

차량기능기초

과제 1



국민대학교
KOOKMIN UNIVERSITY

학번 : 20161298

학과 : 자동차공학과

이름 : 김 태 현

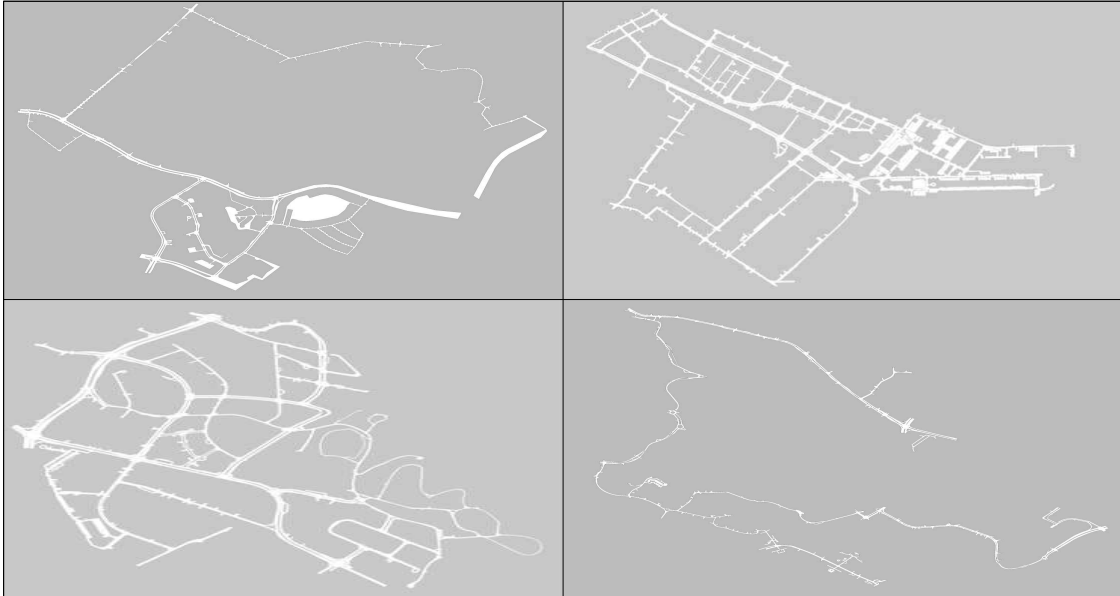
1) 자율주행 인지에 관련된 3종 이상의 공개 Data Set 조사, 정리

누씬(nuScene) Data Set

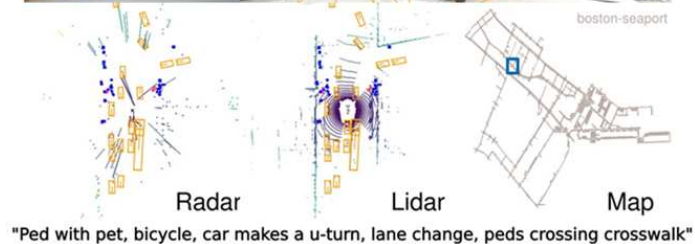
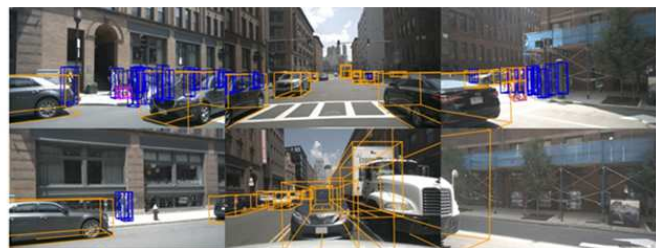
누씬은 모셔널 회사에 의해 자율주행을 위한 공공 대규모 data set이다.

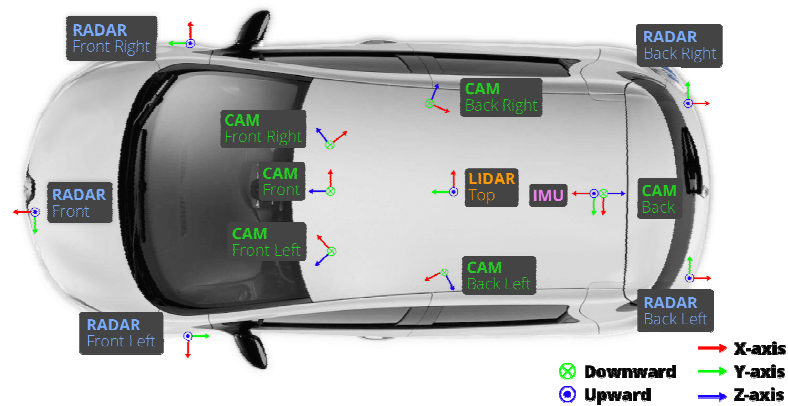
교통신호가 밀집되어 있고 운전하기 어려운 상황들이 있는 보스턴과 싱가포르에서 1000가지의 driving scene을 수집하였다.

우선 Scene planning 수집항목이 있다.



위 사진들은 포스턴의 해변항구 및 싱가포르의 one North, Queenstown 과 Holland Village에서의 장면이다. 위치, 시간, 날씨조건 등 다양한 조건에 의한 까다로운 시나리오를 포착하기 위해 신중히 검열된 운전 경로이다. 자전거 등과 같은 차종 빈도 분포의 균형을 위해 더 많은 Scene들을 포함시키고 있다. 각각 20초마다 1000개의 Scene들을 전문가들이 수동으로 선별합니다. 이 1000개의 주행 장면은 라이다를 통한 이미지 분할 기술을 적용한 것으로 14억개의 포인트를 포함하고 있다. 또한 위에 언급한 대로 무단횡단, 대형 교차로, 약천 후, 야간 주행, 강렬한 햇빛 등 까다로운 시나리오에 직접 일일이 주석을 단 데이터가 누이미지스이다.





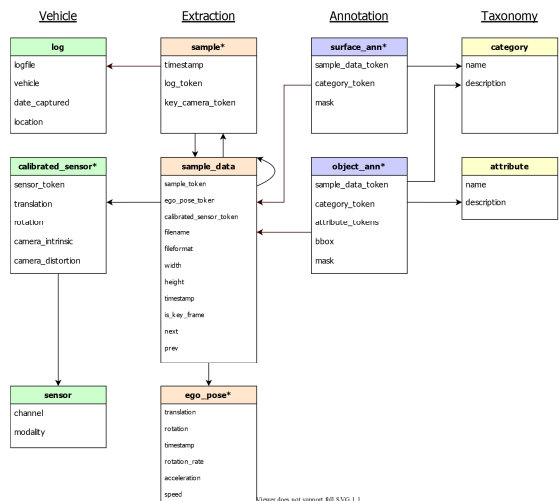
자료를 수집하기 위해 Apiz 는
 Spinning LIDAR x 1
 Long range RADAR sensor x5
 camera x6를 사용하였다. 6개의 메라는 360도를 볼수 있게끔 설치되어있다.

Driving Data를 수집한 이후, 주석 전문가에게 2Hz 로 동기화된 키프레임들(image, LIDAR, RADAR) 왼쪽 사진과 같이 샘플화합니다.

Category	nuScenes cuboids	Cuboid ratio	Lidarseg points	Point ratio
animal	787	0.07%	5,385	0.01%
human.pedestrian.adult	208,240	17.86%	2,156,470	2.73%
human.pedestrian.child	2,066	0.18%	9,655	0.01%
human.pedestrian.construction_worker	9,161	0.79%	139,443	0.18%
human.pedestrian.personal_mobility	399	0.03%	8,723	0.01%
human.pedestrian.police_officer	727	0.06%	9,158	0.01%
human.pedestrian.stroller	1,072	0.09%	8,809	0.01%
human.pedestrian.wheelchair	503	0.04%	12,168	0.02%
movable_object.barrier	152,087	13.04%	9,305,106	11.79%
movable_object.debris	3,016	0.26%	66,661	0.08%
movable_object.pushable_pullable	24,605	2.11%	718,641	0.91%
movable_object.trafficcone	97,959	8.40%	736,239	0.93%
static_object.bicycle_rack *	2,713	0.23%	163,126	0.21%

nuImages schema

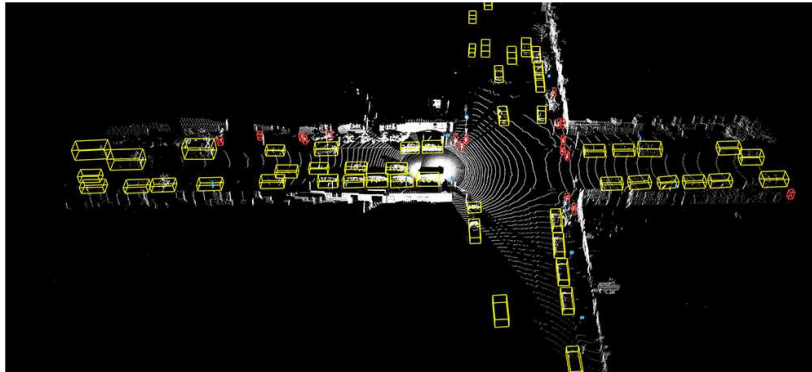
Asterisks (*) indicate modifications compared to the nuScenes schema.



교정, 지도, 차량 좌표 같은 메타 데이터와 모든 주석들은 관계적인 데이터베이스에 포함됩니다. 오른쪽을 보면 데이터베이스 테이블을 볼 수 있는데 각 행은 고유한 기본키인 토큰으로 식별할 수 있다. sample_token 같은 외래키는 sample 테이블의 토큰으로 링크할 수 있다.

카테고리가 똑같이 유지하는 동안 속성이 변경할 수 있는 것이 인스턴스의 속성이다. 예를 들면, 주차/정지/이동중인 차량이나 자전거에 운전자가 있는지 없는지 바뀔 수 있는 것이다. nuImage의 속성은 nuScenes의 속성의 상위 집합이다.

Waymo Open DataSet



웨이모는 10만 개 이상의 세그먼트에 대한 객체배적 및 해당 3D 맵을 사용하여, 각 20초씩 길고 흥미로운 상호 작용을 위해 마이닝된 새로운 모션 Dataset에는 570 시간 이상의 고유 데이터가 포함되어 있습니다. 웨이모는 다양한 도시 환경에서 24시간 내내 캡처된 다양한 도로 유형과 주행 조건을 특징으로 가진 모션 데이터 세트입니다.

웨이모는 3DLidar 레이블을 사용합니다. 라이더 레벨은 차량 프레임에 있는 3D 7-DOF 경계 상자이며 전역적으로 고유한 추적 ID가 있습니다. 경계상자는 피치가 0, 롤이 0 이다. 차량의 전방측과 정렬하기 위해 Z 축을 중심으로 차량 프레임 +X 축을 회전하는데 필요한 각도가 방향입니다.

2D 카메라 라벨은 카메라 이미지에 2D 경계 상자 레이블을 제공합니다. 카메라 레이블은 전 세계적으로 전 세계적으로 고유한 추적 ID가 있는 꼭 맞는 축 정렬 2D경계 상자이다. 경계상자는 개체가 보이는 부분에만 덮여있습니다.

웨이모의 데이터형식은 글로벌 프레임/차량프레임/센서프레임/Lidar 구면 좌표가 있습니다.

글로벌 프레임은 차량의 시작 위치로 원점이 설정됩니다. Up(z) 는 중력 벡터와 정렬되고 양의 위쪽을 향합니다. East(x) 는 도선을 따라 직접 동쪽을 가리키고 north(y) 는 북극을 가리킵니다.

차량 프레임은 x축은 앞쪽으로, y축은 왼쪽으로, z축은 위쪽으로 양수입니다. 차량 포즈는 차량 프레임에서 글로벌 프레임으로의 변환을 정의합니다.

센서 프레임은 카메라 프레임을 카메라 렌즈의 중앙에 위치시켜 X축은 렌즈에서 렌즈 배열 아래를 가리키고 Z축이 위 쪽을 가리킵니다. y/z 평면은 카메라 평면과 평행합니다.

Lidar 구면 좌표계는 LIDAR 센서 프레임의 데카르트 좌표계를 기반으로 합니다. 라이더 데카르트 좌표의 점(X,Y,Z) 은 라이더 구면 좌표의 (범위,방위각, 경사) 튜플로 고유하게 변환될 수 있습니다.

Lidar 데이터 세트에는 5개의 라이더(중간 거리 라이더(위쪽) 1개와 단거리 라이더 4개의 데이터가 포함됩니다. 이 때 최대 75미터로 잘린 중거리 라이더의 범위, 25미터로 잘린 단거리 라이더의 범위, 총 5개의 모든 라이더에 대해 가장 강력한 2개의 강도 리턴이 제공되는 제한 상이 적용되어 있습니다. 각 라이더의 포인트 클라우드는 범위 이미지로 인코딩 되고 각 라이더에 대해 두 개의 가장 강력한 수직을 각각에 대해 두 개의 범위 이미지가 제공됩니다. 채널 0은 범위, 채널1 은 라이더 강도, 채널2은 라이더 신장, 채널3은 is_in_nlz로 총 4개의 채널이 있습니다. 여기서 라이더 신장은 공칭 폭을 초과하는 펄스의 신장을 나타냅니다.

기본 4채널 외에도 라이더 투 카메라 프로젝션을 위한 또 다른 6채널을 제공하는데 채널0은 카메라 이름 채널1 은 x 채널2 는 y 채널 3은 두 번째 프로젝션의 카메라 이름 채널 4은 x (이미지 너비에 따른 축) 채널5은 y (이미지 높이에 따른 축)입니다.

카메라 데이터는 5개의 서로 다른 방향과 관련된 5개의 카메라 이미지가 포함되어 있습니다.

전면, 전면 왼쪽, 전면 오른쪽, 측면 오른쪽 및 측면 오른쪽입니다.

각 카메라 이미지는 jpeg 형식으로 하나씩 제공됩니다. 이미지 바이트 외에도 차량포즈, 이미지 센서의 노출 시간에 해당하는 속도 및 롤링 셔터 타이밍 정보도 제공합니다. 이러한 정보는 카메라 프로젝션에 대한 라이더 사용자 정의하는 데에 유용합니다.

Ford & Volkswagen Argoverse

아르고버스는 위의 모셔널 누씬처럼 자율주행 연구를 위한 아르고 AI의 Open Data Set 이다. 아르고버스는 3가지 데이터 유형으로 나뉘어있다.

-3D 추적 데이터세트(Argoverse 3D tracking dataset)

초당 30프레임 영상과 정면용 스테레오 이미지, 장거리 라이다로부터 들어온 포인트클라우드 데이터를 360도 시야가 결합된 총 7대의 카메라로부터 수집한다.

15~30초 정도의 주행 장면이 113개가 있고, 물체를 약 10000개 대상에 모두 3D 추적 라벨링 작업을 한다.

- 움직임 예측 데이터세트(Argoverse motion forecasting dataset)

운전을 하다보면 옆에서 끼어드는 차량이 있으면 속도를 늦추거나, 커브 후 속도를 다시 높이고, 보행자가 지나갈 때에는 정지하는 등 3D 트래킹 하는 대상 이동 방향을 예측하는 시나리오 데이터이다.

각 시나리오 데이터는 추적 대상에 대한 5초 분량의 2D 이미지 버드 아이뷰로 되어있다. 2초는 알고리즘에 입력한 시간이고 다음 3초간은 추적 대상 움직임을 예측하게 하는 식으로 학습된다.

-정밀지도 (Argoverse high-definition maps)

정밀지도는 290km의 고화질 지도 데이터 세트로서 차량이 교차로에 진입하게 되면, 교차로를 지난 다음 어떠한 후속 차선을 따라가야 하는지 알려줄 수 있을 정도로 이해도가 높다. 또한 지면의 높낮이를 인식할 수 있고, 주행 가능 영역인지도 판별이 가능하다.

2) 자율주행 인지에 관련된 2종 이상 Open Source 조사, 정리

다음은 모셔널의 nuScenes- lidarseg tutorial 의 open source 정리이다.

```
!mkdir -p /data /sets /nusenes
# Make the directory to store the nuScenes dataset in.
/data/sets/nusenes 의 경로에 파일을 만드는 명령이다.
!wget https://www.nusenes.org/data/v1.0-mini.tgz
# Download the nuScenes mini split.
nuScenes mini split을 저 홈페이지를 통해 다운로드 한다.
!wget https://www.nusenes.org/data/nuScenes-lidarseg-mini-v1.0.tar.bz2
# Download the nuScenes-lidarseg mini split.
마찬가지로 제가 조사해온 open source인 lidarseg tutorial의 mini split을 다운로드 한다.
!tar -xvf v1.0-mini.tgz -C /data/sets/nusenes
# Uncompress the nuScenes mini split.
!tar -xvf nuScenes-lidarseg-mini-v1.0.tar.bz2 -C /data/sets/nusenes
# Uncompress the nuScenes-lidarseg mini split.
위 두 명령어는 다운로드 한 것들을 압축풀기 하는 것이다.
!pip install nusenes-devkit &>/dev/null
# Install nuScenes.
이제 nuScenes를 설치한다. 여기까지 nuScenes-lidarseg tutorial을 하기 위한 환경구축을 하였다.
```

%matplotlib inline

from nusenes import NuScenes

nusc = NuScenes (version = 'v1.0-mini', dataroot = '/data/sets/nusenes', verbose = **True**)

위 명령어는 필요한 library들을 추출하는 과정이다. 이때 %matplotlib inline은 그래프 표시를 가능케 하는 라이브러리 명령어이다.

nusc .list_lidarseg_categories (sort_by = 'count')

count 로 지정하였으므로 포인트의 수에 따라 클래스들이 올림차순으로 분류된다.

nusc .lidarseg_idx2name_mapping

위 명령어는 맨 왼쪽 열을 보고 각 클래스 이름이 속한 인덱스를 얻을 수 있다. 실행시켜보면

```
{0: 'noise',
1: 'animal',
2: 'human.pedestrian.adult',
3: 'human.pedestrian.child',
4: 'human.pedestrian.construction_worker',
5: 'human.pedestrian.personal_mobility',
6: 'human.pedestrian.police_officer',
7: 'human.pedestrian.stroller',
8: 'human.pedestrian.wheelchair',
9: 'movable_object.barrier',
```

```

10: 'movable_object.debris',
11: 'movable_object.pushable_pullable',
12: 'movable_object.trafficcone',
13: 'static_object.bicycle_rack',
14: 'vehicle.bicycle',
15: 'vehicle.bus.bendy',
16: 'vehicle.bus.rigid',
17: 'vehicle.car',
18: 'vehicle.construction',
19: 'vehicle.emergency.ambulance',
20: 'vehicle.emergency.police',
21: 'vehicle.motorcycle',
22: 'vehicle.trailer',
23: 'vehicle.truck',
24: 'flat.driveable_surface',
25: 'flat.other',
26: 'flat.sidewalk',
27: 'flat.terrain',
28: 'static.manmade',
29: 'static.other',
30: 'static.vegetation',
31: 'vehicle.ego'

```

이런식으로 1부터 시작해서 각 클래스를 얻을 수 있다.

반대로 nusc .lidarseg_name2idx_mapping 은

```

'animal': 1,
'flat.driveable_surface': 24,
'flat.other': 25,
'flat.sidewalk': 26,
'flat.terrain': 27,
'human.pedestrian.adult': 2,
'human.pedestrian.child': 3,
'human.pedestrian.construction_worker': 4,
'human.pedestrian.personal_mobility': 5,
'human.pedestrian.police_officer': 6,
'human.pedestrian.stroller': 7,
'human.pedestrian.wheelchair': 8,
'movable_object.barrier': 9,
'movable_object.debris': 10,
'movable_object.pushable_pullable': 11,
'movable_object.trafficcone': 12,
'noise': 0,
'static.manmade': 28,
'static.other': 29,
'static.vegetation': 30,
'static_object.bicycle_rack': 13,
'vehicle.bicycle': 14,
'vehicle.bus.bendy': 15,
'vehicle.bus.rigid': 16,
'vehicle.car': 17,
'vehicle.construction': 18,
'vehicle.ego': 31,
'vehicle.emergency.ambulance': 19,
'vehicle.emergency.police': 20,
'vehicle.motorcycle': 21,
'vehicle.trailer': 22,
'vehicle.truck': 23

```

위와 같이 인덱스에 대한 클래스 이름으로 매핑을 가져올 수 있다.

예를 들어 샘플을 고르면 다음과 같이 명령어를 쓰면 된다.

```
my_sample = nusc .sample [ 87 ]
```

```
nusc .get_sample_lidarseg_stats (my_sample ['token'], sort_by = 'count')
```

포인트 클라우드에 클래스를 가지고 위와 같은 명령어를 써서 클래스와 각 빈도수를 오름차순으로 인쇄한다.

```

sample_data_token = my_sample ['data'] ['LIDAR_TOP']
nusc .render_sample_data (sample_data_token ,
                           with_anns = False ,
                           show_lidarseg = True )

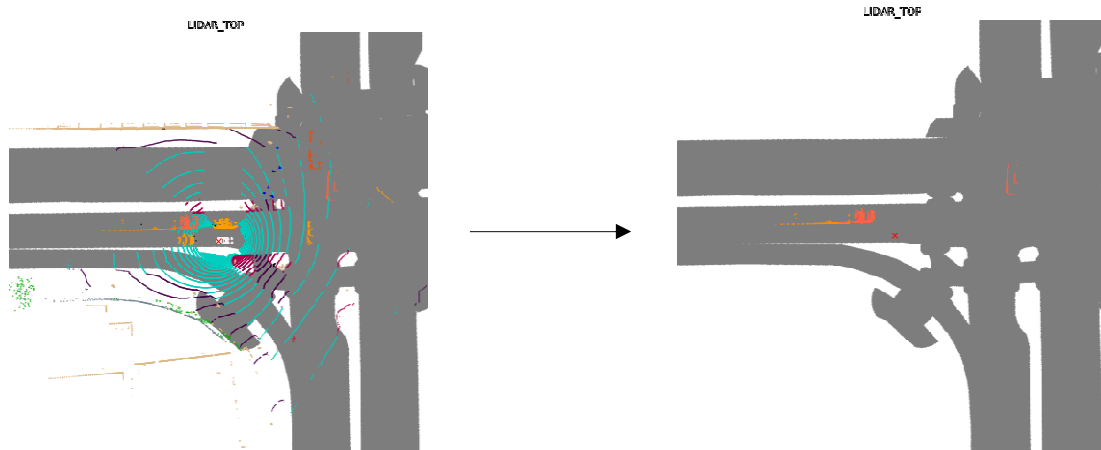
```

라이다의 TOP의 샘플을 가지고 와서 차량으로부터의 거리에 따라 색을 입혀준다.

또한, show_lidarseg=True 로 설정함으로써 포인트클라우드에서의 클래스 라벨을 시각화시켜준다.

```
nusc.render_sample_data (sample_data_token ,
                        with_anns =False ,
                        show_lidarseg =True ,
                        filter_lidarseg_labels =[ 22 ,23 ])
```

마지막에 filter_lidarseg_labels=[22,23] 은 특정 인덱스만 부르고 나머지는 보고 싶지 않을 때 사용하여 필터링한 것이라고 이해할 수 있다.



```
nusc.render_sample_data (sample_data_token ,with_anns =False ,show_lidarseg
= True ,show_lidarseg_legend =True )
```

이것은 범례를 추가하여 각 점 색깔에 맞는 인덱스를 보여주게 한다.

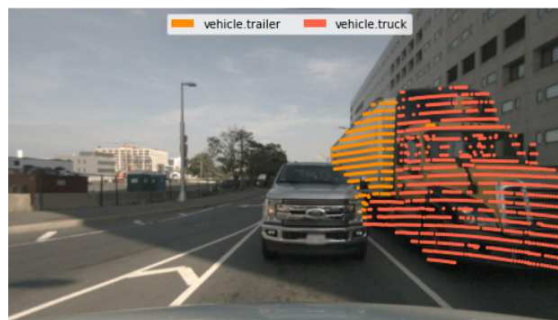
이미지 사진으로 렌더링을 하기 위해서는 다음과 같이 쓴다.

```
nusc.render_pointcloud_in_image (my_sample [ 'token' ],
                                pointsensor_channel ='LIDAR_TOP',
                                camera_channel ='CAM_BACK',
                                render_intensity =False ,
                                show_lidarseg =True ,
                                filter_lidarseg_labels =[ 22 ,23 ],
                                show_lidarseg_legend =True )
```

2번째 줄은, 라이다 탐에 있는 채널을 맞추는 명령어

3번째 줄은 카메라 뒤쪽에 있는 채널을 맞추는 명령어

nuScenes devkit 원본에는 모든 셋팅이 되어있기 때문에 render_intensity는 False로 하되 show_lidarseg= True 로 하여 분류될수 있게끔 한다. 위와 같은 명령어를 입력하게 되면 트럭과 트레일러를 가르키는 라벨을 볼수 있다.




```
!rm -rf waymo-od > /dev/null
!git clone https://github.com/waymo-research/waymo-open-dataset.git waymo-od
!cd waymo-od && git branch -a
!cd waymo-od && git checkout remotes/origin/master
!pip3 install --upgrade pip
!pip3 install waymo-open-dataset-tf-2-1-0==1.2.0
import os
import tensorflow.compat.v1 as tf
import math
import numpy as np
import itertools

tf.enable_eager_execution()

from waymo_open_dataset.utils import range_image_utils
from waymo_open_dataset.utils import transform_utils
from waymo_open_dataset.utils import frame_utils
from waymo_open_dataset import dataset_pb2 as open_dataset

먼저 dataset 패키지를 설치하고 환경구축을 한다.
```

파일 이름을 설정하고 **Dataset** 형식을 결정짓기 위해 두 개의 프레임을 추출한다.

3) 2)의 정리한 코드 중 하나 실행해서 결과 확인

저는 VM ware Workstation 16 player을 이용하여 유튜브를 이용하였습니다.

nuScenes- lidarseg tutorial 다운로드 경로는 ~/data/sets/nuscenes 으로 설정하였습니다.

