

# 차량 탑재형 상·하역 장비의 설계와 딥러닝 객체 인식을 이용한 자동제어 방법

## Design of Vehicle-mounted Loading and Unloading Equipment and Autonomous Control Method using Deep Learning Object Detection

이순교<sup>1</sup>·김선목<sup>2</sup>·우효원<sup>2</sup>·이석<sup>1</sup>·이기백<sup>†</sup>

Soon-Kyo Lee<sup>1</sup>, Sunmok Kim<sup>2</sup>, Hyowon Woo<sup>2</sup>, Suk Lee<sup>1</sup>, Ki-Baek Lee<sup>†</sup>

**Abstract:** Large warehouses are building automation systems to increase efficiency. However, small warehouses, military bases, and local stores are unable to introduce automated logistics systems due to lack of space and budget, and are handling tasks manually, failing to improve efficiency. To solve this problem, this study designed small loading and unloading equipment that can be mounted on transportation vehicles. The equipment can be controlled remotely and is automatically controlled from the point where pallets loaded with cargo are visible using real-time video from an attached camera. Cargo recognition and control command generation for automatic control are achieved through a newly designed deep learning model. This model is designed to be optimized for loading and unloading equipment and mission environments based on the YOLOv3 structure. The trained model recognized 10 types of pallets with different shapes and colors with an average accuracy of 100% and estimated the state with an accuracy of 99.47%. In addition, control commands were created to insert forks into pallets without failure in 14 scenarios assuming actual loading and unloading situations.

**Keywords:** Object Detection, YOLOv3, Logistics, Deep Learning, Autonomous System

### 1. 서 론

현재 물류 산업은 폭발적인 증가세를 보이고 있으며, 물류 작업의 효율성과 비용을 줄이기 위해 거대 물류창고들은 대부분 자동화된 시스템을 채택하고 있다. 자동화된 물류체계는 노동을 대폭 감소시키면서 물류 작업의 효율성을 크게 향상시킨다. 하지만 대규모 물류센터와 달리 기존의 분산화된 소규모 물류창고는 자동화된 물류체계를 채택하지 못하고 있다.

거대 물류창고의 시스템을 그대로 적용시키기에는 공간과 예산이 부족하기 때문이다. 이러한 물류창고들은 여전히 수작업으로 물류작업을 진행하고 있으며, 이에 따라 작업시간과 비용이 효율적이지 못하다. 대표적으로, 전국 각지에 있는 중소규모 군부대는 상·하역에 필요한 물류체계와 장비를 갖추지 못해 모든 화물을 장병들이 직접 손으로 상차 및 하차하고 있으며, 지역의 마트 등 영세 업체들 또한 비슷한 상황이다.

본 논문에서는 먼저 이러한 소규모 물류 환경을 효율화하기 위한 차량 탑재형 상·하역 장비의 고안을 제안한다. 이 장비의 개념은 [Fig. 1]에 요약 표현되어 있다. 평소에는 [Fig. 1(a)]와 같이 상·하역 장비를 활용하고, 이동시 [Fig. 1(b)]와 같이 접은 후, 비교적 크기가 큰 배송차량에 [Fig. 1(c)]와 같이 탑재하여 배송 시 함께 이동한다면, 장비를 보유하지 못한 다수의 소형 물류창고들의 효율이 동시에 개선될 수 있다.

제안하는 접이식 장비는, 기존 디젤기관 지게차와 달리, 운송자가 직접 탑승하지 않는다. 이는 원격 제어가 용이하다는

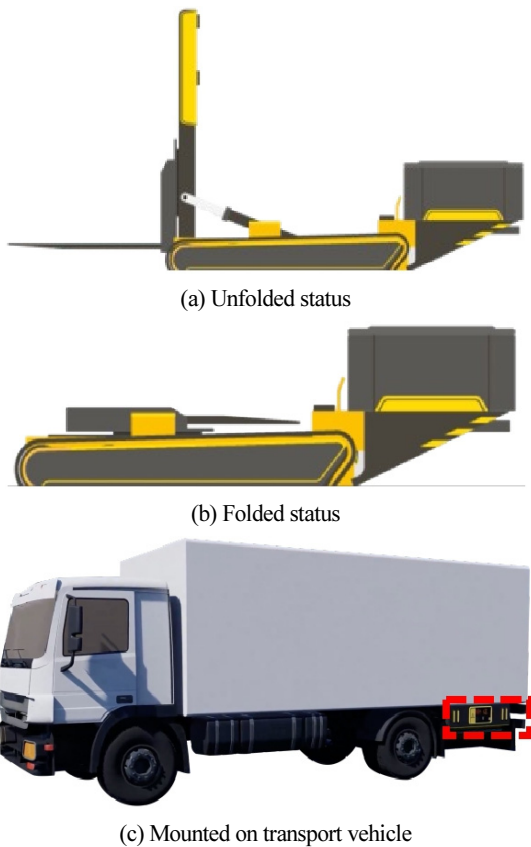
Received : Nov. 6. 2023; Revised : Nov. 30. 2023; Accepted : Dec. 18. 2023

※ This work is supported by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant RS-2021-KA163201). Additionally, this paper was researched through the 2021 Kwangwoon University Outstanding Researcher Support Project

1. Researcher, Future Transport Logistics Research Center, Korea Railroad Research Institute, Uiwang, Korea (sklee, slee@krii.re.kr)

2. Researcher, Dept. Electrical Engineering, Kwangwoon University, Seoul, Korea (nadasunmok@gmail.com, dngydnjs@naver.com)

† Associate Professor, Corresponding author: Dept. Electrical Engineering, Kwangwoon University, Seoul, Korea (kblee@kw.ac.kr)



[Fig. 1] Concept illustration of vehicle-mounted loading and unloading equipment

장점과, 정밀하게 조작하기 위해 기존과 다른 숙련도를 요구한다는 단점을 동시에 가지고 있다. 본 논문에서는 이러한 단점을 극복하기 위해 작업자의 숙련도를 보조해주는 자동제어 방법을 함께 제안한다. 상·하역 장비와 유사한 다른 장비들의 경우 기존에 다양한 자동화 관련 연구가 진행되었다<sup>11-16</sup>. 자동화에서는 목표 대상에 대한 인식이 필수적인데(상·하역 장비의 경우 팔레트이다), 대상에 특별한 마커를 부착해 인식하는 방법<sup>11-13</sup> 대상의 색상 정보를 이용해 영역을 추출하는 방법<sup>14</sup> 등이 연구되었다. 하지만 이 방법들은 자동화를 위해 그 대상을 특수 제작해야 하거나, 대상 주변의 환경이 복잡한 경우 그 객체를 잘 인식하지 못한다는 문제가 있다. 이를 해결하기 위해 최근 딥러닝 기반의 물체 인식연구가 활발히 진행되고 있다<sup>15-16</sup>. 딥러닝 기반의 물체 인식은 기존 인식 방법과 비교해 좀더 다양한 물체를 정확하게 인식 할 수 있는 장점이 있지만, 이러한 기존 연구 결과가 소형 창고 환경에 바로 적용될 수 없고, 특히 상·하역 장비의 경우 수동 원격 제어와 자동 제어 간의 유기적인 연결을 고려 해야 할 뿐 아니라 물체를 정확히 인식한 후에도 포크를 팔레트의 구멍에 삽입하기 까지 다양한 경우에 대한 판단 및 제어 명령 생성을 포함한 알고리즘 개발이 추가로 필요하다.

따라서, 본 논문에서는 우선 차량 탑재형 접이식 상·하역 장비와 이를 이용한 임무에 최적화된 딥러닝 모델을 제안한다. 장비에 부착된 카메라와 딥러닝 모델을 이용해 전방에 있는 팔레트를 인식하고, 인식한 팔레트의 상태(위치 및 자세)를 정확하게 추정한다. 또한 본 논문에서는, 딥러닝 모델로 얻은 상태 정보를 이용해 장비를 자동으로 제어하는 알고리즘을 함께 제안한다. 딥러닝 모델의 학습은 다양한 현장 데이터를 직접 촬영하여 획득하고 라벨링(Labeling)한 데이터를 활용한다. 다양한 팔레트의 크기, 색, 모양 등을 고려해서 10개의 종류의 팔레트를 데이터에 포함하고, 학습하지 않은 팔레트에 대해서도 인식 및 상태 추정을 할 수 있게 최적화한다. 딥러닝 모델의 기반 구조로는 물체 인식 모델로 잘 알려진 YOLOv3를 차용한다<sup>17</sup>. 제작된 하나의 장비를 이용해 여러 연구가 동시에 진행되는 관계로, 원활한 데이터 수집과 다양한 시나리오 테스트를 위해 딥러닝 모델 및 알고리즘의 검증은 실제 설계한 장비 대신 핸드 팔레트 트럭으로 모사한 테스트 장비를 별도 구성해 수행한다.

## 2. 관련연구

### 2.1 상·하역 자동화 장비

상·하역 작업의 자동화는 물류 및 창고 관리 분야에서 기술적 혁신과 효율성 증진의 핵심 요소로 부상한다. 초기 산업 시대의 물류 시설과 기술이 어떻게 현대의 상·하역 자동화 기술 발전에 영향을 미쳤는지를 탐구한 연구가 있었다<sup>18</sup>. 이는 역사적 관점에서 시카고 교차철도의 화물 취급 기술과 산업 건물 디자인의 변화를 조명하고 오늘날 창고 자동화 시스템의 설계와 구현에 중요한 역사적 맥락을 제공한다. 또한 창고 관리 시스템의 구현을 통해 주문 선택 과정을 개선하는 방법 또한 활발히 연구되어 왔다<sup>19</sup>. 이는 상·하역 자동화가 창고 운영의 효율성과 정확성을 어떻게 증진시킬 수 있는지를 간접적으로 증명하며, 결과적으로 이들의 연구는 상·하역 자동화가 창고 운영의 전반적인 효율성과 생산성을 향상시키는 데 중요한 역할을 할 수 있음을 시사한다. 한편 근집 로봇의 관점에서도 상·하역 자동화의 구체적 기술 구현 측면에 집중된 연구도 진행되고 있다<sup>20</sup>. 이 연구는 로봇 기술이 상·하역 작업에 어떻게 적용될 수 있는지에 대한 통찰력을 제공하며, 로봇 기술의 발전이 물류 분야에서 어떻게 혁신을 가져올 수 있는지를 보여준다. 특히, 근집 로봇의 에너지 최적화와 효율적인 작업 수행 방법에 대한 연구가 진행되고 있는데 이는 상·하역 자동화에서 로봇 기술이 궁극적으로 필수 불가결함을 강조한다<sup>21</sup>. 이러한 로봇 기술은 상·하역 과정을 더 빠르고 정확하게 만들며, 인간 노동자의 부담을 줄이고 전반적인 작업 효율을 향상시킬

수 있다. 마지막으로, 혁신적인 기술과 전통적인 기술을 비교 분석함으로써, 상·하역 자동화 기술이 물류 과정에 어떤 혁신적인 변화를 가져올 수 있는지에 관한 연구도 있었다<sup>22)</sup>. 이 연구는 기술 혁신이 물류 및 창고 관리 분야에서 어떻게 효율성과 생산성을 개선할 수 있는지에 대한 중요한 통찰력을 제공한다.

## 2.2 팔레트 인식

### 2.2.1 객체 감지(Object Detection)

객체 감지는 이미지나 비디오에서 객체를 찾는 컴퓨터 비전 분야의 대표적인 임무들 중 하나이다. 보통 지도 학습을 통해 학습 데이터 이미지에 있는 개체의 존재와 위치를 이용해 모델이 훈련된다. 객체 감지의 목표는 객체를 둘러싸는 경계 상자 와 객체 유형을 식별하는 클래스 레이블을 출력하는 것이다.

객체 감지를 위해 제안된 다양한 딥러닝 구조에는 RCNN (Region-based Convolutional Neural Networks)<sup>23)</sup>, SSD (Single Shot multi-box Detector)<sup>24)</sup>, YOLO (You Only Look Once)<sup>25)</sup> 및 그 변형(예: YOLOv2<sup>26)</sup>, YOLOv3<sup>17)</sup>) 등이 있다. 이러한 모델들은 입력 이미지를 받아 관련 클래스 확률과 함께 예측된 경계 상자 정보를 출력한다.

객체 감지 모델의 주요 요소 중 하나는 학습에 사용되는 손실 함수다. 객체 감지에 가장 일반적으로 사용되는 손실 함수는 multi-task loss이며, 이는 localization error (예: 예측된 경계 상자와 ground-truth 상자의 편차) 및 classification error (예: 예측된 경계 상자의 클래스 확률과 ground-truth 클래스 레이블과의 차이)를 고려한다.

최근 몇 년 동안 딥러닝의 발전으로 다양한 벤치마크에서 객체 감지 모델의 성능이 크게 향상되어 자율 주행, 감시, 증강 현실 등과 같은 실제 응용 프로그램에 실용적으로 사용할 수 있게 되었다.

### 2.2.2 YOLOv3

YOLO 계열 모델들은 여러 기법들을 추가해가며 발전해오고 있다. YOLOv2는 기존의 YOLO에 딥러닝 분야에서 대표적으로 성능향상에 도움이 되는 기법들을 다량 적용해보고 그 중 성능이 향상되었던 기법들만 취하는 형태로 개선되었다. YOLOv3는 YOLOv2에서 발견된 단점을 보완하는 형태로 개선되었는데, 대표적인 단점은 작은 객체에 대한 인식률이 좋지 않다는 것이다. 이를 보완하기 위해 YOLOv3에서는 세 개의 각기 다른 크기의 특징 맵(feature map)으로 객체를 탐지한다. 기존에는 많은 컨볼루션 레이어(convolution layer)를 통과한 특징 맵이 일반적인 특징을 잘 나타낸다고 알려져 있어 1개의 특징 맵만을 사용하였다. 하지만, 컨볼루션 레이어를 통과할 수록 특징 맵의 크기가 작아지고 특징 맵 내에서 객체가 담당하

는 영역이 줄어들는데, 이는 작은 객체를 인식하는데 부정적인 영향을 끼친다. YOLOv3에서는 작은 객체에 대한 인식을 위해 최종 레이어를 통과하기 이전에 서로 다른 크기의 특징 맵 2개를 추가로 학습과 예측에 포함 시킨다. 이러한 방식은 테두리, 색상과 같은 단순한 특징들도 예측에 포함시킬 뿐 아니라, 크기가 큰 특징 맵을 통해 작은 객체에 대한 인식률도 향상시켰다.

## 2.3 자세 추정 및 유도 알고리즘

### 2.3.1 팔레트 자세 추정

팔레트 적재 및 상하차 과정에서 팔레트의 정확한 위치와 방향을 파악하는 것은 매우 중요하다. 팔레트 적재 문제에 대한 휴리스틱 접근법이 제안되어 왔으며, 이 방법은 팔레트의 위치와 방향을 고려하여 최적의 적재 방식을 결정하는 데 중요한 역할을 한다<sup>27)</sup>. 또한 L-형 접근 방식을 통해 팔레트의 자세와 적재 순서를 최적화하는 방법도 제시되었고<sup>28)</sup> 최근 연구에는 대규모 복잡한 장소에서 추가적인 하드웨어 설비 없이 무선 액세스 포인트를 사용한 실내 3D 위치 추정에 초점을 맞추는 연구도 진행되고 있다<sup>29)</sup>. 이 연구는 시간 지문을 기반으로 한 3D 동적 위치 추정 모델을 제안하며, 위치 추정의 정확성과 견고성을 향상시키는 데 중요한 역할을 한다. 그러나 이 방법은 무선 신호의 간섭과 환경 변화에 민감할 수 있으며, 실시간 위치 추적에 있어서 일부 지연이 발생할 수 있다는 한계가 있다.

### 2.3.2 팔레트 유도 알고리즘

팔레트 유도 알고리즘은 팔레트의 적재 순서와 방식을 최적화하여 전체 적재 공간의 활용도를 높이는 것을 목표로 하는 연구가 진행되었고<sup>30)</sup>, 팔레트 적재 문제를 해결하기 위한 최적의 분기 및 경계 알고리즘이 소개됐다<sup>31)</sup>. 최근에는 비시야 환경에서 초광대역(UWB) 시간-비행(TOF) 기반 거리 측정 정보의 오류를 분석하고, 이를 개선하기 위한 알고리즘을 제안되었다<sup>32)</sup>. 이 연구는 실내 공간의 사전 정보를 활용하여 앵커 LOS/NLOS 정보 지도를 구축하고, 이를 기반으로 견고한 적응형 확장 칼만 필터링 알고리즘을 설계한다. 그러나 이 방법은 환경 변화에 민감하고 특정 하드웨어에 의존하는 한계가 있다.

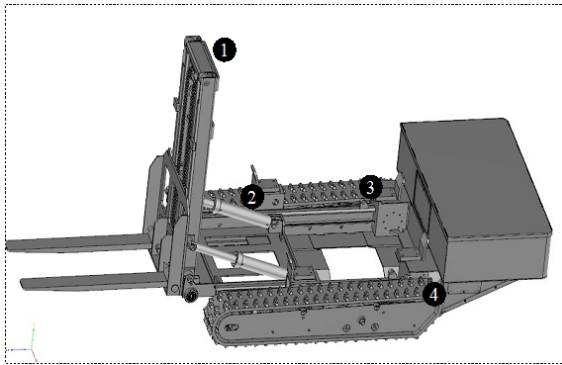
## 3. 방법

### 3.1 상하차 자동화 장비 설계

#### 3.1.1 구조

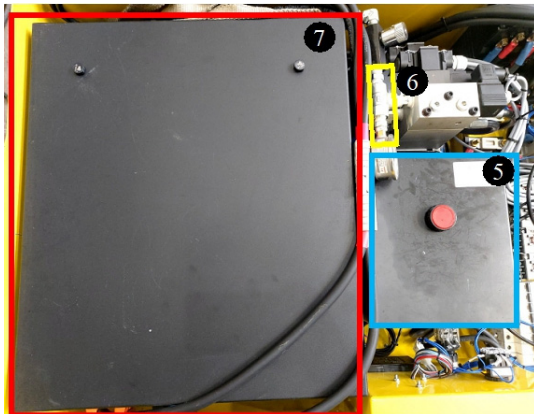
본 논문에서 고안한 장비는 5톤 트럭 하단에 적재를 하기 위하여 접이식 지게차 구조를 가진다. 5톤 트럭 하단의 여유공간을 고려해, 완전히 접었을 때 길이 2,176 mm, 너비 1,210 mm,

높이 645 mm 규격을 갖도록 차체가 설계되었다. 또한 저상형 구조로 트럭 하단 탑재부에 들어가야 하기 때문에 포크가 장착되어 있는 마스트부분이 90도로 접히도록 설계 및 제작하였다. 90도로 접히는 것이 가능하게 하기 위하여 접는 동력을 전



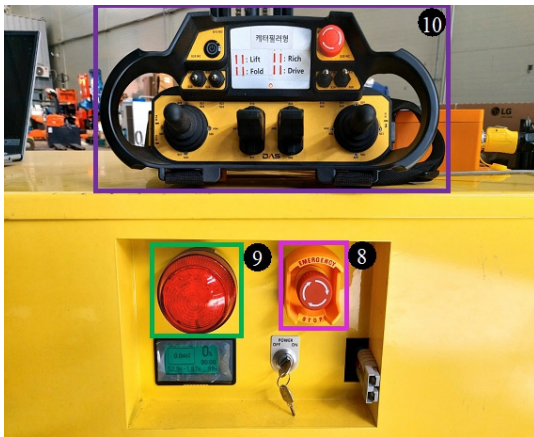
(a) Structure

(1) Mast (2) Hydraulic cylinder for folding  
(3) Hydraulic cylinder for reach (4) Caterpillar



(b) Power module

(5) Power pack (6) Electronic control valve (7) Battery



(c) Peripheral module

(8) Emergency switch (9) Speaker and lamp (10) Remote controller

[Fig. 2] Foldable forklift structure, power unit and peripheral devices

달하는 유압 실린더를 양쪽에 따로 장착하여 안정적으로 접고 펴게 구조를 구성하였다. 또한 자체 중량을 줄이기 위하여 마스트를 절곡하여 무게를 줄이는 동시에 강성을 높게 만들어 제작 하였다. 1톤이 조금 넘는 무게로 1톤을 들기 위해서는 무게 접점이 양쪽 바퀴 끝라인보다 뒤쪽에 위치하여야 하기 때문에 포크 마스트를 앞뒤로 움직일 수 있는 리치 기능을 추가 하였다. 리치 기능을 구현하기 위하여 접는 기능과 마찬가지로 양 옆에 리치 기능을 위한 유압 실린더를 장착하고, 슬라이딩 구조를 채택하여 부드럽게 앞 뒤로 움직이는 구조를 구현 하였다. 마스트 뿐 아니라 포크도 접이식으로 제작하여, 최대한 부피를 줄이도록 설계 및 제작하였다. 구동을 위한 구조는 어느 지형 및 등판 각도에서 주행이 가능하게 하기 위하여 캐터필러 웨도 구조로 설계를 하였다. 모터에 스프로킷을 장착하여 웨도를 움직일 수 있게 하였다. 웨도를 장착하여 탱크와 같은 움직임을 가지게 하고, 그로 인해 울퉁불퉁한 지형 및 높은 등판 각도에서도 구동을 할 수 있게 하였다. [Fig. 2(a)]는 구조의 각 부위를 포함한 그림이다.

### 3.1.2 동력장치

접이식 지게차에는 크게 두가지의 동력원이 들어간다. 지게차의 움직임을 위한 모터와 리프트를 제어하는 유압 파워팩이 장착되어 있다. 지게차의 움직임을 위한 모터는 양 옆에 2.5 Kw의 AC 모터를 장착하고, 60:1의 감속기를 사용하여 속도 및 토크를 조절하였다. AC 모터드라이버 제어기를 이용하여 CAN으로 직접 설정 및 제어하였다. 파워팩은 2.2 Kw DC 파워팩을 사용하여 리프트에 유압을 공급하였다. 각 리프트를 제어하는 실린더에는 전자제어 밸브를 장착하여 제어보드에서 각각의 밸브 제어를 통해 독립적인 움직임을 가질 수 있게 구현하였다. 두가지의 동력장치를 구동하기 위하여 48 V, 200 Ah 배터리를 장착하여 구동하였다. 배터리는 400 A까지 순간 출력이 가능하여 모터 및 파워팩을 동시에 제어 가능하게 제작 하였다. 배터리는 충전시 3시간 안에 완충이 가능하며, 연속 2 시간까지 구동이 가능하다. [Fig. 2(b)]는 동력장치 부위들의 그림이다.

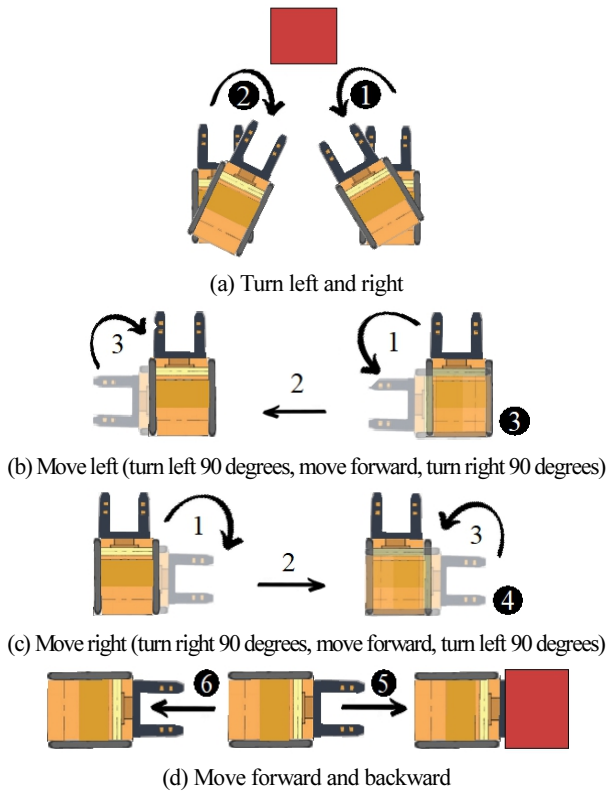
### 3.1.3 주변장치

접이식 지게차에는 항상 비상상황에 대비하여 비상 스위치를 외부에 장착 하였으며, 스피커와 램프를 통하여 동작 및 비상상황에 대한 알림을 하게 된다. 또한 산업용 주파수를 이용하는 조종기를 사용하여 800 m 반경 내에서 원격 조종이 가능한 제어기를 장착하여 언제 어디서든 쉽게 조종 가능하게 구현하였다. 와이파이 클라이언트 장착을 통하여 무선으로 접이식 지게차의 상태 또한 받아 볼 수 있다. [Fig. 2(c)]는 주변장치의 각 부위를 포함한 그림이다. 마지막으로, 물체를 인식하기

위한 Intel사의 RealSense D415 모델카메라와 딥러닝 모델을 연산하기 위한 노트북이 탑재된다.

### 3.1.4 궤도 방식 장비의 구동 모델

장비는 궤도 방식의 구동으로 설계되었으며, 궤도 방식의 경우 동역학 기반의 모델링을 한다고 해도 슬립 등에 의해 오차가 발생하고 결국 카메라 영상을 통한 피드백 제어에 의존해야 한다. 하지만 궤도 방식의 장점으로 제자리 회전이 가능하다는 점과 직진성이 어느정도 보장된다는 점이 있어, 이를



[Fig. 3] Command types of foldable forklift

[Table 1] Control commands of continuous track vehicle

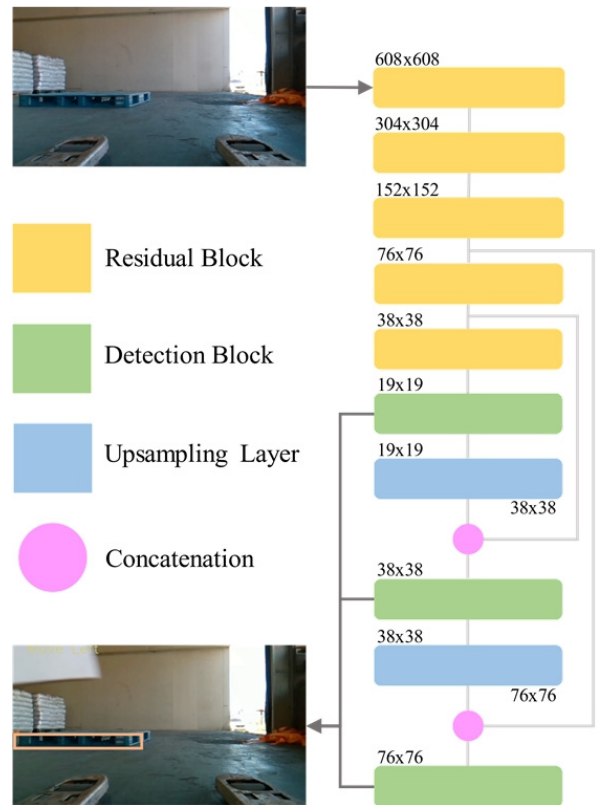
| No. | Command    | Function call | Angular velocity [rad/s] | Velocity [m/s] | Parameters |
|-----|------------|---------------|--------------------------|----------------|------------|
| 1   | Turn left  | Turn          | +0.71                    | 0              | Angle      |
| 2   | Turn right | Turn          | -0.71                    | 0              | Angle      |
| 3   | Move left  | Turn          | +0.71                    | 0              | 90 deg.    |
|     |            | Move          | 0                        | +0.32          | Distance   |
| 4   | Move right | Turn          | -0.71                    | 0              | 90 deg.    |
|     |            | Move          | 0                        | +0.32          | Distance   |
| 5   | Forward    | Move          | 0                        | +0.32          | Distance   |
| 6   | Backward   | Move          | 0                        | -0.32          | Distance   |

모두 감안하면 동역학 모델링을 생략할 수 있고 대신 장치의 구동 명령을 단순화 하는 것이 가능하다. [Fig. 3]은 궤도 방식 장비의 명령 종류를 보여준다. 좌 회전, 우 회전, 좌 이동, 우 이동, 전진, 후진 총 6가지의 명령으로 제어가 가능하고 이는 이후 설명할 제어 알고리즘에도 반영되었다.

[Table 1]에는 6가지의 명령에 대한 호출함수, 회전속도, 직진속도, 매개 변수가 좀 더 자세히 기술되어 있다. 각 명령어에 대한 함수를 호출하면 속도와 매개 변수가 입력으로 들어가고 장비를 제어한다. 회전속도의 단위는 rad/s이며 양수인 경우 시계반대방향, 음수인 경우 시계방향으로 제어한다. 직진속도의 단위는 m/s이고 양수인 경우 직진, 음수인 경우 후진으로 제어한다. 회전속도, 직진속도 모두에서 0은 정지하고 있는 상태를 의미한다. 좌회전과 우회전은 회전운동만을 하기 때문에 직진속도 값은 0이 되고, 직진과 후진은 직선운동만 하기 때문에 회전속도 값이 0이 된다.

### 3.2 팔레트 인식 알고리즘

우리는 객체를 식별하고 위치를 예측하기 위해 객체 탐지 모델로 잘 알려진 YOLOv3<sup>[17]</sup>를 사용했다. 객체 감지 모델은 [Fig. 4]에 기술된 바와 같이 다음과 같은 요소들로 구성된다.



[Fig. 4] Object detection model (YOLOv3) architecture

먼저 다크넷(Darknet-53)<sup>[17]</sup>이라고 불리는 컨볼루션 신경망 기반구조(Backbone)를 사용한다. 다크넷은 잔차블록(Residual blocks)을 기반으로 하며, 이는 잔차 연결(Residual connections)을 사용하여 신경망의 깊이와 효율성을 향상시킨다. 탐지블록(Detection block)은 객체 탐지 모델에서 다크넷 기반구조 네트워크의 뒷부분에 추가되며, 다양한 크기와 종류의 객체를 효과적으로 탐지하기 위한 다양한 스케일의 출력을 생성한다. 이를 위해 앵커박스(Anchor box)와 다중 스케일 특징 맵을 사용하여 객체의 위치와 클래스를 예측하고, 다양한 스케일에서 생성된 출력을 결합하여 모델이 다양한 객체를 다양한 크기와 스케일에서 효과적으로 탐지할 수 있도록 한다. 이는 객체 탐지 모델의 정확도와 다양성을 향상시키는 데 기여한다. 탐지블록 사이에 업샘플링 계층(Upsampling Layer)과 연결(Concatenation)을 사용하는데 이것은 더 높은 해상도의 특징 맵을 더 낮은 해상도의 특징 맵과 결합해서 객체의 정확한 위치를 예측하는데 도움을 준다. 객체 탐지를 위한 다양한 스케일의 특징 맵을 효과적으로 조합하고 더 높은 해상도에서 객체의 위치를 정확하게 예측하기 위해 사용한다. 탐지 블록을 통과한 특징 맵은 각

각 24채널을 가진다. 특징 맵의 한 셀당 3개의 위치 및 클래스 예측값을 가지며, 예측값은 [Table 2]의 구조를 따른다. 크기가 다른 3개의 특징 맵은 총 22,743개의 위치 및 클래스 정보를 포함하고 있으며, 비-최대 억제(Non-max Suppression) 알고리즘을 통해 가장 정확한 예측값을 선택한다. 이를 통해 모델은 다양한 크기의 객체를 감지하고 정확한 경계상자를 생성할 수 있다.

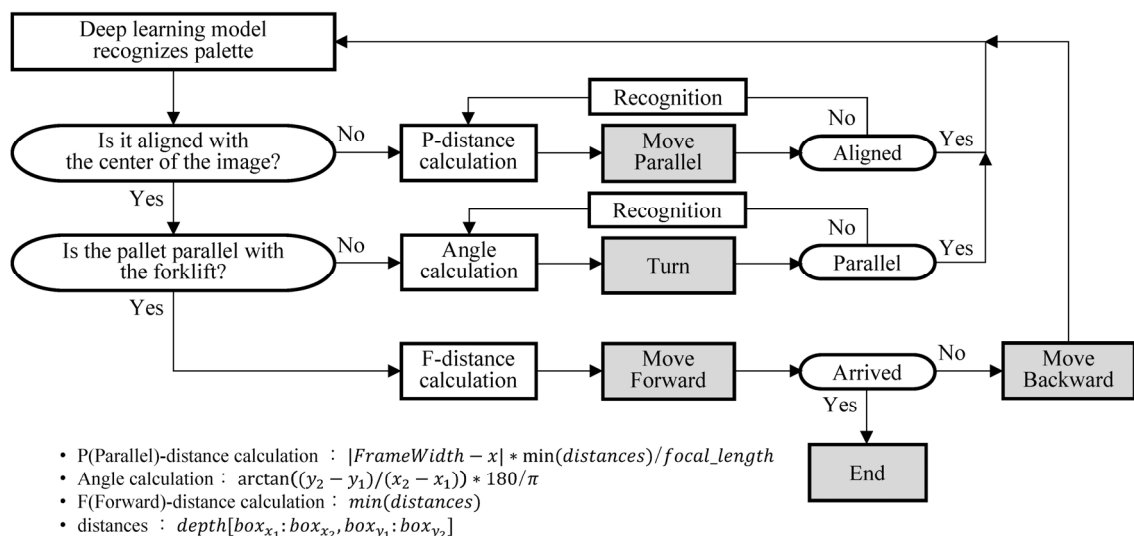
카메라 영상 기반 주행은 1) 딥러닝 모델을 통한 팔레트 인식, 2) 팔레트 상태 추정, 3) 지게차 유도 알고리즘으로 세분화할 수 있다. 먼저 딥러닝 모델을 통해 카메라 영상으로부터 팔레트를 인식하고, 팔레트 영역의 이미지 상세 정보와 클래스 정보를 획득한다. 이후 이 정보들을 취합해 팔레트의 상태를 추정한다. 이렇게 추정된 상태를 바탕으로 설계된 알고리즘을 따라 지게차가 유도된다. 전체 알고리즘의 흐름은 [Fig. 5]의 플로우 차트에서 확인할 수 있다. 플로우 차트에서 distances는 바운딩 박스에 대한 depth값을 의미하는데 여기서  $[box_{x_1} : box_{x_2}, box_{y_1} : box_{y_2}]$ 는 바운딩 박스 안의 영역을 의미한다. 즉 distances는 바운딩 박스 안의 각 픽셀에 대해 카메라와의 거리를 의미한다. 그에 따라 직선거리는 바운딩 박스 영역안에 각 픽셀의 거리 중 가장 가까운 거리를 의미한다.

팔레트 인식 및 위치 계산을 위해서 지도학습 계열의 딥러닝 물체 감지 기법인 YOLOv3를 활용한다. 학습의 경우 출력 벡터(Output vector)와 정답 영역이 얼마나 일치하는지를 손실로 계산하고, 이를 역전파(Back propagation) 하여 학습을 한다.

팔레트 인식단계는 학습된 딥러닝 모델을 통해 물체 이미지가 입력으로 들어왔을 때, 타겟 물체의 위치와 회전 각도를 예측하는 단계이다. 딥러닝 모델을 통해 타겟 물체의 위치정보(중심 픽셀 값, 박스의 가로,세로길이)와 클래스(CCW, CW,

[Table 2] Output vector configuration

|   |                       |
|---|-----------------------|
| A | Box_center_X          |
| B | Box_center_Y          |
| C | Box_width             |
| D | Box_height            |
| E | Object-ness Score     |
| F | Class Score (CCW)     |
| G | Class Score (CW)      |
| H | Class Score (Aligned) |



[Fig. 5] Flowchart of autonomous control method using object detection

Aligned)정보를 얻는다. [Table 2]에서 출력 벡터의 각 요소들을 보면 A와 B는 예측한 박스의 중점좌표를 의미하고 C와 D는 예측한 박스의 너비와 높이를 의미한다. 그리고 E는 박스안에 객체가 있을 가능성을 나타내는 점수이고 F, G, H는 각각 박스안에 객체가 어떤 클래스(class)일지를 나타내는 점수이다. 이 클래스들을 시각화 하면 [Fig. 6]의 예시 사진과 같다. Aligned는 팔레트와 지게차의 각도가 수평이라는 것을 의미한다. 이때 수평의 기준을 팔레트의 각도 -5도에서 +5도 사이로 한다. CCW는 팔레트와 지게차를 수평으로 만들기 위해서는 지게차가 시계 반대방향으로 회전해야 된다는 것을 의미한다. 반대로 CW는 팔레트와 지게차를 수평으로 만들기 위해서는 지게차가 시계방향으로 회전해야 된다는 것을 의미한다.



(a) Class Aligned example (Pallet and forklift are horizontal)



(b) Class CCW example (Counterclockwise rotation is required)



(c) Class CW example (Clockwise rotation is required)

[Fig. 6] Visualization of deep learning model output vector

### 3.3 팔레트 상태 추정 및 유도 알고리즘

#### 3.3.1 팔레트 상태 추정

인식한 팔레트의 영상 내 영역 정보와 클래스 정보를 통해 지게차와의 상대 위치를 포함한 팔레트의 상태를 다음과 같이 추정한다.

- 좌/우 상대 위치 연산: 카메라 중심과 실제 물체 중심의 실제 거리는 다음식으로 연산한다.

f: 카메라의 초점 거리(핀 홀과 이미지 센서 사이거리)

d: 물체와 카메라 사이의 거리

x: 이미지 상의 거리(이미지 중심과 물체중심의 거리)

X: 실제 거리(카메라 중심과 물체의 거리)

$$\frac{x}{f} = \frac{X}{d} \quad (1)$$

$$X = \frac{x \cdot d}{f} \quad (2)$$

- 회전 각도 연산: 지게차가 팔레트와 수평을 이루기 위한 회전 각도를 연산하기 위해 RGB 영상정보와 Depth 영상 정보 중 팔레트 중심에 대한 거리 d를 사용한다. 출력 벡터 중 클래스(CCW, CW, Aligned)는 지게차가 팔레트 대비 어떤 방향으로 회전되어 있는지를 보여준다. 이후 예측한 박스의 위치 정보로부터 팔레트의 영역을 추출하고, 이 영역에 대해 OpenCV라이브러리를 활용해 지면과 맞닿은 직선을 감지한다. 이때 선 감지 임계값인 Thresholds는 50, 선으로 간주할 최소 길이인 MinLineLength는 10, 동일한 선으로 간주할 최대 간격은 MaxLindGap은 1값으로 설정한다. 감지된 직선의 pixel 좌표계상 양 꼭지점 좌표인  $(u_1, v_1), (u_2, v_2)$ 를 입력으로 arctan 함수를 사용해 기울기  $\alpha$ 를 계산한다. 이를 식 (3)과 (4)를 사용하여 실제 팔레트가 회전된 각도  $\varphi$ 로 변환한다. [Table 3]은 실제 장치의 가동 범위 내 다양한 d와  $\alpha$  경우의 수에 대한 실험을 통해 얻은  $\varphi$ 값이며, 이를 바탕으로 근사식 (4)를 도출하였다.

$$\alpha = \frac{\arctan \frac{v_2 - v_1}{u_2 - u_1} * 180}{\pi} \quad (3)$$

$$\varphi = \frac{5.63 - 0.77 \cdot \alpha - 0.02 \cdot d + 0.034 \cdot \alpha \cdot d}{1 + e^{-10(\alpha - 0.5)}} \quad (4)$$

[Table 3] Rotation angle,  $\varphi$  (degree) conversion table

|   |   | Distance to pallet center $d$ [cm] |      |      |      |      |      |      |      |
|---|---|------------------------------------|------|------|------|------|------|------|------|
|   |   | 100                                | 110  | 120  | 130  | 140  | 150  | 160  | 170  |
| Angle of line in the image $\alpha$<br>[degree] | 0 | 0                                  | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
|   | 1 | 6.3                                | 6.4  | 6.6  | 6.7  | 6.9  | 7    | 7.2  | 7.3  |
|   | 2 | 8.9                                | 9.4  | 9.9  | 10.4 | 10.9 | 11.4 | 11.8 | 12.3 |
|   | 3 | 11.6                               | 12.4 | 13.2 | 14   | 14.9 | 15.7 | 16.5 | 17.3 |
|   | 4 | 14.2                               | 15.4 | 16.5 | 17.7 | 18.8 | 20   | 21.2 | 22.3 |
|   | 5 | 16.8                               | 18.3 | 19.8 | 21.3 | 22.8 | 24.3 | 25.8 | 27.3 |
|   | 6 | 19.5                               | 21.3 | 23.1 | 25   | 26.8 | 28.7 | 30.5 | 32.4 |

- 직선 거리 연산 : 팔레트에 삽입하기 위해 지게차와 물체의 직선 거리를 계산 해야 한다. 우리는 거리를 계산하기 위해 RGB-D 영상의 깊이(Depth) 정보를 사용한다. 팔레트 영역 전체의 깊이 정보를 얻고, 가장 가까운 포인트를 실제 거리로 환산한다.

### 3.3.2 지게차 유도 알고리즘

마지막으로 지게차가 팔레트에 포크를 삽입할 수 있도록 자동 유도를 한다. 이 과정은 [Algorithm 1]에 자세히 기술되어 있다.

먼저 딥러닝 모델이 팔레트를 인식한다. 그리고 팔레트영역이 이미지의 중심에 있는지를 확인한다. 팔레트가 이미지 중심에 있지 않은 경우 상대 위치를 계산해 평행 이동을 한다. 만약 평행 이동 이후에도 중심이 맞지 않는다면 팔레트를 다시 인식하고 팔레트 영역이 이미지의 중심에 충분히 근접할 때까지 반복한다. 상대 위치 계산은 식 (1)과 (2)를 이용한다. 팔레트가 이미지의 중심에 있다면, 다음은 지게차와 팔레트의 회전 각도가 수평인지 여부를 확인한다. 즉, 인식된 팔레트의 클래스가 Aligned이 아니라면, 회전 각도를 계산해 인식된 클래스가 Aligned이 될 때까지 회전한다. 이때 회전 각도는 식 (3)과 (4)를 이용해 계산한다. 만약 팔레트와 이미지 중심이 충분히 근접하고, 팔레트의 각도도 수평에 가까운 경우는 카메라의 깊이 정보로 팔레트와 카메라의 거리를 계산한 후 직진을 한다. 카메라의 스펙상 50 cm이하의 깊이 정보는 얻을 수 없기 때문에, 직진하는 경우 두 번에 나눠서 직진을 한다. 먼저 50~100 cm사이까지 직진을 하고, 그 다음에 정확한 거리를 계산 후 전진 및 삽입 후 종료된다. 만약에 깊이정보를 얻을 수 없다거나 팔레트를 인식하지 못하는 경우 후진해서 팔레트를 다시 인식하고 알고리즘을 반복한다.

[Algorithm 1] Forklift guidance algorithm

#### Algorithm 1:

```

Data: RGB Camera Frame(6fps)
Data: Depth Camera Frame(6fps)
Result: Insert to palette
 $F \leftarrow$  Camera Focal Length
 $W \leftarrow$  Frame Width
 $f \leftarrow$  YOLOv3
 $h \leftarrow$  Crop Frame by box info
 $g \leftarrow$  Line Detection
manual driving;
while True do
   $w, h, u, v, class = f(RGBFrame)$ ;
  if not( $(9/20) * W < u < (11/20) * W$ ) then
    while not( $(9/20) * W < u < (11/20) * W$ ) do
       $w, h, u, v, class = f(RGBFrame)$ ;
       $distances = h(DepthFrame, w, h, u, v)$ 
       $distance = |0.5 * W - x| * \min(distances) / F$ 
       $Move\_parallel(distance)$ 
    end while
  else if  $class! = Aligned$  then
    while  $class! = Aligned$  do
       $w, h, u, v, class = f(RGBFrame)$ ;
       $u_1, v_1, u_2, v_2 = g(h(DepthFrame, w, h, u, v))$ 
       $angle = \arctan((v_2 - v_1) / (u_2 - u_1)) * 180 / \pi$ 
       $Turn(class, angle)$ 
    end while
  else
     $distances = h(DepthFrame, w, h, u, v)$ 
     $distance = \min(distances)$ 
    if  $distance \geq 100cm$  then
       $Move\_forward(distance - 50cm)$ 
    else if  $100cm > distance \geq 50cm$  then
       $Move\_forward(distance)$ 
      break;
    else
       $Move\_backward(50cm)$ 
    end if
  end if
end while

```

## 4. 평가방법 및 결과

### 4.1 데이터 셋

팔레트 검출을 위한 YOLOv3 network의 학습 속도와 성능 개선을 위해서 PASCAL VOC 2007 data set을 이용해서 사전 학습한 가중치를 초기값으로 사용한다. 그리고 크기, 모양, 색 등이 다른 10개 종류의 팔레트로 3,000개의 데이터를 수집했고 2,700개를 학습 데이터, 300개를 테스트 데이터로 사용했다. 데이터는 직접 라벨링을 수행하였고, 3개의 클래스(CCW, CW, Aligned)로 분류했다. 각 클래스 별 데이터의 개수는 [Table 4]에 정리되어 있다.



[Table 4] Dataset

| Class                | CCW | CW  | Aligned |
|----------------------|-----|-----|---------|
| # of train data (EA) | 900 | 900 | 900     |
| # of test data (EA)  | 100 | 100 | 100     |

[Table 5] Qualitative comparison with other methodologies

| Check list                   | Proposed method | [1] | [2] | [3] | [4] | [5] | [6] | [7] |
|------------------------------|-----------------|-----|-----|-----|-----|-----|-----|-----|
| Deep learning application    | O               | X   | X   | X   | X   | O   | O   | O   |
| Fork insertion guidance      | O               | X   | O   | O   | O   | X   | O   | O   |
| Various palettes             | O               | O   | O   | O   | X   | O   | O   | O   |
| Markers free                 | O               | X   | X   | X   | O   | O   | O   | O   |
| External setting unrequired  | O               | O   | X   | O   | O   | O   | O   | X   |
| Recognition of other objects | O               | X   | X   | X   | O   | O   | O   | O   |
| Mountable on vehicle         | O               | X   | X   | X   | X   | X   | X   | X   |
| Remote control supplement    | O               | X   | X   | X   | X   | X   | X   | O   |
| Dedicated forklift design    | O               | X   | X   | X   | X   | X   | X   | O   |

#### 4.2 다른 방법론과의 정성적 비교

본 논문이 제안하는 방법과 다른 방법론을 체크 리스트를 통해 정성적으로 비교한 결과는 [Table 5]에 정리되어 있다. 딥러닝을 사용하여 실시간 제어를 통해 포크 삽입까지 유도되고, 특정 팔레트가 아닌 다양한 종류의 팔레트를 인식 가능하게 설계해 다른 방법들에 비해 범용성과 실용성 측면에서 우수하다. 그리고 인식을 위한 특정 마커나 외부 환경에 대한 맞춤 세팅이 필요하지 않고, 팔레트 뿐만 아니라 다른 객체도 인식할 수 있도록 확장이 가능하다. 기존의 지게차들과 달리 수송차에 탈 부착이 가능하고 원격 조종 또한 가능하도록 전용 지게차를 설계한 점 또한 제안한 방법의 차별적인 요소이다.

#### 4.3 팔레트 인식 및 분류 결과

[Fig. 7]은 10개 종류의 팔레트에 대한 딥러닝 모델 인식 결과를 보여준다. 크기, 모양, 색등이 다른 10개 종류의 팔레트, 총 300개의 테스트 데이터에 대해서 인식 성공률 100%, mAP (Mean Average Precision) 99.47%를 달성하였다. 또한 모든 팔레트를 1개의 클래스로 학습시킨 경우 인식 성공률과 mAP 모두 100%의 결과를 얻었다.



(a) Class CCW

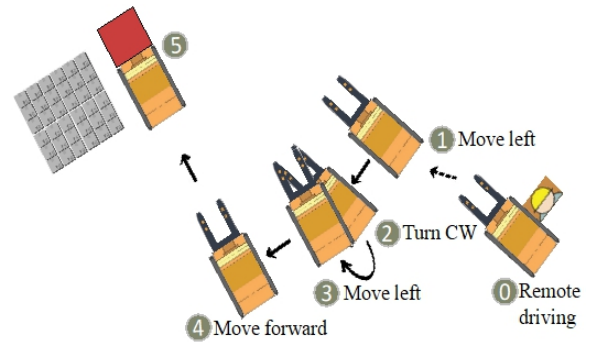


(b) Class Aligned



(c) Class CW

[Fig. 7] Recognition test results for 10 types of palettes



[Fig. 8] Test scenario

#### 4.4 시나리오 테스트 결과

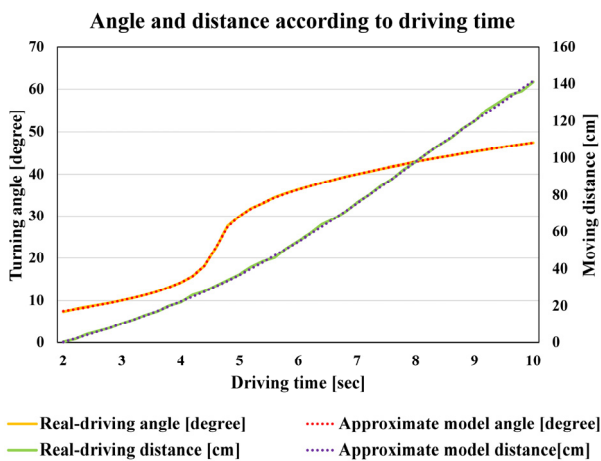
[Fig. 8]은 알고리즘을 통해 제어되는 상·하역 장치의 테스트 시나리오 그림이다. (0) 먼저 관리자가 원격으로 상·하역 지게차를 팔레트가 보이는 위치로 이동시킨다. 그 이후에 (1) 팔레트 영역이 이미지의 중앙에 위치했는지 확인한다. 그렇지 않다면 좌/우이동을 통해 팔레트 영역이 이미지의 중앙에 오도록 이동한다. 팔레트 영역이 이미지 중앙에 위치한다면, (2)

인식한 팔레트의 클래스가 Aligned인지 확인한다. 클래스가 Aligned이라는 것은 팔레트와 지게차가 수평으로 위치되어 있다는 것을 의미한다. Aligned이 아니라면 회전을 통해 Aligned이 될 때까지 회전 이동한다. 회전을 마치면, (3) 팔레트 영역의 이미지의 중앙에서 다시 벗어나게 되고 다시 좌/우 이동을 통해 팔레트가 이미지의 중앙에 오게 한다. (4) 마지막으로 클래스가 Aligned이고 팔레트가 이미지의 중앙에 있다면, 전진해서 포크를 팔레트에 삽입한다.

실제 접이식 상·하역 장치는 아직 소유권이 제작 업체에 있어 현재 운용이 가능한 시간과 범위가 제한적이다. 이러한 상황에서 최대한 실제와 근접한 테스트를 위해 손 지게차를 활용하되, 실제 접이식 상·하역 장치를 운용할 때 얻는 영상과 정확히 일치하도록 각도록 [Fig. 9] 와 같이 세팅을 하였다. 실제 장치를 통해 얻은 이미지와 동일한 결과를 얻도록 여러 번에 걸쳐 영상을 비교해가며 카메라의 위치, 영상에서 포크의 위치 등을 고려해서 정밀하게 세팅을 하였다. 실제 접이식 지게차와 동일한 방법으로 영상취득, 인식, 명령 생성 과정을 거치고 해당 명령의 수행만 사람이 직접 조작하여 테스트를 진행하였다.



[Fig. 9] Hand forklift equipment for testing



[Fig. 10] Actual forklift driving results

특히 명령 수행을 하는 과정에서 실제 상·하역 장치가 궤도 방식임을 감안해 움직임을 정밀하게 모사하였다. 사람이 대신 운행하더라도, 실제 접이식 상·하역 장치의 동역학을 최대한 정확하게 반영하기 위해서 우선 실제 장치의 모든 회전 명령 및 전후 이동 명령에 대한 가능한 제어 입력과 그 구동 결과를 수치로 기록하였고, 이를 [Fig. 10]의 실선으로 시각화 하였다. 회전 명령의 경우 구동시간을 1초부터 10초까지 0.1초씩 증가 시키며 회전 각도를 측정하고, 전후 이동 명령의 경우 구동시간을 2초부터 10초까지 0.2초씩 증가시키며 이동거리를 측정하였다. 측정결과 초기에 가속구간이 있었고, 일정시간 후에 등속구간이 있고, 멈춘 후에도 관성에 의한 움직임이 있음을 확인했고, 이 가변적인 속도를 포함한 결과를 기반으로 각도와 시간에 관한 동역학 모델을 근사해 이를 식 (5)와 (6)으로 정리하였다.

$$t = -7.1088 \cdot 10^{-6}a^4 + 1.1326 \cdot 10^{-3}a^3 - 0.055315a^2 + 1.1253a - 3.7986 \quad (5)$$

$$t = -1.1 \cdot 10^{-8}d^4 + 5.22 \cdot 10^{-6}d^3 - 0.0008797 \cdot d^2 + 0.1079 \cdot d + 1.9 \quad (6)$$

식 (5)에서 a는 원하는 회전 각도를 의미하고 t는 a 만큼 회전시키기 위한 제어 입력 시간을 의미한다. 식 (6)에서 d는 원하는 직선 이동 거리를 의미하고 t는 d만큼 이동시키기 위한 제어 입력 시간을 의미한다. 손 지게차를 이용해 테스트 할 때 해당 식을 이용해 동역학을 근사하였으며, [Fig. 10]의 실선과 점선으로 표현된 결과를 보면 근사모델이 해당 범위 내에서 실측값을 충분히 잘 반영하는 것을 알 수 있다. [Fig. 11]는 실시간 테스트 영상을 캡처한 결과이다.

[Fig. 11(a)]를 보면 팔레트의 중심과 이미지의 중심이 일치하지 않는 것을 볼 수 있다. 중심을 맞추는 평행이동을 위해 Move Left의 명령을 한다. 이때 좌/우 상대위치 연산을 통해 계산된 거리만큼 이동하게 되고 중심이 일치 할 때까지 이동한다. [Fig. 11(b)]를 보면 팔레트와 이미지의 중심이 일치한 것을 볼 수 있다.

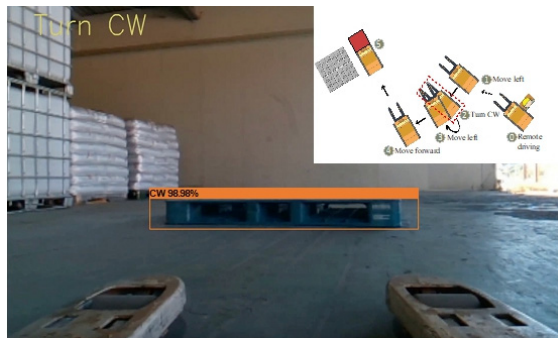
[Fig. 11(b)]에서 팔레트가 클래스 CW로 분류된 것을 볼 수 있다. 이는 팔레트와 지게차가 수평이 아니라는 것을 의미하고, 그림의 좌측상단을 보면 시계방향으로 각도조절 하라는 명령을 내린 것을 볼 수 있다. 직선 검출을 통해 계산된 각도만큼 각도 조절을 하고 각도 조절은 클래스가 Aligned로 분류될 때까지 한다. [Fig. 11(c)]를 보면 팔레트가 Aligned로 분류된 것을 확인할 수 있다.

[Fig. 11(c)]에서 팔레트가 Aligned로 분류됐다는 것은 팔레트와 지게차가 수평임을 의미한다. 하지만 팔레트가 현재 이미지의 중앙에 위치하지 않기 때문에 좌측 상단에 명령으로

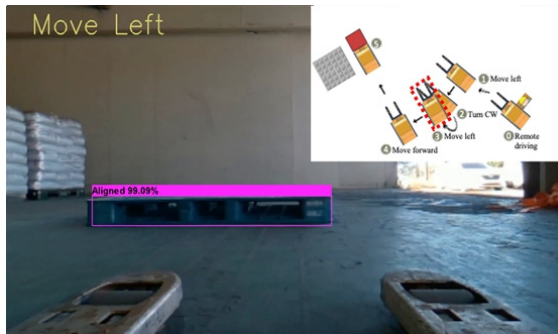
Move left가 출력된 것을 볼 수 있다. 이때 인식된 팔레트 영역의 중앙과 이미지 중심의 차이를 계산해서 일치할 때까지 이동한다. [Fig. 11(d)]를 보면 팔레트 영역이 이미지 중앙에 위치한 것을 볼 수 있다.



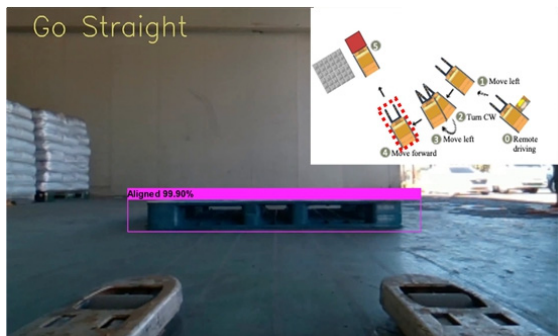
(a) Center misalignment, command: Move Left



(b) Center match & class CW, command: Turn CW



(c) Center misalignment, command: Move Left



(d) Center aligned & class Aligned, command: Move Forward

[Fig. 11] Result of test scenario

마지막으로 [Fig. 11(d)]을 보면 팔레트 인식결과 Aligned이고 팔레트가 이미지의 중앙에 위치하고 있기 때문에 Go Straight 명령이 출력된 것을 확인할 수 있고 팔레트까지의 거리를 영상의 깊이 정보로부터 계산해 전진하며 포크를 삽입한다.

최종적으로 14개의 서로 다른 시작 환경을 갖는 시나리오에 대해, 무작위로 10종 중 한 종의 팔레트를 선택하고, 시나리오 당 3회씩 반복 테스트한 결과, 모든 경우 포크 삽입까지 성공적으로 자동 제어되는 것을 확인했다. 단, 초기에는 사용자가 원격으로 팔레트가 보이는 위치까지 지게차를 조종한다는 가정에 따라, 시작 환경에서 팔레트가 보이지 않는 경우는 배제하였다.

## 5. 결론

제안된 차량 탑재형 상·하역 장비와 딥러닝 기반의 자동 제어 시스템을 통해, 소규모 물류 창고 및 상·하역 환경에서의 효율성을 향상시키는 연구를 제안했다. 제안된 시스템을 통해 작은 규모의 물류 창고 및 상·하역 환경에서 효율성을 향상시키고, 수작업에 비해 물류 작업 속도를 향상시킬 수 있다. 또한 원격 제어와 딥러닝 모델을 결합하여, 작업자의 필요한 숙련도를 줄이고 팔레트를 정확하게 식별하고 추적할 수 있다. 다양한 팔레트의 크기, 색상, 모양 등을 고려하여 모델을 최적화하여 다양한 작업 시나리오에 적용할 수 있고, 이러한 기술을 도입함으로써, 노동 비용을 절감할 수 있다. 즉, 본 연구는 다양한 환경에서 노동과 비용을 절감하고 생산성을 향상시킬 가능성을 가지고 있다. 이러한 연구가 지속되어 향후 물류 자동화 및 로봇 기술산업이 발전할 것을 기대한다.

## References

- [1] G. Garibotto, S. Masciangelo, P. Bassino, and M. Ilic, "Computer vision control of an intelligent forklift truck," *Conference on Intelligent Transportation Systems*, Boston, MA, USA, pp. 589-594, 1997, DOI: 10.1109/ITSC.1997.660540.
- [2] M. Li, S. Gu, G. Chen, and Z. Zhu, "A RFID-based Intelligent Warehouse Management System Design and Implementation," *2011 IEEE 8th International Conference on e-Business Engineering*, Beijing, China, pp. 178-184, 2011, DOI: 10.1109/ICEBE.2011.28.
- [3] M. Seelinger and J.-D. Yoder, "Automatic Pallet Engagment by a Vision Guided Forklift," *2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 4068-4073, 2005, DOI: 10.1109/ROBOT.2005.1570744.
- [4] J. Pagès, X. Armangué, J. Salvi, J. Freixenet, and J. Martí, "A Computer Vision System for Autonomous Forklift Vehicles in Industrial Environments," *9th Mediterranean Conference on Control and Automation MEDS*, Dubrovnik, Croatia, pp. 1-6, 2001, [Online], <http://eia.udg.es/~qsalvi/papers/2001-MEDS.pdf>.

- [5] T. Li, B. Huang, C. Li, and M. Huang, "Application of convolution neural network object detection algorithm in logistics warehouse," *The Journal of Engineering*, vol. 2019, no. 23, pp. 9053-9058, Dec., 2019, DOI: 10.1049/joe.2018.9180.
- [6] Y.-Y. Li, X.-H. Chen, G.-Y. Ding, S. Wang, W.-C. Xu, B.-B. Sun, and Q. Song, "Pallet detection and localization with RGB image and depth data using deep learning techniques," *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Dalian, China, pp. 306-310, 2021, DOI: 10.1109/CACRE52464.2021.9501390.
- [7] J. Ren, Y. Pan, P. Yao, Y. Hu, W. Gao, and Z. Xue, "Deep Learning-Based Intelligent Forklift Cargo Accurate Transfer System," *Sensors*, vol. 22, no. 21, pp. 8437, Nov., 2022, DOI: 10.3390/s22218437.
- [8] S. V. Carata, M. Ghenescu, and R. Mihaescu, "Real-Time Detection of Unrecognized Objects in Logistics Warehouses Using Semantic Segmentation," *Mathematics*, vol. 11, no. 11, pp. 2445, May, 2023, DOI: 10.3390/math11112445.
- [9] H. Yin, C. Chen, C. Hao, and B. Huang, "A Vision-based inventory method for stacked goods in stereoscopic warehouse," *Neural Computing and Applications*, vol. 34, pp. 20773-20790, Jul., 2022, DOI: 10.1007/s00521-022-07551-4.
- [10] C. Mok, I. Baek, Y. S. Cho, Y. Kim, and S. B. Kim, "Pallet recognition with multi-task learning for automated guided vehicles," *Applied Sciences*, vol. 11, no. 24, Dec., 2021, DOI: 10.3390/app112411808.
- [11] Y. Shao, Z. Fan, B. Zhu, M. Zhou, Z. Chen, and J. Lu, "A Novel Pallet Detection Method for Automated Guided Vehicles Based on Point Cloud Data," *Sensors*, vol. 22, no. 20, Oct., 2022, DOI: 10.3390/s22208019.
- [12] M. Zaccaria, R. Monica, and J. Aleotti, "A Comparison of Deep Learning Models for Pallet Detection in Industrial Warehouses," *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, pp. 417-422, 2020, DOI: 10.1109/ICCP51029.2020.9266168.
- [13] Y. Mo, Z. Sun, and C. Yu, "EventTube: An Artificial Intelligent Edge Computing Based Event Aware System to Collaborate With Individual Devices in Logistics Systems," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1823-1832, Feb., 2023, DOI: 10.1109/TII.2022.3189177.
- [14] M. Liu, X. Xu, X. Wang, Q. Jiang, and C. Liu, "Intelligent monitoring method of tridimensional storage system based on deep learning," *Environmental Science and Pollution Research*, vol. 29, pp. 70464-70478, May, 2022, DOI: 10.1007/s11356-022-20658-4.
- [15] H. Xiong, J. Wu, Q. Liu, and Y. Cai, "Research on abnormal object detection in specific region based on Mask R-CNN," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, May, 2020, DOI: 10.1177/1729881420925287.
- [16] Z. Li, K. Lu, Y. Zhang, Z. Li, and J.-B. Liu, "Research on Energy Efficiency Management of Forklift Based on Improved YOLOv5 Algorithm," *Journal of Mathematics*, vol. 2021, Dec., 2021, DOI: 10.1155/2021/5808221.
- [17] J. Redmon, and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, Apr., 2018, DOI: 10.48550/arXiv.1804.02767.
- [18] T. Allison, "Freight-handling technologies and industrial building design: freighthouse and warehouse facilities of the Chicago junction railway, 1900-30," *Industrial Archaeology Review*, vol. 36, no. 2, pp. 109-127, Dec., 2014, DOI: 10.1179/0309072814Z.00000000034.
- [19] A. Anđelković and M. Radosavljević, "Improving order-picking process through implementation of warehouse management system," *Strategic Management*, vol. 23, no. 1, pp. 3-10, Jan., 2018, DOI: 10.5937/STRAMAN1801003A.
- [20] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1-41, Jan., 2013, DOI: 10.1007/s11721-012-0075-2.
- [21] M. Al-Obaidy and R. Al-Azawi, "Cluster-based Algorithm for Energy Optimization of Swarmed Robots using Swarm Intelligence," *2019 Sixth HCT Information Technology Trends (ITT)*, Ras Al Khaimah, United Arab Emirates, pp. 202-207, 2019, DOI: 10.1109/ITT48889.2019.9075119.
- [22] J. A. Cano, F. Salazar, R. A. Gómez-Montoya, and P. Cortés, "Disruptive and conventional technologies for the support of logistics processes: a literature review," *International Journal of Technology*, vol. 12, no. 3, pp. 448-460, Jul., 2021, DOI: 10.14716/ijtech.v12i3.4280.
- [23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 580-587, 2014, DOI: 10.1109/CVPR.2014.81.
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Computer Vision-ECCV 2016: 14th European Conference*, Amsterdam, Netherlands, pp. 11-14, 2016, DOI: 10.1007/978-3-319-46448-0\_2.
- [25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779-788, 2016, DOI: 10.1109/CVPR.2016.91.
- [26] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 7263-7271, 2017, DOI: 10.1109/CVPR.2017.690.
- [27] G. Scheithauer and J. Terno, "A new heuristic for the pallet loading problem," *Operations Research Proceedings 1995*, Passau, Germany, pp. 84-89, 1996, DOI: 10.1007/978-3-642-80117-4\_15.
- [28] E. G. Birgin, R. Morabito, and F. H. Nishihara, "A note on an L-approach for solving the manufacturer's pallet loading problem," *Journal of Operations Research Society*, vol. 56, no. 12, pp. 1448-1451, Mar, 2005, DOI: 10.1057/palgrave.jors.2601960.
- [29] L. Wang, S. Shang, and Z. Wu, "Research on Indoor 3D Positioning Algorithm Based on WiFi Fingerprint," *Sensors*, vol. 23, no. 1, pp. 153, Dec., 2022, DOI: 10.3390/s23010153.

[30] K. C. Wu and C. J. Ting, "A two-phase algorithm for solving the manufacturer's pallet loading problem," *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, Singapore, pp.1574-1578, 2007, DOI: 10.1109/IEEM.2007.4419457.

[31] S. Ahn, K. Yoon, and J. Park, "A best-first branch and bound algorithm for the pallet-loading problem," *International Journal of Production Research*, vol.53, no. 3, pp.835-849, Jul., 2014, DOI: 10.1080/00207543.2014.935824.

[32] H. Zhang, Q. Wang, C. Yan, J. Xu, and B. Zhang, "Research on UWB Indoor Positioning Algorithm under the Influence of Human Occlusion and Spatial NLOS," *Remote Sensing*, vol. 14, no. 24, pp. 6338, Dec., 2022, DOI: 10.3390/rs14246338.



**이 순 교**

2011 고려대학교 전자및정보공학(공학사)  
 2011 고려대학교 경영학(경영학사)  
 2020 고려대학교 산업경영공학(공학박사)  
 2021 한국해양수산개발원 전문연구원  
 2021~현재 한국철도기술연구원 선임연구원

관심분야: Logistic System, Supply Chain Finance, Embedded System



**이 석**

1995 성균관대학교 산업공학과(공학사)  
 1997 KAIST 산업공학과(공학석사)  
 2002 KAIST 산업공학과(공학박사)  
 2015~현재 한국철도기술연구원 책임연구원

관심분야: 물류자동화, 물류센터 운영 최적화, 철도물류 효율화



**김 선 목**

2017 광운대학교 전기공학과(공학사)  
 2020 광운대학교 전기공학과(박사수료)  
 2020~현재 광운대학교 전기공학과(박사)

관심분야: 머신러닝, 딥러닝, 컴퓨터비전, 자율주행



**이 기 백**

2005 KAIST 전기및전자공학과(공학사)  
 2014 KAIST 전기및전자공학과(공학박사)  
 2014~현재 광운대학교 부교수

관심분야: Intelligent Robot, Artificial Intelligence, Computational Intelligence



**우 효 원**

2022 서울과학기술대학교 전자공학과(공학사)  
 2022~현재 광운대학교 전기공학과(석사)

관심분야: 딥러닝, 컴퓨터비전, 자율주행, 안티드론