

作业说明

一、作业内容

1. 基于 ransac 的地面去除（借鉴第 4 课内容）
2. 去除靠近地面和高于地面 2 米的点云后，投影到地面进行聚类（借鉴第 4 课内容）
3. 构建数据集，训练 PointNet，对聚类后的点云进行分类。（借鉴第 5 课内容）
4. 进行检测结果的评估，结果在 dt 文件夹中

二、作业细节

1. 数据集构建

- 1.1 根据 KITTI 数据集中的 label 和 calib 文件，将 Car、Cyclist 和 Ped 的 boundingbox 中的点云提取出来，分别构建成三个类型障碍物的样本。
- 1.2 利用第 4、5 节课的内容，将 KITTI 点云文件中的障碍物聚类提取出来，如果某个类的所有点云都不在 Car、Cyclist 和 Ped 的 boundingbox 中，记录为 Genobj 的一个样本。（Genobj 样本非常多，没有全部提取出来）。
- 1.3 一共得到了 21616 个 Car 样本；1230 个 Cyclist 样本；2803 个 Genobj 样本；3745 个 Ped 样本。（链接: <https://pan.baidu.com/s/10WZ2OIESjRsNcTq51t5CaA> 密码: 8Il7）

2. Dataset 类构建

- 2.1 四种类型的样本数量差距较大，本次作业是通过控制训练的样本中，四个种类的训练样本数量保持接近来抑制样本的不平衡问题。采用 Focal Loss 应该更加科学，时间问题还没有尝试。
- 2.2 按照训练、测试、验证的比例为 8:1:1 来分配样本。
- 2.3 为了用到 GPU 加速，每次读取 batch size 个样本进行训练，但问题在于，不同障碍物的点云数量不一致，无法多个样本一行读取，因此，这里记录了包含点云数量最多的障碍物的点云数，然后将其他障碍物的点云补(0.0, 0.0, 0.0)，使得所有障碍物点云数一致。

3. 网络训练与测试

- 3.1 在训练时，本来想在 model 的 forward 函数中，将所有原点坐标行都删掉（PointNet 是对每个点进行 MLP，在求 max，没有要求每个样本的点云数量一致），但这样做就无法一次计算 batch size 个输入了，所以就没有去除。希望让网络自己学习到原点坐标是无用行这一规律。

3.2 网络训练结果

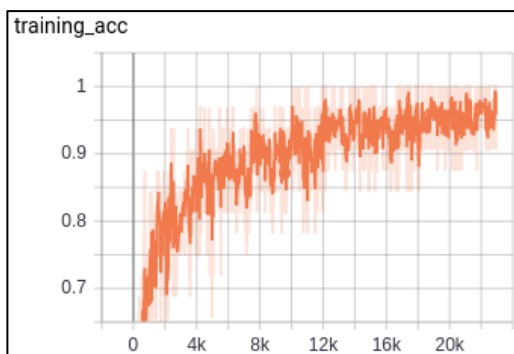


Fig 1. 训练准确率图

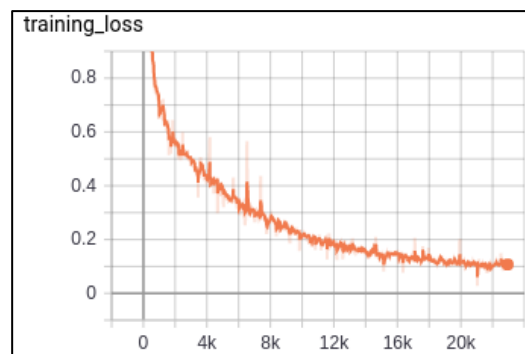


Fig 2. 训练 loss 图

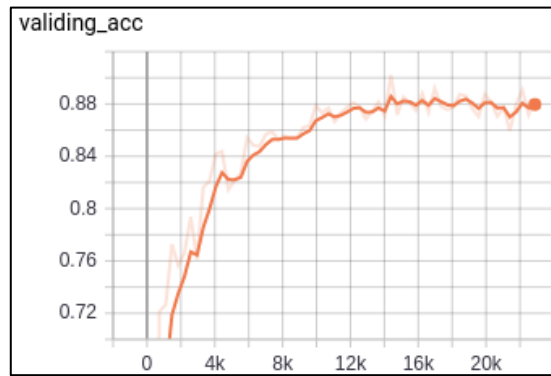


Fig 3. 验证准确率图

最终在 validation dataset 上的最高准确率为 90.2%

3.3 检测结果

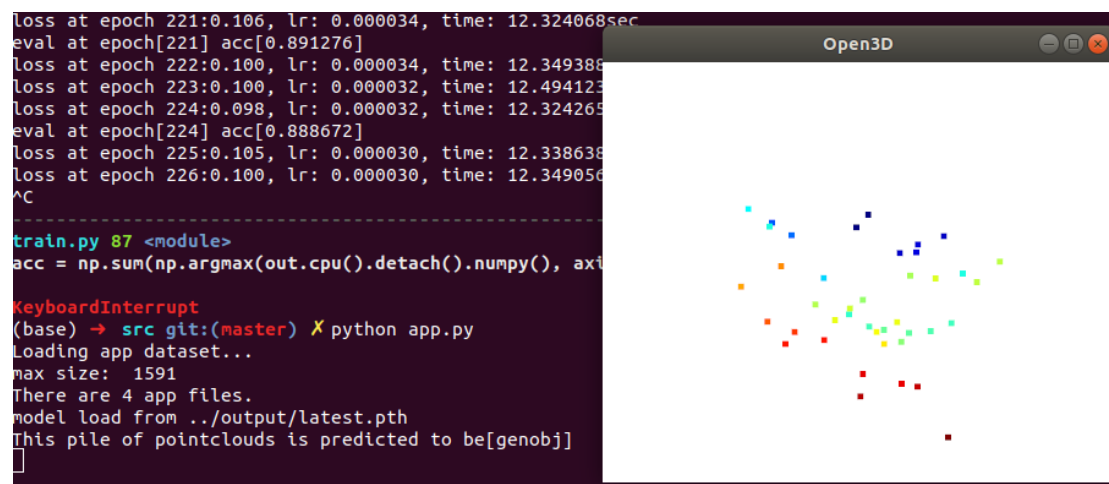


Fig 4. Genobj 检测结果

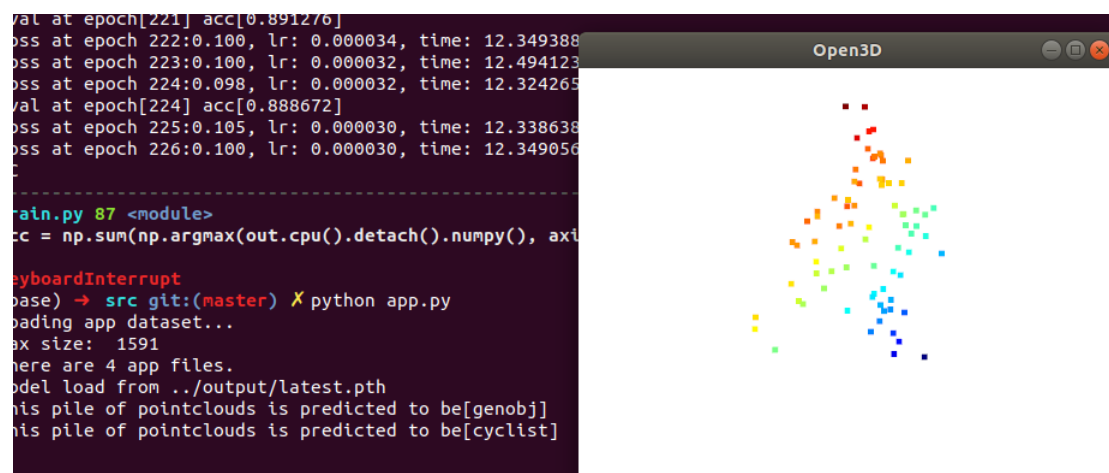


Fig 5. Cyclist 检测结果

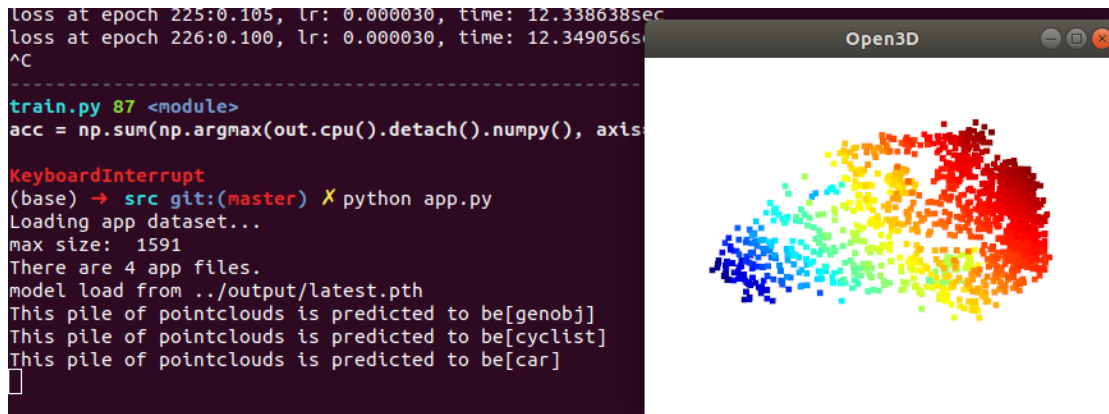


Fig 6. Car 检测结果

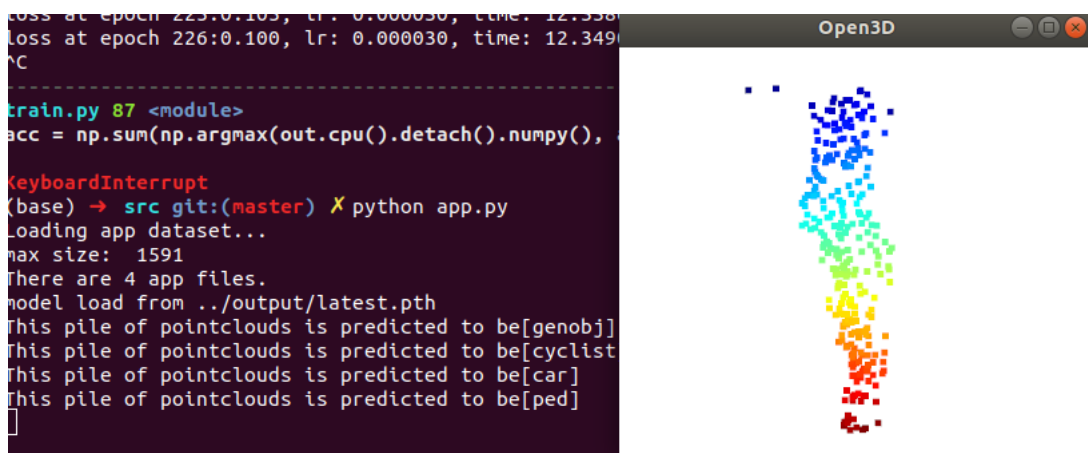


Fig 7. Ped 检测结果

4. 检测结果评估

4.1 truncated、occluded、alpha、bbox、rotation_y 和 score 是 mock 的数据；type、dimensions 和 location 是算出来的数据

4.2 type 是网络的分类结果

4.3 dimensions 中的 height 是计算该障碍物距离地面最远的点云的距离；width 和 length 是点云投影到地面坐标系后，在二维平面内的 x 和 y 轴上的跨度值，取较大的为 length，较小的为 width。

4.4 location 是点云坐标的平均值。

三、问题反馈

上面的检测结果可视化，是执行 app.py 得到的。该代码从四个类型障碍物对应的验证集样本中各挑选一个障碍物点云文件进行障碍物的分类。分类结果还是比较准确的。但是，如果执行 detection.py 文件，跑一个完整的地面去除、聚类、分类的 pipeline，得到的检测结果并不好。很多点云实例看起来不像 Car 的也被分类为 Car 了，精度非常差。

可能的结果如下：

1. 地面去除，为了去干净，把距离地面 0.3 米内的点都删除了，导致很多车的车轮子部分都被过滤掉了。而训练时用的 Car 样本是从 label 的 bbox 中提取到的完整点云，细节更加多。导致训练精度挺高，实际检测结果较差。

2. 训练的时候，每个障碍物的点云数量都补充零值到一个固定的 size，但实际检测的时候，直接把聚类结果的点云放进去推理了，导致差别过大。

四、文件介绍

dataset: 样本数据存在这里面

- | —— car
- | —— ped
- | —— genobj
- | —— cyclist
- | —— train_val_test_split.py: 生成 train_val_test.json 的脚本
- | —— train_val_test.json: 记录训练、测试、验证样本文件名的文件

mypointnet: 神经网络相关代码

- | —— output: 训练过程中&训练结果文件
 - | —— runs: tensorboard 输出
 - | —— latest.pth: 表现最好的权重参数文件
- | —— src: 源码
 - | —— app.py: 利用网络进行点云分类并可视化
 - | —— dataset.py: 数据集预处理代码
 - | —— model.py: 模型实现代码
 - | —— test.py: 模型测试代码
 - | —— train.py: 模型训练代码
 - | —— util.py: 辅助函数实现
- | —— third_lib: 第三方辅助函数
- | —— clustering: 地面去除&点云聚类代码
- | —— data_generator.py: 样本数据生成代码
- | —— detection.py: 完整的点云检测 pipeline 代码
- | —— dt: 检测结果文件