JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 11, pp. 151-165, November 2026
https://doi.org/10.9708/jksci.2026.31.11.151

EdgeGround-VLA: A Lightweight Model Fusing Heterogeneous Grounding Encoders for Goal-Directed Navigation in Wheeled Robots

Min-Woo Lee*, In-Yeop Choi**

*Student, Dept. of Applied Artificial Intelligence Engineering, Kangnam University, Gyeonggi-do, Korea
**Professor, Dept. of Computer Science, Kangnam University, Gyeonggi-do, Korea

[Abstract]

Recent research in mobile robot navigation has been actively progressing toward language-instructed navigation using large-scale Vision-Language Models (VLMs), as exemplified by Mobility VLA. However, such VLM-based approaches require substantial computation and depend on pre-built infrastructure, which limits their applicability to resource-constrained mobile robot platforms. To address this problem, this paper proposes EdgeGround-VLA, a lightweight on-device navigation model. The proposed model is designed to fuse the representations of two vision encoders with distinct grounding characteristics and combine them with a lightweight action head, converting a goal instruction into a 3-degree-of-freedom (3-DoF) movement command. EdgeGround-VLA is designed such that only 1.128M parameters are trained out of a total of 459M parameters. And it operates on-device without any cloud connectivity or pre-built infrastructure. Real-world experiments were conducted by deploying the model on a robot equipped with an NVIDIA Jetson Orin NX, achieving an average goal-reaching success rate of 92.0%. In addition, we decomposed the entire inference pipeline of the proposed model by component to measure latency and analyze the bottleneck sections, thereby identifying the components that should be prioritized for optimization when scaling to ultra-low-resource platforms such as the Raspberry Pi. These results demonstrate the potential for implementing on-device VLA-based goal-directed navigation on ultra-low-resource mobile robots while reducing dependency on pre-built infrastructure.

▸ **Key words**: On-Device AI, VLA, Open-Vocabulary Object Detection, Omnidirectional Mobile Robot, Lightweight Deep Learning

• First Author: Min-Woo Lee, Corresponding Author: In-Yeop Choi
*Min-Woo Lee (minwool0357@gmail.com), Dept. of Applied Artificial Intelligence Engineering, Kangnam University
**In-Yeop Choi (billychoi@kangnam.ac.kr), Dept. of Computer Science, Kangnam University
• Received: 2026. 08. 26, Revised: 2026. 10. 01, Accepted: 2026. 10. 19.

Copyright © 2026 The Korea Society of Computer and Information
http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945
www.kci.go.kr

[요 약]

최근 모바일 로봇 내비게이션 분야에서는 Mobility VLA와 같이 대규모 Vision-Language Model(VLM)을 활용한 언어 지시 기반 내비게이션 연구가 활발히 진행되고 있다. 그러나 이러한 VLM 기반 추론은 높은 연산량을 요구하며, 사전 구축된 인프라에 대한 의존성으로 인해 연산 자원이 제한된 모바일 로봇 플랫폼에 적용하는 데 한계가 있다. 이러한 문제를 해결하기 위해 본 논문에서는 경량의 온디바이스 내비게이션 모델인 EdgeGround-VLA를 제안한다. 제안 모델은 서로 다른 그라운딩(grounding) 특성을 갖는 두 비전 인코더의 표현을 융합하고 경량 액션 헤드를 결합하여, 목표 지시를 3-DoF 이동 명령으로 변환하도록 설계하였다. 제안 모델은 전체 459M개의 파라미터 중 1.128M개의 파라미터만 학습하는 구조로 설계되었으며, 클라우드 연결이나 사전 구축된 인프라 없이 온디바이스에서 동작한다. NVIDIA Jetson Orin NX가 탑재된 로봇에 배포하여 실기 실험을 수행한 결과, 평균 92.0%의 목표 도달 성공률을 달성하였다. 또한 제안 모델의 전체 추론 파이프라인을 구성 요소별로 분해하여 지연시간을 측정하고 병목 구간을 분석하였다. 이를 통해 Raspberry Pi와 같은 초저자원 플랫폼으로 확장할 경우 우선적으로 최적화해야 할 구성 요소를 식별하였다. 이러한 결과는 사전 구축된 인프라에 대한 의존성을 줄이면서 초저자원 모바일 로봇에서 온디바이스 VLA 기반 목표 지향 내비게이션을 구현할 수 있는 가능성을 보여준다.

▸ **주제어**: 온디바이스AI, 비전-언어-액션, 개방형 어휘 객체 탐지, 모바일 로봇, 경량 딥러닝

I. Introduction

모바일 로봇이 사람의 언어 지시를 이해하고 실내 환경에서 목표 지점까지 이동하는 내비게이션 기술은 서비스 로봇, 배송 로봇, 실내 안내 로봇 등 다양한 응용 분야에서 핵심적인 기술로 자리 잡고 있다. 최근에는 Mobility VLA[1]와 같이 긴 문맥(long-context)의 추론 능력이 있는 VLM(Vision- Language Model)과 토폴로지 그래프를 결합한 계층적 VLA(Vision-Language-Action) 내비게이션 정책이 제안되었다. 이러한 모델은 복잡한 추론 및 멀티모달 지시에 대해 각각 86%와 90%의 end-to-end 성공률을 보고하였다[1].

그러나 이러한 대규모 VLM 기반 접근은 다음과 같은 배포상의 제약을 동반한다. 첫째, Mobility VLA의 대규모 VLM 추론은 클라우드 API 기반으로 수행되며, 목표 프레임 탐색에 약10~30초의 지연이 발생한다. 둘째, 목표 프레임을 탐색하기 위해 사전에 수집된 데모 투어 영상과 이를 기반으로 오프라인으로 구축한 토폴로지 그래프를 필요로 한다. 이러한 제약은 저비용 모바일 로봇 플랫폼이나 네트워크 연결이 불안정한 환경에서의 실용적 배포를 어렵게 만든다.

본 논문에서는 이러한 배포 제약을 해결하기 위해, 클라우드 인프라나 사전 구축된 투어 영상 및 맵에 의존하지 않고 온디바이스에서 동작하는 경량 목표 지향 주행(goal-directed navigation) 모델을 제안한다. 구체적으로, 개방형 어휘 객체 탐지(Open Vocabulary Object Detection) 모델인 OWLv2[2]와 그라운딩 멀티모달 모델인 Kosmos-2[3]의 비전 인코더의 장점을 이용하였다.

OWLv2는 개방형 어휘 객체 탐지를 통해 목표 객체의 위치 정보를 실시간으로 제공하고, 실시간 위치 그라운딩 출력을 표현한다. Kosmos-2의 비전 인코더는 phrase grounding 목적으로 사전 학습되어 얻어진 시각 표현을 제공한다. 본 연구에서는 두 인코더의 가중치를 고정한 채, 서로 다른 특성을 갖는 시각 표현을 경량 프로젝션 계층을 통해 융합한다. 이런 방법으로 대규모 VLM의 추론 없이도 목표 지향 행동을 생성할 수 있는 경량 파이프라인을 구성하였다. 융합된 표현은 소형 MLP 액션 헤드를 거쳐 이산 행동으로 직접 변환된다. 제안 모델의 전체 파이프라인은 바퀴형 로봇에 탑재된 NVIDIA Jetson Orin NX 보드에 배포되어 동작하며, 실제 주행 실험을 통해 성능을 정량적으로 검증하였다.

본 논문에서 다루는 연구 문제와 범위는 다음과 같이 정의한다. 본 연구는 범용적인 언어 지시 기반 내비게이션보다는 저자원 연산 환경에서의 목표 지향 주행에 초점을 둔다. 이에 따라 학습 과정에서 접하지 않은 새로운 물체나 표현에 대한 제로샷(zero-shot) 일반화와 사전 구축된 투

www.kci.go.kr

어 영상이나 맵 없이 임의의 환경을 탐색하는 tour-free/map-free 내비게이션은 연구 범위에서 제외한다. 실험에서는 사전에 정의된 목표 물체를 나타내는 단일 텍스트 프롬프트를 사용한다. 그리고 소수의 고정된 목표 지점을 대상으로 목표 도달(goal-reaching) 성능을 평가한다. 특히 본 연구에서 텍스트 입력은 매 추론마다 고정된 단일 텍스트 프롬프트로 주어진다. 사용자가 매번 다른 자연어 지시를 내리는 형태가 아니다. 이러한 이유로 본 연구의 모델은 이미지와 텍스트를 입력으로 받아 로봇 행동을 출력하는 VLA 구조를 갖지만, 실험에서는 사전에 정의된 제한된 어휘(constrained vocabulary)를 사용하였다. 즉, 본 연구가 다루는 문제는 사전에 정의된 소수의 고정 목표 지점에 대한 목표 도달 문제이다. Mobility VLA가 지향하는 범용적 언어 내비게이션이 아니라 연산 자원과 인프라가 제한된 환경에서의 경량 내비게이션을 제시하는 것을 목적으로 한다.

본 논문의 주요 기여는 다음과 같다. 첫째, 경량 Grounding-VLA 구조를 제안한다. OWLv2와 Kosmos-2가 제공하는 상이한 시각-언어 그라운딩 표현을 융합하여, 대규모 VLM 기반 클라우드 추론 및 사전 구축 인프라에 대한 의존성을 줄이면서 목표 지향 주행이 가능한 경량 VLA 구조를 설계하였다. 둘째, 온디바이스 로봇 시스템을 구현하고 실기 환경에서 검증하였다. 클라우드 연결이나 사전 구축된 내비게이션 인프라에 의존하지 않고, 제안된 모델을 옴니휠 로봇에 배포하고 구동하여 목표 도달 성능을 검증(제한된 어휘와 고정 목표 지점 조건)하였다. 셋째, 엣지 배포를 위한 추론 파이프라인을 정량적으로 분석하였다. 전체 perception-to-action 파이프라인을 구성 요소별로 분해하여 각 단계의 지연시간을 측정하고 주요 병목 구간을 식별하였다. 이를 통해 초저자원 플랫폼으로 확장할 때 우선적으로 최적화해야 할 구성 요소를 제시하였다.

II. Preliminaries

1. Related works

1.1 Mobility VLA

Mobility VLA는 VLM과 오프라인 토폴로지 그래프를 결합한 계층적 VLA 내비게이션 정책을 사용한다. 고수준 정책은 사용자의 멀티모달 지시와 데모 투어 영상을 입력받아 목표 프레임을 탐색한다. 그리고 저수준 정책은 이를 토폴로지 그래프상의 최단 경로로 변환하여 웨이포인트(waypoint) 행동을 생성한다. 저자들은 토폴로지 그래프 없이 VLM이 직접 웨이포인트를 생성하도록 한 경우 0%의 성공률을 보였다고 보고하였다. 이를 근거로 그래프 기반 저수준 정책의 필요성을 실증하였다. 다만 이 구조는 클라우드 VLM 호출과 사전 투어 촬영을 전제로 하여, 자원이 제한된 플랫폼으로의 이식이 어렵다.

1.2 GNM (General Navigation Model)

GNM[4]은 다양한 cross-embodiment 학습 데이터로 학습된 범용 목표 조건화(goal- conditioned) 시각 내비게이션 정책을 사용한다. 이 모델은 여러 종류의 로봇을 제로샷으로 제어할 수 있으며 새로운 로봇이나 하위 과제에 효율적으로 파인튜닝 될 수 있다. 배포 시에는 로봇의 관측을 노드로, 학습된 정책의 시간적 거리 추정치를 엣지로 하는 토폴로지 그래프와 결합하여 사용한다.

하지만 GNM은 처음부터 이미지 목표(goal image) 기반으로 설계되어 자연어 지시 자체를 입력으로 받지 않는다. 그리고 장거리 내비게이션을 위해서는 여전히 사전 주행을 통한 토폴로지 그래프 구축이 전제된다. 즉 여러 로봇에 대한 범용성은 확보했으나, 언어 인터페이스의 부재와 사전 매핑 의존성이라는 한계는 그대로 남아 있다.

1.3 ViNT (Visual Navigation Transformer)

ViNT[5]는 비전 기반 로봇 내비게이션에 범용 사전학습 모델의 성공을 목표로 하는 파운데이션 모델로 다양한 하위 내비게이션 과제에 효율적으로 적응할 수 있다. 미탐색 환경에 대해서는 토폴로지 그래프 기반 전역 플래너를 이용해 탐색할 수 있다. ViNT는 목표 조건화 내비게이션에서 최고 수준의 성능을 보인다. 하지만 자유 탐색(undirected exploration)은 수행할 수 없으며 외부의 서브골 제안 메커니즘이 별도로 필요하다. 또한 ViNT는 서브골 이미지를 수집하기 위한 초기 주행(initial run)이 필요하다. 즉 GNM 대비 적응성은 향상되었으나, 사전 투어를 통한 서브골/토폴로지 그래프 확보라는 전제는 동일하게 유지된다.

1.4 NoMaD (Goal Masked Diffusion Policy)

NoMaD[6]는 ViNT를 확장하여 목표 조건화 내비게이션과 자유 탐색을 하나의 정책으로 통합함으로써 ViNT의 한계를 직접 겨냥한 후속 연구이다. 목표 조건화 정책의 한계를 파악하고, ViKiNG의 설정을 따라 정책을 환경 경험의 토폴로지 그래프 형태의 일화적 메모리와 결합하였다. 실제로 미탐색 실내·실외 환경 모두에서 벽과 장애물을 회피하고 다중모드 행동 분포를 생성하는 탐색 능력을

www.kci.go.kr

보였다. 이는 ViNT의 "자유 탐색 불가"라는 한계를 해소한 것이다. 하지만 이는 오히려 장거리·대형 환경에서는 여전히 토폴로지 그래프에 대한 의존이 구조적으로 필수적임을 재확인시켜 준다.

Mobility VLA와 GNM, ViNT, NoMaD로 이어지는 일련의 연구는 공통적으로 사전 투어 주행과 토폴로지 그래프 구축이라는 사전 구축 인프라를 전제로 한다. 이를 실행하기 위한 상당한 온보드 또는 클라우드 연산 자원을 요구한다.

이처럼 앞선 연구들은 범용성과 자유 탐색 능력의 확장이라는 축을 따라 발전해 왔다. 이와 달리 본 연구는 사전 주행 및 토폴로지 그래프 구축과 클라우드 VLM에 의존하는 기존의 계층적 내비게이션 구조를 배제한다. 그리고 그라운딩 기반의 경량 모델만으로 목표 지향 주행을 온디바이스에서 수행한다는 점에서 차별성을 갖는다.

III. The Proposed Scheme

본 연구는 클라우드 인프라나 사전 구축 맵 없이, 고정된 목표 지점까지 이동하는 경량형 온디바이스 목표 지향 주행 모델을 제안한다. 제안된 모델의 파이프라인에서 중요한 역할을 하는 것은 그라운딩 인코더이다. 그라운딩 인코더는 컴퓨터 비전 및 자연어 처리(NLP) 분야에서 텍스트 명령어를 입력 받아 이미지나 영상 속에서 해당 물체의 구체적인 위치(Bounding Box 등)를 찾아낸다.

1. Model Architecture

제안한 파이프라인은 Fig. 1과 같이 (1) 이종 그라운딩 인코더 기반 시각 특징 추출, (2) 특징 융합, (3) 액션 헤드의 세 단계로 구성된다.

1.1 Heterogeneous Grounding Encoder

입력 이미지(1280×720 RGB)는 두 개의 독립적인 인코더에 병렬로 입력된다. 최근에는 로봇 조작(manipulation) 분야에서도 개방형 어휘 파운데이션 모델 기반 그라운딩이 적극 활용되고 있다[7].

첫째, OWLv2는 detection-grounded 표현을 추출한다[8]. 둘째, Kosmos-2의 비전 인코더(CLIP ViT-L/14[9], 0.303B)는 phrase-grounded 표현을 추출한다. 두 인코더는 학습 과정에서 파인튜닝 되지 않고 고정된(frozen) 상태로 사용되어, 별도의 도메인 특화 파인튜닝 없이도 사전 학습된 그라운딩 지식을 재사용한다.

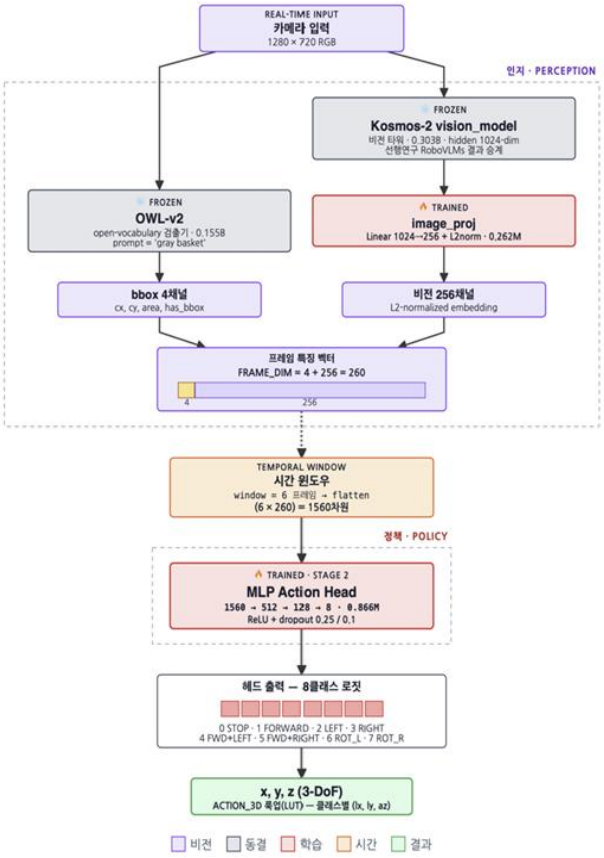


Fig. 1. Model Architecture

Kosmos-2 인코더에서 추출된 특징은 Linear projection(1024→256)을 통해 동일한 차원의 임베딩 공간으로 매핑 된 후 L2 정규화를 적용하여 스케일을 통일한다. 각 프레임의 특징 벡터는 총 260차원(FRAME_DIM = 4 + 256)으로 구성되며, bbox 채널과 비전 채널의 두 부분으로 나뉜다.

첫 번째 부분은 OWLv2의 그라운딩 출력으로부터 얻어지는 4채널의 bbox 특징(cx, cy, area, has_bbox)이다. 두 번째 부분은 Kosmos-2의 vision_model에서 추출된 1024차원 비전 특징을 image_proj를 통해 256차원으로 투영한 후 L2 정규화를 적용한 비전 채널이다. 이 두 부분을 concatenation로 결합하여 얻어진 260차원의 프레임 특징 벡터가 MLP 행동 헤드의 최종 입력으로 사용된다.

본 논문에서 제안하는 모델이 서로 다른 그라운딩 인코더를 융합한 이유는 OWLv2와 Kosmos-2의 단독 사용 시 가지고 있는 한계 때문이다. 실제로 모델의 파이프라인은 초기에 Kosmos-2 자체의 phrase-grounding만으로 그라운딩을 수행하였다. 그러나 테스트시에 대부분의 프레임에서 검출 실패(non-detection)가 발생하였다. 그 원인은 phrase-grounding의 격자(Grid)화된 위치 토큰 구조로 인해 양자화 오차(Quantization Error)가 발생하여 물리

www.kci.go.kr

적 좌표의 정밀도가 부족한 것에 기인한다[10]. 이 문제를 해결하기 위해 PaliGemma2를 거쳐 최종적으로 OWLv2를 명시적 위치 검출기로 채택하였다.

OWLv2는 정적인 사물 탐지에는 탁월하지만 단일 프레임 기반의 4채널 기하학적 정보만 제공한다. 그러므로 로봇이 목표에 다가감에 따라 발생하는 동적 상태 변화(Dynamic State Transition) 및 시계열적 맥락을 모델링하는 데 정보의 병목(Information Bottleneck)이 존재한다[11]. 이러한 구조적 한계로 인해 OWLv2의 명시적 bbox만으로는 가려짐·방해물·부분 시야와 같은 장면 전반의 맥락 정보가 유실된다. 따라서 본 연구는 두 인코더를 상호 보완적 요소로 파악하고, OWLv2의 정밀한 공간 위치 정보와 Kosmos-2의 고정된(frozen) 밀집 시각 표현을 융합하는 하이브리드 프레임워크를 제안한다[12]. 이는 명시적 개방형 어휘 객체 탐지와 암묵적 시각 맥락을 결합함으로써, 각 모델이 독립적으로 사용 시에 발생하는 구조적 한계를 상호 보완하였다.

1.2 Lightweight MLP Action Head

본 연구에서는 Kosmos-2 VLM을 기반으로 하여 액션 예측 기능을 추가하였다. 융합된 256차원 특징은 1초에 6프레임으로 6개의 Windows를 입력 받아 평탄화(flatten)하여 1560차원으로 된다. 1560차원이 2개의 hidden layer로 구성된 MLP 행동 헤드에 입력된다. MLP의 구성은 1560차원을 입력 받아 Hidden layer 1층(512 노드)과 Hidden layer 2층(128 노드)을 거쳐 8차원으로 출력한다. 연속적인 좌표 회귀나 diffusion[13]/flow-matching[14] 기반 action expert 대신 경량 분류기를 사용함으로써 온디바이스 실시간성을 확보하는 것을 목표로 한다. 액션 헤드 설계 시 연속형(regression) 방식이나 LSTM·Transformer 기반의 시퀀스 헤드 구조도 고려하였다. 그러나 본 연구는 8-class 이산 분류 방식을 최종적으로 채택하였다. 그 근거는 다음과 같다. 연속 값을 직접 회귀하거나 시간적 문맥을 모델링하는 방식은 일반적으로 더 큰 규모의 데이터를 필요로하는 경향이 있다. 반면에 사전에 정의된 고정 속도값(룩업 테이블)의 조합으로 행동을 표현하는 이산 분류 방식은 상대적으로 작은 규모의 데이터에서도 안정적인 학습이 가능할 것으로 판단하였다. 본 연구의 데이터셋 규모(225 에피소드)를 고려할 때, 이러한 데이터 효율성은 실용적으로 중요한 설계 기준이었다. 1560차원의 입력 값은 액션 헤드를 거쳐 이산 행동 공간에 대한 8개 행동 클래스 중 하나로 분류된다. 행동 클래스는 정지(0), 직진(1), 횡이동(2, 3), 대각 직진(4, 5), 제자리 회전(6, 7)으로 구성된다. 선택된 행동 클래스는 Table 1에서 보듯이 고정된 룩업 테이블(Fixed LUT)을 참조하여 3자유도(3-DoF)로 매핑 된다.

Table 1. Lookup Mapping Table

class idx	class name	x	y	z
0	STOP	+0.00	+0.00	+0.00
1	FORWARD	+1.15	+0.00	+0.00
2	LEFT	+0.00	+1.15	+0.00
3	RIGHT	+0.00	-1.15	+0.00
4	FWD+L	+1.15	+1.15	+0.00
5	FWD+R	+1.15	-1.15	+0.00
6	ROT_L	+0.00	+0.00	+0.25
7	ROT_R	+0.00	+0.00	-0.25

2. Hardware Platform

제안 파이프라인은 바퀴형 모바일 로봇에 탑재된 NVIDIA Jetson Orin NX[15] 상에서 온디바이스로 실행된다. 카메라는 1280×720 해상도의 RGB 영상을 실시간으로 제공하며, 별도의 클라우드 서버나 외부 네트워크 연결 없이 이미지 입력부터 행동 출력까지의 전 과정이 로봇 내부에서 처리된다. 본 연구에서 사용한 모바일 로봇은 Fig. 2와 같이 3축 옴니휠(Omni-wheel)을 120° 간격으로 배치한 Kiwi drive 방식의 전방향(holonomic) 구동 로봇이다. 로봇은 두 개의 보드로 구성이 되어 있다. Top 보드(Board)는 NVIDIA Jetson Orin NX 보드가 탑재되어 있다. 이 보드에 EdgeGround-VLA 모델을 배포하여 런타임시 추론을 수행한다. Motor 보드는 Cortex-M4 CPU를 탑재하며, Top 보드와는 CAN(Controller Area Network) 인터페이스를 통해 통신한다. 런타임시에 Top 보드에서 전달되는 3Dof 값을 받아 바퀴를 제어한다.



Fig. 2. Serbot II

www.kci.go.kr

전방향으로 구동되므로 회전 없이도 횡이동 및 대각 이동이 가능하며, 이는 8클래스 이산 행동 공간(정지·직진·횡이동·대각이동·회전) 설계의 물리적 근거가 된다. 순수 횡이동·대각 이동은 일반적인 차동 구동(differential drive) 방식으로는 물리적으로 구현할 수 없다. 그러나 본 로봇의 3축 옴니휠 구조는 별도의 회전 보정 없이 직접 수행 가능하다.

3. Collection of data

액션 헤드 학습에 사용된 데이터는 Fig. 3과 같이 사람이 조이스틱을 이용해 로봇을 직접 원격 조작(teleoperation)하는 방식으로 수집되었다[16].

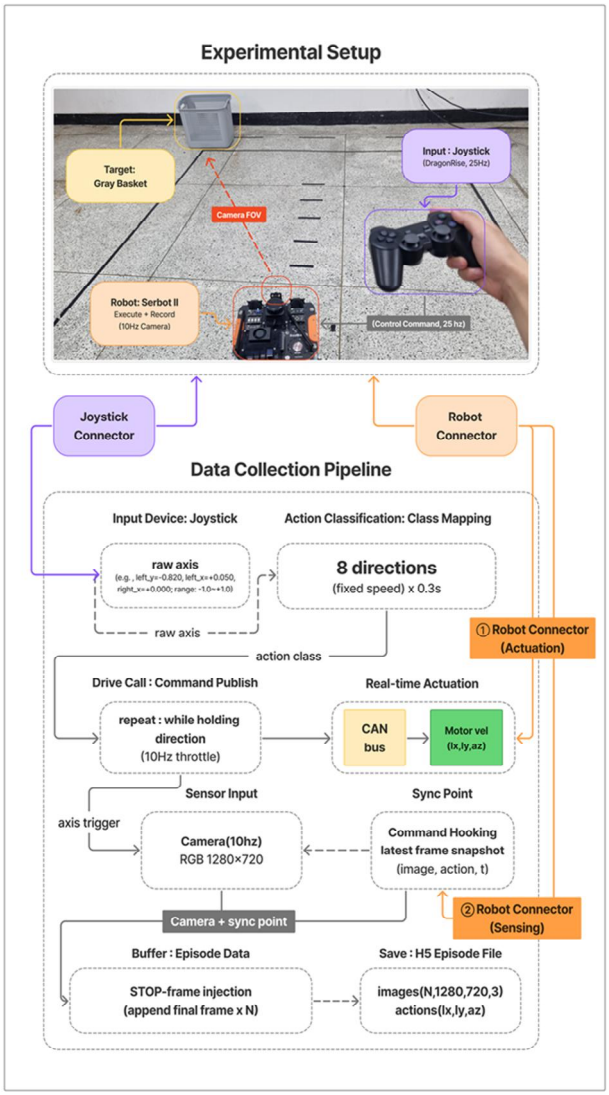


Fig. 3. Collection of Data

각 에피소드는 하나의 독립된 HDF5(.h5) 파일 규격으로 저장된다. 각 파일은 RGB 1280×720 해상도 이미지 시퀀스와 이에 대응하는 행동 라벨과 수집 당시의 실행 설정을 기록한 런타임 설정을 포함한다. 수집된 데이터는 1개의 시작 위치(중앙)와 5개의 경로 유형(중앙·좌(약좌/강좌)·우(약우/강우))으로 구성되며, 총 225개 에피소드(16,599 프레임)를 수집하였다. 경로 유형별 구성은 중앙 45 에피소드, 좌측 90 에피소드, 우측 90 에피소드로 좌/우 에피소드 수가 정확히 대칭을 이루도록 수집되었다. 실제로 bbox 중심 좌표(cx)의 라벨 평균값은 0.4954로 이론적 중심 값인 0.5에 매우 근접한다. 이것으로 데이터셋 자체에는 좌우 공간적 편향이 존재하지 않음을 확인하였다.

4. Dataset

데이터셋은 SPLIT_SEED=42로 고정된 시드 하에 VAL_RATIO=0.15로 분할되어, 192개의 훈련(training) 에피소드와 33개의 검증(validation) 에피소드로 구성된다. 각 프레임의 시각 특징은 L2 정규화 규격을 적용하여 사전에 캐싱하였다. 데이터셋의 구성은 Table 2와 같고, 액션 클래스는 Table 3과 같이 분류된다.

Table 2. Dataset Composition

Item	Value
Episodes	225
Total Frame	16,599
Train / Val	192 / 33 (VAL_RATIO=0.15, SPLIT_SEED=42)
Vision Cash	exp73_v6_vis_cache.pt Result of Stage 1
bbox Annotation	OWLv2-based bbox Labeling

Table 3. Action Class Distribution

Class	Name	Frame	Ratio
0	STOP	1,452	8.75%
1	FORWARD	7,641	46.03%
2	LEFT	716	4.31%
3	RIGHT	858	5.17%
4	FWD+L	2,532	15.25%
5	FWD+R	3,117	18.78%
6	ROT_L	136	0.82%
7	ROT_R	147	0.89%

5. Perform Model Training

5.1 Training by Module

제안 파이프라인의 각 구성요소는 서로 다른 학습 상태를 가진다. OWLv2와 Kosmos-2 비전 인코더는 모두 고정된 상태로 사용되어 사전 학습된 그라운딩 지식을 그대로 재사용한다. 전체 파이프라인에서 실제로 학습되는 구성요소는 두 인코더의 출력을 정렬하는 Stage 1의 이미지 프로젝션(projection)과 Stage 2의 MLP 액션 헤드 두 부분뿐이다.

www.kci.go.kr

또한 Kosmos-2의 언어 디코더는 경량화를 위해 설계 단계에서 제외되어 본 파이프라인에서 호출되지 않는다. 각 구성 요소의 학습 상태와 그 근거는 Table 4와 같다.

Table 4. Training status of pipeline components

Module	Status	Usage
OWLv2	frozen	Text prompt only
Kosmos-2 Vision Encoder	frozen	Feature extractor
Kosmos-2 Language Decoder	frozen	Anchor matching(train only)
image_proj	Training (stage1)	Stage 1: anchor matching -> reused in Stage 2
MLP Action Head	Training	Discrete action classification

5.2 Hyperparameters used during Training

훈련시에 사용된 하이퍼파라미터는 Table 5와 같다. 하이퍼파라미터 설정시에 주의가 필요한 3가지의 보정 계수는 다음과 같다. threshold는 OWLv2가 "이 물체를 검출했다"고 판단하는 신뢰도 기준선으로, 오검출과 미검출 사이의 균형을 잡는 값이다. OWLv2의 검출 신뢰도 기준값은 ROC 실측 결과에서 오탐률 0%와 정탐률 94.9%를 달성한 최적값 0.25를 기준으로 설정하였다. Jetson Orin NX 실기 환경에서는 검출 민감도를 높이기 위해 0.20으로 완화 적용하였다. 이후 실제 주행 실험을 통해 해당 기준값의 적정성을 검증하고 최종 확정하였다.

bbox_scale은 area 신호가 비전 특징에 묻혀 무시되지 않도록, 스케일을 맞춰주는 보정 계수이다. 로봇이 목표에 가까워질수록 카메라에 보이는 목표 물체의 면적(area)이 커진다. 반면 함께 입력되는 image_proj의 256차원 비전 특징은 L2 정규화가 되어 있어 스케일이 항상 일정하다. 면적 값(0.02~0.56)은 L2 정규화된 256차원 비전 특징에 비해 절대 크기가 너무 작아, 모델이 거리 신호를 무시할 위험이 있다. bbox_scale=3.0을 곱해 면적의 크기를 비전 특징과 비슷한 수준으로 키움으로써, 이 신호가 학습에 유효하게 반영되도록 한다.

class weight는 다수 클래스인 FORWARD에 치우친 학습이 소수 클래스인 “ROT_xx(회전)” 등을 무시하지 못하도록 손실 함수에 균형을 부여하는 보정 계수이다[17]. FORWARD가 46.03%로 편중되어 있어, 이것만 찍어도 정확도가 46%에 달해 소수 클래스(ROT_L·ROT_R)를 무시하고 학습이 끝날 위험이 있다. 이를 막기 위해 클래스별 빈도의 역수를 정규화한 가중치를 Cross-Entropy에 적용하여 소수 클래스의 오분류 페널티를 키운다.

Table 5. Hyperparameters

Parameter	Value	Description
threshold	0.25(default) 0.20(deployment)	groudning confidence threshold
window	6 frame	consecutive input frames
bbox_scale	3.0	bbox coordinate scale
num_classes	8	8-class action logit
optimizer	AdamW, lr=5e-4	AdamW optimizer
epochs	300	Training iterations
class weight	1/frequency normalization	Corrects FORWARD's 46& imbalance
scheduler	CosineAnnealingLR	cosine learning rate decay

5.3 Explanation of Training Methods

학습 분할은 에피소드 단위로 15%를 검증용으로 떼어냈으며, 프레임 단위가 아니라 에피소드 단위로 분할하였다. 전체 학습 파이프라인은 Fig. 4와 같다. 학습 대상은 Stage 1에서 image_proj를 이미지에서 추출한 특징을 방향을 서술하는 5개 문장의 텍스트 특징과 대조하여 정렬시키는 학습을 수행한다. 그리고 Stage 2에서 MLP 헤드(0.866M)를 학습하는 구조이다. 전체 학습 파라미터는 1.128M이다. OWLv2와 Kosmos-2 비전은 두 단계 모두 고정된 상태로 유지된다.

image_proj는 카메라가 찍은 한 장의 이미지가 5단계 방향(강좌·약좌·중앙·약우·강우) 중 어디에 가장 가까운지 맞히도록 학습된다(Fig 4, Stage 1). 이미지에서 뽑은 특징과 방향을 서술하는 5개 문장에서 뽑은 특징 사이의 유사도(코사인 유사도)를 비교한다. 정답 방향 문장과는 유사도를 높이고 나머지 문장과는 낮추는 방식으로 학습한다. 이 과정을 거치면 image_proj는 별도의 방향 표시 없이도 카메라 영상만 보고 목표물의 대략적인 위치를 가늠할 수 있게 된다.

www.kci.go.kr

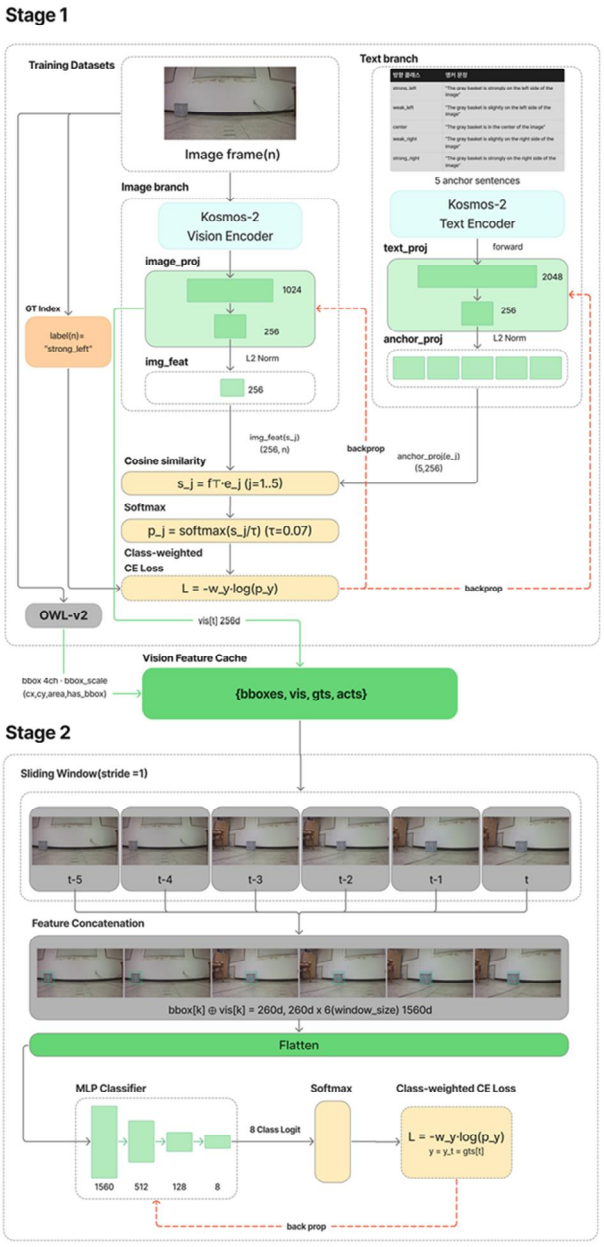


Fig. 4. Pipeline Structure Diagram of Stage 1/ Stage 2

5.4 image_proj Training

image_proj는 강도별 방향을 서술하는 5개의 텍스트 앵커(예: "많이 왼쪽에 있음"/"살짝 왼쪽에 있음"/"중앙에 있음" 등)와의 사전 학습되는 구성요소로, Kosmos-2 비전 특징(1024차원)을 256차원으로 사상한다. 액션 헤드 학습에 사용한 225 에피소드 데이터 셋의 방향 구조(강좌·약좌·중앙·약우·강우, 각 45 에피소드)를 그대로 살려 image_proj를 Table 6과 같이 5-class로 학습하였다. 라벨은 서빙에 쓰는 검출기(OWLv2)의 출력을 그대로 사용해 학습·서빙 간 일관성을 확보했다.

Table 6. Anchor Sentences for 5-Direction Classification

Path	Frame	Anchor Sentence	Class Weight (training)
Strong Left	3611	"The gray basket is strongly on the left side of the image"	0.91
Weak Left	3299	"The gray basket is slightly on the left side of the image"	1.00
Center	2819	"The gray basket is in the center of the image"	1.17
Weak Right	3268	"The gray basket is slightly on the right side of the image"	1.01
Strong Right	3602	"The gray basket is strongly on the right side of the image"	0.91

image_proj의 훈련 정확도는 Fig. 5와 같이 96.10%이고, 검증 정확도는 94.09%로 잘 일반화된 학습 패턴이다.

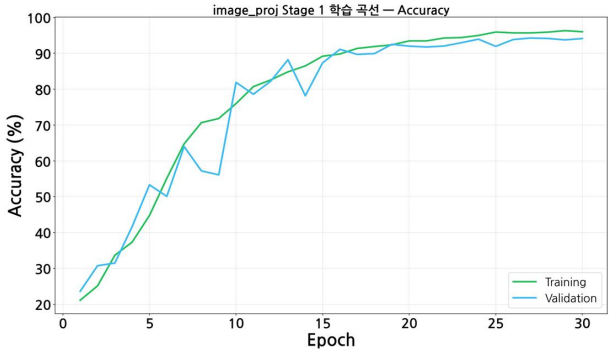


Fig. 5. Accuracy Graph of image_prj

손실(loss)을 보면, Fig. 6과 같이 훈련 손실은 0.1072이고, 검증 손실은 0.1728이다. 이 결과값은 학습이 잘된 상태임을 나타낸다.

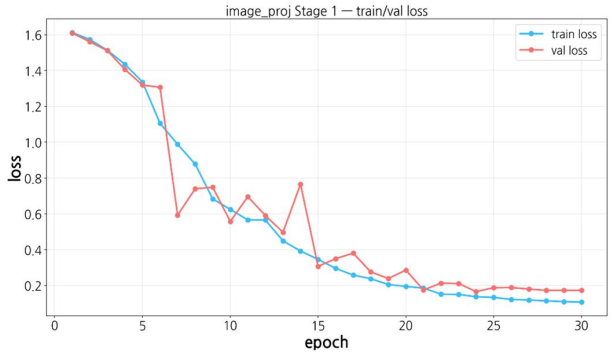


Fig. 6. Loss Graph of image_prj

5.5 Action Head Training

액션 헤드에서 학습 시에 Fig. 7과 같이 에폭(epoch) 225에서 훈련 정확도는 91.44%이고, 검증 정확도는

www.kci.go.kr

74.04%로 17.40%의 격차가 발생한다. 이 수치는 훈련이 과적합 되었음을 의심하게 된다. 그러나 이러한 현상은 원격조작 기반 모방 학습(imitation learning)에서 잘 알려진 현상이며, 여러 원인이 문헌에 정리되어 있다[18].

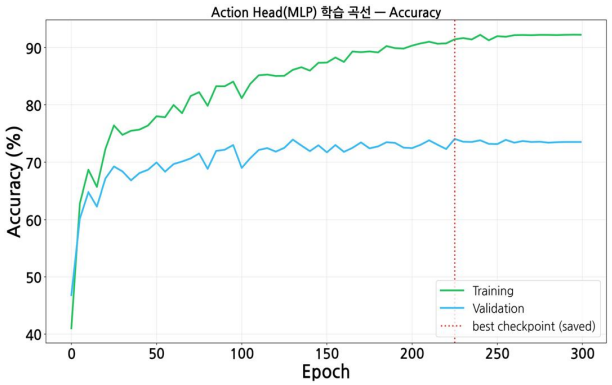


Fig. 7. Accuracy Graph of Action Head

이 격차를 그대로 성능 저하로 해석하는 것은 적절하지 않다. 검증 정확도는 매 장면(프레임)에서 사람이 조작한 단 하나의 행동과 일치하는지를 묻는 반면 실기 성공률은 에피소드 단위로 목표에 도달했는지를 묻는다. 특정 시점에서 목표에 접근할 수 있는 행동은 일반적으로 하나가 아니다. 예를 들어 목표가 약간 왼쪽에 있을 때 FOWARD와 FWD+LEFT는 모두 접근 가능하다. 그러기에 검증 정확도와 실기 성공률은 차이가 날 수 있다. 이 주장은 Fig. 8의 검증 혼동행렬로 확인된다.

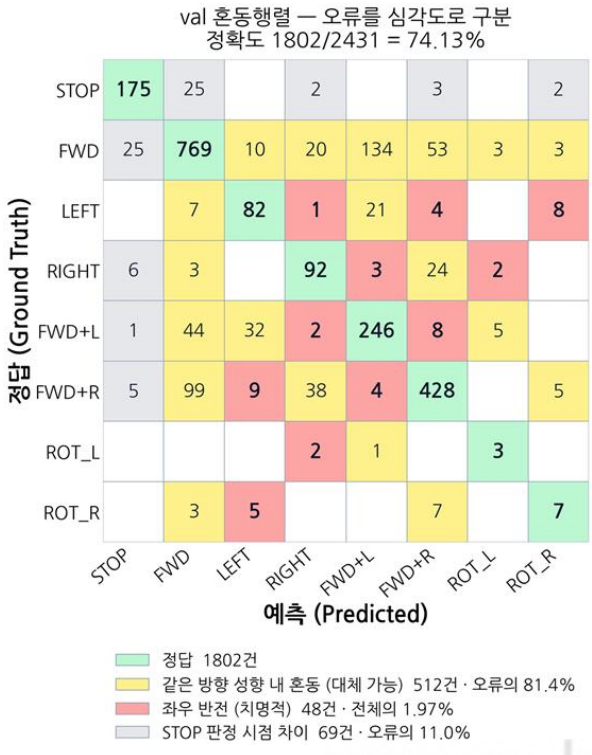


Fig. 8. Validation Confusion Matrix

위의 혼동행렬에서 보이는 오류 629건을 분해하면 Table 7과 같다.

Table 7. Severity Breakdown of Action Head Validation Errors

Error Type	Count	Total (%)	Error (%)
Left-Right Flip(Critical)	48	1.97%	7.6%
Same-Direction Confusion (Substitutable Action))	512	21.06%	81.4%
Stop Timing Difference	69	2.84%	11.0%

좌우 방향이 완전히 틀린 치명적 오류는 전체 2431 프레임 중에서 48개로 1.97%이다. 오류의 81.4%는 틀린 행동이라기보다 같은 방향으로 접근하는 다른 유효한 선택이다. 주행 시에 실제로 실패가 되는 좌우 반전은 전체 판단의 1.97%에 불과하다.

검증 손실은 에폭 20 이후 0.926에서 1.953까지 상승한다. 액션 헤드의 체크포인트는 검증 손실(validation loss)이 아닌 검증 정확도(validation accuracy)를 기준으로 선정하였다. Fig.7 에서 확인되는 바와 같이, 최종 채택된 체크포인트의 에폭 225는 검증 정확도가 최고 수준에 도달한 시점이다. 하지만 이 시점의 검증 손실(약 1.7~1.9)은 Fig. 9의 그래프와 같이 학습 초반(epoch 10~30)의 최저점(약 0.85 ~ 0.95)보다 오히려 높다.

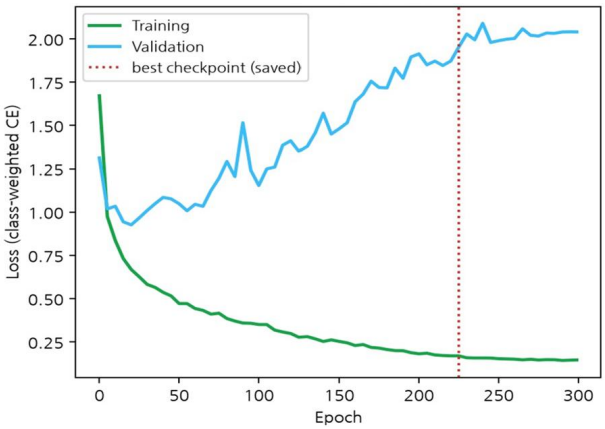


Fig. 9. Loss Graph of Action Head

이는 학습이 진행됨에 따라 모델이 정답을 맞히는 비율은 유지·개선하면서도, 틀리는 소수의 샘플에 대해서는 점점 더 오분류하는 방향으로 나아갔음을 시사한다. 즉 정확도 기준으로 선정된 체크포인트는 이미 '가장 잘 맞히는'

www.kci.go.kr

시점의 것이다. 오분류가 발생하는 경우도 대부분 기능적으로 유사한 인접 행동 클래스(예: 직진과 오른쪽/왼쪽 직진) 사이에 국한된다. 그래서 폐루프 주행에서는 다음 프레임의 재정정(Re-correction)을 통해 목표 도달에 치명적인 영향을 미치지 않았을 가능성이 크다.

6. Deploy the Trained EdgeGround-VLA Model on the Robot

학습이 완료된 EdgeGround-VLA 모델을 실제 로봇(Serbot II)에 탑재하는 절차는 Fig. 10과 같다. 먼저 서버에서 학습된 두 체크포인트인 행동 헤드 체크포인트와 image_proj 체크포인트(5-class 방향 대조 학습 가중치)를 로봇에 탑재된 Jetson Orin NX로 이전하여 저장한다. Jetson 보드의 추론 서버(inference server)가 추론 시에 이 두 체크포인트를 로드한다. 그리고 HTTP API를 통해 카메라 프레임을 수신하고 예측된 행동을 반환한다.

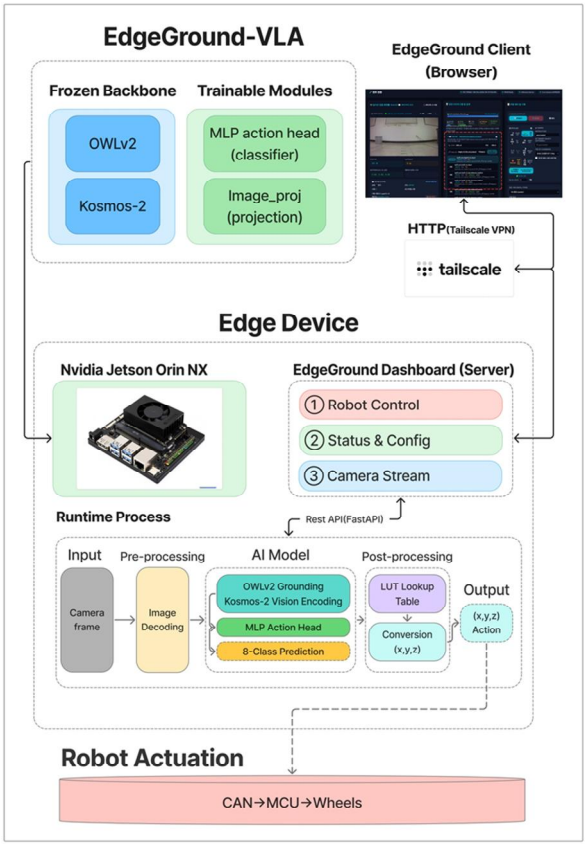


Fig. 10. EdgeGround-VLA model deploy and runtime process

7. Runtime Inference Control Logic

제안한 파이프라인은 학습된 액션 헤드의 출력만으로 동작하지 않는다. 실기 배포 시 발생하는 예외 상황과 지연시간 제약을 다루기 위해 Fig. 11과 같이 별도의 서빙(serving) 제어 로직을 함께 포함한다. 이는 크게 (1) 타겟 미검출 시 폴백 처리, (2) 그라운딩 호출 주기 제어(skip_n), (3) 회전 시 안전 제어, (4) 학습된 정지(STOP) 판정과 콜드스타트 가드의 네 가지로 구성된다.

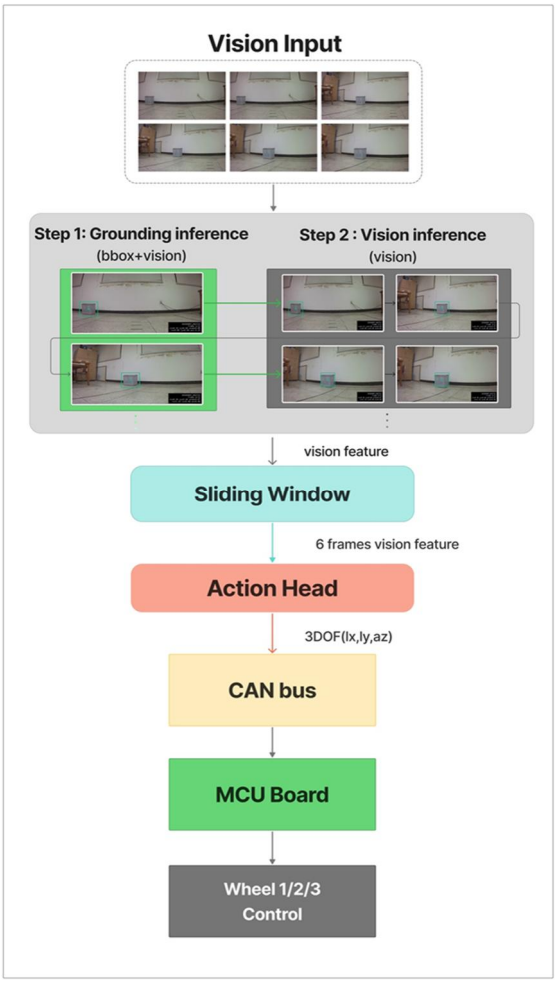


Fig. 11. Runtime Inference Control Logic of EdgeGround-VLA model

7.1 Fallback Processing When Target is not Detected

OWLv2가 타겟 검출에 실패한 경우, 위치 채널에는 고정된 fallback 값(cx=0.5, cy=0.6, area=0.06)과 함께 has_bbox=0 신호가 주입된다. 학습 데이터셋에는 미검출 프레임이 존재하지 않아 모델이 이러한 상황에 대한 응답을 학습할 기회가 없었다. 그러므로 본 연구는 이 문제를 모델의 학습된 대응에서 수행하지 않았다. 대신 서빙 단계의 제어 로직(강제 재그라운딩 및 정지 명령 치환)으로 우회 처리하였다.

www.kci.go.kr

7.2 Grounding Call Cycle Control and Safety Control during Rotation

Table 10에서 확인한 바와 같이 OWLv2 그라운딩은 Jetson Orin NX 상에서 1901.7ms(fp32 기준) 지연된다. 이는 전체 추론 지연의 약 97%를 차지한다. 그러므로 매 프레임마다 그라운딩을 호출하면 제어 루프 자체가 성립하지 않는다. 이러한 이유로 grounding_skip_n=3을 적용하여 3프레임마다 한 번(1, 4, 7, … 프레임)만 그라운딩을 수행하도록 하였다. 그 사이의 프레임에서는 직전에 얻은 bbox를 유지한다. 이때 액션 헤드(MLP)의 순전파는 약 2ms 걸리므로 매 프레임마다 수행 가능하다. 그라운딩과 행동 결정의 갱신 주기를 분리함으로써 실효 그라운딩 주기 약 1.3Hz 수준에서 프레임 단위의 행동 출력을 유지할 수 있다. 다만 이 방식은 bbox 정보가 최대 2프레임까지 지연되는 대가를 수반한다. 이 지연은 로봇이 정지해 있거나 직진하는 상황에서는 큰 문제가 되지 않는다. 하지만 회전 중에는 시야 자체가 빠르게 변화하므로 오래된 bbox 정보가 곧바로 조향 오차로 이어진다. 따라서 skip_n 기반 그라운딩 주기 제어는 다음의 회전 안전 규칙과 반드시 함께 적용되어야 하며, 두 메커니즘은 하나의 결합된 제어 정책으로 동작한다.

1) 타겟 미검출이 발생하면 skip 주기를 무시하고 즉시 재그라운딩을 수행한다.

2) 회전 동작이 완료된 직후에는 skip_n 여부와 관계없이 즉시 재그라운딩을 강제한다. 이는 회전으로 인해 시야가 이미 바뀐 상태에서 오래된 bbox를 그대로 사용하는 것을 방지하기 위함이다.

3) 제어 계층(move_and_stop_ramped)에서 단일 회전 동작의 지속 시간을 0.4초로 엄격히 제한하여, 과회전으로 인해 타겟이 카메라 시야 밖으로 벗어나는 상황을 방지한다.

4) 회전 명령이 2회 연속으로 발생하는 경우, 이를 정지 명령([0, 0])으로 대체하는 안전 치환을 적용한다.

7.3 Trained Stop Judgment and Cold Start Guard

본 파이프라인의 정지(STOP) 판단은 별도의 기하학적 규칙이 아니라, 액션 헤드가 학습한 STOP 클래스의 분류 결과를 그대로 사용한다. 다만 액션 헤드는 6프레임 크기의 시간 윈도우를 입력으로 사용한다. 그런데 주행을 시작하는 시점에는 과거 프레임이 존재하지 않아 첫 프레임을 6회 반복하여 패딩한다. 이 패딩된 입력에 대해 STOP 클래스가 출력되면 로봇이 출발과 동시에 정지해버리는 오작동이 발생할 수 있다. 이를 방지하기 위해 초기 3회 추론까지는 정지(STOP) 판정을 무시하는 콜드스타트 가드(cold-start guard)를 적용하였다. 이는 window=6 구조에서 파생된 안전장치이다.

IV. Experimental Results

1. Goal-Reaching Success Rate

제안한 모델의 목표 주행 도달 성능을 검증하기 위해, VLA 모델이 탑재된 바퀴형 로봇을 가지고 테스트를 수행하였다. 테스트 환경은 Fig. 12와 같이 2m x 2m 규모의 실내 공간으로 구성된다. 테스트는 5개의 고정 목표 지점(중앙, 약우, 강우, 약좌, 강좌)을 설정하였다. 이는 로봇 정면 기준 좌우 대칭으로 배치하여 접근 각도 범위를 균형있게 커버하도록 설계한 것이다. 약좌/강좌, 약우/강우의 2단계 구분은, 본 연구의 8-클래스 이산 행동 공간이 각 방향에 대해 횡이동과 대각 이동이라는 2단계의 접근 강도를 구분하는 것과 대응하도록 설계하였다.



Fig. 12. Practical Test – started from the center

테스트는 각 목표 지점에 대해 20회씩 총 100회의 실기 에피소드를 수행하였다. 출발점은 훈련시에 사용했던 중앙 위치를 기준으로 하였다. 각 에피소드는 로봇이 출발 지점에서 목표 지점까지 도달하는 과정을 하나의 단위로 측정하였다. 실험 결과는 Table 8과 같이 95.0%의 성공률을 보였다.

Table 8. Seen-start success rate

Target Location	Success Count	Success Rate
Center	20/20	100%
Strong Left	19/20	95%
Weak Left	19/20	95%
Strong Right	19/20	95%
Weak Right	18/20	90%
Overall	95/100	95.0%

www.kci.go.kr

해당 실험에 대한 일반화 성능을 확인하기 위하여, 출발 위치를 Fig. 13과 같이 훈련시에 사용하지 않은 지점에서 수행하였다. POS1과 POS2는 테스트 공간의 중앙을 기준으로 좌우 각 1m 이내(즉 맵 전체 폭의 절반에 해당하는 범위)에서 무작위로 설정되었다. 목표 방향 자체는 동일하게 유지한 채 로봇의 시작 위치만 바꿔서 위치 일반화 성능을 방향 난이도와 분리해 검증할 수 있도록 했다. 목표 방향은 카메라 초기 시야각 안에 목표가 들어오는 범위 내에서 좌우 극단(강좌/강우)까지 포함하도록 설정했다. 그리고 동일한 5방향 세트를 2개 출발 위치에서 반복 측정해 위치 변화에 따른 성능 저하를 정량적으로 분석하고자 했다.

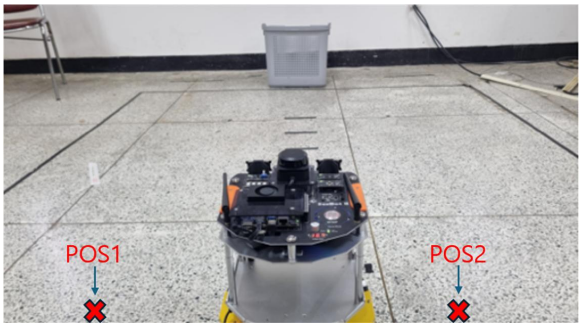


Fig. 13. Practical Test – started from the POS1 and POS2

POS1과 POS2 지점에서 출발하여, 각 목표 지점에 대해 20회씩 총 200회의 실기 에피소드를 수행하였다. 그 결과 Table 9와 같이 평균 90.5%의 목표 도달 성공률을 보였다. 이는 훈련 시 사용했던 중앙 출발 지점 기준 성능(95.0%)과 비교해 4.5%p 낮은 수준이다. 이는 본 테스트 환경이 허용하는 범위 내에서의 출발 위치 변화에 대해 성능 저하가 크지 않음을 보여준다.

Table 9. Unseen-start success rate

Target Location	POS1		POS2	
	Success Count	Success Rate	Success Count	Success Rate
Center	18/20	90%	19/20	95%
Strong Left	18/20	90%	19/20	95%
Weak Left	19/20	95%	19/20	95%
Strong Right	20/20	100%	17/20	85%
Weak Right	16/20	80%	16/20	80%
Overall	91/100	91.0%	90/100	90.0%

다만 본 실험은 단일 고정 텍스트 프롬프트와 목표를 고정시키고, 출발 위치를 변경하며 검증을 하였다. 목표 물체의 위치나 종류를 변경한 경우의 제로샷 일반화 성능은 별도로 검증되지 않았다. 출발-목표 간 상대적 기하 관계가 유사하더라도 시각 그라운딩에 기반한 본 시스템은 배경, 조명, 목표 주변 환경 등 장면 전체의 시각적 조건에 따라 성능이 달라질 수 있다. 그러므로 목표 지점 변경에 대한 성능은 이번 결과로부터 직접 추정하기 어려우며 이를 향후 연구로 남긴다.

실패 케이스의 유형을 실기 데이터를 기반하여 정성적으로 분류시에 두 가지 유형이 발견되었다. 첫째는 POS1 지점 실험에서 발견된 그라운딩 손실로 인한 목표 이탈이다. 그라운딩 인코더(OWL-v2)가 목표를 프레임 시작부터 끝까지 한 번도 검출하지 못하는 경우이다. 이 경우 정책은 그라운딩 신호 없이 학습된 fallback 행동에 의존해 주행하다가 목표 지점에 도달하지 못하고 종료된다. 두 번째는 POS2 지점 실험에서 발견된 정책의 방향 과잉대응이다. 그라운딩 자체는 정상 작동하지만, 정책이 특정 방향(예: 우측)으로 필요 이상으로 꺾어서 목표가 화면 중앙에서 점점 가장자리로 이동하다가 결국 화면 밖으로 밀려나는 경우이다. 목표가 시야를 벗어난 뒤에는 그라운딩도 자연히 실패로 전환되어 이탈이 가속화되어 테스트 영역을 벗어났다.

위 두 가지 실패 유형에 대해 다음과 같은 개선 방향을 계획하고 있다. 첫째, 그라운딩 손실에 대해서는 목표의 화면 내 위치에 따라 그라운딩 호출 주기를 동적으로 조정하는 적응형 skip_n 제어를 도입할 예정이다. 둘째, 방향 과잉대응에 대해서는 좌우 반전 기반 데이터 증강을 통한 데이터 효율성 개선과, 연속된 회전/대각 명령에 대한 감쇠 제어를 함께 적용하여 완화할 계획이다. 다만 본 논문에서는 이러한 개선안의 실제 구현과 정량적 효과 검증까지는 포함하지 못하였으며, 이는 후속 연구로 남긴다.

검증 정확도에 비해 에피소드 단위를 쓰는 실기 성공률이 높은 것에는 구조적 이유가 있다. 검증 정확도에서 쓰는 프레임 단위 채점은 재정정 기회를 반영하지 못한다. 반면 에피소드 단위 지표는 폐루프 제어에서 로봇은 매 판단 시점마다 갱신된 관측으로 다시 판단한다. 그러므로 한 시점에서 대체 행동을 선택하더라도 다음 시점에 목표 위치가 재측정되어 보정된다. 프레임 단위 채점과 폐루프 배포 사이의 이러한 간극은 모방학습 문헌에서도 지적된 바 있다[19][20]. 이러한 이유로 실기 성공률이 seen-start와 unseen-start의 평균 92%까지 도달할 수 있다.

또한 본 연구의 결과는 클라우드 상의 VLM 호출이나 사전 투어·맵 구축 없이 OWLv2(0.155B)와 Kosmos-2 비전 인코더(0.303B)를 모두 고정된 상태로 재사용하고, 실제로는 1.128M개의 경량의 파라미터만 학습하는 파이프라인을 통해 달성되었다는 점에서 의의가 있다.

www.kci.go.kr

2. Component-wise Latency Bottleneck Analysis

제안한 파이프라인이 초저자원 플랫폼으로 확장될 가능성을 논의하기 위해, Jetson Orin NX 상에서 파이프라인 구성 요소별 추론 지연시간을 분해 측정하였다. 측정 결과는 Table 10과 같다.

Table 10. Component-wise latency decomposition

Component	Parameters	Latency	Ratio
OWLv2 Grounding (fp32)	0.155B	1901.7 ms	~97%
OWLv2 Grounding (fp16)	0.155B	962.1 ms	–
Kosmos-2 Vision Encoding	0.303B	53.7 ms	~3%
MLP Action Head	0.866M	<1 ms	~0%

Table 10에서 가장 주목할 결과는 파라미터 수가 더 적은 OWLv2(0.155B)의 그라운딩 단계가 전체 추론 지연의 약 97%(FP32 기준)를 차지한 반면 약 2배 많은 파라미터를 갖는 Kosmos-2(0.303B)는 약 3%에 불과하다는 점이다. 이는 모델의 파라미터 수만으로 추론 지연을 판단하기 어렵다는 것을 보여준다. 또한 FP16로 정밀도 변환 시 OWLv2의 지연시간이 1901.7 ms에서 962.1 ms로 약 49% 감소하여, 병목 단계의 경량화가 전체 추론 지연 감소에 효과적임을 확인하였다.

전체 파이프라인의 종단 지연은 FP32 사용 시는 약 1.96초, FP16로 정밀도 변환 시는 약 1.02초로 측정되었다. 이는 수십 ms 수준의 연속 제어에는 적합하지 않다. 하지만 본 연구에서는 이산 행동을 주기적으로 갱신하는 방식으로 로봇을 제어한다. 실제 주행에서는 새로운 그라운딩 결과가 도착하기 전까지 직전 행동 명령을 10Hz로 재발행하여 이동을 유지하므로, 종단 지연이 로봇의 정지 시간으로 직접 이어지지 않는다. 실제 주행 300회에서 평균 92%의 목표 도달 성공률을 달성한 결과는 이러한 제어 방식이 본 연구의 목표 도달 환경에서 실용적으로 동작함을 보여준다.

이러한 결과는 향후 Raspberry Pi와 같은 초저자원 플랫폼으로 확장할 때 OWLv2 그라운딩 단계의 최적화가 핵심임을 시사한다. 특히 양자화 및 지식 증류를 적용하면 전체 지연의 대부분을 차지하는 병목을 줄일 수 있을 것으로 예상된다. 다만 초저자원 하드웨어 환경으로의 실제 이식과 성능 검증은 본 연구의 범위를 넘어서는 부분으로, 향후 연구를 통해 검토하고자 한다.

V. Conclusions

본 논문에서는 대규모 VLM과 사전 구축된 내비게이션 인프라에 대한 의존성을 줄이기 위해 목표 지향 경량 온디바이스 모델인 EdgeGround-VLA를 제안하였다. 제안한 모델은 추론 시 OWLv2, Kosmos-2, MLP 액션 헤드가 모두 함께 forward pass에 관여하며, 이들을 합친 총 파라미터는 459M이다. 이 규모는 SmolVLA(500M) [21], VLA-Adapter(500M)[22] 등 최근의 경량 VLA와 유사한 수준으로, 대형 VLA(OpenVLA 7B 등)에 비해 훨씬 작다. 다만 이 중 실제로 학습되는 파라미터는 1.128M에 불과하며, 이는 백본(OWLv2, Kosmos-2)을 고정된 상태로 유지하고 adapter/policy만 학습하는 VLA-Adapter의 설계 철학과 일치한다.

또한 제안 모델은 NVIDIA Jetson Orin NX가 탑재된 옴니휠 로봇에서 평균 92.0%의 목표 도달 성공률을 달성하였다. 이를 통해 제한된 목표 도달 환경에서 대규모 VLM 없이도 실용적인 수준의 목표 지향 주행이 가능함을 확인하였다.

다만 본 연구의 성과는 사전에 정의된 소수의 고정 목표 지점과 제한된 어휘 조건에서 검증된 것이며, 학습 시 접하지 않은 물체나 표현에 대한 제로샷 일반화는 본 연구의 범위에 포함되지 않는다.

VI. Future Work

본 연구는 다음과 같은 한계를 가진다. 첫째, 제한된 어휘 조건과 사전에 정의된 5개의 고정 목표 지점을 대상으로 목표 도달 성능을 평가하였다. 그리하여 학습 과정에서 접하지 않은 새로운 물체나 표현에 대한 제로샷 일반화는 지원하지 않으며, 이는 본 연구의 주요 한계이다. 이러한 제약하에서 본 연구는 대규모VLM과 사전 구축 인프라에 대한 의존도를 낮추는 VLA 기반 경량의 목표 지향 내비게이션을 제시하는 데 초점을 두었다. 둘째, 사전 투어 영상이나 맵 구축 없이 새로운 환경에서 즉시 주행하는 tour-free/map-free 내비게이션은 연구 범위에서 제외하였다. 셋째, 실험 결과는 통제된 조건 하에서의 성능이며, 장애물이 존재하는 비통제 환경에 배포할 경우 충돌 위험이 존재한다. 향후 연구에서는 충돌 회피 알고리즘 등의 별도 안전 계층을 본 파이프라인과 결합하는 방향으로 확장할 계획이다.

www.kci.go.kr

전체 perception-to-action 파이프라인의 지연시간을 분석한 결과, FP32 환경에서 전체 추론 지연의 약 97%가 OWLv2 그라운딩 단계에서 발생함을 확인하였다. 이는 파라미터 수만으로 엣지 추론 성능을 판단하기 어렵고, 실제 연산 병목에 대한 최적화가 중요함을 보여준다. 향후에는 OWLv2 그라운딩 단계에 대한 양자화 및 지식 증류를 통해 추가적인 경량화를 수행하고, Raspberry Pi와 같은 초저자원 플랫폼으로의 확장 가능성을 검증할 예정이다.

향후 연구에서는 이러한 한계를 보완하기 위해 다음과 같은 방향으로 연구를 확장할 예정이다. 첫째는 OWLv2 그라운딩 단계에 양자화[23] 및 지식 증류[24]를 적용하여 추론 지연을 추가로 감소시킬 예정이다. 둘째는 Raspberry Pi와 같은 초저자원 플랫폼에 실제 이식하여 성능 및 자원 효율성을 벤치마크할 예정이다. 셋째는 텍스트 프롬프트를 다양화하여 언어가 행동 결정에 기여하는지를 검증하고, 이를 바탕으로 미학습 목표 표현에 대한 제한적인 제로샷 일반화 가능성까지 확인할 예정이다.

REFERENCES

[1] D. Shah, M. Equi, N. Sundaresan, F. Xia, F. Xia, C. Fu, A. Zeng, J. Tompson, C. Sun, D. Sadigh, J. Vasquez, K. Xu, T. Zhang, S. Levine, and Q. Vuong, "Mobility VLA: Multimodal Instruction Navigation with Long-Context VLMs and Topological Graphs," arXiv preprint arXiv:2407.07775, 2024.

[2] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling Open-Vocabulary Object Detection," arXiv preprint arXiv:2306.09683, 2023.

[3] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei, "Kosmos-2: Grounding Multimodal Large Language Models to the World," arXiv preprint arXiv:2306.14824, 2023.

[4] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "GNM: A General Navigation Model to Drive Any Robot," IEEE ICRA, 2023.

[5] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "ViNT: A Foundation Model for Visual Navigation," CoRL, 2023.

[6] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration," IEEE ICRA, 2024.

[7] S. Griebenow, O. Ozdemir, C. Weber, and S. Wermter, "Open-Vocabulary Robotic Object Manipulation using Foundation Models," Proc. of ESANN 2025, 2025.

[8] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby, "Simple Open-Vocabulary Object Detection with Vision Transformers," arXiv preprint arXiv:2205.06230, 2022.

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," arXiv preprint arXiv:2103.00020, 2021.

[10] Z. Peng et al., "Kosmos-2: Grounding multimodal large language models to the world," arXiv preprint arXiv:2306.14824, 2023.

[11] H. Gani, A. G. Perera, M. Naseer, S. Khan, and F. S. Khan, "Video OWL-ViT: Temporally-consistent open-world localization in video," arXiv preprint arXiv:2308.11093, 2023.

[12] R. Sapkota, M. Karkee, “Object detection with multimodal large vision-language models," Social Science Research Network (SSRN) / arXiv Collection, 2025/2026.

[13] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," arXiv preprint arXiv:2303.04137, 2023.

[14] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "π0: A Vision-Language-Action Flow Model for General Robot Control," arXiv preprint arXiv:2410.24164, 2024.

[15] NVIDIA, "Jetson Orin NX Series Technical Specifications," NVIDIA Developer Documentation, https://developer.nvidia.com /embedded/jetson-orin, 2026.

[16] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," arXiv preprint arXiv:2304.13705, 2023.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," arXiv preprint arXiv:1708.02002, 2017.

[18] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the Limitations of Behavior Cloning for Autonomous Driving," arXiv preprint arXiv:1904.08980, 2019.

[19] S. Ross, G. J. Gordon, and J. A. Bagnell, "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning," arXiv preprint arXiv:1011.0686, 2011.

[20] P. de Haan, D. Jayaraman, and S. Levine, "Causal Confusion in Imitation Learning," arXiv preprint arXiv:1905.11979, 2019.

[21] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti, S. Alibert, M. Cord, T. Wolf, and R. Cadène, "SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics," arXiv preprint arXiv:2506.01844, 2025.

www.kci.go.kr

[22] Y. Wang, P. Ding, L. Li, C. Cui, Z. Ge, X. Tong, W. Song, H. Zhao, W. Zhao, P. Hou, S. Huang, Y. Tang, W. Wang, R. Zhang, J. Liu, and D. Wang, "VLA-Adapter: An Effective Paradigm for Tiny-Scale Vision-Language-Action Model," arXiv preprint arXiv:2509.09372, 2025.

[23] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A Survey of Quantization Methods for Efficient Neural Network Inference," arXiv preprint arXiv:2103.13630, 2021.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," arXiv preprint arXiv:1503.02531, 2015.

Authors



Min-Woo Lee is currently an undergraduate student majoring in Artificial Intelligence in the Department of Applied Artificial Intelligence Engineering at Kangnam University, Korea, where he has been enrolled since 2022. Min-Woo Lee research interests include Physical AI, Embodied AI, vision-language-action (VLA) models, and AI Agent.



In-Yeop Choi received the B.S. degrees in Computer Science from Hanyang University in 2001 and the Integrated Master's-PhD degree in Computer Science from Korea University, Korea, in 2023.

Dr. Choi joined the faculty of the Department of Computer Science at Kangnam University, Gyeonggi-do, Korea, in 2023. He developed embedded systems at the company for 15 years. He is currently a Professor in the Department of Computer Science, Kangnam University. He is interested in Physical AI, Embodied AI, VLA(Vision Language Action), and Embedded System.

www.kci.go.kr