

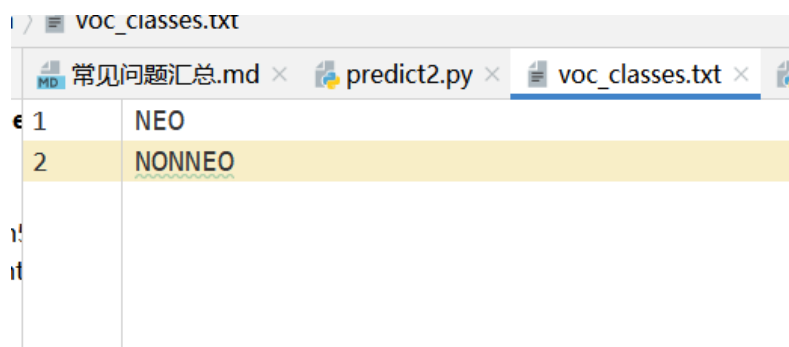
yolo-predict 自动分类代码说明

姚方浩

版本更新 6.7

注意：运行前需检查 `voc_classes.txt` 是否正确！

如果不正确程序会出现 `mismatch` 报错。



1 predict2.py 操作步骤

1.1 准备工作，检查文件有无缺失和错误

检查 `voc_classes.txt` 中标签是否有误

检查 `logs/` 中的权重是否匹配

检查是否有需要调用的 `yolo_predict2.py`

检查其他文件是否完整

1.2 需要根据自己的需要，修改 yolo_predict2.py 中的 confidence

```
predict2.py
— predict2.py × yolo_predict2.py × yolo_predict1.py ×
43 # phi = 2代表efficientnet-B2-yolov3
44 # ..... 以此类推
45 #-----#
46 "phi" : 2,
47 #-----#
48 # 只有得分大于置信度的预测框会被保留下来
49 #-----#
50 "confidence" : 0.1,
51 #-----#
52 # 非极大抑制所用到的nms_iou大小
53 #-----#
54 "nms_iou" : 0.3,
55 "max_boxes" : 100,
```

1.3 对应地，修改 predict2.py 中的 confidence_num

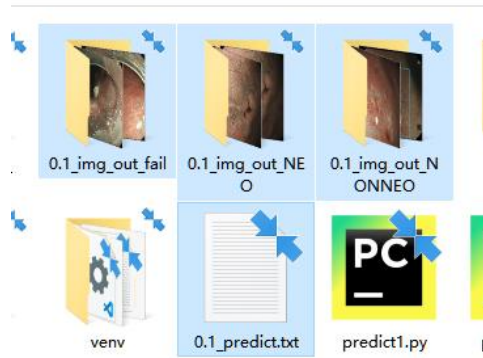
```
t2.py
— predict2.py × yolo_predict2.py × predict1.py ×
70 import logging
71 logging.basicConfig(level=logging.DEBUG,
72                     format='%(asctime)s - %(levelname)s
73                     logging.disable(logging.CRITICAL) # 不显示debug日志记录。
74
75 #-----#
76 # !!! 唯一需要手动设置的地方: confidence_num
77 confidence_num = 0.1
78 #-----#
79
80 # 创建图片输出的文件夹
81 dir_save_path_NEO = str(confidence_num) + str('_') + st
```

1.4 运行 predict2.py 即可

```
simhei.ttf if __name__ == "__main__" > elif mode == "dir_predi
运行: predict2 (1) x
b'NEO 0.70' 45 107 572 735
b'NONNEO 0.67' 152 235 559 711
-----
王勇卿(2).jpg
42%|██████████| 162/388 [02:19<03:07, 1.20it/s]Found 2 boxes
b'NEO 0.69' 30 31 570 670
b'NONNEO 0.12' 38 219 438 619
-----
王勇卿(3).jpg
42%|██████████| 163/388 [02:20<03:17, 1.14it/s]Found 2 boxes
b'NEO 0.31' 272 320 598 636
b'NONNEO 0.41' 272 320 598 636
-----
王勇卿(5).jpg
42%|██████████| 164/388 [02:21<03:36, 1.03it/s]Found 2 boxes
b'NEO 0.79' 232 237 645 698
b'NONNEO 0.16' 271 242 665 697
-----
王勇卿(6).jpg
-----
if __name__ == "__main__" > elif mode == "dir_
运行: predict2 (1) x
b'NEO 0.76' 94 163 522 539
b'NONNEO 0.37' 163 98 555 507
-----
龚秋芳1.jpg
100%|██████████| 387/388 [06:36<00:00, 1.22it/s]Found 2 b
b'NEO 0.79' 5 67 360 517
b'NONNEO 0.22' 14 69 414 515
-----
龚秋芳2.jpg
100%|██████████| 388/388 [06:37<00:00, 1.02s/it]
进程已结束，退出代码为 0
```

1.5 输出结果

输出分为三个文件夹，并生成一个 txt 文档，如下所示：



2 predict2.py 代码的简要说明

2.1 对 yolo.py 进行了修改，并保存为新程序 yolo_predict2.py

仅修改了一处地方，在 detect_image() 函数的返回值添加了 out_scores 和 out_classes 两项。

out_scores 表示预测出的框的置信度大小，例如 `tf.Tensor([0.4555555])`

out_classes 表示预测出的框的类别，例如 `tf.Tensor([0])`，0 表示 NEO，1 表示 NONNEO。

```

predict2.py x yolo_predict2.py x predict1.py x
217 text_origin = np.array([left, top + 1])
218
219 for i in range(thickness):
220     draw.rectangle([left + i, top + i, right - i, bottom - i], outline=self.color
221 draw.rectangle([tuple(text_origin), tuple(text_origin + label_size)], fill=self.c
222 draw.text(text_origin, str(label,'UTF-8'), fill=(0, 0, 0), font=font)
223 del draw
224
225 # 打印结果为 tf.Tensor[0]、 tf.Tensor[0 1]等等 0表示NEO, 1表示NONNEO
226 # print(out_classes)
227
228 return image, out_scores, out_classes # 和原版相比, 添加了 out_scores, out_classes
229
230 def detect_heatmap(self, image, heatmap_save_path):

```

2.1 对 predict.py 进行了修改, 并保存为新程序 predict2.py

```

predict2.py
74 #-----#
75 # !!! 唯一需要手动设置的地方: confidence_num
76 confidence_num = 0.1
77 #-----#
78
79 # 创建图片输出的文件夹
80 dir_save_path_NEO = str(confidence_num) + str('_') + str(dir_save_path) + "_NEO/"
81 dir_save_path_NONNEO = str(confidence_num) + str('_') + str(dir_save_path) + "_NONNEO/"
82 dir_save_path_fail_output = str(confidence_num) + str('_') + "img_out_fail/"
83
84 if not os.path.exists(dir_save_path_NEO):
85     os.makedirs(dir_save_path_NEO)
86 if not os.path.exists(dir_save_path_NONNEO):
87     os.makedirs(dir_save_path_NONNEO)
88 if not os.path.exists(dir_save_path_fail_output):
89     os.makedirs(dir_save_path_fail_output)
90
91 # 统计识别和未识别图片个数
92 num_NEO = 0
93 num_NONNEO = 0
94 fail = 0
95
96

```

根据需手动设置的置信度阈值, 大于该值输出预测框, 小于该值不输出

输入和输出文件夹路径设置

```

predict2.py x yolo_predict2.py x
102 # 修改detect_image 的返回值
103 r_image, out_scores, out_classes = yolo.detect_image(image)
104
105 # 打印测试结果
106 print('-----')
107 print(img_name)
108 logging.debug(out_scores)
109 out_scores_size = out_scores.numpy().size
110 logging.debug(out_scores_size)
111
112 threshold = tf.constant([confidence_num]) 创建置信度阈值的张量threshold
113
114 # 比较 out_scores 和 threshold
115 flag = tf.greater(out_scores, threshold) 比较预测得到的out_scores和阈值threshold大小, 如果前者大于后者, flag为True
116 if any(flag): # any()语法: 只要flag中存在True时即返回True 如果预测出多个框, 只要有一个大于阈值, any(flag)即为True, 进入循环
117     # flag2 = (predicted_class == 'NEO')
118     flag2 = (0 in out_classes) # 如果NEO (标签为0) 在预测结果中, 返回True
119     if flag2:
120         flag2检测预测框中是否有预测到NEO类型 (标签为0), 如果有, 保存至NEO文件夹
121         r_image.save(os.path.join(dir_save_path_NEO, img_name.replace(".jpg", ".png")), quality=95, subsampling=0)
122         num_NEO += 1
123     else:
124         # 如果没有, 保存至NONNEO文件夹
125         r_image.save(os.path.join(dir_save_path_NONNEO, img_name.replace(".jpg", ".png")), quality=95, subsampling=0)
126         num_NONNEO += 1
127 else:
128     # 没有预测出框的, 保存至fail文件夹
129     r_image.save(os.path.join(dir_save_path_fail_output, img_name.replace(".jpg", ".png")), quality=95, subsampling=0)
130     fail += 1
131
132

```

调用yolo_predict2.py中的方法

```

predict2.py x yolo_predict2.py x
126 else:
127     r_image.save(os.path.join(dir_save_path_fail_output, img_name.replace(".jpg"
128     fail += 1
129
130 # 修改中---预测结果保存至txt中 统计图片总数total、识别出肿瘤的比例accuracy,
131 total = num_NEO + num_NONNEO + fail 并存至txt中
132 accuracy = num_NEO / total
133 f = open(os.path.join(os.getcwd(), str(confidence_num) + '_predict.txt'), 'a')
134 f.write("预测的图片总数: " + str(total))
135 f.write("\n")
136 f.write("---识别出置信框: " + str(num_NEO + num_NONNEO))
137 f.write("\n")
138 f.write(" |---识别为肿瘤性: " + str(num_NEO))
139 f.write("\n")
140 f.write(" |---识别为非肿瘤: " + str(num_NONNEO))
141 f.write("\n")
142 f.write("---未识别出置信框: " + str(fail))
143 f.write("\n")
144 f.write("识别肿瘤性的比例: " + str(accuracy*100) + "%")
145 f.close()
146
147 #####以上为修改部分#####

```