



# Sim2Real 접근을 통한 Isaac Sim 기반 자율주행 인공지능의 실제 주행 구현 기술 개발

고국원<sup>a,\*</sup>

## Development of Real-World Implementation Technology for Isaac Sim-Based Autonomous Driving AI Using a Sim2Real Approach

Kuk Won Ko<sup>a,\*</sup><sup>a</sup> Department of Future Mobility Engineering, Halla University

### ARTICLE INFO

#### Article history:

Received	11	October	2025
Revised	18	November	2025
Accepted	19	November	2025

#### Keywords:

Autonomous mobile robot (AMR)  
CNN-based localization  
Isaac sim  
Sim2Real  
Virtual twin

### ABSTRACT

A zero-shot Sim2Real transfer framework is developed, in which autonomous driving intelligence is trained entirely within a high-fidelity NVIDIA Isaac Sim virtual twin environment and deployed directly on a physical autonomous mobile robot (AMR). Two core models, a YOLOv8-based line-detection model and a CNN-based LiDAR localization model, were developed for navigation. Both were trained using synthetic sensor data generated solely in a virtual environment without real-world data collection. The trained models were applied to a real differential-drive AMR without additional calibration or fine-tuning. Real-world experiments in a warehouse environment showed that the proposed system achieved stable line-following and reliable global pose estimation and performed at a level comparable to or exceeding that of the ROS2 AMCL baseline. These results demonstrate that virtual-twin-based simulations can effectively support the direct deployment of AI models to real AMR platforms, reduce the development time, and minimize the physical data acquisition demand.

## 1. 서론

최근 물류 산업에서는 Autonomous Mobile Robot(AMR)을 이용한 자동화 시스템이 급속도로 확산되고 있다. 이에 따라 공장 설계 단계에서부터 AMR 운용까지 통합적으로 고려하는 Virtual Twin 기반 공장 개발 기술의 중요성이 커지고 있다.

Virtual Twin 기술은 실제 공장의 설비·라인·물류 동선을 3차원으로 재현하여, 설계 초기 단계에서 AMR의 경로 최적화, 충돌 회피, 자율주행 경로 계획 등을 시뮬레이션할 수 있는 장점을 지닌

다. 이를 통해 설계·시공·운영 전 주기에 걸쳐 오류를 사전에 검증하고, 생산 중단 없이 시스템 개선을 반복할 수 있는 지능형 공장 구현이 가능하다. 이러한 Virtual Twin 기술은 이미 글로벌 기업에서 활발히 활용되고 있다.

Amazon Robotics는 수천 대의 AMR을 디지털 트윈 환경에서 시뮬레이션하고, 물류 흐름 최적화를 위한 인공지능 학습을 병행함으로써 실제 창고 운영 효율을 향상시키고 있다<sup>[1,2]</sup>.

BMW Group은 NVIDIA Omniverse를 기반으로 실제 공정과 동일한 수준의 Virtual Factory를 구축하고, 로봇의 충돌 검사, 작

\* Corresponding author. Tel.: +82-33-760-1579

E-mail address: kuksauto@hanmail.net (Kuk Won Ko).

업 경로 및 라인 변경 시나리오를 가상 환경에서 검증함으로써 생산성 향상과 테스트 비용 절감 효과를 입증하였다<sup>3,4)</sup>.

또한 NVIDIA는 산업형 디지털 트윈을 위한 시뮬레이션 및 로봇 학습 도구로 Isaac Sim을 제공하며, Virtual Twin-AI-AMR 융합 기술은 차세대 제조 인프라의 핵심 기술로 자리 잡고 있다<sup>5,6)</sup>.

자율주행 인공지능의 학습과 검증을 위해서는 현실과 유사한 물리·시각적 환경을 제공하는 시뮬레이터가 필수적이다.

그러나 기존의 Gazebo나 Webots 등 범용 시뮬레이터는 그래픽 품질과 센서 모델의 정밀도가 낮아 AI 학습용 환경으로는 적합하지 않다. 특히 영상 인식 기반 학습(deep learning visual perception)의 경우, 현실감이 떨어지는 데이터셋이 생성되어 실제 적용 시 성능 저하가 발생하는 한계가 있다.

이러한 한계를 극복하기 위해 최근에는 Unreal Engine 5(UE5)와 NVIDIA Isaac Sim을 활용한 고정밀 시뮬레이션 연구가 활발히 진행되고 있다<sup>7,8)</sup>.

현재 활용되는 시뮬레이션 접근법은 크게 두 가지로 구분된다.

첫 번째는 Unreal Engine 5(UE5) 기반의 시각 중심 시뮬레이션이다. UE5는 고해상도 Physically Based Rendering(PBR), Lumen 글로벌 조명, 실시간 Ray Tracing 기술을 통해 현실과 구분이 어려운 수준의 시각적 사실감을 구현할 수 있다. 이러한 특성으로 인해 UE5는 AI 비전 인식, 자율주행차 학습용 영상 데이터셋 합성(synthetic dataset) 등 시각 중심 연구에 널리 활용되고 있다. 그러나 물리 충돌, 마찰, 센서 노이즈 등 정밀한 물리 기반 로봇 제어 시뮬레이션에는 한계가 존재한다.

두 번째는 NVIDIA Isaac Sim을 이용한 물리 기반 시뮬레이션이다. Isaac Sim은 Omniverse 플랫폼 위에서 동작하며, PhysX 5 물리엔진과 RTX 렌더링을 통해 LiDAR, IMU, 카메라 등 실제 센서의 물리 특성을 정밀하게 재현할 수 있다. 또한 ROS2 인터페이스를 지원하여 실제 제어기와 통신 검증이 가능하며, AI 학습과 로봇 제어, 환경 인식, Localization·Mapping 등 복합 기능의 통합 검증이 가능한 Virtual Twin 환경을 제공한다. 따라서 Isaac Sim은 물류 로봇의 실제 운용 환경을 가상에서 정밀하게 재현할 수 있는 시뮬레이터로 활용성이 매우 높다.

본 연구에서는 이러한 두 가지 접근 중 Isaac Sim 기반 Simulation-to-Reality Consistency(Sim2Real) 시뮬레이션 기법을 선택하였다. 그 이유는 본 연구의 목표가 단순한 데이터 생성이 아니라, 시뮬레이터에서 학습된 인공지능을 실제 로봇에 즉시 적용할 수 있는 Sim2Real 구조 구현에 있기 때문이다. Isaac Sim은 실제 물리 계수를 반영한 센서 모델과 로봇 동역학 기반 제어 구조를 포함하고 있어, 학습된 인공지능 모델이 추가 보정 없이 실제 AMR에서 안정적인 주행을 수행할 수 있도록 한다.

본 연구에서는 이러한 두 가지 접근 중 Isaac Sim 기반

Simulation-to-Reality Consistency(Sim2Real) 시뮬레이션 기법을 선택하였다. 그 이유는 본 연구의 목표가 단순한 데이터 생성이 아니라, 시뮬레이터에서 학습된 인공지능을 실제 로봇에 즉시 적용할 수 있는 Sim2Real 구조 구현에 있기 때문이다. Isaac Sim은 실제 물리 계수를 반영한 센서 모델과 로봇 동역학 기반 제어 구조를 포함하고 있어, 학습된 인공지능 모델이 추가 보정 없이 실제 AMR에서 안정적인 주행을 수행할 수 있도록 한다.

기존 연구에서는 시뮬레이터에서 학습한 정책(policy)을 실제 로봇에 전이하기 위해 튜닝이나 추가 보정(fine-tuning) 단계를 거치는 경우가 많았다<sup>9-11)</sup>. 이들 연구는 강화학습(RL) 기반 정책 학습을 통해 Isaac Sim 또는 Gazebo에서 학습된 제어 정책을 실제 로봇에 전이하였으나, 대부분 특정 환경이나 센서 구성에 최적화된 모델로 제한되었다. 반면 본 연구에서는 Isaac Sim 환경에서 학습된 인공지능 모델을 추가 보정 없이(zero-shot) 실제 자율주행 AMR에 적용하였으며, 맵 없이 LiDAR 데이터만으로 위치를 추정하는 CNN 회귀모델을 활용한 점에서 기존 RL 기반 Sim2Real 연구들과 차별성을 갖는다.

이러한 Virtual Twin 기반 학습 구조를 도입함으로써, 공장 완공 전부터 제어 알고리즘을 사전 검증할 수 있으며, 구축 이후에는 개발 기간과 비용을 획기적으로 절감할 수 있다. 따라서 본 연구에서는 Sim2Real 접근을 통한 Isaac Sim 기반 자율주행 인공지능의 실제 주행 구현 기술 개발을 목표로 하였으며, Virtual Twin 환경에서 학습된 인공지능 모델이 실제 AMR 시스템에서 추가 학습 없이 안정적으로 동작함을 실험적으로 검증하였다.

## 2. Virtual Twin 환경 구축 및 인공지능 학습 구조 설계

### 2.1 Virtual Twin 환경 구성

본 연구에서는 AMR의 자율주행 인공지능 학습을 위해 NVIDIA Isaac Sim을 기반으로 한 Virtual Twin 환경을 구축하였다. Isaac Sim은 Omniverse 플랫폼에서 동작하며, 실제 물리 계수(마찰계수, 중력, 관성 등)와 센서 모델을 정밀하게 구현할 수 있다. 이를 통해 현실과 유사한 공장 시뮬레이션을 구성하고, AI 학습 및 검증을 위한 가상 데이터를 안정적으로 확보할 수 있다.

Virtual Twin 모델은 실제 물류 공장의 평면도와 CAD 데이터를 기반으로 제작하였다. 이를 통해 공장 내부의 레이아웃, 통로 폭, 적재 공간, 충전 스테이션 등 주요 구조 요소를 동일하게 재현하였다. 각 요소는 Universal Scene Description(USD) 포맷으로 변환하여 Isaac Sim에 임포트하였으며, 충돌체(collision mesh)와 마찰 계수를 실제 바닥 재질(에폭시, 타일 등)에 맞게 설정하였다.

본 연구에서 구축한 Virtual Twin 환경은 실제 AMR과 동일한 센서 구성과 주행 공간을 기반으로 하여, 전체 시스템 구조를 실제

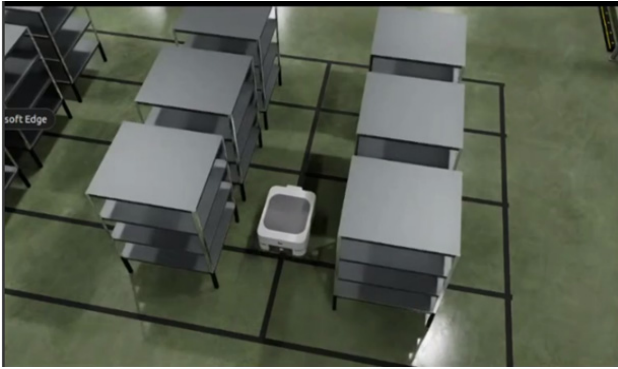


Fig. 1 Virtual twin environment implemented in NVIDIA Isaac Sim

계와 유사하게 재현하였다. 이러한 Virtual Twin 환경의 전체 구성은 Fig. 1에 나타내었다.

센서 구성은 실제 AMR과 동일하게 RGB 카메라, 2D LiDAR, IMU, 휠 엔코더로 구성하였다. 카메라 모델은 90° 시야각과 30fps로 설정하였으며, 센서 노이즈 모델을 추가하여 실제 환경에서의 왜곡·조도 변화를 모사하였다. LiDAR는 360° 회전형 2D 센서를 사용하여 거리 분해능 0.25°로 구현하였고, IMU 센서는 3축 가속도와 자이로 데이터를 ROS2 토픽 형태로 송신하도록 설정하였다. AMR의 주행 환경은 1/1 스케일로 모델링하였으며, 라인 주행 구간, 장애물 회피 구간, 랙(rack) 진입 구간으로 구성하였다. 각 구간은 물리 기반 조명(physically based lighting)과 RTX 실시간 렌더링을 적용하여 카메라 입력이 실제 환경의 조명 변화에 근접하도록 하였다.

## 2.2 ROS2 Communication Structure

본 연구에서는 ROS2 Humble Hawksbill 기반 환경에서 Isaac Sim과 ROS2 노드 간의 데이터 송수신을 구현하였다<sup>[12]</sup>. 시뮬레이터와 ROS2 간 통신에는 NVIDIA의 공식 ROS 확장 모듈인 isaac\_ros2\_bridge를 사용하였다. 이 브리지는 Omniverse Isaac Sim 내에서 ROS2 메시지 타입(sensor\_msgs, geometry\_msgs, nav\_msgs)을 직접 퍼블리시·서브스크라이브할 수 있도록 지원한다. 시뮬레이터 내부에서는 PhysX 엔진이 실제 AMR의 운동을 계산하며, isaac\_ros2\_bridge를 통해 ROS2 제어 명령(/cmd\_vel)이 Isaac Sim의 물리 엔진에 직접 전달된다. 또한 좌표계 일관성 유지를 위해 ROS2의 TF 트리(/map, /odom, /base\_link)를 동기화하였으며, AMR의 위치 정보는 nav\_msgs/Odometry 메시지를 통해 실시간으로 반환된다.

## 2.3 Dataset Generation for Line Tracking

Isaac Sim 환경에서 AMR의 인공지능 기반 라인 주행(line



Fig. 2 Sample images of instance-segmentation dataset for line tracking

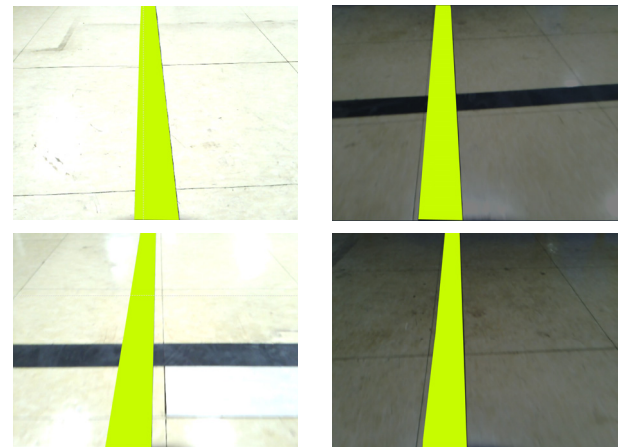


Fig. 3 Example of labeling results generated using Roboflow

following)을 구현하기 위해 가상 공장 바닥에 주행 경로 라인을 설정하고, 전방 카메라로 라인 영상을 수집하였다. 수집된 영상은 YOLOv8 Instance Segmentation 모델 학습용 데이터셋으로 활용하였다. 인공지능 학습 영상의 해상도는 640 × 640으로 설정하였으며, Instance Segmentation 기법을 적용하여 라인 영역과 배경을 픽셀 단위로 구분하였다. 이를 통해 모델이 라인의 형태, 폭 등의 특징을 명확히 학습할 수 있도록 구성하였다<sup>[13]</sup>.

Fig. 2에서와 같이 데이터셋은 조명 변화, 반사, 그림자, 노면 질감 등 다양한 환경 조건을 포함하여 모델의 일반화 성능을 향상시키도록 설계하였다. 총 1,000 장의 이미지를 수집하였으며, 이

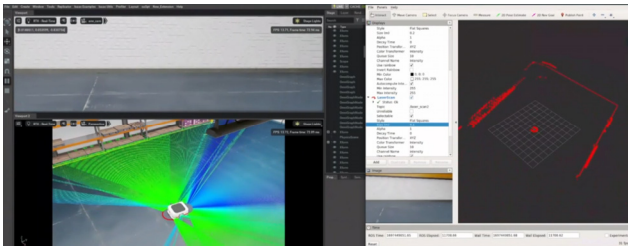


Fig. 4 Example of LiDAR dataset collected in Isaac Sim for localization training

중 800 장은 학습(training)에, 200 장은 검증(validation)에 사용하였다. 라벨링(labeling) 작업은 온라인 데이터셋 관리 도구인 Roboflow를 이용하여 수행하였다. 라벨링 결과는 Fig. 3에 나타내었다.

#### 2.4 Dataset Generation for Localization Learning

AMR의 자기 위치 추정(localization)을 위해 ROS2 환경에서 널리 사용되는 Localization 기법은 다음과 같다.

대표적인 방식으로는

- 1) AMCL (Adaptive Monte Carlo Localization),
- 2) EKF 기반 robot\_localization,
- 3) Graph-SLAM(slam\_toolbox, cartographer),
- 4) ICP 정합(Iterative Closest Point) 방식 등이 있다.

이들 방법은 ROS2에서 안정적으로 활용되고 있으나, 대부분 사전 지도(map)가 필요하며 초기 위치가 부정확하거나 환경이 동적으로 변하는 경우 성능이 급격히 저하되는 한계를 가진다. 특히 공장이 건설 중이거나 구조가 지속적으로 변경되는 산업 환경에서는 정확한 지도를 구축하기 어렵기 때문에, AMCL 및 SLAM 계열 알고리즘의 활용이 제한적이다. 이에 따라 본 연구에서는 2D LiDAR 데이터를 활용한 CNN 기반 회귀(Localization Regression) 모델을 적용하였다<sup>[15]</sup>. 제안된 방법은 사전 지도 없이 LiDAR 데이터만으로 현재 위치를 추정할 수 있으며, 기존의 확률적 방법(예: AMCL)에 비해 계산량이 적고, 초기 위치 정보가 없어도 실시간 추정이 가능한 장점을 가진다. 또한 Sim2Real 환경에서의 적용성이 높아 학습된 모델을 실제 AMR 시스템에 직접 이식할 수 있다.

Localization 학습을 위한 데이터 셋은 Fig. 4에서와 같이 Isaac Sim 환경에서 수집하였다. AMR을 다양한 위치와 자세로 이동시키면서, 360° 범위의 2D LiDAR 스캔 데이터를 획득하였다. 각 스캔은 1° 간격의 360개 거리값으로 구성되며, 동시에 로봇의 실제 위치(x, y)와 방향( $\theta$ )을 Ground Truth로 기록하였다. 최대 측정 거리는 15 m로 설정하였으며, 총 3,000세트의 데이터를 수집하였다. 이 중 2,700세트는 학습용, 300세트는 검증용으로 사용하

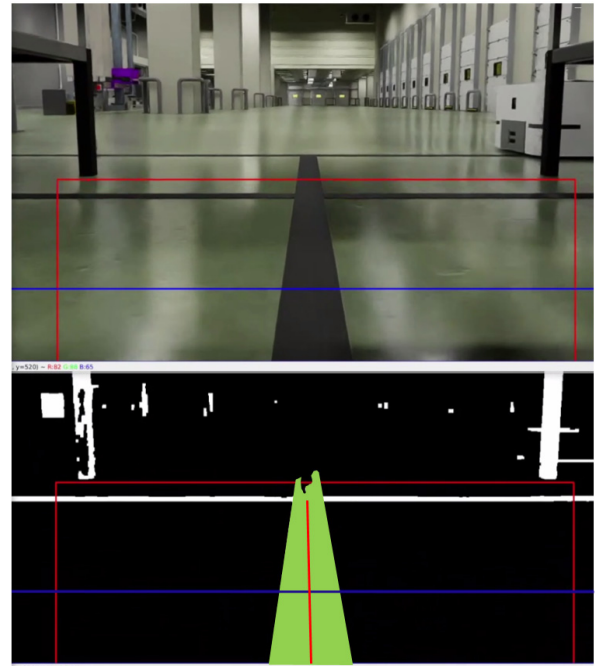


Fig. 5 YOLOv8-based line segmentation and centerline extraction result in Isaac Sim

였다. 수집된 LiDAR 데이터는 360차원 입력 벡터로 정규화하여 Convolutional Neural Network(CNN) 기반 회귀 모델의 입력으로 사용하였다.

### 3. Sim2Real 자율주행을 위한 인공지능 구현

#### 3.1 YOLOv8을 이용한 라인 중심 검출

라인 추출을 위해서는 YOLOv8 Instance Segmentation 모델을 사용하였다. Isaac Sim 환경에서 수집된 라인 주행 영상(640 × 640)을 학습 데이터로 사용하였으며, 학습은 Adam Optimizer (learning rate = 0.001), Batch Size 16, Epoch 100의 조건에서 수행하였다.

Fig. 5는 YOLOv8 Instance Segmentation을 이용하여 라인 영역을 검출하고 중심선을 추출하는 과정을 시각적으로 보여준다. 상단 이미지는 Isaac Sim에서 수집된 원본 RGB 영상이며, 하단 이미지는 세그멘테이션을 통해 추출된 라인 영역(녹색)과 그로부터 생성된 중심선(centerline)이 표시된 결과이다.

라인 중심선은 라인으로 분리된 픽셀 영역 내에서 가로 방향(row-wise)으로 유효 픽셀의 중심 좌표를 계산한 뒤, 각 행에서 얻어진 중심점을 연결하는 방식으로 생성하였다. 이러한 방법을 적용함으로써 시뮬레이션 환경뿐 아니라 실제 AMR 카메라 영상에서도 안정적이고 일관된 라인 중심 검출 성능을 얻을 수 있었다.

모델은 다양한 조명 및 반사 조건에서도 라인을 안정적으로 검

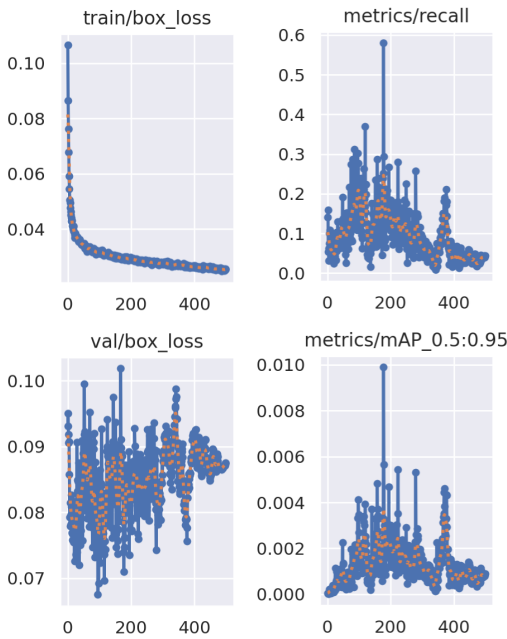


Fig. 6 Training loss curves of YOLOv8 model

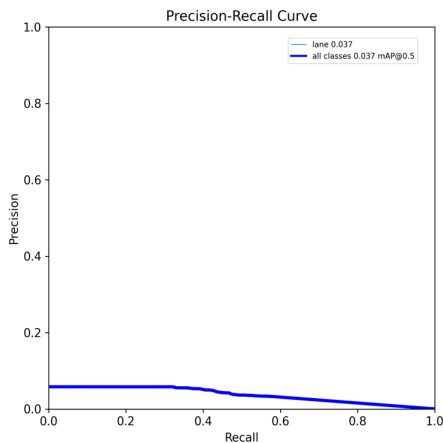


Fig. 7 Precision-recall curve of YOLOv8 segmentation model

출하였으며, 중심선은 후처리 과정을 통해 실시간으로 계산되어 조향각 산출에 활용되었다. 이 중심 정보는 주행 제어 알고리즘에서 실시간 입력으로 사용되어, AMR이 주행 중 라인 중심을 안정적으로 추종할 수 있도록 하였다. 학습 로그 분석 결과, 훈련 100 epoch 동안 손실 함수(loss)는 지속적으로 감소하여 안정적으로 수렴하였다. Fig. 6은 학습 과정에서의 손실 변화 곡선을 보여주며, 훈련 손실(train/box\_loss, train/obj\_loss)과 검증 손실(val/box\_loss, val/obj\_loss)이 모두 일정 수준에서 수렴함을 확인할 수 있다.

최종 성능 지표는 mAP@0.5 = 0.972, Precision = 0.985, Recall = 0.963으로 나타났다. 또한 검증 데이터셋의 결과를 비교한 결과, 세그멘테이션 경계의 일관성과 중심선 검출의 정확도가 높았다. 모델의 정량적 성능 평가는 Precision-Recall 관계를 기준으로 수행하였다. Fig. 7은 검증 데이터셋에 대한 Precision-Recall Curve

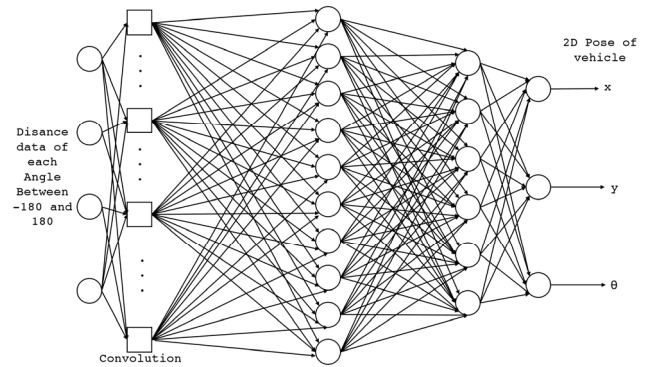


Fig. 8 Proposed CNN model based localization

Table 1 The training results of various CNNs

CNN structure (input-hidden layers-output)	R <sub>2</sub> (validation accuracy)	RMS position error [m]	RMS heading error [°]
360-128-64-32-32-16-3	0.941	0.034	2.097
360-256-128-64-32-16-3	0.967	0.029	1.872
360-256-128-64-64-32-3	0.974	0.026	1.815

를 나타내며, 모델의 평균 정확도(mAP@0.5)는 0.972로 높은 검출 성능을 보였다. 이는 YOLOv8 모델이 Isaac Sim에서 생성된 합성 영상에서도 라인 영역을 안정적으로 인식함을 의미한다.

### 3.2 CNN 기반 Localization 학습 및 결과

AMR의 자기 위치 추정(localization)을 위한 CNN 기반 회귀 모델은 Isaac Sim 환경에서 AMR을 다양한 위치와 자세에서 이동 시키며, 360° 회전 범위를 갖는 2D LiDAR 센서로부터 거리 정보를 획득하였다. Localization CNN 모델은 Fig. 8과 같이 1차원 합성곱 계층(1D convolutional layers)을 이용하여 LiDAR 거리 벡터의 공간적 패턴을 추출하도록 설계되었다. 입력으로는 360차원의 LiDAR 거리 데이터를 사용하였으며, 두 개의 1D 합성곱 계층과 풀링 계층(MaxPooling)을 통해 지역적 거리 분포 특징을 추출하였다. 이후 완전연결층(fully connected layers)을 거쳐 로봇의 위치(x, y)와 자세각(theta)을 출력하는 회귀 구조로 구성하였다. 활성화 함수는 ReLU를 사용하였으며, 손실 함수는 Mean Squared Error(MSE)로 정의하였다

학습은 Adam Optimizer(learning rate = 0.0009), Batch Size = 32, 최대 Epoch = 10,000으로 수행하였으며, Early Stopping을 통해 과적합을 방지하였다. 손실 함수는 평균제곱오차(mean squared error, MSE)로 정의하였다. Table 1은 다양한 CNN 구조에 따른 성능을 비교한 결과를 나타낸다. 최적 모델인 (360-256-128-64-64-32-3)은 RMS 위치 오차 0.026 m, 자세각 오차 1.8°로 가장 우수한 성능을 보였으며, 본 연구의 최종 Localization 모



Fig. 9 Real-world warehouse test field and AMR platform used for Sim2Real validation

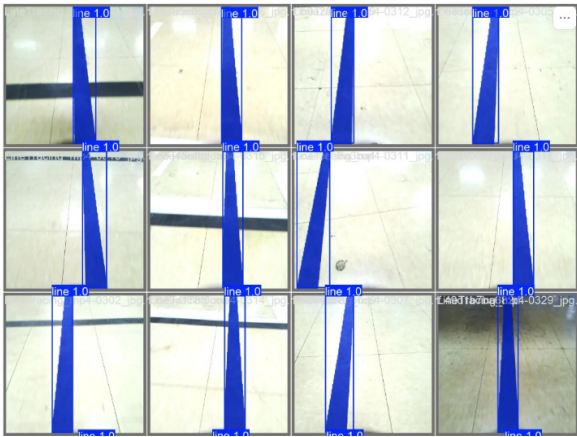


Fig. 10 Segmentation results of YOLOv8 model in real-world line-following experiment

델로 채택되었다.

#### 4. Sim2Real 검증

##### 4.1 실험 환경 구성

연구의 실험은 Virtual Twin 환경에서 학습된 인공지능 모델의 실제 적용 가능성을 검증하기 위하여 실제 현장(real-world test field)에서 수행하였다. Fig. 9는 실험이 진행된 경기도 소재 C사의 물류창고 환경과 사용된 AMR 플랫폼의 외관을 나타낸다. 실제 실험에 사용된 AMR은 차동구동(differential-drive) 방식으로 설계되었으며, 전방 RGB 카메라와 전·후방 2D LiDAR 등 센서는 실험의 정확도 확보를 위해 실제 장비 사양에 맞게 구성하였다. RGB 카메라는 1280 × 720 해상도, 30fps로 설정되었고, LiDAR는 SOSLAB GL-320 모델을 사용하여 수평 FOV 180°, 감지 범위 20 m, 각 분해능 0.18°, 40 Hz 스캔 속도로 구성하였다.

ROS2 Humble 환경에서 동작하는 제어 노드는 YOLOv8 기반 라인 인식 결과를 수신하여 Cross-Track Error(CTE)를 계산하였으며, CNN Localization 모델은 LiDAR 벡터를 입력받아 AMR

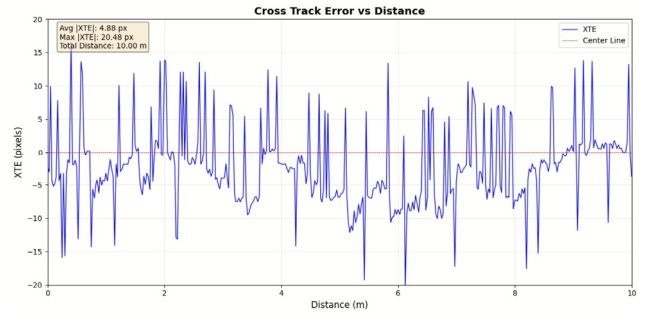


Fig. 11 Line-following experimental results of the AMR in a 10 m straight track

의 위치(x, y)와 자세각(θ)을 실시간으로 추정하였다. 학습된 두 AI 모델은 실험 중 추가 튜닝이나 파라미터 보정 없이 실제 AMR에 적용되었으며, 조명 밝기 및 반사 환경을 다양하게 변경하여 모델의 강건성을 검증하였다.

##### 4.2 라인 추종 실험

라인 인식 실험은 실제 AMR에 장착된 RGB 카메라(해상도 1280 × 720)로 촬영된 영상에서 640 × 640 영역을 중심부 기준으로 crop하여 YOLOv8 Instance Segmentation 모델의 입력으로 사용하였다<sup>[13-14]</sup>. 모델은 영상 내에서 라인 영역을 픽셀 단위로 분리(segmentation)하고, 세그멘테이션 결과로부터 라인의 중심선을 계산하였다. 다양한 조명 조건(밝음, 반사, 그림자)에서 실제 주행 중 촬영된 1,658장의 영상을 분석한 결과, YOLOv8 모델은 100%의 정확도로 라인 영역을 검출하였다.

Fig. 10은 실제 환경에서 촬영된 이미지의 세그멘테이션 결과를 나타낸 것으로, 모델이 조명 변화나 바닥 반사 조건에서도 일관되게 라인 형태를 인식함을 보여준다.

라인 추종 실험은 세그멘테이션 결과로부터 라인의 중심 좌표( $x_c$ )와 영상의 중심( $x_m$ )과의 차이를 식(1)과 같이 Cross-Track Error (CTE)로 정의하여 AMR이 주행 중 라인 중심을 얼마나 정확히 추종하는지를 평가하였다.

$$CTE[i] = x_c[i] - x_m[i] \quad (1)$$

여기서  $i$ 는 CTE는 프레임 index이다. CTE는 640 × 640 해상도 영상에서 픽셀 단위 기준으로 분석하였다. 영상의 중심( $x_m, y_m$ ) = (320, 320)을 기준으로 하여 CTE가 0에 가까울수록 차량이 라인 중앙을 정확히 주행하는 것을 의미한다.

라인의 주행은 PD 제어를 사용하여 AMR의 좌우 바퀴 속도차이를 조정하는 방법을 사용하였다. PD 제어기의 파라미터는 여러 차례의 실험을 통하여 단계적으로 조정하였으며, 최종적으로  $K_p_{vision}$ 은 0.05,  $K_d_{vision}$ 은 0.12로 설정하였다. Fig. 11은 10 m

**Table 2 LiDAR range accuracy validation(0.5-10.0 m)**

Distance (m)	Samples (N)	Mean range (m)	Bias (m)	Std (m)	RMSE (m)
0.5	50	0.498	-0.002	0.006	0.006
1.0	50	0.99	-0.01	0.007	0.012
1.5	50	1.501	0.001	0.008	0.008
2.0	50	1.989	-0.011	0.01	0.015
2.5	50	2.503	0.003	0.01	0.01
3.0	50	2.995	-0.005	0.009	0.011
3.5	50	3.499	-0.001	0.009	0.009
4.0	50	3.989	-0.011	0.011	0.016
4.5	50	4.495	-0.005	0.01	0.011
5.0	50	4.98	-0.02	0.012	0.023
5.5	50	5.486	-0.014	0.011	0.018
6.0	50	5.991	-0.009	0.01	0.013
6.5	50	6.493	-0.007	0.011	0.013
7.0	50	6.98	-0.02	0.012	0.024
7.5	50	7.491	-0.009	0.01	0.013
8.0	50	7.982	-0.018	0.012	0.021
8.5	50	8.487	-0.013	0.012	0.018
9.0	50	8.982	-0.018	0.012	0.022
9.5	50	9.486	-0.014	0.011	0.018
10.0	50	9.985	-0.015	0.012	0.019

라인 주행 결과를 나타내며, 실험 결과 평균 절대 오차(Avg |CTE|)는 4.88 pixel, 최대 오차(Max |CTE|)는 20.48 pixel로 측정되었다. 이는 실제 환경에서 YOLOv8 기반 영상 인식과 PD 제어기의 조합이 안정적인 라인 추종 성능을 구현함을 보여준다.

**4.3 라이다 위치 인식 실험**

실험은 Fig. 9와 같이 물류창고 내부의 주행 구역을 2.5 m × 2.5 m 크기의 격자(grid)로 구분하고, 각 교차점에 대한 실제 좌표(ground truth)를 기준으로 예측 좌표를 비교하였다. 먼저 AMR에 장착된 LiDAR의 거리 측정 성능을 검증하기 위한 실험을 수행하였으며, 10 m 거리 구간에서 정확도는 2 cm, 정밀도(표준편차)는 1 cm 이하로 나타났다. Table 2는 실험 결과의 일부를 제시한다.

비교 대상으로는 ROS2에서 많이 사용되는 Adaptive Monte Carlo Localization(AMCL)과 Graph-SLAM 기반 방식인 SLAM Toolbox를 선택하였다. 두 방법 모두 지도 기반(scan-matching 기반 또는 particle-filter 기반) Localization 기법으로, 초기 위치(initial pose)를 필요로 하며, 연속적인 오도메트리 및 누적 관측을 바탕으로 상대 위치 변화를 추정하는 구조를 갖는다. 반면, 본 연구에서 제안된 CNN Localization 모델은 단일 LiDAR 스캔(360D vector)만을 입력으로 절대 위치(x, y, θ)를 직접 회귀(direct regression)하는 map-free global localization

**Table 3 Sample results of the proposed CNN-based localization model**

GT(x, y) [m]	Proposed CNN (x, y) [m]	CNN position error [m]	CNN heading error [°]
(2.50,0.00)	(2.52, 0.01)	0.022	2.16
(2.50,2.50)	(2.53, 2.49)	0.032	-2.08
(0.00,2.50)	(-0.01, 2.52)	0.022	-1.23
(1.25,1.25)	(1.26, 1.22)	0.032	1.26
(3.00,1.25)	(3.02, 1.21)	0.045	-2.28
(1.25,3.00)	(1.27, 2.99)	0.022	-1.37
(3.00,3.00)	(2.99, 3.02)	0.022	-2.18

**Table 4 Sample results of ROS2 AMCL-based localization**

GT(x, y) [m]	AMCL (x, y) [m]	AMCL Position Error [m]	AMCL Heading Error[°]
(2.50,0.00)	(2.47, -0.04)	0.050	2.10
(2.50,2.50)	(2.55, 2.57)	0.086	1.24
(0.00,2.50)	(-0.02, 2.60)	0.102	2.06
(1.25,1.25)	(1.29, 1.33)	0.089	2.75
(3.00,1.25)	(2.93, 1.33)	0.106	-1.94
(1.25,3.00)	(1.32, 3.05)	0.087	2.48
(3.00,3.00)	(2.95, 2.92)	0.094	-2.28

**Table 5 Sample results of ROS2 SLAM toolbox-based localization**

GT(x, y) [m]	SLAM toolbox (x, y) [m]	SLAM toolbox position error [m]	SLAM toolbox Heading error [°]
(2.50,0.00)	(2.45, -0.03)	0.054	2.41
(2.50,2.50)	(2.57, 2.58)	0.108	2.01
(0.00,2.50)	(0.03, 2.63)	0.134	-2.81
(1.25,1.25)	(1.31, 1.34)	0.108	2.84
(3.00,1.25)	(2.92, 1.36)	0.132	-2.51
(1.25,3.00)	(1.33, 3.08)	0.112	1.95
(3.00,3.00)	(2.94, 2.88)	0.126	-2.13

**Table 6 Comparative summary of localization accuracy between the proposed CNN model, ROS2 AMCL and SLAMTOOLBOX**

Evaluation parameter	Proposed CNN model	ROS2 AMCL	ROS2 SLAM toolbox
RMS X error [m]	0.018	0.059	0.084
RMS Y error [m]	0.021	0.052	0.089
RMS position error [m]	0.028	0.088	0.112
RMS heading error [°]	1.79	2.12	2.36

방식으로, 기존 방법과 차별성을 가진다.

Table 3은 CNN Localization 모델의 예측 결과 일부를 나타내며, 모든 측정 위치에서 중심 좌표 오차는 ±5 cm 이내, 자세각 오차는 평균 2° 이하로 나타났다. Table 4는 동일 좌표에서 측정된 AMCL

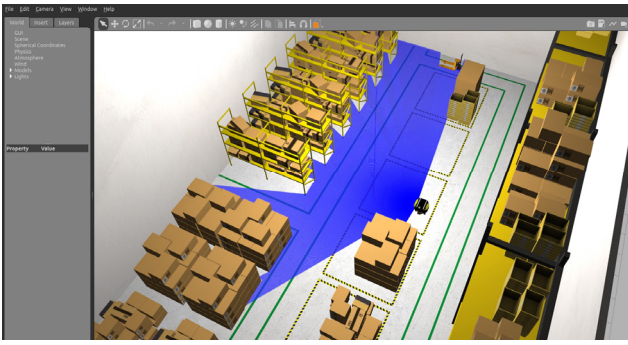


Fig. 12 Extended validation scenario constructed in the Gazebo simulation to assess the generalization ability of the proposed CNN-based LiDAR localization model

Table 7 Localization accuracy of the proposed CNN model evaluated at unseen robot poses in the Gazebo simulation environment

Gazebo (x, y) [m]	Proposed CNN (x, y) [m]	CNN position Error [m]	CNN heading Error [°]
(12.50,0.00)	(12.51, -0.01)	0.014	-1.53
(12.50,2.50)	(12.49, 2.53)	0.032	2.08
(10.00,2.50)	(10.01, 2.47)	0.032	2.02
(11.25,1.25)	(11.25, 1.23)	0.020	-1.41
(13.00,1.25)	(12.98, 1.24)	0.022	1.97
(11.25,3.00)	(11.24, 3.01)	0.014	1.75
(13.00,3.00)	(13.01, 3.02)	0.022	2.18

결과를 제시하였으며, Table 5는 SLAM Toolbox(Graph-SLAM)의 예측 성능을 나타낸다. AMCL과 SLAM Toolbox는 초기 위치 설정을 제공하지 않으면 정상적으로 수렴하지 못하는 구조적 제약을 지니는 반면, 제안된 CNN 모델은 초기 포즈를 제공하지 않고도 일관된 절대 좌표를 추정할 수 있다.

세 방법의 RMS 위치 오차 및 자세각 오차를 요약한 결과는 Table 6에 나타내었다. CNN Localization 모델은 RMS Position Error 0.028 m로 AMCL(0.088 m)과 SLAM Toolbox(0.110 m) 대비 더 높은 정확도를 보였으며, Heading Error 또한 가장 낮았다. 이는 Virtual Twin 환경에서 다양한 위치·자세 조합을 포함한 학습 데이터셋을 구성함으로써 전역 위치 일반화 능력을 확보한 결과이다.

본 연구에서 제안한 CNN 기반 LiDAR 위치 인식 모델의 확장성(scalability)과 환경 일반화 능력(generalization ability)을 검증하기 위하여, Fig. 12와 같이 Gazebo 시뮬레이터 환경에서 추가 실험을 수행하였다. Gazebo 환경은 Isaac Sim과 센서 노이즈 특성 및 반사 패턴이 상이하므로, 제안된 모델이 특정 시뮬레이터에 과적합되지 않고 다양한 환경에서도 안정적으로 동작하는지 평가하는 데 적합하다.

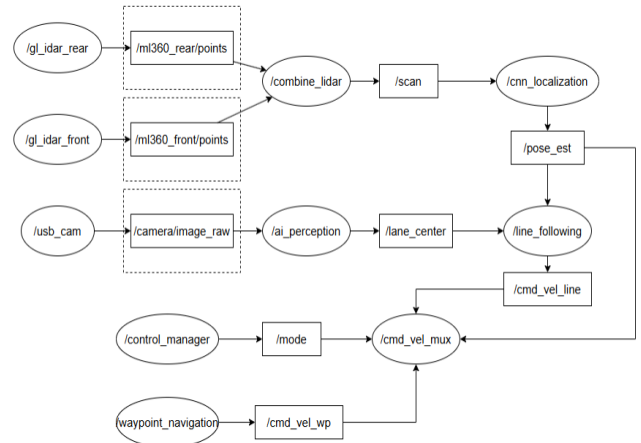


Fig. 13 ROS2-based integrated driving architecture for autonomous mobile robot

학습 데이터는 Gazebo 환경에서 임의로 설정한 200개 위치에서 로봇의 자세를 2° 간격으로 회전시키며 총 3600개의 라이다 데이터를 수집하여 구성하였다. 학습에는 360-256-128-64-64-32-3 구조의 신경망을 사용하였으며, 최종적으로 R<sup>2</sup> (validation accuracy) = 0.96, RMS Position Error = 0.021 m, RMS Heading Error = 2.114°를 얻었다.

학습에 포함되지 않은 새로운 로봇 위치에서 모델을 평가한 결과는 Table 7에 나타내었다. 모든 테스트 위치에서 위치 오차는 최대 0.032 m(3.2 cm) 이내로 나타났으며, 자세각 오차는 2° 내외로 유지되었다. 이는 제안된 CNN Localization 모델이 Gazebo와 같은 다른 시뮬레이션 환경에서도 안정적으로 위치를 추정할 수 있음을 의미한다.

#### 4.4 ROS2 주행 구조

본 연구에서 구현한 ROS2 기반 주행 시스템은 Fig. 13과 같이 관제 노드(control manager), AI 인식 노드(AI perception), 라인 주행 노드(line-following), 웨이포인트 주행 노드(waypoint navigation), AI LiDAR 위치 인식 노드(CNN localization), 그리고 명령 병합 노드(cmd\_vel\_mux) 로 구성된다.

관제 노드는 전체 주행 모드(라인·웨이포인트·자율)를 제어하고, 시스템 상태 및 비상 정지를 관리한다.

AI 인식 노드는 RGB 카메라 영상을 입력받아 YOLOv8 기반으로 라인 중심을 검출하고 /lane\_center를 퍼블리시한다.

라인 주행 노드는 이 데이터를 이용해 Cross-Track Error(CTE)를 계산하고 조향 명령(/cmd\_vel\_line)을 출력한다. 웨이포인트 주행 노드는 사전 정의된 목표점 리스트를 기반으로 CNN Localization 노드에서 제공되는 위치 정보를 이용해 /cmd\_vel\_wp를 생성한다. AI LiDAR 위치 인식 노드는 2D LiDAR 입력으로부터 로봇의

위치와 자세(x, y,  $\theta$ )를 회귀 예측하여 /pose\_est를 퍼블리시하며, 이는 모든 주행 노드의 공통 참조 정보로 사용된다. 마지막으로, cmd\_vel\_mux 노드는 여러 주행 노드에서 생성된 속도 명령을 통합·우선순위화하여 /cmd\_vel 토픽으로 모터 드라이버에 전달한다.

이와 같은 ROS2 기반 통합 구조를 통해 Virtual Twin 환경에서 학습된 인공지능 모델이 실제 AMR 제어 시스템에 자연스럽게 연동되며, 다양한 주행 모드(line following, waypoint navigation 등)를 실시간으로 전환·관리할 수 있다.

## 5. 결론

본 연구에서는 NVIDIA Isaac Sim 기반 Virtual Twin 환경을 활용하여 Autonomous Mobile Robot(AMR)의 자율주행 인공지능을 학습하고, 실제 환경과 동일한 조건에서 검증할 수 있는 Sim2Real 기반 학습 구조를 제안하였다. 제안된 방법은 단순 시뮬레이션 수준을 넘어, 공장 설계-AI 학습-자율주행 검증을 통합한 지능형 가상 공장(virtual factory)을 구현 목표로 하였다.

제안된 방법은 Virtual Twin 환경에서 YOLOv8 Instance Segmentation을 이용한 라인 인식(line following)과 CNN 기반 Localization Regression을 이용한 위치 추정을 결합하여 AI 모델이 실제 로봇 제어로 직접 연결될 수 있도록 하였다. 이를 통해 기존의 지도 기반(localization with map) 방식이 갖는 초기 위치 의존성과 환경 변경에 대한 취약성을 개선할 수 있었다.

본 연구를 통하여, Virtual Twin 환경에서 생성된 합성 데이터를 통해 인공지능 모델을 사전 학습·검증함으로써 개발 기간과 비용을 획기적으로 단축할 수 있음을 확인하였다. 또한 Isaac Sim 기반의 고정밀 물리 시뮬레이션을 통해 실제 센서 노이즈, 조명 변화, 반사 등 다양한 현실적 요소를 반영함으로써 실세계 수준의 AI 학습 환경을 제공하였다. 이로써 실제 데이터 수집과 주행 실험에 필요한 비용과 인력 투입을 크게 줄이고, 동시에 AI 모델의 학습 효율성과 신뢰성을 향상시킬 수 있었다. 마지막으로, ROS2 기반 제어 구조를 연동하여 Sim2Real 일관성을 확보함으로써, 가상 환경에서 학습된 AI가 별도의 추가 조정 없이 제한된 환경이지만 AMR에 적용 가능함을 확인하였다. 복잡한 환경에서는 실제 운영하면서 얻은 데이터를 보조적으로 학습할 경우 학습 시간과 비용을 줄이면서 적용 가능할 것으로 판단된다.

향후 연구에서는 본 구조를 확장하여 랙 도킹(rack docking), 충전 도킹(charging docking) 등 AMR의 복합 작업 환경을 포함한 AI 기반 Full-Stack 자율주행 시스템 개발을 할 계획이다. 이를 통해 설계·생산·운영이 통합된 산업형 지능 공장(industrial physical AI) 구현을 산업용 로봇·AGV·서비스 로봇 등으로 확장함으로써 AI

기반 지능형 공장구현에 기여할 것으로 기대된다.

## References

- [1] Savushkin, N., 2024, viewed 30 November 2025, Warehouse Automation in logistics: Case study of Amazon and Ocado, Bachelor Thesis, Metropolia University of Applied Sciences, Finland. <<https://www.theseus.fi/handle/10024/857682>>.
- [2] Aggarwal, H., Malhotra, P., 2026, Smart Warehousing: Automation and Real-Time Monitoring, IGI Global, PA, USA., <https://doi.org/10.4018/979-8-3693-9856-2.ch009>.
- [3] Nassif, J., Tekli, J., Kamradt, M., 2024, Synthetic Data: Revolutionizing the Industrial Metaverse, Springer, Cham, Switzerland, 33-74, [https://doi.org/10.1007/978-3-031-47560-3\\_3](https://doi.org/10.1007/978-3-031-47560-3_3).
- [4] Lüdemann-Ravit, B., Heieck, F., 2023, viewed 30 November 2025, Architecture Options to Orchestrate Digital Twins in an Industrial Metaverse for the Predictive Production with AI Methods, PATTERNS 2023: The Fifteenth International Conference on Pervasive Patterns and Applications, 6-11, <[https://personales.upv.es/thinkmind/dl/conferences/patterns/patterns\\_2023/patterns\\_2023\\_1\\_20\\_78002.pdf](https://personales.upv.es/thinkmind/dl/conferences/patterns/patterns_2023/patterns_2023_1_20_78002.pdf)>.
- [5] Nambiar, S., Paul, R. C., Ikechukwu, O. C., Jonsson, M., Tarkian, M., 2025, Digital Twin-Enabled Adaptive Robotics: Leveraging Large Language Models in Isaac Sim for Unstructured Environments, Machines, 13:7 620, <https://doi.org/10.3390/machines13070620>.
- [6] Myrvold, J. U., 2023, Digital Twin of the KUKA KMR iiwa: Developing a Simulation Framework Using ROS 2 and NVIDIA Isaac Sim, Master's Thesis, Norwegian University of Science and Technology (NTNU), Norway, <<https://nva.sikt.no/registration/0198e906fe11-28c7fb72-80d2-4479-a26e-df6ccb634140>>.
- [7] Li, X., Park, J., Reberg-Horton, C., Mirsky, S., Lobaton, E., Xiang, L., 2024, Photorealistic Arm Robot Simulation for 3D Plant Reconstruction and Automatic Annotation using Unreal Engine 5, Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 5480-5488, <https://doi.org/10.1109/CVPRW63382.2024.00557>.
- [8] Samaylal, S., 2024, Real-Time Digital Twin with Reinforcement Learning for Industrial Manipulator Applications: Designing of ABB Yumi Robot Digital Twin in NVIDIA Isaac Gym and Building RL Application, Master's Thesis, Tampere University, Finland, <<https://trepo.tuni.fi/bitstream/handle/10024/161966>>.

- SamaylalSandeep.pdf?sequence=2>
- [9] Salimpour, S., Peña-Queralt, J., Paez-Granados, D., Heikkonen, J., Westerlund, T., 2025, Sim-to-Real Transfer for Mobile Robots with Reinforcement Learning: from NVIDIA Isaac Sim to Gazebo and Real ROS 2 Robots, arXiv preprint, arXiv:2501.02902, <https://doi.org/10.48550/arXiv.2501.02902>.
- [10] Albardaner, J., San Miguel, A., García, N., Dalmau-Moreno, M., 2024, Sim-to-Real Gap in RL: Use Case with TIAGo and Isaac Sim/Gym, arXiv preprint, arXiv:2403.07091, <https://doi.org/10.48550/arXiv.2403.07091>.
- [11] Ampuero, G. C., Hermosilla, G., Varas, G., Clark, M. T., 2025, Deep Reinforcement Learning for Sim-to-Real Robot Navigation with a Minimal Sensor Suite for Beach-Cleaning Applications, *Appl. Sci.*, 15:19 10719, <https://doi.org/10.3390/app151910719>.
- [12] Sterk-Hansen, A., Saghaug, B. H. H., Hagen, D., Aftab, M. F., 2023, A ROS 2 and TwinCAT Based Digital Twin Framework for Mechatronics Systems, 11th International Conference on Control, Mechatronics and Automation (ICCA), 485-490, <https://doi.org/10.1109/ICCA59762.2023.10374978>.
- [13] Ko, K. W., 2025, Development of a Mobility AI Education Platform and Its Application to First-Year Students, *Asia-pacific Journal of Convergent Research Interchange*, 11:9 25-41, <https://doi.org/10.47116/apjcri.2025.09.03>.
- [14] Lee, S., Yoo, K.-h., Kang, Y.-g., Kim, J.-m., Lee, C. S., 2023, Strawberry-harvesting Robot System with Object Detection Based on YOLO, *J. Korean Soc. Manuf. Technol. Eng.*, 32:2 101-108, <https://doi.org/10.7735/ksmte.2023.32.2.101>.
- [15] Choi, T. Y., Son, M., Shin, M. S., Lim, Y. H. B., Jo, Y. H., Ko, K. W., 2024, Development of AI Model for Lidar Localization for FITENTH, 24th International Conference on Control, Automation and Systems (ICCAS), 104-105, <https://doi.org/10.23919/ICCAS63016.2024.10773105>.



**Kuk Won Ko**

Professor in the Department of Future Mobility, Halla University. His research interests include AI-based Autonomous Mobility, Robotics, Sensor Fusion, and Intelligent Vehicle Systems.

E-mail: [kuksauto@hanmail.net](mailto:kuksauto@hanmail.net)