

RoboDrive Challenge Track3竞赛代码说明

代码百度网盘链接：链接: <https://pan.baidu.com/s/1GgsvH-hfXoEWxvAyONgbA?pwd=dkgt> 提取码: dkgt

1.项目说明

项目代码部分有3个文件夹，SurroundOcc主要完成算法训练，SurroundOcc_eval完成算法结果提交，tools部分完成phase2数据pkl文件生成及多模型结果融合。

本文档第一部分为项目整体说明，第二部分为算法原理说明，第三部分为算法训练与结果提交说明。

由于算法最终精度为多模型融合的结果，目前以单个模型为例说明算法(9.02分)流程，其他模型给出模型与log文件，如果需要，本文档后续会继续更新。

如果在结果复现过程中遇到bug或者精度不一致的情况，请邮件或微信(zhangbingyang1001)联系。

2.算法原理说明

#	SCORE	FILENAME	SUBMISSION DATE	SIZE (BYTES)	STATUS	✓	
1	9.63	occ3d_v1.zip	04/07/2024 06:31:59	46401	Finished		+
2	9.63	occ3d_v1.zip	04/07/2024 07:03:24	46578	Finished		+
3	8.0	vovnet-occ.zip	04/07/2024 09:48:01	47287	Finished		+
4	8.92	resnet_v3_epoch6.zip	04/28/2024 03:22:30	21790	Finished		+
5	9.94	fusion_v1.zip	04/28/2024 07:27:23	21875	Finished		+
6	9.4	baseline.zip	04/28/2024 08:09:12	21960	Finished		+
7	9.02	resnet_v1.zip	04/29/2024 02:08:45	22454	Finished		+
8	10.32	fusion_v2.zip	04/29/2024 04:52:41	22422	Finished	✓	+
9	10.05	fusion_v3.zip	04/29/2024 11:45:33	22640	Finished		+
10	10.14	fusion_v4.zip	04/30/2024 07:29:28	22894	Finished		+
11	8.96	resnet101_occ_v1_plus.zip	04/30/2024 13:58:44	23087	Finished		+

本次竞赛数据处理部分与官方给出的baseline代码保持一致，尝试了SurroundOcc与FB-Occ，SurroundOcc在验证集性能要高于FB-Occ(猜测原因为基于transformer的路线优于深度估计的路线)，在SurroundOcc部分，主要探索了三个方面的模型优化：

(1).SurroundOcc模型微调

在官方baseline的基础上微调，在微调时去掉多层Occ特征Loss，仅保留最后一层(200*200)的Loss，使用更小的学习率，Focal Loss, Lovasz_softmax Loss,新增unknow class label等方式，多次微调，主要是探索算法在超参数与loss微调后的性能提升，提交两个结果baseline.zip和occ3d_v1.zip,结果分别是**9.4**与**9.63**。

最后一次微调训练配置与log文件在SurroundOcc_eval/work_dirs/surroundocc文件夹。

(2).模型优化

本部分主要探索了不同网络结构的算法性能，主要是resnet101 + 最大voxel size(100,100,8)，resnet101 + 最大voxel size(200,200,16), vovnet-99 + 最大voxel size(100,100,8)的对比，提交三个模型resnet_v3_epoch6.zip, resnet_v1.zip, vovnet-occ.zip,分数分别为**8.92,9.02和8.0**。

上述三个模型都采用了训练+微调的模式，得出的结论为：vovnet-99在验证集性能高于resnet，但泛化能力较差；提升voxel size可以进一步提升算法性能；vovnet-99 backbone虽然结果较差，有助于多模型融合性能提升（第三部分说明）。

(3).多模型融合

本部分主要探索了多个模型融合对算法的性能提升：

第一个版本融合三个模型结果(occ3d_v1,vovnet-occ,resnet_v3_epoch6)，性能得到初步提升，取得分数:9.94;

第二个版本融合(1)与(2)部分的全部5个模型，取得分数10.32;

第三和四个版本探索vovnet对结果的影响，融合三个模型，其他两个相同，第三个分别为resnet和vovnet，分数分别为10.05和10.14，说明提升融合模型的多样性可提升模型的融合性能。

ToDoList

算法原理与细节进一步补充

3.算法训练与提交流程

本部分以第二部分resnet101 + 最大voxel size(200,200,16)的网络结构为例，给出算法训练与提交流程。

其他模型在SurroundOcc_eval/work_dirs有5个模型及其对应的配置.py文件和训练log文件(9.4分的配置与log文件丢失，可与surroundocc文件夹模型共用py文件)，根据3.2部分结果提交流程并替换.py文件与模型文件可获得对应模型的结果，最优结果为5个模型的融合。

3.1 数据准备

参考baseline代码，按照如下结构准备data文件夹数据，并将其软连接至SurroundOcc/data文件夹与SurroundOcc_eval文件夹，其中eval部分不需要nuscnescs数据集。

```
data
├── nuscnescs -> /home/zby/data/nuscnescs
├── nuscnescs_infos_train.pkl
├── nuscnescs_infos_val.pkl
├── nuscnescs_occ -> /home/zby/data/nuscnescs_occ
├── robodrive_infos_temporal_test_phase1.pkl
├── robodrive_infos_temporal_test.pkl
├── robodrive-phase2 -> /home/zby/data/robodrive-phase2
└── robodrive-release -> /home/zby/data/robodrive-release

3 directories, 5 files
```

3.2 算法流程

参考<https://github.com/weiyithu/SurroundOcc> 或<https://github.com/robodrive-24/toolkit/tree/main/track-3> 准备训练环境;

训练

进入SurroundOcc文件夹，训练resnet101 + 最大voxel size(200,200,16)模型：

```
./tools/dist_train.sh ./projects/configs/resnet101/resnet101_occ_v1.py 2
./work_dirs/resnet101
```

在train.sh里，记录了其他方法的训练脚本。

结果提交

进入SurroundOcc_eval文件夹，获得提交测试结果：

```
./tools/dist_submit.sh work_dirs/resnet101/resnet101_occ_v1.py  
work_dirs/resnet101/latest.pth 1
```

将生成的pred.pkl文件使用zip指令压缩后提交，在submit.sh里，记录了其他方法的提交流程。

结果融合

在tools文件夹，运行结果融合脚本：

```
python submit_fusion.py
```

可以修改pkl_list尝试其他组合的融合方法。

ToDoList

全部模型的训练及提交模型补全；

训练log描述。