

# 자율주행차량의 주차를 위한 딥러닝 기반 주차경로계획 수립연구

## Parking Path Planning For Autonomous Vehicle Based on Deep Learning Model

김지환\* · 김주영\*\*

\* 주저자 : 국립한국교통대학교 교통정책학과 석사

\*\* 교신저자 : 국립한국교통대학교 교통정책학과 교수

Ji hwan Kim\* · Joo young Kim\*\*

\* Master. of Transportation Planning & Management., Korea National University of Transportation

\*\* Professor. of Transportation Planning & Management, Korea National University of Transportation

† Corresponding author : Joo young Kim, jykim@ut.ac.kr

Vol. 23 No.4(2024)  
August, 2024  
pp.110~126

pISSN 1738-0774  
eISSN 2384-1729  
<https://doi.org/10.12815/kits.2024.23.4.110>

Received 19 June 2024  
Revised 15 July 2024  
Accepted 1 August 2024

© 2024. The Korean Society of  
Intelligent Transport Systems. All  
rights reserved.

### 요약

자율주행차의 요소 중 하나인 경로계획(Path-planning)을 제안한다. 실제 주차장을 참고하여 수직주차와 수평주차로 주차장의 차로 너비, 주차 공간의 너비, 길이 등 주차장 구조와 주차 환경을 다양하게 설정한다. 출발점과 도착지점 등 각도와 환경을 다양하게 설정하여 경로데이터를 수집하고 수집한 데이터를 Deep Learning model에 넣어 학습시켜 자동주차경로계획 모델을 제안한다. 분석결과, 기 알고리즘(Hybrid A-star, Reeds-Shepp Curve)과 딥러닝 모델 모두 장애물에 충돌하지 않고 비슷한 경로를 생성하지만, 거리와 소모시간이 각각 0.59%, 0.61% 감소하여 효율적인 경로가 생성되었다. 또한, Switching point도 1.3개에서 1.2개로 감소하여 직진과 후진을 최대한으로 줄여 운전자의 피로를 줄일 수 있을거라 생각된다. 마지막으로 경로생성시간은 42.76% 감소하여 효율적이고 신속한 경로생성이 가능하여 향후 자율주행 중 자율주행차의 경로계획생성에 활용될 수 있으며, 차량작동에 따라 이동하는 주차로봇의 경로생성에도 활용될 수 있을 것으로 보인다.

핵심어 : Path-planning, Deep Learning, Reeds-Shepp Curve, Hybrid A-star, Matlab/Simulink

### ABSTRACT

Several studies have focused on developing the safest and most efficient path from the current location to the available parking area for vehicles entering a parking lot. In the present study, the parking lot structure and parking environment such as the lane width, width, and length of the parking space, were varied by referring to the actual parking lot with vertical and horizontal parking. An automatic parking path planning model was proposed by collecting path data by various setting angles and environments such as a starting point and an arrival point, by putting the collected data into a deep learning model. The existing algorithm(Hybrid A-star, Reeds-Shepp Curve) and the deep learning model generate similar paths without colliding with obstacles. The distance and the consumption time were reduced by 0.59% and 0.61%, respectively, resulting in more efficient paths. The switching point could be decreased from 1.3 to 1.2 to reduce driver fatigue by maximizing straight and backward movement. Finally, the path generation time is reduced by 42.76%, enabling efficient and rapid path generation, which can be used to create a path plan for autonomous parking during autonomous driving in the future, and it is expected to be used to create a path for parking robots that move according to vehicle construction.

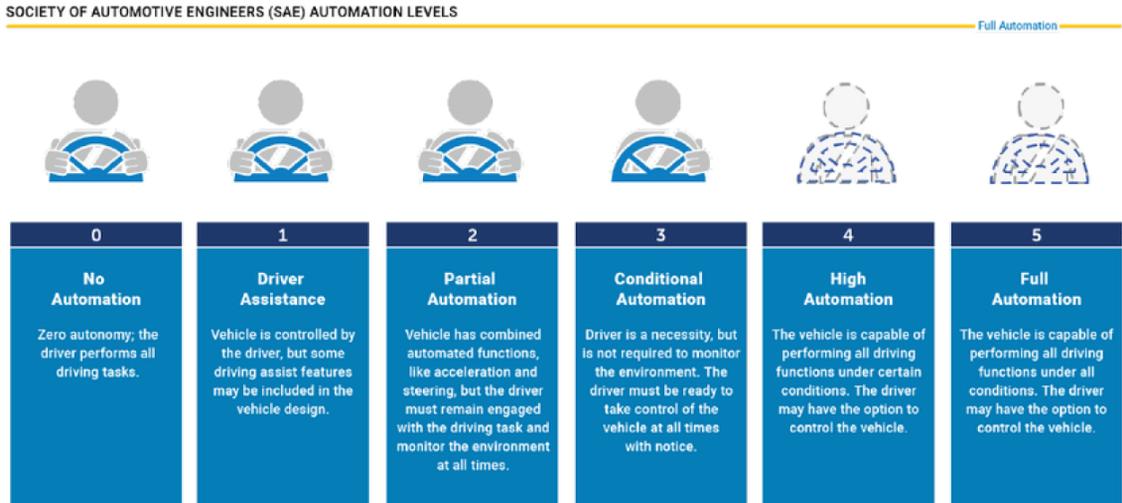
Key words : Path-planning, Deep learning, Reeds-shepp curve, Hybrid A-star, Matlab/Simulink

# I. 서론

## 1. 개요

주차문제는 1990년대 중반부터 부각되어 40여년이 지난 지금도 여전히 해결되지 않고 있고, 자동차 수는 계속 증가하여 현재진행중인 심각한 문제이다. 도심의 차량밀집지역의 주차수요와 주차공급이 부족하기 때문이다. 도심의 차량밀집지역의 주차 수요는 시간대와 장소에 달라지기 때문에 주차수요에 맞는 주차장 확보율이 중요한 것이 아니다. 대부분의 운전자들이 주차를 할 자리가 없어 어쩔 수 없이 불법주정차를 하게 된다. 불법 주차로 인해 교통혼잡도 늘어나는 등 국내 교통문제의 많은 부분을 차지한다. 이에 주차 수요와 공급에 맞는 유동성이 필요하다. 현재 스마트 주차 관제시스템을 도입하려는 추세가 들어난다. 이러한 추세는 스마트 주차 관제시스템 국내의 시장 성장세를 보면 알 수 있다. 국내의 스마트 주차관제 시스템 시장은 2019년 기준 시장규모가 약 6조 3천억원의 규모를 보이며 연평균 약 8.7%의 증가율을 보인다. 주차로봇을 통해 주차 이용자와 공급자의 효율을 증가시키고 효율적인 주차면 관리를 통해 주차문제를 넘어 교통문제까지 해결하려는 움직임이다. 또한, 국내외 주차로봇 시장이 활성화 되고, 이에 자율주행에 관한 연구도 계속해서 진행되고 있다. 자율주행 자동차란 운전자의 개입 없이 주변 환경을 인식하고, 주행 상황을 판단하여 차량을 제어함으로써 목적지까지 주행하는 자동차를 말한다. 자율주행 자동차가 교통사고를 줄이고, 연료를 절감하고, 교통효율성을 높이면서 편의를 증대시킬 수 있는 미래 모빌리티로 각광 받으며 상용화를 목표로 연구와 개발에 박차를 가하고 있다. 미국자동차기술회 SAE(Society of Automotive Engineers)의 자율주행 시스템 6단계<sup>1)</sup>에 따르면 현재는 2단계인 부분자동화와 3단계인 조건부자동화에 속한다.

최근 10여년간 교통안전문제들로 인한 비효율지표의 심각성을 인식한 선진국들은 교통사고를 획기적으로 예방하기 위한 협력형 지능형교통시스템(cooperative ITS) 구상에 집중하고 있다.



<Fig. 1> Automation Levels

1) Levels of Driving Automation, SAE(Society of Automotive Engineers) (2021)

우리가 상상하는 운전자 개입없이, 운전면허가 없는 사람이 탑승해도 목적지에 도착하여 주차까지 자율화 되는 단계는 최소 4단계인 고도자동화 또는 최종단계 5단계 완전자동화가 되어야 가능하다. 즉, 운전자의 개입이 없는 완전자동화 단계의 자율주행을 위해서는 자동주차가 필수요소이다. 주차장 환경은 일반 구조화된 도로에 비해 도로의 너비가 좁으며, 다양한 장애물이 복잡하게 존재하기 때문에 이에 대한 주차경로계획이 필요하다(Kim et al., 2021). 이처럼 도로위에서의 자율주행도 중요하지만 자율주차도 그만큼 중요하다. 자율주차는 주차 가능 영역을 탐색하고, 차량의 위치에서 장애물을 피해 주차 가능 영역까지 가장 안전하고 효율적인 경로로 주차하는 것이다. 주차는 운전을 시작한지 얼마 안된 초보 운전자에게 어려움을 느끼게 하는 요소이다. 국내 손해보험 회사에서 제공한 자료를 보면 자동차보험 사고의 약 30%가 주차사고로 분류되고 있고(Kim, 2022), 그 중 후진주차사고의 비율은 약 53.8%를 차지한다(Lee, 2016).

<Table 1> Ratio of parking accidents among traffic accidents

Total number of cases	Number of parking accident	Parking accident rate
8,610,616	2,597,023	30.2%

또한 교통안전정보관리시스템에서 통계한 운전경력별 교통사고를 살펴보면 1년 미만의 운전경력을 가진 운전자의 교통사고가 가장 많고 경력이 늘수록 교통사고도 비례적으로 줄어드는 것을 볼 수 있다.

<Table 2> Traffic accidents by driving experience

Driving experience	Number of occurrences				
	Total	Dead	Fatal	Slight	Minor
Less than 1 year	5,490	75	1,218	3,891	306
Less than 2 year	4,611	55	1,038	3,277	241
Less than 3 year	4,471	50	978	3,215	228
Less than 4 year	4,348	53	1,000	3,099	196
Less than 5 year	4,351	62	964	3,137	188

이처럼 주차를 어려워하는 초보 운전자들이 많고, 장애 주차공간 다양성에 대한 대비와 미래 새로운 모빌리티 시대의 주차 인프라를 개선하기 위해 자율주차는 필수요소로 생각된다. 본 연구의 목표는 자율주차의 요소 중 하나인 경로계획(Path-planning)이다. 주차장에 들어선 차량이 현재 위치에서 주차 가능 영역까지 가장 안전하고 효율적인 경로를 생성하는 것을 목표로 한다. 수직주차와 수평주차로 주차장의 차로 너비, 주차공간의 너비, 길이 등 주차장 구조와 주차 환경을 다양하게 설정한다. 이러한 다양한 환경에서 RRT star (Rapidly-exploring Random Tree star), Reeds-Shepp Curve, Hybrid A-star 등의 알고리즘을 사용하여 경로를 생성한다. 출발점과 도착지점 등 각도와 환경을 다양하게 설정하여 경로데이터를 수집하고 이렇게 수집한 데이터를 Deep Learning model에 넣어 학습시켜 자동주차경로계획 모델을 제안한다.

## II. 선행연구

### 1. RRT star 알고리즘

RRT\*알고리즘은 RRT 알고리즘을 개선하기 위해 제안된 알고리즘 중 하나로 생성된 경로의 질을 높이는 방법을 제안한다. RRT\*알고리즘은 식 (1)과 같이 RRT 알고리즘에 없는 특징인 접근적 최적성으로 인해 생성된 노드 수가 무한에 가까워질수록 생성되는 경로는 최적 경로를 향해 근사하는 성질을 갖는다. RRT\*알고리즘을 사용하면 시작점, 목표점으로 하는 하나의 경로를 생성할 수 있으며 알고리즘의 진행에 따라 생성된 노드들은 트리 구조를 갖는다.

$$k = \gamma \left( \frac{\log(n)}{n} \right)^{1/d} \dots\dots\dots (1)$$

```

Algorithm 1: RRT*
1  T.Initialize(x_init);
2  for i = 0 to i = N do
3    x_rand ← Random_Sample;
4    x_nearest ← Nearest(T, x_rand);
5    x_new ← Steer(x_nearest, x_rand);
6    if Obstacle_Free(x_new) then
7      x_near ← Near(T, x_new, |k|);
8      x_min, e_min-new ← Choose_Parent(x_near, x_new);
9      T.Insert(x_new, e_min-new);
10   foreach x_near ∈ X_near do
11     if c(x_near) > c(x_new) + c(e_new-near) then
12       T.Reconnect(x_new, x_near, e_new-near);
13 return T
    
```

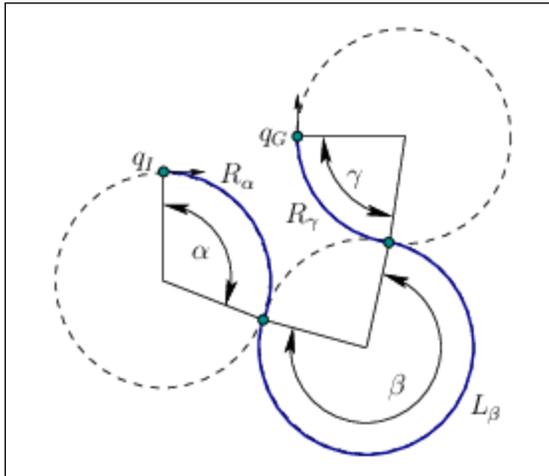
<Fig. 2> RRT\* algorithm flow

RRT\*알고리즘의 과정은 <Fig. 2>와 같다. <Fig. 2>의 7번째 줄은 최적 부모 노드 선택과 재연결 과정이 수행되는 특정 범위 내의 노드들을 구하는 과정이다. 특정 범위는  $x_{(\neq w)}$ 를 중심, 상수  $k$ 를 반지름으로 하는 영역이고  $k$ 값은 식(1)에 의해 결정되며  $r$ 는 재연결 상수,  $n$ 은 노드수,  $d$ 는 차원 수를 의미한다. 이는 알고리즘의 반복으로 노드의 수가 많아질수록 재연결 관정의 연산 양이 증가하므로 반지름을 줄여  $x_{(\eta \epsilon \alpha \gamma)}$  내의 노드를 일정 수준으로 유지하여 재연결 과정에 필요한 연산의 양을 줄이기 위함이다(Lee, 2022). <Fig. 2>에서 10~12번째 줄은 알고리즘이 접근적 최적성을 갖도록 하는 재연결 과정이다. 새로 생성된 노드인  $x_{(\neq w)}$ 의 인근 노드인  $x_{(\eta \epsilon \alpha \gamma)}$ 에 대한 비용 함수의 값이  $x_{(\neq w)}$ 를 거쳐 갈 때 더 줄어드는 경우 간선의 연결을 변경하는 것이다. RRT\* 알고리즘의 최적 경로로의 수렴은 임의표본추출(Random sampling)에 의해 생성된 노드들의 위치에 의해 성능이 좌우되므로 경로의 최적화를 위해서는 많은 노드의 생성이 필요하다. 이는 최초의 경로를 찾은 후 생성되는 노드들에 대해 재연결 과정을 진행함으로써 더 효율적인 경로의 생성을 기대하기 때문이며 이러한 방법을 통해 경로의 최적화가 수행될 수 있음이 알려져 있다.(Lee, 2022) 무작위로 표본추출된 상태로 상태공간의 탐색트리를 생성을 한다. 시작점에서 트리 안에 있는 가장 가까운 점, 노드를 하나 선택하여 벡터를 생성한다. 무작위로 트리안에서 점과 노드를 선택하여 그 안에서 가장 가까운 쪽으로 붙여 트리를 확장하게 된다. 이 과정을 반복하면 목적지까지 가는 경로를 찾아내게 된다. 인위적인 트리의 확장이라는 특징 때문에 탐색영역이 넓은 고차원의 공간을 효율적으로 탐색할 수 있다. 목적지점에 탐색 트리가 도달하

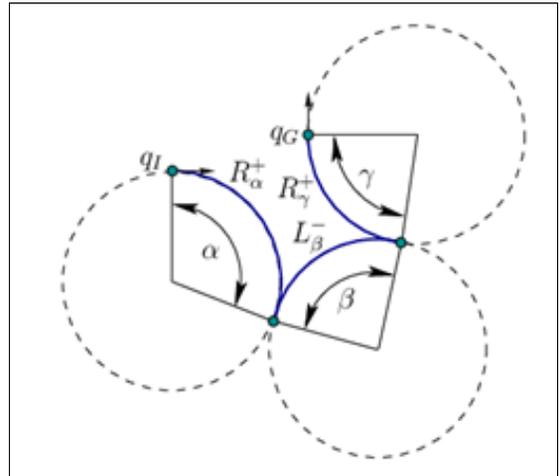
는 그 순간 검색과정이 종료되지 않고 계속 진행하여 좀 더 효율적인 경로가 만약 있다면 기존 경로를 변경할 가능성이 높다.

### 2. Reeds-Shepp Curve 알고리즘

Reeds-Shepp Curve는 자동차의 최단경로를 고려하는 알고리즘이다. Dubins Curve에서 파생되어 유일한 차이점은 반대 방향으로의 이동을 허용한다는 것이다.



<Fig. 3> Dubins Curve



<Fig. 4> Reed-Shepp Curve

Dubins Curve 알고리즘은 차량의 최소 회전반경과 직선만을 이용하여 출발점에서 도착점까지 가장 짧은 경로를 탐색하는 방법이다. <Fig. 5>에서 볼 수 있듯이 Reeds-Shepp Curve 알고리즘은 같은 방식으로 경로를 탐색하지만, 추가로 전진뿐만 아니라 후진으로도 경로를 탐색할 수 있으며, 이는 출발점에서 도착점까지 Dubins Curve 알고리즘보다 더 짧은 경로를 생성할 수 있다는 장점이 있다(Kim, 2022).

Base word	Sequences of motion primitives
$C C C$	$(L^+R^-L^+)(L^-R^+L^-)(R^+L^-R^+)(R^-L^+R^-)$
$CC C$	$(L^+R^-L^-)(L^-R^+L^+)(R^+L^+R^-)(R^-L^-R^+)$
$C CC$	$(L^+R^-L^-)(L^-R^+L^+)(R^+L^-R^-)(R^-L^+R^+)$
$CSC$	$(L^+S^+L^+)(L^-S^-L^-)(R^+S^-R^+)(R^-S^+R^-)$
$C C C C$	$(L^+S^+R^+)(L^-S^-R^-)(R^+S^+L^+)(R^-S^-L^-)$
$CC_3 C_3C$	$(L^+R_1^+L_2^+R^+)(L^-R_1^-L_2^-R^-)(R^+L_2^+R_1^+L^+)(R^-L_2^-R_1^-L^-)$
$C C_3C_3 C$	$(L^+R_1^+L_2^+R^+)(L^-R_1^-L_2^-R^-)(R^+L_2^+R_1^+L^+)(R^-L_2^-R_1^-L^-)$
$C C_{\pi/2}SC$	$(L^+R_{\pi/2}^+S^-R^+)(L^-R_{\pi/2}^-S^+R^-)(R^+L_{\pi/2}^+S^-L^+)(R^-L_{\pi/2}^-S^+L^-)$ $(L^+R_{\pi/2}^+S^-L^-)(L^-R_{\pi/2}^-S^+L^+)(R^+L_{\pi/2}^+S^-R^-)(R^-L_{\pi/2}^-S^+R^+)$
$CSC_{\pi/2} C$	$(L^+S^+L_{\pi/2}^+R^+)(L^-S^-L_{\pi/2}^-R^-)(R^+S^+R_{\pi/2}^+L^+)(R^-S^-R_{\pi/2}^-L^-)$ $(R^+S^+L_{\pi/2}^+R^-)(R^-S^-L_{\pi/2}^-R^+)(L^+S^+R_{\pi/2}^+L^-)(L^-S^-R_{\pi/2}^-L^+)$
$C C_{\pi/2}SC_{\pi/2} C$	$(L^+R_{\pi/2}^+S^-L_{\pi/2}^+R^+)(L^-R_{\pi/2}^-S^+L_{\pi/2}^-R^-)$ $(R^+L_{\pi/2}^+S^-R_{\pi/2}^+L^+)(R^-L_{\pi/2}^-S^+R_{\pi/2}^-L^-)$

<Fig. 5> Example of Reeds-Shepp Curve

### 3. A-star 알고리즘

A-star 알고리즘은 대표적인 휴리스틱 최단 경로 탐색 알고리즘이다. A\* 알고리즘은 우선 탐색으로 휴리

스틱 추정값을 사용하는 평가 함수에 의해 선정된 값 중 최솟값을 가진 노드를 선택 최단 경로를 산출한다 (유영근·박용진, 2013). 탐색 노드 n에서의 평가 함수를  $f(n)$ 이라고 했을 때 평가 함수  $f(n)$ 는 식(2)과 같다.

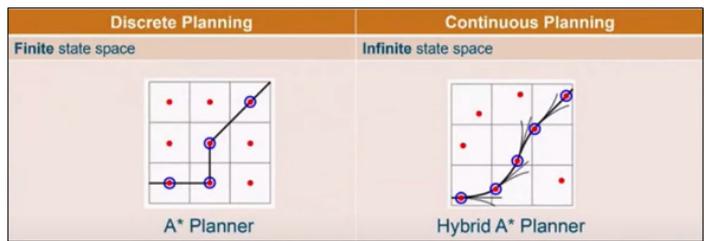
$$f(n) = g(n) + h(n) \dots\dots\dots (1)$$

여기서,  $g(n)$  : 시작 노드로부터 탐색 노드 n까지의 거리  
 $h(n)$  : 탐색 노드 n으로부터 목표 노드까지의 거리

복잡한 경로를 탐색할 경우, A\* 알고리즘은 최적의 경로를 탐색하는 데 실패할 확률이 높으며 경로를 찾는 것을 보장하지 못한다는 문제점이 있다(Lee, 2023). 그렇지만 다른 경로와 비교하지 않고 목적 노드를 찾아가기 때문에 시간이 짧게 걸린다는 장점이 있고, 탐색이 잘못되었을 경우 뒤로 돌아와서 다시 다른 방향의 경로를 탐색하는 특징을 가지고 있는 목표 지향적인 알고리즘이다. 현재 위치에서 주변에 있는 미방문 (unvisited) 노드들을 모두 고려 대상 노드로 설정을 해서 시작점에서 현재 위치를 지나서 도착노드까지의 거리를 계산을 한다. 이 거리를 비용(cost)로 지정을 한다.  $g(n)$ 과  $h(n)$ 을 더한 비용이 새로 계산된 비용보다 낮다면 경로를 변경해서 설정하게 된다. 더 이상의 미방문 노드를 가지고 있지 않다면 해당 노드는 방문 (visited)노드로 설정하고 더 이상 검색을 수행하지 않고 검색 작업을 마친다. 이후, 미방문 노드 목록에 있는 노드들 중에서 가장 낮은 비용을 가진 노드를 현재 검색할 노드로 선정을 하여 다시 검색작업을 수행한다. 이 과정을 반복하면 가장 낮은 비용을 가진 방향으로 지속적으로 검색을 수행하다 장애물을 만나면 회피한다. 장애물을 회피하면 다시 가장 가까운 경로로 검색을 수행하여 경로를 찾는다.

#### 4. Hybrid A-star 알고리즘

A-star 알고리즘은 차량의 움직임에 적절하지 않다. A-star 알고리즘을 통해 얻은 경로는 자동차의 연속성을 반영하지 못하기 때문에 Hybrid A-star 알고리즘을 사용해야한다.

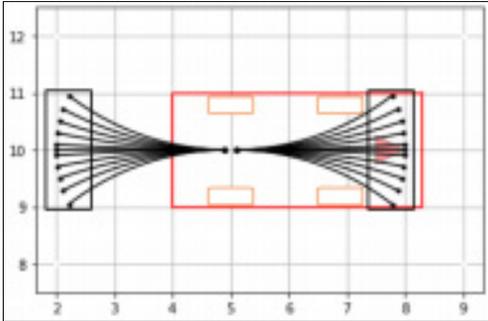


<Fig. 6> Difference between A\* and Hybrid A\*

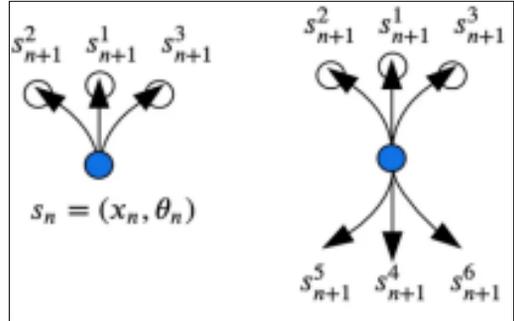
자동차의 경우 경로 생성의 중심은 차량의 뒷바퀴 중심으로 설정하는 것이 좋다. 직진인 가운데를 중심으로 왼쪽과 오른쪽, 즉, 좌회전과 우회전은 조향각을 설정하여 다양한 방향으로 움직일 수 있다. 하지만, 조향각을 너무 많이 늘리면 탐색해야하는 노드 수가 증가하므로 경로생성시간이 오래 걸리는 문제가 발생해 적당한 조향각으로 설정해야한다.

<Fig. 8>의 경우 역방향 동작을 갖도록 확장될 수 있으므로 6가지 확장이 가능하다. 이 경우 Reeds-Shepp Curve를 따르는 최단 경로로 임의의 두 상태를 연결할 수 있다. 현재위치에서 차량이 이동할 수 있는 모든

노드의 비용을 계산하고 그중 가장 낮은 비용을 가진 노드를 선택한다. 선택된 노드에서 다시 차량의 조향각을 고려하여 다음 노드들을 탐색하고 확장해나간다.



<Fig. 7> Creation route based on rear wheel

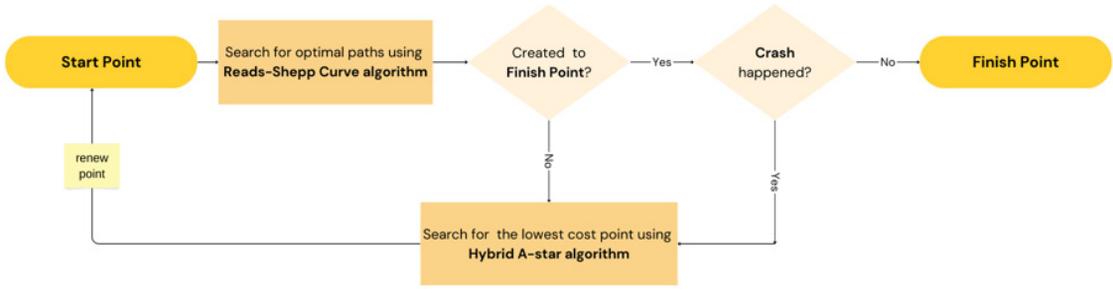


<Fig. 8> Expansion route based on rear wheel

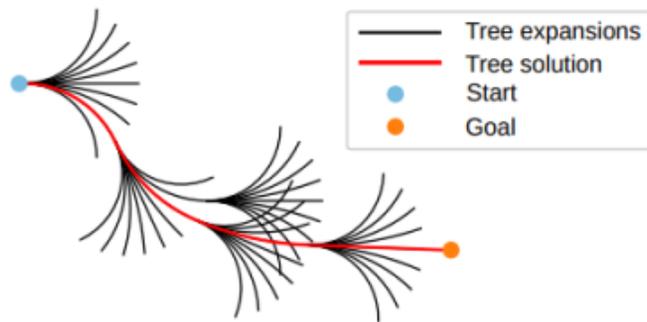
### III. 본 론

#### 1. Path Planning 알고리즘

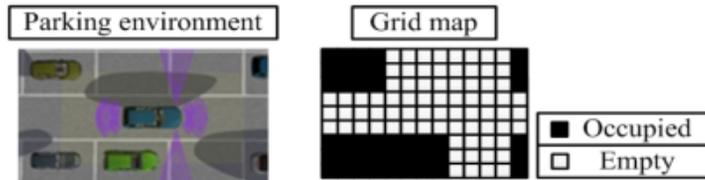
앞서 살펴본 주차경로계획에 사용되는 알고리즘은 장점과 단점이 존재한다. RRT 알고리즘의 경우 확률적인 경로 생성이라는 특성을 가지고 있어 다양한 형태의 지형에 대응하고 장애물 회피에도 유리하여 다양한 환경에 적응이 용이하다. 하지만, 확률적 경로 생성이라는 말 자체처럼 최적의 경로를 보장하지는 않는다. 즉, 효율성과 최적성이 떨어질 수 있다. 또한, 다양한 문제나 요지에 따라 튜닝이 필요하여 구현 또한 쉽지 않고, 준용하지 않는 추세이다. 따라서 본 연구에서는 RRT 알고리즘을 제외한 Hybrid A-star 알고리즘과 Reeds-Shepp Curve 알고리즘을 사용한다. Hybrid A-star 알고리즘과 Reeds-Shepp Curve 알고리즘을 혼합하여 단점을 보완할 수 있다. Hybrid A-star 알고리즘은 시작점부터 도착점까지 모든 경로를 검색하여 최적 경로를 찾기 때문에 계산량이 많아 맵의 크기에서는 경로 연산 시간이 길어질 수 있다. Reeds-Shepp Curve 알고리즘은 차량의 운동 모델을 고려하여 부드러운 경로를 생성하는데 유리하지만 이로 인해 복잡한 환경에서는 충돌 회피와 장애물 회피에 제한이 있다. 이에 Reeds-Shepp Curve 알고리즘으로 시작점부터 도착점까지 경로 탐색을 하다가 장애물 및 충돌 회피 시에만 Hybrid A-star 알고리즘을 이용하면 빠르고 안전한 경로계획이 가능하다. 따라서 본 연구에서는 먼저 Reeds-Shepp Curve 알고리즘으로만 경로를 검색한다. 장애물 여부와 충돌 여부를 확인하고 충돌하지 않았다면 경로검색을 마친다. 만약 Reeds-Shepp Curve 알고리즘으로만 경로를 검색하다 장애물이 있거나 장애물과 충돌 및 경로를 검색하지 못하였을 때에는 Hybrid A-star 알고리즘을 이용하여 경로를 탐색한다. 검색한 노드 중에서 비용이 가장 낮은 노드를 선택하고 그 검색된 위치를 출발점으로 설정하고 Reeds-Shepp Curve 알고리즘을 이용하여 도착점까지 경로를 탐색한다. 그러나 또 다시 경로 검색 중 장애물과의 충돌이나 경로검색 자체가 안 된다면 Hybrid A-star 알고리즘을 이용한다. 시작점에서 도착점까지 Reeds-Shepp Curve 알고리즘을 이용하지만 경로가 탐색이 원활하지 않은 경우 Hybrid A-star 알고리즘을 사용하여 도착점에 도착할 때까지 반복 사용한다.



<Fig. 9> Path planning Flow Chart



<Fig. 10> Node proliferation of Hybrid A-star algorithm



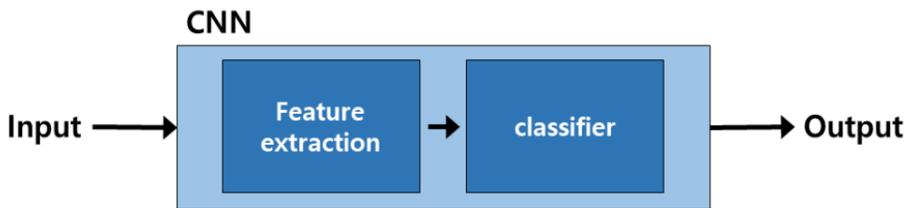
<Fig. 11> Grid map

최적의 경로를 결정하는 것은 많은 노드 중에 최적의 노드를 선택하는 것, 즉, 비용이 가장 적은 노드를 선택하는 것이다. 따라서 노드에 적용되는 비용함수가 중요하다. 최적의 경로란 가장 움직임을 덜 쓰고, 가장 거리가 가깝게 가는 것을 뜻한다. 따라서, 거리가 추가됨에 따라 비용이 증가하도록 설정하였다. 이는 후진의 경우에도 비용이 추가되도록 하였다. 조향각에 변화가 없다면 비용이 추가로 들지 않지만, 조향각이 변경된다면 비용이 추가되도록 설정하였다. 또한, 반복 움직임을 방지하기 위해 방문했던 노드를 재방문 시 비용이 추가되도록 설정하였다.

두 알고리즘을 혼합하여 사용하여도 경로를 찾지 못하는 경우가 발생한다. 일반적인 주차환경에서는 경로가 잘 생성되지만 장애물이 있는 제한적인 주차환경에서는 경로가 생성되지 않는 문제가 발생하였다. 이는 장애물을 회피하기 위한 섬세한 경로전환을 할 수 없는 문제가 있기 때문이다. 이에, 본 연구에서 장애물인 차량의 너비, 주차장의 너비, 주차장 차량의 너비 등을 고려하여 Grid resolution을 0.1m로 설정하였다. Grid

resolution은 그리드, 즉, 격자를 형성하는데 사용되는 격자 셀의 크기를 나타냅니다. Grid resolution이 낮을수록 격자가 촘촘하게 설정되며, 검색 노드 수가 증가하여 경로 생성시간이 오래 걸리지만 좀 더 최적화된 경로가 생성된다. 기존 Grid resolution이 2m일 때보다 Grid resolution이 0.5m일때 탐색한 노드의 수는 약 4배 가까이 증가하는 것을 확인하였다. 하지만 Grid resolution을 0.5m로 설정하면 같은 출발점에서 도착점까지 이동 거리가 기존 대비 12.3% 줄어든 11.75m가 나왔다. 이는 좀 더 최적화된 경로가 생성 된 것을 확인할 수 있다 (Kim, 2022).

## 2. Deep Learning 개요

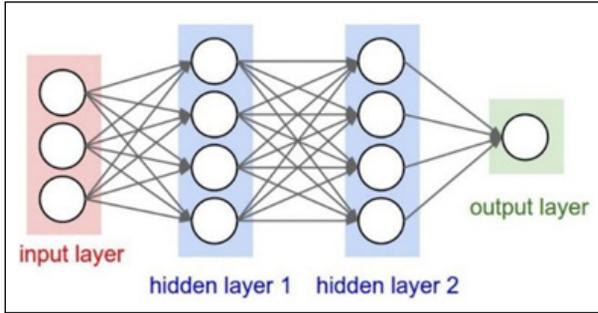


<Fig. 12> CNN progress

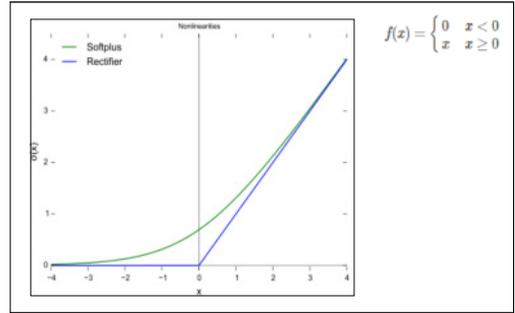
합성곱 신경망(CNN, Convolution Neural Network)은 <Fig. 12>와 같이 특징 추출(Feature extraction)과 분류기(Classifier)로 구성된다. 특징 추출은 Convolution layer와 Pooling(Sub-sampling) layer로 구성되어 있으며, 입력데이터(input)의 고유한 특징(invariance)을 찾는 단계이다. 이는 입력데이터를 필터가 순회하며 합성곱(convolution)을 계산하고, 그 결과를 이용하여 Feature map을 만드는 것이다. Convolution layer는 Input data에 필터를 적용한 뒤 활성화 함수를 반영하는 요소이며, Pooling(Sub-sampling) layer는 spatial 차원의 Down-Sampling을 수행한다. 분류기는 Fully connected layer로 구성되어 있으며, 찾아진 특징들을 가지고 class를 고르는 단계이다. Fully connected layer는 이전 계층의 모든 노드가 다음 계층의 모든 노드에 연결된 계층이다. 이는 특징 추출의 결과를 취하여 입력데이터를 정의된 라벨로 분류하는 데 사용한다(Cho, 2020). 본 연구에서는 이미지 라벨링을 수행하기 위해 YOLO를 사용하였으며, 주차하는 차량이 도착지점에 위치하는지 위치하지 않는지를 기준으로 Class를 정의하고 라벨링하였다. 후술하겠지만, 주차장 맵의 크기를 동일하게 설정하여 Bounding Box 영역을 설정하기 용이하였다.

본 연구에서는 Deep Learning을 이용하여 주차경로검색 방법을 제안한다. Deep Learning 구조는 Fully Connected Layer를 사용하였다. Fully Connected Layer(이하 FCL)은 한 Layer의 모든 뉴런이 그 다음 Layer의 모든 뉴런과 연결된 상태를 말한다. 1차원 배열의 형태로 평탄화된 행렬을 통해 이미지를 분류하는데 사용되는 계층이다.

활성화 함수(Activation Function)은 특정 조건을 만족하게 되면 활성화되었다는 신호를 다음 노드로 보내 활성화하고, 만약 조건을 만족하지 못했으면 해당 노드를 비활성화하는 함수이다. 본 딥러닝 모델에서는 ReLU(Rectified Linear Unit)함수를 사용하였다. 입력값이 음수면 0으로 비활성화 하고, 입력값이 양수면 해당 값을 출력하는 함수이다. 이 함수는 unit-step function과 동일하다. 따라서 sigmoid 활성화 함수에 비해 소실되는 기울기가 적어 학습에 용이하다. 또한, 비선형성을 이용해서 사전훈련 없이도 신경망 훈련을 가능하게 하며, 더 복잡한 경우에도 빠르고 효과적인 훈련을 가능하게 한다.

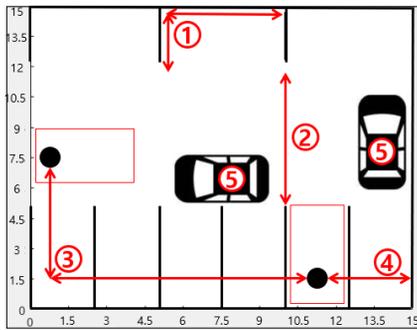


<Fig. 13> Fully Connected Layer

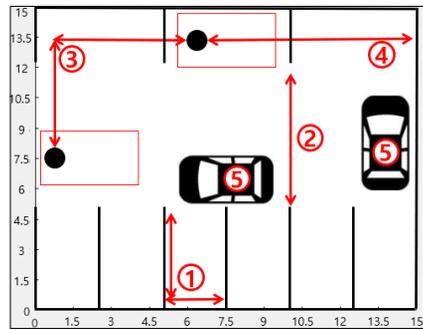


<Fig. 14> Rectified Linear Unit

### 3. Deep Learning Input



<Fig. 15> Example of Deep learning Input (Vertical parking)



<Fig. 16> Example of Deep learning Input (Horizontal parking)

<Fig. 15>과 <Fig. 16>는 딥러닝 input 값을 나타낸다. <Fig. 15>과 <Fig. 16>의 1번은 주차 공간의 크기이다. 우리나라의 주차공간의 크기는 다양하지만 주차장법 시행규칙에 따르면 다음과 같다.

<Table 3> Parking lot regulation

Classification	Before the revision		After the revision	
	Width	Length	Width	Length
Light type	2.0m	3.6m	2.0m	3.6m
General type	2.3m	5.0m	2.5m	5.0m
Expandable type	2.5m	5.1m	2.6m	5.2m
Handicapped only	3.3m	5.0m	3.3m	5.0m
Two-wheeled vehicle only	1.0m	2.3m	1.0m	2.3m

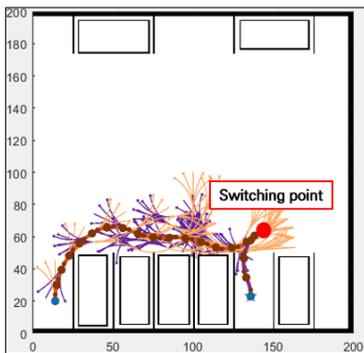
2019년 3월 1일부터 개정안으로 시행되었지만, 변경된 주차단위 구획은 새로이 신축되거나 설치되는 시설물에 한해 적용된다. 또한, 개정 후 변경된 점은 일반형의 너비와 확장형의 너비, 길이로 나머지는 개정전과 동일하다. 본 연구에서는 주차공간을 확장형을 제외한 개정안의 최소 기준에 따라 너비 2.5m, 길이 5m로 설

정하였다. 2번은 차로의 너비이다. 차로의 너비는 최소 6m 이상 되도록 정해져있어 본 연구에서는 6m 이상으로 다양하게 설정하였다. 최종 도착점이 어디든 시작점과의 거리와 각도가 같다면 경로도 같다. 즉, 각 시작점, 도착점의 위치가 중요한 것이 아니라 시작점과 도착점 사이의 거리와 각도가 경로를 결정하기 때문에 3번은 시작점과 도착점사이의 x축 차이와 y축 차이, 각도로 설정하였다. 4번은 도착점과 벽까지의 거리이다. 도착점에 주차를 할때 벽과 너무 가까우면 충돌이 일어나는지 구별하여 경로를 다시 생성하기 위해 필요하다. 마지막으로 5번은 주차장 내 장애물이다. 장애물은 주차된 차량과 보행자로 나뉜다. 주변 장애물들의 존재 여부가 경로에 큰 영향을 주기 때문에 장애물을 피해갈수 있도록 장애물의 위치좌표와 크기 등을 설정하였다. 여기서 장애물의 위치좌표는 차량의 경우 차량의 뒷바퀴 중심축을 기준으로 하여 장애물이 차지하고 있는 grid를 계산하였고, 보행자의 경우 보행자 중심을 기준으로 grid를 계산하였다. 또한 주차되어있는 차량과 주차하는 차량은 중형 혹은 대형으로 설정하여 길이 4.8m, 너비 1.8m로 설정하였고, 보행자의 경우 보행 여유공간까지 포함하여 원지름 2m인 원으로 설정하였다. 우리나라 차량 분류기준은 다음과 같다.

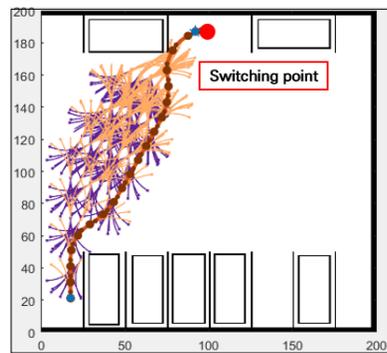
<Table 4> Vehicle classification criteria

Classification		Light	Small	Medium	Large
Displacement		~ 1,000cc	1,000 ~ 1,600cc	1,600 ~ 2,000cc	2,000cc ~
Car Body Size	Length	~ 3.6m	~ 4.7m	If any of the length, width, or height is exceeded	When all lengths, widths, and heights are exceeded
	Width	~ 1.6m	~ 1.7m		
	Height	~ 2.0m	~ 2.0m		

#### 4. Deep Learning Output



<Fig. 17> Example of Deep learning Output (Vertical parking)



<Fig. 18> Example of Deep learning Output (Horizontal parking)

<Fig. 17>와 <Fig. 18>은 딥러닝 Output을 나타낸다. 수직주차, 수평주차 모두에서 나오는 Switching point는 전진과 후진이 바뀌는 지점을 말한다. 시작점과 도착점 등 Input이 입력되면 output인 Switching point까지 주행한다. Switching point에 도착하여 전진과 후진이 바뀌게 되면 그 좌표가 시작점으로 인식되어 또 다음 Switching point까지 경로를 다시 검색한 후 주행한다. 즉, Switching point가 시작점으로 인식되어 input값을 갱신하고, 또 다음 Switching point에 도착하면, 시작점과 input값이 도착점까지 도착할 때까지 갱신되는 방법

이다. Switching point에서 도착점까지 경로를 다시 검색하기 때문에 최적의 경로를 선택하게 된다. 최종적으로 Switching point가 아니라 도착점이라면 경로 탐색이 종료된다.

#### IV. 시나리오 환경

본 연구에서는 다양한 주차환경을 구현하기 위해 서울교통공사 역세권 환승주차장 현황에서 안내하는 환승주차장을 참고하여 시나리오 환경을 구현하였다. 본 연구에서는 상용 소프트웨어인 MATLAB/Simulink를 사용하였다. 시나리오 환경은 직각 주차에 한정해서 설정하였다.

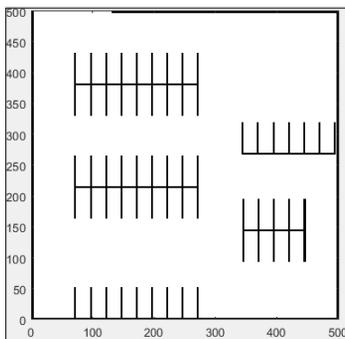
서울교통공사 역세권 환승 주차장은 총 27개역에 29개소가 있고 주차장 구획 수는 총 7,650면이 존재한다. 역세권 환승 주차장의 기준은 출구 300m 이내 위치한 주차장을 기준으로 선정하였다. 본 연구에서는 지하 주차장과 주차타워를 제외하고 구현이 어려운 주차장을 제외하여 총 29개 중 15개의 주차장을 이진 맵으로 구현하였다. 주차장 구조의 다양성을 위해 현재 운영 중인 주차장을 참고하는 것이기 때문에 주차장 구획수와 주차각도를 수정하여 구현하였다. 또한, 이진 맵의 크기가 방대하다면 메모리 요구량이 과다해지고 X축과 Y축의 길이 상이하면 결과의 가시성과 인지성이 저하될 수 있어 맵의 크기를 50×50m로 통일하였다. 또한, 알고리즘 구현의 컴퓨터 사양은 다음과 같다.

<Table 5> Vehicle classification criteria

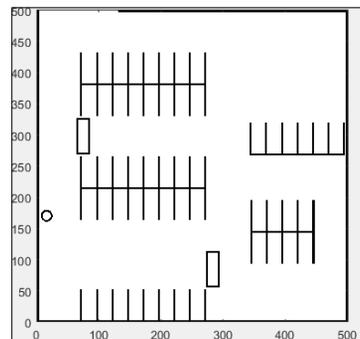
Processor	Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19 GHz
RAM	8.00GB
System	64-bit operating system, x64-based processor
Edition	Windows 11 Pro
version	23H2

##### 1. 일반적인, 제한적인 주차환경

<Fig. 19>과 <Fig. 20>은 일반적인 주차 환경을 나타낸 이진 맵과 제한적인 주차환경을 나타낸 예시 그림



<Fig. 19> Example of parking environment setting (General parking environment)

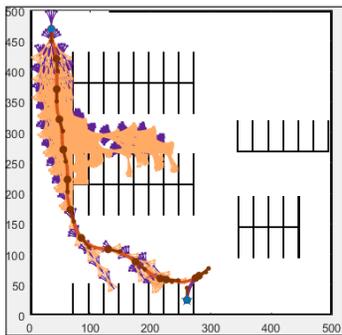


<Fig. 20> Example of parking environment setting (Restricted parking environment)

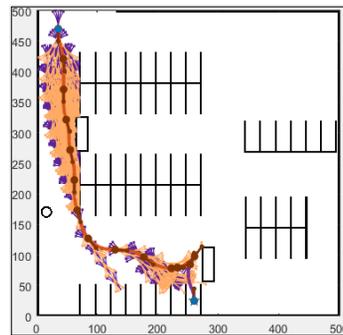
이다. 목표로 하는 주차구역 외 다른 주차구역은 모두 주차가 되어있는 상태로 설정하며, 수직주차와 수평주차에 따라 주차 경로가 달라지기 때문에 목표 주차구역을 변경하며 다양한 주차경로를 계획하였다. 전과 동일하게 목표 주차구역 외 다른 주차구역은 모두 주차가 되어있는 상태이며, 수직과 수평주차를 다양하게 설정하였다. <Fig. 19>는 일반적인 주차환경이다. <Fig. 20>는 주차경로에 주정차하고 있는 자동차나 보행자가 있는 상황이다. <Fig. 19>와 <Fig. 20>처럼 다양한 환경으로 나눈 이유는 장애물에 따른 주차경로의 유연성과 회피성을 확인하여 주차경로생성계획 모델의 성능을 검증하기 위함이다.

## 2. 기 알고리즘 사용 결과

일반적인 주차환경과 제한적인 주차환경에서의 Hybrid A-star 알고리즘과 Reeds-Shepp Curve 알고리즘을 혼합한 기 알고리즘 경로 중 하나이다. 시작점에서 도착점까지 장애물을 회피하고 충돌없이 경로가 생성되었고 이 생성된 경로의 거리, Switching point 개수, 소모시간, 경로생성시간 값을 도출했다. 제한적인 주차환경에서는 불법주차의 장애물을 회피하기 위해 우회 또는 조향각 조정 등의 이유로 생성된 경로의 거리, Switching point 개수, 소모시간, 경로생성시간 값이 증가한 것을 확인하였다.



<Fig. 21> Example of Algorithm generation path (General parking environment)



<Fig. 22> Example of Algorithm generation path (Restricted parking environment)

<Table 6> Results of Algorithm model path

Classification	Distance(m)	Switching point	Time spent(s)	Path generation time(s)
General parking environment	61.4	2	22.10	4.257
Restricted parking environment	64.8	2	23.33	3.846

## 3. 딥러닝 학습과정과 사용 결과

좌표와 y좌표 거리의 차, 장애물 좌표, 주차장 차로길이, 주차자리 폭과 너비 등 Input을 수집, output인 Switching point에서 도착점까지 경로를 다시 검색하기 때문에 최적의 경로를 선택하게 되는 메커니즘이다. 기 알고리즘을 통해 얻은 데이터로 딥러닝 모델에 입력되는 Input과 결과값 Output은 다음과 같다.

시작점과 도착점 등 Input이 입력되면 output인 Switching point까지 주행한다. output인 Switching point가 시작점으로 인식되어 input값을 갱신하고, 또 다음 Switching point에 도착하면, 시작점과 input값이 도착점까지

도착할 때까지 갱신되는 방법이다. 최종적으로 Switching point가 아니라 도착점이라면 경로 탐색이 종료된다. 딥러닝의 결과 값인 Switching point를 서로 이어 경로를 생성한다.

<Table 7> Data used for Deep learning

INPUT	OUTPUT
Size of parking space	Switching Point
Width of a lane	
X-axis and Y-axis differences between the starting and arriving points	
Distance from the arrival point to the wall	
Obstacle Location in Parking Lot	

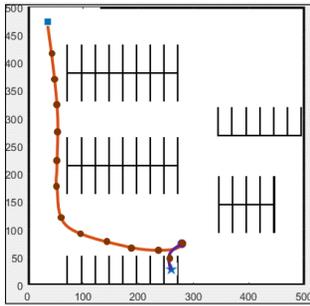
이를 위해 Deep Learning은 Hybrid A-star 알고리즘과 Reeds-Shepp Curve 알고리즘을 혼합하여 생성한 경로 즉, 위의 기 알고리즘 사용 결과에서 생성된 경로를 학습시킨다. 상용 소프트웨어인 MATLAB/Simulink를 이용해 x좌표와 y좌표 거리의 차, 장애물 좌표, 주차장 차로길이, 주차자리 폭과 너비 등 Input을 수집하고 탐색된 경로의 switching point인 Output을 수집하였다. 수집한 데이터를 장애물 및 충돌이 일어났거나 도착지점에 도착하지 못한 경로 등 결측치를 제외하고 계산 해본 결과 총 데이터 수는 약 10만 개로 나타났다. 위에 설명한 내용을 기반으로 주차경로계획 모델을 설정하였고, Learning Environment는 Matlab내 애드온인 Deep Learning Toolbox와 Navigation Toolbox를 사용하였다. 모멘텀을 사용한 확률적 경사하강법을 이용해 신경망을 훈련시키는 옵션 세트를 설정하였다. Epoch 5회마다 학습률을 0.2배만큼 줄일 수 있어, train 진행할 최대 Epoch 횟수를 50으로 설정하고, 각 반복마다 2048개의 관측값을 갖는 mini batch Size를 설정하였다. 모델의 성능과 신뢰도를 향상하고 새로운 데이터에 대한 예측력을 높이기 위해 Train셋과 Test셋의 비율은 8:2으로 설정하였다. 이렇게 학습한 딥러닝 모델의 정확도를 확인하기 위해 Test셋으로 확인한 결과, 정확도는 91.9%로 나타났다. 또한, 정밀도는 94.5%, 재현율은 83.0%, F1 score는 88.7%로 나타났다.

<Table 8> Deep Learning performance metrics

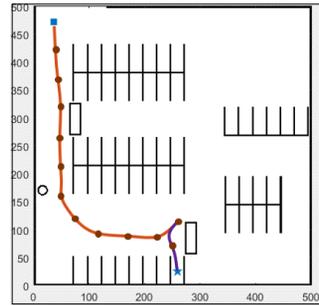
Classification	Result
Accuracy	0.919
Precision	0.945
Recall	0.830
Specificity	0.966
F1-Score	0.887

<Table 9> Results of Deep Learning model path

Classification	Distance(m)	Switching point	Time spent(s)	Path generation time(s)
General parking environment	61.2	2	22.03	2.397
Restricted parking environment	64.7	2	23.29	2.347



<Fig. 23> Example of Deep Learning model generation path (General parking environment)



<Fig. 24> Example of Deep Learning model generation path (Restricted parking environment)

### V. 결 론

본 연구에서는 기 알고리즘과 딥러닝을 통한 주차경로계획 생성 모델을 비교하기 위해 시작점과 도착점을 동일한 곳으로 설정하였다. 그 결과, 두 방법으로 나온 경로는 큰 차이 없이 비슷한 경로가 생성되었다. 먼저 일반적인 주차환경에서, 거리의 경우 평균적으로 0.53% 감소하였고 소모시간은 평균 0.54% 감소, Switching point는 평균 1.3개에서 1.1개로 변화하였고, 경로생성시간은 평균 42.40% 감소하였다. 제한적인 주차 환경에서는 거리의 경우 평균적으로 0.67% 감소하였고 소모시간은 평균 0.68%, Switching point는 평균

<Table 10> Comparison of Results in General Parking Environments

Classification	Distance(m)	Switching point	Time spent(s)	Path generation time(s)
Algorithm model	58.7	1.3	21.14	5.513
Deep Learning model	58.4	1.1	21.02	3.176
rate of change(%)	-0.53%	-10.53%	-0.54%	-42.40%

<Table 11> Comparison of Results in Limited Parking Environments

Classification	Distance(m)	Switching point	Time spent(s)	Path generation time(s)
Algorithm model	77.5	1.4	27.91	8.261
Deep Learning model	77.0	1.3	27.72	4.709
rate of change(%)	-0.67%	-4.76%	-0.68%	-42.99%

<Table 12> Comparison of Results between models

Classification	Distance(m)	Switching point	Time spent(s)	Path generation time(s)
Algorithm model( <i>a</i> )	68.1	1.3	24.52	6.887
Deep Learning model( <i>b</i> )	67.7	1.2	24.37	3.942
Variation of change( <i>b - a</i> )	-0.4	-0.1	-0.15	-2.945
rate of change( $\frac{b-a}{a} \times 100$ )(%)	-0.59%	-7.50%	-0.61%	-42.76%

1.4개에서 1.3개로 변화하였고, 경로생성시간은 평균 42.99% 감소하였다.

본 연구에서는 주차장을 구현하여 일반적인 주차환경과 제한적인 주차환경으로 나누어 Hybrid A-star 알고리즘과 Reeds-Shepp Curve 알고리즘을 혼합하여 주차경로생성 알고리즘을 만들고, 이를 Deeping Learning을 통해 추정차된 자동차, 보행자 같은 장애물을 회피하는 주차경로생성 알고리즘을 제안하였다. 서울교통공사에서 제공하는 역세권 환승 주차장을 참고하여 15개의 다양한 환경에서 검증하였다.

기 알고리즘과 본 연구에서 제안한 Deeping Learning 모델을 비교한 결과, 장애물에 충돌하지 않고 비슷한 경로를 생성하지만, 거리와 소모시간이 각각 0.59%, 0.61% 감소하여 보다 효율적인 경로가 생성되었다. 또한, Switching point도 1.3개에서 1.2개로 감소하여 직진과 후진을 최대한으로 줄여 운전자의 피로를 줄일 수 있을거라 생각된다. 마지막으로 경로생성시간은 42.76% 감소하여 보다 효율적이고 신속한 경로생성이 가능한 것을 확인하였다.

이러한 연구결과는 향후 자율주행 중 자율주차의 경로계획생성에 활용될 수 있으며, 차량작도에 따라 이동하는 주차로봇의 경로생성에도 활용될 수 있을 것으로 보인다. 더 나아가, 차량 종류나 제원에 따른 작도와 최소, 최대 조향각을 설정할 수 있다는 점을 보았을 때, 각 차량에 따른 최적의 자동주차경로생성에 활용할 수 있다고 사료된다.

그러나 본 연구에서 정확하고 안전한 최적의 주차경로계획생성을 위해서는 주차장의 높이, 경사도 등의 지형을 적용하여 수행되어야하나 이를 반영하지 못한 점과 보행하고 있는 보행자와 출입하고 있는 차량, 주차하고 있는 차량 등 움직이는 장애물이 아닌 정지되어있는 장애물을 회피하는 경로만 수행하는 단편적인 분석만 수행했다는 점에서 현실성을 반영하지 못한 한계점을 가진다. 또한, Distance, Switching point, Time spent, Path generation time 중 Path generation time만 유의미한 감소가 있고, 나머지 지표는 큰 차이가 없는 것으로 분석되었다. 이는 제안한 알고리즘 모형이 기존의 기 알고리즘의 메커니즘과 동일하여 이미 최적화된 경로와 별 다른 차이가 없는 것으로 판단되어 더 효율적이고 신속한 경로 생성에 적용하기 보다는 경로 생성시간을 줄이는 모형으로 적용하는게 적절한 것으로 보인다. 향후 연구 과제로는 앞서 서술했던 움직이는 장애물을 설치하고 이를 실시간으로 인식, 회피함과 동시에 최적을 경로를 생성하는 모델을 만들고자 한다. 나아가, 주차하려는 자동차 1대의 최적경로생성인 Path-planning 뿐만 아니라 여러 자동차들이 동시다발적으로 주차를 할 때 가장 최적의 방법을 찾아내는 Route-planning을 반영하여 주차장 내 주차면 활용도를 높일 수 있을 것으로 기대된다.

## ACKNOWLEDGEMENTS

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2022R1C1C1005640)

## REFERENCES

- Kang, D. H., Lee, S. H., Kang, C. M. and Chung, C. C.(2018), "Parking path planning using path folding method(2018)", *Domestic Studies of the Society of Control Robot Systems Published by the Society of Liquor*, vol. 2018, no. 6, pp.92-293.
- Kim, D. H. and Jeong, W. J.(2008), "Creating parking routes for automatic parking using M-space",

- Journal of Korea University, Robotics Association*, vol. 3, no. 1, pp.1-8.
- Kim, J. H.(2022a), *Parking path planning in constrained parking spaces using deep learning*, Master's Degree Papers, Hanyang University.
- Kim, J. H.(2022b), "Automatic parking system implemented with MATLAB and Simulink | ADAS and autonomous stocks. New features of MATLAB and the utilization of deep learning for row systems", *MathWorks Korea*, Youtube, 1941-2020.
- Kim, J. M.(2016), *Development of an autonomous parking system based on real-time route planning of autonomous vehicles*, Master's thesis, Kookmin University.
- Kim, M. S., Ahn, J. W., Kim, M. S., Shin, M. Y. and Park, J. H.(2021), "A comparative analysis of path planning and tracking performance according to the consideration of vehicle's constraints in automated parking situations", *Journal of Korea Robotics Society*, vol. 16, no. 3, pp.250-259.
- LaValle, S. M.(2006), *Planning algorithms*, Cambridge, UK: Cambridge University Press.
- Lee, J. W.(2016), *Research on domestic parking accidents and reverse automatic braking devices*, Korea Insurance Development Institute Automotive Technology Research Institute, 2016 Han The National Automobile Engineering Association's estimate.
- Lee, S. I.(2022), "Improving RRT\* family algorithm performance in circle and sphere obstacle environments using sampling range limitations", *Ajou University, Journal of the Korean Aerospace Society*.
- Lee, S. J.(2023), "Study on deep learning-based parking space optimal path search algorithm for autonomous parking systems", *Gyeonggi University Kyodo, 2023 Summer of the Korean Society of Information Technology*.
- Lei, C., Hsin, G., Hao, L. Z., Xin J. and Jun, Z.(2022), "Multi-maneuver vertical parking path planning and control in a narrow space", *Robotics and Autonomous Systems*, vol. 149, 103964.
- MathWorks(2023a), *Use Hybrid A Star to generate code for path planning*, MathWorks Help Center.
- MathWorks(2023b), *Planner Hybrid A Star*, MathWorks Help Center.
- MathWorks(2023c), *Train deep learning-based sampler of motion planning*, MathWorks Help Center.
- Ministry of Government Legislation(2021), *Enforcement rules of the automobile management act (Attachment 1)*.
- Reeds, J. A. and Shepp, L. A.(1990), "Optimal paths for a car that goes both forwards and backwards", *Pacific Journal of Mathematics*, vol. 145, no. 2, pp.367-393.
- SAE(Society of Automotive Engineers)(2021), *Levels of driving automation*.
- Seoul Transportation Corporation(2023), *Customer convenience facilities - Transfer parking lot*, Seoul Transportation Corporation usage information.
- Thoresen, M., Nielsen, N. H., Mathiassen, K. and Pettersen, K. Y.(2021), "Path Planning for UGVs based on Traversability Hybrid A\*", *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 1216-1223.
- Vorobieva, H., Glaser, S., Minoiu-Enache, N. and Mammar, S.(2015), "Automatic parallel parking in tiny spots: path planning and control", *In IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp.396-410.
- Yoo, Y. G. and Park, Y. J.(2013), "A study on the reverse application method for improving the accuracy of the shortest path search of the A\* algorithm", *Han Journal of the Korean ITS Society*, vol. 12, no. 6, pp.1-9.