Nuscenes数据集总结

一. 简介

nuScenes 数据集是由Motional（前身为 nuTonomy）团队开发的自动驾驶公共大规模数据集。在波士顿和新加坡收集了 1000 个驾驶场景.

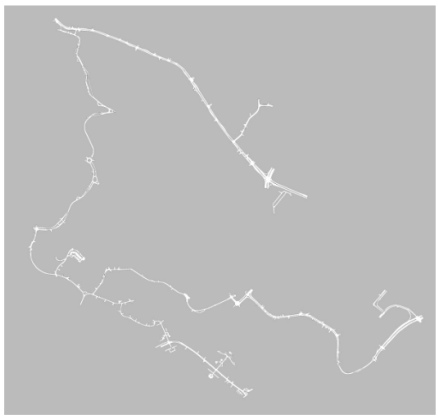
nuScenes 数据集的灵感来自开创性的KITTI数据集。nuScenes 是第一个提供来自自动驾驶汽车的整个传感器套件（6 个摄像头、1 个激光雷达、5 个雷达、GPS、IMU）的数据的大规模数据集。与 KITTI 相比，nuScenes 包含 7 倍以上的对象注释。

二.数据采集

1.场景规划

在波士顿和新加坡收集了大约 15 小时的驾驶数据。对于完整的 nuScenes 数据集，发布了来自波士顿海港和新加坡 One North、皇后镇和荷兰村地区的数据。

波士顿海港



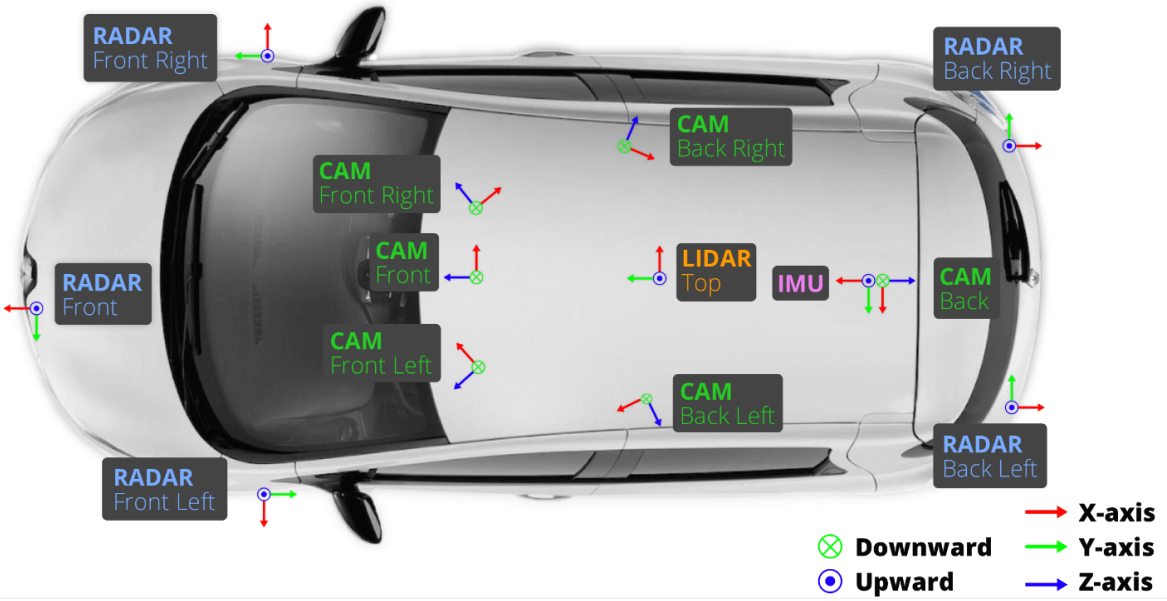
新加坡皇后镇

新加坡一北

新加坡荷兰村

2.汽车设置

使用两辆具有相同传感器布局的雷诺 Zoe汽车在波士顿和新加坡行驶。数据是从研究平台收集.



相机（CAM）有六个，分别分布在前方（Front）、右前方（Front Right）、左前方（Front Left）、后方（Back）、右后方（Back Right）、左后方（Back Left）；激光雷达（LIDAR）有1个，放置在车顶（TOP）；毫米波雷达有五个，分别放置在前方（Front）、右前方（Front Right）、左前方（Front Left）、右后方（Back Right）、左后方（Back Left）。

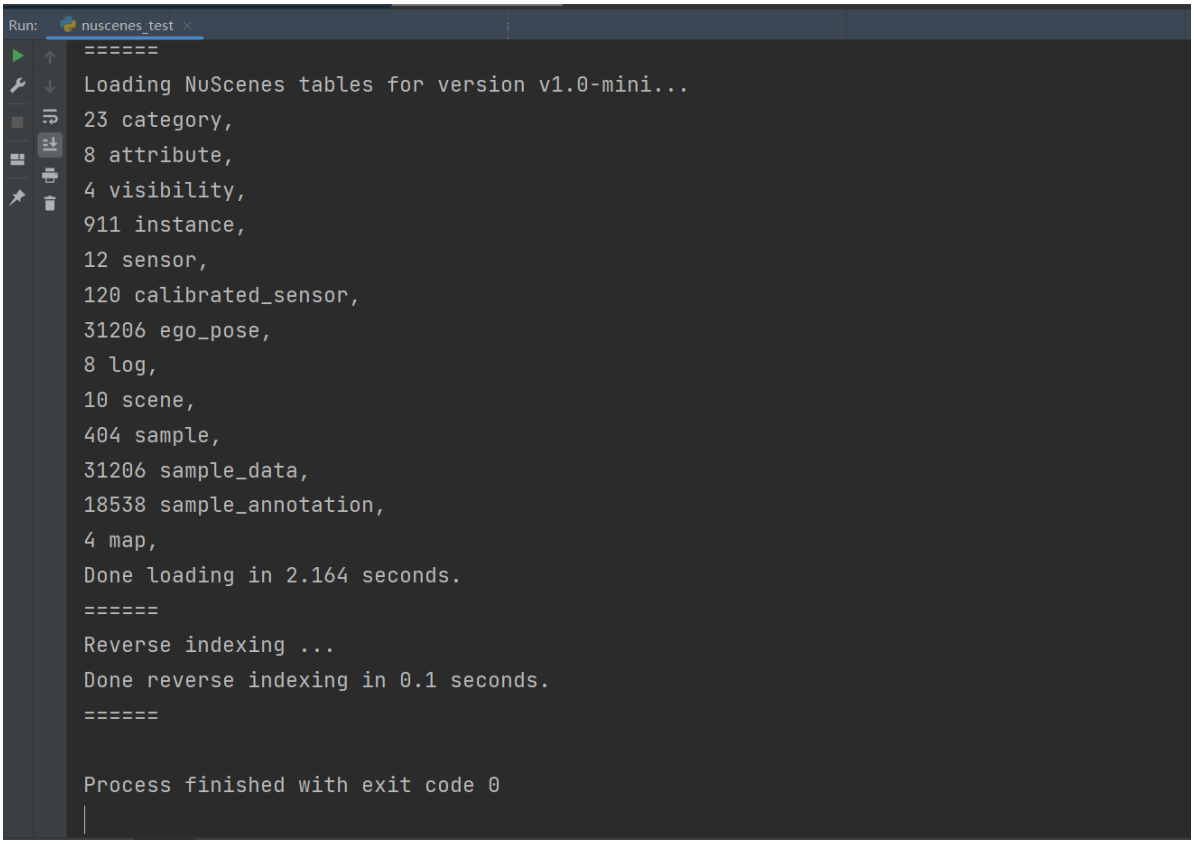
三.数据集的使用

1.导入nuscenes-devkit库

```
pip install nuscenes-devkit
```

2.加载数据集信息

```
from nuscenes.nuscenes import NuScenes
nusc = NuScenes(version='v1.0-mini', dataroot='数据集的具体路径', verbose=True)
```



3.场景scene

使用如下代码，查看当前数据集中所有的场景

```
nusc.list_scenes()
```

```
scene-0061, Parked truck, construction, intersectio... [18-07-24 03:28:47]   19s, singapore-onenorth, #anns:4622
scene-0103, Many peds right, wait for turning car, ... [18-08-01 19:26:43]   19s, boston-seaport, #anns:2046
scene-0655, Parking lot, parked cars, jaywalker, be... [18-08-27 15:51:32]   20s, boston-seaport, #anns:2332
scene-0553, Wait at intersection, bicycle, large tr... [18-08-28 20:48:16]   20s, boston-seaport, #anns:1950
scene-0757, Arrive at busy intersection, bus, wait ... [18-08-30 19:25:08]   20s, boston-seaport, #anns:592
scene-0796, Scooter, peds on sidewalk, bus, cars, t... [18-10-02 02:52:24]   20s, singapore-queensto, #anns:708
scene-0916, Parking lot, bicycle rack, parked bicyc... [18-10-08 07:37:13]   20s, singapore-queensto, #anns:2387
scene-1077, Night, big street, bus stop, high speed... [18-11-21 11:39:27]   20s, singapore-hollandv, #anns:890
scene-1094, Night, after rain, many peds, PMD, ped ... [18-11-21 11:47:27]   19s, singapore-hollandv, #anns:1762
scene-1100, Night, peds in sidewalk, peds cross cro... [18-11-21 11:49:47]   19s, singapore-hollandv, #anns:935

Process finished with exit code 0
```

使用如下代码查看具体某个场景的信息

```
my_scene = nusc.scene[0]
print(my_scene)
```

```
{'token': 'cc8c0bf57f984915a77078b10eb33198', 'log_token': '7e25a2c8ea1f41c5b0da1e69ecfa71a2', 'nbr_samples': 39, 'first_

Process finished with exit code 0
```

4.样本sample

每个scene大约持续20s，那sample就是每0.5秒进行一次采样。也可以这样理解sample和scene，sence相当于20s的视频，sample就是每0.5s取一帧的图像。

使用如下代码，获取具体场景的具体的一个sample的token值

```
first_sample_token = my_scene['first_sample_token']  #获取第一个sample的token值
print(first_sample_token)
```

```
ca9a282c9e77460f8360f564131a8af5

Process finished with exit code 0
```

使用如下代码，通过sample的token值获取sample的具体信息

```
my_sample = nusc.get('sample', first_sample_token)
print(my_sample)
```

```
{'token': 'ca9a282c9e77460f8360f564131a8af5', 'timestamp': 1532402927647951, 'prev': '', 'next': '39586f9d59004284a7114a
Process finished with exit code 0
```

5.样本数据 sample_data

可视化前方的毫米波雷达传感器

通过如下代码，获取sample的data数据

```
print(my_sample['data'])
```

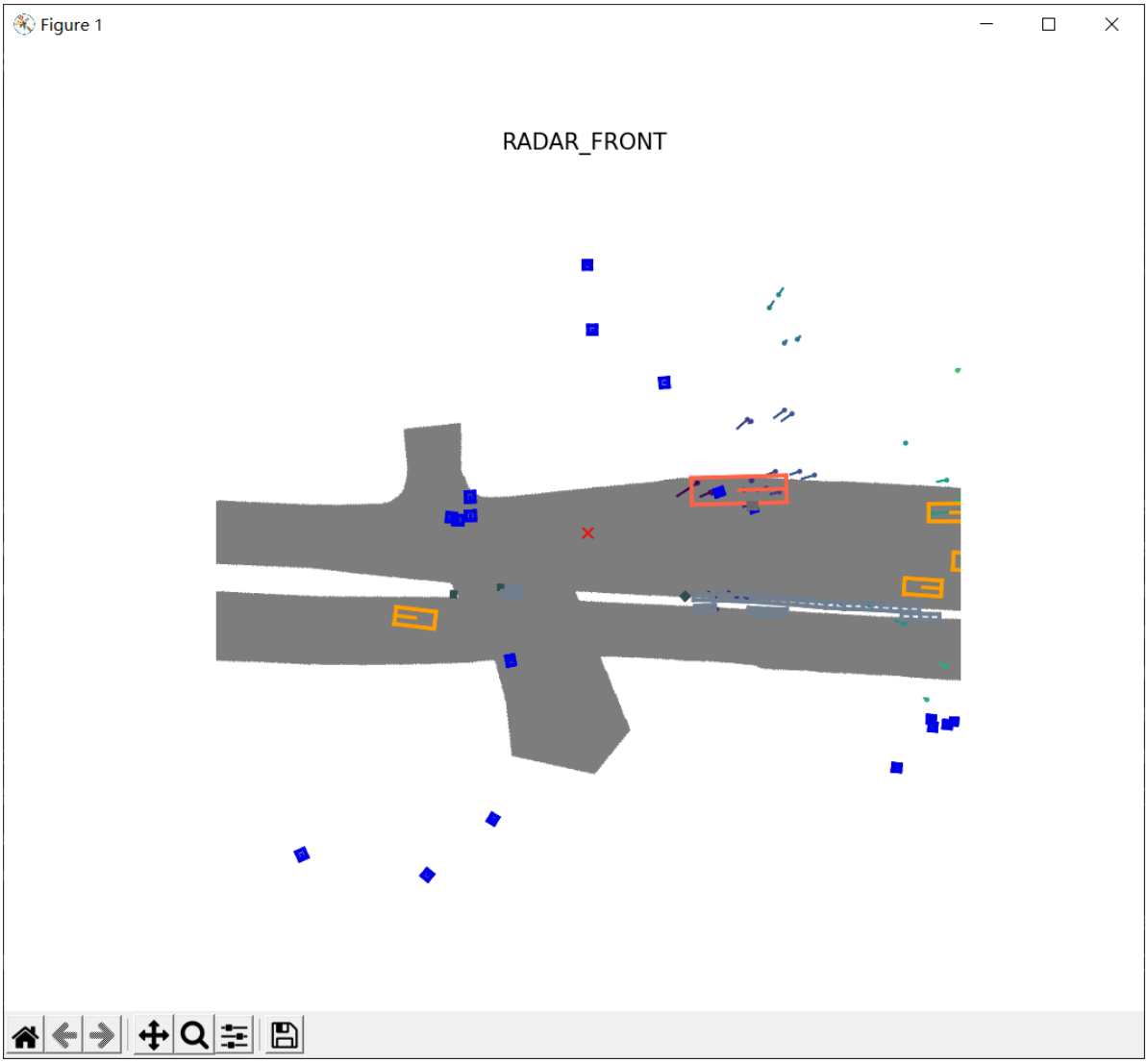
```
{'RADAR_FRONT': '37091c75b9704e0daa829ba56dfa0906', 'RADAR_FRONT_LEFT': '11946c1461d14016a322916157da3c7d', 'RADAR_FRONT_
Process finished with exit code 0
```

(1)获取具体sample中的具体传感器的信息

```
sensor_radar = 'RADAR_FRONT'  #这里选择的传感器为前方的毫米波雷达传感器
radar_front_data = nusc.get('sample_data',my_sample['data'][sensor_radar])
print(radar_front_data)
```

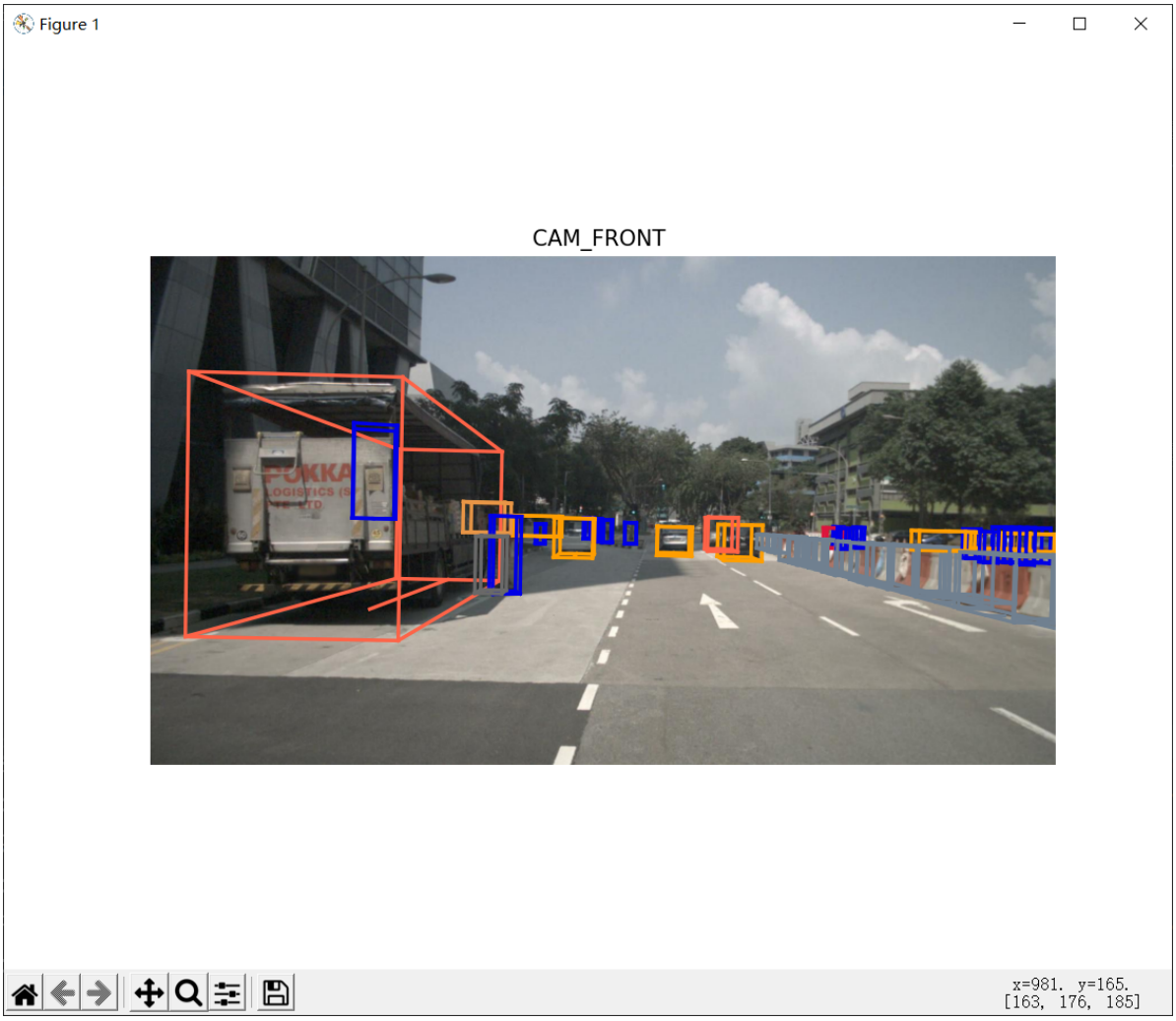
(2)通过传感器信息的token值进行可视化

```
nusc.render_sample_data(radar_front_data['token'])
```



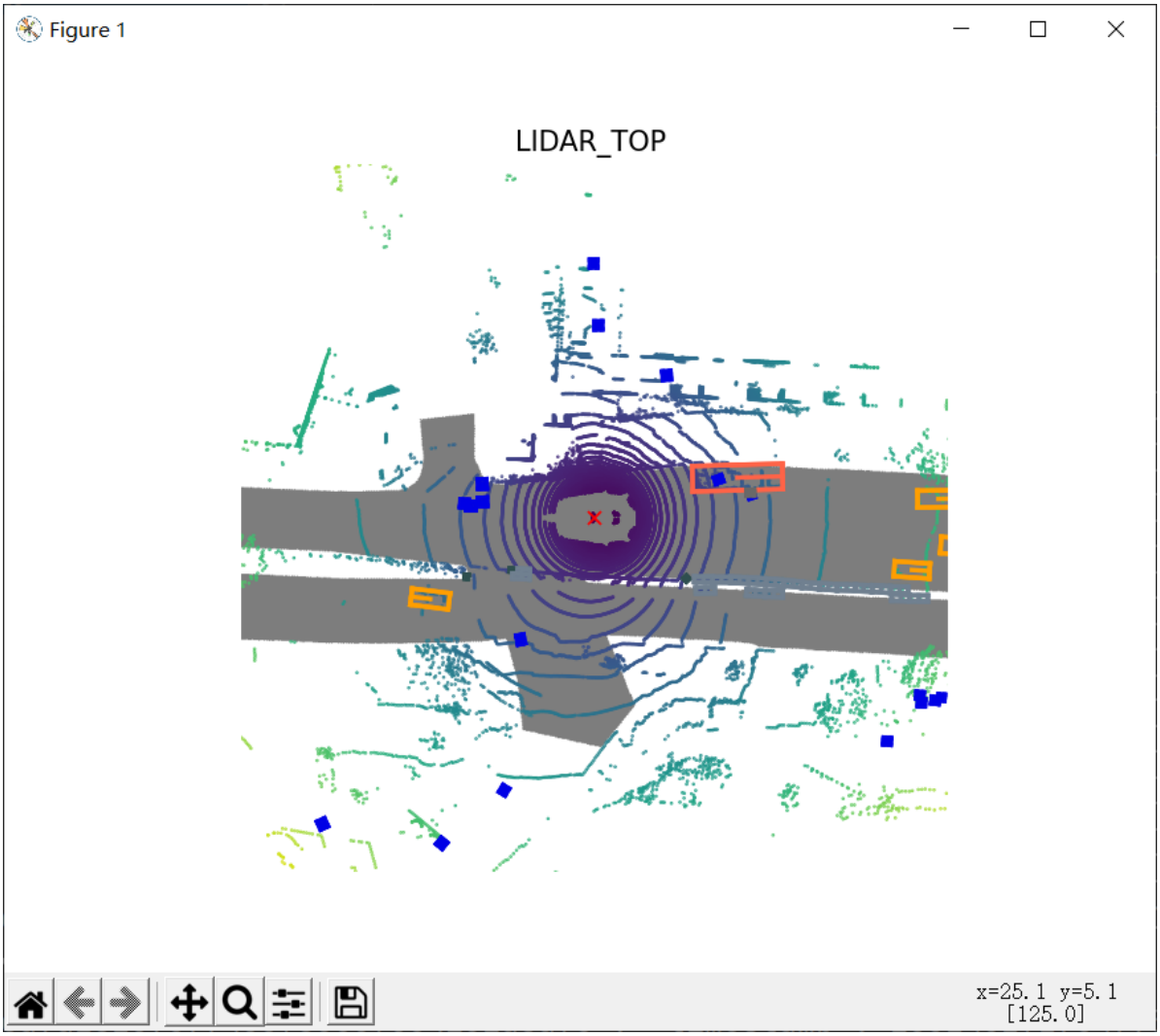
可视化前方的相机

```
# 可视化前方的相机
sensor_CAM_FRONT = 'CAM_FRONT'  #这里选择的传感器为前方的毫米波雷达传感器
CAM_FRONT_data = nusc.get('sample_data', my_sample['data'][sensor_CAM_FRONT])
print(CAM_FRONT_data)
nusc.render_sample_data(CAM_FRONT_data['token'])
```



可视化顶部激光雷达

```
# 可视化顶部激光雷达
sensor_LIDAR_TOP = 'LIDAR_TOP'  #这里选择的传感器为前方的毫米波雷达传感器
LIDAR_TOP_data = nusc.get('sample_data', my_sample['data'][sensor_LIDAR_TOP])
print(LIDAR_TOP_data)
nusc.render_sample_data(LIDAR_TOP_data['token'])
```



6.样本标注 sample_annotation

在sample_data中已经展示了传感器采集到的信息，这一部分将展示样本标注的信息，方法与之前是类似的。

(1)获取sample的标注数据，然后输出相关信息

```
my_annotation_token = my_sample['anns'][18]
my_annotation_metadata = nusc.get('sample_annotation',my_annotation_token)
my_annotation_metadata
```

(2)通过sample标注信息的token值，进行可视化

```
# 获取特定sample的标注信息
my_annotation_token = my_sample['anns'][18]
my_annotation_metadata = nusc.get('sample_annotation', my_annotation_token)
print(my_annotation_metadata)
# 可视化
nusc.render_annotation(my_annotation_metadata['token'])
time.sleep(3)
```

7.实例 instance

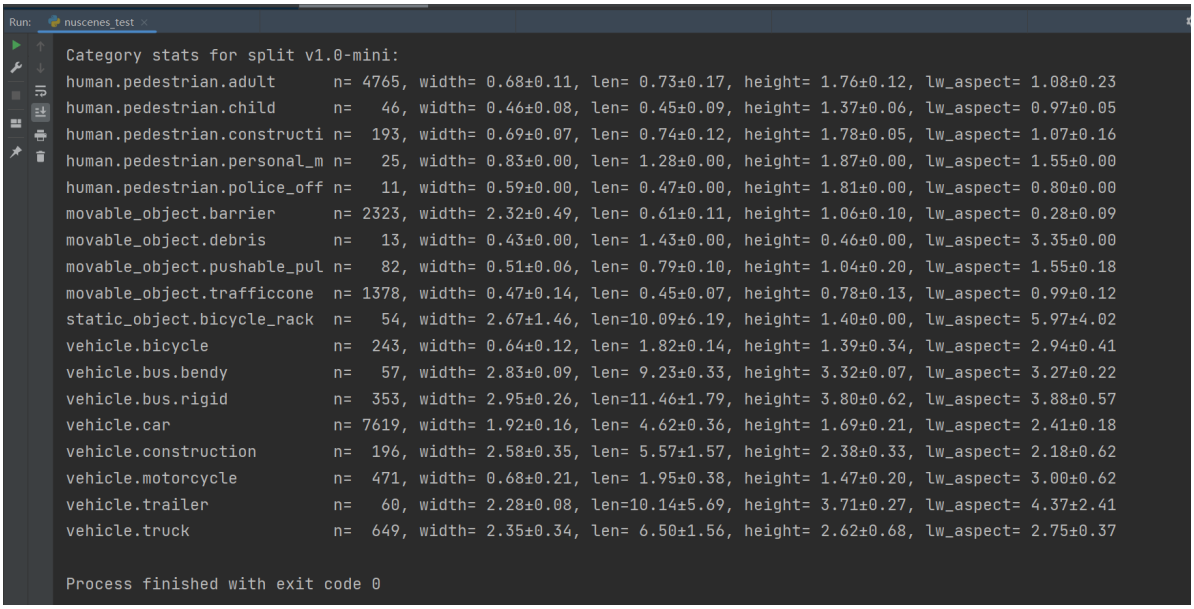
通过nusc.instance[0]获取具体的实例，instance表示某个实例对象，例如一辆汽车

```
# 获取某个实例对象，并输出其信息
my_instance = nusc.instance[5]
print(my_instance)
# 通过实例的token值，进行可视化
instance_token = my_instance['token']
nusc.render_instance(instance_token)
time.sleep(3)
```

8.类别categories

使用如下代码展示数据集中所有的种类

```
nusc.list_categories()
```



9.属性attributes

通过如下代码展示数据集的具体属性

```
nusc.list_attributes()
```

```
cycle.with_rider: 305
cycle.without_rider: 434
pedestrian.moving: 3875
pedestrian.sitting_lying_down: 111
pedestrian.standing: 1029
vehicle.moving: 2715
vehicle.parked: 4674
vehicle.stopped: 1545

Process finished with exit code 0
```

10.可视化 visibility

```
# 选取当前sample标注信息中的一个token值
anntoken = my_sample['anns'][9]
nusc.render_annotation(anntoken)
time.sleep(3)
```

11.传感器sensor

通过下面代码展示数据集中传感器信息

```
print(nusc.sensor)
```

```
[{'token': '725903f5b62f56118f4094b46a4470d8', 'channel': 'CAM_FRONT', 'modality': 'camera'}, {'token': 'ce89d4f3050b5892

Process finished with exit code 0
```

12.校准传感器 calibrated_sensor

通过下面代码获取传感器的校准信息

```
# 展示传感器的校准信息
sensor_token = nusc.calibrated_sensor[0]
print(sensor_token)
```

```
{'token': 'f4d2a6c281f34a7eb8bb033d82321f79', 'sensor_token': '47fcd48f71d75e0da5c8c1704a9bfe0a', 'translation': [3.412,

Process finished with exit code 0
```

13.车辆姿态 ego_pose

通过下面代码获取车辆姿态信息

```
# 车辆姿态ego_pose信息
print(nusc.ego_pose[0])
```

```
{'token': '5ace90b379af485b9dcb1584b01e7212', 'timestamp': 1532402927814384, 'rotation': [0.5731787718287827, -0.00158116

Process finished with exit code 0
```

14.日志 log

通过下面代码获取日志信息

```
# 日志信息
print(nusc.log[0])
```

```
{'token': '7e25a2c8ea1f41c5b0da1e69ecfa71a2', 'logfile': 'n015-2018-07-24-11-22-45+0800', 'vehicle': 'n015', 'date_captur

Process finished with exit code 0
```

15.地图 map

```
# 地图信息
print(nusc.map[0])
```

```
{'category': 'semantic_prior', 'token': '53992ee3023e5494b90c316c183be829', 'filename': 'maps/53992ee3023e5494b90c316c183

Process finished with exit code 0
```