

## 자율주행 인지 관련 Dataset 조사 및 분석

차량지능기초

이재구 교수

소프트웨어학부 2학년

20203401 이희주

2021.04.07 제출

### 1. 자율주행 인지와 관련된 공개 DataSet

#### 1) KAIST Multispectral Pedestrian Detection Benchmark

국내 유일한 공개 Dataset이다. 컬러 이미지와 열 영상 두 가지 방법으로 촬영되었으며, HD급 화질로 촬영된 9만 5000여 장의 보행자 Dataset을 가지고 있다.

자율주행을 위한 Dataset 중 국내의 것은 KAIST의 데이터가 유일하다. 국가마다 교통신호체계, 교통통제설비가 다르기 때문에 해외의 Dataset으로 학습을 시킬 경우 국내 교통체계를 정확히 인식하기 어렵다. 즉, 현재 우리나라는 국내 교통체계를 반영한 국내 자율주행을 위한 데이터가 현저히 부족한 상태인 것이다.

또한, 세계 최초로 경찰 및 교통안전요원에 대한 Dataset 구축을 시도한 Dataset이기도 하다.

#### 2) KITTI Dataset

독일에서 수집된 Dataset으로, 한국과 다른 환경에서 데이터 축적이 진행되어 독일 지역에만 유효한 정보들만이 담기게 되고, 그 결과 일반성을 보장하지 못하게 된다. 또한 학습에 사용된 데이터 이외의 다른 데이터에 대해서는 잘 동작하지 못하는 과적합 문제도 발생할 가능성이 있다. 예를 들자면, KITTI Dataset을 이용해 자율주행자동차의 인지 모델 학습을 진행할 경우 독일의 교통환경에만 과적합되고, 국가마다 다른 표지판이나 버스의 외관 디자인 등에 대한 정보는 가지고 있지 않으므로 국내 표지판과 버스를 인지하지 못할 수도 있다.

또한 KITTI의 경우 매우 맑은 날씨에서만 데이터가 수집되었는데, 이는 사계 변화가 뚜렷하고 날씨가 급격하게 바뀌는 국내 사정에는 적합하지 못한 Dataset이다. 또한 비와 눈 같은 악천후 상황이 포함되지 않았다는 점은 딥러닝 기반의 객체 인지 모델의 강건성에 한계를 유발하게 된다.

#### 3) BDD100K

가상 주행 장면의 Dataset으로 새롭고 다양한 대규모의 정보를 저장하고 있다. 보다 현실적인 주행 시나리오를 다루며 다양한 환경 영역에서 관심 있는 범주의 위치 배열과 모습 변화를 더 많이 포착한다.

BDD100K는 주간, 야간 및 다양한 날씨 역시 포함하지만, 미국에서 수집하였기 때문에 이를 기반으로 한 모델이 국내의 도로 환경에도 동등한 성능을 낼 수 있을지 보장하지 못한다.

#### 4) Waymo Open Dataset

25개 도시에서 약 1000만 마일에 달하는 자율주행 테스트를 하며 다양한 도로와 주행 환경에서 다양한 센서 데이터로 수집한 Dataset으로, 자율주행뿐만 아니라 머신러닝을 연구하는데 중요한 자료로 사용된다. 이 Open Dataset은 약 1TB의 크기를 갖는 대용량 데이터이기도 하다.

처음 시범을 보인 것은 2019년 8월으로, 고해상도의 센서 데이터와 1950개의 세그먼트에 대한 라벨이 있는 perception 데이터셋으로 시작하였다. 이 데이터셋은 지속적으로 업데이트되어 마지막으로 2020년 3월에 업데이트되었다. 그리고 2021년 3월에는 총 103,354개의 세그먼트에 대해 물체의 궤적과 이에 해당하는 3D 지도를 포함하는 motion 데이터셋이 추가되었다.

1000개의 주행 세그먼트 데이터가 포함되어 있는데 각 세그먼트는 센서당 10Hz(20만 프레임)으로 수집된 20초 동안의 주행 데이터로 구성되어 있다. 이러한 연속 주행 영상은 다른 도로에서 사용자의 행동을 추적하고 예측하는 모델을 개발하는데 활용할 수 있다. 또한 각 세그먼트에는 5개의 고해상도 웨이모 라이다(LiDAR)와 5개의 전면 카메라 센서 데이터가 제공된다.

Dataset에는 지역과 날씨, 시간 등이 다른 다양한 주행환경에서 수집된 자료들이 들어있는데, 테스트 주행 구간은 피닉스, 커클랜드, 마운틴 뷰, 캘리포니아와 샌프란시스코 등의 도심과 교외 구간이다. 또한 밤, 낮, 새벽, 황혼, 태양 빛이 강할 때와 비 내리는 날씨 등 시간과 날씨에 따른 다양한 주행 환경 데이터가 제공된다.

아울러 자동차, 보행자, 자전거, 표지 4가지로 구분해 라벨을 표기한 이미지와 라이다 프레임이 포함되어 있다. 전체 라벨은 120만 개의 2D 라벨과 약 1200만 개의 3D 라벨을 포함하고 있다. 센서 데이터는 미드 레인지 라이다 1개, 단거리 라이다 4개, 전면과 측면에 장착된 카메라 5대, 동기화된 라이다와 카메라 데이터, 카메라 투영에 대한 라이다, 센서 교정 및 차량 자세 자료로 구성되어 있다.

실제 차량으로 실제 도로를 주행하면서 이러한 데이터를 수집하고 Dataset을 만드는 작업에는 매우 많은 시간과 비용이 드는데 비해, 실제 차량과 라이다 센서를 가지고 주행하지 않아도 알고리즘을 검증할 수 있는 데이터를 제공한다는 장점을 가지고 있는 Dataset이다.

그러지만 사용자가 실제로 차량을 운행해서 얻는 데이터와 Dataset이 제공하는 데이터가 서로 같지 않을 가능성 역시 존재한다.

## 2. 자율주행 인지와 관련된 Open Source

### 1) 아펙스.OS(Apex.OS)

오픈소스 로봇 운영체제인 ROS에서 착안해 고안된 오픈소스 운영체제다. 더 정확히 말하자면 ROS2에 원류를 두고 있으며 ROS2와 API 차원에서 호환될 수 있다.

자율주행자동차, 자율주행로봇, 드론 등 높은 수준의 안전성을 요구하는 시스템의 개발에 적용할 수 있으며, 각종 임베디드 시스템을 이용해 확보한 센서 데이터를 실시간으로 처리할 수 있다는 특징을 가지고 있다. 또한 개발자들에게 필요한 '풀 스택(full stack)'을 제공하는 대신 자신의 필요에 따라 구현할 수 있도록 기본 인프라 환경만을 제공한다.

requirements.txt에서는 전체 코드 수행의 구현 환경을 정의하고 있다.

```
# The order of packages is significant, because pip processes them in the order of appearance.
# Changing the order has an impact on the overall integration process, which may cause wedges
in the gate later.

pbr>=2.0.0 # Apache 라이선스 2.0 적용
anyjson>=0.3.3 # 자유 소프트웨어 저작권

six>=1.9.0 # six = Python 2, 3의 compatibility 라이브러리 # MIT
eventlet!=0.18.3,>=0.18.2 # MIT
iso8601>=0.1.11 # MIT
netaddr!=0.7.16,>=0.7.13 # BSD
oslo.concurrency>=3.8.0 # Apache-2.0
oslo.utils>=3.20.0 # Apache-2.0
PyYAML>=3.10.0 # MIT
jsonschema>=2.0.0,<3.0.0,!2.5.0 # MIT
```

또한 test\_requirements.txt에서는 아펙스 테스트 수행을 위해 필요한 조건들을 나열하고 있다.

```
# The order of packages is significant, because pip processes them in the order of appearance.
# Changing the order has an impact on the overall integration process, which may cause wedges
in the gate later.

hacking<0.11,>=0.10.2 # Apache-2.0

coverage>=4.0 # Apache-2.0
fixtures>=3.0.0 # Apache-2.0/BSD
python-subunit>=0.0.18 # Apache-2.0/BSD
sphinx>=1.5.1 # BSD
oslosphinx>=4.7.0 # Apache-2.0
testrepository>=0.0.18 # Apache-2.0/BSD
testtools>=1.4.0 # MIT
mock>=2.0 # BSD
mox>=0.5.3 # Apache-2.0
```

## 2) Waymo

Waymo의 코드가 공유되어 있는 GitHub에서 코드에 대한 설명을 찾으면 다음과 같이 나온다.

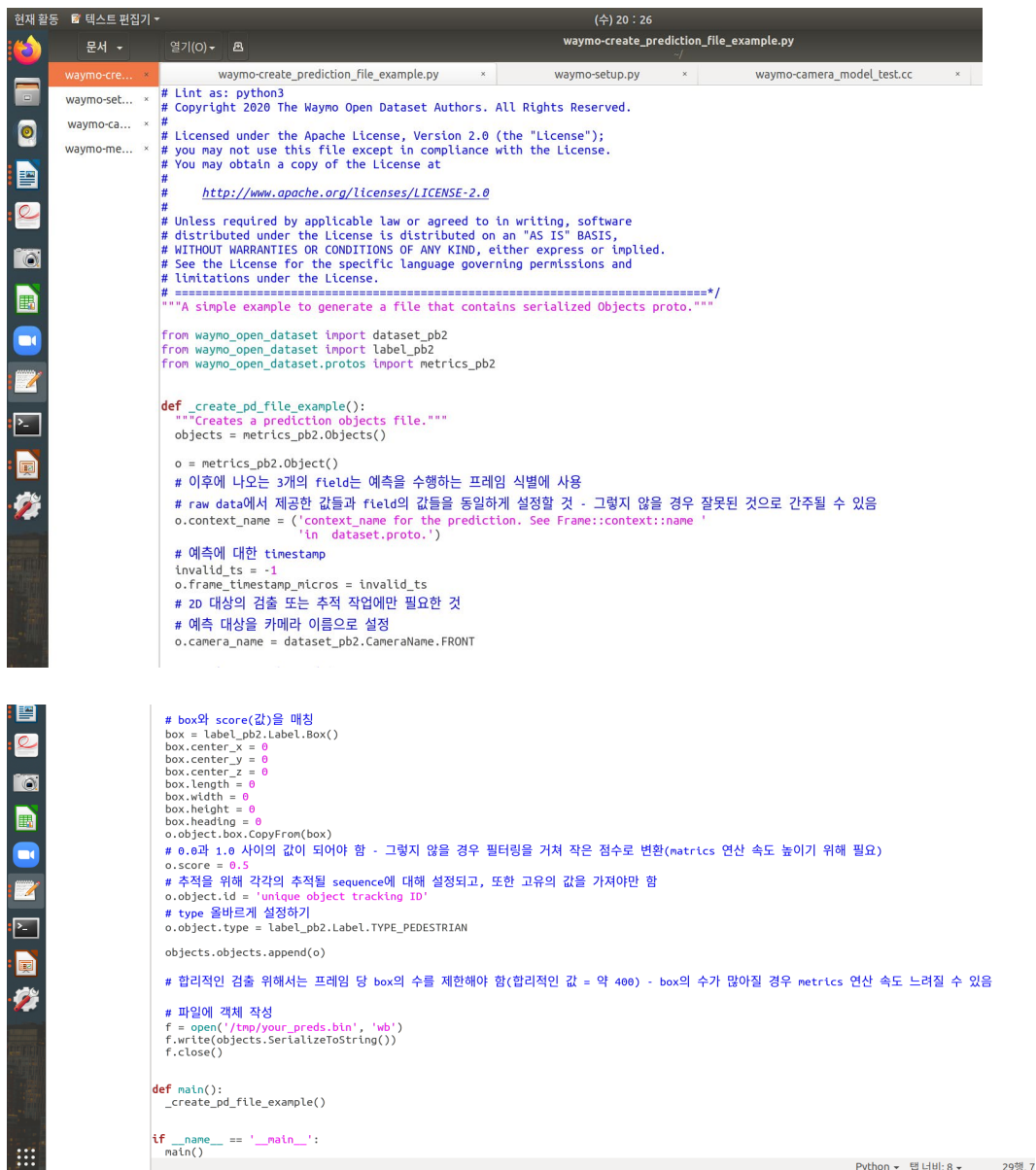
```
Run inference and dump the predictions in protos/metrics.proto:
Objects format. Example code can be found in metrics/tools/create_submission.cc:
example_code_to_create_a_prediction_file.
There is also a python version in metrics/tools/create_prediction_file_example.py.
```

자율주행자동차는 GPS(위성위치확인시스템)을 통해 현재 위치와 목적지를 끊임없이 모니터링하는 것은 기본이고, 여기에 다양한 센서가 더해져 도로의 다양한 정보를 확보한다. 여기에서 도로의 정보란 주변의 차량이나 사물, 사람에 대한 정보를 가리킨다. 이렇게 다양하게 수집된 데이터를 자율주행자동차가 종합하여 해석하고, 그 결과를 토대로 방향을 돌리거나 속도를 조절하고, 정지하는 등 운전을 수행하게 된다.

이러한 기술을 기반으로 구글은 2012년 3월 28일 시각장애인을 무인자동차에 태우고 시험 주행에 성공하였다. 여전히 법률 상으로는 온전히 자동차가 100% 운전을 수행하는 자동차는 허용하지 않고 있다. 운전석에 운전이 가능한 사람이 반드시 착석해 혹시 모를 사고에 대비해야 하며, 위험 상황에 대처하는 방법 등에 대한 교육을 이수하고 특별 면허를 취득해야 자율주행자동차를 운전할 수 있는 제한도 존재한다. 그렇지만 이러한 제한은 자율주행 기술이 점차 발달함에 따라 조금씩 사라질 것으로 예상된다.

### 3.2의 정리한 코드 중 하나 실행

- Waymo의 코드 실행waymo-create\_prediction\_file\_example.py 중



```
(수) 20 : 26
waymo-create_prediction_file_example.py

# Lint as: python3
# Copyright 2020 The Waymo Open Dataset Authors. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# =====*/
"""A simple example to generate a file that contains serialized Objects proto."""

from waymo_open_dataset import dataset_pb2
from waymo_open_dataset import label_pb2
from waymo_open_dataset.protos import metrics_pb2

def _create_pd_file_example():
    """Creates a prediction objects file."""
    objects = metrics_pb2.Objects()

    o = metrics_pb2.Object()
    # 이후에 나오는 3개의 field는 예측을 수행하는 프레임 식별에 사용
    # raw data에서 제공한 값들과 field의 값들을 동일하게 설정할 것 - 그렇지 않을 경우 잘못된 것으로 간주될 수 있음
    o.context_name = ('context_name for the prediction. See Frame::context::name '
                     'in dataset.proto.')

    # 예측에 대한 timestamp
    invalid_ts = -1
    o.frame_timestamp_micros = invalid_ts
    # 2D 대상의 검출 또는 추적 작업에만 필요한 것
    # 예측 대상을 카메라 이름으로 설정
    o.camera_name = dataset_pb2.CameraName.FRONT

    # box와 score(값)을 매칭
    box = label_pb2.Label.Box()
    box.center_x = 0
    box.center_y = 0
    box.center_z = 0
    box.length = 0
    box.width = 0
    box.height = 0
    box.heading = 0
    o.object.box.CopyFrom(box)
    # 0.0과 1.0 사이의 값이 되어야 함 - 그렇지 않을 경우 필터링을 거쳐 작은 점수로 변환(metrics 연산 속도 높이기 위해 필요)
    o.score = 0.5
    # 추적을 위해 각각의 추적될 sequence에 대해 설정되고, 또한 고유한 값을 가져야만 함
    o.object.id = 'unique object tracking ID'
    # type 올바르게 설정하기
    o.object.type = label_pb2.Label.TYPE_PEDESTRIAN
    objects.objects.append(o)

    # 합리적인 검출 위해서는 프레임 당 box의 수를 제한해야 함(합리적인 값 = 약 400) - box의 수가 많아질 경우 metrics 연산 속도 느려질 수 있음

    # 파일에 객체 작성
    f = open('tmp/your_preds.bin', 'wb')
    f.write(objects.SerializeToString())
    f.close()

def main():
    _create_pd_file_example()

if __name__ == '__main__':
    main()
```

## waymo-setup.py 중

```
현재 활동 텍스트 편집기 (수) 20 : 26
waymo-setup.py
waymo-cre... x waymo-create_prediction_file_example.py x waymo-setup.py x waymo-camera
waymo-set... x # Copyright 2019 The Waymo Open Dataset Authors. All Rights Reserved.
waymo-ca... x # Licensed under the Apache License, Version 2.0 (the "License");
waymo-me... x # you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the license is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# =====
#Waymo 상에서의 기본적인 설정
"""Setup script for pip package."""
#필요한 것들 import 해오기
from setuptools import find_packages
from setuptools import setup
from setuptools.command.install import install
from setuptools.dist import Distribution

__version__ = '1.3.0' #버전 설정
REQUIRED_PACKAGES = [
    'tensorflow-gpu== ' + '.'.join('TF_VERSION'.split('-')),
]
project_name = 'waymo-open-dataset-tf-TF_VERSION'

class BinaryDistribution(Distribution):
    """This class is needed in order to create OS specific wheels."""

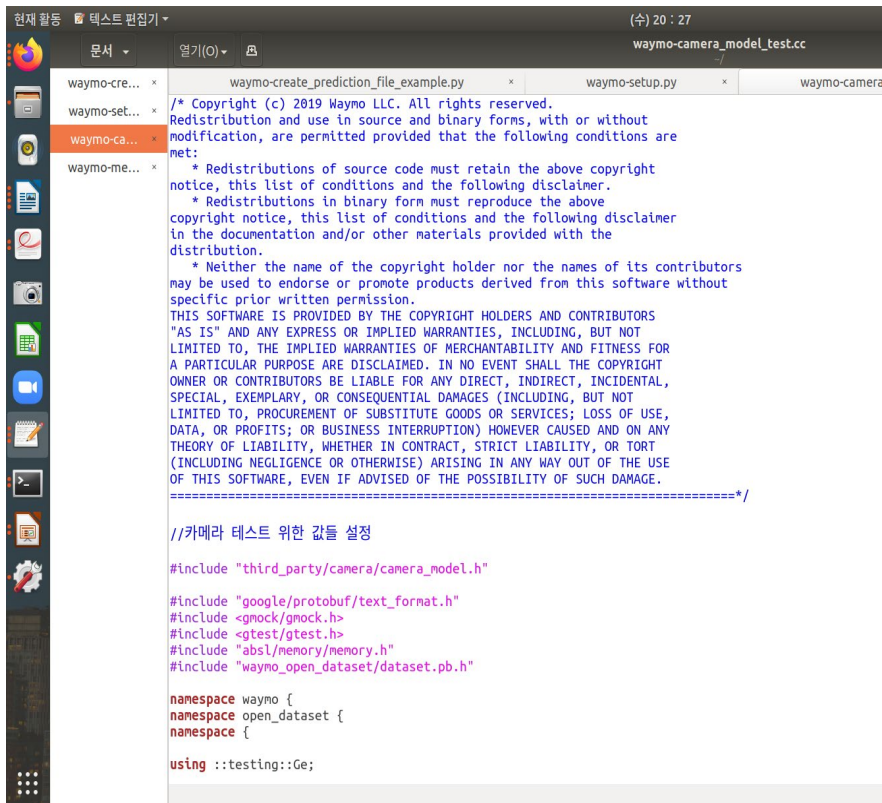
    def has_ext_modules(self):
        return True
```

```
현재 활동 텍스트 편집기 (수) 20 : 27
waymo-setup.py
waymo-cre... x waymo-create_prediction_file_example.py x waymo-setup.py x
waymo-set... x class InstallCommand(install):
waymo-ca... x """Override install command.
waymo-me... x
Following:
https://github.com/bigartm/bigartm/issues/840.
"""

def finalize_options(self):
    install.finalize_options(self)
    if self.distribution.has_ext_modules():
        self.install_lib = self.install_platlib
```

```
setup(
    name=project_name,
    version=__version__,
    description='Waymo Open Dataset libraries.',
    author='Waymo Open Dataset Authors',
    author_email='waymo-open-dataset@google.com',
    url='https://waymo.com/open',
    packages=find_packages(include=['waymo_open_dataset*'], exclude=[]),
    include_package_data=True,
    python_requires='>=3',
    install_requires=REQUIRED_PACKAGES,
    zip_safe=False,
    cmdclass={
        'install': InstallCommand,
    },
    distclass=BinaryDistribution,
    # PyPI package information.
    classifiers=[
        'Development Status :: 5 - Production/Stable',
        'Intended Audience :: Developers',
        'Intended Audience :: Education',
        'Intended Audience :: Science/Research',
        'License :: OSI Approved :: Apache Software License',
        'Programming Language :: Python :: 3.6',
        'Programming Language :: Python :: 3.7',
        'Programming Language :: Python :: 3.8',
        'Topic :: Scientific/Engineering',
        'Topic :: Scientific/Engineering :: Artificial Intelligence',
        'Topic :: Software Development',
        'Topic :: Software Development :: Libraries',
        'Topic :: Software Development :: Libraries :: Python Modules',
    ],
    license='Apache 2.0',
    keywords='autonomous driving dataset machine learning',
)
```

## waymo-camera\_model\_test.cc 중



```
/* Copyright (c) 2019 Waymo LLC. All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of the copyright holder nor the names of its contributors
may be used to endorse or promote products derived from this software without
specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
=====*/

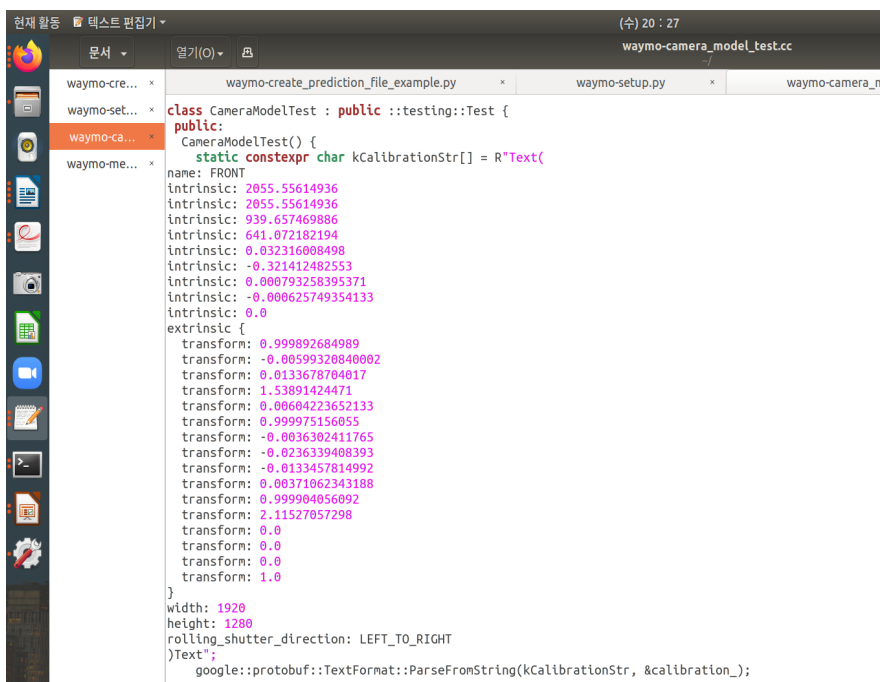
//카메라 테스트 위한 값들 설정

#include "third_party/camera/camera_model.h"

#include "google/protobuf/text_format.h"
#include <gmock/gmock.h>
#include <gtest/gtest.h>
#include "absl/memory/memory.h"
#include "waymo_open_dataset/dataset.pb.h"

namespace waymo {
namespace open_dataset {
namespace {

using ::testing::Ge;
```



```
class CameraModelTest : public ::testing::Test {
public:
  CameraModelTest() {
    static constexpr char kCalibrationStr[] = R"Text(
name: FRONT
intrinsic: 2055.55614936
intrinsic: 2055.55614936
intrinsic: 939.657469886
intrinsic: 641.072182194
intrinsic: 0.032316008498
intrinsic: -0.321412482553
intrinsic: 0.000793258395371
intrinsic: -0.000625749354133
intrinsic: 0.0
extrinsic {
  transform: 0.999892684989
  transform: -0.00599320840002
  transform: 0.0133678704017
  transform: 1.53891424471
  transform: 0.00604223652133
  transform: 0.999975156055
  transform: -0.0036302411765
  transform: -0.0236339408393
  transform: -0.0133457814992
  transform: 0.00371062343188
  transform: 0.999904056092
  transform: 2.11527057298
  transform: 0.0
  transform: 0.0
  transform: 1.0
}
width: 1920
height: 1280
rolling_shutter_direction: LEFT_TO_RIGHT
)Text";
  google::protobuf::TextFormat::ParseFromString(kCalibrationStr, &calibration_);
}
```

```
원재 활동 텍스트 편집기 (수) 20 : 27 waymo-camera_model_test.cc

waymo-cre... x waymo-create_prediction_file_example.py x waymo-setup.py x waymo-camera_r
waymo-set... x
waymo-ca... x static constexpr char kCameraImageStr[] = R"Text(
waymo-me... x name: FRONT
image: "dummy"
pose {
  transform: -0.913574384152
  transform: -0.406212769482
  transform: -0.0193141875914
  transform: -4069.03497872
  transform: 0.406637479491
  transform: -0.913082565675
  transform: -0.0304333457449
  transform: 11526.3118079
  transform: -0.00527303457417
  transform: -0.0356569976572
  transform: 0.999350175676
  transform: 86.504
  transform: 0.0
  transform: 0.0
  transform: 0.0
  transform: 1.0
}
velocity {
  v_x: -3.3991382122
  v_y: 1.50920391083
  v_z: -0.0169006548822
  w_x: 0.00158374733292
  w_y: 0.00212493073195
  w_z: -0.0270753838122
}
pose_timestamp: 1553640277.26
shutter: 0.000424383993959
camera_trigger_time: 1553640277.23
camera_readout_done_time: 1553640277.28
)Text";
google::protobuf::TextFormat::ParseFromString(kCameraImageStr, &camera_image_);
}

protected:
  CameraCalibration calibration_;
  CameraImage camera_image_;
};
```

```
원재 활동 텍스트 편집기 (수) 20 : 27 waymo-camera_model_

waymo-cre... x waymo-create_prediction_file_example.py x waymo-setup.py x
waymo-set... x TEST_F(CameraModelTest, RollingShutter) {
waymo-ca... x CameraModel camera_model(calibration_);
waymo-me... x camera_model.PrepareProjection(camera_image_);

double x, y, z;
camera_model.ImageToWorld(100, 1000, 20, &x, &y, &z);

double u_d, v_d;
EXPECT_TRUE(camera_model.WorldToImage(x, y, z, /*check_image_bounds=*/true,
&u_d, &v_d));

EXPECT_NEAR(u_d, 100, 0.1);
EXPECT_NEAR(v_d, 1000, 0.1);
EXPECT_NEAR(x, -4091.88016, 0.1);
EXPECT_NEAR(y, 11527.42299, 0.1);
EXPECT_NEAR(z, 84.46667, 0.1);
}

TEST_F(CameraModelTest, GlobalShutter) {
  calibration_.set_rolling_shutter_direction(CameraCalibration::GLOBAL_SHUTTER);
  camera_model(calibration_);
  camera_model.PrepareProjection(camera_image_);

  double x, y, z;
  camera_model.ImageToWorld(100, 1000, 20, &x, &y, &z);
```

```
waymo-ca... x double u_d, v_d;
waymo-me... x EXPECT_TRUE(camera_model.WorldToImage(x, y, z, /*check_image_bounds=*/true,
&u_d, &v_d));

EXPECT_NEAR(u_d, 100, 0.1);
EXPECT_NEAR(v_d, 1000, 0.1);
EXPECT_NEAR(x, -4091.97180, 0.1);
EXPECT_NEAR(y, 11527.48092, 0.1);
EXPECT_NEAR(z, 84.46586, 0.1);
}

TEST_F(CameraModelTest, SubPixelChangeInPrinciplePointChangesPoseTimeOffset) {
  int center_x = static_cast<int>(calibration_.intrinsic(2));
  int center_y = static_cast<int>(calibration_.intrinsic(3));
  CameraCalibration calibration_a = calibration_;
  calibration_a.set_intrinsic(2, center_x);
  calibration_a.set_intrinsic(3, center_y);
  CameraCalibration calibration_b = calibration_;

  // principle point 이동
  const double sub_pixel = 0.1;
  calibration_a.set_intrinsic(2, center_x + sub_pixel);
  calibration_a.set_intrinsic(3, center_y + sub_pixel);
  CameraModel camera_a(calibration_a);
  CameraModel camera_b(calibration_b);

  camera_a.PrepareProjection(camera_image_);
  camera_b.PrepareProjection(camera_image_);

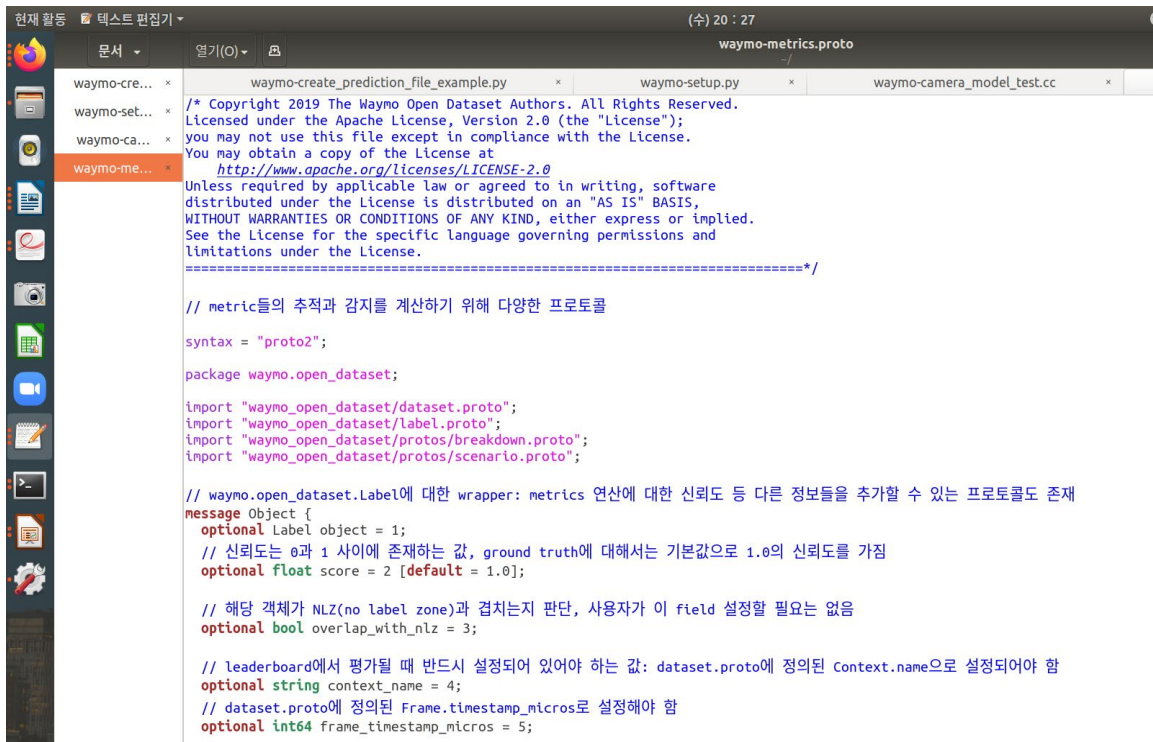
  const double readout_time = camera_image_.camera_readout_done_time() -
    camera_image_.camera_trigger_time() -
    camera_image_.shutter();

  const double min_seconds_per_col =
    readout_time / std::max(calibration_.width(), calibration_.height());
  EXPECT_THAT(std::abs(camera_a.t_pose_offset() - camera_b.t_pose_offset()),
    Ge(sub_pixel * min_seconds_per_col));
}

} // namespace
} // namespace open_dataset
} // namespace waymo
```



## waymo-metrics.proto 중



```
/* Copyright 2019 The Waymo Open Dataset Authors. All Rights Reserved.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
=====*/

// metric들의 추적과 감지를 계산하기 위해 다양한 프로토콜

syntax = "proto2";

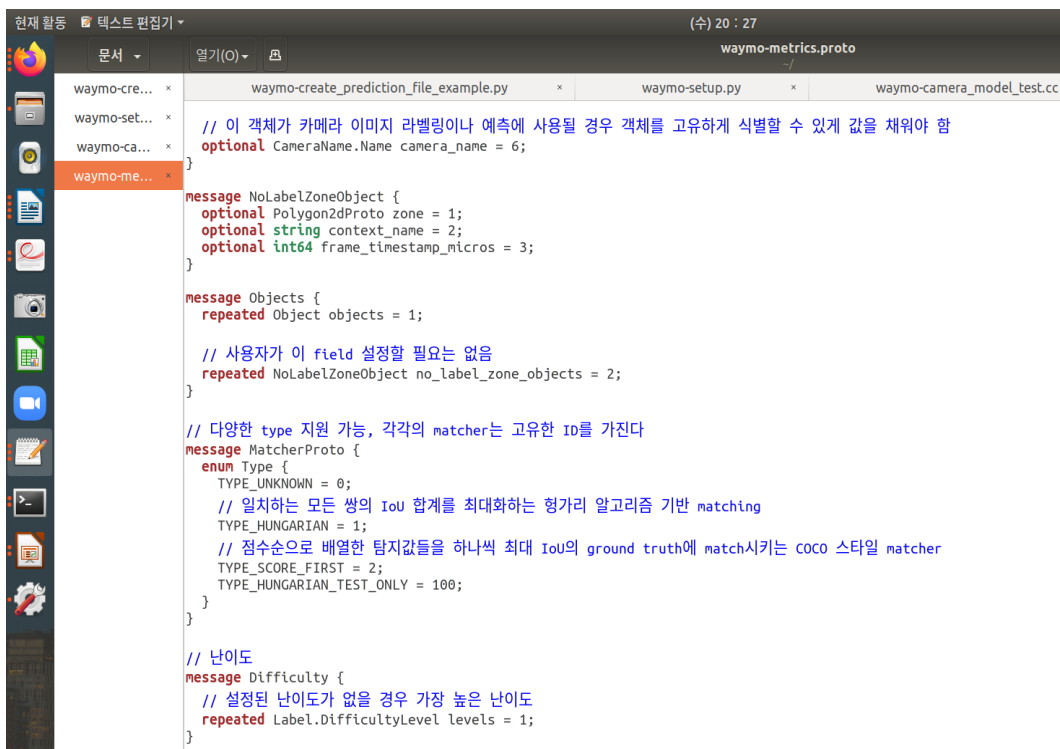
package waymo.open_dataset;

import "waymo_open_dataset/dataset.proto";
import "waymo_open_dataset/Label.proto";
import "waymo_open_dataset/protos/breakdown.proto";
import "waymo_open_dataset/protos/scenario.proto";

// waymo.open_dataset.Label에 대한 wrapper: metrics 연산에 대한 신뢰도 등 다른 정보들을 추가할 수 있는 프로토콜도 존재
message Object {
  optional Label object = 1;
  // 신뢰도는 0과 1 사이에 존재하는 값, ground truth에 대해서는 기본적으로 1.0의 신뢰도를 가짐
  optional float score = 2 [default = 1.0];

  // 해당 객체가 NLZ(no label zone)과 겹치는지 판단, 사용자가 이 field 설정할 필요는 없음
  optional bool overlap_with_nlz = 3;

  // leaderboard에서 평가될 때 반드시 설정되어 있어야 하는 값: dataset.proto에 정의된 Context.name으로 설정되어야 함
  optional string context_name = 4;
  // dataset.proto에 정의된 Frame.timestamp_micros로 설정해야 함
  optional int64 frame_timestamp_micros = 5;
```



```
// 이 객체가 카메라 이미지 라벨링이나 예측에 사용될 경우 객체를 고유하게 식별할 수 있게 값을 채워야 함
optional CameraName.Name camera_name = 6;
}

message NoLabelZoneObject {
  optional Polygon2dProto zone = 1;
  optional string context_name = 2;
  optional int64 frame_timestamp_micros = 3;
}

message Objects {
  repeated Object objects = 1;

  // 사용자가 이 field 설정할 필요는 없음
  repeated NoLabelZoneObject no_label_zone_objects = 2;
}

// 다양한 type 지원 가능, 각각의 matcher는 고유한 ID를 가진다
message MatcherProto {
  enum Type {
    TYPE_UNKNOWN = 0;
    // 일치하는 모든 쌍의 IoU 합계를 최대화하는 헝가리 알고리즘 기반 matching
    TYPE_HUNGARIAN = 1;
    // 점수순으로 배열한 탐지값들을 하나씩 최대 IoU의 ground truth에 match시키는 COCO 스타일 matcher
    TYPE_SCORE_FIRST = 2;
    TYPE_HUNGARIAN_TEST_ONLY = 100;
  }
}

// 난이도
message Difficulty {
  // 설정된 난이도가 없을 경우 가장 높은 난이도
  repeated Label.DifficultyLevel levels = 1;
}
```

```
현재 활동 텍스트 편집기 (수) 20 : 28 waymo-metrics.proto

waymo-cre... x waymo-create_prediction_file_example.py x waymo-setup.py x waymo-camera_model_test.cc x waymo-...
waymo-set... x
waymo-ca... x
waymo-me... x

// 탐지/추적 metrics를 계산하는 configuration
message Config {
  repeated float score_cutoffs = 1;
  // score_cutoffs 설정되지 않은 경우 예측값의 분포를 기반으로 cutoff, num_descore_cutoffs 설정
  // NOTE: 이 field는 더 이상 사용되지 않음, 위의 score_cutoffs를 수동으로 설정
  // TODO: clean this up.
  optional int32 num_desired_score_cutoffs = 2;
  // 사용자는 id만 지정하면 됨, 다른 정보는 지정할 수 없음
  repeated Breakdown.GeneratorId breakdown_generator_ids = 3;
  // breakdown_generator_ids와 같은 크기를 가짐, 각 입력은 각각에 해당하는 breakdown generator에 대해 고려되어야 하는 난이도를 나타냄
  repeated Difficulty difficulties = 4;
  optional MatcherProto.Type matcher_type = 5;
  // label 유형별로 인덱싱, label 종류 + 1 = 크기, 값은 0.0과 1.0 사이에 존재해야 함
  repeated float iou_thresholds = 6;
  optional Label.Box.Type box_type = 7;

  // 평균 정밀도 계산 위해 P/R 곡선 샘플링할 때 delta 호출
  optional float desired_recall_delta = 8 [default = 0.05];

  ////////////////////////////////////////////////////////////////////////////////////
  // 사용자는 다른 정보는 수정할 수 없음
  ////////////////////////////////////////////////////////////////////////////////////
  // 다른 정보가 수정되었을 경우 이 값 아래의 모든 정밀도는 0으로 간주
  optional float min_precision = 9 [default = 0.0];
  // 최소의 정확도 설정 - 이보다 낮을 경우 모두 거짓으로 간주
  optional float min_heading_accuracy = 10 [default = -1.0];
}
```

```
현재 활동 텍스트 편집기 (수) 20 : 28 waymo-metrics.proto

waymo-cre... x waymo-create_prediction_file_example.py x waymo-setup.py x waymo-camera_model_test.cc x waymo-...
waymo-set... x
waymo-ca... x
waymo-me... x

message TrackingMeasurement {
  // 거짓 음성의 개수
  optional int32 num_misses = 1;
  // 거짓 양성의 개수
  optional int32 num_fps = 2;
  // mismatch의 개수
  optional int32 num_mismatches = 3;
  // match된 모든 객체들에 대한 matching cost의 합계
  optional double matching_cost = 4;
  // match된 모든 객체들의 개수
  optional int32 num_matches = 5;
  // ground truth 객체들의 총 개수(즉, 라벨링된 객체들)
  // Total number of ground truth objects (i.e. labeled objects).
  optional int32 num_objects_gt = 6;

  // 이 measurement 계산 위해 사용된 score cutoff
  optional float score_cutoff = 7;
}
```

```
현재 활동 텍스트 편집기 (수) 20 : 28 waymo-metrics.proto

waymo-cre... x waymo-create_prediction_file_example.py x waymo-setup.py x waymo-camera_model_test.cc x waymo-...
waymo-set... x
waymo-ca... x
waymo-me... x

message TrackingMeasurements {
  repeated TrackingMeasurement measurements = 1;
  optional Breakdown breakdown = 2;
}

message TrackingMetrics {
  // 정확도를 추적하는 다중 객체(누락, 불일치 및 fp의 합계)
  optional float mota = 1;
  // 정밀도를 추적하는 다중 객체(matching_cost / num_matches)
  optional float motp = 2;
  // 누락 비율 (num_misses / num_objects_gt).
  optional float miss = 3;
  // 불일치 비율 (num_mismatches / num_objects_gt).
  optional float mismatch = 4;
  // 거짓 양성 비율 (num_fps / num_objects_gt).
  optional float fp = 5;

  optional float score_cutoff = 6;

  optional Breakdown breakdown = 7;
  // 원래 측정값
  optional TrackingMeasurements measurements = 8;
}
```

#### 4. 참고문헌

- 1) 이용구, ICT R&D 동향, 정보통신기획평가원 주간기술동향, pp. 36~42, 2020. 4. 15.
- 2) 윤승제 et al., 자율주행 차량의 학습 데이터 자동 생성 시스템 개발, 한국ITS학회논문지, vol. 19 no. 5, pp. 162~177, 2020.10.
- 3) 김달훈(2019.08.26), “연구형 자율주행 데이터 무료 제공” ... 웨이모, 개방형 데이터세트 공개, [https://www.ciokorea.com/news/129266\(2021.04.03\)](https://www.ciokorea.com/news/129266(2021.04.03)).
- 4) AIMMO(2020.07.27), 자율주행용 학습 데이터 가공, [https://medium.com/aimmosubscribe/%EC%9E%90%EC%9C%A8%EC%A3%BC%ED%96%89%EC%9A%A9-%ED%95%99%EC%8A%B5-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EA%B0%80%EA%B3%B5-e8dd01a2254b\(2021.04.03\)](https://medium.com/aimmosubscribe/%EC%9E%90%EC%9C%A8%EC%A3%BC%ED%96%89%EC%9A%A9-%ED%95%99%EC%8A%B5-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EA%B0%80%EA%B3%B5-e8dd01a2254b(2021.04.03)).
- 5) Redesign X(리디자인엑스)(2020.07.14), CVPR 2020에서 소개된 자율주행 오픈소스 데이터셋 TOP5, [https://rdx-live.tistory.com/90\(2021.04.03\)](https://rdx-live.tistory.com/90(2021.04.03)).
- 6) Ambika Choudhury(2020.01.07), Top 5 Autonomous Driving Dataset Open-Sourced At CVPR 2020, [https://analyticsindiamag.com/top-5-autonomous-driving-dataset-open-sourced-at-cvpr-2020/\(2021.04.03\)](https://analyticsindiamag.com/top-5-autonomous-driving-dataset-open-sourced-at-cvpr-2020/(2021.04.03)).
- 7) WAYMO, About, [https://waymo.com/open/about/\(2021.04.07\)](https://waymo.com/open/about/(2021.04.07)).
- 8) 로봇신문(2020.02.05), ‘아펙스.AI’, 자율주행차용 오픈소스 운영체제 발표, [http://www.irobotnews.com/news/articleView.html?idxno=19508\(2021.04.07\)](http://www.irobotnews.com/news/articleView.html?idxno=19508(2021.04.07)).
- 9) 장형준 et al., 가상 3D 라이다 기반 객체 분류 딥러닝 학습 데이터셋 구축 방법에 관한 연구, 한국자동차공학회논문집, vol. 28, no.6, pp.427~437.
- 10) waymo-research / waymo-open-dataset, [https://github.com/waymo-research/waymo-open-dataset/tree/master/waymo\\_open\\_dataset\(2021.04.07\)](https://github.com/waymo-research/waymo-open-dataset/tree/master/waymo_open_dataset(2021.04.07)).