主要程式碼都位於”整理”資料夾內

1. 偵測

本研究共有YOLOv4、U-net、EAST三種偵測模型，其對應的程式分別位於yolo、unet、EAST三個資料夾。

1.1yolo資料夾內有.py檔 yolo.py 為YOLOv4偵測字串區域的程式，執行後將會把位置"../../groundtruth/image"的圖像做偵測，並將結果存於"../yolo-Result-best - rulebase/"。

1.2 unet資料夾內有.py檔 detectionUnet.py 為U-net偵測字串區域的程式，執行後將會把位置"../../groundtruth/image"的圖像做偵測，並將結果存於"../unet-Result-best- rulebase /"。

1.3 EAST資料夾內有.py檔 testFlooder.py 為U-net偵測字串區域的程式，執行後將會把位置"../../groundtruth/image"的圖像做偵測，並將結果存於"../EAST-Result/"。

1. 字串辨識

字串辨識資料夾為OCR ，資料夾內testmarge.py為辨識的程式碼。

調用方法:

python testmarge.py --modelName <設定檔名>--typedetect <裁切圖片資料夾名稱> --typedata <圖片輸入類型>

範例:

python testmarge.py --modelName STARNet --typedetect marge --typedata RGB

其結果為調用”./cfg/ <設定檔名>.py”之設定檔建立模型，並辨識”../cutImage /<裁切圖片資料夾名稱>/ <圖片輸入類型>”內的圖像，最後將結果存於"../CrossTest-Result/ <圖片輸入類型>\_ <設定檔名>\_ <裁切圖片資料夾名稱>"

1. 雜項

3.1cutImage.py 用以將偵測結果裁切成辨識圖片

調用方法:

python cutImage.py --flooder <資料夾名稱>

範例:

python cutImage.py --flooder yolo-Result-best

其結果為將<資料夾名稱>下的偵測結果存於”./cutImage/<資料夾名稱>” 資料夾下

3.2 merge.py 會將YOLOv4與U-net的結果合併，必須先分別偵測後，產生"./unet-Result-best-rulebase/", "./yolo-Result-best-rulebase/"兩資料夾，合併後的結果存於”./marge/”

3.3 mergeOCRresult.py 可將字串辨識結果投票，使用前須先觀察並修改程式下方的設定方式(目前無法直接執行)

3.4 test.py用以計算偵測結果之準確率

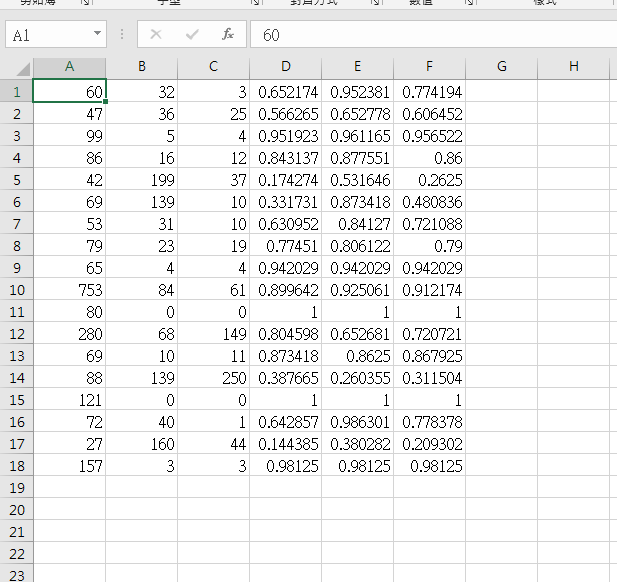
調用方法:

python test.py --flooder <資料夾名稱>

範例:

python test.py --flooder unet-Result-Rulebase

其結果為將計算”./<資料夾名稱>”下的偵測評估並存於”./csv/<資料夾名稱>.csv”，csv如下圖



A~F欄位依序是，TP, FP, FN, precision, recall, F1

3.5 testCrossTest.py用以計算偵測加辨識結果之準確率

調用方法:

python testCrossTest.py --flooder <資料夾名稱>

範例:

python testCrossTest.py --flooder marge\_CRNN\_BINARY

其結果為將計算”. ./CrossTest-Result/<資料夾名稱>”下的偵測評估並存於”./ CrossTest /<資料夾名稱>.csv”，csv同3.4