

자율주행 차량의 학습 데이터 자동 생성 시스템 개발

Development of Autonomous Vehicle Learning Data Generation System

윤 승 제* · 정 지 원** · 홍 준** · 임 경 일*** · 김 재 환**** · 김 형 주*****

* 주저자 : 모라이 기업부설연구소 연구원

** 공저자 : 모라이 대표이사

*** 공저자 : 차세대융합기술연구원 경기도자율주행센터 선임연구원

**** 공저자 : 차세대융합기술연구원 경기도자율주행센터 센터장

***** 교신저자 : 차세대융합기술연구원 경기도자율주행센터 선임연구원

Seungje Yoon* · Jiwon Jung* · June Hong* · Kyungil Lim** · Jaehwan Kim** · Hyungjoo Kim**

* Mobility Research and Artificial Intelligence

** Advanced Institutes of Convergence Technology

† Corresponding author : Hyungjoo Kim, hyungjoo@snu.ac.kr

Vol.19 No.5(2020)

October, 2020

pp.162~177

pISSN 1738-0774

eISSN 2384-1729

<https://doi.org/10.12815/kits.2020.19.5.162>

Received 19 August 2020

Revised 23 September 2020

Accepted 20 October 2020

© 2020. The Korea Institute of Intelligent Transport Systems. All rights reserved.

요 약

자율주행시스템에서 다양한 센서를 기반으로 한 외부환경 인지는 주행안전성과 직접적인 관계가 있다. 최근 머신러닝/심층 신경망 기술의 발전으로 심층 신경망 기반의 인지 모델이 사용됨에 따라, 인지 알고리즘의 올바른 학습과 이를 위한 양질의 학습데이터가 필수적으로 요구된다. 그러나 자율주행에 발생할 수 있는 모든 상황을 데이터를 수집하는 것은 현실적인 어려움이 많다. 해외와 국내의 교통 환경의 차이로 인지 모델의 성능이 저하되기도 하며, 센서가 정상동작을 못하는 악천우에 대한 데이터는 수집이 어려우며 질적인 부분을 보장하지 못한다. 때문에, 실제 도로가 아닌 시뮬레이터 내 가상 도로 환경을 구축하여 합성 데이터를 수집하는 접근법이 필요하다. 본 논문에서는 국내 실정에 맞게 국내 도로 상황을 모사한 시뮬레이터 환경 안에 날씨와 조도, 차량의 종류와 대수, 센서의 위치를 다양화하여 학습데이터를 수집하였고, 보다 더 좋은 성능을 위해 적대적 생성 모델을 활용하여 이미지의 도메인을 보다 실사에 가깝게 바꾸고 다양화 하였다. 그리고 위 데이터로 학습한 인지 모델을 실제 도로 환경에서 수집한 시험 데이터에 성능 평가를 진행하여, 실제 환경 데이터만으로 학습한 모델과 비슷한 성능을 내는 것을 보였다.

핵심어 : 자율주행 시뮬레이션, 실도로 데이터셋, 가상환경 데이터셋, 인공지능 학습데이터

ABSTRACT

The perception of traffic environment based on various sensors in autonomous driving system has a direct relationship with driving safety. Recently, as the perception model based on deep neural network is used due to the development of machine learning/in-depth neural network technology, a the perception model training and high quality of a training dataset are required. However, there are several realistic difficulties to collect data on all situations that may occur in self-driving. The performance of the perception model may be deteriorated due to the difference between the overseas and domestic traffic environments, and data on bad weather where the sensors can not operate normally can not guarantee the qualitative part. Therefore, it is necessary to build a virtual road environment in the simulator rather than the actual road to collect the training data. In this paper, a training dataset collection process is suggested by diversifying the weather, illumination, sensor

position, type and counts of vehicles in the simulator environment that simulates the domestic road situation according to the domestic situation. In order to achieve better performance, the authors changed the domain of image to be closer to due diligence and diversified. And the performance evaluation was conducted on the test data collected in the actual road environment, and the performance was similar to that of the model learned only by the actual environmental data.

Key words : Autonomous simulation, Real road dataset, Virtual environment dataset, AI learning data

I. 서 론

자율주행시스템(Autonomous Driving System)은 인지·판단·제어 등이 포함되는 일련의 자율주행 과정에서 외부환경을 인지하는 것이 주행안전성과 직접적인 관계가 있다, 이를 위해서 LiDAR, Radar, Camera 등의 다양한 센서를 차량에 추가로 장착하여 주행안전성을 최대화 시킨다. 이렇듯 외부환경을 인지하기 위해서 다양한 종류의 센서가 사용되고, 검출 영역과 목적에 맞춰 활용되는 센서의 개수와 위치가 다양하게 정의되어 질 수 있다. 최근 머신러닝 기술의 발전으로 인지율(Recognition rate)이 크게 향상되었지만 이를 위해 반드시 선행돼야 하는 부분은 인지 알고리즘의 ‘학습(Learning)’ 과정이며, 이를 위해서는 ‘학습데이터(Learning Dataset)’가 필수적으로 요구된다. 인지 성능의 향상이란 머신러닝 또는 딥러닝(Deep learning) 알고리즘의 적용만을 통해서 이뤄낼 수 없으며, 양질의 데이터와 이를 기반으로 한 학습, 그리고 테스트와 피드백의 종합적인 과정을 반복함으로써 얻을 수 있다.

최근 자율주행 주행환경이 실제 환경의 도시환경으로 이동함으로써 안전한 자율주행을 위해서는 더 빠르고 정확한 외부환경의 인지 성능이 요구된다. 이를 위해서 실제 환경에서의 동적 및 정적 객체에 대한 양질의 학습데이터를 얻는 과정이 선행되어야 한다. 하지만 매 순간 변화하는 실제 환경에서의 학습데이터 생성은 시간과 비용이 많이 들고 불확실성을 포함할 수 있다는 단점이 존재한다. 인지 성능의 향상이란 머신러닝 또는 딥러닝 알고리즘의 적용만을 통해서 이뤄낼 수 없으며, 양질의 데이터와 이를 기반으로 한 학습 그리고 테스트와 피드백의 종합적인 과정을 반복함으로써 얻을 수 있다. 모든 과정이 유기적인 관계로 결속되기 때문에 한 가지 과정이라도 어긋난다면 긍정적인 결과를 예측하기 어렵다. 데이터의 경우 머신러닝 또는 딥러닝에 있어서 시작이면서 끝이라 할 수 있을 정도로 중요한 부분을 차지한다고 할 수 있다.

그리고 많은 데이터가 존재하더라도, 인지 학습을 위한 데이터는 무엇보다 다양한 환경정보를 많이 담아야 인지 모델의 일반화 성능을 향상 시킬 수 있다. 특정 지역에서 얻은 정보는 그 지역 고유의 특징, 규정 등에 한정 되어 있다고 해도 무방하다. KITTI Dataset의 경우, 독일에서 수집되었으며, 한국과 다른 환경에서 데이터 축적이 진행됐기 때문에 데이터에는 고스란히 독일 지역에만 유효한 정보들만이 담기게 된다 (Gaiger et al., 2013). 이는 데이터셋이 일반성을 보장하지 못한다 뜻이며, 대부분의 머신러닝을 기반으로 한 모델들이 학습에 이용된 데이터외의 데이터에는 잘 동작하지 못하는 과적합 문제에 빠지게 된다. KITTI Dataset을 통해 인지 모델 학습을 진행하면 독일의 교통환경에 과적합되어 국가마다 다른 표지판, 버스의 외관 디자인 등에 대한 일반적인 정보가 없으므로, 국내 표지판과 버스를 인지하지 못하는 한계를 가지게 된다. 또한, KITTI의 경우 매우 맑은 날씨에만 데이터가 수집되어 사계변화가 뚜렷하고 날씨가 급격하게 바뀌는 국내 사정에 적합하지 못하는 단점을 가진다. 이렇듯 국내 자율주행 산업의 발전과 상용화를 위해서는 국내 환경에 적합한 한국형 Dataset 구축이 필요한 이유이다.

또한 데이터가 인지 모델 학습에 사용이 되려면 정답 데이터인 라벨도 필수적으로 확보가 되어야 한다.

그러나 적게는 5000장, 많게는 10만장이 넘는 데이터에 라벨을 수작업으로 하는데 많은 시간과 비용이 들어간다. 그리고 라벨링도 인적 요인으로 인한 잘못 지정될 수 있어서 학습 결과와 모델 성능의 신뢰성을 떨어뜨린다. 때문에, 단 시간에 대량의 데이터 라벨링을 정확하게 실행하는 것이 요구된다.

특히 시뮬레이션 플랫폼에서는 국내 도로환경(실제 국내에 존재하는 도로, 국내 표지판, 구급차 등 특수차량) 및 악천우 등 제약없이 구현이 가능하며, 대량의 데이터셋 구축이 가능한 장점을 가진다. 이에 본 연구에서는 실제 환경에 준하는 환경 구축, 나아가 센서 모델링 기술을 통해 구축된 가상환경에서의 데이터셋을 생성하는 시스템을 제시하고자 한다. 실제 환경에서의 안전한 자율주행을 이끄는 인지 성능 향상을 위해서 실제와 준하는 가상의 주행환경 생성 및 센서 모델링 기술의 적용으로 양질의 학습데이터 생성을 목표로 한다.

네이버랩스 같은 경우는 virtual KITTI 라는 가상의 데이터셋을 생성하여 발표하였다(Gaidon et al., 2016). 날씨와 카메라 구도, 시간에 대한 조도 변화를 게임 엔진을 통하여 생성하고 다양화하였다. 그러나 실사성이 떨어지는 단점이 있었으며, 실사성을 높이기 위해서 게임 엔지니어들의 수작업과 그에 따른 인건비가 든다. 그에 반해, 엔비디아와 토론토 대학의 연구는 포토 리얼리즘에 치중한 이미지 수천 장보다는 다양한 도메인에서 생성한 수십만의 이미지가 더 좋은 일반화 성능을 낸다는 연구 결과를 내놓은 적이 있다(Tremblay et al., 2018). 이는 포토 리얼리즘을 높이기 위해 들어가는 시간과 비용을 더 많은 수량과 가짓수로 보완하는 것이 가능하다는 점을 제시한 것이다. 허나, 일반화를 위해 얼마나 많은 데이터를 넣어야하는지 사전에 판단하기가 힘들며, 증강된 데이터 양만큼 학습 시간이 늘어난다는 단점도 존재하여, 적정 수준 이상의 학습 자원을 요구한다.

가장 최근에 접근하는 방식은 적대적 생성 네트워크(Generative Adversarial Networks)를 사용하여 (Goodfellow et al., 2014), 실사에 가까운 이미지를 생성하고, 이를 학습에 응용하는 것이다. Ian Goodfellow가 GAN을 발표한 이후 수많은 GAN에 대한 연구들이 대거 등장 했으며, PIX2PIX와 CycleGAN 같은 이미지 생성 모델들이 등장하였다(Isola et al., 2017; Zhu et al., 2017). 자율주행용 학습 데이터 생성에 응용한 케이스 중 가장 최근에 나온 것이 웨이모에서 발표한 surfel GAN이며, 조명 정보를 포함한 표면 요소를 기반으로 적대적 생성학습을 통해 조명 같은 포면 요소 기준으로 실사성을 개선하여 객체 인식 성능을 높였다고, CVPR2020에 발표하였다(Yang et al., 2020).

본 논문은 다음과 같이 구성되어 있다. 1장에서는 연구배경 및 목적, 선행연구와의 차이점을 설명한다. 2장에서는 자율주행 차량의 학습데이터 자동생성 시스템인 MORAI SIM에 대해서 설명하고, 3장에서는 방법론 적용 및 분석 결과를 도출한다. 마지막으로 4장에서는 본 연구 결과를 요약하고 시사점을 제시한다.

II. 학습데이터 자동 생성 시스템 개발

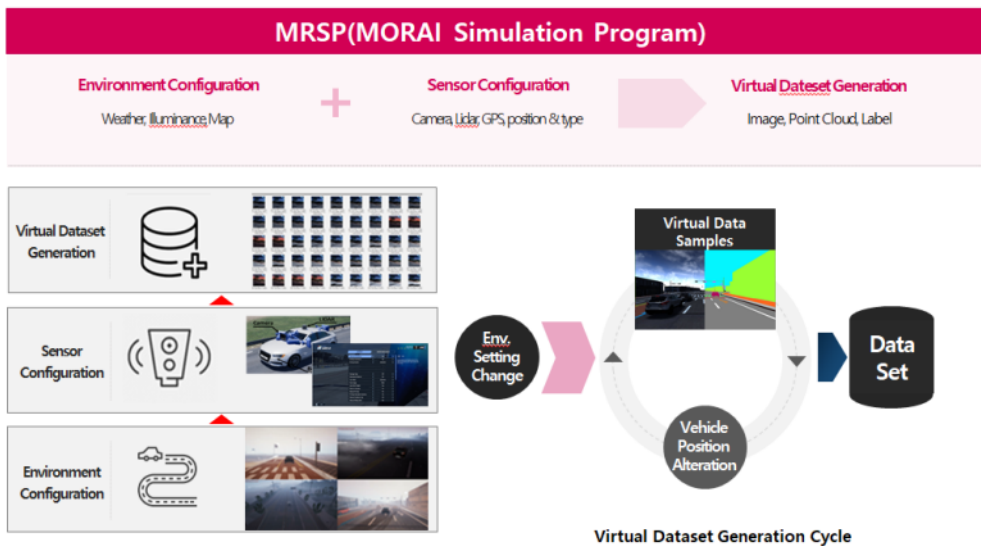
1. 학습데이터 자동 생성 시스템 개요

본 연구의 가상 주행 환경 상에서 자동으로 데이터셋을 생성하기 위한 프로그램인 MORAI SIM의 구성도는 <Fig. 1>과 같다. 우선 초기에 환경과 센서에 대한 설정을 입력한다. 환경의 가상 환경의 날씨와 조도, 광원의 방향, 차량의 갯수, 차가 수집을 하면서 이동할 위치 등을 선택할 수 있다. 이는 실제 데이터셋 취득 시 맑은 날씨에 주로 수집하여 다양한 날씨에 대한 데이터가 없다는 점을 보완하기 위한 것이며, 자율주행 연구를 하는데 다양한 경우의 수를 제공해주는 이점이 있다.

환경 설정 외에도 센서에 대한 설정도 가능하다. 자율주행 차량마다 설정되는 센서의 종류와 개수, 위치

등이 다르기 때문에 실제 구현하고자 하는 자율주행 차량의 센서 설정에 주변 환경이 어떻게 측정될 수 있는지를 미리 확인할 수도 있어, 적합한 센서의 위치들을 찾아 구축하는데 도움을 준다.

센서 설정과 환경 설정 등이 다 끝나면, 가상 환경 상에서 차량을 매 스텝 진행시켜가면서 센서 데이터를 얻기 시작한다. 이때 센서 데이터는 카메라 이미지와 라이다로부터 나오는 점군 데이터와 GPS센서로의 측위 데이터 등이 있으나, 이것만으로는 데이터로써 의미가 없다. 자율주행용 데이터셋이 의미가 있으려면 정답 데이터인 라벨이 있어야하는데, 가상의 주행환경은 실제와 다르게 이런 정답데이터도 자동으로 생성시켜주는 장점이 있어 실제 데이터셋의 정답데이터를 만드는데 필요한 인력과 시간을 줄여주는 장점을 가지고 있다. 이렇게 데이터를 순차적으로 축적시켜서 대량의 데이터셋을 단시간에 만들어 내는 것이 자동생성 프로그램의 역할이다.



<Fig. 1> MORAI SIM dataset automatic generation system

2. MORAI SIM 구성 및 기능

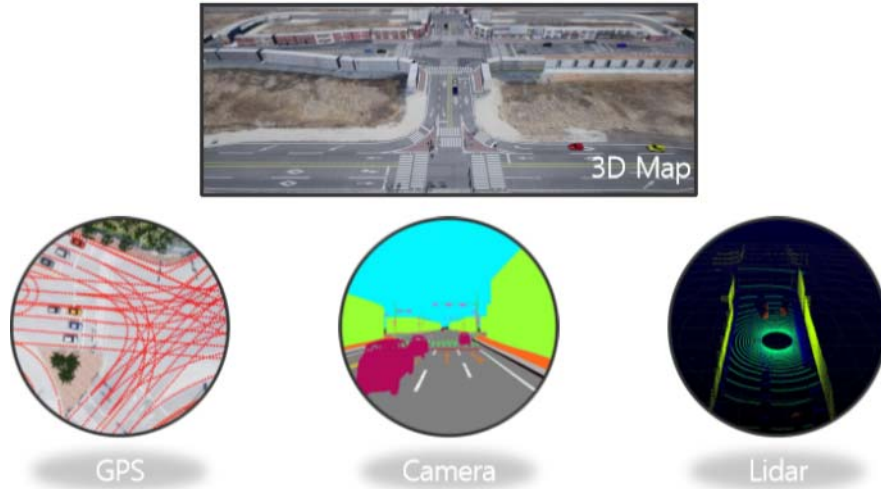
1) 가상의 주행환경 물리 엔진 및 3D Map 구축

가상의 학습 데이터를 생성하는 MORAI SIM을 만드는데 가장 먼저 해야 하는 것은 가상 환경을 만들 개발 엔진 선정이었다. 본 논문에서 선정한 개발 엔진은 Unity다. Unity는 UnrealEngine과 함께 게임 개발에 쓰이는 개발 엔진이며, 객체의 색상과 질감 표현 뿐만 아니라, 중력과 충돌, 투과 같은 물리 조건 등도 구현이 가능하다. 이 때문에 자율주행 연구에도 Unreal Engine과 Unity가 사용되고 있다. Unreal Engine을 쓴 대표적인 자율주행용 시뮬레이터는 CARLA(Dosovitskiy et al., 2017)와 Airsim 이다. 특히 CARLA는 강화학습 용도로 쓰이고 있어 자율주행 챌린지 대회를 열었고, 그 결과가 논문으로 나온바 있다(Chen et al., 2019). Unity의 경우에는 LGSVL simulator가 대표적이며, 자율주행용 데이터셋 구축용도로는 네이버랩스의 virtual kitti가 대표적이다(Gaidon et al., 2016).

본 논문에서 선정한 개발 엔진은 Unity인데, 실사성은 Unreal Engine 못지 않으며, ML-agent 등으로 강화학습용 환경으로 개발이 가능했고 클라우드 상으로 시뮬레이터를 올려서 클라우드 연산으로 서버를 연결할 수

있는 곳 어디서나 접속해서 사용할 수 있도록 하는 기능이 추후에 추가 가능했기 때문이다.

게임엔진으로 Unity를 선택한 후, 실제 국내 도로 환경을 <Fig. 2>처럼 3D Map으로 옮겨서 구현하였다. 앞서 언급한대로 공개되어 있는 자율주행 데이터셋은 대부분 독일이나 미국같은 해외 기준으로 되어 있어, 국내에만 존재하는 표지판과 틀게이트 같은 기타 교통 인프라에 대한 정보가 포함되지 않기 때문에 학습 후 국내 환경에 적용하기에 검증이 되지 않는다. 이를 해결하기 위해 국내에만 존재하는 교통 인프라를 가상환경 상에서 추가하였다. 이때 가상환경에서 생성되는 데이터셋은 국내 표지판과 특수 차량 등에 대한 인식을 향상에 기여할 것이다(Tremblay et al., 2018).



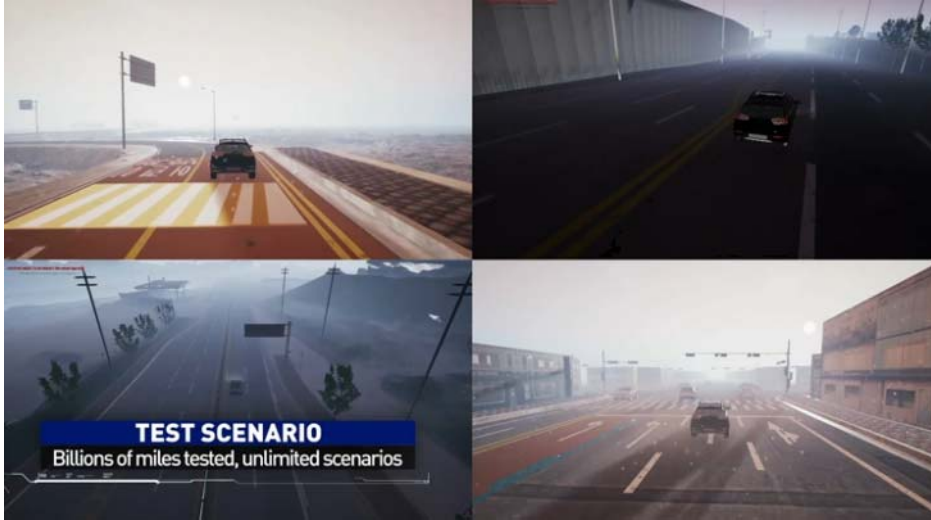
<Fig. 2> Sensor information available from MORAI SIM

2) 가상의 주행환경 내 센서 데이터 정의

MORAI SIM은 가상의 주행 환경을 생성하고 센서의 정답 데이터를 실시간으로 생성해 센서 데이터를 가공하는 작업 없이 무한대에 가까운 데이터를 생성 할 수 있으며, <Fig. 2>에서는 시뮬레이터에 적용된 3D 환경과 센서데이터의 예시가 나타나 있다. 가상의 주행환경에 존재하는 건물과 도로, 신호등, 차량 등의 객체들을 제공하는 것이 가능하며, 실제 국내에서 존재하는 건물 양식들과 표지판, 노면 표시 등으로 구축되어 있다. 그리고 가상 환경 내 GPS도 실제 들어오는 GPS와 동일한 프로토콜로도 제공이 되며, Lidar의 경우도 실제 상용화 되어 있는 센서와 동일한 프로토콜과 채널방식으로 구현되어 있다.

3) 다양한 주행환경 조건 생성

MORAI SIM에서는 <Fig. 3>처럼 일반적인 주행 상황뿐만 아니라 빛 조건, 그림자, 날씨, 노면 등을 다양하게 변경하면서 데이터셋을 구축 할 수 있다. 유명한 오픈 데이터셋인 KITTI의 경우는 맑은 날씨에만 수집되어 있다는 점 때문에 비와 눈 같은 악천후 상황이 포함되지 않아 딥러닝 기반의 객체 인지 모델의 강건성에 한계를 유발한다. cityscapes의 경우에도 독일 내 여러 도시에 수집하였으나 이 또한 악천후를 포함하진 않았다. BDD100K는 주,야간 및 다양한 날씨를 포함하지만, 미국에서 수집되었기 때문에 이를 기반으로 한 모델이 국내의 도로 환경에도 동등한 성능을 낼 수 있을지 보장하지 못한다. 이를 보완하기 위해 MORAI SIM은 가상으로나마 악천후를 재현해서 객체 인지 모델 학습을 위한 데이터셋의 정보들을 보강하고자 한다.



<Fig. 3> Creation of various driving environments of MORAI SIM

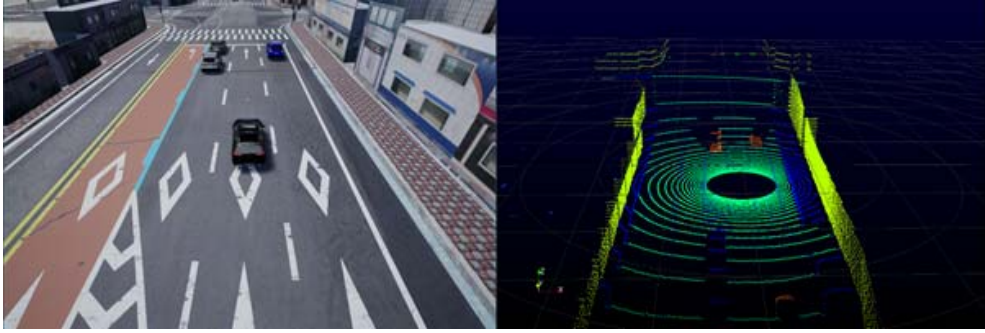
4) 센서의 정답 데이터

자율주행 용 데이터셋 수집 및 구축에 가장 어려운 점 중 하나는 바로 정답데이터를 사람이 직접 정하는 라벨링 작업이다. 실제 라벨링 작업은 사람의 손으로 하는 게 일반적이다. 당연히 장시간이 작업이 요구되며, 이는 대량 데이터 구축을 어렵게 한다. 또한, 정답 데이터를 만들어야하는 사람의 인건비도 고려해야한다. 그리고 장기간 작업 시의 사람의 실수와 숙련도 미달로 인해 라벨링이 잘못 되어 데이터셋의 품질을 떨어뜨리게 된다. 이는 데이터셋을 사용하여 만든 모델의 신뢰도 또한 하락시킨다. 때문에 자동으로 대량으로 데이터를 제공함과 동시에, 실수를 유발하는 인적 요인을 배제시킬 수 있는 가상 환경 내 데이터 수집이 장점을 가지고 있다.

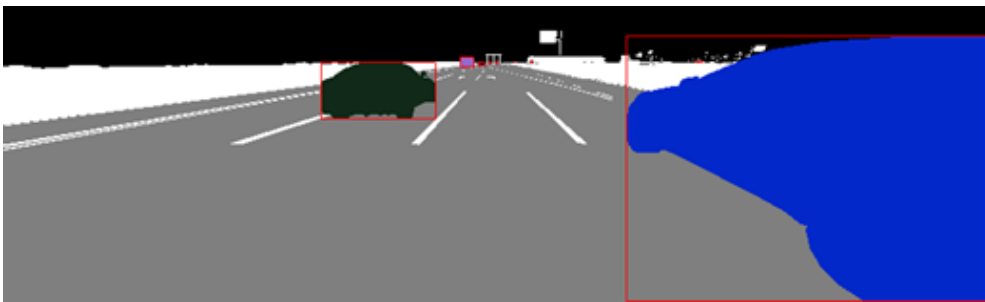
사용자가 획득할 수 있는 정답데이터의 예시는 <Fig. 4, 5>와 같은 점군 데이터와 이미지이고 센서의 Raw 데이터와 학습에 사용될 수 있도록 가공된 정답 데이터로 구성한다. 사용자가 센서 데이터를 별도의 가공 없이 바로 학습에 사용가능한 데이터를 제공 받을 수 있도록 구현하였다.

<Fig. 4>는 각 점군 데이터가 차량, 도로, 건물 등 어떤 클래스에 속해 있는지를 나타내는데, 이는 수집 과정에 동시에 나오는 정답데이터이다. 점군 데이터는 이미지보다도 정답데이터를 매기기 어렵기 때문에, 가상 주행 환경으로 인한 자동 생성이 큰 이점이 된다.

<Fig. 5>는 가상 주행 환경을 구축하는데 쓰인 엔진을 통해서, 각 객체의 클래스 정답을 Fig. 왼쪽처럼 칼라맵 형태로 출력하는 게 가능하다. 이를 통해서 사람이 개입하지 않고, 자동으로 칼라맵의 픽셀에 맞는 bounding box로 변환할 수 있으며, 그 bounding box의 좌상단, 우하단 좌표를 KITTI 데이터셋의 양식과 같이 텍스트 파일형태로 변환이 가능하다. 결국 사람이 bounding box를 이미지 한 장 마다 직접 작성하는 과정을 생략시킬 수 있는 장점이 있다.



<Fig. 4> Point cloud data correct answer data



<Fig. 5> Image correct answer data

Ⅲ. 방법론 적용 및 분석

1. 가상환경 데이터 수집

1) 가상 주행환경 데이터 수집

가상의 주행환경에서 얻은 데이터들은 각자 다른 조건에 수집이 되었다. 변경했던 조건들은 아래와 같다. 이미지 해상도는 총 4가지(800x240, 640x480, 1280x720, 1920x1080) 로 구성이 되어 있다. 그리고 데이터를 수집할 때마다 카메라의 각도를 변경시켜가면서 수집을 하였고, 날씨도 실제 데이터에 포함되어 있지 않았던 눈, 비, 안개를 포함한 악천후와 주간 및 야간 상황을 재현하였으며, 차량도 기존 승용차 외에도 국내에서만 운행되는 버스 및 특수 차량 모델까지 구현을 하였다. 이와 같이 총 2만여 장의 데이터를 취득하였다. 그리고 III-2-3)에서처럼 각 프레임에 대응되는 정답 데이터까지 반드시 포함시켜야 한다.



<Fig. 6> Virtual driving environment image data obtained under various conditions

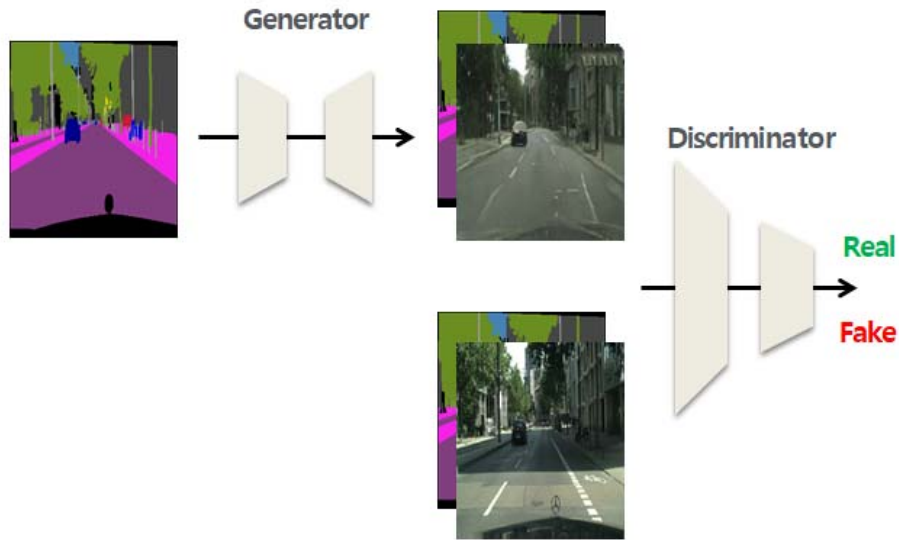
2) 실제 이미지로의 Neural Style Transfer

가상의 주행환경에서 얻은 데이터 외에도 더 다양한 도메인의 성능을 올리기 위해, style transfer를 시도했다. style transfer, image-to-image translation 등으로 불리는 이 문제는 한 이미지를 다른 이미지의 스타일을 가지는 새로운 이미지로 생성하는 것을 말한다. 그리고 지난 5년간 딥뉴럴넷과 Generative Adversarial Network(GAN)등의 발달로(Goodfellow et al., 2014), 딥뉴럴넷을 사용한 여러 neural style transfer 알고리즘들이 등장하였다(Isola et al., 2017; Zhu et al., 2017). segmentation 라벨을 실제 사진으로 바꿔준다던가, 흑백사진을 컬러로, 주간 사진을 야간으로 바꾸는 등의 문제들이 style transfer 문제들이다.

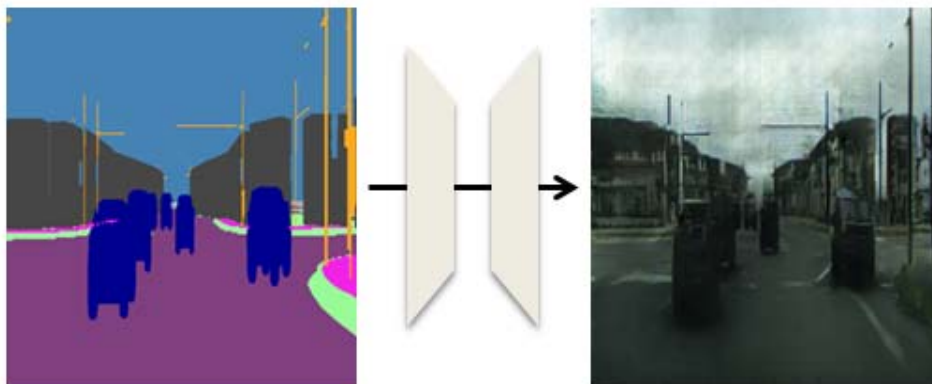
그러나 이 neural style transfer를 실제로 구현하는데 몇 가지 기술적 및 이론적인 어려움이 따르는데, segmentation 라벨을 얻어서 실제 사진처럼 얻으려면 사람이 직접 라벨을 지정해야하는 등의 수작업이 필요하며, 대량으로 얻는데 많은 시간이 걸린다(Zhu et al., 2017). 반면에 가상 주행 환경의 경우, 데이터 수집 시 카메라 센서로 얻은 이미지와 정답 데이터를 같은 프레임 상에서 사람의 수고를 들이지 않고 단기간에 대량으로 얻어내는 것이 가능하기 때문에, style transfer를 시도하기가 유리하다.

가상 환경의 데이터를 가지고 Image-to-image translation 를 하는데 Pix2Pix 모델을 사용하였다(Isola et al., 2017). Conditional GAN의 한 종류인 Pix2Pix는 이미지 생성을 위한 어떤 조건 이미지를 입력으로 주면, 그 조건을 기반으로 모델에 학습된 스타일대로 조건 이미지를 스타일 변환시켜준다. 이 때문에 무작위로 이미지가 생성되지 않고, 가상환경에서와 같은 조건 대로 생성하는게 가능해진다. 여기서 조건은 가상 환경에서 얻은 정답데이터를 쓰고, 이를 기반으로 Pix2Pix를 적용하는 전체 과정은 <Fig. 8>과 같다. U-net 구조를 띄는 생성자는 정답데이터를 받아서 임의의 이미지를 생성해내고, 판별자는 실제 이미지와 생성자가 만든 이미지를 가지고 진짜 이미지를 판별한다. 판별자는 생성자의 가상 이미지를 구분해내려는 방향으로 학습하고, 생성자는 판별자를 속이는 방향으로 학습을 한다. 이 두 모델이 적대적으로 학습을 하면, 생성자는 실제 이미지와 비슷한 가상 이미지를 만들어 낼 수 있다는 뜻이다. Pix2Pix를 학습시킬 때는 segmentation 학습이 많이 쓰이는 cityscapes 데이터셋을 가지고 시도하였다.

실제로 만들어낸 결과를 <Fig. 7>로 보면 매우 아쉬운 점을 볼 수 있다. 건물과 도로, 신호등과 전신주 등은 cityscapes의 도메인대로 생성된 걸 볼 수 있으나, 가장 중요한 차량의 세부적인 생성결과는 아쉬움이 남는다. 이는 모델 하나로 전방의 도로 상황을 전부 다 표현하는데 한계가 있기 때문이다. 전체 이미지에 큰 비중을 차지하는 하늘과 건물 도로는 표현이 가능하지만, 차량이 차지하는 픽셀의 비중이 매우 적어 생성 결과가 떨어지더라도 학습과정에서 매겨지는 loss에 큰 영향을 미치지 않는다.



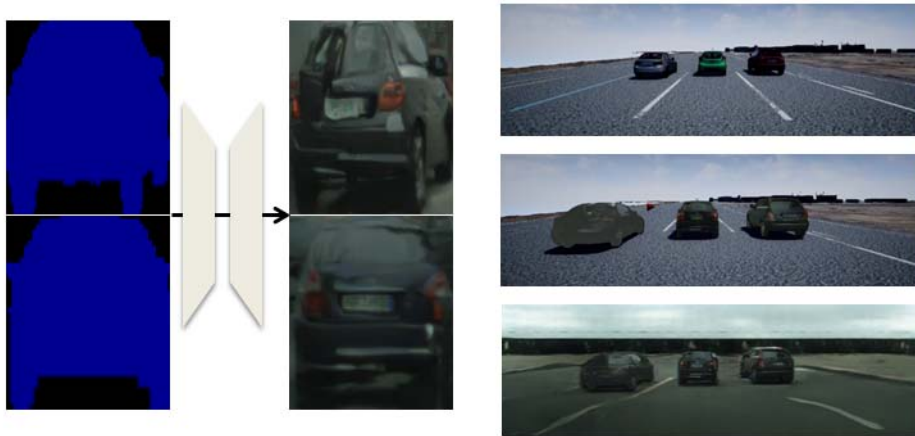
<Fig. 7> Pix2Pix based virtual-real driving image style transfer



<Fig. 8> Pix2Pix style transfer result

이를 보완하기 위해 차량만 스타일 변환하도록 하는 모델을 하나 더 만들어서 실험을 진행하였다. cityscapes에 차량만 마스킹하고 확대하여 새로운 이미지 데이터를 만들어 학습시킨 결과, 차량의 범퍼와 표지판 등 차량의 세세한 부분도 생성이 되는 것을 확인할 수 있다. 그리고 시뮬레이터 이미지 데이터는 그림

과 같이 가상 주행환경으로 얻은 데이터, 차량만 변환한 데이터, 전체 지형을 변환한 데이터, 총 3가지 그룹으로 구성하였다.



<Fig. 9> Improved vehicle generation results and training data composition

2. 학습데이터 생성 검증 결과

1) 객체 인지 모델 및 성능지표 선정

가상 환경으로 만들어진 데이터셋을 검증하는 방법으론 객체인식 모델을 가상 환경 이미지로 트레이닝 시킨 후, 실제 카메라 이미지로 트레이닝 시킨 것과 비교해보는 것으로 시도했다. 가상 환경의 이미지가 의미가 있다면, 객체 인식 모델의 학습 결과에 의미 있는 성능 변화를 줄 거라는 가정 하에 진행한 것이다.

선정된 객체인식 모델은 YOLO-v3를 사용하였다(Redmon et al., 2019). 이전 객체 인지 모델들에 비해 속도와 정확도 전부 적절한 결과를 내놓고 있다고 저자는 설명하고 있는데, back bone 모델로는 darknet53을 사용하였고, 학습을 하는데 overfitting을 방지하는 batch normalization과 data augmentation을 사용하여 regularization 기법들을 사용하였다. 특히 이전에 쓰이는 ResNet-101과 ResNet-152 등에 비해 FPS가 높으면서도, 정확도는 큰 차이가 없으며, floating point operation/s에서 가장 높은 성능을 달성하여 GPU를 효율적으로 사용한다는 점을 저자는 보여주고 있다.

그 밖에 성능을 평가하기 위한 성능 지표로는 객체 인지 문제에 가장 많이 사용하는 mean Average Precision(mAP)를 사용하였다. 보통 객체 분류로 많이 쓰이는 precision과 recall은 객체의 클래스를 분류하는 threshold 에 따라 그 값이 바뀌어 강건한 성능지표로 쓰이기 힘들다. 그리고 객체의 추정위치까지 평가해주는 않는다. 그래서 먼저 객체의 추정 위치 지표로는 예측된 bounding box와 라벨링된 bounding box간의 Intersection over union(IoU)를 계산해서 IoU가 특정 값 이상일 때 객체 성능을 평가한다. 본 논문에선 0.5로 정했다. 그리고 IoU가 0.5 이상일 때, threshold를 바꿔가면서 precision과 recall을 구하고 recall 별 최대 precision의 평균을 내는데, 이것을 Average Precision(AP)라 한다. mAP는 AP가 모든 클래스에 대해서 구하여 평균을 낸 값을 말한다. mAP가 높을수록 정확하고, 작을수록 부정확하다.

YOLO-v3의 COCO mAP-50의 결과를 보면 Single Shot Detector(SSD)(Liu et al., 2015), YOLO-v2(Redmon et al., 2016) 보단 정확하고, RetinaNet과 Faster R-CNN (Ren et al., 2016)등에는 못 미쳤지만, inference time 에서

는 RetinaNet과 Faster R-CNN보다 더 우수한 측면을 보였기 때문에 선정하였다. 자율주행을 위한 인지 모델의 경우는 정확도뿐만 아니라 inference time이 50ms 이내로는 나와야 하는 점을 고려하여 비교하였다.

2) 데이터셋 구성

가상 환경으로 만들어진 데이터셋과 비교하기 위한 실제 데이터셋 비교군으로는 KITTI와 Berkeley Deep Drive 100K(BDD100K)를 사용하였고(Yu et al., 2020), 다른 가상 데이터로는 Virtual KITTI를 사용하였다(Gaidon et al., 2016). 그리고 학습에는 참여시키지 않는 테스트용 데이터로는 KITTI의 validate 데이터와 KODAS를 사용하였다. BDD100K는 KITTI와 cityscapes에 비해 눈, 비, 흐림 등 다양한 날씨를 포함하고, 주간에만 수집된 두 데이터셋에 비해 야간과 터널 등 조도가 낮은 상황에서도 수집이 됐기 때문에 객체 인지 모델을 일반화가 가능하도록 학습시키는 것이 가능하다.

우선 평가해보고자 하는 요소는 두 가지로, 첫 번째는 논문에서 제시하는 가상환경의 데이터와 Virtual KITTI 둘 중 하나를 학습시키면, 두 YOLO-v3 모델 중 어디가 실제 KITTI에 더 잘 동작하는지의 여부다. 만일 논문에서 제시한 가상 주행 환경의 데이터가 KITTI에 잘 맞는 도메인이라면, Virtual KITTI보다 더 좋은 성능을 내놓을 것이라는 가정이다. 실제로 학습 데이터와 검증 테스트 용 데이터 간의 도메인 차이가 커지면 커질수록 학습된 모델은 테스트 데이터에 대해 인식 결과가 부정확해진다. 때문에 실제 KITTI 데이터에 두 가지 가상 환경 데이터를 동등한 비율로 조합하여 후보군을 9개 만들어서 성능을 비교하였다.

두 번째는 과연 KITTI와의 다른 데이터셋에서 인식 결과가 잘 나오는지의 여부이다. 두 번째 실험의 비교를 위한 데이터셋은 아래와 같이 3가지로 나누어서 구성하고 각각의 데이터셋을 YOLO-v3의 구조로 동등하게 학습시켰다. 학습은 셋 다 80epoch으로 실행하였다. 1과 2는 BDD100K와 논문에서 제시되는 가상환경 이미지가 YOLO-v3 성능에 어떻게 기여하는지 비교하는 것이며, 3번의 경우에는 각각의 데이터 셋이 어떤 상황에서 영향을 주는지를 보여주기 위함이다.

<Table 1> Dataset configuration for verification of training data generation

Experiment Number	Dataset Number	Dataset configuration
1	0	Only real KITTI
	1	Virtual KITTI all & KITTI 5000
	2	Virtual KITTI all & KITTI 2000
	3	Morai SIM Data all & KITTI 5000
	4	Morai SIM Data all & KITTI 2000
	5	Morai SIM Data 2000 & KITTI 3000
	6	Morai SIM Data 3000 & KITTI 2000
	7	Morai SIM Data 4000 & KITTI 1000
	8	Morai SIM Data & KITTI 1000
2	1	KODAS 125 + BDD100K 20000
	2	KODAS 125 + Morai SIM Data 20000
	3	KODAS 125 + BDD100K 20000 + Morai SIM Data 20000

3) 학습 결과

첫번째 실험의 테스트 결과로 KITTI validate 데이터에 대한 각 모델 inference 결과의 mAP와 recall을 산출하고, 이를 아래 표와 같이 나타내었다. mAP의 결과들을 보면 당연히 제일 많은 데이터가 들어간 3번의 경우가 제일 좋았으며, 동등한 비율로 실제 KITTI 데이터에 두 가상 데이터를 넣었을 때는 논문에서 제시하는 MORAI SIM의 데이터가 KITTI validate 데이터에 더 높은 mAP를 나타냄을 보였다. 이는 Virtual KITTI보다, MORAI SIM 데이터가 KITTI의 도메인에도 모델의 인지 기능을 유지할 수 있는 일반성을 높여준다는 것으로 해석할 수 있다.

그리고 실제 KITTI와 MORAI SIM의 데이터의 총합을 5000개로 동등하게 조정해서 비율만 달리 하여 비교한 5,6,7번의 경우를 보면 MORAI SIM의 비율이 점점 높아질수록 학습된 인지 모델의 성능이 떨어지는 것을 알 수 있는데, 이는 아무리 MORAI SIM 데이터가 Virtual KITTI보다 실제 KITTI와의 유사도가 높더라도 본래 원본만큼은 아니라는 점을 시사한다. 그리고 KITTI 데이터 3000개에 Morai 데이터 2만여 개를 추가한 9번의 경우가 실제 KITTI 5000개의 데이터로만 학습한 것과 근접하지만 성능이 조금 낮은 것은, KITTI 데이터 2000개의 차이를 메꾸기 위해 가상환경 데이터가 10배 이상 필요하다는 것을 의미한다. 물론 실제 데이터 2000개를 구하는 거보단, 그 10배 이상의 가상환경 데이터를 구하는 것이 작업 시간이나 기타 들어가는 기회 비용 측면에서 더 유리하다는 점을 고려해야한다.

<Table 2> Learning data generation verification result

Dataset Number	mAP (IOU>0.5)
0	0.761
1	0.670
2	0.606
3	0.764
4	0.701
5	0.718
6	0.590
7	0.584
8	0.584
9	0.743

두번째 실험의 테스트는 KODAS 데이터 셋에 간단하게 inference해보았으며, 그 결과 중 하나는 아래와 같다. 가상환경 이미지와 실제 이미지 둘 다 맑은 날씨 상에서 차량이 일렬로 배치된 상황이 포함되어 있으므로, 인지된 차량 대수도 차이가 없음을 확인할 수 있다. 이는 MORAI SIM의 데이터가 악천후가 아닌, 자율주행을 하는데 센서가 날씨의 악영향을 받지 않는 조건에선 자율주행용 인지 모델이 적절한 성능을 낼 수 있도록 기여할 수 있음을 보여준다.



<Fig. 10> Detection results of a KODAS example image with the yolo v3 models trained in 3 datasets on Experiment no.2

차에 대해서만 한정하여 mAP를 비교한 결과는 아래와 같으며, 비록 데이터 개수는 125개로 적고 학습데이터에서 나온 KODAS 데이터와 비슷한 도메인이지만, 가상으로 만든 이미지 데이터가 BDD100K보다 더 높은 걸 확인할 수 있었다.

<Table 3> Learning data generation verification result

mAP (IOU>0.5)	Car
1	0.863
2	0.909
3	0.866

4) Failure case

가상 환경으로 만들어진 데이터셋도 몇몇 케이스에서는 한계를 보였는데, 아래 그림처럼 악천후 상황에서 차량을 전혀 인지하지 못했다. 가상환경 데이터셋이 재현을 못했던 폭우로 인한 물 맺힘 현상 때문에, 아래 상황은 out of distribution 데이터라고 정의내릴 수 있다. 때문에 이런 out of distribution 혹은 novel observation 입력에 취약한 딥뉴럴넷 모델들의 고질적인 문제로 올바른 inference를 하지 못한다. 1번의 경우는 차가 아닌 표지판을 차로 오인지 했으나, BDD100K와 가상환경 데이터셋 둘 다 포함해서 학습한 3번 모델의 경우는 이런 오인지가 없는데, 이는 가상환경 데이터셋이 재현하지 못한 이미지엔 취약하더라도 data augmentation으로 할 수 있는 regularization과도 같은 역할을 해서 저 오인지를 방지해줬다고 볼 수 있다. 2번의 경우 가상환경 데이터셋이 앞유리에 물이 맺힌 상황을 재현 못했기 때문에 위와 같이 폭우 상황에서 수집한 이미지 내 차량을 인지 못했다. 대신 1번에서 발생하는 차량 오인지를 3번에서 없애주는 regularization 역할을 했다.



<Fig. 11> Object unknown case of model trained with virtual environment dataset

두 번째 상황의 경우는 2번이 작은 bounding box로 객체를 인지하지 못하는데, 이는 가상 환경 데이터의 최소 bounding box의 횡종 최소 크기가 25 픽셀이기 때문이다. 특히 가상환경 데이터는 resolution을 다양하게 쪼갬기 때문에, 자동 생성 하에 동일해서 써야할 최소 픽셀 크기를 그 이하로 줄이기 힘들었다. 이 점은 최소 15픽셀로 정의되어 있던 타 데이터에 비해 작은 물체 식별에 불리함을 낳았다. 그러나 BDD100K로 학습했던 모델인 1번의 경우, 국내 버스를 인지하지 못하는 것을 확인 할 수 있는데, 이는 BDD100K의 차량들이 미국 내에 운행 중인 차량에 한정되어 있어, 국내 버스 샘플은 BDD100K 기준으로 out of distribution이기 때문이다. 이 또한 위의 상황과 마찬가지로 novel observation에 취약한 모델의 한계를 보여주었으며, 이를 보완하기 위해 국내 버스를 포함한 데이터셋으로 학습시켜야 인지가 가능함을 3번으로 확인할 수 있다. 라벨링할 bounding box를 되도록 가능한 한 작게 해줘야 함을 보여줬으며, BDD100K 같이 미국 차량만 포함되어 있는 데이터가 국내의 도메인에 취약했음을 확인하였다.



<Fig. 12> Object unrecognized case due to the minimum pixel setting of the virtual environment dataset

IV. 결 론

본 연구에서는 자율주행이 필요한 객체 인지 모델을 학습하는데 국내 교통 환경을 잘 반영한 데이터의 필요성과 실제 데이터를 대량으로 수집하기 어려운 점을 설명하였고, 이를 간단한 절차로 대량 생성이 가능한 가상 환경 데이터셋으로 해결해보자 하는 방법을 제안하였다.

해외에서 공개된 자율주행용 데이터셋은 국내 도로 교통 환경 상에서만 존재하는 정보들을 포함하지 않기에 객체 인지 모델이 국내 환경에 대한 일반적인 성능을 보장 못하기에, 국내 도로 환경에 대한 정보가 들 어간 국내 데이터셋이 필요함을 설명하였다. 또한 이 데이터를 구축하는데 많은 시간과 비용이 소요되고, 센서가 동작하지 못하는 날씨 등의 외부요인으로 수집이 어려우며, 인적 요인으로 라벨링에 실수가 발생하는 등으로 대규모의 데이터를 단기간에 생성하는데 가상 환경 상에서 자율주행용 데이터를 생성시키는 프로그램이 필요함을 설명하면서, MORAI SIM을 설명하였다.

본 논문에선 MORAI SIM를 제작하는데 Unity를 사용하여 표지판과 기타 국내 도로 인프라를 렌더링하고 3D Map을 재현하여 국내 도로 환경 정보를 가상 데이터셋이 포함하도록 만들었다. 실제 데이터 수집 및 구축의 어려운 점인 Bounding Box 라벨링은 가상환경 상에서 객체별 Segmentation 라벨을 직접 받아 생성함으로 해결하였고, 가상 환경 상의 학습 데이터 수집은 실제 데이터 수집에 비해 다양한 Resolution과 카메라 각도, 날씨 조건을 제공하면서 데이터를 생성하였다. 그리고 실제와의 유사성을 개선해보기 위해 GAN의 한 종류인 Pix2Pix 로 가상환경의 Segmentation 이미지를 실제 이미지처럼 Style Transfer 하였다.

BDD100K로 학습한 모델과 가상 환경 이미지로 학습한 모델을 비교하여 차량 인지 성능에 비슷한 결과를 낼 수 있음을 확인하였고, 두 데이터를 혼합하여 학습하면 Data Augmentation로 인한 성능 향상과 국내 도로 환경에서만 볼 수 있는 이미지들에 대한 정확한 인지 결과를 낼 수 있는 것도 상대적으로 개선된 mAP 값들로 증명되었다.

아직 가상환경의 데이터만 가지고는 실제 이미지 데이터의 물체 인식을 수행하는 데에는 한계가 있으며, 실제 이미지 데이터를 1천개 정도 포함하더라도 가상 환경의 데이터 이미지가 배수로 필요하다는 결론도 나왔기에, 모델의 domain adaptation 성능을 개선해서 해결해보는 연구도 필요할 것으로 사료된다. 그밖에 향후 연구 사항으로 게임 엔진으로 구현된 가상환경 데이터의 실사성을 개선하는 것으로, 이미지 외에 Lidar 포인트 클라우드의 실제 잡음을 GAN을 통해서 재현시킴으로서 3D object detection 과 lidar SLAM 연구에도 기여하는 것도 가능할 것으로 판단된다.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1C1C1003296) and Advanced Institutes of Convergence Technology (No. AICT-2020-0005).

REFERENCES

- Chen D., Zhou B., Koltun V. and Krähenbühl P.(2019), “Learning by Cheating,” *Conference on Robot Learning(CoRL)*.
- Dosovitskiy A., Ros G., Codevilla F., Lopez A. and Koltun V.(2017), “CARLA: An Open Urban Driving Simulator,” *Conference on Robot Learning(CoRL)*.
- Gaidon A., Wang Q., Cabon Y. and Vig E.(2016), “Virtual Worlds as Proxy for Multi-Object Tracking Analysis,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.4340-4349.
- Geiger A., Lenz P., Stiller C. and Urtasun R.(2013), “Vision meets Robotics: The KITTI Dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp.1231-1237.
- Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. and Bengio Y.(2014), “Generative Adversarial Networks,” *NIPS*.
- Isola P., Zhu J. Y., Zhou T. and Efros A.(2017), “Image-to-Image Translation with Conditional Adversarial Networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.1125-1134.
- Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C. and Berg A.(2015), “SSD: Single Shot MultiBox Detector,” *European Conference on Computer Vision(ECCV)*, pp.21-37.
- Redmon J. and Farhadi A.(2016), “YOLO9000: Better, Faster, Stronger,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.7263-7271.
- Redmon J. and Farhadi A.(2019), *YOLO v3: An Incremental Improvement*, University of Washington.

- Redmon J., Divvala S., Grishick R. and Farhadi A.(2015), “You Only Look Once: Unified, Real-Time Object Detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.779-788.
- Ren S., He K., Girshick R. and Sun J.(2016), “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *In IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Rong G., Shin B. H., Tabatabaee H., Lu Q., Lemke S., Možeiko M., Boise E., Uhm G., Gerow M., Mehta S., Agafonov E., Kim T. H., Sterner E., Ushiroda K., Reyes M., Zelenkovsky D. and Kim S.(2020), “LGSVL Simulator: A High Fidelity Simulator for Autonomous Driving,” *ITSC*.
- Tremblay J., Prakash A., Acuna D., Brophy M., Jampani V., Anil C., To T., Cameracci E., Boochoon S. and Birchfield S.(2018), “Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization,” *In CVPR Workshop*.
- Yang Z., Chai Y., Anguelov D., Zhou Y., Sun P., Erhan D., Rafferty S. and Kretzschmar H.(2020), “SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.11118-11127.
- Yu F., Chen H., Wang X., Xian W., Chen Y., Liu F., Madhavan V. and Darrell T.(2020), “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.2636-2645.
- Zhu J. Y., Park T., Isola P. and Efros A.(2017), “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, pp.2223-2232.