

YOLOE 기반 Text-Free 모델 협업 구조를 이용한 장단기 기억 기반 객체 탐지 시스템

Adaptive Object Detection with Long- and Short-Term Memory via YOLOE Text-Free Model Collaboration

김 지 호¹ · 광 정 훈² · 경 도 현³ · 구 재 완⁴ · 감 병 우⁵ · 이 준 우⁶ · 서 갑 호[†]
Jiho Kim¹, Jeonghoon Kwak², Dohyun Kyung³, Jaewan Koo⁴,
Byoung-Woo Gam⁵, Joonwoo Lee⁶, Kap-Ho Seo[†]

Abstract: Autonomous robots often operate in semi-structured environments such as orchards and warehouses, where unpredictable obstacles frequently appear. Conventional detectors relying on fixed classes cannot adapt to unknown objects, while continual or reinforcement learning approaches are impractical for robots with limited onboard resources. This paper presents a YOLOE-based Text-Free collaborative detection system with long- and short-term memory management. The Text model detects known objects, and the Free model is selectively activated for unknown objects, which are stored in short-term memory while critical classes remain in long-term memory. Implemented on a Jetson Orin Nano and tested in greenhouses, fields, and orchards, the system dynamically expanded its detectable object set without offline retraining. Results confirm reliable real-time obstacle response with reduced unnecessary stops, highlighting the framework's adaptability and practical value for autonomous robots in dynamic environments.

Keywords: Open-Vocabulary Object Detection, Long- and Short-Term Memory, Semi-Structured Environment, Autonomous Robot

1. 서 론

자율주행 로봇은 농업, 물류, 서비스 산업 등 다양한 분야에서 활용 가능성이 점차 확대되고 있다^[1-3]. 특히 과수원, 비닐하우스

스, 창고와 같은 반구조적(semi-structured) 환경은 로봇의 도입 효과가 크지만, 전형적인 도로 환경과 달리 일정한 구조가 존재하면서도 배치가 시시각각 변화하고 예기치 못한 장애물이 빈번히 출현한다는 특성을 가진다. 따라서 이러한 환경에서 로봇이 안정적으로 동작하기 위해서는 단순한 경로 추종이나 정적 장애물 회피를 넘어, 동적으로 변화하는 환경에 적응할 수 있는 객체 탐지 및 대응 능력이 필수적이다.

기존 연구들은 이러한 문제를 해결하기 위해 여러 접근을 시도해 왔다^[4,5]. 고정 클래스 기반 객체 탐지기는 사전에 정의된 클래스 집합을 학습하여 높은 정확도로 탐지를 수행하지만, 실제 주행 환경에서 등장하는 미지 객체(Unknown object, UO)에 대해서는 대응이 어렵다^[6,7]. 그 결과 새로운 장애물은 인식하지 못하거나, 반대로 무해한 물체(예: 풀, 천막 등)에 불필요하게 반응하여 주행 효율성을 저해한다. 센서 융합 기반 접근 역시 LiDAR와 RGB-D 카메라를 결합하여 장애물의 거리, 형태, 질감을 분석함으로써 대응 성능을 높이려 하였으나, 여전히 인식 가능한 객체가 학습 시점에 한정된다는 제약이 따른다^[8,9]. 최근 주목받는

Received : Sep. 23, 2025; Revised : Mar. 11, 2026; Accepted : May. 14, 2026

※ This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs (MAFRA) (02219411).

1. Assistant Manager, Daedong Robotics (DDR), Seoul, Korea (kimjiho@daedong.co.kr)
2. Senior Researcher, Advanced Institute of Convergence Technology (AICT), Suwon, Korea (jeonghoon@snu.ac.kr)
3. Researcher, Korea Institute of Robotics and Technology Convergence (KIRO), Seoul, Korea (kyoungdh@kiri.re.kr)
4. Senior Researcher, Korea Institute of Robotics and Technology Convergence (KIRO), Pohang, Korea (jwko03236@kiri.re.kr)
5. Executive Director, Daedong Robotics (DDR), Seoul, Korea (bwgam@daedong.co.kr)
6. Associate Professor, Department of Electrical Engineering, Kyungpook National University, Daegu, Korea (jwl@knu.ac.kr)

† Chief Researcher, Corresponding author: Korea Institute of Robotics and Technology Convergence (KIRO), Pohang, Korea (neoworld@kiri.re.kr)

지속적 학습(continual learning)^[10]이나 강화학습(reinforcement learning)^[11,12] 기반 기법은 환경 적응성을 개선할 수 있으나, 대규모 데이터와 연산 자원을 요구하거나 실제 환경에서 충돌을 수반하는 학습을 수행해야 하는 한계가 있다. 특히 자율주행 로봇은 제한된 하드웨어 리소스로 동작하기 때문에 실시간 학습을 위한 고비용 연산을 수행하기 어렵고, 모든 상황을 포괄하는 방대한 데이터를 메모리에 상시 탑재하는 것도 현실적으로 불가능하다.

이러한 배경에서 본 논문은 YOLOE^[13] 기반 Text-Free 협업 구조와 기억 관리 전략을 결합한 새로운 객체 탐지 프레임워크를 제안한다. 본 논문에서 Text 모델이란 메모리에 저장된 클래스 이름을 텍스트 프롬프트로 입력받아 해당 객체만을 탐지하는 모델을 의미하며, Free 모델이란 별도 클래스 지정 없이 장면 내 모든 객체를 자율적으로 탐지하는 모델을 의미한다. 제안 시스템은 Text 모델을 통해 알려진 객체를 효율적으로 탐지하고, Free 모델을 필요 시 보조적으로 활성화하여 UO를 탐지한 뒤 단기 기억(Short-Term, ST)에 반영한다. 장기 기억(Long-Term, LT)에는 사람이나 농기계처럼 반드시 인식해야 하는 핵심 객체를 지속적으로 유지하며, 단기 기억은 일정 시간이 지나면 삭제되어 메모리 효율을 보장한다. 이러한 구조는 추가 학습 과정 없이도 탐지 대상을 유연하게 확장할 수 있으며, 불필요한 연산 낭비를 최소화하면서도 환경 변화에 적응할 수 있도록 한다.

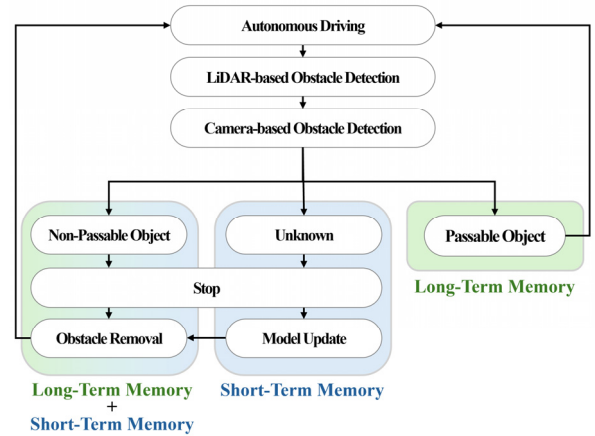
본 논문의 주요 기여는 세 가지로 요약된다. 첫째, 오프라인 재학습 없이도 새로운 객체를 즉시 반영할 수 있는 적응형 탐지 구조를 제안한다. 둘째, 탐지 가능한 클래스 집합을 동적으로 확장함으로써 기존 센서 융합 방식보다 높은 환경 적응성을 확보한다. 셋째, 장·단기 기억 기반 메모리 관리 전략을 도입하여 제한된 하드웨어 환경에서도 실시간 탐지를 구현한다. 이를 통해 로봇은 다양한 환경에서 발생하는 장애물에 대해 신뢰성과 유연성을 동시에 확보할 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 제안하는 YOLOE 기반 Text-Free 협업 구조와 장단기 기억 관리 메커니즘을 설명하고, 3장에서는 이를 기반으로 한 장애물 대응 제어 절차를 기술한다. 4장에서는 다양한 환경에서 수행한 실험 결과와 성능 평가를 제시하며, 5장 결론에서는 본 연구의 의의와 향후 연구 방향을 논의한다.

2. YOLOE 기반 객체 탐지 시스템

2.1 시스템 개요

본 논문에서 제안하는 객체 탐지 시스템은 [Fig. 1]과 같이 YOLOE 기반 Text-Free 협업 구조와 장단기 기억 관리 전략을 결합하여, 로봇이 주행 중 직면하는 다양한 장애물에 실시간으로 대응할 수 있도록 한다.



[Fig. 1] The overview of the obstacle detection system

로봇은 전방 LiDAR로 장애물 존재 여부를 감지한 후, 카메라 영상을 YOLOE 모델로 분석한다. Text 모델은 알려진 객체를 빠르게 탐지하며, 만약 탐지 결과가 없을 경우 Free 모델이 조건부로 호출되어 UO를 인식한다. Free 모델을 통해 새롭게 발견된 객체는 단기 기억에 임시 저장되며, 이후 Text 모델이 해당 객체를 탐지할 수 있도록 메모리가 갱신된다. 장기 기억은 사람, 농기계 등 반드시 인식해야 하는 핵심 객체를 유지하여 안정성을 보장한다. 이러한 절차를 반복함으로써 시스템은 환경 변화에 적응하면서도 불필요한 연산 자원 낭비를 최소화한다.

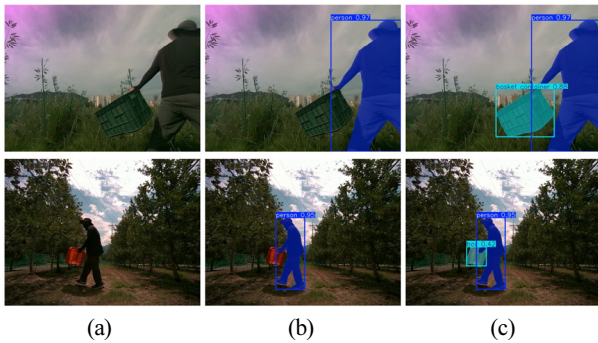
2.2 YOLOE와 추론 모드

YOLOE는 기존 YOLO 시리즈가 가진 폐쇄형 어휘(closed vocabulary) 한계를 극복한 개방형 어휘(open vocabulary) 객체 탐지 모델이다. Text 프롬프트 모드에서는 메모리에 저장된 클래스만 탐지하여 효율성을 높이고 오탐을 줄일 수 있다. 반면 Prompt-Free 모드에서는 프롬프트 없이 장면 전체를 탐지하여 새로운 객체를 인식할 수 있다.

본 논문에서는 두 모드를 상호 보완적으로 사용한다. 평상시에는 Text 모델을 사용하여 알려진 객체를 효율적으로 탐지하고, LiDAR에서 장애물이 감지되었으나 카메라에서 탐지가 이루어지지 않을 경우 Free 모델이 호출되어 새로운 객체를 탐지한다. 이렇게 발견된 객체는 ST로 등록되어 이후 Text 모델의 탐지 대상이 된다. [Fig. 2]는 Free 모델을 통해 신규 객체가 등록되고 이후 Text 모델에서 동일 객체가 인식되는 과정을 보여준다.

2.3 장애물 분류와 메모리 관리

탐지된 객체는 특성에 따라 구분된 후 메모리에 저장·관리되어, 로봇이 환경 변화에 적응할 수 있도록 설계되었다. 객체의 구분은 [Table 1]과 같이 크게 알려진 객체(Known Object, KO)



[Fig. 2] Example of inference results before and after short-term memory update: (a) Original image, (b) Result before Text model update, (c) Result after Text model update

와 미지 객체로 나뉘며, 보조적으로 무시 객체(Ignorable Object, IO)와 비객체(Non-Object, NO)가 포함된다. 메모리 관리 관점에서는 KO, IO, NO가 LT 객체로 관리되며, UO는 ST 객체로 관리된다.

KO는 로봇이 항상 인식해야 하는 핵심 객체 집합으로, 사람이나 농기계와 같이 안전에 직접적인 영향을 미치는 대상이다. 이들은 시스템 구동 전부터 정의되어 있으며, 주행 내내 유지된다. 반면 UO는 주행 과정에서 새롭게 발견된 객체가 임시로 저장되는 영역으로, Free 모델을 통해 등록된다. 이는 일정 시간 동안만 유지되며, 이후 자동으로 삭제되어 메모리 효율성을 확보한다. IO는 탐지되더라도 로봇의 동작에 직접적인 영향을 주지 않는 대상이다. 낙엽, 풀, 얇은 가지처럼 실제로 존재하지만 주행을 멈출 필요가 없는 객체들이 여기에 포함된다. 마지막으로 NO는 하늘이나 그림자처럼 물리적 실체가 없는 배경 요소로, 탐지 과정에서 필터링된다.

앞서 정의한 장애물 분류는 CSV 파일을 통해 관리되며, Text 모델은 이를 참조하여 객체 탐지를 수행한다. CSV 파일은 총 네 개의 필드로 구성되어 있으며(Table 2), 각 장애물의 카테고리, 인덱스, 명칭, 그리고 등록 시각을 포함한다. 이 중 등록 시각을 나타내는 update_time 속성은 마주한 장애물의 최신성을 나타내는 지표로 활용되며 단기 기억 메모리 업데이트를 위한 핵심 장치로 기능한다. 로봇의 주행 시 KO, UO, IO에 해당하는 클래스 명칭은 프롬프트에 상시 반영되는 반면, 주행과 무관한 NO 클래스는 프롬프트에서 제외되어 탐지 대상에서 배제된다.

2.4 단기 기억 업데이트 메커니즘

본 논문에서 제안하는 시스템의 핵심 기능 중 하나는 단기 기억 객체의 업데이트이다. 이 절차는 LiDAR에서 장애물의 존재가 감지되었음에도 불구하고, YOLOE Text 모델에서 해당 객체가 인식되지 않을 때 수행된다.

[Table 1] Taxonomy of obstacle types and corresponding robot behaviors

Type	Category	Examples	Memory	Robot action
Known Object (KO)	Human	adult, child	LT (updated by server)	Stop
	Vehicle, Machinery, Structure	tractor, vehicle, ladder, fence, rock		Stop (until confirmed)
Ignorable Object (IO)	Vegetation / Minor	grass, small branches, plantations		Pass-through / Slow
Non-Object (NO)	Background / Scene	sky, shadow, background features		Ignore (no action)
Unknown Object (UO)	Detected by Free model	temporary tools, newly observed items	ST (updated by itself)	Stop (until confirmed)

[Table 2] Structure of the CSV file defining obstacle classes

Field Name	Description	Example
category	Type of obstacle among the five categories	NO
object_id	Index number assigned to each class	NO_000
object_name	Name of the class	person
update_time	Timestamp when the entry was registered or updated in the CSV file	1756281666

이 경우 로봇은 안전을 위해 즉시 정지하고, YOLOE를 Free 모델 모드로 전환하여 전방 장면을 재분석한다. Free 모델이 신뢰도 높은 새로운 객체를 탐지하면 해당 객체는 단기 기억 객체로 등록된다. 이후 Text 모델은 갱신된 메모리를 기반으로 동일 객체를 탐지할 수 있게 되며, 주행 중 동일 객체가 다시 나타날 경우 Free 모델 호출 없이도 즉시 인식이 가능하다.

단기 기억 장애물은 최대 100개까지 저장하여 관리하며, update_time 속성을 기준으로 가장 오래된 항목부터 자동으로 삭제되도록 설계하였다. 이를 통해 시스템은 새로운 장애물에 신속하게 적응하면서도, 일시적이거나 특수한 장애물로 인해 메모리가 불필요하게 누적되는 것을 방지하여 메모리 효율성을 유지할 수 있다. 이와 같은 단기 기억 업데이트 메커니즘은 로봇이 미지 객체를 실시간으로 수용하고 즉각적으로 탐지 범위를 확장할 수 있도록 하며, 제한된 연산 자원 환경에서도 안정적으로 동작할 수 있게 한다.

3. Text-Free 협업 구조 기반 제어 방법

본 장에서는 제안한 YOLOE 기반 객체 탐지 시스템을 실제 로봇 주행 제어에 적용하는 절차를 설명한다. 제안된 절차는 LiDAR 기반의 장애물 감지, 카메라 기반의 객체 판정, Text-

Free 협업 구조에 의한 미지 객체 대응을 거쳐 최종적으로 주행 제어로 이어진다([Table 3]).

로봇은 먼저 LiDAR 센서를 통해 전방 주행 영역의 장애물 존재 여부를 신속하게 판정한다. LiDAR 단계에서는 ROI (Region of Interest)를 사전 정의하여, 영역 내 LiDAR 스캔 포인트의 수가 거리 기반 동적 임계값을 초과하면 장애물이 존재하는 것으로 판정한다. 장애물이 감지되면 카메라 처리 단계로 전달한다.

카메라 단계에서는 YOLOE Text 모델을 활용하여 ROI 내 객체의 의미를 판정한다. Text 모델은 KO, UO, IO 클래스 집합을 프롬프트로 사용하여 탐지를 수행하며, 인식된 객체는 기억 관리 모듈에 매핑된다. KO 또는 UO로 판정된 객체는 정지 대상으로 처리되고, IO 객체는 무시 또는 저속 통과 대상으로 간주된다.

만약 LiDAR가 장애물 존재를 보고했음에도 Text 모델에서 어떤 클래스에도 매핑되지 않는 경우, 해당 객체는 Unknown으로 처리된다. 이때 로봇은 안전을 위해 정지 상태로 전환하고, YOLOE Free 모델을 조건부 호출하여 ROI를 재분석한다. Free 모델이 신뢰도 높은 결과를 제공하면 해당 객체는 단기 기억에 등록되고, 이후부터는 Text 모델 탐지 과정에 포함된다. 반대로 Free 모델에서도 유효한 객체가 검출되지 않으면, 로봇은 일정 시간 정지 상태를 유지한 뒤 장애물이 해소된 것으로 판단하고 주행을 재개한다.

최종적으로 로봇은 LiDAR-카메라-기억 관리 모듈의 결과를 종합하여 주행 정책을 결정한다. LT 또는 ST 객체가 탐지되면 즉시 정지하며, IO 객체만 존재하는 경우에는 저속 주행을 수행한다. 장애물이 사라지면 정상 주행으로 복귀한다. 이러한 절차를 통해 로봇은 불필요한 정지를 최소화하면서도 안전성을 확보할 수 있다.

4. 실험 결과 분석

본 장에서는 제안한 YOLOE 기반 Text-Free 협업 구조와 장단기 기억 기반 객체 탐지 시스템의 성능을 실제 로봇 플랫폼과 환경에서 검증한 결과를 서술한다. 실험은 자율주행 로봇 RT100-TA(대동로보틱스社)를 대상으로 수행되었으며, 농업 현장에서 농자재 이송, 수확물 운반, 방제 보조 등의 작업을 수행하는 자율주행 농업 로봇을 활용하였다. 또한 예측 불가능한 객체가 수시로 출현하는 과수원 환경에서 다양한 장애물 조건을 구성하여 제안 시스템의 동작을 검증하였다.

4.1 로봇시스템의 구성

RT100-TA은 소형 전동 구동 기반의 실외 주행 로봇으로, 전방 장애물 인식을 위해 2D LiDAR (RPLiDAR S3)와 RGB-D 카메라(Intel D456)를 탑재하였다. 또한 정밀한 위치 측정을 위해

[Table 3] Pseudocode of the Text-Free Object Detection Pipeline

```

procedure OBSTACLE_HANDLING_PIPELINE():
  # 1. LiDAR-based obstacle detection
  G ← LiDAR.detect()
  if G.exists == false:
    return {found: false}

  # 2. ROI setting (projecting LiDAR boundaries onto the camera)
  ROI ← project_to_camera(G)

  # 3. Camera-based Text model detection
  prompts ← KO ∪ UO ∪ IO
  preds ← YOLOE_TEXT(ROI, prompts)

  valid ← []
  for p in preds:
    if p.conf <  $\theta_{text}$ : continue
    if not depth_check(p.region): continue
    mem_cls ← MEMORY_LOOKUP(p.label)
    valid.append({label: p.label, class: mem_cls})

  # 4. Text model result evaluation
  if valid is not empty:
    p* ← select_nearest(valid)
    return {found: true, label: p*.label, class: p*.class}

  # 5. Unknown object handling → Call Free model
  hypo ← YOLOE_FREE(ROI)
  if hypo.conf ≥  $\theta_{free}$  and hypo.label not in {KO, UO, IO, NO}:
    ST.upsert(label=hypo.label, timestamp=now())
    return {found: true, label: hypo.label, class: "UO"}

  # 6. Still unrecognized → Maintain stop
  return {found: false}

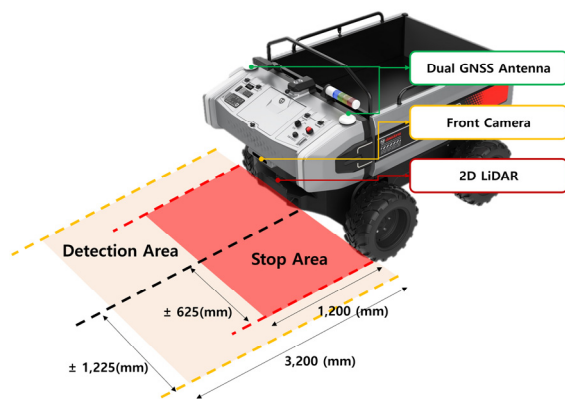
```

RTK-GPS 모듈(H&I社)을 장착하였다. LiDAR는 주행 유효 영역 내 장애물의 존재를 빠르게 판정하고, RGB-D 카메라는 YOLOE 기반 탐지 및 깊이 정보를 활용하여 장애물의 성격을 판별한다. [Fig. 3]과 같이 RT100-TA에 장애물 및 위치를 인식하기 위한 센서가 부착되어 있다.

로봇의 주행 유효영역(E)는 정지영역과 감지영역으로 두 가지 구분이 있다. 로봇의 폭과 속도를 고려하여 정지영역은 로봇의 중심기준으로 ± 625 mm의 폭과 1,200 mm 거리로 설정되었으며, 해당 부분에서는 로봇이 정지를 수행한다. 감지영역은 $\pm 1,225$ mm의 폭과 3,200 mm 거리로 설정되었으며, 장애물과의 거리를 고려해서 서서히 감속하는 영역이다.

4.2 실험 환경

실험 환경은 [Fig. 4]와 같이 나무 열이 평행하게 배치된 통로 구조와 불균일한 잔디 및 풀이 존재하는 실제 농장 과수원으로 설정되었다. 과수원 통로의 폭은 일반적으로 수 미터 내외이며, 나무 열 간격과 지면의 굴곡은 구간마다 차이가 있어 주행 안정



[Fig. 3] Robot platform, RT100-TA



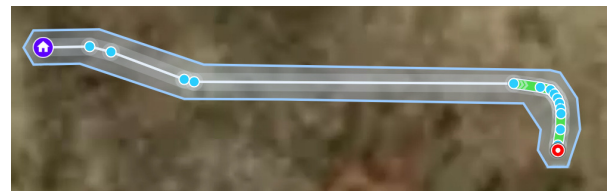
[Fig. 4] Experimental scenes: (a) RT100-TA in operation, (b) surrounding environment of the robot

성에 영향을 미친다. 또한 잔디와 풀의 높이가 일정하지 않아 저상·중상 높이의 물체를 잘못 장애물로 인식하는 경우가 발생할 수 있다. 환경에는 지주, 지압선, 관수 설비, 말뚝, 방풍망과 같은 고정 구조물이 항상 존재하며, 천막이나 비닐벽과 같은 임시 구조물은 시간이나 작업 상황에 따라 출현하거나 이동한다. 더불어 작업자, 소형 차량, 사다리과 같은 동적 객체도 수시로 등장하거나 배치가 변동되므로, 다양한 형태의 장애물이 함께 고려되어야 하는 도전적인 조건을 제공한다.

과수원 환경에서는 [Fig. 4]의 (b)와 같이 제초 작업이 오래 수행되지 않으면 풀이 과도하게 성장하여 LiDAR가 이를 장애물로 오인식하는 사례가 관찰된다. 또한 바람에 의한 풀의 움직임과 먼지 입자는 LiDAR 및 깊이 센서에 간헐적인 노이즈를 유발하여, 실제 장애물이 존재하지 않음에도 불구하고 장애물로 탐지되는 상황이 발생할 수 있다. 더 나아가, 과수원과 유사한 구조를 가진 노지 환경에서는 [Fig. 5]와 같이 경로를 설정하고, 시나리오별 주행을 통해 장애물 객체 인식을 수행하였다.

4.3 장애물 객체 설정

실험에 사용된 객체는 네 가지 범주(KO, UO, IO, NO)로 분리되었다. KO는 사람, 트랙터, 사다리, 차량 등 안전상 반드시



[Fig. 5] Path setup in the orchard

탐지해야 하는 214개 클래스를 포함하였다. UO는 사전에 정의된 범주가 아니라, 자율주행 중 Text 모델이 인식하지 못한 객체가 발생했을 때 Free 모델을 통해 보충되는 클래스이다.

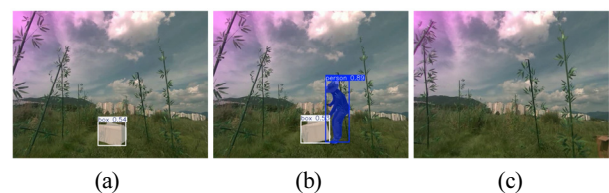
IO는 풀, 잔디, 작은 나뭇가지처럼 탐지되더라도 주행에 직접적인 영향을 주지 않는 30개 클래스로 구성되었다. NO는 하늘, 구름, 지형 등 총 322개 클래스를 포함하며, 실제 장애물이 아닌 배경 요소로 분류되어 탐지 과정에서 필터링된다.

4.4 실험 분석

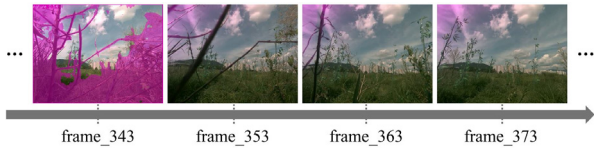
제한한 라이더 전처리-YOLOE Text-Free 협업-LT/ST 메모리 파이프라인이 실제 과수원 주행 환경에서 일관되게 동작하는지를 상황별로 확인하였다. LiDAR가 주행 유효영역 내 존재 여부를 먼저 보고하고, 카메라 경로는 이미지의 객체 판정과 깊이 검증을 수행하였다. 로봇의 움직임에 대한 정책은 최종적으로 KO/UO/IO 매핑 결과에 따라 정지/저속/재개를 선택하였다.

(1) KO/UO 존재 시 정지 및 재개: LiDAR가 전방 존재를 보고하고, RGB-D 경로에서 KO 또는 UO로 판정된 경우 로봇은 즉시 정지하였다. 장애물이 제거되거나 주행 유효영역(E)의 정지영역 밖으로 이탈하면, 동일 위치에서 정지 해제 후 재개가 이루어졌다. [Fig. 6]과 같이 이는 사람이 동체(예: person), 작업 장비(예: tractor, ladder)와 같은 고위험·가변 객체에 대해 정책이 보수적으로 작동함을 보여준다.

(2) IO만 존재 시 무시/저속 통과: [Fig. 7]와 같이 LiDAR가 존재를 보고했으나 KO/UO는 검출되지 않고 IO만(예: power pole, plantation)이 전방에 분포한 경우, 시스템은 IO로 매핑된 객체를 정지 사유로 사용하지 않고 저속 통과를 선택하였다. 이



[Fig. 6] Robot response to the presence and removal of LT obstacles: (a) the robot stops due to a box on the path, (b) the obstacle is removed by a person (also an LT obstacle), (c) the robot resumes autonomous driving after removal

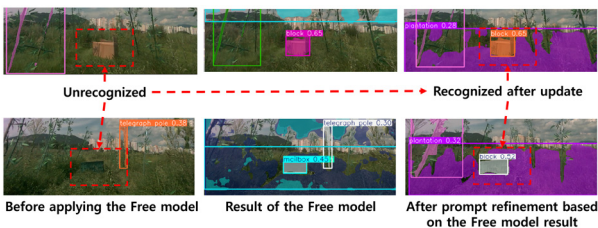


[Fig. 7] Autonomous driving through an IO at low speed

때 IO가 장애물의 반경 안에 있더라도 IO 정책 우선순위에 의해 정지로 전환되지 않았다. 해당 결과는 식생·잔재로 인한 불필요 정지 억제라는 설계 목표를 현장에서 확인하였다.

(3) UO 객체 갱신: 알지 못한 객체에 대한 성능 개선 테스트를 위해서 사람 및 과수를 제외한 과수환경에서 나타날 수 있는 다른 물체들을 KO로 설정하지 않고 테스트를 수행하였다. [Fig. 8]의 사례에서 LiDAR가 장애물 존재를 보고하였고, 전방 주행 유효영역(E) 내에 KO/UO/IO가 반경 내에 없는 조건에서, 카메라 경로는 초기 프레임에서 의미 미정 상태를 보고하였다. 이때 자율주행 정책은 정지를 유지한 채 Free 모델을 조건부 호출하였고, Free 결과로 block(박스) 라벨이 제안되어 UO 객체에 신규 등록되었다. 동일 구간 재주행 시에는 Text 모델에 UO 객체가 추가된 프롬프트를 즉시 활용하여 Free 호출 없이 곧바로 단기 위험으로 판정하였고, 객체 재발견 시 UO의 타임스탬프가 갱신됨을 확인하였다. 이는 제안한 UNKNOWN→Free→ST→Text 전환 루프가 현장에서 지연을 최소화하며 적응한다는 점을 보여주었다. 하지만 block이 사전에 설정되지 않았지만 주행 유효영역(E) 안에 새로운 객체가 나타나 Free 모델로 설정되어 추가되었다. Free 모델에서 mailbox로 인지되었지만 최종적으로 이전의 block로 설정된 확률이 더 높아 block로 출력되는 것을 확인하였다. 사전에 정의되지 않은 객체를 UO로 설정 가능하다는 것을 볼 수 있다.

(4) 모듈간 연계성: 모든 시나리오에서 LiDAR의 장애물 존재 검출 → ROI 투영 → Text/거리 검출 → KO/UO/IO 패핑 → 정책의 실행 순서가 유지되었고, Free 모델은 정지 상태에서만 호출되었다. 이러한 결과는 제안한 시스템이 다양한 환경적 조건에서 실시간 적응형 탐지를 수행할 수 있음을 보여준다. 특히 IO 설정을 통해 불필요한 정지를 줄일 수 있었으며, Free 모델과 UO 업데이트 과정을 통해 탐지 가능한 객체 집합을 즉시 확장할 수 있었다.



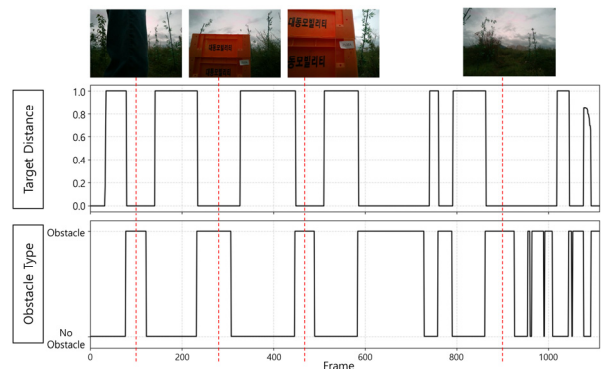
[Fig. 8] Text model updated by Free model

이러한 결과는 제안한 시스템이 다양한 환경적 조건에서 실시간 적응형 탐지를 수행할 수 있음을 보여준다. 특히 IO 정책을 통해 불필요한 정지를 줄일 수 있었으며, Free 모델과 UO 업데이트 과정을 통해 탐지 가능한 객체 집합을 즉시 확장할 수 있었다.

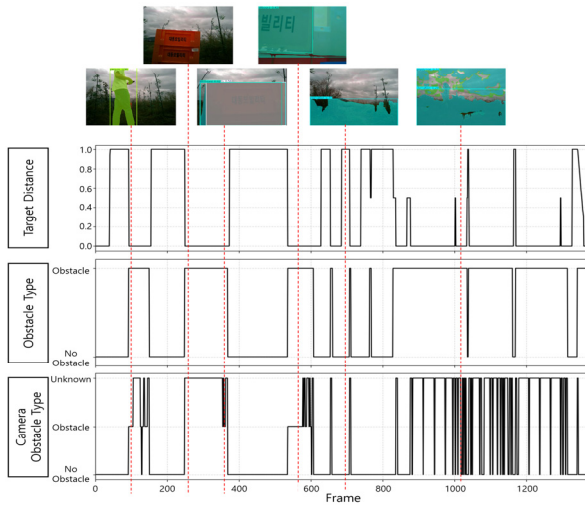
제안한 방법을 검증하기 위한 시나리오는 제초 작업이 이루어지지 않아 풀이 자란 환경에서 로봇의 이동 경로에 상자가 놓여 있는 상황을 가정하였다. 로봇이 상자에 도달하기 전에는 작업자가 경로에서 서 있다가 작업자가 사라지면 로봇의 진행 방향에 상자가 나타나게 된다. 로봇은 장애물로 인해 정지하고 작업자가 이를 확인한 후 장애물을 치운다. 이후 또 다른 작업자가 경로에 상자를 다시 배치하면 로봇은 다시 정지하게 되고, 작업자가 상자를 제거한 후 로봇은 이동을 재개한다. 최종적으로 로봇은 제초작업이 덜 이루어진 구역에 도착한 뒤, 최종 목적지로 이동하는 시나리오이다.

시나리오 기반으로 LiDAR만을 사용하여 주행한 결과는 [Fig. 9]에 나타나 있다. 로봇은 장애물에 대해 정지 동작을 수행하였으나, 풀과 같이 무시하고 통과할 수 있는 객체에도 반응하여 불필요하게 정지하는 한계를 보였다.

제안한 방법을 적용하여 첫 번째 구동을 수행한 결과는 [Fig. 10]과 같다. 작업자는 사전에 장애물로 등록되어 있어 정지 동작이 정상적으로 수행되었다. 상자는 초기에는 등록되지 않았기 때문에 인식되지 않았으나, 이후 Free 모델을 통해 상자가 trailer truck, ticket booth, billboard의 객체로 Text 모델이 갱신되면서 상자가 장애물로 인식되는 과정을 확인할 수 있었다. 두 번째로 상자가 등장했을 때는 Text 모델이 이미 업데이트되어 있었기 때문에 정상적으로 인식되었다. 풀이 높게 자란 구간에서는 로봇의 전방 이동 명령 거리 값(Target Distance)이 절반으로 조정된 상태에서, 로봇이 정지하지 않고 주행을 지속함을 확인할 수 있다. 마지막 구간에서 UO가 다수 발생한 것은 IO 객체의 인식 정확도가 낮아 주행 속도를 높이지 못한 문제점을 가졌다.



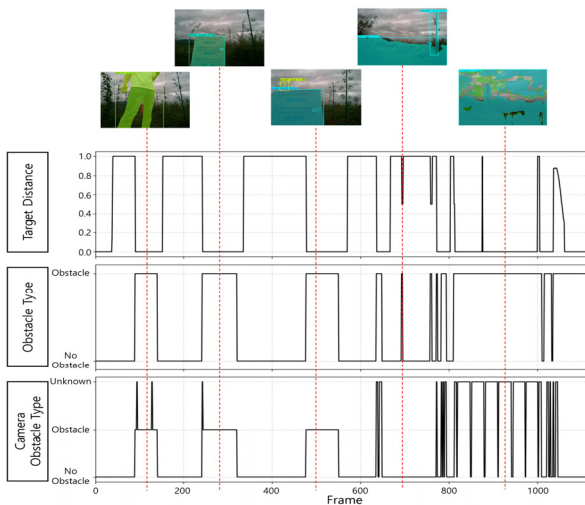
[Fig. 9] Driving result using LiDAR-based obstacle recognition



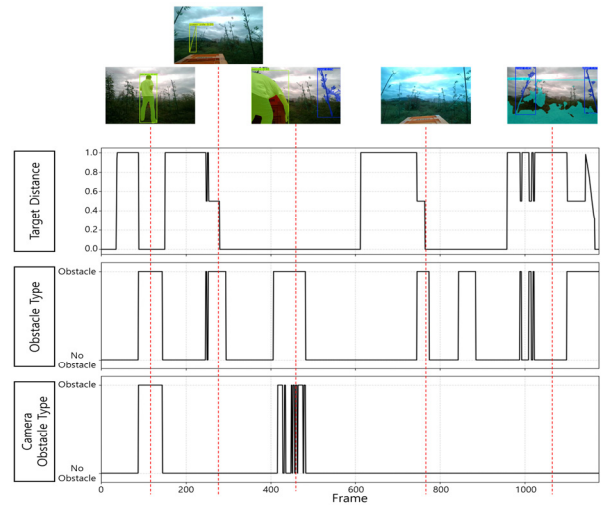
[Fig. 10] First driving result using the proposed method

동일한 시나리오에서 제안한 방법으로 두 번째 구동을 수행한 결과는 [Fig. 11]과 같다. 상자가 이미 장애물로 등록되어 있었기 때문에 Text 모델에서 즉시 인식되었다. 다만 일부 각도에서는 상자가 미인식되었으나, Free 모델을 통해 Text 모델이 보완 업데이트되면서 ‘block’으로 등록되는 것을 확인할 수 있었다.

고정된 장애물 인식 모델을 사용하는 기존 방법으로 주행한 결과는 [Fig. 12]와 같다. 작업자는 사전 등록되어 있어 정상적으로 인식되었으나, 학습되지 않은 상자는 인식되지 않았다. LiDAR는 상자를 장애물로 판정하여 속도를 줄이기는 했지만, 카메라 기반 장애물 인식에서는 앞의 상자를 인식하지 못해 충돌 두 차례 발생하였다. 제초작업이 이뤄지지 않은 구간에서는 장애물이 없다고 판단하여 속도를 낮추며 통과하는 양상을 보였다.



[Fig. 11] Second driving result using the proposed method



[Fig. 12] Driving result using an existing camera-based obstacle recognition method

[Table 4] Comparison of the number of collisions between the existing method and the proposed method

Iteration	Potential collisions	Existing method	Proposed method	
		Actual collisions	Actual collisions	Number of Free model calls (details)
1	3	2	0	20 (trailer truck, ticket booth, billboard)
2	3	2	0	9 (block)

제안한 방법과 카메라를 활용한 기존 방법의 비교 결과는 [Table 4]에 정리하였다. 충돌 가능 횟수는 총 3회였다. 기존 방법에서는 2회의 충돌이 발생하였지만, 제안한 방법에서는 충돌이 발생하지 않았다. 제안한 방법을 활용한 첫 번째 주행에서는 Free 모델이 총 20회 호출되어 3종 객체가 신규 UO로 등록되었다. 두 번째 주행에서는 해당 객체들이 업데이트 과정 없이 즉시 인식되었으며, Free 모델 호출 횟수가 9회로 감소하고 1종의 객체가 추가 등록되었다. 즉, 제안한 방법은 반복 주행을 통해 로봇의 탐지 가능 영역이 점진적으로 확장되며 환경 내 적응 비용이 감소하는 효율성을 보였다.

5. 결론 및 향후 연구

본 연구에서는 YOLOE 기반 Text-Free 협업 구조와 장·단기 기억 관리 전략을 결합한 새로운 객체 탐지 프레임워크를 제안하였다. 제안 시스템은 기존 고정 클래스 기반 탐지기가 가진 한계, 즉 미지 객체에 대한 즉각적인 대응이 불가능하다는 문제를 해결하기 위해 설계되었다. Text 모델은 알려진 객체를 효율

적으로 탐지하고, Free 모델은 필요 시에만 조건부로 활성화되어 새로운 객체를 탐지한 뒤 단기 기억에 반영한다. 장기 기억은 반드시 유지되어야 하는 핵심 객체를 지속적으로 관리하고, 단기 기억은 일정 시간이 지나면 삭제되어 메모리 효율성을 확보한다. 이러한 구조를 통해 시스템은 오프라인 재학습 없이도 탐지 가능한 클래스 집합을 동적으로 확장하며, 제한된 하드웨어 리소스 환경에서도 실시간 적응성을 보장할 수 있었다.

실험은 Jetson Orin Nano 플랫폼을 기반으로 비닐하우스, 개활지, 과수원 등 다양한 비정형 환경에서 수행되었다. 결과적으로, 기존 모델에서 탐지되지 않았던 객체들이 Free 모델과 기억 관리 과정을 통해 Text 모델의 탐지 범위에 즉시 반영되는 것을 확인하였다. 이를 통해 로봇은 불필요한 정지를 줄이면서도 안전성을 유지할 수 있었으며, 제안 방식이 실제 주행 환경에서 실시간 적응형 장애물 대응을 효과적으로 달성함을 입증하였다.

본 연구는 자율주행 로봇의 인지 체계에서 추가 학습 없이도 새로운 객체를 반영할 수 있는 실시간 적응형 인식 전략을 제시했다는 점에서 의미가 크다. 이는 농업, 물류, 실내외 서비스 로봇 등 다양한 응용 환경에서 안정성과 실용성을 동시에 확보할 수 있는 기반 기술로 활용될 수 있다.

향후 연구에서는 첫째, 단기 기억의 유지 방식과 삭제 조건을 다양한 환경 요인에 따라 최적화하여 메모리 관리의 안정성과 신뢰성을 강화할 필요가 있다. 둘째, 3D LiDAR나 초음파 센서 등 다른 센서 모달리티와의 융합을 통해 인식 신뢰도를 높이고, 보다 복잡한 환경에서도 안정적인 성능을 보장할 수 있도록 확장할 계획이다.

References

- [1] X. Li, N. Ma, Y. Han, S. Yang, and S. Zheng, "AHPPEBot: Autonomous Robot for Tomato Harvesting based on Phenotyping and Pose Estimation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, pp. 18150-18156, 2024, DOI: 10.1109/ICRA57147.2024.10610454.
- [2] T. Lackner, J. Hermann, C. Kuhn, and D. Palm, "Review of autonomous mobile robots in intralogistics: state-of-the-art, limitations and research gaps," *Procedia CIRP*, vol. 130, pp. 930-935, 2024, DOI: 10.1016/j.procir.2024.10.187.
- [3] Y. Yan et al., "The Navigation of Home Service Robot Based on Deep Learning and Machine Learning," *Journal of Robotics*, vol. 2024, Article ID 5928227, 2024, DOI: 10.1155/2024/5928227.
- [4] H. Li, K. Huang, Y. Sun, X. Lei, Q. Yuan, J. Zhang, and X. Lv, "An autonomous navigation method for orchard mobile robots based on octree 3D point cloud optimization," *Frontiers in Plant Science*, vol. 15, 2025, DOI: 10.3389/fpls.2024.1510683.
- [5] T. N. Syed et al., "Enhancing Autonomous Orchard Navigation: A Real-Time Convolutional Neural Network-Based Obstacle Classification System for Distinguishing 'Real' and 'Fake' Obstacles in Agricultural Robotics," *Agriculture*, vol. 15, no. 8, pp. 827-856, 2025, DOI: 10.3390/agriculture15080827.
- [6] N. Adiuku, N. P. Avdelidis, G. Tang, A. Plastropoulos, and Y. Diallo, "Mobile robot obstacle detection and avoidance with NAV-YOLO," *International Journal of Mechanical Engineering and Robotics Research*, vol. 13, no. 2, pp. 219-226, 2024, DOI: 10.18178/ijmerr.13.2.219-226.
- [7] M. Y. Park, W. Jung, M. Kim, and K. M. Park, "Robot Task Planning Based on Voice Command and Object Detection," *The Journal of Korea Robotics Society*, vol. 19, no. 4, pp. 347-353, 2024, DOI: 10.7746/jkros.2024.19.4.347.
- [8] H. Liu, C. Wu, and H. Wang, "Real time object detection using LiDAR and camera fusion for autonomous driving," *Scientific Reports*, vol. 13, no. 8056, 2023, DOI: 10.1038/s41598-023-35170-z.
- [9] J. Roh, B. Kim, D. Kim, J. Kim, and D. H. Sim, "Reliable Autonomous Reconnaissance System for a Tracked Robot in Multi-floor Indoor Environments with Stairs," *The Journal of Korea Robotics Society*, vol. 19, no. 2, pp. 149-158, 2024, DOI: 10.7746/jkros.2024.19.2.149.
- [10] E. Hajizada, B. Swaminathan, and Y. Sandamirskaya, "Continual Learning for Autonomous Robots: A Prototype-based Approach," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Abu Dhabi, UAE, pp. 13988-13995, 2024, DOI: 10.1109/IROS58592.2024.10802683.
- [11] H. Le, S. Saeedvand, and C. C. Hsu, "A Comprehensive Review of Mobile Robot Navigation Using Deep Reinforcement Learning Algorithms in Crowded Environments," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 4, 2024, DOI: 10.1007/s10846-024-02198-w.
- [12] V. M. Tran and G. Kim, "Mapless Navigation with Distributional Reinforcement Learning," *The Journal of Korea Robotics Society*, vol. 19, no. 1, pp. 92-97, 2024, DOI: 10.7746/jkros.2024.19.1.092.
- [13] A. Wang, L. Liu, H. Chen, Z. Lin, J. Han, and G. Ding, "YOLOE: Real-Time Seeing Anything," *arXiv:2503.07465*, 2025, DOI: 10.48550/arXiv.2503.07465.



김 지 호

2021 경북대학교 심리학과(문학사)
심리정보융합(공학사)
2021~2023 POSTECH 인공지능연구원 연구원
2023~2025 (주)AIMMO 연구원
2025~2026 (주)대동로보틱스 대리

관심분야: 인공지능, 컴퓨터비전, 로봇지능



감 병 우

1996 울산대학교 전기공학과(공학사)
2019 울산대학교 전기전자정보시스템공학과
(공학석사)
1996~2001 대우자동차
2001~2024 (주)대동공업 전무
2024~현재 (주)대동로보틱스 전무

관심분야: 시스템 제어, 임베디드 시스템



곽 정 훈

2015 계명대학교 게임모바일콘텐츠학과(공학사)
2017 동대학 컴퓨터공학과(공학석사)
2021 동국대학교 멀티미디어공학과(공학박사)
2022~2025 한국로봇융합연구원 선임연구원
2024~2025 (주)대동로보틱스 차장
2025~현재 차세대융합기술연구원 선임연구원

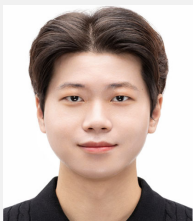
관심분야: 인공지능, 딥러닝, 로봇제어



이 준 우

2007 부산대학교 전자전기공학과(공학사)
2009 KAIST 로봇공학(공학석사)
2014 동 대학 전기공학(공학박사)
2015~현재 경북대학교 전기공학과 부교수

관심분야: 군집 로봇틱스, 진화연산, 메타휴리스틱스



경 도 현

2021 국립강릉원주대학교 컴퓨터공학과(공학사)
2023 동국대학교 자율사물지능학과(공학석사)
2024~2025 한국로봇융합연구원 주임연구원

관심분야: 인공지능, 딥러닝, 사물지능제어



서 갑 호

1999 고려대학교 전기공학과(공학사)
2001 KAIST 전기및전자공학(공학석사)
2009 동대학(공학박사)
2009~현재 한국로봇융합연구원 수석연구원
2021~현재 경북대학교 로봇 및 스마트 시스템
공학과 겸임교수
2024~2026 (주)대동로보틱스 상무

관심분야: 시스템 제어, 농업로봇, 모바일로봇



구 재 완

2017 계명대학교 의용공학과(공학사)
2020 경북대학교 기계공학과(공학석사)
2019~현재 한국로봇융합연구원 선임연구원
2024~2026 (주)대동로보틱스 과장

관심분야: 기계설계, 구조해석