

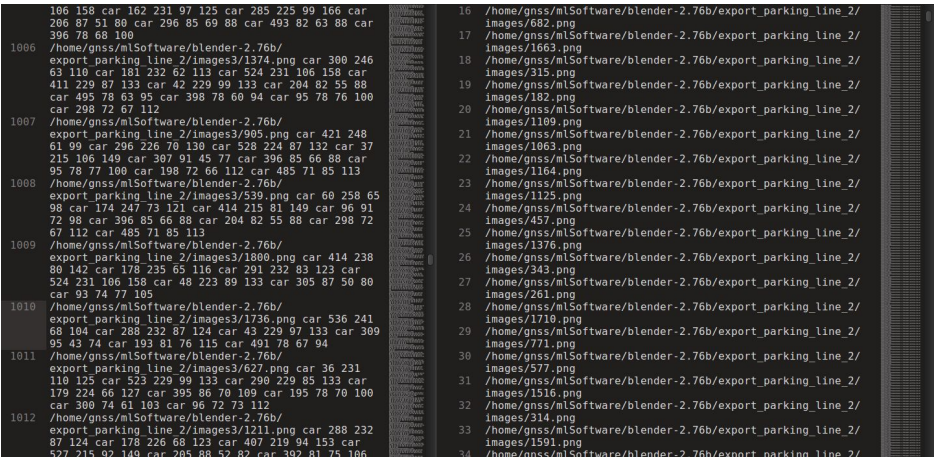
工作记录

今天的主要工作就是 3 点

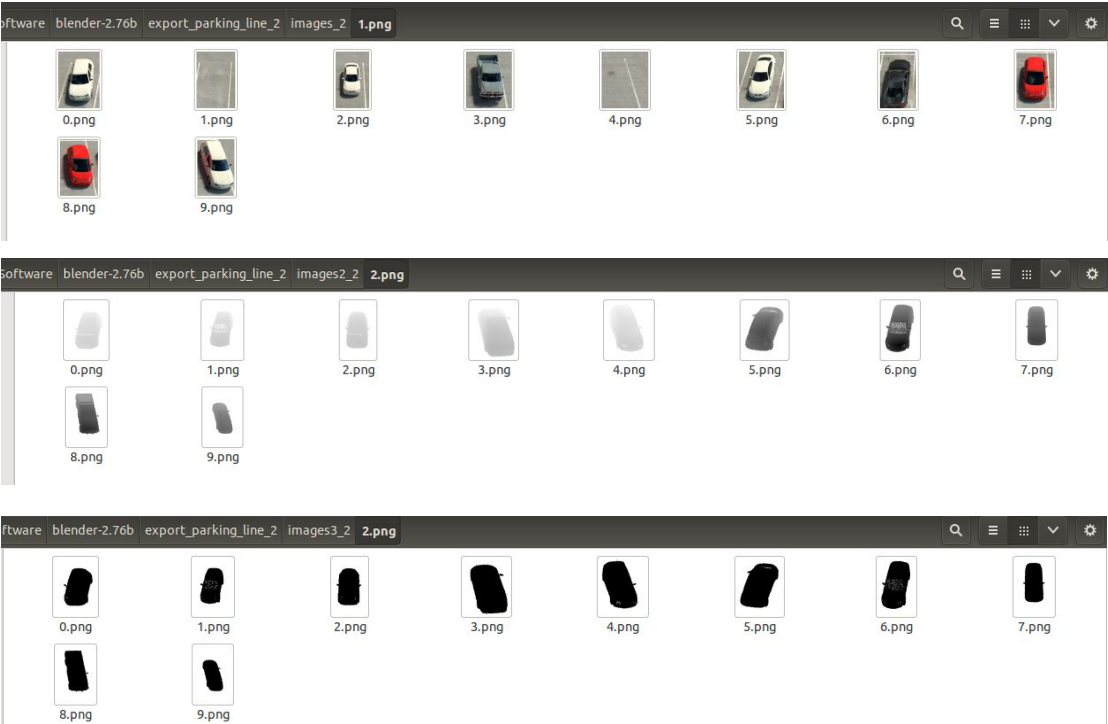
1.数据整理

a) 整理之前生产的图片并处理：

- 1. Train.txt
- 2. Annotation.txt



相关图片切割：



2. 加快图片生产效率

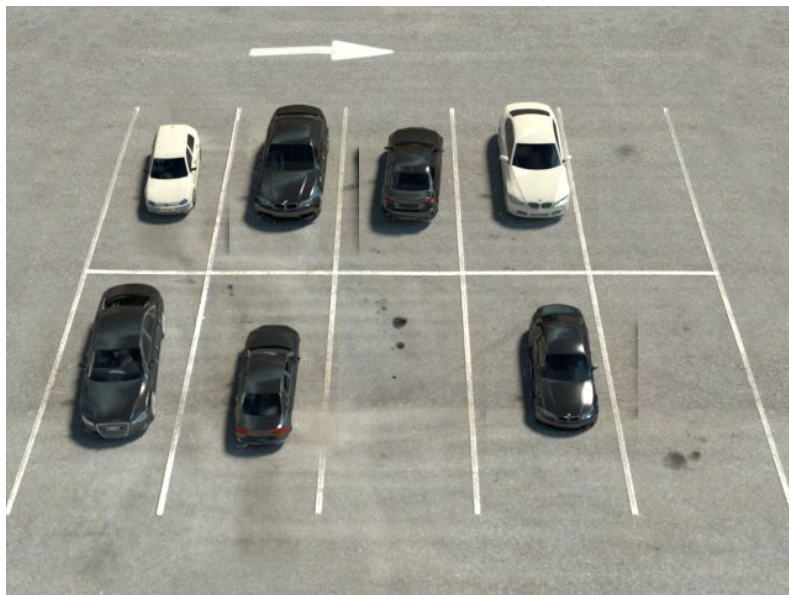
i. 可行方案 1:

1. 模型中有 30 种车, 生产 30 张停车场图片 (一张图片中有 10 个停车位, 10 个停车位上均停满一种车, RGB 和 Depth 两种), 同时生产一张没有车的停车位图片
2. 对 30 张图片进行切割组合 (切割出每个停车位, 然后在没有车的停车位图片上进行黏贴复制), 所有过程随机, 可以快速得到大量图片

实现 (循环太多, 计算量太大, 不适合, 需要继续更改代码):

```
for num_car in range(car_num):
    file_new_dirs=random.sample(file_dirs,10)           #get new file_dirs
    for i in range(len(posis_)):                         #get posis
        for file_dir in file_new_dirs:                  #get new file_dir
            img_lists=walk_img(file_dir)                #get img_lists
            #print img_lists[2].split('.'),'\n'
            img_posi_new=random.sample(img_posi_list,10)
            for img_posi_new_ in img_posi_new:
                for img_list in img_lists:
                    if img_list.split('.')==img_posi_new_:
                        img=cv2.imread(file_dir+'/'+str(img_posi_new_)+'.png')
                        rgb_img[int(posis_[img_posi_new_][1]):int(posis_[img_posi_new_][2]),int(posis_[img_posi_new_][3]):int(posis_[img_posi_new_][4])]=img
                        cv2.imwrite(file_dir+'/'+str(img_posi_new_)+'.png',rgb_img)
```

改进方案, 数组保存所有的图片地址, 再随机操作, 加快打印速度, 已实现:



ii. 可行方案 2:

多开几台主机, 多机同时工作, 耗费资源, 暂不采用

3. 多角度图片生产

10-80 度视角的图片



1.png



2.png



3.png



4.png



5.png



6.png



7.png



8.png

