

# 도심 환경에서의 자율주행을 위한 멀티 모달기반 딥러닝 모델 연구

심우일<sup>1,0</sup>, 박태화<sup>2</sup>, 김경중<sup>2</sup>

세종대학교<sup>1</sup>, 광주과학기술원<sup>2</sup>

piranas2067@gmail.com, taehwa-p@gist.ac.kr, kjkim@gist.ac.kr

## MultiModal Deep Learning for Autonomous Driving in Urban Environment

Wooil Shim<sup>1,0</sup>, Tae-Hwa Park<sup>2</sup>, Kyung-Joong Kim<sup>2</sup>

Depart of Computer Science and Engineering, Sejong University<sup>1</sup>

Institute of Integrated Technology, Gwangju Institute of Science and Technology<sup>2</sup>

### 요 약

자율주행 분야에서 카메라 RGB 이미지를 기반으로 하는 연구들이 많이 이루어지고 있다. 이는 다른 센서들에 비해 가격이 저렴하고, 사람의 운전 방식과 유사하기 때문이다. 하지만 카메라 RGB 이미지만을 사용하는 것은 에이전트의 앞에 어떤 물체가 있는지, 어느 정도 거리에 떨어져 있는지와 같은 주행에 있어서 필요한 정보들을 효율적으로 알지 못한다. 따라서 본 연구에서는 Segmentation 이미지와 Depth 이미지를 기반으로 한 멀티모달기반 딥러닝 모델을 제안한다. 대표적인 자율주행 자동차 시뮬레이터인 CARLA에서 학습 및 검증 실험을 진행하였고, 도심환경에서 자율주행 자동차가 다른 차량과 충돌하지 않으며, 움직이는 시나리오인 CARL 벤치마크 실험 결과 충분히 안전하게 주행함을 알 수 있었다.

### 1. 서 론

사람이 운전을 할 때는 전방에 어떤 물체가 있는지, 어느 정도의 거리에 떨어져 있는지, 교차로에서 어떤 길로 가야하는지 등 주변 상황을 파악한다. 예를 들어 전방에 차량이 멈춰있다면, 운전자는 가속 페달을 밟지 않을 것이고, 교차로에서 좌회전을 해야 하는 상황이라면, 핸들을 왼쪽으로 꺾을 것이다. 자율주행 분야에서도 사람의 주행 방식을 따라 RGB 이미지와 같은 시각적인 정보를 입력 데이터로 사용하여 종단간 학습 모델을 설계하려는 시도들이 많이 이루어지고 있다 [1][2][3][4][5]. 하지만 사람의 주행방식을 따라 하기 위해서 에이전트가 RGB 이미지와 같은 저차원 입력 데이터를 이용하여 학습하는 데는 한계가 있다. 그 이유는 종단간 학습 방법에서 RGB 이미지만을 이용할 경우에는 차량의 전방에 어떤 물체가 있는지 어느 정도 거리에 떨어져 있는지 효율적으로 알 수 없기 때문이다. 또한 그러한 정보를 알고 있어도 교차로에서 어느 길로 가야하는지와 같은 모호한 상황에서는 에이전트가 행동 결정을 하는데 어려움이 있다.

따라서 본 연구에서는 사람의 주행 방식과 유사한 객체 정보를 파악할 수 있는 Segmentation 이미지와 그 객체와의 거리를 시각화한 Depth 이미지, 내비게이션 정보를 이용한 멀티모달 기반 자율주행 딥러닝 모델을 제안한다. 제안한 모델은 행동에 따라 모방학습과 강화학습을 사용하여 에이전트가 안전하게 주행하는 것을 목표로 하였다. 모방학습과 강화학습을 같이 사용한 이유는 에이전트가 전문가 데이터를 통해 학습하여, 안전성과 다양한 시행착오를 통해 제안한 모델의 성능을 높이기 위함이다. 대표적인 도심환경 자율주행 자동차 시뮬레이터인 CARLA에서 학습 및 검증 실험을 진행하였고, CARLA 벤치마크를 통해 학습된 에이전트의 안전성 및 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 배경 및 관련연구에 대한 내용 3장은 제안하는 모델의 구조에 대한 설명, 4장은 실험 결과, 5장은 결론 및 향후 연구에 대한 내용을 기술한다.

### 2. 배경 및 관련 연구

#### 2.1 자율주행 연구

자율주행 관련 연구는 크게 두 가지 방향으로 나뉘어진다. 첫 번째는 전통적인 자율주행 방식으로, 주행 에이전트를 인식, 경로 계획, 제어와 같은 여러 가지 하위 작업으로 나누어 동작하게끔 하는 것이다[6][7]. 각 하위 작업이 순차적으로 동작하여 자동차 제어한다. 그와 반대로 입력 데이터에서 차량을 직접 매핑하려는 종단간 학습 방법이 있다. 학습한 에이전트는 입력 데이터를 통해 차량을 제어한다. 전문가의 주행 데이터를 활용하는 모방학습[1][2][3][5]이나 무수히 많은 시행착오를 통해 학습하는 강화학습[4]을 통해 에이전트를 학습한다. 본 연구에서는 종단간 학습 방법에 초점을 두었다.

#### 2.2 DDPG (Deep Deterministic Policy Gradient)[4]

DDPG는 심층 강화학습 알고리즘 중 연속 행동을 다루기 위해 개발된 알고리즘이다. DDPG는 Actor와 Critic라는 2개의 신경망으로 구성된다. Actor는 최적의 행동을 학습하고, Critic은 앞으로 얻을 보상의 기댓값을 최대화하는 방향으로 학습을 한다. Actor가 환경과 상호작용을 하며 그 과정에서 얻는  $(s_t, a_t, r_t, s_{t+1})$ 쌍을 리플레이 메모리에 저장하여 학습을 한다. Actor와 Critic 신경망의 손실함수는 아래와 같다.

$$y_t = r_t + \gamma Q(s_{t+1}, \mu(s_{t+1}|\theta^\mu)) - \theta^Q$$

수식 1 목표함수 정의

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

수식 2 Critic 손실함수의 정의

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) |_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s | \theta^\mu) \nabla_{s_i}$$

수식 3 Actor 손실함수의 정의

### 2.3 Conditional Imitation Learning(CIL)[5]

모방학습의 학습 방법은 전문가의 데이터를 학습하기 때문에 전문가가 없는 환경에서 성능이 좋지 못하다. 이러한 문제를 해결하기 위해서 그림 1과 같은 방법을 제안하였다.

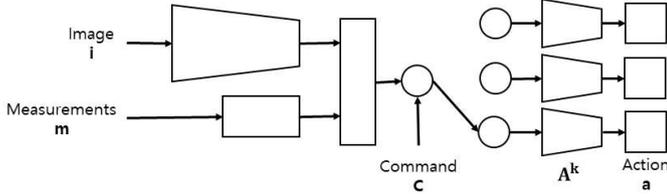


그림 1 Conditional Imitation Learning의 구조

그림 1의 구조는 이미지와 속도를 입력 받아 command(내비게이션 정보)에 따라 하위 신경망을 선택하는 구조이다. 그림 1의 구조를 통해 기존 모방학습보다 더 유연한 모델을 만들 수 있다.

### 3. 제안하는 모델

본 연구에서 제안한 모델은 그림 2와 같다. 행동을 출력하는 신경망을 그림 2와 같이 두 계층으로 나누었다. 입력  $i_1$ 는 그림 3의 (a)와 같은 카메라를 통해 입력된 Segmentation 이미지와 Depth 이미지이고,  $i_2$ 는 그림 3의 (b)와 같은 Segmentation 이미지이다. 입력 이미지 크기는 모두 [84, 84]이다. 출력은 핸들의 각도와 엑셀 두 가지이다.

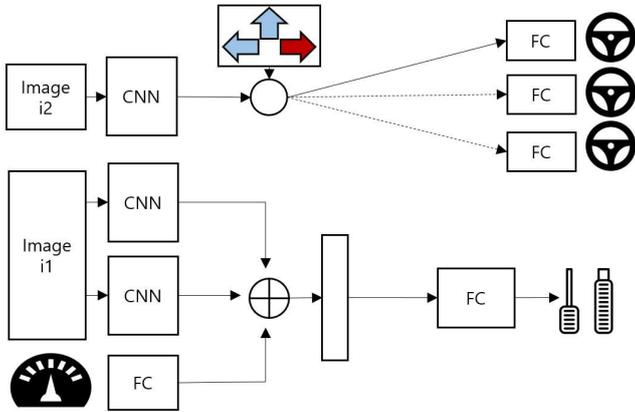
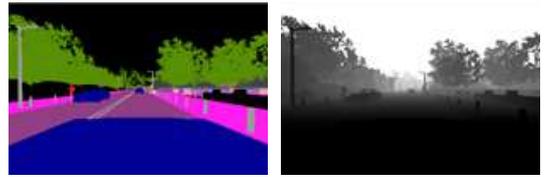


그림 2 제안하는 모델의 구조

모델의 세부 구조는 행동에 따른 신경망으로 구성하였다. 에이전트가 엑셀보다 핸들을 조종하기가 어렵다고 판단하여 핸들에 대한 행동은 모방학습을 통해 학습하고, 엑셀에 대한 행동은 강화학습을 통해 학습한다. 핸들에 대한 신경망은 도로와 차선만을 Segmentation 이미지와 내비게이션 정보(직진, 좌회전, 우회전 등)를 이용하여 핸들 각도를 제어한다. 하위 모듈은 CIL[5] 구조와 동일하게 직진, 좌회전, 우회전, 차선 유지 4개의 모듈이 있으며, 내비게이션 정보에 따라 하위 모듈을 선택하도록 구성하였다. 내비게이션 정보는 CARLA 시뮬레이터에서 제공하는 기능을 사용하였다. 엑셀에 대한 신경망은 Segmentation 이미지와 Depth 이미지, 현재 차량의 속도를 입력으로 최적의 엑셀 값을 출력한다.



(a)



(b)

그림 3 제안하는 모델의 입력  $i_1$ (a)와 입력  $i_2$ (b)

엑셀을 학습시키기 위한 강화학습은 DDPG[4] 알고리즘을 사용하였고, 보상함수는 수식 4와 같다.  $V$ 는 차량의 최고 속도를 1로 정의하여 0과 1사이로 정규화한 값이며,  $W$ 는 핸들의 각도를 [-0.5, 0.5] 사이로 클리핑한 값이다.  $C$ 는 차량의 충돌 여부를 나타낸다. 이를 통해 안전하면서 어느 정도의 속도를 낼 수 있다.

$$R = -(\cos 1.8\pi V) \times (\cos 2\pi W) - C \times 5$$

수식 4 강화학습 보상함수 정의

### 4. 실험



(a)



(b)



(c)



(d)

그림 4 Town 1 환경의 조감도(a) 및 실제 이미지(b)와 Town 2 환경의 조감도(c) 및 실제 이미지(d)

#### 4.1 실험 환경

실험을 위해 도심 환경 자율주행 시뮬레이터인 CARLA[8]를 사용하였다. CARLA는 오픈 소스 기반 자율주행 자동차 시뮬레이터로 다양한 주행 환경을 제공하고 있다. 또한 시뮬레이터에는 LiDAR, 카메라, GPS, 내비게이션 정보 등의 다양한 센서

정보를 제공하고 있으며, 사용자 요구에 따라 추가로 설정할 수 있다. 제안한 모델의 성능을 평가하기 위해 CARLA 시뮬레이터를 선택한 이유는 최근 자율주행 연구의 성능 평가 지표로 많이 사용되고 있기 때문이다.

그림 4의 (a)와 (c)는 실험에 사용한 환경들의 조감도이고, 그림 4의 (b)와 (d)는 각 환경의 실제 이미지이다. Town 1 환경은 총 2.9km 도로에 11개의 교차로가 있고, Town 2 환경은 총 1.4km 도로에 8개의 교차로가 존재한다. 본 실험은 맑은 날씨를 가정하고, 에이전트의 주행 성능을 평가하였다. 학습 단계에는 Town 1 환경에서만 학습을 하고, 평가 단계에서는 학습 단계에서 보지 못한 Town 2 환경에서 성능 평가를 하였다.

**4.2 실험 결과**

CARLA 벤치마크[8]를 통해 제안한 방법에 대한 성능 평가를 진행하였다. 벤치마크에는 표 1과 같이 4개의 Task가 있고, 난이도가 점점 올라가도록 구성되어 있다. 각 Task마다 정해진 시작지점과 도착지점이 있으며, 에이전트가 시작지점에서 시간 내에 도착지점에 도착하면 성공하는 시나리오이다.

표 1 CARLA 벤치마크 설명

	설명
Task1	에이전트가 직진을 잘하는지 평가
Task2	에이전트의 좌회전, 우회전을 잘하는지 평가
Task3	에이전트의 여러 번의 교차로를 통과하여 목적지에 도착하는지 평가
Task4	Task 3과 동일하지만, 환경의 복잡성이 커짐(다른 차량과 보행자가 있는 추가)

표 2 Town01 벤치마크 비교(표 안의 숫자는 성공률을 나타낸다.)

	MP	IL	RL	CIL	Ours
Task1	98	95	89	98	98
Task2	82	89	34	89	92
Task3	80	86	14	86	90
Task4	77	83	7	83	83

표 3 Town02 벤치마크 비교(표 안의 숫자는 성공률을 나타낸다.)

	MP	IL	RL	CIL	Ours
Task1	92	97	74	97	96
Task2	61	59	12	59	75
Task3	24	40	3	40	70
Task4	24	38	2	38	70

표 2와 표 3은 모델 별로 각 Task에서의 성공률을 나타낸 표이다. MP는 에이전트가 인식, 주행모델로 나뉘어 동작하는 모델이고, IL은 모방학습 한 모델, RL은 A3C[9] 알고리즘으로 학습한 모델이며, CIL[5]은 내비게이션 정보를 기반으로 한 모방학습 모델이다. 표 1에서는 학습 단계에서 충분히 Town 1 환경에 대해 학습하였기 때문에 강화학습(RL) 모델을 제외하고 높은 성능을 나타내었다. 강화학습 모델은 자율주행 환경의 복잡도가 높고, 에이전트의 행동의 범위가 크기 때문에 쉽게 학습하지 못했다. 하지만, 학습 단계에서 보지 못했던 Town 2 환경에서는 난이도가 높은 Task 3과 Task 4에서 제안한 모델을 제외하고 낮은 성능을 나타냈다. 그 이유는 대부분의 모델이 RGB 이미지를 기반으로 학습하였는데, Town 1과 Town 2에서 이미지의 특징들이 완전히 동일하지 않기 때문에 Town 2

에서의 성능이 높지 않다. 반면 제안한 모델은 Segmentation 정보와 Depth 정보를 이용하여 학습하였기 때문에 에이전트 전방에 어떤 객체가 어느 정도 거리에 떨어져 있는지 알 수 있어 다른 차량이나 보행자와의 충돌을 피할 수 있었다.

**5. 결론 및 향후 연구**

본 연구에서는 멀티모달 기반 자율주행을 위한 새로운 구조를 제안하였다. 표 2와 표 3에서 확인할 수 있듯이 제안한 모델이 높은 성능을 나타내었다. 본 연구를 통하여 Segmentation 정보와 Depth 정보를 융합하여 사용하는 방법이 단일 정보만 사용하는 방법보다 자율주행의 성능을 높일 수 있다는 것을 보였다. 본 연구에 사용한 Segmentation 이미지와 Depth 이미지는 시뮬레이터에서 제공하는 정확한 값이다. 따라서 도로 환경이 달라도 Segmentation 정보나 Depth 정보가 정확하게 입력된다. 향후 연구에는 RGB 이미지에서 Segmentation 이미지로 변형하고, Depth 이미지를 표현하는 방법을 추가하여 성능 비교를 해 볼 예정이다. 또한 전방 카메라로만 에이전트의 행동을 제어하도록 하였는데, 사람은 전방 뿐 아니라 다양한 방향의 시야를 이용하여 주행을 하기 때문에 다중 카메라를 사용한 연구도 진행할 예정이다.

**6. 감사의 글**

본 연구는 UD180026RD 위탁연구의 일환으로 방위사업청과 국방과학기술연구소의 지원으로 수행되었음 \* 교신 저자

**참고 문헌**

- [1] Pomerleau, Dean A. "Alvin: An autonomous land vehicle in a neural network." Advances in neural information processing systems. 1989.
- [2] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," arXiv:1712.00409, 2016.
- [3] Xu, Huazhe, et al. "End-to-end learning of driving models from large-scale video datasets." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
- [4] Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971 (2015).
- [5] Codevilla, Felipe, et al. "End-to-end driving via conditional imitation learning." 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018.
- [6] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.
- [7] Schwarting, Wilko, Javier Alonso-Mora, and Daniela Rus. "Planning and decision-making for autonomous vehicles." Annual Review of Control, Robotics, and Autonomous Systems (2018).
- [8] Dosovitskiy, Alexey, et al. "CARLA: An Open Urban Driving Simulator." Conference on Robot Learning. Pp1-16 2017.
- [9] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. 2016.