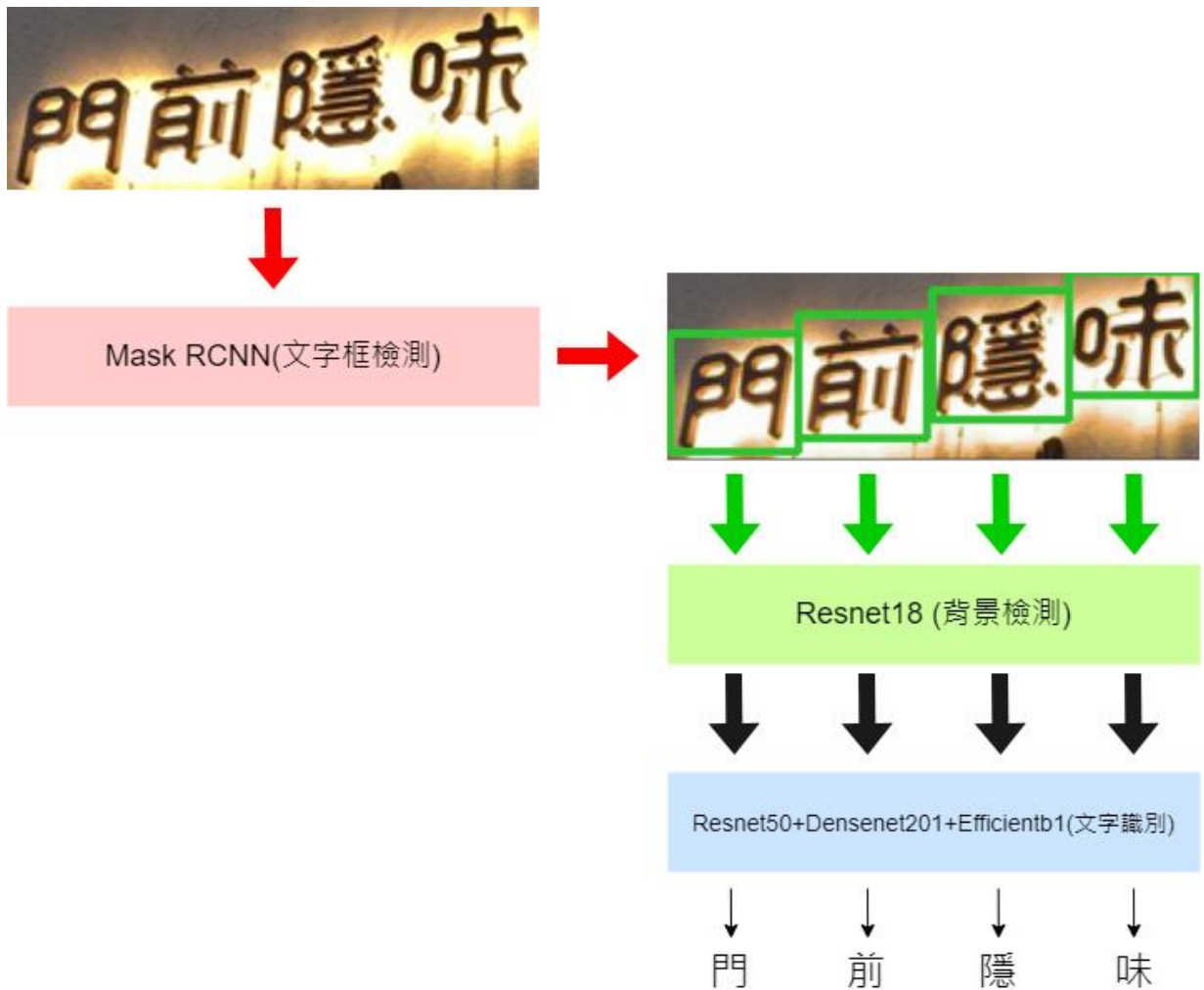
背景識別模型：Resnet18

字元識別模型：Resnet50, Densenet201, Efficientnet-b1

推論步驟：

1. 透過偵測模型檢測文字框
2. 以背景識別模型判斷文字框內的圖像是否為文字
3. 個別將為文字的圖像透過 3 種字元識別模型 ensemble (平均) 識別

流程圖：



參、 資料處理

對訓練資料較少見的字(少於 400 字) 以隨機字型進行影像生成。

字型：setofont, HanyiSentyTang, Kaisotai-Next-UP-B,
Senobi-Gothic-Regular, SNSanafonkaku, HanaMinA, RiiT_F, Brush,
Calligraphy, Advertisement, Stamp

範例：



肆、 訓練方式

模型	影像處理	資料增強	優化
字元偵測模型	灰階	水平翻轉、模糊化	SGD lr:5e-3
背景識別模型		RGBShift、	Adam lr: 1e-4
字元識別模型		Elastictransform、對比、模糊化、反色、旋轉、雜訊、大小變換、長寬比變換	

伍、 分析與結論

分析:

1. 若使用常見的識別架構如 CRNN 等等，垂直文字的識別準確度會比較低，也需要進行更多樣化的資料生成和方向判別，而這次的訓練數據大概有 40% 是垂直文字，因此藉上回比賽經驗採取兩階段(偵測+識別)的策略，成效還不錯。
2. 上次偵測的比賽幾乎是以原圖來訓練，這次經過調研後發現模糊化的圖像對文字框的檢測會很有幫助。
3. 採用簡單的 ensemble (平均) 可以提升不少的識別準確率，在 public leaderboard 的分數提升了 0.03 分左右。

待改善:

1. 若以識別模型的 logits 或中間的 layer 作為字元的 embedding 再接上語言模型(BERT、CNN、RNN 等)，則可透過語義來修正圖像識別錯誤的情況。
2. 有不少的繁體中文沒出現在訓練資料集，因此還需要進行更多的影像生成，應該會提升不少識別準確度。
3. 還有很多沒發現到的問題

陸、 程式碼

如附件

柒、 使用的外部資源與參考文獻

無

聯絡資料

● 隊伍

隊伍名稱	Private leaderboard 成績	Private leaderboard 名次
GPU 不夠了	0.868158	6

● 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
吳亦振(Yi-Chen Wu)	國立清華大學統計學研究所(畢業)	0972771286	a0972771286@gmail.com