https://doi.org/10.7840/kics.2024.49.12.1671

심층 강화학습 기반 하이브리드 액션을 이용한 자율주행 차량의 고속도로 주행 판단 연구

김 성 준•, 신 규 민*, 전 준 서*, 방 지 윤*, 김 준 영**, 정 소 이°

A Study on Highway Driving Decision Making with Hybrid Action for Autonomous Vehicles Using Deep Reinforcement Learning

Seongjun Kim•, Kyu-min Shin*, Jun-seo Jeon*, Ji-yoon Bang*, Junyoung Kim**, Soyi Jung°

요 약

다차선 고속도로 환경에서는 다양한 교통 상황이 발생하며 교통 수칙을 준수하면서 주행해야 하며 이는 자율주행에서 매우 어려운 과제이다. motion planning에 대한 기존의 판단 방법은 규칙 기반 판단 방법으로 복잡한 환경에서 안전을 보장할 수 없으며, 이 문제를 해결하기 위해 심층 강화학습을 적용한 판단 방법에 관한 연구가 진행되고 있다. 본 논문에서는 심층 강화학습 속 하이브리드 액션을 이용한 proxiaml policy optimization(PPO)에 기반한 판단 방법을 제시한다. 심층 강화학습 모델의 상태 공간은 차선 정보, 속도를 포함한 자차량(ego)의 상태, 주변 차량의 상태를 포함하며, 연속 종 방향 액추에이터 값과 이산 횡 방향 차선 변경 판단을 출력으로 한다. 차선 변경을 위해 고정된 스티어링(steering)을 사용하는 것이 아닌 pure pursuit을 통해 액추에이터 단계의 스티어링을 제어한다. 100번의 테스트를 진행하여 충돌률을 포함한 평가 지료를 제시하며, 실험 결과는 하이브리드 액션을 이용한 에이전트가 연속 또는 이산 행동을 취하는 에이전트보다 더 안전하다는 결과를 보여준다

키워드 : 자율주행, 심층강화학습, 차로변경, PPO, 고속도로, 하이브리드 액션
Key Words : Autonomous Driving, Deep Reinforcement Learning, Lane-Change, PPO, Highway, Hybrid Action

ABSTRACT

Various traffic situations on multi-lane highways pose challenges for autonomous driving, requiring adherence to traffic rules. Traditional rule-based decision-making struggles with safety in complex environments, leading to research on deep reinforcement learning (DRL). This paper proposes a decision-making method based on proximal policy optimization (PPO) with hybrid actions. The DRL model inputs the states of the ego vehicle and surrounding vehicles, outputting continuous longitudinal control and discrete lateral lane changes. For lane changes, actuator-level steering is controlled via pure pursuit. Experiments show that agents with hybrid actions are safer than those using only continuous or discrete actions.

※ Equally Distributed
※ 본 연구는 2022년 한국연구재단의 지원을 받아 수행됨 (NRF-2022R1A2C2004869).
• First Author : Ajou University Department of Electrical and Computer Engineering, seongjun6935@gmail.com, 학생회원
° Corresponding Author : Ajou University Department of Electrical and Computer Engineering, sjung@ajou.ac.kr, 종신회원
* Department of Electrical and Computer Engineering, Ajou University, rbals1120@ajou.ac.kr; chevy1999@ajou.ac.kr; bangjun@ajou.ac.kr
** Department of Artificial Intelligence Convergence Network, Ajou University, junzero0615@ajou.ac.kr, 학생회원
논문번호 : 202407-146-A-RN, Received July 14, 2024; Revised August 16, 2024; Accepted September 5, 2024

Ⅰ. 서 론

자율주행에 대한 관심도가 커짐 따라 최근 자율주행 기술이 크게 발전하였으며, 안전, 교통 혼잡, 에너지, 환경 등 도로 위의 문제들을 해결하는 데 기여할 것으로 전망되고 있다[1]. 국제자동차기술자협회(The Society of Automotive Engineers, SAE)에서는 자율주행 시스템(Autonomous Driving Systems, ADS)을 Level 0부터 5까지 6개의 단계로 구분하고 있다[2]. 현재 상용화된 기술은 Level 2-3의 수준에 해당하며, 이를 넘어선 Level 4 이상의 자율주행 기술은 운전자가 필요 없는 자율주행을 의미한다[3]. 또한, Level 4 이상의 자율주행에서부터는 시스템이 판단하기 때문에 그에 따라 판단에 대한 고도화된 기술을 필요로 하게 된다.

자율주행 시스템(ADS)은 일반적으로 크게 센싱(sensing), 인지(perception), 측위(localization), 맵 생성(map building), 경로 생성 및 판단(planning & decision-making), 제어(control)의 모듈로 구성된다[4,5]. 특히, 판단은 차선 변경, 다른 차량 추월 결정 등을 하는 가장 결정적인 모듈 중 하나로 잘못된 판단은 사고로 이어질 수 있으며, 실제 세계에 대한 불확실성 및 복잡도로 인하여 자율주행 차량의 판단은 여전히 어려운 과제로 남아 있다[6,7]. 특히 고속도로에서는 다양한 교통 상황에서 교통 수칙을 준수하면서 다른 차량을 회피해서 주행하는 시나리오는 사람도 쉽게 판단하기 어려운 문제이다[8].

현재까지 판단에 대한 문제를 해결하기 위해 다양한 연구가 진행되었다. 판단은 3개의 방법으로 분류할 수 있다. 첫 번째 방법은 규칙 기반(rule-based)의 판단이다. Stanford 대학에서 Volkswagen 회사와 협업하여 개발한 자율주행 차량에서는 finite state machine (FSM)을 사용하여 판단하였고[9], defense advanced research projects agency (DARPA) 챌린지에서는 intelligent driver model (IDM)과 minimizing overall braking induced by lane changes (MOBIL)을 적용하여 성공적인 판단을 이루었다[10,11]. 또한, Pérez는 퍼지 논리에 기반하여 자율주행 차량의 추월 결정에 대한 방법을 제시하였다. 규칙 기반의 판단 방법은 구현하기가 간단하다는 장점이 존재하지만, 이를 구현하기 위해서 방대한 사전 지식과 여러 교통 상황들이 필요하다는 한계가 있다. 또한 규칙 기반의 판단 방법으로는 복잡한 교통 상황을 모두 표현하기에는 어려움이 존재하며, 정의되지 않는 상황에 대해서는 안전한 주행 판단에 대하여 보장할 수 없게 된다. 앞서 언급한 Stanford에서 FSM에 기반하여 개발한 자율주행 차량의 경우, 테스트 주행 시 FSM에서 정의하지 않는 상황이 발생하였다[9].

두 번째 방법은 불확실성을 다루기 위한 확률 기반의 방법으로, Schubert와 Wanielik은 베이지안 네트워크를 사용하여 인지부터 판단까지의 불확실성을 고려한 판단 방법을 제시하였다. 하지만 이 역시 복잡한 모델과 동적인 결정 과제에 관한 적용이 어렵다는 한계가 존재한다[6].

마지막으로 머신러닝(Machine Learning, ML)에 기반한 방법이다. 머신러닝은 크게 지도학습(supervised learning), 비지도 학습(unsupervised learning), 강화학습(reinforcement learning)으로 구분할 수 있으며, 지도학습은 사람의 주행 데이터를 기반으로 판단에 대한 정책을 학습할 수 있지만 적용이 어렵다는 한계점이 존재한다. 딥러닝은 학습이 가능할 정도의 방대한 양의 데이터가 필요하며, 이를 위한 데이터 취득은 굉장히 시간적으로 오래 걸리는 일이며[12], 공개된 데이터셋에 대해서는 인지를 위한 데이터셋으로 판단에 대한 적용이 어렵다. 판단은 주변 환경뿐만 아니라 운전자의 심리적 요소까지 고려해야 하는 요소로 이는 데이터를 취득하고 이를 정량화하여 라벨링 하는 것 역시 한계가 존재한다[5]. 이와 다르게 강화학습은 환경과 상호작용하여 스스로 학습하기 때문에 불확실성과 복잡도에 대해 강인한 판단이 가능한 장점을 기반으로, 강화학습 기반 판단 결정 연구가 진행되고 있다[7,13].

불확실성 및 복잡한 교통 상황에서 자율주행 차량의 안전한 판단을 위해서 강화학습 연구가 활발히 진행되고 있다[14]. 고속도로 환경 역시 자율주행 차진행되었다. 행동 공간(action space)으로 목표 차선의 좌표 (x, y)를 정의하였으며[12], 차선 변경 및 종 방향에 대한 가속과 감속을 행동 공간을 이산 및 연속으로 정의하여 최적화하였다[6,10-21]. 또한, 행동 공간에 대해서 가속에 대한 이산 행동에 대해서 다시 크기에 따라서 여러 개로 나누는 방법을 제시하였다[16,20]. 하지만, 위에서 언급한 방법들은 모두 한 번에 종 방향 또는 횡 방향에 관한 결정만을 내리게 되며, 일반적으로 차량은 종 방향과 횡 방향 동시에 결정을 내리게 된다. 또한, 종 방향과 횡 방향에 대해서 모두 이산적이거나 연속적인 행동 결정을 내리고 있으며, 이는 이 둘을 혼합한 행동보다 더 좋은 성능을 보인다고 확신할 수 없다. 이에 본 논문에서는, 에이전트의 행동 공간을 다양한 형태로 구분하여 실험을 진행하며 성능의 우수성을 검증하였다.

나머지 논문의 구성은 다음과 같다. 2장에서는 심층 강화학습 방법론에 관해서 소개하며, 3장에서는 논문에서 제안하는 hybrid action의 설계 및 나머지 행동 공간에 대하여 설명한다. 4장에서는 보상 설계를 위한 환경

구성 및 학습을 위한 시나리오 구성과 방법에 대해 설명한다. 또한, 각 행동 공간별 충돌률, 평균 속도, 평균 차로 오차를 통해 결과를 비교하며, 마지막 5장에서는 결론을 내린다.

Ⅱ. 심층 강화학습 방법론

2.1 마르코프 결정 프로세스

강화학습은 환경과 상호 작용하면서 에이전트 또는 프로세스를 제어하는 알고리즘이다. 에이전트는 환경으로부터 상태와 보상을 받고 환경과의 상호 작용 과정에서 누적된 long-term 수익을 극대화하고자 한다[15].

마르코프 결정 프로세스(Markov Decision Process, MDP)는 강화학습 문제에 대한 기본적인 공식이다. MDP는 $\{S, A, P, R, \gamma\}$로 정의할 수 있다. S는 상태 공간(state space), A는 행동 공간(action space), R은 보상 함수(reward function), P는 상태 전이 확률(state transition probability), γ은 감쇠인자(discount factor)이다. 정책(policy) π는 모든 상태에서 행동을 선택할 때 사용되는 상태 공간에서 행동 공간으로 매핑(mapping)하는 함수$\pi : S \rightarrow A$이다[2]. 강화학습의 목표는 기대 보상을 최대화하는 최적 정책 π^*를 찾는 것이다.

2.2 가치 기반 강화학습 기법

가치 기반 강화학습 기법은 가치 함수를 Q-함수로 나타내어, 최적의 가치 함수를 찾는 것을 목적으로 한다[22]. Q-함수는 주어진 상태에서 각 행동의 예상 보상을 평가하는 예측 함수로, 수식 (1)과 같이 벨만(bellman) 방정식을 이용하여 업데이트된다[19].

$$Q^*(s,a) = E_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') | (s,a)\right] \tag{1}$$

수식(1)의 가치 함수를 신경망 파라미터 θ를 기반으로 나타낸 알고리즘을 deep Q-network (DQN)이라 하며, 기존의 Q-학습은 Q-가치를 Q-table에 저장해서 학습했던 반면, DQN에서는 신경망의 출력이 Q-가치를 잘 나타내도록 수식 (2)와 같이 신경망을 갱신하는 방법을 이용한다[5].

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \tau \max_{a' \in A} Q(s',a') - Q(s,a)\right] \tag{2}$$

2.3 정책 기반 강화학습 기법

가치 기반 강화학습 기법은 반드시 행동 공간을 이산적으로 정의해야 한다. 하지만 실제 세계에서는 이산적으로 행동 공간을 나타내기 어려우며, 이산 행동 공간의 차원을 매우 크게 하여 연속적으로 나타낼 수 있다. 하지만 이 방법은 "차원의 저주"에 의해 적용되기 어려우며, 따라서 가치 기반 학습 기법이 아닌 정책 기반 강화학습 기법에 관한 연구가 제시되었다[22].

정책 기반 강화학습 기법은 주어진 상태에서 직접적으로 정책을 추정하는 방법으로, 일반적으로 정책은 신경망 파라미터에 의해 $\pi_\theta(a|s)$로 정의될 수 있다[18]. 따라서, 정책 기반 강화학습 기법은 발생하는 가치 함수의 변화를 토대로 파라미터 θ를 갱신하여, 최적의 정책을 찾고자 하며, 최적의 정책은 수식 (3)과 같은 목적함수 $J(\pi_\theta)$를 최대화하는 θ를 찾는 것이 목적이다.

$$J(\pi_\theta) = E[\sum_{t=0}^{T} r_t] \tag{3}$$

수식 (4)를 이용하여 최적의 정책을 찾기 위해 목적함수 $J(\pi_\theta)$를 갱신해야 하며, 이를 위해서 policy gradient (PG)라는 방식이 사용되어 왔다. 이는 목적함수 $J(\pi_\theta)$를 최대화하기 위해 경사 상승(gradient ascent)을 통해 수식 (5)처럼 갱신하게 된다[19,23].

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta) \tag{4}$$

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q_\pi(s,a)] \tag{5}$$

Multi-step MDP에서는 $Q_\pi(s,a)$을 학습의 bias를 줄이기 위해서 advantage function $A(s,a)$를 사용하여 나타낼 수 있으며, 이를 monte-carlo PG라 한다[19,23].

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) A(s,a)] \tag{6}$$

2.4 액터-크리틱 강화학습 기법

액터-크리틱(actor-critic) 강화학습 기법은 정책 기반 학습 기법과 가치 기반 강화학습의 이점을 합친 기법이다[10]. 액터는 환경과 상호작용하여 정책을 생성하고 행동을 선택하는 역할을 하며, 크리틱은 가치 함수에 기반하여 액터의 정책을 평가하게 된다[22,24]. 본 논문에서는 액터-크리틱 강화학습 기법 중 하나인 proximal policy optimization(PPO)을 통해 횡 방향 이산 행동, 종 방향 연속 행동을 출력으로 하는 hybrid action을

제시하고자 한다.

Ⅲ. 심층 강화학습 기반 주행 판단 모델

3.1 시스템 아키텍처

본 장에서는 시뮬레이션 기반의 자율주행 시스템의 아키텍처에 관해 설명한다. 본 시스템은 3차선 고속도로 환경에서 에이전트 차량이 주행하는 상황이며, 아키텍처는 그림 1과 같으며, 그림 2에서는 주행 판단 모델에 대한 의사결정 코드를 제시한다. 시스템은 크게 3가지로 구분할 수 있다. 먼저 에이전트 차량이 환경과 상호작용하여 상태를 입력으로 하여 네트워크를 통과하면서 에이전트 차량의 행동에 대한 메뉴버(maneuver)를 출력으로 내보내는 부분이 있다. 두 번째는 메뉴버를 바탕으로 에이전트 차량의 액추에이터(actuator)로 넣기 위한 세부 입력을 계산하는 부분이 존재하며, 마지막으로 에피소드가 새롭게 시작될 때마다 시나리오를 불러오는 부분이 구성된다.

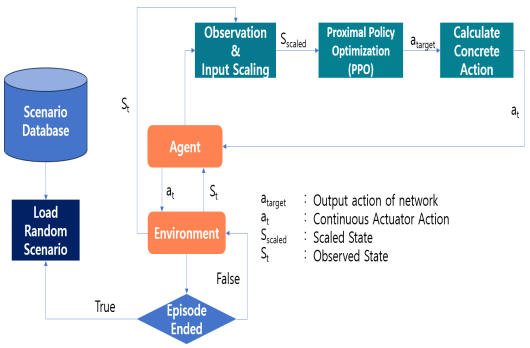

그림 1. 심층 강화학습 기반 주행 판단 모델 아키텍처
Fig. 1. Decision-making model based on deep reinforcement learning architecture.

Algorithm 1 Decision-making model based on deep reinforcement learning

```
for step = 1 to N_e do
  if episode end then
    Select a scenario and ego's start lane randomly
    if episode > N then
      Activate {1,...,N_T} cars randomly
    end if
  end if
  Other cars control based on the selected scenario
  Get current state S_t
  for Surrounding Vehicle = 1 to 6 do
    if Surrounding Vehicle is empty then
      dx_s ← 2 × x_threshold
      dy_s ← 0
      dv_s ← v_max
    end if
  end for
  Scaling S_t by division
  Make decision according to a_t ∈ [a_x, y_lc]
  Take input a_x to longitudinal actuator
  Calculate lateral actuator value based on Pure pursuit
  Get the next state S_{t+1} and r_t(s, a)
  if collision or reach to maximum distance then
    episode end
  end if
end for
```

그림 2. 심층 강화학습 기반 주행 판단 모델 의사결정 코드
Fig. 2. Decision-making model based on deep reinforcement learning pseudo-code.

자율주행 차량은 LiDAR와 카메라 센서 등 여러 센서를 이용하여 주변 환경을 인지할 수 있다. 객체 탐지 알고리즘은 주변 차량의 상대 위치 x, y를 인지할 수 있으며, 이를 트래킹(tracking) 알고리즘을 통해 상대 속도를 추정할 수 있다. 또한, 측위를 통하여 정밀지도에서 에이전트의 위치를 인지할 수 있으며, 정밀지도는 신호등, 차로 중심 등 다양한 정보를 포함한다[5].

본 논문에서는 자율주행 차량의 판단에 대한 검증을 하기 위하여 인지 알고리즘을 사용하는 것이 아닌 주변 차량의 위치와 속도, 그리고 차로 중심에 대해서 시뮬레이터 정보를 사용하여 검증한다.

3.2 시스템 모델 기반 마르코프 결정 프로세스

강화학습은 MDP로 정의되는 문제를 푸는 방법론으로 도로 위의 차량의 주행을 제어하기 위해 이 장에서는 MDP를 정의하였다.

3.2.1 상태

상태 공간은 크게 2가지 측면으로 정의하였다. 먼저, 에이전트 차량의 상태를 기준으로 속도(v_{ego}), 각 차로의 중심과 에이전트 차량 중심의 y 좌표 차이(dLy_i), 에이전트 차량이 주행하고자 하는 목적 차로 중심과의 y 좌표 차이(dL_{target})를 정의하였다. 두 번째로는 주변 차량(Surrounding Vehicles, SV)에 관한 상태로 에이전트 차량과의 x, y 좌표 차이(dx_s,dy_s), 속도 차이(dv_s)로 정의하였다.

$$S = [v_{ego}, dLy_i, dL_{target}, dx_s, dy_s, dv_s] \quad (7)$$
$$i = 1,2,3,\ s = 1,2,\ldots,6$$

3.2.2 행동

행동 공간은 종 방향과 횡 방향을 구분하여 나타내었으며, 속도 제어를 위해 종 방향 행동(a_x)은 액추에이터의 제어 값으로 직접 사용되며, 범위는 [-1, 1]이다. 그림 3과 같이 횡 방향은 차량의 현재 차로를 기준으로 목표 차로를 설정(y_{lc})하게 된다. 차선 유지(Lane Keeping, LK), 좌측 차선 변경(Lane Change Left, LCL), 우측 차선 변경(Lane Change Right, LCR)으로 수식 (8)과 같이 정의한다.

$$A = [a_x, y_{lc}],\ \ y_{lc} \in \{LK, LCL, LCR\} \quad (8)$$

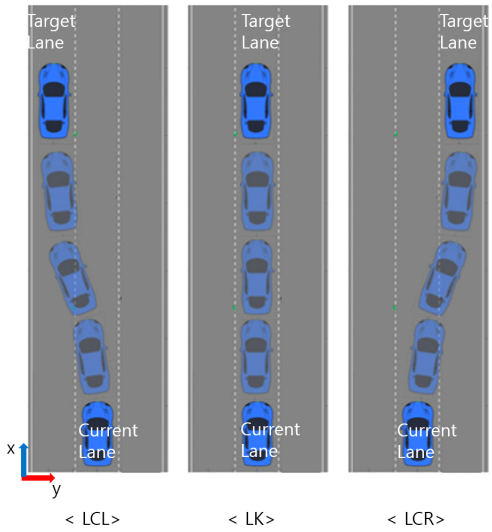

그림 3. 행동 공간 정의
Fig. 3. Definition of action space.

3.2.3 보상

보상 함수는 총 5가지 측면에서 정의하였으며, 모든 보상 상수($C_{object \in collision, speed, laneError, lc, wd}$)는 양수로 정의하였다. 먼저, 충돌을 방지하기 위해 다른 차량 또는 가드레일(guard-rail)과 충돌 상황을 방지하기 위해서 수식 (10)와 같이 충돌 상황 발생 시 음의 보상을 부여하였다[7]. 두 번째로 속도 측면에 있어서 미리 설정한 에이전트 차량의 제한 속도 범위 $[v_{\min}, v_{\max}]$ 내에서 높은 속도에 대해서 높은 보상을 주기 위해서 에이전트 차량 속도(v)에 대해서 수식 (11)와 같이 나타내었다[7,9]. 에이전트 차량은 수식 (8)에서 정의한 대로 횡 방향에 대해서 현재 차로를 유지하거나 양옆 차로로 차선 변경을 하려 한다. 또한 차선 변경을 하는 중에 현재 차로를 유지하는 행동을 하고자 할 때, 차량은 양옆 차로로 차선 변경을 하지 못하고, 현재 차로의 중심이 아닌 바깥에서 주행을 하게 된다. 따라서 이를 방지하기 위해 수식 (12)와 같이 목표 차로의 y좌표와 현재 차로의 y좌표 차이(d_{lane})의 크기에 비례하여 음의 보상을 부여하도록 설정하였다. 다음으로는 잦은 차선 변경을 제한하기 위해서 차선 변경을 하여 옆 차로로 이동할 때마다 수식 (13)과 같이 음의 보상을 부여하였다[15]. 에이전트 차량의 학습의 속도를 높이기 위해서 양 끝 차로에 있을 때 가드레일 방향으로 차선 변경을 하지 않고 주행하는 차로를 유지하도록 하였다. 즉, 1차로에서 LCL, 3차로에서 LCR 일 때 LK를 하도록 했다. 따라서 이러한 상황이 발생하였을 경우, 수식 (14)와 같이 음의 보상을 부여하도록 설정하였으며[15], 총 보상(r_t)은 수식 (9)와 같이 이를 모두 더한 결과이다[7].

$$r_t = r_{collision} + r_{speed} + r_{laneError} + r_{lc} + r_{wd} \tag{9}$$

$$r_{collision} = \begin{cases} -C_{collision}, \text{if } collision \\ 0 \quad\quad\quad\quad, otherwise \end{cases} \tag{10}$$

$$r_{speed} = C_{speed} \frac{v - v_{\min}}{v_{\max} - v_{\min}} \tag{11}$$

$$r_{laneError} = -C_{laneError} * |d_{lane}| \tag{12}$$

$$r_{lc} = \begin{cases} -C_{lc}, \text{if } lane\ change \\ 0 \quad\quad\quad, otherwise \end{cases} \tag{13}$$

$$r_{wd} = \begin{cases} -C_{wd}, \text{if } wrong\ decision \\ 0 \quad\quad\quad, otherwise \end{cases} \tag{14}$$

3.3 PPO 구조도

PPO는 정책이 갱신되는 범위를 제한하여 PG에서 발생하는 과한 갱신을 방지하여 안정적인 신경망을 갱신할 수 있게 하는 알고리즘이다[23].

본 논문에서는 효율적으로 학습시키기 위해서 상태 공간(S_t)을 스케일링(scaling)하여 그 결과(S_{scaled})를 네트워크 입력으로 사용하도록 입력 스케일링을 진행한다. 또한, 하나의 모델에서 연속과 이산 행동을 동시에 사용하기 위한 심층 네트워크 구조를 제시한다.

3.3.1 입력 스케일링

수식 (7)에서 정의한 상태 공간을 스케일링해주었으며, 스케일링은 각 상태를 특정 값으로 나눠 주어 네트워크의 입력을 구성하는 상태 공간끼리 차원(dimension)의 차이가 크지 않도록 설정하였다. 특정 값은 표 1을 참고하였으며, $x_{threshold}$는 주변 차량의 인지 범위를 의미한다.

표 1. 딥러닝 네트워크 입력장치 스케일링
Table 1. Input Scaling to the neural network.

v_{ego}	$v_{\max}$
dLy_i	$Lane\ Width$
dL_{target}	$Lane\ Width$
dx_s	$x_{threshold}$
dy_s	$Lane\ Width$
dv_s	$v_{\min}$

3.3.2 심층 네트워크 구조

하나의 딥러닝 구조에서 이산행동과 연속 행동을 동시에 출력을 나타내어야 한다. 그림 4는 네트워크 구조를 시각화하여 나타내고 있다. 네트워크는 23개의 입력층과 256개의 노드로 구성되는 은닉층 3개로 구성되어 있다. 또한, 이산 행동 공간과 연속 행동 공간에 대해서 따로 출력을 내기 위해서 마지막 은닉층의 출력을 이산 행동 공간은 노드의 크기가 3인 은닉층으로, 연속 행동 공간의 경우 노드의 크기가 1인 은닉층을 통과하게 된다. 또한, 각 행동 공간의 출력을 내보내기 위해서 이산 행동 공간은 softmax를 거치게 되고 연속 행동 공간의 경우 가우시안 샘플링(gaussian sampling)을 거친 후 [-1,1]로 클리핑(clipping) 후 최종적으로 네트워크에서 출력을 산출한다.

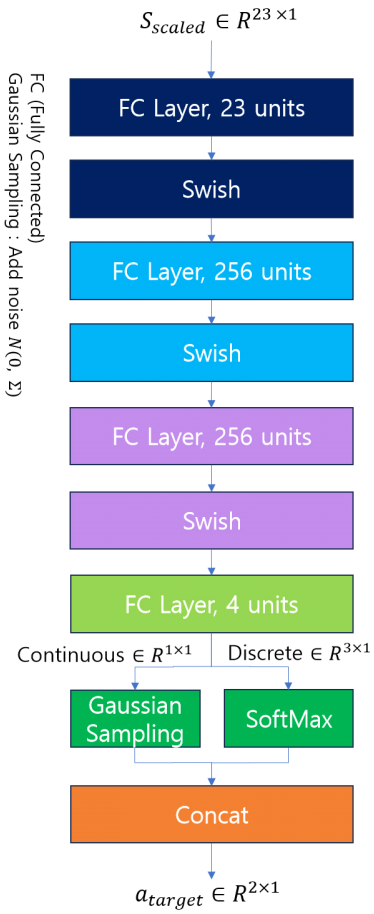

그림 4. 심층 뉴럴 네트워크 구조
Fig. 4. Structure of deep neural network.

3.4 Other MDP 정의

본 논문에서 제시하는 hybrid action은 차량 액추에이터의 종 방향은 연속 행동, 횡 방향은 이산 행동으로 정의하여 고속도로 환경을 주행한다. 종 방향은 네트워크의 출력으로 닫힌 구간 [-1, 1]에서의 연속적인 값이 결과로 나오게 된다. 이 결과가 양수일 때는 스로틀(throttle)을 제어하며, 음수일 때는 브레이크(brake)를 제어하도록 설정하였다. 횡 방향은 네트워크의 결과가 차선 변경에 대한 3가지 이산 행동으로 목표 차로를 주행하기 위해 고정된 스티어링(steering)이 아닌 pure pursuit 방법을 통해 스티어링을 계산하여 제어하도록 하였다.

본 논문에서는 hybrid action 외의 다른 행동 유형을 비교하여 hybrid action을 최적의 행동 유형으로 제시하는 바이다. 하지만 hybrid action의 MDP는 다른 행동 유형에 적용되기 어려운 부분이 존재한다. 따라서 본 장에서는 single discrete, double discrete, only continuous의 행동 유형에 대해서 MDP 정의를 내린다.

표 2에서는 다른 행동 유형의 상태에 대해서 나타내고 있으며, single discrete는 이산적으로 종 방향과 횡 방향의 행동 공간이 하나의 차원에서 정의되어 동시에 종 방향 또는 횡 방향 결정을 내리도록 하는 방식이다. 횡 방향 행동 공간에 대해서는 hybrid action과 동일하게 LCL, LCR, LK로 구성되며, 종 방향에 대해서 ACC와 DEC는 각각 가속과 감속을 나타내며 이를 포함하여 크기가 5인 행동 공간을 나타낸다. 또한, 가속과 감속에 대해서 에이전트 차량의 현재 목표 속도에서 2.5km/h를 더하거나 빼서 목표 속도를 수정하도록 하였으며, 액추에이터에 대한 종 방향 제어는 비례-적분-미분(Proportional Integral Differential, PID)제어를 사용하였다. 에피소드 시작 에이전트의 목표 속도는 80km/h로 설정하였다.

Double discrete는 이산적으로 종 방향과 횡 방향을 동시에 결정을 내릴 수 있도록 다른 차원에서 정의되었으며, 횡 방향 행동 공간은 hybrid action과 동일하다. 또한, 종 방향의 경우 single discrete에서 NONE이 추가되었으며, NONE은 어떠한 가속 또는 감속을 하지

표 2. 각 행동 유형 별 행동 공간 정의
Table 2. Definition of action space of other action types.

Action types	Actions space
single discrete	$[LCL, LCR, LK, ACC, DEC]$
double discrete	$[[ACC, DEC, NONE]$ $,[LCL, LCR, LK]]$
only continuous	$[a_x, steer]$

않는 행동을 의미한다.

Only continuous는 종 방향과 횡 방향 행동 공간이 연속적으로 따로 정의되어 있으며, 종 방향은 hybrid action과 동일하게 제어하고, 횡 방향은 에이전트 차량의 액추에이터의 스티어링의 제어 값을 출력으로 내보낸다. 또한, 스티어링의 최대 범위는 1.5°로 설정하였다.

Only continuous에서는 dL_{target}을 설정할 수 없다. 따라서 표 3과 같이 각 행동 유형별 MDP 상태 공간을 다르게 하였으며, discrete 유형에 대해서는 dv_{target}을 추가하였고, 이는 에이전트 차량의 목표 속도와 현재 속도의 차이를 나타낸다. 또한, only continuous 유형은 가드레일까지의 거리(dGy_j)와 헤딩(ψ_{ego})을 추가하였다.

표 3. 각 행동 유형 별 상태 공간 정의
Table 3. Definition of state space of other action types.

Action types	State(S_t)
single & double discrete	$[v_{ego}, dL_{target}, dv_{target}, dLy_i, dx_s, dy_s, dv_s] \in R^{24}$
only continuous	$[v_{ego}, \psi_{ego}, dLy_i, dGy_j, dx_s, dy_s, dv_s] \in R^{25}$

Ⅳ. 시뮬레이션 결과

4.1 유니티 시뮬레이션 환경

본 연구에서는 ML-Unity를 활용하여 시뮬레이터 환경을 구성하였으며, 3차선 고속도로에서 에피소드 당 최대 1,500m를 주행하도록 설정하였다. 차량은 폭은 1.7m, 축거는 4.0m, 최대 스티어링은 25°의 차량을 활용하여 학습하였다. 주행 환경을 구성하는 파라미터로 $v_{\max}$는 120km/h로 설정하였으며, $v_{\min}$은 60km/h로 설정하였다. 또한, 제어기는 50Hz, 주변 차량 인지는 20Hz, 판단은 10Hz마다 갱신하게 설정하였다.

그림 5. 차량 주행 시뮬레이터 환경
Fig. 5. Vehicles driving simulator environment.

시뮬레이션 환경을 구성하기 위해서 unity asset을 사용하였으며, 차량은 'NWH Vehjcle Physic2', 3차선 고속도로 환경은 'Modular Highways & Freeways'을 사용하였다. 또한 'Box Collider'를 사용하여, 차량 간의 충돌과 가드레일과의 충돌을 센싱하였다. 그림 5는 차량 에이전트의 정면과 bird eye view(BEV)를 시각적으로 보여주고 있다.

4.1.1 차량 상태 정의

MDP의 수식 (7) 상태 공간 정의에서 주변 차량(SV)의 상태를 사용하며, 수식 (13)에서는 차선 변경에 대하여 보상 함수를 정의하고 있다. 따라서 이 장에서는 주변 차량과 차선 변경에 대한 세부적인 정의를 내리도록 한다.

주변 차량은 그림 6처럼 에이전트 차량을 기준으로 양옆 차선과 종 방향으로 x 좌표 차이 100m($x_{threshold}$)를 기준으로 하여 설정하였으며, 그림 6과 같이 주변 차량의 인덱스 s에 recognition을 설정한다. 주변 차량이 네트워크의 입력으로 들어가는 순서를 지정하여 주변 차량의 상태에 대한 순서로 인한 다양성을 줄이려 시도하였다. s=1부터 6까지 front vehicle left (FVL), front vehicle in-lane (FVI), front vehicle right (FVR), rear vehicle left (RVL), rear vehicle in-lane (RVI), rear vehicle right (RVR)로 recognition을 설정한다. 또한, 에이전트 차량이 주행하는 과정에서 항상 해당 recognition의 주변 차량이 존재하지 않으며, 이때는 충분히 큰 값으로 ($[dx_s, dy_s, dv_s] = [200m, 0m, 100km/h]$) 초기화를 진행한다. 차선 변경의 경우 에이전트 차량의 중심을 기준으로 차선을 넘었을 때를 차선 변경으로 정의하였다.

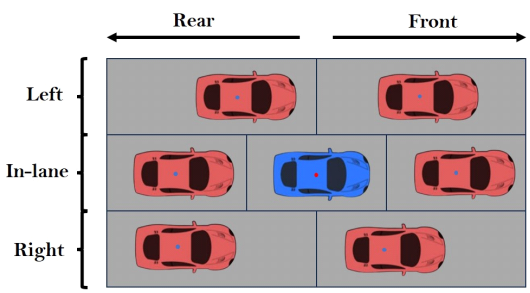

그림 6. 주변 차량 recognition 정의
Fig. 6. Definition of surrounding vehicles.

4.2 시뮬레이션 시나리오

일반적으로 고속도로 위에서 차량은 가속 또는 감속 및 차선 변경을 할 수 있다. 본 논문에서는 차량이 정속

으로 한 차로만을 주행하는 상황만을 시나리오로 학습하여 4가지 행동 유형을 평가하고자 하며, 추후 주변 차량의 급감속과 끼어들기 등에 대한 상황을 탐색하고자 한다. Mei Zhang과 Kai Chen은 2차로부터 4차로까지 주변 차량이 차선 변경 없이 정속 주행하는 시나리오를 6개 제시하였으며[4], 본 논문에서는 4차로에 대해서 좌측 3개 차로와 우측 3개 차로로 나누어 그림 7과 같이 8개의 시나리오로 재구성하였다. 에피소드가 시작될 때마다 무작위로 8개 중에 하나의 시나리오를 선택하도록 하여 학습을 진행한다. 시나리오가 선택된 이후에 에이전트 차량의 시작 차선을 무작위로 선택하여 시나리오에 대해서 더 다양하게 학습하도록 설정하였다. 또한, 1,500번의 에피소드를 학습했을 경우 시나리오를 구성하는 차량의 수를 두 대 이상이 구성되게 무작위로 선택하여 시나리오를 구성하여, 다양한 상황에 대해 학습하도록 시도하였다. 총 1,000,000 step을 학습시켰으며, Test의 경우에는 총 100번의 에피소드를 구성하여 결과를 테스트하였다.

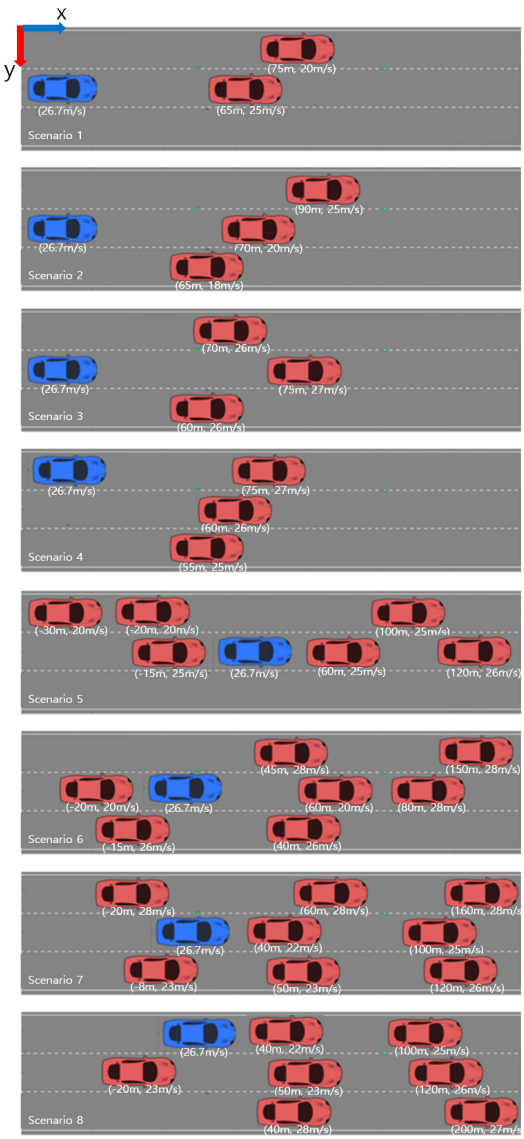

그림 7. 학습 및 테스트를 위한 시나리오 구성
Fig. 7. Configuration of scenarios for train and test.

4.3 시뮬레이션 결과

본 논문에서는 강화학습의 학습 과정을 평가하기 위해 에피소드 진행에 따른 누적 보상의 크기 변화 양상과 정책 및 가치 손실에 대한 변화 양상을 확인한 후 학습 과정에 대해서 추가 평가 지표로 학습의 진행에 따른 평균 속도에 대한 양상과 평균 차로 오차, 차선 변경 수, 50번 에피소드 당 충돌 횟수를 평가한다. Test에 대한 평가 지표로는 충돌률과 평균 속도, 평균 차로 오차를 제시하여, 결과를 평가하도록 한다.

4.3.1 학습에 따른 보상 및 손실 양상

본 논문에서 제시하는 hybrid action이 누적 보상값이 제일 높게 형성되는 것을 그림 8에서 확인할 수 있으며, 가치 손실은 그림 9와 같이 에피소드가 진행됨에 따라 작아지는 것을 확인할 수 있다. 하지만, 학습이 약 700k가 되는 지점에서 누적 보상은 감소하고, 손실은 증가하는 양상을 보이는데, 이는 학습한 에피소드가 1500번을 넘어가는 지점으로 시나리오를 구성하는 차량의 수를 무작위로 선택하는 지점이다. 따라서 이를 통해 기존의 시나리오에 대해서 과적합되어 학습했음을 추정할 수 있으며, 강화학습에서 탐색은 학습 초기에는 빈번하게 일어나지만, 학습 후반에서는 탐색보다는

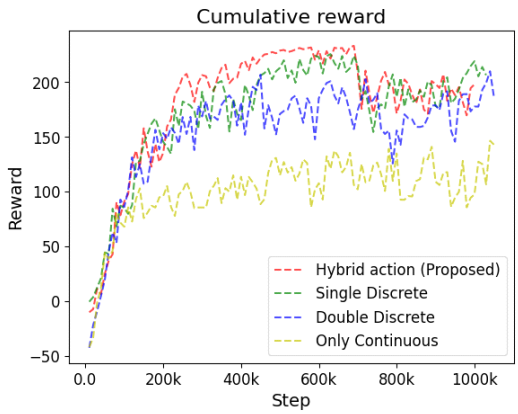

그림 8. 학습 누적 보상 결과
Fig. 8. Cumulative reward of train.

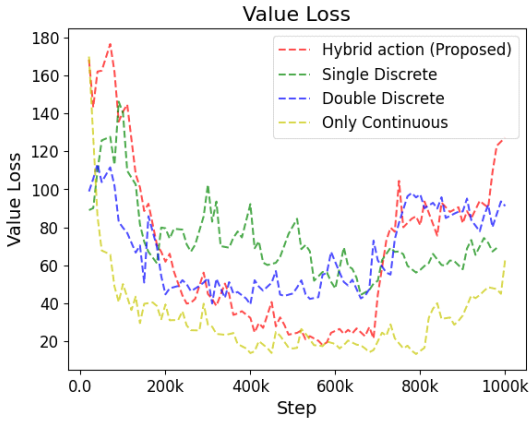

그림 9. 학습 가치 손실 결과
Fig. 9. Value loss of train.

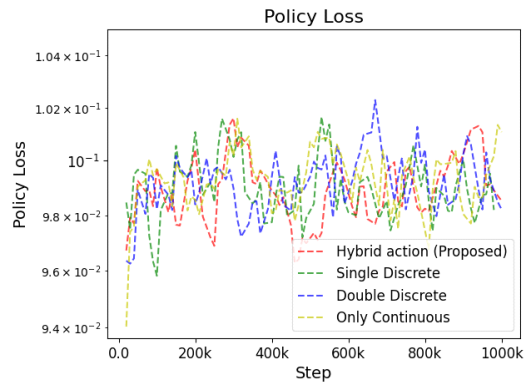

그림 10. 학습 정책 손실 결과
Fig. 10. Policy loss of train.

활용을 주로 행하기 때문에 학습의 개선이 느리게 일어나고 있다.

4.3.2 행동 유형별 학습 결과

1,500번 지점부터 학습의 성능을 비교하기 어려우므로 700k의 step을 기준으로 하여 이전까지의 학습 결과를 비교하였다. 그림 11-14와 같이 학습의 과정에서 평균 속도, 차로 변경 수, 평균 차로 오차, 50 에피소드 별 충돌 횟수 모두 개선되고 있음을 확인할 수 있다. 하지만, only continuous의 경우 평균 속도, 평균 차로 오차 모두 다른 행동 유형보다 성능이 낮다. 이는 다른 행동 유형은 횡 방향 행동 공간에 대해서 목표 차로를 설정하여 제어하기 때문에 가드레일과의 충돌이 없으며, 차로 중앙을 유지하려 한다. 반면에 only continuous는 횡 방향을 목표 차로가 아닌 연속 행동을 액추에이터의 스티어링으로 사용하기 때문에 주변 차량과의 충돌 뿐만 아니라 차로 유지와 가드레일과의 충돌 상황이 빈

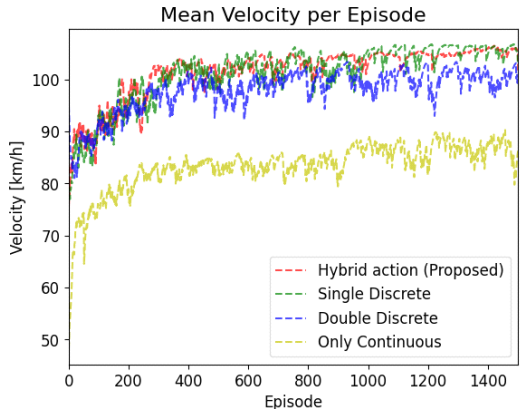

그림 11. 학습 평균 속도
Fig. 11. Mean velocity of train.

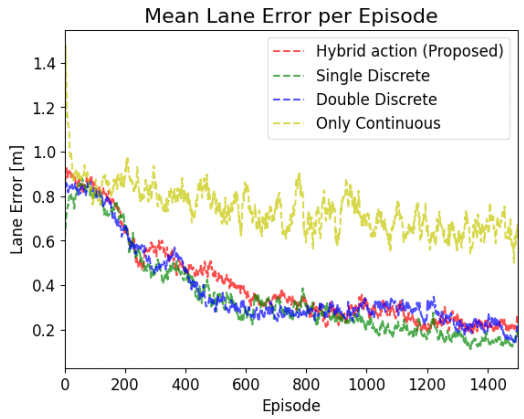

그림 12. 학습 평균 차로 오차
Fig. 12. Mean lane error of train.

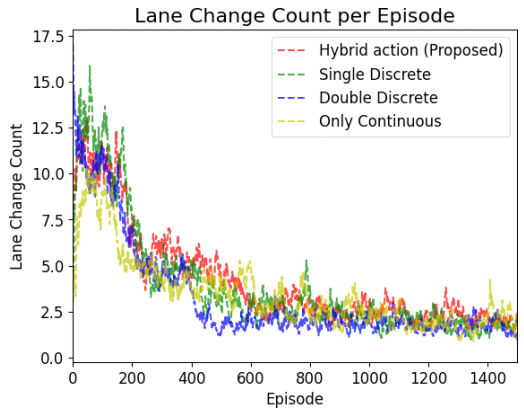

그림 13. 학습 차선 변경 수
Fig. 13. Lane change count of train.

번하게 일어난다. 따라서 본 논문에서 정의한 보상함수로 적용하기 어려운 것으로 확인된다. 또한, 50 에피소

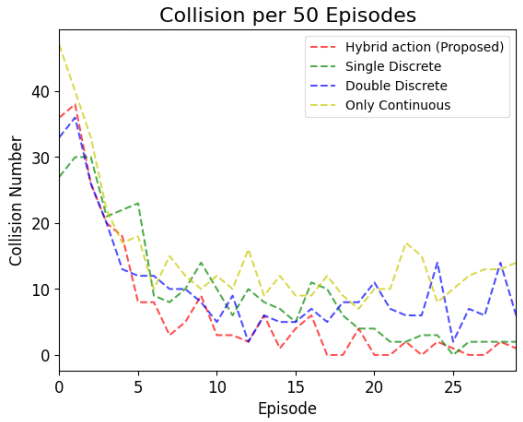

그림 14. 학습 50 에피소드 당 충돌 횟수
Fig. 14. Collision count per 50 episodes of train.

드마다 충돌 횟수 역시 다른 행동 유형에 비해 크게 나타나며, 학습 과정에서 single discrete와 hybrid action은 차이는 두드러지지 않는 것으로 나타난다.

4.3.3 Test 결과 평가

표 4의 결과를 보면 본 논문에서 제시하는 hybrid action이 충돌률이 1%, 평균 속도는 106.62km/h, 평균 차로 오차는 0.152m로 최적의 성능을 나타내고 있으며, 다음으로는 single discrete가 좋은 성능을 나타내고 있다. double discrete의 경우 충돌률이 상대적으로 크며, only continuous는 충돌률뿐 아니라 평균 속도와 차로 오차에서 모두 다른 행동 유형과 비교하면 성능이 많이 떨어지고 있다. 그림 15과 16은 각각 평균 속도와 평균 차로 오차에 대해서 테스트한 시나리오의 결과를 시각적으로 보여주고 있다. PPO는 하나의 샘플을 여러 번의 업데이트를 사용하므로 수렴 속도가 빠르지만, 하이퍼 파라미터에 민감하다. 실험을 통해 표 5와 같이 파라미터의 크기를 설정하였다. 보상함수는 $r_{collision}, r_{lc}, r_{wd}$

표 4. 테스트 비교 결과
Table 4. Test comparison results.

	Collision rate[%]	v_{mean} [km/h]	$\lvert d_{lane} \rvert$ [m]
hybrid action	1	106.62	0.152
single discrete	2	103.93	0.154
double discrete	6	101.61	0.199
only continuous	11	89.94	0.653

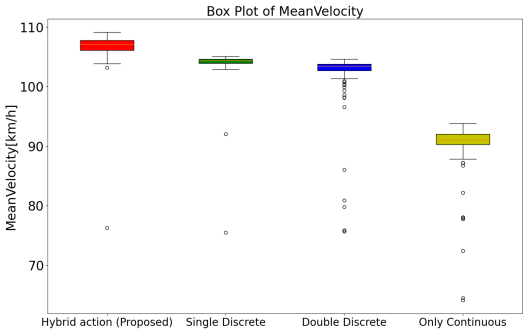

그림 15. 테스트 평균 속도 박스 플랏
Fig. 15. Box plot of mean velocity.

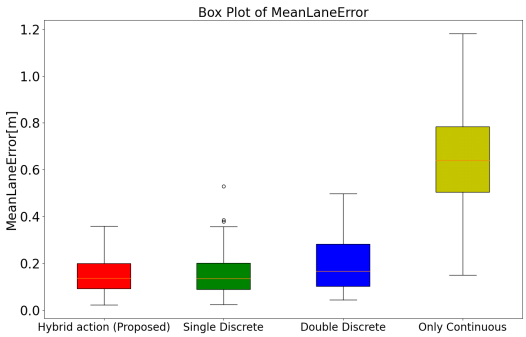

그림 16. 테스트 평균 차로 오차 박스 플랏
Fig. 16. Box plot of mean lane error.

표 5. 심층강화학습 파라미터
Table 5. Parameter of deep reinforcement learning.

Discount Factor γ	0.99
Experience Buffer size N_E	12,000
Batch size N_B	64
Maximum epoch N_e	1000,000
Learning rate l	0.003

와 같이 특정 이벤트에 의해 발생하는 보상과 $r_{speed}, r_{laneError}$와 같이 에이전트의 상태에 의해 연속적으로 발생하는 보상으로 구분될 수 있다. 이벤트에 의해 발생하는 보상에서 충돌은 에이전트의 주행에 있어서 필수적으로 막아야 하는 상황이며, 이와 별개로 r_{lc}와 r_{wd}는 크게 설정하였다[15]. 또한 r_{speed}는 유일한 양의 보상으로 이상적인 주행을 위한 가장 직접적인 보상이 되며, 누적 r_{speed}가 $r_{collision}$에 비해 너무 크게 할 경우, 에이전트는 제한된 속도 범위에서 속도를 높이는 것이 충돌보다 더 이상적인 결정이라고 판단한다. 반대로 너무 작게 할 경우, 충돌을 방지하기 위해 가속에 있어서

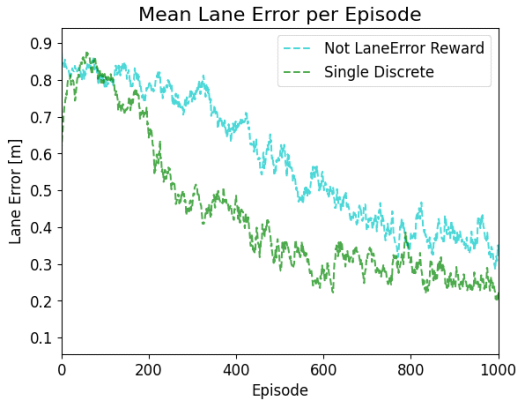

그림 17. 차로 오차 보상에 따른 평균 차로 오차 비교
Fig. 17. Comparison of mean lane error based on lane error reward.

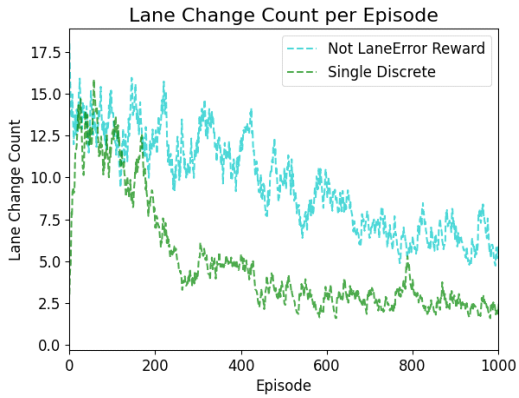

그림 18. 차로 오차 보상에 따른 차선 변경 수 비교
Fig. 18. Comparison of lane change count based on lane error reward.

소극적인 판단을 하게 된다. 따라서 실험을 통해 1500m를 에이전트가 주행하는 동안 충돌이 없는 상황에서 r_t가 $C_{collision}$의 약 4배가 되도록 C_{speed}를 설정하였다. $r_{laneError}$는 차로의 중심을 유지하도록 역할을 해주는 동시에 차선 변경을 제한시킬 수 있다. 따라서 $C_{laneError}$는 C_{lc}의 크기와 매 프레임마다 보상을 얻는 점을 고려하여 설정하였으며, 그림 17과 18에서 $r_{laneError}$의 유무에 따라 평균 차로 오차와 차선 변경의 수의 영향을 주는 것을 확인할 수 있다. 설정한 상수(C)는 표 6에 나타내었다.

표 6. 보상함수 상수
Table 6. Constants of reward function.

$C_{collision}$	50.0
C_{speed}	0.6
C_{lc}	2.0
$C_{laneError}$	0.04
C_{wd}	1.0

V. 결 론

본 논문에서는 심층 강화학습을 이용하여 3차선 고속도로에서 차량을 회피하면서 주행하기 위해 행동 공간을 4가지로 구분하여 최적의 방법을 제시하였다.

각 행동 유형에 따라 상태 공간을 다르게 정의하였으며, 결과를 비교하기 위해 학습 과정에서의 누적 보상과 손실뿐만 아니라 평균 속도, 평균 차로 오차, 차선 변경 횟수, 50 에피소드별 충돌 횟수를 평가 지표로 제시하여 비교하였다. 테스트에서는 충돌률, 평균 속도, 평균 차로 오차를 비교하여 hybrid action을 최적의 행동 유형으로 제안하였다. 그러나 최적의 행동 유형에서도 100번의 테스트 중 1번의 충돌이 발생하였으며, 이는 매우 위험한 결과이다. 강화학습은 결과를 설명하기 어렵다는 단점이 존재한다. 그렇기에 강화학습만을 적용한 판단이 어려우며, 따라서 규칙 기반을 섞어 안전한 주행이 가능한 모델을 개발할 계획이다.

References

[1] H. Lee, T. Kim, H. Kim, and S.-H. Hwang, “Reinforcement learning based autonomous emergency steering control in virtual environments,” *J. Drive and Control*, vol. 19, no. 4, pp. 110-116, Dec. 2022. (https://doi.org/10.7839/ksfc.2022.19.4.110)

[2] S. Malik, M. A. Khan, H. El-Sayed, J. Khan, and O. Ullah, “How do autonomous vehicles decide?” *Sensors*, vol. 23, no. 1, Dec. 2022. (https://doi.org/10.3390/s23010317)

[3] J. Lee and S.-J. Yoo, “Implementation of digital virtual environment model considering obstacles and traffic lights, and research on multi-lane autonomous driving based on deep reinforcement learning,” *J. KICS*, vol. 49, no. 6, pp. 862-873, Jun. 2024. (https://doi.org/10.7840/kics.2024.49.6.862)

[4] M. Zhang, K. Chen, and J. Zhu, “An efficient planning method based on deep reinforcement

learning with hybrid actions for autonomous driving on highway," *Int. J. Mach. Learn. and Cybernetics*, vol. 14, pp. 3483-3499, Jun. 2023.
(https://doi.org/10.1007/s13042-023-01845-2)

[5] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transport. System (TITS)*, vol. 23, no. 6, pp. 4909-4926, Jun. 2022.
(https://doi.org/10.1109/TITS.2021.3054625)

[6] X. Xu, L. Zuo, X. Li, L. Qian, J. Ren, and Z. Sun, "A reinforcement learning approach to autonomous decision making of intelligent vehicles on highways," *IEEE Trans. Syst., Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3884-3897, Oct. 2020.
(https://doi.org/10.1109/TSMC.2018.2870983)

[7] H. Deng, Y. Zhao, Q. Wang, and A.-T. Nguyen, "Deep reinforcement learning based decision-making strategy of autonomous vehicle in highway uncertain driving environments," *Automotive Innovation*, vol. 6, pp. 438-452, Aug. 2023.
(https://doi.org/10.1007/s42154-023-00231-6)

[8] M. Moghadam, A. Alizadeh, E. Tekin, and G. H. Elkaim, "A deep reinforcement learning approach for long-term short-term planning on Frenet frame," in *Proc. IEEE Int. Conf. Automation Science and Engineering (CASE)*, pp. 23-27, Lyon, France, Aug. 2021.
(https://doi.org/10.1109/CASE49439.2021.9551598)

[9] S. Tang, H. Shu, and Y. Tang, "Research on decision-making of lane-changing of automated vehicles in highway confluence area based on deep reinforcement learning," in *Proc. CAA Int. Conf. Vehicular Control and Intelligence (CVCI)*, pp. 29-31, Tianjin, China, Oct. 2021.
(https://doi.org/CVCI)

[10] P. Wolf, K. Kurzer, T. Wingert, F. Kuhnt, and J. M. Zöllner, "Adaptive behavior generation for autonomous driving using deep reinforcement learning with compact semantic states," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, pp. 26-30, Changshu, China, Jun. 2018.
(https://doi.org/10.1109/IVS.2018.8500427)

[11] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," *IEEE Trans. Intell. Vehicles (TIV)*, vol. 5, no. 2, pp. 294-305, Jun. 2019.
(https://doi.org/10.1109/TIV.2019.2955905)

[12] M. P. Ronecker and Y. Zhu, "Deep Q-network based decision making for autonomous driving," in *Proc. IEEE Int. Conf. Robotics and Automatation Sciences (ICRAS)*, pp. 154-160, Wuhan, China, Aug. 2019.
(https://doi.org/10.1109/ICRAS.2019.8808950)

[13] Y. Ye, X. Zhang, and J. Sun, "Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment," *Transport. Res. Part C*, vol. 107, pp. 155-170, Oct. 2019.
(https://doi.org/10.1016/j.trc.2019.08.011)

[14] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transport. System (TITS)*, vol. 23, no. 2, pp. 740-759, Feb. 2022.
(https:/doi.org/10.1109/TITS.2020.3024655)

[15] A. Alizadeh, M. Moghadam, Y. Bicer, N. K. Ure, U. Yavas, and C. Kurtulus, "Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment," in *Proc. IEEE Int. Conf. Intell. Transport. Systems (ITSC)*, pp. 27-30, Auckland, NZ, Oct. 2019.

[16] S. Nageshrao, E. Tseng, and D. Filev, "Autonomous highway driving using deep reinforcement learning," in *Proc. IEEE Int. Conf. Syst. Man and Cybernetics (SMC)*, pp. 2326-2331, Bari, Italy, Oct. 2019.
(https://doi.org/10.1109/SMC.2019.8914621)

[17] Z. Bai, W. Shangguan, B. Cai, and L. Chai, "Deep reinforcement learning based high-level

driving behavior decision-making model in heterogeneous traffic," in *Proc. Chin. Control Conf. (CCC)*, pp. 27-30, Guangzhou, China, Jul 2019. (https://doi.org/10.23919/ChiCC.2019.8866005)

[18] J. Wang, Q. Zhang, D. Zhao, and Y. Chen, "Lane change decision-making through deep reinforcement learning with rule-based constraints," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, pp. 1-6, Budapest, Hungary, Jul. 2019.

[19] S. Aradi, T. Becsi, and P. Gaspar, "Policy gradient based reinforcement learning approach for autonomous highway driving," in *Proc. IEEE Conf. Control Technology and Applications (CCTA)*, pp. 21-24, Copenhagen, Denmark, Aug. 2018. (https://doi.org/10.1109/CCTA.2018.8511514)

[20] C.-J. Hoel, K. Wolff, and L. Laine, "Automated speed and lane change decision making using deep reinforcement learning," in *Proc. IEEE Int. Conf. Intell. Transport. Systems (ITSC)*, pp. 2148-2155, Maui, HI, USA, Dec. 2018. (https://doi.org/10.1109/ITSC.2018.8569568)

[21] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 5385-5392, Paris, France, Aug. 2020. (https://doi.org/10.1109/ICRA40945.2020.9197132)

[22] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," *IEEE Trans. Neural Netw. and Learn. Syst. (TNNLS)*, vol. 35, no. 4, pp. 5064-5083, Apr. 2024. (https://doi.org/10.1109/TNNLS.2022.3207346)

[23] S. G. Park and D. H. Kim, "Autonomous flying of drone based on PPO reinforcement learning algorithm," *J. Inst. Control, Robotics and Syst.*, vol. 26, no. 11, pp. 955-963, Nov. 2020. (https://doi.org/10.5302/J.ICROS.2020.20.0125)

[24] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Syst. with Appl.*, vol. 231, Nov. 2023. (https://doi.org/https://doi.org/10.1016/j.eswa.2023.120495)

김 성 준 (Seongjun Kim)

2020년 3월~현재 : 아주대학교 전자공학과 학사과정

<관심분야> 자율주행, 무인시스템, 머신러닝

신 규 민 (Kyu-min Shin)

2019년 3월~현재 : 아주대학교 전자공학과 학사과정

<관심분야> 자율주행, 머신러닝, 제어공학, 로보틱스

전 준 서 (Jun-seo Jeon)

2019년 3월~현재 : 아주대학교 전자공학과 학사과정

<관심분야> 전자공학, 임베디드 시스템, 자율주행

방 지 윤 (Ji-yoon Bang)

2021년 3월~현재 : 아주대학교 전자공학과 학사과정

<관심분야> 전자공학, 제어공학, 로보틱스, 자율주행

김 준 영 (Junyoung Kim)

2023년 2월 : 한림대학교 소프트웨어학부 학사

2023년 3월~현재 : 아주대학교 AI융합네트워크학과 석사과정

<관심분야> 모빌리티, 자율주행, 저궤도 위성통신, 인공지능

정 소 이 (Soyi Jung)

2013년 2월 : 아주대학교 전자공학과 공학사

2015년 2월 : 아주대학교 전자공학과 공학석사

2021년 2월 : 아주대학교 전자공학과 공학박사

2021년 3월~2021년 8월 : 고려대학교 정보통신기술연구소 연구교수

2021년 3월~2022년 8월 : 한림대학교 소프트웨어학부 조교수

2022년 9월~현재 : 아주대학교 전자공학과 조교수

<관심분야> 모빌리티, 자율주행, 이동통신, 저궤도 위성통신, 인공지능