

一、预测模型，将结果保存wider同文件夹形式的txt结果

二、用预测结果计算Map

最近在评估目标检测模型，整理一份完整的规范，以供以后评估使用。

附件：（完整项代码）



eval_tools.zip
1.73MB

整体步骤为：

1. 预测模型，将结果保存
2. 根据预测结果和wider 给出的测试mat，计算Map

一、预测模型，将结果保存wider同文件夹形式的txt结果

1. 按需修改args 参数
2. 书写模型的load函数，返回模型
3. 书写模型测试函数，返回numpy格式的boxes、probs
4. python widerface_val.py

```
1 #utils.py
2 #! /usr/bin/env python
3 # -*- coding: utf-8 -*-
4 # author: shengdan in 2019.10.18
5
6 import os
7 import cv2
8
9 #写入 wider 标签格式txt文件
10 def writeResToTxt(args, boxes, probs, imgPath):
11     '''write result into txt.
12     Args:
13         args: Custom parameter.
14         boxes: Model prediction box results, type: numpy.
15         probs: Model prediction box's confidence results, type: numpy.
16
17     Returns:
18         Generate the file.
19     '''
20     save_folder = args.save_folder
21     widerface_root = args.widerface_root
22     save_txt = imgPath.replace(widerface_root, save_folder).replace('.jpg', '.txt')
23     save_dir = save_txt.replace(os.path.basename(save_txt), '')
24
25     if not os.path.exists (save_dir):
26         os.makedirs(save_dir)
27
28     f = open(save_txt, 'w')
29     f.write(os.path.basename(imgPath)+'\n')
30     f.write(str(len(boxes)) + '\n')
31
32     for i in range(len(boxes)):
33         box = boxes[i, :]
34         label = probs[i]
35         line = str(box[0])+ ' '+str(box[1]) + ' '+str(box[2]-box[0])+ ' '+str(box[3]-box[1])+ ' '+str(
```

```

1  #! /usr/bin/env python
2  # -*- coding: utf-8 -*-
3  # author: shengdan in 2019.10.18
4
5  import argparse
6  import sys
7  import os
8  import cv2
9  import time
10 from utils import writeResToTxt
11
12 #*****参数变量*****
13 parser = argparse.ArgumentParser(description='detect_video')
14 parser.add_argument('--net_type', default="mb_tiny_RFB_fd", type=str, help='The network architecture')
15 parser.add_argument('--input_size', default=320, type=int, help='define network input size, default (320, 320)')
16 # parser.add_argument('--trained_model', default='./weights/WIDERFace_DSFD_RES152.pth', type=str, help='trained model path')
17 parser.add_argument('--threshold', default=0.6, type=float, help='score threshold')
18 parser.add_argument('--candidate_size', default=1000, type=int, help='nms candidate size')
19 parser.add_argument('--device', default="cuda:0", type=str, help='cuda:0 or cpu')
20 parser.add_argument('--label_path', default='../models/voc-model-labels.txt', type=str, help='label path')
21 parser.add_argument('--save_folder', default='./WIDERFace_mb_tiny_RFB_fd_320/', type=str, help='Dir to save results')
22 parser.add_argument('--widerface_root', default='/home/datalab/ex_disk/work/shengdan/dataset/Face_Landmarks', type=str, help='WIDERFace root')
23 args = parser.parse_args()
24
25 #*****模型加载*****
26 def loadUltraModel(args):
27     '''load model.
28     Args:
29         args: Custom parameter.
30     Returns:
31         eval model.
32     '''
33     #模型加载参数, 返回预测模型
34     return predictor
35
36 #*****模型预测*****
37 def ultraDetectFace(args, orig_image, predictor):
38     '''detect face.
39     Args:
40         args: Custom parameter.
41         orig_image: cv2 image BGR.
42         predictor: eval model.
43     Returns:
44         boxes.numpy(): x1,y1,w,h.
45         probs.numpy(): conf.
46     '''
47     # boxes, labels, probs = predictor.predict(image, args.candidate_size / 2, args.threshold)
48     return boxes.numpy(), probs.numpy()
49
50 if __name__ == "__main__":
51     predictor = loadUltraModel(args)
52     num = 0

```

```

53     widerface_root = args.widerface_root
54     for root, dirs, files in os.walk(widerface_root, topdown=False):
55         for name in files:
56             imgPath = os.path.join(root,name)
57             orig_image = cv2.imread(imgPath)
58
59             boxes,probs = ultraDetectFace(args,orig_image,predictor)
60             writeResToTxt(args,boxes,probs,imgPath)
61             num +=1
62             print('Dealed :',num)

```

生成结果：

```

(py3.6_torch0.4) shengdan@baihu:/home/datalab/ex_disk/work/shengdan/Ultra-Light-Fast-Generic-Face-Detector-1MB/eval_ols/WIDERFace_mb_tiny_RFB_fd_320$ ls
11--Meeting 17--Ceremony 53--Raid
(py3.6_torch0.4) shengdan@baihu:/home/datalab/ex_disk/work/shengdan/Ultra-Light-Fast-Generic-Face-Detector-1MB/eval_ols/WIDERFace_mb_tiny_RFB_fd_320$ cd 11--Meeting/
(py3.6_torch0.4) shengdan@baihu:/home/datalab/ex_disk/work/shengdan/Ultra-Light-Fast-Generic-Face-Detector-1MB/eval_ols/WIDERFace_mb_tiny_RFB_fd_320/11--Meeting$ ls
11_Meeting_Meeting_11_Meeting_Meeting_11_102.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_468.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_108.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_507.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_176.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_529.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_189.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_560.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_206.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_573.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_223.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_574.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_250.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_587.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_26.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_633.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_287.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_639.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_295.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_644.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_320.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_663.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_339.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_702.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_349.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_71.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_375.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_774.txt
11_Meeting_Meeting_11_Meeting_Meeting_11_385.txt 11_Meeting_Meeting_11_Meeting_Meeting_11_807.txt

```

二、用预测结果计算Map

```

1 # before evaluating
2 python setup.py build_ext --inplace
3
4 # after
5 python evaluation.py -p <your prediction dir> -g <groud truth dir>

```

运行结果：

```

Reading Predictions : 100%| 61/61 [00:00<00:00, 333.54it/s]
Processing easy: 100%| 61/61 [00:17<00:00, 3.36it/s]
Processing medium: 100%| 61/61 [00:17<00:00, 3.32it/s]
Processing hard: 100%| 61/61 [00:17<00:00, 3.33it/s]
===== Results =====
Easy Val AP: 0.7339617503641326
Medium Val AP: 0.5942347587452728
Hard Val AP: 0.31169635423243847

```