

자차배송을 위한 적재 및 경로 최적화

B반 2조 | 김아연 김한탁 류지연 서지윤 오석훈 하지민



Contents



01 추진배경

02 프로젝트 목표 및 개요

03 활용 기술

04 기업 사례

05 논문 분석

06 전기수 분석

07 시나리오

08 진행 상황

09 추진 일정

추진 배경

POINT 01

중국 e-커머스의 공습



- 중국 이커머스 플랫폼(C커머스)
- 알리익스프레스(알리)·테무·웨이인 등
- 쿠팡, 3조원 이상을 투자해 2027년까지
로켓배송(당일·익일배송) 지역 전국 확장 계획

POINT 02

국내 e-커머스의 라스트마일 배송 경쟁



- 국내 e-커머스는 새로운 배송형태인
신 라스트마일 배송 시스템 구축을 계획
- 2016년, 익일배송의 비율이 2%였으나
2022년에는 일반적인 배송 방식으로 자리 잡음

*라스트 마일
: 물류 센터에서 구매자 집 앞까지 주문한 물품이 배송되는 전과정

POINT 03

쿠팡플렉스를 시작으로 일반인 자차 배송 확산



- 일반인 인력을 활용한 자차 배송 시스템이
e커머스 시장 전반으로 확산
- 자차 배송 시스템은 급증하는 주문량에도
쿠팡이 로켓배송 속도를 유지하는 비결의 하나

프로젝트 목표 및 개요

적재 공간 및 배송 경로 최적화

강화학습 기반 AI 모델 개발

Unity 시뮬레이션 환경 구축

자차배송을 위한 적재 및 경로 최적화

C커머스의 등장으로 국내 e-커머스 기업들의
라스트마일 배송 경쟁이 심화되어
새로운 라스트마일 배송 시스템인 자차 배송
대상으로 적재 및 경로 최적화 필요성 증가



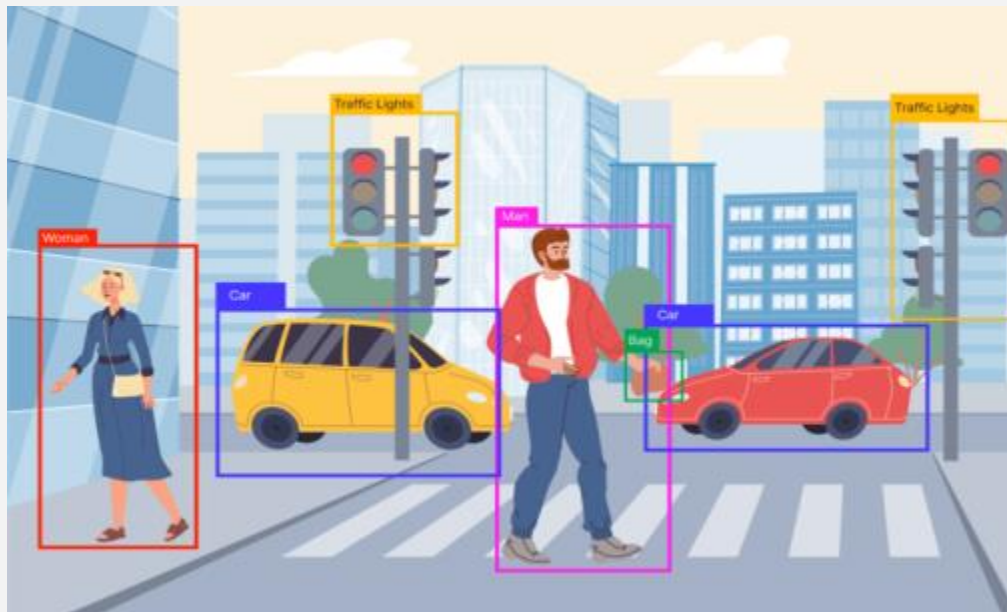
배송 효율성 및 생산성 향상

신규 라이더의 진입장벽 완화

고객만족도 향상

POINT 01

Object Detection



찾고자 하는 객체의 특징을 사전에 추출하고
주어진 영상 내 해당 객체의 특징을 검출하여
식별 및 분류하는 방법

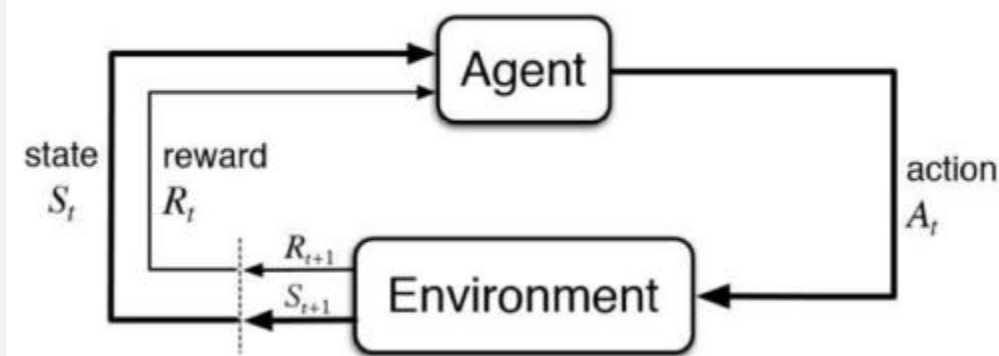


여러 차량의 이미지를 딥러닝 모델을 이용해 학습
차량의 종류를 인식하여
해당 차량의 트렁크 내 적재 가능 용량을 알려주도록 함

POINT 02

강화학습

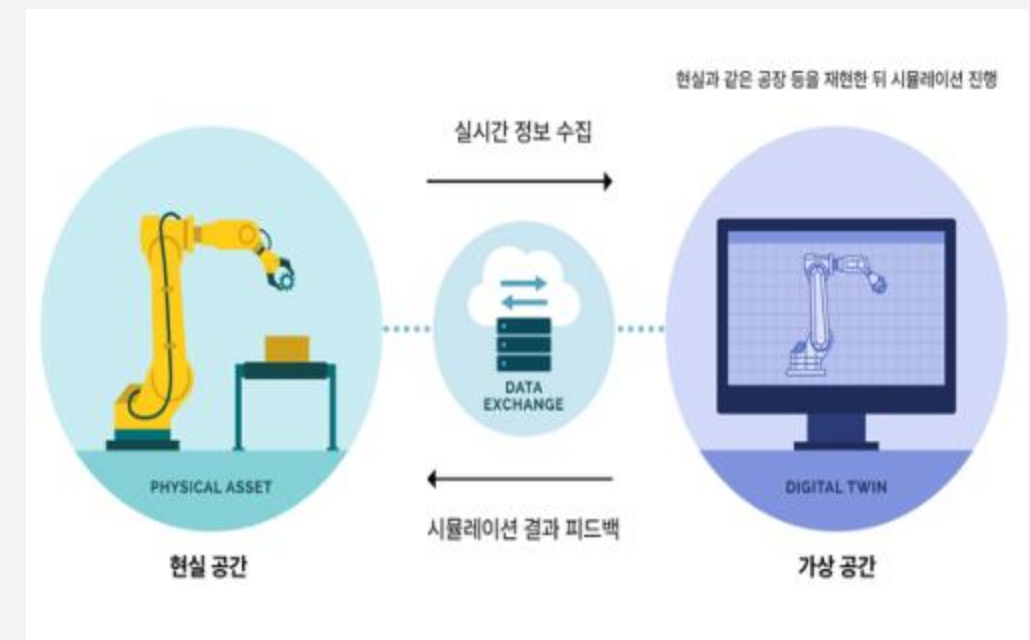
Learn from mistake



- Agent : 로봇팔
- Action : 로봇팔이 상자를 들어 특정 위치로 이동
- Environment : 가상환경 속 물류센터
- State : 자동차 별 트렁크 점유율
- Reward : 쌓은 상자의 부피(무게, 부피가 높을 수록 아래 깔리도록), 배송지의 우선순위가 높은 경우(맨 위에 쌓이도록)

POINT 03

디지털 트윈



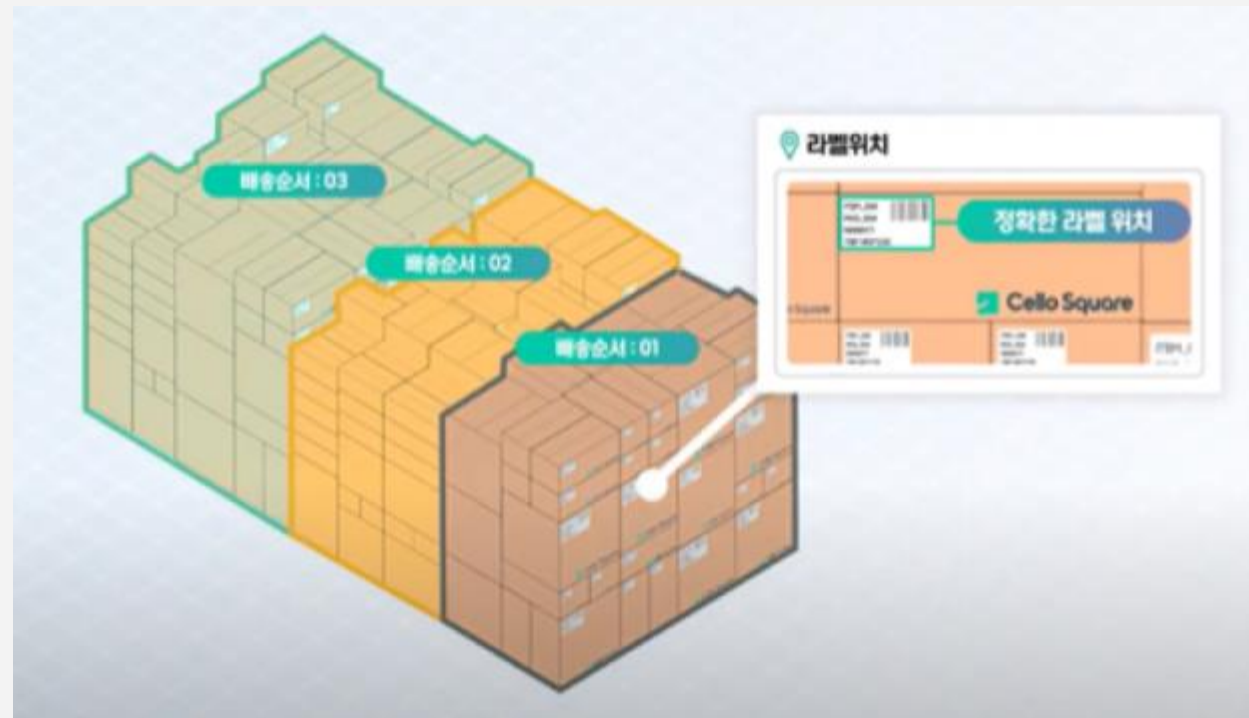
컴퓨터에 현실 속 사물의 쌍둥이를 만들고,
현실에서 발생할 수 있는 상황을
컴퓨터로 시뮬레이션함으로써 결과를 미리 예측하는 기술



유니티 내 로봇 팔(agent) 강화학습을 통해
차량 내부 부피 공간의 배송지 우선순위를 고려한
최적 적재 알고리즘 적용 및 디지털 트윈 구축

POINT 01

삼성 SDS 첼로스퀘어 - 적재 최적화 서비스



- 창고 운영의 생산성과 효율성 향상
 - 화물을 실을 때 고려해야하는 적재 공간, 배송 순서, 라벨 위치 등 모든 요소를 고려하여 시뮬레이션하고, 화물에 맞는 컨테이너 수량 및 적재율을 안내

POINT 02

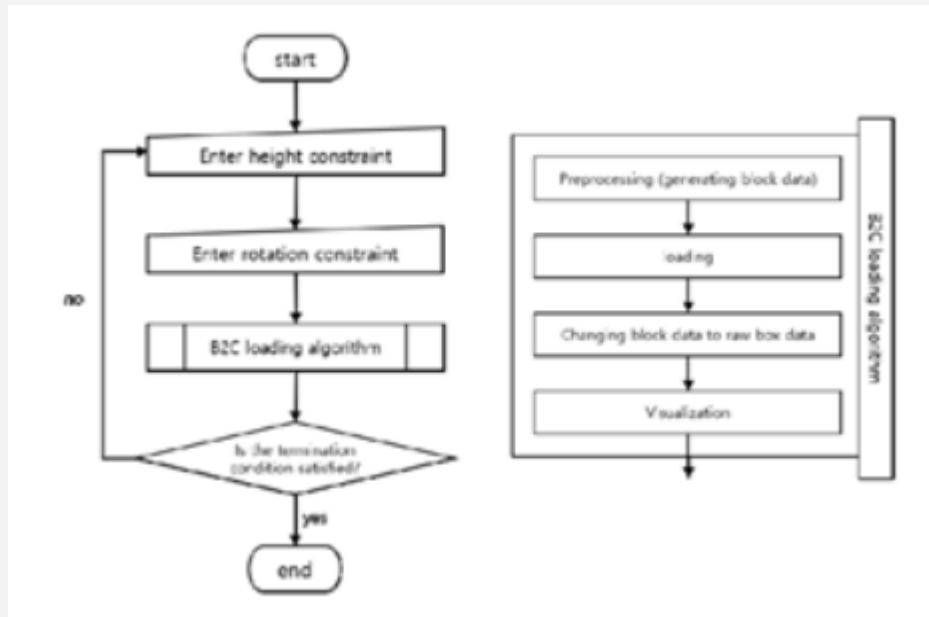
LG CNS & UNITY



- LG CNS는 유니티의 3D엔진 기반으로 고객사 제조공장 공간과 설비를 가상화
- 가상화 공간에 각종 공장과 물류센터 데이터를 연계해 실제 공장을 원격 운영하는 메타버스 환경을 구축

POINT 01

B2B, B2C 물류 환경을 고려한 3차원 차량 적재 알고리즘 개발



- Clustering: 배송 경로 정보와 물건의 배송지 정보를 통해 물건의 배송 권역 구분 및 배송 순서 할당
- Grouping 과정을 통해 배송 권역이 구분된 제약 조건 만족하는 블록 생성
- 정렬된 블록의 순서에 따라 배치를 시작하여 트럭을 모두 채울 때까지 반복

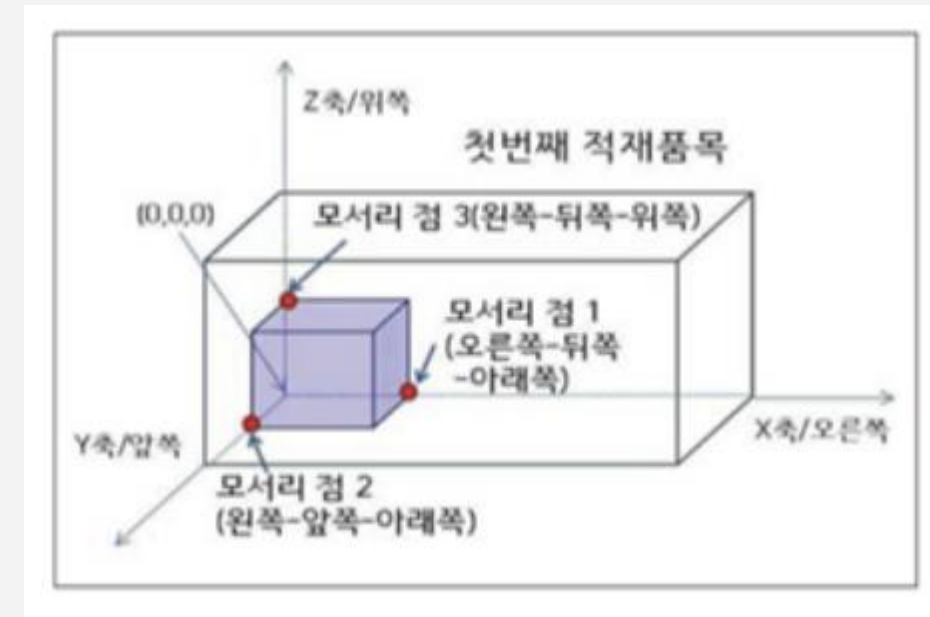


서로 다른 타입의 적재물 간의 안정적 적재 가능
최적 배송을 위한 적합한 의사결정 가능

류영태, 이태훈, 송병덕, 이상덕. (2022). B2B, B2C 물류 환경을 고려한 3차원 차량 적재 알고리즘 개발. 로지스틱스연구, 30(6), 89-103.

POINT 02

배송 지점과 품목의 우선순위 제약 조건하 3차원 품목 적재 해법



- 배송 지점의 우선순위와 도착순서, 품목의 우선순위를 고려
- 타부 서치를 적용하여 추가 품목의 적재 가능 여부를 확인 후 추가 적재가 가능한 적재 방법을 선택하는 휴리스틱 방법론을 사용한 적재 방안



지점 방문 및 품목 우선순위 제한 사항을 포함한 제시를 통해
다양한 문제 적용 가능

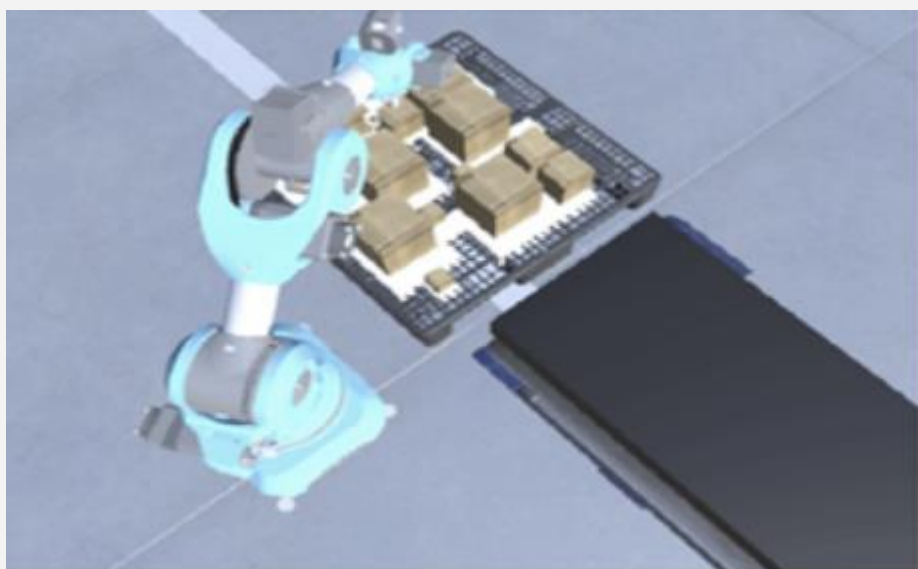
*타부서치: 한정된 시간 내에 최적의 해 대신 현실적으로 만족할 만한 수준의 해를 구현

*휴리스틱 알고리즘: 경험, 규칙, 또는 추정치와 같은 지식을 기반으로 문제 해결을 위한 가이드 라인을 제시하여 어려운 문제 해결
송수민, 이문걸. (2014). 배송 지점과 품목의 우선순위 제약 조건하 3차원 품목 적재 해법. 로지스틱스연구, 22(1), 17-28, 10.15735/ks.2014.22.1.002

전기수 분석

POINT 01

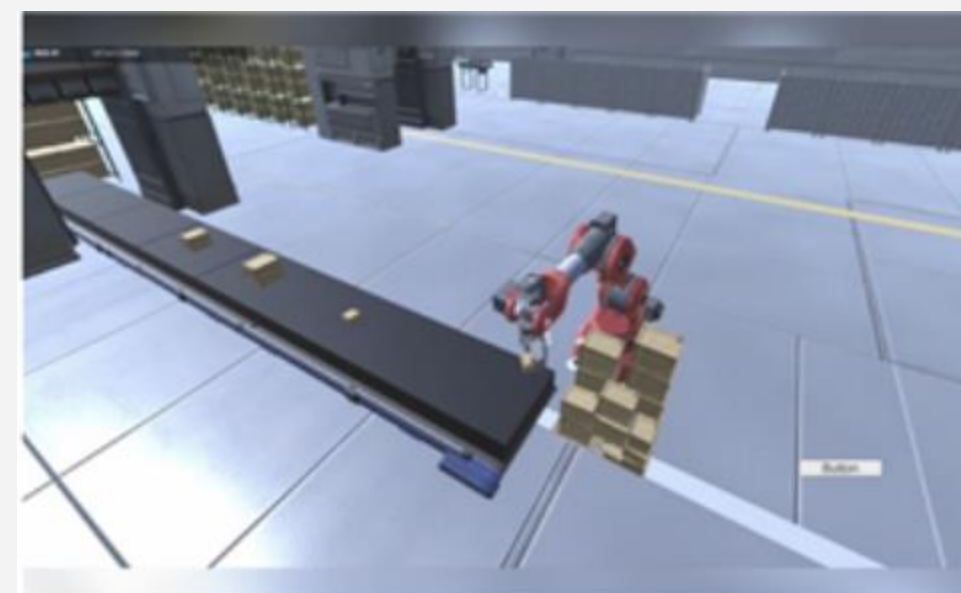
21기 C3조 PBA Twins



- 강화학습 기반 디지털 트윈을 활용한 물류 최적화 시스템
- 주요기능: Unity(가상환경 구현), PPO(강화학습 알고리즘), Bin Packing

POINT 02

22기 C4조 'CPost(C4st)'



- 강화학습을 통한 3D 물류 적재 최적화 시스템
- 주요기능: Unity(가상환경 구현), ACTKR(강화학습 알고리즘), ROS(로봇팔 제어), 3D Bin Packing System(3차원 물류 적재 시스템)

차별점

- 밀도와 무게만을 고려하여 최적 적재 방안을 도출 -> 밀도, 무게 뿐만 아니라 배송지의 우선순위를 고려하여 물건의 최적 적재 방안을 도출
- 표준 산업용 팔레트 사이즈인 110cm x 80cm x 170cm를 기준 -> 차 종에 따른 트렁크의 실제 크기를 바탕으로 다양한 팔레트 크기를 새롭게 설정

시나리오



POINT 01

차량 Object Detection -> 차종 인식

Car-1



POINT 02

차종에 맞는 트렁크 부피 공간 생성



POINT 03

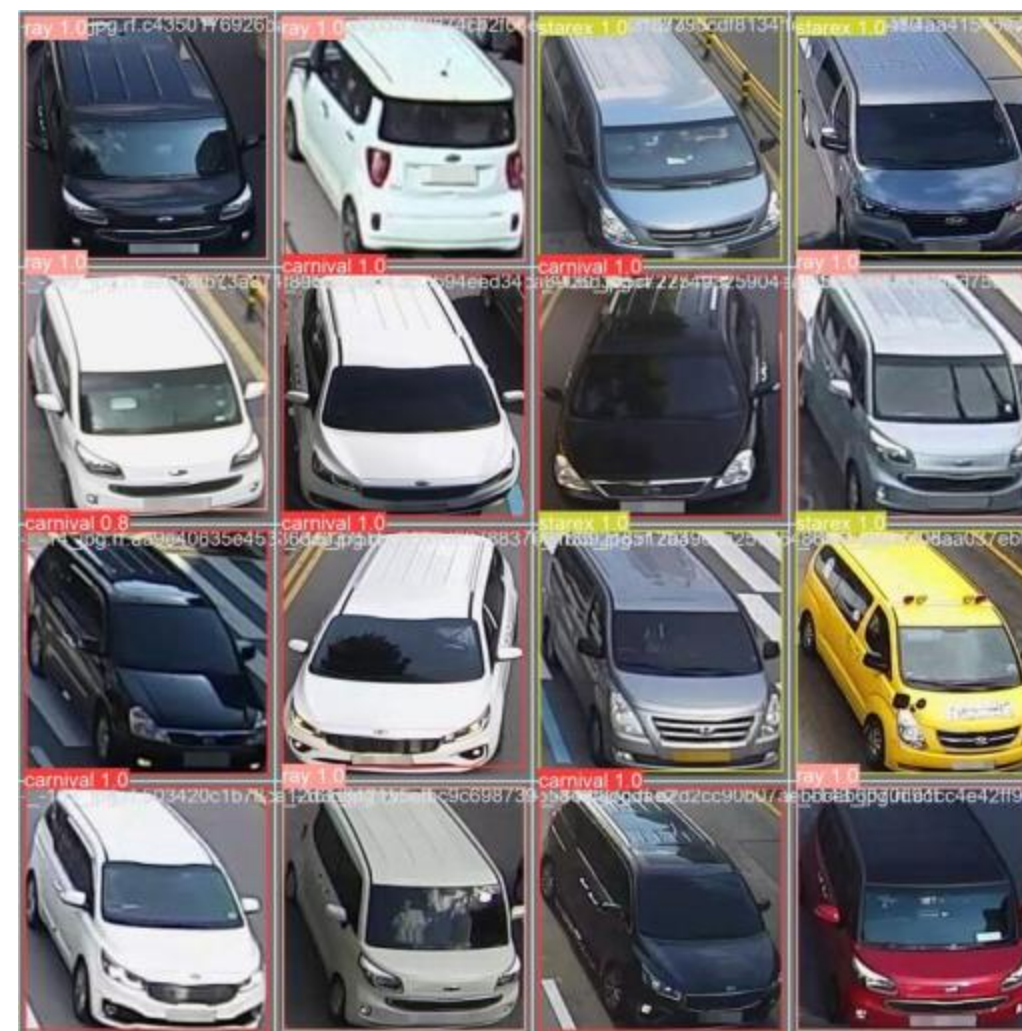
Random 박스 사이즈에 대한
강화학습 알고리즘 기반
적재 최적화 및 배송 경로 최적화



Object Detection

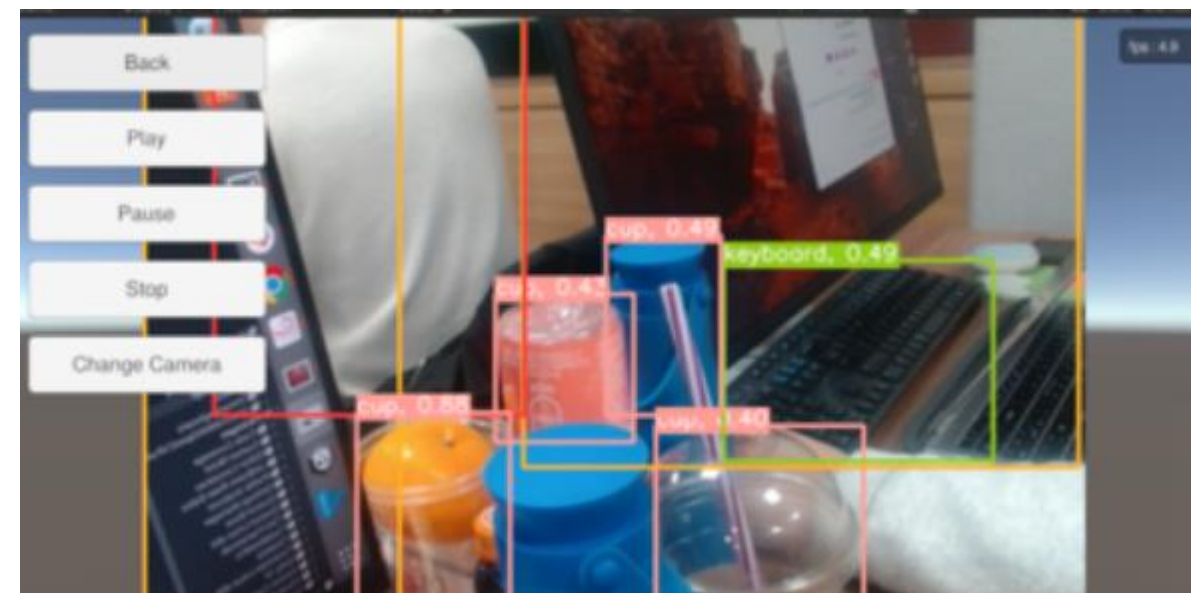
| | 현재 진행 상황 |
|---|-------------------------------------|
| ✓ | AIHUB '자동차 차종/연식/번호판 인식용 영상' 데이터 수집 |
| ✓ | 차량 종류별 사진 라벨링(Roboflow 이용) |
| ✓ | 위를 활용한 YOLOv8 모델 학습(epoch=200) |
| ✓ | 모델 성능 확인 |

| | 향후 계획 |
|--|-----------------------------|
| | 차종에 따른 트렁크 부피 값 반환할 수 있도록 함 |
| | 최종 모델 성능 개선(바운딩 박스, 정확도 등) |



Unity

| | 현재 진행 상황 |
|---|--------------------------------|
| ✓ | Unity 외 다른 디지털 트윈 환경 분석 |
| ✓ | 유니티 환경 Setting & Unity와 ROS 통신 |
| ✓ | 전 기수(22기 C4조) 시뮬레이션 |
| ✓ | 유니티 내 YOLOv5 모델 구현 |
| | 향후 계획 |
| | YOLOv5 모델에 차종 객체 인식 C# 스크립트 작성 |
| | 차량 인식 후 차종에 맞는 트렁크 공간 생성 |
| | 강화학습 알고리즘 ML-Agents 이용하여 연동 |
| | 배송지 고려 알고리즘 적용한 로봇팔 시뮬레이션 |

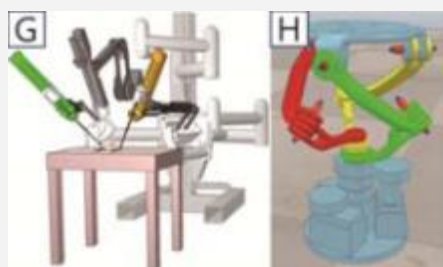


Unity

현재 진행 상황 ✓ Unity 외 다른 디지털 트윈 환경 분석

Coppeliasim

- 로보틱스 분야에서 가장 많이 사용되는 교육용 시뮬레이터
- 초기 공장 자동화 시스템의 시뮬레이션 수행하기 위해 개발
- ROS 지원은 없지만 CoppeliaSim API 프레임워크를 통한
- libsimExtROS이라는 ROS 인터페이스를 통해 ROS1/ROS2 연동이 가능



| 장점 | 단점 |
|--|---|
| <ul style="list-style-type: none"> - 실시간 시뮬레이션 루프가 가능하며 시뮬레이션 중 장면 수정 가능 - 사용할 물리 엔진 쉽게 변경 가능 - 편리한 접근성과 다양한 로봇 모델 지원 | <ul style="list-style-type: none"> - 시뮬레이션 속도 느림 |

open AI gym (Issac Gym)

- Isaac Sim 시뮬레이터를 이용해 Open AI Gym과 같이 강화학습을 진행하기 위해 사용되는 Isaac Sim 내 하나의 툴

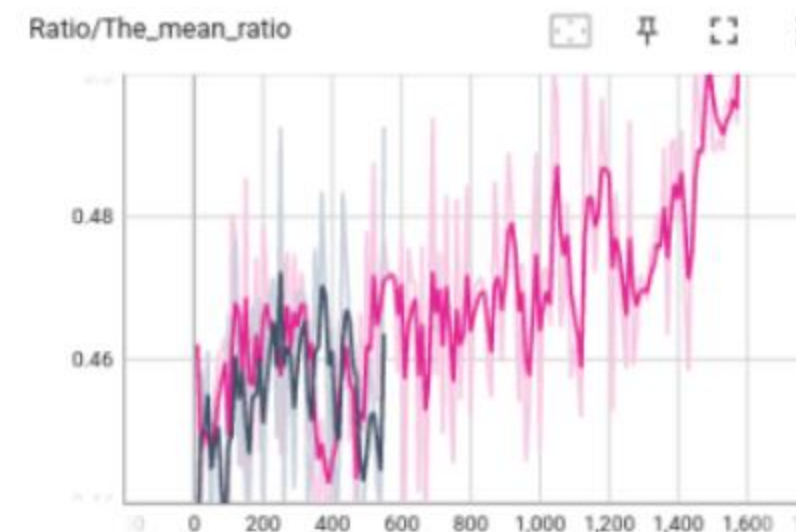


| 장점 | 단점 |
|--|--|
| <ul style="list-style-type: none"> - 하나의 시뮬레이션 환경에서 다수의 로봇들이 학습을 동시에 진행할 수 있음 - NVIDIA RTX 기술을 통해 사실적 묘사가 가능 | <ul style="list-style-type: none"> - 계속 진행 중인 업데이트로 인해 기능 상의 변경 가능성이 존재 - 관련 예제 및 문서 부족 - 커뮤니티 부족 |

알고리즘

| | 현재 진행 상황 |
|---|---|
| ✓ | 전 기수(22기 C4조, 23기 B3조), 논문 원본 제약조건 수정 |
| ✓ | 알고리즘 3가지(a3c, policy_gradient, TRPO) 추가 |
| ✓ | box 랜덤 난수 조건 추가 |
| ✓ | continous domain 에서의 밀도 제약 조건 추가 |

| | 향후 계획 |
|--|---|
| | Object Detection 모델을 통한 트렁크 사이즈별 팔레트 구현 |
| | 배송지 제약 조건 추가 수정 |



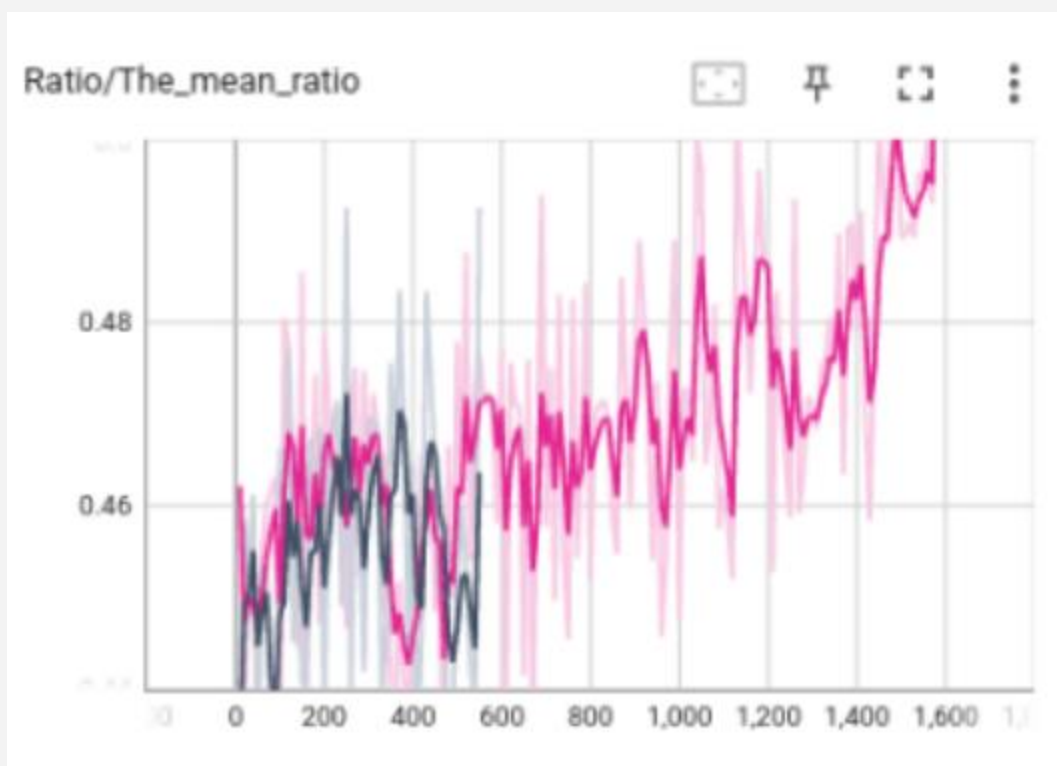
```

9 item_size_set=[]
10 for x in range(1000):
11     random_numbers = [
12         random.uniform(20,50), # 24에서 54 사이의 난수
13         random.uniform(15,40), # 19에서 39 사이의 난수
14         random.uniform(10,35)  # 9에서 34 사이의 난수
15     ]
16     item_size_set.append(random_numbers)
17

```

알고리즘

현재 진행 상황 ✓ 알고리즘 3가지(a3c, policy_gradient, TRPO) 추가



- 알고리즘 별(trpo(검정), acktr(자주색)) 학습 횟수에 따른 평균 ratio(적재량)를 비교하는 그래프
- 해당 그래프를 통해 각 알고리즘을 동일한 학습 횟수로 ratio를 비교할 예정

- update 20번 한 학습 결과

```
##### acktr #####
Time version: 0411acktr_c_2-2024.04.11-23-44-51 is training
Updates 20, num timesteps 6720, FPS 510
Last 10 training episodes: mean/median reward 4.4/4.5, min/max reward 3.4/5.1
The dist entropy 3.84254, the value loss 68.17518, the action loss -4.55340
The mean space ratio is 0.4379827116762125, the ratio threshold is 0.5185155121857854
```

- update 110번 한 학습 결과 → 팔레트의 ratio가 높아짐

```
##### acktr #####
Time version: 0411acktr_c_2-2024.04.11-23-44-51 is training
Updates 110, num timesteps 35520, FPS 519
Last 10 training episodes: mean/median reward 4.8/4.8, min/max reward 4.1/5.4
The dist entropy 3.85894, the value loss 242.09029, the action loss 2.17443
The mean space ratio is 0.4805276296653309, the ratio threshold is 0.5653443536097226
```


알고리즘

현재 진행 상황 ✓ box 랜덤 난수 조건 추가

- '마켓컬리', '쿠팡', '우체국' 박스 크기를 조사
- 박스 규격의 가로·세로·높이의 max, min 값의 범위를 정함
- 그 안에서 균등한 분포로 랜덤하게 box 사이즈가 구현되도록 함

```
9 item_size_set=[]
10 for x in range(1000):
11     random_numbers = [
12         random.uniform(20,50), # 24에서 54 사이의 난수
13         random.uniform(15,40), # 19에서 39 사이의 난수
14         random.uniform(10,35)  # 9에서 34 사이의 난수
15     ]
16     item_size_set.append(random_numbers)
17
```

알고리즘

현재 진행 상황 ✓ continous domain에서의 밀도 제약 조건 추가

- 전기수(22기 C4조) discrete domain이었지만, 이번 프로젝트에는 continous domain에서의 밀도 제약 조건 추가

| | 전기수 | 현재 |
|------|----------------------------|-----------------------|
| 강화학습 | discrete domain | continous domain |
| 설명 | 강화학습 시 상태와 행동이 구별되는 값 | 강화학습 시 상태와 행동이 연속적인 값 |
| 예시 | 체스 말은 정해진 위치로만 움직일 수 있는 경우 | 로봇 팔의 관절 각도와 같은 문제 |

```

68
69 # 밀도 저장
70 def get_box_density(self):
71     density = self.next_den
72     return density
73
74 def reset(self):
75     self.box_creator.reset()
76     self.packed = []
77     self.space.reset()
78     self.box_creator.generate_box_size()
79     cur_observation = self.cur_observation()
80     return cur_observation
81

```

```

205 box_ratio = self.get_box_ratio()
206 box_density = self.get_box_density()
207 self.box_creator.drop_box() # remove current box from the list
208 self.box_creator.generate_box_size() # add a new box to the list
209
210 reward = box_ratio * 10 + 0.007 * box_density * (self.bin_size[2] - z) #일단 밀도에 따른 reward 차이 있는지
211

```

추진 일정

| SUN | MON | TUE | WED | THU | FRI | SAT |
|---------------------------------|-----|-----|-------|-------|--------|--------|
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| | | | | | 3주차 발표 | OD 마무리 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 강화학습 알고리즘 개발, Unity 시뮬레이션 환경 구축 | | | | 모델 통합 | | |
| 21 | 22 | 23 | 24 | | | |
| 테스트 및 완성 | | | 최종 발표 | | | |

+ Q & A +



이상으로 발표를 마치겠습니다. 감사합니다!

THANK YOU FOR
ATTENTION

