

Evaluation of Precision and Recall

1.介紹

evaluatePrecisionRecall.py是以libs/PMXLIB_Evaluation.py為Library的程式，其功能包含了寫XML及計算Precision & Recall。其中計算Precision & Recall包含了計算全畫面、畫線區域內兩種。

2.操作方式

步驟1.

執行evaluatePrecisionRecall.py時，輸入的指令由 (SSD_LOG) -> (YOLO_LOG) -> (GT_PATH) -> (Pic_PATH) -> (其他功能) 所構成，可以用help (-h) 指令查詢。

```
python3 evaluatePrecisionRecall.py --ssd_path {ssd_log .txt} --yolo_path {yolo_log .txt} \
--gt_path {GT xml folder} -pic_path {image folder} -res {resolution} --draw_line {Read or image path}
```

步驟2.

執行程式所獲得的檔案 (.txt) 會被回存到資料夾當中。

3.Example

計算全畫面的Precision & Recall，假設此次需要進行計算的相關資料路徑如下，且照片解析度為預設的1920*1080。

```
ssd_path: ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt
yolo_path: ../example/logs/20201104-part2_yolo_608x608.txt
gt_path: ../example/GT/20201104-part2
pic_path: ../example/pics/20201104-part2
```

步驟1.

依據上方所提供的資料路徑執行evaluatePrecisionRecall.py，指令如下。若有需要自訂解析度，可輸入-res指令。

```
python3 evaluatePrecisionRecall.py \
--ssd_path ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt \
--yolo_path ../example/logs/20201104-part2_yolo_608x608.txt \
--gt_path ../example/GT/20201104-part2 \
--pic_path ../example/pics/20201104-part2
```

步驟2.

取得計算後的檔案 (.txt) 。

```
huangkaijiedeMacBook-Pro-2 pmx_tools/AI_Evaluation <master> » cd outPut
huangkaijiedeMacBook-Pro-2 AI_Evaluation/outPut <master> » ls
20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k_Eval.txt
20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k_Eval_draw.txt
```

```

Load log from ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt.
Load pics from ../example/pics/20201104-part2.
Load log from ../example/logs/20201104-part2_yolo_608x608.txt.
Load pics from ../example/pics/20201104-part2.

===== loadAItxt from ssd =====

===== loadAItxt from yolo =====

===== Read xml files from ../example/GT/20201104-part2 =====

vertex_list: None

===== SSD vs. YoloV4 =====

Create Table
Count AI in GT: 4485

Precision: 0.887698986975398
Recall: 0.6414975946454716
TP: 3067
TP_TABLE: [411, 1039, 1617, 0]
AI_TABLE: [469, 1135, 1849, 2]
GT_TABLE: [479, 1382, 2623, 1]
sum_AI: 3455
sum_GT: 4781

RESULTS:


|         |    | Precision | Recall |
|---------|----|-----------|--------|
| Bike    | -> | 0.88      | 0.86   |
| Vehicle | -> | 0.92      | 0.75   |
| Person  | -> | 0.87      | 0.62   |



===== SSD vs. GT =====

```

狀況1：手動畫點計算畫線區域內的Precision & Recall

假如此次要以手動畫點的方式，完成對畫線區域內Precision & Recall的計算，相關資料路徑如下。

```

ssd_path: ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt
yolo_path: ../example/logs/20201104-part2_yolo_608x608.txt
gt_path: ../example/GT/20201104-part2
pic_path: ../example/pics/20201104-part2
sample pic path: ../example/pics/20201104-part2/20201104-part2_00000.jpg

```

步驟1.

執行evaluatePrecisionRecall.py，並開啟 -dl功能（輸入sample pic path）。

```

python3 evaluatePrecisionRecall.py \
--ssd_path ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt \
--yolo_path ../example/logs/20201104-part2_yolo_608x608.txt \
--gt_path ../example/GT/20201104-part2 \
--pic_path ../example/pics/20201104-part2 \
--draw_line ../example/pics/20201104-part2/20201104-part2_00000.jpg

```

步驟2.

進入指定圖片後，需用滑鼠在照片上畫點形成區域。過程中若需重置視窗，可輸入 **C**。完成後可輸入 **Q** 來關閉視窗。



步驟3.

執行程式所獲得的檔案會存放到下方路徑中。

```

huangkaijiedeMacBook-Pro-2 AI_Evaluation/outPut <master*> » ls
20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k_Eval.txt
20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k_Eval_draw.txt

Load log from ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt.
Load pics from ../example/pics/20201104-part2.
Load log from ../example/logs/20201104-part2_yolo_608x608.txt.
Load pics from ../example/pics/20201104-part2.

===== loadAI.txt from ssd =====

===== loadAI.txt from yolo =====

===== Read xml files from ../example/GT/20201104-part2 =====
vertex_list: [[0, 314.94054054054055], [1920, 195.5891891891892], [1920, 1080], [0, 1080], [0, 0], [0, 314.94054054054055]]

===== SSD vs. YoloV4 =====

Create Table
Count AI in GT: 4083

Precision: 0.8991161231331911
Recall: 0.7135945815191098
TP: 2950
TP_TABLE: [406, 975, 1569, 0]
AI_TABLE: [463, 987, 1829, 2]
GT_TABLE: [477, 1248, 2357, 1]
sum_AI: 3281
sum_GT: 4134

RESULTS:

```

		Precision	Recall
Bike	->	0.88	0.85
Vehicle	->	0.99	0.78
Person	->	0.86	0.67

```

===== SSD vs. GT =====

Create Table
Count AI in GT: 577

Precision: 0.8993288590604027
Recall: 0.6967071057192374

```

狀況2：讀取座標計算畫線區域內的Precision & Recall

假如此次要以讀取座標的方式，完成對畫線區域內Precision & Recall的計算，相關資料路徑如下。

```
ssd_path: ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt
yolo_path: ../example/logs/20201104-part2_yolo_608x608.txt
gt_path: ../example/GT/20201104-part2
pic_path: ../example/pics/20201104-part2
```

步驟1.

執行evaluatePrecisionRecall.py，並開啟 -dl功能（輸入 read）。

```
python3 evaluatePrecisionRecall.py \
--ssd_path ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt \
--yolo_path ../example/logs/20201104-part2_yolo_608x608.txt \
--gt_path ../example/GT/20201104-part2 \
--pic_path ../example/pics/20201104-part2 \
--draw_line read
```

步驟2.

指令輸入後，會顯示已經被儲存的座標，此時僅需輸入前方編號即可繼續進行畫線工作。

```
(base) primax1220@vm:~/VMworkplace/repo/pmx_tools/AI_Evaluation$ python3 evaluatePrecisionRecall.py \
> --ssd_path ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt \
> --yolo_path ../example/logs/20201104-part2_yolo_608x608.txt \
> --gt_path ../example/GT/20201104-part2 \
> --pic_path ../example/pics/20201104-part2 \
> --draw_line read
0 : [[0, 526.6223404255319], [1920, 230.45212765957447], [1920, 1080], [0, 1080], [0, 0], [0, 526.6223404255319]]
1 : [[0, 1038.4373259052925], [1920, -491.14484679665736], [1920, 1080], [0, 1080], [0, 0], [0, 1038.4373259052925]]
2 : [[0, 325.7659137577002], [1920, 392.78850102669406], [1920, 1080], [0, 1080], [0, 0], [0, 325.7659137577002]]
3 : [[0, -2.9699999999999704], [1920, 886.63], [1920, 1080], [0, 1080], [0, 0], [0, -2.9699999999999704]]
4 : [[0, 602.224043715847], [1920, 455.3387978142076], [1920, 1080], [0, 1080], [0, 0], [0, 602.224043715847]]
5 : [[0, 219.47381546134665], [1920, 401.4189526184539], [1920, 1080], [0, 1080], [0, 0], [0, 219.47381546134665]]
6 : [[0, 18.9315068493151], [1920, 676.4657534246576], [1920, 1080], [0, 1080], [0, 0], [0, 18.9315068493151]]
7 : [[0, -329.20000000000005], [1920, 54262.8], [1920, 1080], [0, 1080], [0, 0], [0, -329.20000000000005]]
8 : [[0, 11.868421052631579], [1920, 22.394736842105264], [1920, 1080], [0, 1080], [0, 0], [0, 11.868421052631579]]
9 : [[0, 42.91803278688525], [1920, 239.9814682822523], [1920, 1080], [0, 1080], [0, 0], [0, 42.91803278688525]]
10 : [[0, 99.32197244379984], [1920, 959.7715736040609], [1920, 1080], [0, 1080], [0, 0], [0, 99.32197244379984]]
11 : [[0, -901.4000000000001], [1920, 4282.6], [1920, 1080], [0, 1080], [0, 0], [0, -901.4000000000001]]
12 : [[0, 368.3853820598007], [1920, 389.6478405315615], [1920, 1080], [0, 1080], [0, 0], [0, 368.3853820598007]]
13 : [[0, 539.8728702490171], [1920, 278.1690694626474], [1920, 1080], [0, 1080], [0, 0], [0, 539.8728702490171]]
14 : [[0, -210.17962466487927], [1920, 1277.4343163538874], [1920, 1080], [0, 1080], [0, 0], [0, -210.17962466487927]]
15 : [[0, 555.2083333333334], [1920, 275.2083333333333], [1920, 1080], [0, 1080], [0, 0], [0, 555.2083333333334]]
Which vertex list do you want to draw: 0
```

步驟3.

執程式所獲得的檔案會存放到下方路徑中。

```
huangkaijiedeMacBook-Pro-2 AI_Evaluation/outPut <master*> » ls
20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k Eval.txt
20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k Eval draw.txt
```

```

Load log from ../example/logs/20201104-part2_ssd_v5.1.1_640x360_Data20210420-retrain5b-160k.txt.
Load pics from ../example/pics/20201104-part2.
Load log from ../example/logs/20201104-part2_yolo_608x608.txt.
Load pics from ../example/pics/20201104-part2.

===== loadAITxt from ssd =====

===== loadAITxt from yolo =====

===== Read xml files from ../example/GT/20201104-part2 =====
vertex_list: [[0, 526.6223404255319], [1920, 230.45212765957447], [1920, 1080], [0, 1080], [0, 0], [0, 526.6223404255319]]

===== SSD vs. YoloV4 =====

Create Table
Count AI in GT: 3363

Precision: 0.9006013441811107
Recall: 0.7466275659824047
TP: 2546
TP_TABLE: [406, 970, 1170, 0]
AI_TABLE: [460, 982, 1383, 2]
GT_TABLE: [477, 1243, 1642, 1]
sum_AI: 2827
sum_GT: 3410

RESULTS:

```

		Precision	Recall
Bike	->	0.88	0.85
Vehicle	->	0.99	0.78
Person	->	0.85	0.71

```

===== SSD vs. GT =====

Create Table
Count AI in GT: 470

Precision: 0.897172236503856
Recall: 0.7425531914893617
TP: 349

```