

大创项目

1. 嵌入Yolo去鬼影

去鬼影采用的是PaddlePaddle内PaddleDetection中的内置的PP-Yolo（COCO80类预训练参数），Paddle中内置的infer函数不输出bounding box，为获得bounding box使用改写的infer2label.py文件可生成包含bounding box的json文件。（此处为了观察Yolo检测结果，并改进拼接算法，并未去除画出预测框的代码，在实际使用中注释相关代码）

该部分需要完成的任务：

- 将 *infer2label.py* 函数中的主体部分嵌入到主程序中（因为历程中仅针对两张图片编写程序，需改动的地方较大，需要编程能力较强的人负责）
- Paddle可以对模型进行修剪和蒸馏，提高性能。具体方法参照Paddle的PaddleSlim的教程（此部分为提升模块可以选择放弃，但使用后在不损失精度的前提下可以提升大约**40%**的性能）
- COCO数据集中的80类物体大多为不可移动或不常移动的物体，所以可以筛选出可能发生移动的类别进行训练，可以提升最终效果。

2. 图像拼接速度提升

问题现已解决，仅需将程序整合进原有程序。现方法为使用缩小到原尺度长宽0.1~0.2的图像进行特征点提取和匹配，计算仿射矩阵，然后乘放缩矩阵得到原图像的仿射矩阵。（还原矩阵未找到文献，是我通过数据特征寻找的）

经测试速度可提升约**40倍**（对两张手机拍摄的图片进行特征提取仅需0.886s）

因为测试图片均为图书馆拍摄，场景中物体较多，特征较为明显，对于大尺度、特征点稀疏的图片放缩到0.1~0.2可能效果不好，需进行测试。

3. 图像融合算法

个人建议抓紧分配人手去把**加权平均融合**和**双线性插值算法**程序写出来，算法并不复杂，对每个像素进行处理即可。不用使用太复杂的算法，github上代码很多但是要嵌入比较麻烦，自己写就可以。

附录：

1. Yolo ipynb 文件结构：

- PaddleDetection(git clone <https://github.com/PaddlePaddle/PaddleDetection>)
- image（存储待拼接图片）
- output(存放输出)

2. 使用环境

opencv_python == 3.4.2.17

paddle == 2.0.1

PS：infer2label.py放在PaddleDetection/tools目录下

