

ISSN 2765-2351

# 물류 과학 기술 연구

Journal of Logistics Science & Technology

2022년 12월 3권 2호

THE SOCIETY OF LOGISTICS  
SCIENCE AND TECHNOLOGY

# 물류과학기술연구

Journal of Logistics Science & Technology

제 3권 제 2호

2022년 12월

## 목 차

1. 정시배송가능 시간대(time window) 제시를 위한 다중정류소(multi-depot) 다회전(multi-trip) 차량경로문제: 온라인 할당 및 오프라인 재할당 접근법

이기주, 정동진, 채준재 / 1

2. UAM과 드론을 사용한 도시물류상의선택적 이중구조 오리엔티어링 문제

김동균, 차형주, 정태수 / 18

3. 추가학습이 불필요한 이미지 특징 유사도 기반상품 식별 시스템

유영재, 윤혜정, 김준오, 박예술, 장병탁 / 36

4. 횡단보도 보행자 인지 향상을 위한ByteTrack 기반 다중객체추적

노유진, 이유진, 이상민 / 56

5. 도시철도 물류 시스템의 통합관제시스템 설계 및 수평이송장치 경로 생성 알고리즘 연구

황선우, 김주욱, 김영민, 최준우, 한수민, 박상원, 장경호 / 78

# 정시배송가능 시간대(time window) 제시를 위한 다중정류소(multi-depot) 다회전(multi-trip) 차량경로문제: 온라인 할당 및 오프라인 재할당 접근법

이기주, 정동진, 채준재<sup>†</sup>  
한국항공대학교 항공교통물류학과

## Multi-depot Multi-trip Vehicle Routing Problem for Offering Available Delivery Time Windows: Online Assignment and Offline Reassignment Approach

Keyju Lee, Dongjin Jeong, Junjae Chae<sup>†</sup>  
School of Air Transport, Transportation and Logistics, Korea Aerospace University

The last mile delivery efficiency is becoming prominent, as its demand is boosting. As the last mile delivery provider and customer are both interested in differentiated services, on-demand delivery and time-reserved delivery services are receiving attention. This research characterizes the problem into multi-depot multi-trip vehicle routing problem for offering available delivery time windows (MD&T VRPOTW). A mathematical formulation for the problem that includes previous schedule constraints and current-time constraints is provided. A strategy for reducing complexity is suggested. The problems were solved using CPLEX, and the complexity reduction was proven effective in online assignments. The experiments on offline re-assignment showed that over 10% of delivery distance (time) can be reduced when the schedules were re-assigned or re-optimized after about 10 delivery scheduled had been accumulated. This research and directed future research should contribute to the improvement of on-demand delivery and time-reserved delivery services.

**Keywords:** On-demand Delivery, Vehicle Routing, Vehicle Assignment, Vehicle Reassignment

---

논문접수일 : 2022.10.08.

심사완료일 : 2022.12.19.

게재확정일 : 2022.12.19.

본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호 22HCLP-C163182-02)

<sup>†</sup> Corresponding Author: jchae@kau.ac.kr

## 1. 서론

전세계 도시지역의 주요한 화두인 라스트마일(Boysen et al., 2021)은 물류 및 유통업계에서 상품이 소비자에게 전달되는 마지막 단계를 의미하며, 라스트마일배송의 속도, 품질, 편의성이 소비자에게 차별화된 경험을 제공하기 때문에 그 중요도가 커지고 있다(Korea Maritime Institute, 2022). 특히 코로나 19의 영향으로 전자상거래의 비중이 급속하게 증가하였으며(Bhatti et al., 2020), 이는 라스트마일배송에 대한 수요 증가로 이어졌다. 대한민국에서 코로나 19가 본격 확산된 2020년 택배물량은 전년 대비 약 21%의 급등세를 나타냈다(Park et al., 2021). 대한민국의 전자상거래는 소매시장의 37%를 차지하고 있으며, 음식배달과 즉시배송과 같은 서비스에 힘입어 향후 5년간 45%까지 증가할 것으로 전망된다(Morgan Stanley, 2022).

Dablanc et al. (2017)의 연구에 의하면 도시지역에서의 이동(mobility)환경은 Uber(2020)와 같이 스마트폰 앱을 기반으로 하는 주문형(on-demand) 운송 서비스로 변형되었고, 이러한 혁신은 도시지역의 화물운송 즉, 물류부문까지 재편할 것으로 예상된다. 해당 연구에서는 실시간 배송요청과 라스트마일배송에 많은 잠재적 개선 가능성이 존재함을 드러냈다. 국내에서도 고객이 원하는 시간대(time window)에 예약배송하는 서비스(Korea Logistics News, 2022; SK Telecom, n.d.-a) 또는 즉시배송 서비스(Kyunghyang, 2022; Maeil Business, 2021)와 같이 차별화된 배송서비스에 대한 관심과 수요가 증가하고 있다. 시간대예약배송 서비스와 즉시배송 서비스는 고객이 문을 열어줘야 하는 경우(가전제품 방문설치 및 점검, 가구의 배송과 조립 및 설치, 대형폐가전 방문수거 등) 및 고객과의 대면이 필요한 경우(자동차용품 방문장착 등)와 같이 고객이 배송지에 있을 예정인 상황(attended home delivery)에서 활용될 수 있다. 현재 이러한 서비스 대부분은 방문기사의 전화응대를 통해 계획되고 있으며, 이에 대한 효율적인 배차 솔루션을 제시할 수 있다면 자원관리의 개선과 운영비용 감소 효과를 기대할 수 있다.

본 연구에서는 이러한 문제형태를 정시배송가능 시간대(time window) 제시를 위한 다중정류소(multi-depot) 다회전(multi-trip) 차량경로문제(vehicle routing problem)로 정의한다. 차량경로문제는 거점에서 출발한 다수의 차량이 배송 등의 서비스를 요구하는 고객을 특정한 순서대로 방문하고 다시 거점으로 복귀하는 최소 비용의 경로를 결정하는 문제이다. 본 연구에서 제시하는 차량경로문제 유형은 다수의 거점(multi-depot)에 다수의 차량이 소속되어 있는 상황에서 임의의 위치에서 실시간으로 발생하는 고객의 배송요청에 대응한다. 각 차량의 기확정된 배송스케줄을 고려했을 때 고객위치로 배송 가능한 모든 시간대(time window)를 도출한 뒤 고객에게 제시(offer)하며, 고객은 제시된 시간대 중 하나를 선택할 수 있다(Figure 1). 각 차량은 소속된 정류소에서 출발하여 돌아올 때까지 여러 고객을 서비스할 수 있으나, 차량의 용량제약이 고려되어야 한다. 각 차량은 출차 및 회차를 반복하는 다회전(multi-trip) 배차가 가능하며, 마지막에는 소속된 정류소로 되돌아와야 한다. 본 연구의 수리모델 작성은 온라인(실시간) 할당의 특성인 빠른 응답시간 등을 반영하기 위해 고객의 배송요청과 차량스케줄이 순차적으로 누적되는 상황을 가정하였으며, 기확정된 할당은 유지되도록 제약하였다. 이러한 가정과 제약으로 배송요청이 누적되어감에 따라 계획이 지역최적해를 나타낼 수 있다. 따라서, 주기적 오프라인(비실시간) 재할당을 통해 전역최적해로 수정하기 위한 전략을 함께 제시한다.

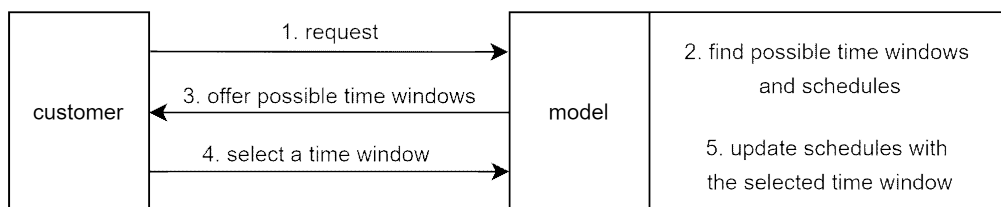


Figure 1. A simplified process of delivery request and offering possible time windows

## 2. 문헌연구

### 2.1. 관련 연구

Uber(2020)의 서비스로 대표되는 실시간 차량할당문제 즉, 승차공유(ride-sharing) 또는 차량호출(ride-hailing)에 대한 연구가 활발하게 이루어지고 있다. 해당 문제는 여정-차량(trip-vehicle) 할당문제(S. Li et al., 2020; Mori & Samaranayake, 2021) 또는 실시간배차(online vehicle dispatch)(Zhao et al., 2019) 문제로 정의된다. 여정-차량 할당문제에서의 여정(trip)은 승객이 설정한 기점에서 종점으로의 이동을 의미하며, 해당 여정을 가장 효율적으로 수행할 수 있는 차량에 할당하는 것을 목표로 한다. Alonso-Mora et al.(2017)의 연구에서는 여정-차량 할당문제를 다수 승객의 동승이 가능한 형태로 일반화했고, 실시간으로 큰 규모의 문제 해결을 위한 알고리즘을 제시했다. Zhou & Roncoli(2022)의 연구 또한 큰 규모의 문제에 대한 효율적인 솔루션을 제시했다. 그들의 연구에서는 여정-차량 할당문제를 더욱 현실적으로 발전시켜 도로상의 교통체증과 여정-차량 할당 자체에서 발생하는 내인성(endogenous) 교통체증을 함께 고려할 수 있도록 하였다.

여정-차량 할당문제와 유사한 형태로 음식배달경로(food/meal delivery routing) 문제(Reyes et al., 2018)가 있다. 해당 문제에서는 음식이 준비된 이후에 픽업이 가능하다는 조건이 추가적으로 고려되며, 배달완료까지 소요되는 시간을 제약하거나 초과시간을 최소화하기 위해 노력한다. 음식이 준비되는 시간을 확률적으로 고려한 연구(Ulmer et al., 2021), 장거리 배달 시의 환적(transfer)을 고려한 연구(J. Li et al., 2022) 등 다양한 파생연구가 이루어졌다. 음식배달경로 문제의 최적해를 실시간으로 도출하는 것은 현실적으로 어려우며 빠른 응답시간을 보장하기 위해 구조적(constructive) 알고리즘 연구(Wang et al., 2021), 딥강화학습을 활용한 연구(Jahanshahi et al., 2022) 등을 포함하여 기계학습된 모델을 활용한 솔루션이 제시되었다.

동적(dynamic)차량경로문제(Ojeda Rios et al., 2021; Pillac et al., 2013)는 확률적으로 존재하는 값들이나 시간이 지남에 따라 드러나게 되는 상황에 대응하기 위한 차량경로문제를 포괄하는 개념이며, 특히 최근 지리정보시스템 등의 기술발전으로 인해 주목받고 있다. 동적차량경로문제의 한가지 유형으로, 미리 계획된 차량경로를 실행하는 도중에 실시간으로 발생하는 새로운 주문들을 경로에 포함시키거나 새로운 차량을 할당하고자 하는 문제가 포함된다. 새로운 주문들은 예측된 경우(Predictive)와 예측되지 않은 경우(Unexpected)가 혼합될 수 있으며(Kucharska, 2019), 동적 할당에 의해 발생할 수 있는 비효율은 재최적화(re-optimization)를 과정을 통해 개선할 수 있다(AbdAllah et al., 2017).

시간대(time window)제약 차량경로문제(El-Sherbeny, 2010)에서는 각 고객이 배송을 원하는 시간대가 설정되어 있고(예를 들어, 17시~18시), 다수의 차량이 모든 고객의 배송시간대를 준수하거나, 미준수율을 최소화하여 배송을 완료하고자 한다. 해당 유형의 문제에 대해 정확한(exact) 알고리즘(Baldacci et al., 2012)과 메타-휴리스틱 알고리즘(Dixit et al., 2019) 등 다양한 접근법이 제시되었다. 또한, 시간대제약 차량경로문제와 명칭이 유사하지만 그 내용이 다른 시간대할당 차량경로문제(Spliet & Gabor, 2015)가 있다. 시간대할당 차량경로문제에서의 고객은 매일 또는 반복적으로 방문해야 하는 소매점이며, 고객이 시간대를 설정하는 것과는 반대로 공급자가 배송시간대를 약속하기 위한 문제이다. 시간대할당 차량경로문제는 할당할 고객배송시간을 연속적(continuous)이지 않은 이산(discrete)시간으로 가정한 연구(Spliet & Desaulniers, 2015), 제품군의 의존성을 고려한 연구(Neves-Moreira et al., 2018) 등 소매점 운영의 현실성을 보다 구체적으로 고려할 수 있도록 발전했으며, 정확한 해법으로 분기절단법(branch-and-cut) 알고리즘이 제시되었다(Dalmeijer & Spliet, 2018).

## 2.2. 차별점 및 연구 기여

앞서 분석한 관련 연구와의 비교를 통해, 아래 Table 1에 나타난 바와 같이 본 연구(정시배송가능 시간대 제시를 위한 다중정류소 다회전 차량경로문제, MD&T VRPOTW)의 차별점을 확인할 수 있다. MD&T VRPOTW에서 고객이 요청하는 것은 여정이 아닌 도달(arrival) 또는 배송(delivery)이며, 실시간 대응이 가능하다. 고객의 위치와 수요량은 미리 알 수 없고, 무작위로 발생하게 된다. 고객의 요청이 최대한 빨리 또는 즉시(on-demand) 충족되어야 하는 문제유형들과는 다르게 고객이 선택한 시간에 맞춰 정시(on-time)배송해야 한다. 만약, 고객에게 제시된 시간대 중 가장 이른 시간이 선택된다면 즉시배송의 형태에도 대응될 수 있다. MD&T VRPOTW는 배송가능 시간대를 고객에게 제시(offer)하기 위한 모델이며, 고객은 배송가능한 것으로 제시된 다수의 시간대 중에서 하나를 선택할 수 있다. 다중정류소가 고려되어 다수의 차량출발 위치가 존재하며, 각 차량은 소속 정류소를 출발하고 복귀하기를 반복하는 다회전 배차가 가능하다.

Table 1. Comparison of MD&T VRPOTW and other related works

	Trip-vehicle assignment problem (TVAP)	Meal delivery routing problem (MDRP)	Dynamic VRP (DVRP)	VRP with time window (VRPTW)	Time window assignment VRP (TWAVRP)	Multi-depot/trip VRP for offering available delivery time windows (MD&T VRPOTW)
Request	Trip	Trip	Arrival	Arrival	Arrival	Arrival
Real-time	O	O	O	X	X	O
Unknown customer	O	O	X/O	X	X	O
On-demand /On-time	Demand	Demand	Demand	Time	Time	Demand/Time
Arrival time decision	Provider	Provider	Provider	Customer	Provider	Customer
# of Arrival time options	Single	Single	Single	-	Single	Multiple
Vehicle start locations	Multiple	Multiple	Single	Single	Single	Multiple
Vehicle return	X	X	O	O	O	O
Vehicle capacity	X	O	O	O	O	O

## 3. MD&T VRPOTW 문제의 모형화

본 연구에서는 MD&T VRPOTW를 혼합정수계획법으로 구성하였다. 해당 문제는  $V$ 가 노드의 집합,  $A = \{(i, j) | i, j \in V, i \neq j\}$ 가 아크의 집합을 의미하는 그래프  $G = (V, A)$  상에서 정의된다. 이외에 MD&T VRPOTW 문제 모형화를 위해 필요한 표기법(notation)을 아래 Table 2에 제시하였다. 두 노드  $(i, j)$  사이의 이동비용인  $c_{i,j}$ 는 차량의 이동 소요시간을 의미한다. 노드 집합은  $V = V_c \cup V_{dstart} \cup V_{dend} \cup V_{dcopy}$ 의 합으로 구성되며, 각 집합은 고객 노드  $V_c = \{1, \dots, n\}$ , 출발정류소 노드  $V_{dstart} = \{n+1, \dots, n+w\}$ , 도착정류소 노드  $V_{dend} = \{n+w+1, \dots, n+2w\}$ , 복제정류소 노드  $V_{dcopy} = \{n+2w+1, \dots, n+2w+(n-1)w\}$ 로 표현된다. 해당 모델에서  $w$ 는 다중정류소(multi-depot)의 개수를 의미한다. 출발정류소 노드와 도착정류소 노드는 물리적으로 동일한 지리좌표 상에 존재하지만, 수리모델 구성의 편의성을 위해 구분하였다. 복제정류소 노드 또한 출발 및 도착정류소와 동일한 지리좌표를 가지며, 복제정류소는 차량의 다회전(multi-trip) 배차를 모델링하기 위해 존재한다. 즉, 각 차량의 경로는 반드시 출발정류소 노드를 출발하여 마지막에는 도착정류소 노드에 도달해야하며, 그 사이에 고객노드와 복제정류소 노드를 포함할 수 있도록 구성된다.

각 노드  $i$ 에 대한 수요량을  $p_i$ , 서비스 소요시간을  $s_i$ , 도착해야 하는 시간대의 시작시간(earliest)을  $e_i$ , 도착해야 하는 시간대의 종료시간(latest)을  $l_i$ 로 정의한다. 노드  $i$ 가 고객 노드 집합  $V_c$ 에 포함되지 않는 경우는 모두 정류소를 나타내므로  $p_i$ 와  $s_i$  값이 모두 0으로 설정된다. 노드  $i$ 가  $V_{dstart}$ 에 포함된다면  $e_i$ 와  $l_i$  값은 모두 0으로 설정하며,  $V_{dstart}$  및  $V_{dcopy}$ 에 포함된다면  $e_i$ 는 0으로,  $l_i$ 는 차량의 업무종료시간으로 설정한다(예를 들어, 분 단위로 표현하되 10시간 근무한다면 600으로 설정). 차량은 각 정류소에 소속되어 있는 차량들의 합집합인  $K = K_1 \cup \dots \cup K_w = \{1, \dots, k\}$ 로 표기하며( $K_1$ 은 1번 정류소에 소속된 차량 집합), 모든 차량은 동일한 용량인  $Q$ 를 가진다.

해당 수리모델의 결정변수로는  $x_{i,j,k}$ ,  $t_{i,k}$ ,  $q_{i,k}$  세 가지가 있다. 노드  $(i,j)$ 로의 이동을 차량  $k$ 가 수행하는 경우  $x_{i,j,k}$ 의 값은 1이며, 그렇지 않으면 0의 값을 가진다.  $t_{i,k}$ 의 값은 노드  $i$ 에 차량  $k$ 가 도달했을 때의 시간을 나타내며,  $q_{i,k}$ 는 노드  $i$ 에 차량  $k$ 가 도달했을 때 차량  $k$ 에 적재된 용량을 나타낸다. MD&T VRPOTW 수리모델의 목적함수와 제약조건을 아래와 같이 표현한다.

Table 2. Notations used for modeling MD&T VRPOTW

Index	Definition
$i$	index for nodes
$j$	index for nodes
$w$	index for multiple depots
$k$	index for vehicles
Set	Definition
$V$	set of all nodes, $V_c \cup V_{dstart} \cup V_{dend} \cup V_{dcopy}$
$V_c$	set of customer nodes
$V_{dstart}$	set of starting depot nodes, $V_{dstart_1} \cup \dots \cup V_{dstart_w}$
$V_{dend}$	set of ending depot nodes, $V_{dend_1} \cup \dots \cup V_{dend_w}$
$V_{dcopy}$	set of copied depot nodes, $V_{dcopy_1} \cup \dots \cup V_{dcopy_w}$
$V_{dcopy_w}$	set of copied depot nodes at depot $w$
$K$	set of all vehicles, $K_1 \cup \dots \cup K_w$
$K_w$	set of vehicles at depot $w$
$S'$	set of $(i,j,k)$ tuples of $x_{i,j,k} = 1$ in the previous solution (schedule)
Parameter	Definition
$c_{i,j}$	cost (time) of traveling from node $i$ to $j$
$s_i$	service time required at node $i$
$p_i$	amount of load to be delivered to node $i$
$e_i$	earliest possible time for arriving at node $i$
$l_i$	latest possible time for arriving at node $i$
$Q$	vehicle capacity for the amount of load
$T$	a number to denote the present time
$M$	a very large positive number
Decision Variable	Definition
$x_{i,j,k}$	1 if vehicle $k$ travels from $i$ to $j$ , 0 otherwise
$t_{i,k}$	arrival time of vehicle $k$ at node $i$
$q_{i,k}$	amount of available space on vehicle $k$ after servicing node $i$

$$\text{minimize } Z = \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{i,j} \times x_{i,j,k} \quad (1)$$

$$\text{subject to } \sum_{i \in V - V_{dend}} \sum_{k \in K} x_{i,j,k} = 1, \forall j \in V_c \quad (2)$$

$$\sum_{j \in V - V_{dstart}} \sum_{k \in K} x_{i,j,k} = 1, \forall i \in V_c \quad (3)$$

$$\sum_{i \in V} x_{i,j,k} = \sum_{i \in V} x_{j,i,k}, \forall j \in V_c \cup V_{dcopy}, k \in K \quad (4)$$

$$\sum_{j \in V_c \cup V_{dend_w} \cup V_{dcopy_w}} x_{i,j,k} = 1, \forall i \in V_{dstart_w}, k \in K_w \quad (5)$$

$$\sum_{i \in V_c \cup V_{dstart_w} \cup V_{dcopy_w}} x_{i,j,k} = 1, \forall j \in V_{dend_w}, k \in K_w \quad (6)$$

$$x_{i,j,k} = 0, \forall i \in V_{dstart_w}, j \in (V_{dend} - V_{dend_w}) \cup (V_{dcopy} - V_{dcopy_w}), k \in K_w \quad (7)$$

$$x_{i,j,k} = 0, \forall i \in (V_{dstart} - V_{dstart_w}) \cup (V_{dcopy} - V_{dcopy_w}), j \in V_{dend_w}, k \in K_w \quad (8)$$

$$x_{i,j,k} = 0, \forall i \in V_{dstart_w}, j \in V, k \notin K_w \quad (9)$$

$$x_{i,j,k} = 0, \forall i \in V, j \in V_{dend_w}, k \notin K_w \quad (10)$$

$$x_{i,j,k} = 0, \forall i \in V_c, j \in V_{dend_w} \cup V_{dcopy_w}, k \notin K_w \quad (11)$$

$$x_{i,j,k} = 0, \forall i \in V, j \in V_{dstart}, k \in K \quad (12)$$

$$x_{i,j,k} = 0, \forall i \in V_{dend}, j \in V, k \in K \quad (13)$$

$$t_{i,k} + s_i + c_{i,j} \leq t_{j,k} + M \times (1 - x_{i,j,k}), \forall i \in V, j \in V_c \cup V_{dend} \cup V_{dcopy}, k \in K \quad (14)$$

$$t_{i,k} \geq 0, \forall i \in V, k \in K \quad (15)$$

$$t_{i,k} \geq e_i, \forall i \in V, k \in K \quad (16)$$

$$t_{i,k} \leq l_i, \forall i \in V, k \in K \quad (17)$$

$$q_{i,k} \geq 0, \forall i \in V, k \in K \quad (18)$$

$$q_{i,k} \leq Q, \forall i \in V, k \in K \quad (19)$$

$$q_{i,k} + p_j \leq q_{j,k} + M \times (1 - x_{i,j,k}), \forall i \in V, j \in V_c, k \in K \quad (20)$$

$$q_{i,k} + p_j \leq Q + M \times (1 - x_{i,j,k}), \forall i \in V, j \in V_c, k \in K \quad (21)$$

$$q_{i,k} + p_j + M \times (1 - x_{i,j,k}) \geq 0, \forall i \in V, j \in V_c, k \in K \quad (22)$$

$$\sum_{i \in V} x_{i,j,k} = 1, \forall (i,j,k) \in S' \quad (23)$$

$$t_{i,k} + M \times x_{i,j,k} \geq T, \forall (i,j,k) \in S' \quad (24)$$

수식(1)에 나타난 MD&T VRPOTW 수리모델의 목적함수는 각 차량의 운행거리(시간) 합계를 최소화하고자 한다. 제약조건 (2)와 (3)에 의하면 모든 고객 노드에 대해 정확히 한 번의 유입(inflow)과 유출(outflow)이 존재해야 한다. 즉, 모든 고객은 반드시 한 번씩 서비스되어야 한다. 제약조건 (4)는 고객 노드 및 복제정류소 노드에 대해 유입량과 유출량이 일치하도록 한다. 제약조건 (5)에 의해 각 차량은 반드시 자신이 소속된 출발정류소 노드로부터 출발하며, 제약조건 (6)에 의해 각 차량은 반드시 자신이 소속된 도착정류소 노드로 복귀해야 한다. 제약조건 (7)은 특정 출발정류소 노드로부터 출발한 차량이 타



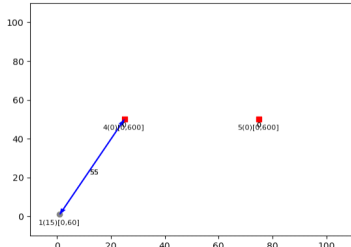
도착정류소 노드 또는 타 복제정류소 노드로 유입되지 않도록, 그리고 제약조건 (8)은 차량이 특정 도착정류소로 유입될 때 타 출발정류소 또는 타 복제정류소 노드로부터 접근한 것이 아니도록 제한한다. 제약조건 (9)는 차량이 소속되지 않은 출발정류소 노드로부터는 출발할 수 없도록, 제약조건 (10)은 차량이 소속되지 않은 거점으로는 복귀할 수 없도록 제한한다. 제약조건 (11)은 고객 노드로부터 도착정류소 노드 또는 복제정류소 노드로 유입될 때 차량의 정류소 소속이 일치해야 함을 나타낸다. 제약조건 (12)와 (13)은 출발정류소 노드로는 어떤 흐름도 유입되지 않도록, 도착정류소 노드로부터는 어떤 흐름도 유출되지 않도록 한다. 제약조건 (14)와 (15)는 서브투어제거(sub-tour elimination) 제약조건이며 Miller-Tucker-Zemlin(MTZ)의 구성방법(Desrochers & Laporte, 1991)을 활용하였다. 제약조건 (16)과 (17)은 고객 노드에 대해 설정된 도착시간대(시간구간)를 준수하도록 한다. 제약조건 (18)과 (19)는 차량이 경로상에서 가질 수 있는 적재량의 최소 및 최대량을 제약한다. 제약조건 (20)~(22)는 차량이 경로를 진행함에 따라 순차적으로 변경되는 적재량의 변화를 업데이트하기 위해 사용되었다. 제약조건 (23)과 (24)에 대해서는 이어지는 하위항목들에서 자세히 설명하도록 한다.

### 3.1. 기확정 스케줄 제약조건 (constraints for previous schedules)

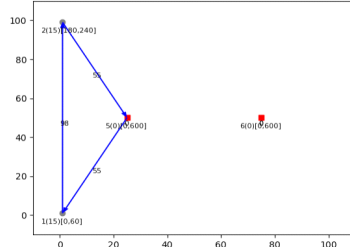
제약조건 (23)은 기확정 스케줄 제약조건을 나타낸다. 본 연구에서 제시하는 MD&T VRPOTW 모델은 실시간으로 들어오는 고객주문에 대한 스케줄링을 진행하며, 기확정된 스케줄을 누적시킨다. 이에 따라 해당 모델은 기확정 스케줄에 대한 제약조건을 포함하고 있다. 먼저, MD&T VRPOTW 모델을 통해 도출되는 차량의 경로 즉,  $x_{i,j,k} = 1$ 인  $(i,j,k)$  조합에 대한 집합을  $S$ 로 정의하고, 바로 직전 고객주문까지 도출된 차량의 경로  $x'_{i,j,k} = 1$ 인  $(i,j,k)$  조합에 대한 집합  $S'$ 를 정의한다(첫 번째 고객의 주문은 직전 주문이 존재하지 않으므로  $S' = \emptyset$  즉, 공집합으로 정의).

아래에 제시된 Figure 2(a)~(u)에서는 첫 번째 고객으로부터 스물한 번째 고객에 이르기까지 MD&T VRPOTW 모델에 의해 할당되고 생성된 배송경로를 예시로 나타내고 있다. 해당 예시에서는 두 개의 정류소가 있으며, 각 정류소에는 두 대의 가용차량이 존재하는 상황을 가정하였다. 각 차량의 경로는 색상(blue, green, red, cyan)으로 구분되며, 1회전에 최대 2개 고객을 처리할 수 있는 용량을 가진다. 각 노드의 번호 뒤 소괄호에는 해당 노드에서 소요되는 서비스시간  $s_i$ 를, 대괄호에는 방문시간대인  $e_i$ 와  $l_i$ 를 기입하였다. 참고로 해당 예시에서는  $100 \times 100$  크기의 좌표평면에 고객노드 위치를 임의로 생성하였으며, 두 정류소의 위치는 (30, 50)과 (70, 50)으로 가정하였다. 노드 간의 이동 소요시간은 좌표평면상의 직선거리를 사용하며, 고객이 선택 가능한 시간대를 0과 600 사이에서 60 길이의 구간으로 임의 설정하였다. 고객노드에서 소요되는 서비스시간은 모두 15로 설정하였다. 보다 구체적인 실험 데이터에 대한 설명은 본문 4.1절에 제시된 내용을 참조함으로써 대체한다.

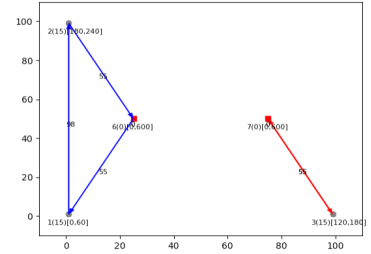
제약조건 (23)에 의하면, 차량에 일단 할당된 고객 노드는 추후 재할당되기 전까지 해당 차량에 할당된 것으로 강제된 상태로 스케줄이 누적된다. 만약 이전 스케줄에서 차량  $k$ 가 고객  $j$ 를 방문하는 것으로 결정되었다면( $x_{i,j,k} = 1$ ), 이후에도 차량  $k$ 는 고객  $j$ 를 방문해야만 한다. 다만, 차량  $k$ 가 고객  $j$ 로 이동하기 위한 직전 노드는 반드시 기존의  $i$ 일 필요는 없다. Figure 2(e)에서 녹색 차량의 경로는 [거점→고객(5)→거점]과 같이 표현될 수 있다. 이후 Figure 2(f)에서의 경로는 [거점→고객(6)→고객(5)→거점]으로 나타난다. 즉, 해당 차량은 기존에 고객(5)를 방문하기로 결정되었으므로 고객(5)를 반드시 방문해야 하지만, 고객(5)를 방문하기 직전 노드는 반드시 기존과 동일한 노드(거점)일 필요는 없다.



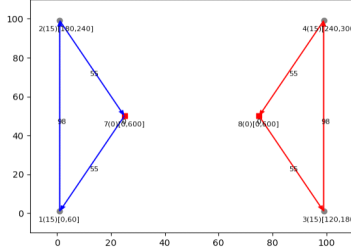
(a) node1 assigned to vehicle1



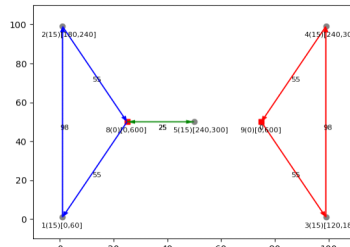
(b) node2 assigned to vehicle1



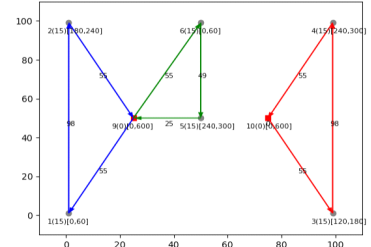
(c) node3 assigned to vehicle3



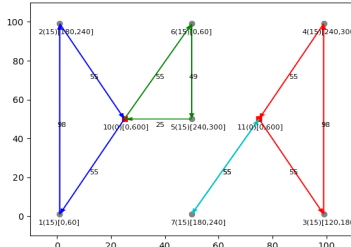
(d) node4 assigned to vehicle3



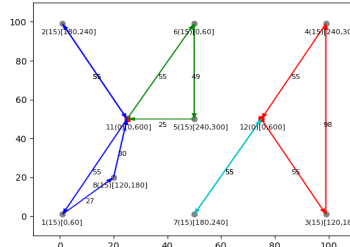
(e) node5 assigned to vehicle2



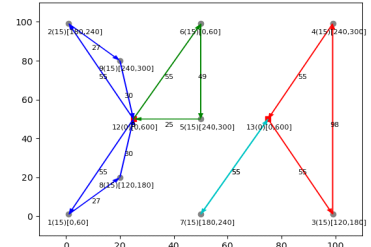
(f) node6 assigned to vehicle2



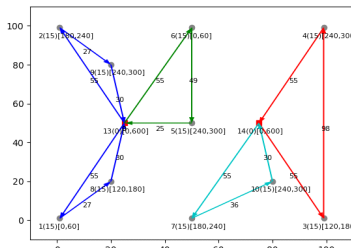
(g) node7 assigned to vehicle4



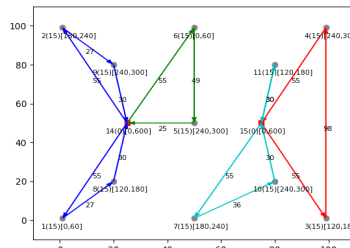
(h) node8 assigned to vehicle1



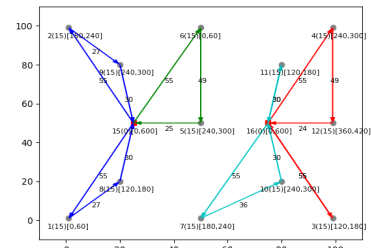
(i) node9 assigned to vehicle1



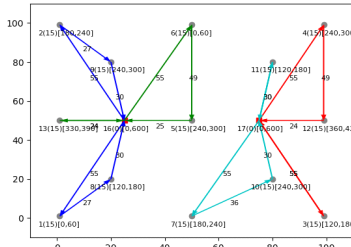
(j) node10 assigned to vehicle4



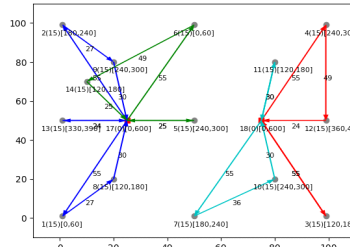
(k) node11 assigned to vehicle4



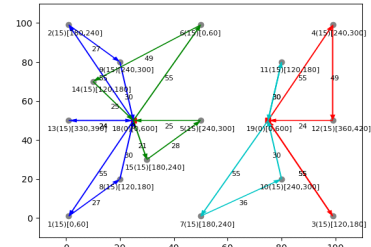
(l) node12 assigned to vehicle3



(m) node13 assigned to vehicle2



(n) node14 assigned to vehicle2



(o) node15 assigned to vehicle2

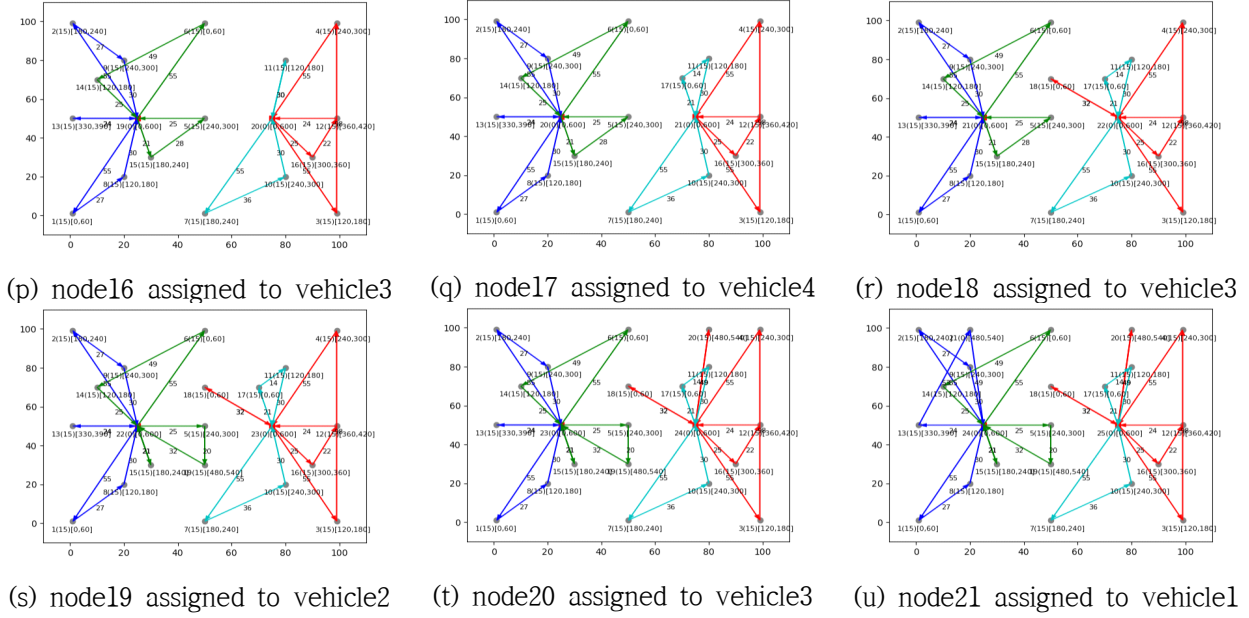


Figure 2. Example results of MD&T VRPOTW model's vehicle assignment with 2-depot, 2-vehicle per depot, and multi-trip assumptions (vehicles are distinguished by colors, service time required for each customer: 15, customer time window length: 60, full time window length: 600)

### 3.2. 현재시간 고려 제약조건 (constraints for current time consideration)

제약조건 (24)는 현재시간 고려 제약조건을 나타낸다. 현재의 고객주문의 요청에 대해, 시간상으로 현재의 시간을 의미하는  $T$ 보다 과거에 운행되는 것으로 결정된 스케줄은 변경할 수 없다. 즉, 기존에 확정된 노드의 도착시간은 현재시간 이후로만 변경할 수 있도록 제약해야 하며, 이는 제약조건 (24)와 같이 표현할 수 있다. 아래의 Figure 3에서는 현재시간  $T$ 의 값이 다를 때 서로 다른 결과가 도출될 수 있음을 예시하고 있다. Figure 3(a)는 현재 주문 고객(12) 이전까지 누적된 배송경로를 나타내고 있다. 현재 주문 고객(12)는  $T$  값이 100일 때, 차량(3)에 할당되는 것이 최적으로 나타나며,  $T$  값이 200일 때에는 차량(3)에 할당하기 위한 스케줄을 생성할 수 없어 차량(4)에 할당된다. Figure 3(a)의 기확정 스케줄에서  $x_{3,4,3} = 1$ 이었기 때문에 새로운 스케줄 누적 시  $x_{3,4,3}$ 의 값을 0으로 변경하여 고객(3) 이후에 연결될 노드를 변경하고자 한다면, 제약조건 (24)에 의해  $t_{3,3} \geq T$ 를 만족해야 한다. 이때, 차량(3)이 고객(3)에 도착하는 시간을 나타내는  $t_{3,3}$  값은 제약조건 (16)과 (17)에 의해 최소 120, 최대 180의 구간으로 제한된다. Figure 3(b)처럼 현재시간  $T$  값이 100인 경우  $x_{3,4,3}$  값을 0으로 수정하더라도  $t_{3,3} = 120 \geq T = 100$ 과 같이 조건을 만족할 수 있으나,  $T$  값이 200인 Figure 3(c)에서는  $x_{3,4,3}$  값을 0으로 수정할 수 없다.

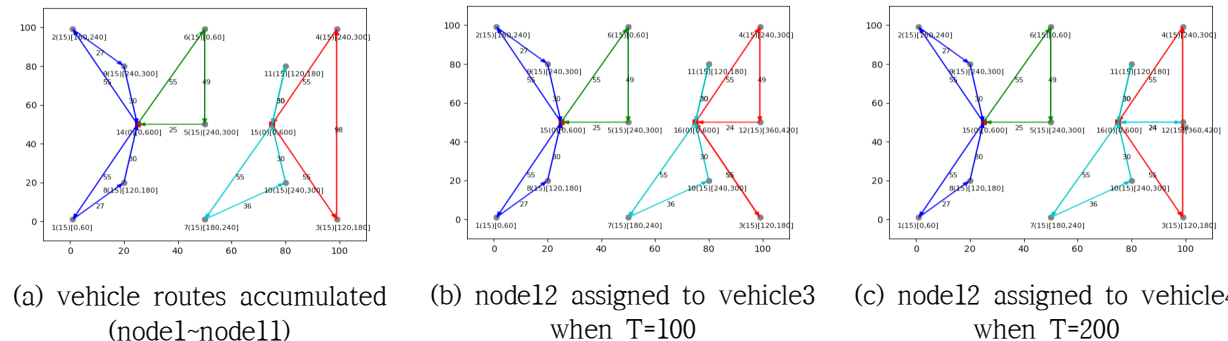


Figure 3. Example results of MD&T VRPOTW model's vehicle assignment with different present time

### 3.3. 오프라인 재할당

본 연구에서 제시된 MD&T VRPOTW 모델에서는 배송요청을 순차적으로 누적시킴에 따라 배송계획이 지역최적해를 나타낼 수 있기 때문에 주기적인 오프라인(비실시간) 재할당을 적용할 필요가 있다. 오프라인 재할당 결과는 제약조건 (23)을 제거한 MD&T VRPOTW 모델을 풀어냄으로써 제시할 수 있다. 아래의 Figure 4에서는 재할당을 통해 배송계획을 개선할 수 있음을 예시로서 보인다. Figure 4(a)는 재할당 없이 13개의 고객주문을 순차적으로 누적시킨 결과를 나타내며, Figure 4(b)는 재할당 이후의 결과를 나타낸다.

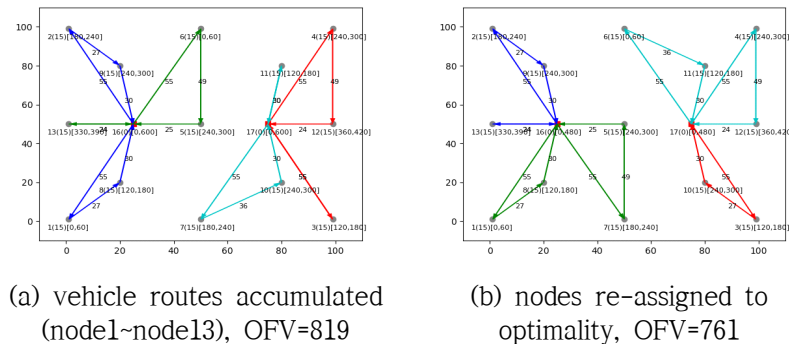


Figure 4. An example result of MD&T VRPOTW model's vehicle re-assignment

제약조건 (23)을 모델에서 제외하는 것은 문제의 복잡성을 증가시킨다. 이로 인해 해를 찾기 위해 소요되는 시간이 증가할 수 있으며, 제한시간 내에 풀어낼 수 있는 문제의 크기가 감소할 수 있다. 이어지는 실험에서는 재할당의 효과와 재할당 문제풀이의 한계 수준에 대한 분석을 포함하고 있다.

## 4. 실험

앞선 MD&T VRPOTW 수리모형 설명에 따르면, 해당 문제는  $(n-1)w$  개수만큼의 복제정류소 노드  $V_{dcopy}$ 를 포함하게 된다( $n$ 은 고객 노드의 수,  $w$ 는 다중정류소의 수). 예를 들어, 다중정류소가 2개 존재하고, 과거에 10개의 고객주문이 누적된 상태라면, 11번째 주문에 대한 수리모형에서는 20개의 복제정류소 노드를 생성하게 된다. 복제정류소는 차량의 다회전배차(multi-trip)를 모델링하기 위해 필요하지만, 차량경로문제에서 노드의 개수는 직접적으로 차원(dimension) 즉, 복잡성에 영향을 미치기 때문에 불필요하게 복제된 노드가 있다면 문제에서 제외시키는 것이 합리적이다. 본 연구에서는 MD&T VRPOTW 모델에서 복제정류소의 개수를 아래와 같이 축소할 수 있음을 제시한다.

직전 고객의 주문까지 도출된 차량의 경로  $S'$ 에서 차량이 소속된 정류소를 확인함으로써 각 정류소별로 할당된(assigned) 고객주문의 수로 정의된  $a_w$ , 그리고 각 정류소별로 할당 및 연결(assigned and connected)된 고객주문의 수로 정의된  $ac_w$ 를 계산할 수 있다. 아래의 Figure 5에 예시로 나타난 상황에서 정류소(1, 왼쪽)에 할당된 고객 노드의 개수가 5개( $a_1 = 5$ ), 정류소(2, 오른쪽)에 할당된 고객 노드의 개수가 3개( $a_2 = 3$ ), 정류소(1, 왼쪽)에 할당되었으면서 경로가 연속으로 연결된 고객 노드의 개수가 2개( $ac_1 = 2$ ), 정류소(2, 오른쪽)에 할당되었으면서 경로가 연속으로 연결된 고객 노드의 개수가 1개( $ac_2 = 1$ )로 계산된다.

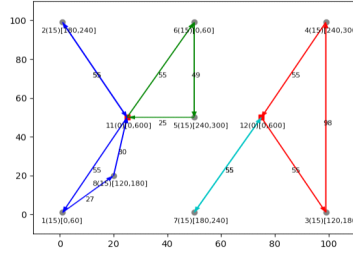


Figure 5. An example state for calculating  $a_w$  and  $ac_w$  values

Figure 5 상황 직후에 새로 들어온 고객주문에 대한 MD&T VRPOTW 모델에서 필요한 각 정류소  $m$ 에 대한 복제정류소의 개수  $|V_{dcopy_m}|$ 는 아래의 식과 같이 표현할 수 있다.

$$|V_{dcopy_m}| = (n-1) - \left( \sum_{w \in W, w \neq m} a_w \right) - ac_m \quad (25)$$

한 대의 차량이 모든 고객 노드를 서비스하고, 한 번의 회차에 하나의 고객만을 서비스하는 최악의 경우(worst case)를 가정해보자. 이때에는 차량이 최초 출발정류소 노드를 출발하여 마지막 도착정류소에 도달하기 전까지 총  $n$ 번의 회차를 완료하기 위해서  $(n-1)$ 개의 복제정류소를 필요로 한다. 하지만 본 연구에서 제시한 바와 같이 기확정 스케줄 제약조건을 통해 주문을 차량에 강제하게 된다면,  $(n-1)$ 개보다 적은 수의 복제정류소만으로도 문제를 해결할 수 있다. 예를 들어, 이미 타 정류소에 기확정된 고객주문의 개수만큼(자신을 제외한 타 정류소에 할당된 고객주문  $a_w$ 의 합) 복제정류소 개수를 줄일 수 있다. 추가적으로, 자신의 정류소에 기확정된 고객주문이라 하더라도 차량의 경로가 연속으로 연결된 경우 즉,  $ac_m$ 값만큼의 복제정류소를 생략할 수 있다.

본 연구에서 진행한 실험 중, 주문을 순차적으로 누적시키기 위한 실험(온라인 할당)에서는 수식 (25)에 표현된 방법을 통해 복제정류소의 개수를 감소시키는 전략을 이용하여 문제의 복잡성을 감소시켰으며, 이후 수리모델을 이용한 해를 도출하였다. 반면, 스케줄 재할당과 관련한 실험(오프라인 재할당)에서는 각 정류소별 복제정류소의 개수를  $(n-1)$ 개로 설정하였다. MD&T VRPOTW 모델의 솔루션 도출을 위해 CPLEX Optimization Studio를 활용하였으며, Intel(R) Core(TM) i5-11400 @2.60GHz 중앙처리장치(CPU)와 16GB의 메모리(RAM) 하드웨어가 활용되었다.

#### 4.1. 실험 데이터

실험을 위해 총 30개의 에피소드를 생성하였다. 각 에피소드는 순차적으로 발생하는 고객주문 30개를 포함한다. 각 고객주문은 균일분포  $U \sim (0, 100)$ 에 의한 위도와 경도를 가진다. 고객은 선택 가능한 10개의 시간대에 대해 임의정렬된 선호도 우선순위를 가진다. 선택 가능한 시간대는 0분에서 600분까지의 시간을 60분 단위로 분할한 총 10개의 구간을 말한다. 고객은 MD&T VRPOTW 모델을 통해 제시되는 선택가능시간대 중에서 자신의 우선순위에 따라 첫 번째로 가능(feasible)한 시간대를 선택한다. 고객이 선택한 시간대에 해당하는 스케줄을 기존 스케줄로서 업데이트한다. 공통적으로 모든 고객의 주문량은 1, 각 차량의 용량은 2로 설정하였다. 하나의 정류소에는 두 대의 차량이 소속되어 있으며, 총 두 개의 정류소가 존재한다. 노드 간의 직선거리를 차량의 운행에 소요되는 시간으로 가정하였으며, 현재시간  $T$ 는 0으로 가정하였다.

## 4.2. 온라인 할당 성능 분석

각 에피소드에 대해 고객주문과 그로 인한 차량스케줄을 누적하면서 MD&T VRPOTW 모델의 온라인 할당 성능에 대한 분석을 실시하였다. Figure 6에서는 요약된 분석결과를 확인할 수 있다. 제한된 시간 (60초) 내에 해결할 수 있었던 문제크기(고객 노드의 수  $n$ )는 평균 16.6, 최대 25, 최소 9개로 나타났다. 해결 가능했던 문제크기의 분산은 7.97로 계산되며, 해결 가능한 문제크기의 95% 신뢰구간이 11.07~22.13으로 나타난다. Figure 6(a)에서는 실제 실험에서 확인된 도수분포와 정규  $N \sim (16.6, 7.97)$ 로 그려낸 분포를 함께 확인할 수 있다. 실시간 응답성을 고려하여 시간제한을 60초 이내로 설정하여 실험을 진행하였으며, 문제 크기에 따른 평균 소요시간을 Figure 6(b)에서 확인할 수 있다. Figure 6(b)의 평균 소요시간은 각 문제크기에 대한 30회의 실험에 대한 평균값이다. 이러한 평균 소요시간은 한 개 시간대에 대한 해를 찾는 데 필요한 시간을 의미하며, 10개 시간대의 실현가능성(feasibility)을 모두 확인하고 차량스케줄을 생성하여 고객에게 제공하기 위해서는 약 10배의 시간이 소요된다.

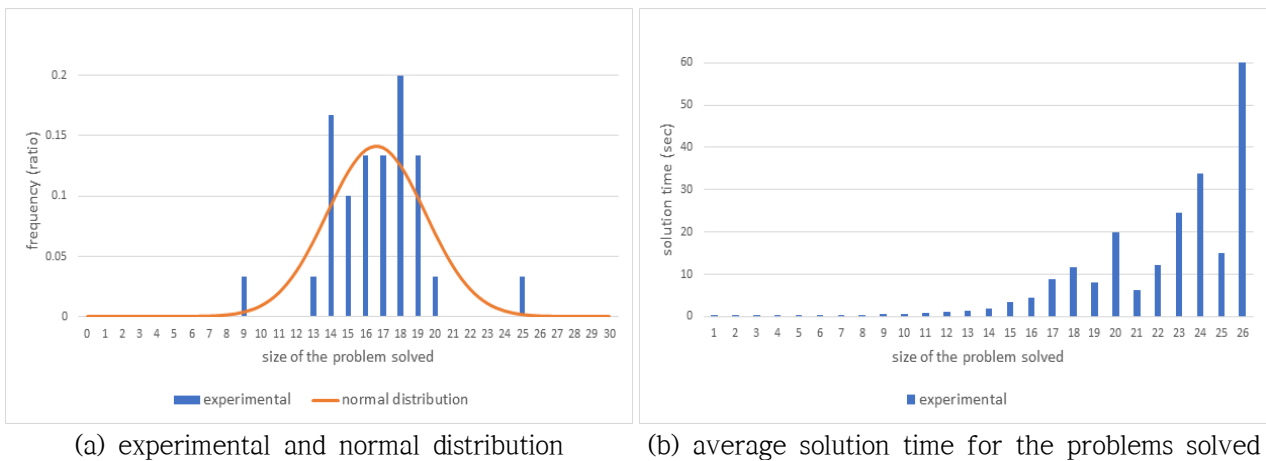


Figure 6. Experimental results of MD&T VRPOTW model's online vehicle assignment

## 4.3. 오프라인 재할당 성능 분석

오프라인 재할당에서는 제시된 MD&T VRPOTW 모델의 기확정 스케줄 제약조건 (23)을 제거해야 하며, 수식 (25)에 표현된 복제정류소를 감소시키는 전략을 사용하지 않는다. 앞서 생성한 동일한 에피소드를 사용하여 오프라인 재할당 실험을 진행하였으며, 분석결과는 Figure 7에 제시된 바와 같다. 재할당의 경우 제한시간이 600초로 설정되었다. 제한시간 내에 해결할 수 있었던 문제의 크기는 평균 11.5, 최대 14, 최소 6, 분산 3.36으로 확인되었으며, 95% 신뢰구간은 7.91~15.09로 계산되었다. Figure 7(a)에 실험을 통해 확인된 도수분포와 정규  $N \sim (11.5, 3.36)$ 가 함께 제시되었다. Figure 7(b)의 평균 소요시간은 각 문제크기에 대한 30회의 실험에 대한 평균값이다.

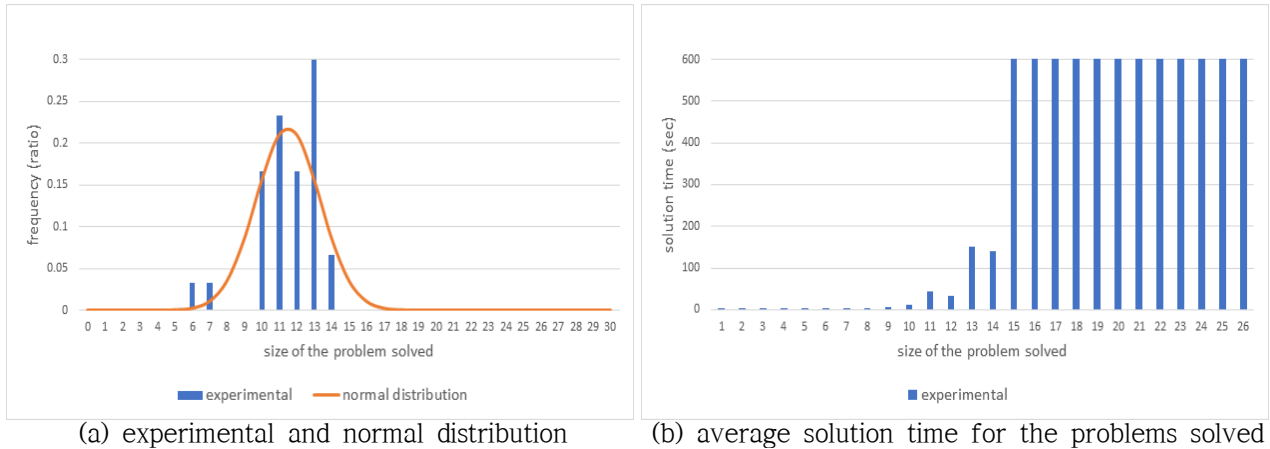


Figure 7. Experimental results of MD&T VRPOTW model's offline vehicle re-assignment

실험결과 분석을 통해 재할당의 효과를 확인할 수 있다. 실시간 할당 시 고객주문이 누적되는 과정에서, 기존 고객주문의 경우 앞서 할당되었던 차량으로 고정시키기 때문에 전체적인 관점에서의 최적상태가 오랜시간 유지되지 못한다. 재할당을 통해 스케줄을 최적의 상태로 개선할 수 있으며, 아래 Figure 8에서 그 효과를 확인할 수 있다. 재할당의 효과는 재할당으로 인한 차량의 운행거리(시간) 감소 비율로 나타낸다. 고객주문의 수가 많이 누적된 상태일수록 재할당 효과가 크며, 누적 고객주문 11개 수준부터는 재할당 효과가 평균 10%를 상회하는 것으로 확인된다. 이러한 결과는 본 실험설계의 가정사항 하에 도출된 것이며, 실제 적용되는 상황에서의 재할당 효과와 차이가 있을 수 있다.

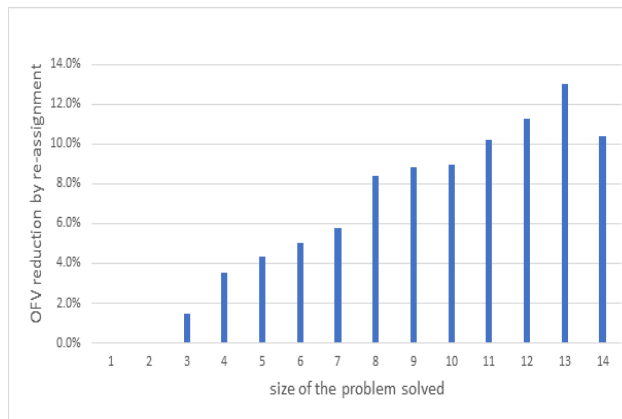


Figure 8. Experimental results on the effect of offline vehicle re-assignment

## 5. 논의

본 연구에서는 두 지리좌표간의 거리(시간)를 직선거리로 가정하였으며, 온전한( $n \times n$ ) 비용매트릭스가 사전에 준비된 것으로 가정하였다. 그러나 실제 운영단계에서 배송시간을 준수하기 위해서는 이와 같은 가정을 하기 어려운 경우가 많다. 비교적 정확한 지리좌표간 거리(시간) 정보를 확보하기 위해 경로엔진을 활용하는 방안이 있다. 구글지도(Google Maps, n.d.)는 상업용 경로엔진의 예시이며, 국내에도 T-map(SK Telecom, n.d.-b)와 네이버지도(Naver Cloud Plaform, n.d.)와 같은 상업용 경로엔진 공급업체가 존재한다. 상업용 경로엔진은 일부 무료 사용량이 제공되기도 하지만, 거리(시간) 요청 건당 약 5원에서 10원 수준의 사용료가 발생하기 때문에 온전한 비용매트릭스를 모두 채우기에는 비용적인 부담이 크다. 무료로 사용 가능한 오픈소스 경로엔진(GraphHopper, 2022, Openrouteservice, 2022, OSRM, 2021, Valhalla, 2021) 활용을 고려해볼 수도 있으나, 오픈소스 경로엔진이 오픈소스 지도(OpenStreetMap, 2015)에 의존하기 때문에 교통



체증과 같이 시간에 따른 변화를 반영하기 어렵다는 치명적인 단점이 있다. 또한, 새로운 도로교통법규나 새로운 도로네트워크에 대한 정보를 반영하는데 오랜시간이 걸린다는 단점이 있다. 결과적으로, 실제 운영 단계에서 MD&T VRPOTW 배송모델이 적용되기 위해서는 상업용 경로엔진을 활용하되, 그 사용량을 전략적으로 감소시키기 위한 방법을 강구해야 한다.

실험에서 CPLEX를 통해 해결 가능한 MD&T VRPOTW 문제의 크기는 온라인 할당의 경우 평균 16.6개, 오프라인 재할당의 경우 평균 11.5개 수준에 불과한 것으로 확인했다. 현실적인 크기의 문제에 대응할 수 있는 빠르고, 비교적 정확한(exact) 알고리즘에 대한 연구가 요구되며, 훨씬 큰 규모의 문제까지 해결할 수 있는 휴리스틱 알고리즘에 대한 연구가 진행될 필요가 있다. 이를 통해 부수적으로 재할당 효과에 대한 분석을 더욱 현실적이고 정확하게 수행할 수 있을 것으로 기대된다. 따라서, MD&T VRPOTW의 확대 적용을 위한 알고리즘 연구가 필요하며, 연구를 진행함에 있어 앞서 언급한 상업적 경로엔진 사용량의 감소 전략이 함께 고려되어야 한다.

## 6. 결론

본 연구에서는 실시간 배송요청에 의한 시간대예약배송 서비스 또는 주문형 배송서비스(즉시배송)를 MD&T VRPOTW(정시배송가능 시간대 제시를 위한 다중정류소 다회전 차량경로문제)로 일반화한다. 문헌 연구를 통해 MD&T VRPOTW 문제유형이 기존의 차량-여정 할당문제, 음식배달경로문제, 동적차량경로문제, 시간대제약 차량경로문제, 시간대할당 차량경로문제 등과의 연구 차별점이 존재하는 것을 확인했다. MD&T VRPOTW 문제를 혼합정수계획법의 수리모형으로 구성하였으며, 특히 기확정 스케줄을 고려하기 위한 제약조건과 현재시간을 고려한 실시간성 제약조건을 조망하였다. MD&T VRPOTW 모델에서는 고객의 주문이 순차적으로 누적되어감에 따라 문제의 복잡도가 증가한다. 본 연구에서는 다회전배차(multi-trip)를 모델링하기 위해 필수적이거나 문제의 복잡성 증가에 큰 영향을 미치는 복제정류소의 개수를 축소하기 위한 전략을 함께 제시하였다.

상용 프로그램인 CPLEX를 활용하여 MD&T VRPOTW 수리모형에 대한 실험을 진행하였다. 실험은 온라인 할당(고객주문의 순차적 누적)과 오프라인 재할당(누적된 고객주문 재최적화)으로 구분하여 설계하였다. 온라인 할당 실험은 실시간 응답성을 고려하여 시간제한을 60초 이내로 설정했으며, 오프라인 재할당의 경우 600초로 제한하였다. 본 연구에서 문제의 복잡성을 완화하기 위해 제시한 복제정류소 개수 감소전략과 기확정 스케줄 제약조건은 온라인 할당 과정에 적용할 수 있다. 완화 전략이 적용되지 않은 오프라인 재할당의 결과를 온라인 할당 결과와 비교함으로써 완화 효과를 확인하였다. 오프라인 재할당의 경우 상용 프로그램을 통해 처리 가능한 고객주문의 수가 평균 11.5, 최대 14, 최소 6개로 나타났으며, 온라인 할당의 경우 평균 16.6, 최대 25, 최소 9개로 확인되었다. 실험에서는 온라인 할당 누적 고객주문의 수가 약 10개 이상일 때 오프라인 재할당을 적용하면 약 10% 이상의 차량운행 거리(시간) 감소효과가 나타나는 것으로 확인되었다.

MD&T VRPOTW 문제는 복잡성이 높아 상용 프로그램으로 현실에서 요구되는 크기의 문제를 해결하기에는 한계가 있을 것으로 판단된다. 이에 따라, 정확한(exact) 알고리즘 및 휴리스틱 알고리즘의 설계를 향후 연구 주제로 제시하고자 한다. 실용화를 위한 향후 연구에서는 현실적인 정시배송을 위해 경로엔진의 활용이 고려되어야 한다. 이때, 경로엔진 사용료 및 알고리즘 응답시간과 관련한 적절한 균형점을 제시할 필요가 있다.

배송서비스 차별화를 위한 노력이 새로운 서비스를 만들어내고 있으며, 새로운 서비스에는 많은 개선 가능성이 존재한다. 본 연구와 추가적인 향후 연구는 즉시배송 및 시간예약배송 서비스의 품질을 개선에 기여한다. 고객에게 폭넓은 시간선택 자유도를 제공함으로써 고객 만족도를 높일 수 있을 것으로 기대되며, 기존의 전화응대를 통한 배송(방문)계획 수립을 고객의 직접선택으로 대체함으로써 자원관리 효율성을 제고할 수 있을 것으로 기대된다. 본 연구의 수행범위에서는 이러한 효과를 수치적으로 제시하기 어려웠으나, 향후 구체적인 사례연구를 통해 확인하고자 한다.



## 참고문헌

- AbdAllah, A. M. F. M., Essam, D. L., & Sarker, R. A. (2017). On solving periodic re-optimization dynamic vehicle routing problems. *Applied Soft Computing Journal*, 55, 1-12. <https://doi.org/10.1016/j.asoc.2017.01.047>
- Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences of the United States of America*, 114(3), 462-467. <https://doi.org/10.1073/pnas.1611675114>
- Baldacci, R., Mingozzi, A., & Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1), 1-6. <https://doi.org/10.1016/j.ejor.2011.07.037>
- Bhatti, A., Akram, H., Basit, H. M., Khan, A. U., Naqvi, S. M. R., & Bilal, M. (2020). Ecommerce Trends During Covid-19. *International Journal of Future Generation Communication and Networking*, June, 1-25. <https://www.researchgate.net/publication/342736799%0Ahttps://www.contactpigeon.com/cp/resources/ebooks/ecommerce-trends-covid19.pdf>
- Boysen, N., Fedtke, S., & Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43(1), 1-58. <https://doi.org/10.1007/s00291-020-00607-8>
- Dablanc, L., Morganti, E., Arvidsson, N., Woxenius, J., Browne, M., & Saidi, N. (2017). The rise of on-demand 'Instant Deliveries' in European cities. *Supply Chain Forum*, 18(4), 203-217. <https://doi.org/10.1080/16258312.2017.1375375>
- Dalmeijer, K., & Spliet, R. (2018). A branch-and-cut algorithm for the Time Window Assignment Vehicle Routing Problem. *Computers and Operations Research*, 89, 140-152. <https://doi.org/10.1016/j.cor.2017.08.015>
- Desrochers, M., & Laporte, G. (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1), 27-36. [https://doi.org/10.1016/0167-6377\(91\)90083-2](https://doi.org/10.1016/0167-6377(91)90083-2)
- Dixit, A., Mishra, A., & Shukla, A. (2019). Vehicle routing problem with time windows using meta-heuristic algorithms: A survey. *Advances in Intelligent Systems and Computing* (Vol. 741). Springer Singapore. [https://doi.org/10.1007/978-981-13-0761-4\\_52](https://doi.org/10.1007/978-981-13-0761-4_52)
- El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, 22(3), 123-131. <https://doi.org/10.1016/j.jksus.2010.03.002>
- Google Maps. (n.d.). Distance Matrix API. Retrieved June 9, 2022, from <https://developers.google.com/maps/documentation/distance-matrix/overview>
- GraphHopper. (2022). Graphopper Routing Engine. GraphHopper. <https://github.com/graphhopper>
- Jahanshahi, H., Bozanta, A., Cevik, M., Kavuk, E. M., Tosun, A., Sonuc, S. B., Kosucu, B., & Başar, A. (2022). A deep reinforcement learning approach for the meal delivery problem. *Knowledge-Based Systems*, 243, 108489. <https://doi.org/10.1016/j.knosys.2022.108489>
- Korea Logistics News. (2022). GS Shop, expected arrival date service "Delivery on customer's preferred date." <https://www.klnews.co.kr/news/articleView.html?idxno=304220>

- Korea Maritime Institute. (2022). Korea Maritime Institute International Logistics Weekly (Vol. 625).
- Kucharska, E. (2019). Dynamic Vehicle Routing Problem—Predictive and Unexpected Customer Availability. *Symmetry*. <https://doi.org/10.3390/sym11040546>
- Kyunghyang. (2022). “Eliminate delivery costs and reduce the minimum order amount” Competition for immediate delivery initiative in full swing. <https://m.khan.co.kr/economy/economy-general/article/202208021556011#c2b>
- Li, J., Yang, S., Pan, W., Xu, Z., & Wei, B. (2022). Meal delivery routing optimization with order allocation strategy based on transfer stations for instant logistics services. *IET Intelligent Transport Systems*, April, 1108–1126. <https://doi.org/10.1049/itr2.12206>
- Li, S., Li, M., & Lee, V. C. S. (2020). Trip-Vehicle Assignment Algorithms for Ride-Sharing. *Combinatorial Optimization and Applications*, LNCS 12577, 685–700. [https://doi.org/10.1007/978-3-030-64843-5\\_9](https://doi.org/10.1007/978-3-030-64843-5_9)
- Maeil Business. (2021). Homeplus Express with “Immediate Delivery in 1 Hour.” <https://www.mk.co.kr/news/business/view/2021/08/749641/>
- Morgan Stanley. (2022). Global Ecommerce Growth Forecast 2022. <https://www.morganstanley.com/ideas/global-ecommerce-growth-forecast-2022>
- Mori, J. C. M., & Samaranayake, S. (2021). On the Request-Trip-Vehicle Assignment Problem. *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21)*, 693, 228–239. <https://doi.org/10.1137/1.9781611976830.21>
- Naver Cloud Plaform. (n.d.). Maps Application Services - NAVER Cloud Platform. Retrieved June 11, 2020, from <https://www.fin-ncloud.com/product/applicationService/maps>
- Neves-Moreira, F., Pereira da Silva, D., Guimarães, L., Amorim, P., & Almada-Lobo, B. (2018). The time window assignment vehicle routing problem with product dependent deliveries. *Transportation Research Part E: Logistics and Transportation Review*, 116(March), 163–183. <https://doi.org/10.1016/j.tre.2018.03.004>
- Ojeda Rios, B. H., Xavier, E. C., Miyazawa, F. K., Amorim, P., Curcio, E., & Santos, M. J. (2021). Recent dynamic vehicle routing problems: A survey. *Computers and Industrial Engineering*, 160(July). <https://doi.org/10.1016/j.cie.2021.107604>
- Openrouteservice. (2022). Openrouteservice: The open source route planner api. <https://github.com/GIScience/openrouteservice>
- OpenStreetMap. (2015). Setting up a local copy of the OpenStreetMap database. OpenStreetMap. [https://wiki.openstreetmap.org/wiki/Setting\\_up\\_a\\_local\\_copy\\_of\\_the\\_OpenStreetMap\\_database,\\_kept\\_up\\_to\\_date\\_with\\_minutely\\_diffs](https://wiki.openstreetmap.org/wiki/Setting_up_a_local_copy_of_the_OpenStreetMap_database,_kept_up_to_date_with_minutely_diffs)
- OSRM. (2021). Open Source Routing Machine: High performance routing engine run on OpenStreetMap data. <https://github.com/Project-OSRM/osrm-backend>
- Park, H., So-Hyoung, K., Jeong, S.-J., & Sangbeom, S. (2021). The Changes in the Parcel delivery service according to the Spread of COVID-19 in Korea. *Journal of Transport Research*, 28(2), 51–66. <https://doi.org/10.34143/jtr.2021.28.2.51>
- Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), 1–11. <https://doi.org/10.1016/j.ejor.2012.08.015>

- Reyes, D., Erera, A. L., Savelsbergh, M. W. P., Sahasrabudhe, S., & O'Neil, R. J. (2018). The Meal Delivery Routing Problem. *Optimization Online*, 1-70.  
[http://www.optimization-online.org/DB\\_FILE/2018/04/6571.pdf](http://www.optimization-online.org/DB_FILE/2018/04/6571.pdf)
- SK Telecom. (n.d.-a). T Direct Shop, arrive at your desired time and place. Retrieved August 31, 2022, from <https://www.tworld.co.kr/poc/html/main/MA.html>
- SK Telecom. (n.d.-b). TMAP API. Retrieved June 11, 2020, from <https://openapi.sk.com/API/detail?svcSeq=4#pay>
- Spliet, R., & Desaulniers, G. (2015). The discrete time window assignment vehicle routing problem. *European Journal of Operational Research*, 244(2), 379-391. <https://doi.org/10.1016/j.ejor.2015.01.020>
- Spliet, R., & Gabor, A. F. (2015). The Time Window Assignment Vehicle Routing Problem. *Transportation Science*, 49(4), 721-731. <https://doi.org/http://dx.doi.org/10.1287/trsc.2013.0510>
- Uber. (2020). Ride with Uber - Request Rides 24/7. Uber. <https://www.uber.com/gb/en/ride/>
- Ulmer, M. W., Thomas, B. W., Campbell, A. M., & Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, 55(1), 75-100. <https://doi.org/10.1287/TRSC.2020.1000>
- Valhalla. (2021). Valhalla: Open Source Routing Engine for OpenStreetMap.  
<https://github.com/valhalla/valhalla>
- Wang, X., Wang, L., Wang, S., Chen, J. fang, & Wu, C. (2021). An XGBoost-enhanced fast constructive algorithm for food delivery route planning problem. *Computers and Industrial Engineering*, 152(December 2020), 107029. <https://doi.org/10.1016/j.cie.2020.107029>
- Zhao, K., Chen, W., & Lye, K. W. (2019). Online vehicle dispatch: From assignment to scheduling. *IEEE International Conference on Data Mining Workshops, ICDMW*, 2018-Novem, 608-611.  
<https://doi.org/10.1109/ICDMW.2018.00094>
- Zhou, Z., & Roncoli, C. (2022). A scalable vehicle assignment and routing strategy for real-time on-demand ridesharing considering endogenous congestion. *Transportation Research Part C: Emerging Technologies*, 139(March). <https://doi.org/10.1016/j.trc.2022.103658>

# UAM과 드론을 사용한 도시물류상의 선택적 이중구조 오리엔티어링 문제

김동균, 차형주, 정태수<sup>†</sup>

고려대학교 산업경영공학과

## Selective Two-echelon Orienteering Problem in urban logistics using UAM and Drones

Dong Kyun Kim, Hyungjoo Cha, Taesu Cheong<sup>†</sup>

School of Industrial Management Engineering, Korea University

In recent days, the Urban Air Mobility (UAM) is emerging as the new transportation after drone. In this study, UAM and drones are used for an on-demand logistics service. Since UAM is considered for delivery, the consideration of vertiports and vertistops (infrastructure for UAM) is inevitable. In this problem, the UAM only visits vertiports and vertistops for the delivery. After the UAM's delivery, at vertiports drone deliver the package to the customer. At vertistops, customers visit vertistop to pick up the package. We present a mathematical model for routing problem using UAM and drones. Since this problem is a general case of Traveling Salesman Problem (TSP), it is hard to get an optimal solution if the size of problem gets bigger. Thus we develop a two-phase Variable Neighborhood Search based heuristic algorithm for large-size problems.

**Keywords:** Urban Air Mobility, Orienteering Problem, MILP, Variable Neighborhood Search, Electric drone

---

논문접수일 : 2022.10.08.

심사완료일 : 2022.12.16.

게재확정일 : 2022.12.19.

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(No.2021R1F1A1059302)이며, 또한 정부(산업통상자원부)의 재원으로 한국에너지기술평가원 지원을 받아 수행된 연구 (No.20229400000090)임.

<sup>†</sup> Corresponding Author: tcheong@korea.ac.kr

## 1. 서론

수도권의 인구 집중으로 인한 차량의 증가는 오랫동안 도심의 심각한 교통체증의 원인이 되어 왔다. 이러한 도시집중화로 인한 교통 혼잡으로 많은 사회간접비용이 발생하고 있다. 한국교통연구원의 연구에 따르면 교통혼잡비용은 2010년 28.5조 원에서 2015년 33.3조 원까지 지속적으로 증가하였으며 2016년, 2017년 교통혼잡비용은 각각 36.3조 원, 38.7조 원으로 꾸준히 증가하는 모습을 보였다 (Cheon et al., 2020). 교통 혼잡으로 인한 문제는 COVID-19로 인한 비대면 배송 수요의 증가와 증가한 1인 가구로 인한 배송 건수 증가와 맞물려 배송업계에도 큰 문제가 되고 있다. 하지만 기술의 발달로 항공 운송수단 운용 가능성이 높아지며 이러한 문제를 해결할 수 있는 Urban Air Mobility (UAM)가 떠오르고 있다. UAM은 운송수단이라기 보다는 하나의 시스템을 의미한다. 현재의 도심에 수직이착륙이 가능한 비행체 (electric Vertical Take Off and Landing, eVTOL)가 새로운 이동수단으로 추가되어 기존의 운송수단과 협력하며, 원활한 이동을 가능하게 하는 시스템이다. 또한, 도심에서 수직이착륙이 가능한 개인용 비행체 (Personal Air Vehicle, PAV)로 이동하는 공중 교통 체계를 의미하기도 한다. UAM에 주로 사용되는 공중 기반의 이동수단인 eVTOL은 보통 사람이 탈 수 있을 정도로 큰 드론의 형태이다. 기존 수직이착륙이 가능한 공중 이동수단인 헬리콥터를 이용할 수도 있겠지만, 헬리콥터는 소음과 관련된 문제가 많아 고려되지 않고 있다 (Jeong et al., 2021).

UAM이 적용되기 위해서는 실제로 사용될 eVTOL과 기반 시설이 필수적으로 고려되어야 한다. 드론의 기술적인 발전에 힘입어 사람이 탑승할 수 있는 큰 드론이 UAM에 활용될 eVTOL로 고려되고 있다. 해외의 경우 대표적으로 UAM의 도입과 이를 위한 eVTOL의 개발에 앞장서고 있는 기업은 미국의 Uber 사이다. 국내에서는 현대모터그룹에서 미국의 Uber 사와 협업하여 2028년 UAM 상용화를 목표로 하고 있다. 최근에는 ‘CES2020’에서 UAM 사업의 첫 비전으로 콘셉트 모델인 ‘S-A1’을 제시하는 등 UAM의 도입을 준비하며 이를 보다 구체화하고 있다 (Hyundai, 2022). 아래 Figure 1은 현대모터그룹에서 제시한 콘셉트 모델 ‘S-A1’의 사진이다.

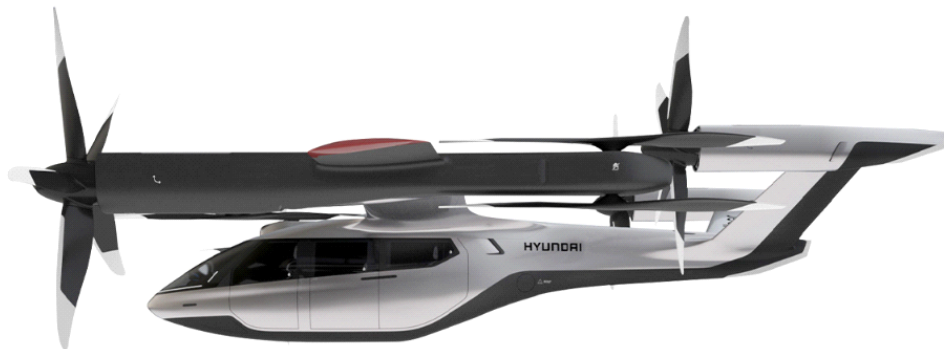


Figure 1. Hyundai Motor Group's UAM vision concept, 'S-A1'

다음으로 UAM의 도입을 위해서는 기반 시설이 필요하다. Fadhil(2018)에서는 이러한 기반 시설을 vertiport (혹은 skyport)와 vertistop으로 구분하였다. Vertiport와 Vertistop은 두 가지 모두 UAM의 수직 이착륙을 위한 station의 개념이라는 점은 동일하다. 둘의 가장 큰 차이점은 vertistop의 경우 UAM의 이착륙과 승객의 승차하만 가능하지만, vertiport는 UAM의 이착륙과 승객의 승차하와 더불어, UAM의 충전, 다른 모빌리티로의 환승 등이 가능한 multi-hub의 역할을 할 수 있다는 점이 가장 큰 차이점이다. Vertiport와 Vertistop 이 둘은 그 역할 및 수행 가능한 기능에도 차이가 있지만, 크기와 규모에도 차이가 있으므로 이를 구분하여 고려하는 것이 중요하다.

UAM의 운영을 위해 기반 시설이 필수적이기 때문에 UAM 단독으로 물품의 배송을 진행하기에는 어려움이 따른다. 따라서 본 연구에서는 UAM과 함께 드론을 사용하여 UAM과 드론의 협업 배송모형을 제안한다. UAM은 배송이 가능한 여러 개의 배송지 (delivery center)를 모두 방문하지 않고 이를 선택적으로 방문하여 중 선택된 임의의 배송지를 방문하며 배송을 진행하고 출발지로 돌아온다. 이때 배송지는 UAM의 특성을 고려하여 vertiport와 vertistop으로 나누어진다. 각 vertiport와 vertistop으로 물품의 배송이 완료되면 배송지의 종류에 따라 서로 다른 방식으로 고객에게 배송을 시작한다. Vertiport의 경우 multi-hub의 역할을 할 수 있어 다른 모빌리티의 운용이 용이하기에 드론을 활용하여 배정된 고객에게 배송을 진행한다. 반면, vertistop의 경우 UAM의 이착륙과 승객의 승하차 정도만 가능한 시설이기 때문에 다른 모빌리티의 운용, 접근 등이 어렵다. 이에 다른 모빌리티를 활용한 배송이 적합하지 않다. 대신 해당 vertistop에 존재하는 보관함에 물품을 보관하고, 해당 vertistop에 배정된 고객이 직접 방문하여 물품을 가져가는 형태로 배송이 진행된다. 본 논문에는 이러한 형태의 배송 모델인 Two-echelon Orienteering Problem with UAM을 제안한다.

제안 모델은 UAM의 이동시간과 실제로 고객이 물품을 받는 시간 모두를 최소화하는 것을 목표로 한다. 각 고객은 배송을 받기를 원하는 시간이 있다. 만약 이 시간에 맞춰서 배송하지 못한다면 고객이 물품을 받은 시간과 원하는 배송시간의 차이만큼의 페널티가 주어지게 된다. 결국 제안 모델에서는 UAM의 이동시간, 실제로 고객이 물품을 받는 시간과 고객이 원하는 시간의 차이, 고객에게 물품 배송이 완료되는 시간 모두를 최소화하는 것을 목적으로 하게 된다.

본 논문에서는 제안 모델에 대한 혼합정수계획법 기반의 수리최적화모형을 제안하고 이와 더불어 Variable Neighborhood Search (VNS) 기반의 2단계의 메타-휴리스틱 알고리즘을 제안한다. 기존 연구들의 대부분은 지상 전기차와 전기 드론의 협업 배송의 형태이다. 차량의 경우 도로를 따라 움직이기 때문에 이동 경로에 대한 제약이 있고 신호대기 시간이나 교통 체증에 빈번하게 노출된다. 반면 UAM은 이동경로가 비교적 자유로우며 신호대기 시간이나 교통 체증의 영향을 받지 않는다. 또한, 일반적인 차량보다 훨씬 빠른 속도로 이동이 가능하기 때문에 본 연구에서 사용되는 UAM이 보다 빠르고 효율적인 배송을 진행할 수 있을 것으로 기대된다.

본 연구의 주요 기여점은 다음과 같다.

첫째, UAM과 전기 드론의 협업 배송을 진행하는 모델인 제안 모델을 처음 제시한다. 제안 모델은 UAM이 방문하는 배송지로 UAM 운용을 위한 기반 시설인 vertiport와 vertistop을 다르게 고려하며, 배송지의 종류에 따라 고객까지의 배송이 다르게 이루어진다.

둘째, 제안 모델에서는 배송을 진행할 때 UAM의 이동 시간과, 각 고객이 배송 받기를 원하는 시간과 실제 배송 받은 시간의 차이, 배송지 사용에 따른 비용을 최소화하는 경로를 찾는다. 이를 통해 배송 시간과 더불어 고객이 느끼는 서비스의 만족도까지 고려하는 모델을 제안한다.

셋째, 제안 모델을 풀기 위한 혼합정수계획법 기반의 수리최적화모형과 동시에 VNS 기반의 2 단계 휴리스틱 알고리즘을 제안한다.

본 논문은 다음과 같은 순서로 구성되어 있다. 2장에서는 드론이 사용된 협업 배송과 UAM에 대한 연구의 흐름을 알아본다. 3장에서는 제안 모델에 대한 정의와 이 문제를 해결하기 위한 혼합정수계획법 기반의 수리최적화모형, VNS 기반의 2단계 휴리스틱 알고리즘을 알아본다. 4장에서는 실험을 통해 수리최적화모형과 휴리스틱 알고리즘의 성능을 평가한다. 마지막으로 5장에서는 본 논문의 결론을 제시한다.

## 2. 관련 연구

UAM의 개념이 처음 등장한 이후, 관련 연구는 꾸준히 진행되고 있다. 국내에서는 Choi et al.(2021)이 UAM의 국내 도입을 위해 국제표준을 고려한 vertiport의 설치기준을 마련하였다. 또한, Jeong et al.(2021)은 K-Means Algorithm을 활용하여 vertiport의 위치를 선정하고, Aviation Environmental Design Tool (AEDT)을 사용해 소음을 분석하여 COEX부터 김포공항까지의 최저소음 경로를 찾는 연구를 진행하였다. 해외에서도 다양한 연구가 진행되었는데 Wu and Zhang(2021)은 UAM과 다양한 이동수단의 연계를 고려한 수리최적화모형을 제안하였다. 이동수단별로 각기 다른 이동시간과 비용을 고려하였으며, vertiport에 진입 가능한지 여부도 구분하였다. Bennaceur et al.(2022)은 승객을 중심으로 접근하였다. UAM을 운영할 때, 승객들의 대기 시간을 최소화할 수 있는 방법을 수리최적화모형과 함께 제안하였으며, 경로생성과 재충전 방안을 동시에 제안해 운영비 감소와 배터리 수명 증가의 효과도 얻었다. Shin et al.(2022)의 연구는 차량과 UAM의 연계와 UAM의 공중 충돌위험을 고려한 수리최적화모형과 유전 알고리즘 (Genetic Algorithm) 기반의 휴리스틱을 함께 제시하였다. 대부분의 UAM을 활용한 경로 문제는 고객들의 운송을 가정하기 때문에 UAM과 함께 활용되는 이동수단은 대부분 차량이나 대중교통이었다. 하지만 본 연구에서는 UAM을 활용한 화물의 배송을 고려하며 이에 따라 기존 연구에서는 함께 활용되지 않았던 드론을 사용한다.

드론을 협업 배송에 고려하는 대표적인 연구는 Murray and Chu(2015)의 단일 차량과 드론의 협업배송 문제이다. 해당 연구에서는 flying sidekick traveling salesman problem (FSTSP)과 parallel drone scheduling traveling salesman problem (PDSTSP)을 제안한다. FSTSP는 단일 차량을 활용하는 경로 문제인 Traveling Salesman Problem (TSP)에 드론을 보조 배송 수단으로 활용한다. 차량은 드론을 탑재하고 배송을 진행하여 특정 노드에서 드론이 차량을 떠나 배송을 수행한 뒤 차량을 떠난 노드가 아닌 다른 노드에서 차량과 합류하는 형태의 배송 방식을 취한다. PDSTSP에서는 차량은 드론이 배송하지 않는 노드들에 대해 배송을 진행하고 드론은 출발지에서 일정 거리 이내의 노드에 대해 배송을 진행한다. Kim and Moon(2018)의 연구는 이러한 PDSTSP의 일반적인 형태인 traveling salesman problem with a drone station (TSP-DS)을 제안한다. 해당 연구에서는 드론이 출발지에서 배송을 진행하는 것이 아니라 특정 노드에 설치한 드론 스테이션에서 배송을 진행한다. 차량이 드론이 배송하지 않는 노드들의 배송을 진행하며 드론 스테이션을 방문하면 드론 스테이션에서 드론이 배송을 시작한다. 차량은 드론 스테이션 방문 후 대기하지 않고 이후 다른 노드들의 배송을 진행하며 출발지로 돌아오는 형태이다. 본 연구는 PDSTSP나 TSP-DS에서 차량의 역할을 UAM이 맡는 형식으로 볼 수 있지만, 해당 연구에서는 그 동안 고려하지 않았던 중간 노드를 고려한다는 점에서 차이점이 존재한다.

지금까지 살펴본 연구들은 모두 TSP의 활용 형태로 주어진 모든 노드를 방문하는 경로를 생성하며, 협업 배송을 진행하는 경우에는 서로 다른 이동수단이 하나의 노드에서 동시에 만나서 이동하는 것을 고려한다. 하지만 본 연구는 주어진 모든 노드를 방문할 필요가 없다. 주어진 노드를 모두 방문하지 않아도 되는 경로를 생성하는 문제는 Location Routing Problem (LRP) 혹은 Orienteering Problem (OP)으로 볼 수 있다. 따라서 본 연구는 LRP, OP로 확장이 가능하다. LRP는 facility location problem (FLR)을 푸는 것을 주된 목적으로 하며 이를 위해 동시에 TSP 혹은 vehicle routing problem (VRP)을 풀게 된다 (Nagy and Salhi., 2007). FLR에서 각 시설에 할당된 고객들이 서비스를 받을 때 시설에서 경로를 이루며 고객에게 서비스를 제공한다면 LRP, 한 명의 고객에게 서비스를 제공하고 다시 시설로 돌아간 뒤 다음 고객에게 서비스를 제공하게 된다면 이는 location-allocation problem이 된다 (Hassanzadeh et al., 2009). 반면 OP의 경우 주어진 시간 혹은 거리에 맞는 경로를 형성하기 위해 노드를 선택하여 방문하게 되며 중간 노드로 고려되는 시설, 혹은 서비스의 제공 방식에 대한 제약은 존재하지 않는다. 본 연구에서는 고객이 경로를 이루어 서비스를 받지 않고 드론을 통해 왕복으로 서비스를 받거나 직접 vertistop을 방문하게 된다. 따라서 다양한 방식으로 서비스의 제공이 가능한, 보다 일반적인 경우를 가정하기 위해 OP 방식을 고려하여 문제를 해결하려고 한다.

본 연구는 Hong et al.(2019)의 연구를 확장 및 발전시킨 연구이다. 해당 연구는 차량을 활용한 on-demand 물류 서비스를 LRP로 풀어내었다. 본 연구에서는 해당 문제를 OP 방식으로 접근한다. OP는 Golden et al.(1987)에 의해 처음 소개되었으며, 앞서 설명하였던 것과 같이 주어진 모든 노드를 방문하는 경로가 아니라 주어진 노드의 일부만을 선택적으로 방문하여 제한 시간 혹은 거리를 넘기지 않는 경로를 찾는 것이 목적이다. 이러한 OP는 방문할 노드를 선택하는 문제인 Knapsack Problem과 선택된 노드들을 방문하는 경로를 생성하는 TSP로 나누어질 수 있다 (Gunawan et al., 2016). OP는 기존 하나의 경로를 찾는 문제에서 여러 개의 경로를 찾는 문제로 변경하여 Team Orienteering Problem (TOP)으로 확장 될 수 있다. Labadie et al.(2012)은 노드별로 특정 시간대에 서비스가 시작되게 하는 제약을 추가하여 OP를 OP with Time Windows (OPTW)로 확장하였다. Verbeeck et al.(2014)은 두 노드 사이의 이동시간이 첫 번째 노드의 출발 시간에 의존적인 문제를 Time Dependent OP (TDOP)로 정의하였다. 이후 TDOP는 노드간 이동시간의 불확실성을 고려한 Ilhan et al.(2008)의 OP with Stochastic Profits (OPSP)로 확장되었다. 노드간 이동시간이 아니라 노드에 부여되는 이익 혹은 점수에 차별점을 부여한 Generalized OP (GOP)도 제안되었다(Geem et al., 2005). 해당 연구에서는 각 노드에 하나의 점수가 부여되는 것이 아니라 특성에 따라 점수의 집합이 부여되고 선택된 노드의 점수 집합 전체의 합이 목적함수가 되는 차이점이 있다. 노드가 아니라 아크를 방문하는 형태로 OP를 고려한 Verbeeck et al.(2014)의 Arc OP (AOP), TOP의 발전 형태로 각 노드에 수요와 점수 값이 함께 부여되는 Archetti et al.(2009)의 Capacitated TOP (CTOP) 등 활발한 연구가 진행되고 있다.

노드간의 이동시간이나 노드에 부여되는 속성, 점수의 변화에 대한 발전이 대부분이었던 기존 연구와는 다르게 본 연구에서는 중간 노드의 추가를 고려한다. 또한, 앞서 언급하였던 Hong et al.(2019)의 연구와는 다르게 배송지의 종류가 vertiport와 vertistop으로 나누어지며, 배송지의 종류에 따라 고객까지의 배송을 다르게 진행한다.

### 3. 문제 정의

본 장에서는 제안 모델에 대한 정의와 혼합정수계획법 기반의 수리최적화모형을 제시하고, 이후 VNS 기반의 2단계 휴리스틱 알고리즘을 제시한다. 제안 모델은 UAM과 드론의 협업 배송모형이다. UAM은 배송 시작지점을 출발하여 물품을 보관할 수 있는 배송지에 방문하여 물품을 배송하고, 다시 배송 시작지점으로 돌아온다. 이때 UAM은 방문 가능한 배송지를 모두 방문할 필요가 없으며, 배송지를 선택적으로 방문하여 보다 효율적인 경로를 찾는다. 배송지는 UAM의 특성을 고려하여 vertiport와 vertistop으로 나누어지며, 물품을 저장할 수 있는 용량과 UAM의 배송 이후 고객에게 직접 물품을 배송하는 방식에 차이가 있다. UAM으로부터 물품을 받은 배송지가 vertiport라면 한 대의 드론을 활용하여 고객에게 직접 배송을 진행한다. 만약 vertistop이라면 고객들이 직접 vertistop으로 방문하여 보관되어있는 물품을 수령한다.

제안 모델의 해를 도출하는 순서는 다음과 같다. 먼저, 고객의 수요량과 배송지의 최대 수용량을 고려하여 배송지에 고객을 배정한다. 모든 고객이 끝나면 UAM은 고객이 배정된 배송지를 따라 경로를 형성하며 배송을 진행한다. 이후, 각 배송지는 특성에 맞게 고객들에게 물품을 제공한다.

제안 모델에서 고객들은 물품을 받을 때 선호하는 시간이 정해져 있다. 제안 모델은 고객들의 선호 시간 이전에 배송을 완료하는 것을 목표로 하지만 이를 실패한 경우에는 불가능해라고 판단하는 대신 일정량의 페널티를 받게 된다. 이를 통해 제안 모델은 UAM의 이동 거리, 고객이 배송을 원하는 시간과 실제 배송을 받은 시간의 차이, 배송지 사용에 따른 비용을 최소화하는 경로를 찾을 수 있다. 아래 Figure 2는 UAM의 배송지의 선택적 방문에 대한 예시를, Figure 3은 vertiport와 vertistop에서의 다른 물품 배송 방식을 그림으로 나타낸다.



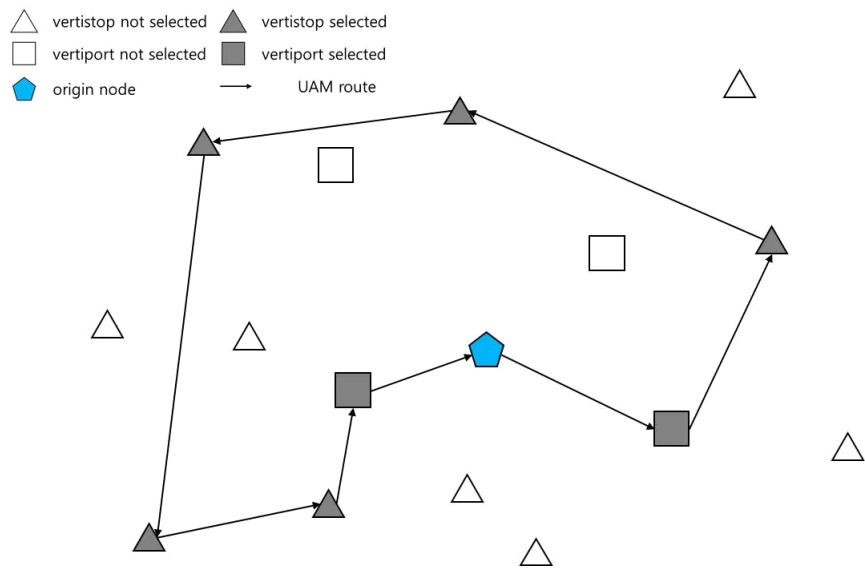
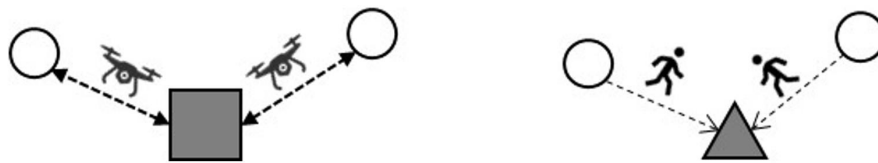


Figure 2. A schematic diagram of the UAM visiting the delivery centers selectively



(a) A drone delivers parcels to customers (b) Customers visiting the vertistop sequentially at vertiport to retrieve their parcels

Figure 3. Different delivery methods of vertiport and vertistop

### 3.1 집합과 파라미터

본 논문에서 제시하는 수리최적화모형에 사용되는 집합과 파라미터의 정의는 아래와 같다.

Table 1. Sets and Parameters

인덱스 집합 및 상수	
$C$	고객 인덱스의 집합
$P$	vertiport 인덱스의 집합
$S$	vertistop 인덱스의 집합
$D$	모든 배송지 인덱스의 집합 ( $P \cup S$ )
$0$	UAM 출발 지점의 인덱스 (디포, depot)
$d_{jk}$	배송지 $j$ 와 $k$ 사이의 거리 ( $j, k \in D, j \neq k$ )
$l_{ij}$	고객 $i$ 와 배송지 $j$ 사이의 거리 ( $i \in C, j \in D$ )
$C_j$	배송지 $j$ 를 사용하는데 드는 비용 ( $j \in D$ )
$p_i$	고객 $i$ 의 선호 배송 시간 ( $i \in C$ )
$demand_i$	고객 $i$ 의 수요량 ( $i \in C$ )
$R_{\max}$	허용 가능한 고객과 배송지 사이의 최대 거리
$v$	UAM의 속도
$v^d$	Drone의 속도
$v^p$	고객이 걸어서 vertistop을 방문할 때의 속도
$\alpha$	UAM 이동 거리에 대한 비용 가중치
$\beta$	고객의 선호 배송 시간과 실제 배송 시간의 차이에 대한 비용 가중치
$\gamma$	배송지 사용 비용에 대한 가중치
$N_j$	배송지 $j$ 의 최대 수용량 ( $j \in D$ )

Table 2. Decision variables

결정변수	
$x_{ij}$	고객 $i$ 의 제품이 배송지 $j$ 로 배송되면 1 아니면 0의 값을 가지는 이진변수 ( $i \in C, j \in D$ )
$y_j$	배송지 $j$ 가 사용되면 1 아니면 0의 값을 가지는 이진변수 ( $j \in D$ )
$z_{jk}$	UAM이 배송지 $j$ 방문 직후 배송지 $k$ 를 방문하면 1 아니면 0의 값을 가지는 이진변수 ( $j, k \in D$ )
$t_j$	UAM이 배송지 $j$ 에 도착하는 시간을 나타내는 연속변수 ( $j \in D$ )
$g_i$	고객 $i$ 의 선호 배송 시간과 실제 배송 시간의 차이를 나타내는 연속변수 ( $i \in C$ )
$t_{ij}$	$g_i$ 를 계산하기 위한 보조 결정변수인 연속변수 ( $i \in C, j \in D$ )
$t'_{ij}$	고객 $i$ 를 방문하기 전 드론이 vertiport $j$ 에 도착한 시간을 나타내는 연속변수 ( $i \in C, j \in P$ )
$w_{ij}^k$	Vertiport $k$ 에 있는 드론이 고객 $i$ 를 방문한 다음 고객 $j$ 를 방문하면 1 아니면 0의 값을 가지는 이진변수 ( $i, j \in C, k \in P, i \neq j$ )
$w'_{ij}$	드론이 고객 $i$ 를 방문하고 vertiport $j$ 를 방문하거나, vertiport $j$ 를 방문하고 고객 $i$ 를 방문하면 1 아니면 0의 값을 가지는 이진변수 ( $i, j \in C \cup P$ )
$u_j$	Depot을 시작으로 계산한 UAM의 배송지 $j$ 의 방문순서를 나타내는 정수변수 ( $u_j \in Z^+$ )

### 3.2 수리최적화모형

아래는 본 논문에서 제안하는 제안 모델의 수리최적화모형이다.

$$\text{Minimize } \alpha \sum_{j \in D} \sum_{\substack{k \in D \\ k \neq j}} d_{jk} z_{jk} + \beta \sum_{i \in C} g_i + \gamma \sum_{j \in D} C_j y_j \quad (1)$$

Subject to

$$\text{s.t. } \sum_{j \in D} x_{ij} = 1, \quad \forall i \in C \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i \in C, j \in D \quad (3)$$

$$\sum_{i \in C} x_{ij} \geq y_j, \quad \forall j \in D \quad (4)$$

$$z_{jk} \leq \frac{y_j + y_k}{2}, \quad \forall j, k \in D, j \neq k \quad (5)$$

$$y_j \geq z_{kj} + z_{jk}, \quad \forall j, k \in D, j \neq k \quad (6)$$

$$\sum_{k \in D} z_{kj} = \sum_{k \in D} z_{jk} = y_j, \quad \forall j \in D \quad (7)$$

$$\sum_{j \in D} z_{0j} = \sum_{j \in D} z_{j0} = 1 \quad (8)$$

$$t_0 = 0 \quad (9)$$

$$t_j + \frac{d_{jk}}{v} \leq t_k + M(1 - z_{jk}), \quad \forall j, k \in D, j \neq k \quad (10)$$

$$t_{ij} \geq t_j - M(1 - x_{ij}) + \frac{l_{jk}}{v^p} x_{ij}, \quad \forall i \in C, j \in S \quad (11)$$

$$t_{ij} \geq t'_{ij} - M(1 - x_{ij}) + \frac{l_{jk}}{v^d} x_{ij}, \quad \forall i \in C, j \in P \quad (12)$$

$$t'_{ij} \geq t_j - M(1 - w'_{ji}), \quad \forall i \in C, j \in P \quad (13)$$

$$t'_{jk} \geq t'_{ik} + 2 \frac{l_{ik}}{v^d} w_{ij}^k - M(1 - w_{ij}^k), \quad \forall i, j \in C, k \in P, i \neq j \quad (14)$$

$$g_i \geq t_{ij} - p_i x_{ij}, \quad \forall i \in C, j \in D \quad (15)$$

$$\sum_{i \in C} \text{demand}_i x_{ij} \leq N_j y_j, \quad \forall j \in D \quad (16)$$

$$\sum_{j \in D} y_j \geq 2 \quad (17)$$

$$\sum_{\substack{j \in C \\ i \neq j}} w_{ij}^k w'_{ik} = \sum_{\substack{j \in C \\ i \neq j}} w_{ji}^k w'_{ki} = y_k, \quad \forall i \in C, k \in P \quad (18)$$

$$\sum_{i \in C} w'_{ij} = \sum_{i \in C} w'_{ji} = y_j, \quad \forall j \in P \quad (19)$$

$$u_0 = 0 \quad (20)$$

$$u_j - u_k + 1 \leq |D| (1 - z_{jk}), \quad \forall j \in D \cup \{0\}, k \in D, j \neq k \quad (21)$$

$$u_j \leq |D| y_j, \quad \forall j \in D \quad (22)$$

$$0 \leq u_j \leq \sum_{k \in D} y_k, \quad \forall j \in D \quad (23)$$

$$x_{ij} \in \{0, 1\}, y_j \in \{0, 1\}, z_{jk} \in \{0, 1\}, \quad \forall i \in C, j, k \in D, j \neq k \quad (24)$$

$$w_{ij}^k \in \{0, 1\}, w'_{ij} \in \{0, 1\}, \quad \forall i, j \in C, k \in D, i \neq j \quad (25)$$

$$t'_{ij} \geq 0, \quad \forall i \in C, j \in P \quad (26)$$

$$g_i \geq 0, t_j \geq 0, t_{ij} \geq 0, \quad \forall i \in C, j \in D \quad (27)$$

식 (1)은 제안 모델이 UAM의 이동거리와 고객이 원하는 배송 시간과 실제 배송 시간의 차이, 배송지 사용에 따른 비용을 최소화하는 것을 목적으로 함을 나타낸다. 식 (2)는 모든 고객은 하나의 배송지에만 배정될 수 있다는 것을 나타낸다. 식 (3)과 (4)는 각각 배송지를 사용하기로 선택한 경우에만 해당 배송지에 고객이 배정될 수 있음을, 배송지에 배정된 고객이 없으면 해당 배송지는 사용하지 않음을 의미한다. 식 (5)–(7)은 사용되는 배송지들만 UAM이 방문한다는 것과 그 때의 흐름균형제약을 나타낸다. 식 (8)을 통해 UAM이 depot에서 출발해서 depot으로 돌아오도록 한다. 식 (9)와 (10)으로 UAM이 배송지에 도착하는 시간을 계산한다. 식 (11)은 vertistop에 할당된 고객들이 물품을 배송 받는 시간을 계산하며, 식 (12)–(14)을 통해 vertiport에서 드론이 고객들에게 물품을 배송하는 시간을 계산한다. 최종적으로 식 (15)를 통해 고객들의 선호 배송 시간과 실제 배송 시간의 차이를 구한다. 식 (16)은 배송지에 배정된 고객들의 수요총량은 배송지의 최대 수용량을 넘길 수 없음을 나타낸다. 식 (17)은 최소 2개의 배송지가 사용되어야함을 의미한다. 식 (18)과 (19)는 vertiport에서 드론이 방문하는 고객들의 순서에 대한 제약이다. 식 (20)–(23)은 부경로 생성을 방지하는 제약이다. 식 (24)와 (25)는 결정변수들의 이진조건을, 식 (26)과 (27)은 결정변수들의 비음조건을 의미한다.

### 3.3 휴리스틱 알고리즘

제안 모델의 경우 UAM이 모든 배송지가 아닌 선택된 배송지만을 방문하는 OP를 확장한 형태의 문제이다. Gunawan et al.(2016)에 따르면 OP는 Knapsack Problem과 TSP로 나누어질 수 있기 때문에 TSP의 일반적인 형태로 볼 수 있다. 이는 해당 문제가 TSP 문제의 한 종류임을 의미하며, TSP는 잘 알려진 NP-hard 문제이다. 이러한 문제의 수리최적화모형은 매우 큰 문제에 대해서 최적해를 다항 시간 (polynomial time) 안에 구하기 힘들다. 따라서 본 논문에서는 이를 해결하기 위해 2단계의 휴리스틱 알고

리즘을 제안한다. 1단계에서는 수리최적화 문제 중 하나인 상자 채우기 문제(bin packing problem)를 통해 휴리스틱에서 사용될 초기해를 구한다. 이후 2단계에서는 VNS 기반의 휴리스틱 알고리즘을 활용하여 해를 개선하게 된다.

### 1) 초기해 생성

초기해 생성에는 상자 채우기 문제의 수리최적화모형을 활용한다. 상자 채우기 문제는 정해진 물품을 모두 담기 위해 필요한 최소 상자의 개수와 어떤 상자에 어떤 물품이 들어갈지를 구하는 문제이다.

아래는 상자 채우기 문제를 풀기 위해 사용된 수리최적화모형이다. 3.1 절에 소개된 인덱스 집합과 상수, 결정변수를 공유한다.

$$\text{Minimize } \gamma \sum_{j \in D} C_j y_j + \delta \sum_{i \in D} \sum_{j \in D} l_{ij} x_{ij} \quad (28)$$

Subject to

$$\sum_{j \in D} x_{ij} = 1, \quad \forall i \in C \quad (29)$$

$$\sum_{i \in C} demand_i x_{ij} \leq N_j y_j, \quad \forall j \in D \quad (30)$$

여기서  $\delta$ 는 배송지와 해당 배송지에 배정된 고객 사이의 거리에 대한 비용 가중치를 의미한다. 식 (28)은 상자 채우기 문제의 목적식으로 배송지의 선정과 고객의 배정을 고려할 때, 배송지 선택에 따른 배용과 배송지와 고객 사이의 거리를 최소화하도록 한다. 식 (29)는 모든 고객이 각각 하나의 배송지에 배정되도록 제한한다. 식 (30)은 배송지에 배정된 고객의 수요량의 합이 해당 배송지의 최대 수용량을 넘지 않아야함을 의미한다.

휴리스틱의 초기해를 이러한 상자 채우기 문제를 수리최적화모형을 통해 풀어서 찾게 된다. 이를 통해 모든 고객을 적어도 하나의 배송지에 배정하기 위해 필요한 최소 배송지들과 해당 배송지들에 배정된 고객들을 구한다. 이렇게 구한 배송지 - 고객 쌍에 대해 경로를 생성한 후, 해당 값이 휴리스틱 알고리즘 2단계의 초기해로 사용 된다. 경로 생성 방법은 다음 절에서 설명한다.

### 2) 경로 생성

본 절에서는 상자 채우기 문제를 통해 구한 해를 휴리스틱 알고리즘의 초기해로 사용하기 전에 배송지 - 고객 쌍에 대해 UAM과 드론의 경로를 생성하는 과정에 대해 설명한다. 본 연구에서는 규칙 기반으로 경로를 생성하도록 하였다. UAM의 경우 상자 채우기 문제를 통해 구한 사용되는 배송지들에 대해서 최근접 이웃(nearest neighborhood) 방식을 적용하여 출발지에서부터 시작해서 가장 가까운 곳으로 이동하도록 하였다. 드론의 경우에도 최근접 이웃 방식을 적용하였지만 이때는 거리가 가장 가까운 고객부터 방문하는 것이 아니라, 배송 선호 시간이 가장 짧은 고객들부터 방문하도록 하였다.

### 3) 변동 이웃 탐색 알고리즘 (Variable Neighborhood Search Algorithm)

VNS는 메타 휴리스틱 기법으로 해 검색을 진행할 때, 이웃해의 구조를 다양하고 체계적으로 변경하는 방식에 기초한 알고리즘이다. VNS는 크게 다음과 같은 3가지 단계로 나뉘어진다.

Shaking : 사용될 이웃해 구조에 따라서 현재 최선해의 이웃해를 생성하는 과정

Local search : 생성된 이웃해들에 대해서 국지적으로 이웃해를 개선하는 과정

Move or Not : 개선된 이웃해가 현재 최선해보다 개선되었다면 현재 최선해를 이웃해로 치환하고, 아니면 다른 이웃해 구조를 시도하는 과정

VNS에서 핵심은 이웃해의 구조를 다양하고 체계적으로 설정하는 것이다. 이렇게 정해진 이웃해 구조를 기반으로 shaking이 진행되기 때문에 VNS의 성능은 이웃해 구조의 설정에 있다고 보아도 과언이 아니다. 따라서 VNS 사용할 때에는 이웃해 구조의 정의가 필수적이며, 아래와 같은 방식으로 해를 개선하게 된다.

#### [이웃해 구조 결정 및 초기해 도출]

**Step 1.** Shaking에 사용될 이웃해 구조를 결정한다.  $\{N_k | k=1, \dots, k_{\max}\}$  해당 이웃해 구조를 기반으로 임의의 해가 주어졌을 때 이웃해 집합을 생성하게 된다.

**Step 2.** 초기해(initial solution), 를 구하고 이를 최선해(best solution), 로 지정한다.

#### [VNS 기반 탐색]

**Step 1.** iteration Max\_Iteration인 경우 Step 2로 간다. 아니면 Step 7로 간다.

**Step 2.** 를 1로 초기화 해준다.

**Step 3.** 미리 결정한 이웃해 구조에 따라서 현재 최선해의 이웃해 집합을 생성한다. (이 때 집합에 속하는 이웃해의 수는 미리 정한  $n_{\text{neighbor}}$ 와 동일)

**Step 4.** 생성된 내의 이웃해에 대하여 국소 탐색(Local search) 방식을 적용하여 해를 개선한다. 이 중 가장 좋은 해를 로 지정한다.

**Step 5.** 만약 가 보다 좋은 해라면, 를 로 변경한 뒤, iteration을 iteration+1로 변경하고 Step 1로 간다. 아니면, 를 로 변경하고 Step 6으로 간다.

**Step 6.** 인 경우 Step 3으로 가고, 그렇지 않다면 Step 1로 간다.

**Step 7.** 를 최종해로 설정한 후 종료한다.

#### 4) 이웃해 탐색 (Shaking)

본 논문에서 가정하는 문제 상황에서 이웃해를 생성하는 방법은 배송지 &#8211; 고객 쌍의 변경을 기반으로 하였다. 배송지 &#8211; 고객 쌍의 변경이 가능한 상황은 vertiport, vertistop의 추가, vertiport, vertistop의 제거, vertiport간의 교환, vertistop간의 교환, vertiport와 vertistop의 교환, vertistop과 vertiport의 교환으로 총 8가지이다. 배송지의 추가, 삭제, 교환이 이루어지면 이에 따라 필수적으로 고객의 배송지 변경이 필요하다. 특히 새로 추가된 배송지에 몇 명의 고객을 배정할지가 문제가 되는데, 이는 기존 배송지들에 배정된 고객의 수 중에서 가장 작은 값만큼 기존 배송지의 고객을 새로운 배송지로 재배정 해주었다. 앞서 언급한 8가지 경우를 기반으로 아래와 같은 8가지 이웃해 구조를 설정하였다.

$k=1$  : 사용되지 않은 배송지 중에서 임의의 vertiport를 추가로 사용하도록 지정한다. 이때, 해당 vertiport에는 기존의 다른 배송지에 배정된 고객들 중 추가되는 vertiport와 가장 가까운 고객을 배정한다.

$k=2$  : 사용되지 않은 배송지 중에서 임의의 vertistop을 추가로 사용하도록 지정한다. 이때, 해당 vertistop에는 기존의 다른 배송지에 배정된 고객들 중 추가되는 vertistop과 가장 가까운 고객을 배정한다.

$k = 3$  : 사용되는 배송지 중에서 임의의 vertiport를 사용하지 않도록 지정한다. 이때, 해당 vertiport에 배정되었던 고객들은 나머지 사용되는 배송지 중 가장 가까운 곳에 최대 수용량을 초과하지 않는 선에서 추가 배정된다.

$k = 4$  : 사용되는 배송지 중에서 임의의 vertistop을 사용하지 않도록 지정한다. 이때, 해당 vertistop에 배정되었던 고객들은 나머지 사용되는 배송지 중 가장 가까운 곳에 최대 수용량을 초과하지 않는 선에서 추가 배정된다.

$k = 5$  : 사용되는 배송지 중에서 임의의 vertiport와 사용되지 않는 배송지 중에서 임의의 vertiport를 교환한다. 이때, 삭제된 vertiport에 배정되었던 고객들은  $k$ 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에  $k$ 가 1, 2일 때와 같은 방식으로 추가된 vertiport에 고객을 배정한다.

$k = 6$  : 사용되는 배송지 중에서 임의의 vertistop과 사용되지 않는 배송지 중에서 임의의 vertistop을 교환한다. 이때, 삭제된 vertistop에 배정되었던 고객들은  $k$ 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에  $k$ 가 1, 2일 때와 같은 방식으로 추가된 vertistop에 고객을 배정한다.

$k = 7$  : 사용되는 배송지 중에서 임의의 vertiport와 사용되지 않는 배송지 중에서 임의의 vertistop을 교환한다. 이때, 삭제된 vertiport에 배정되었던 고객들은  $k$ 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에  $k$ 가 1, 2일 때와 같은 방식으로 추가된 vertistop에 고객을 배정한다.

$k = 8$  : 사용되는 배송지 중에서 임의의 vertistop과 사용되지 않는 배송지 중에서 임의의 vertiport를 교환한다. 이때, 삭제된 vertistop에 배정되었던 고객들은  $k$ 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에  $k$ 가 1, 2일 때와 같은 방식으로 추가된 vertiport에 고객을 배정한다.

각 이웃해 방식은 선택되었을 때 알고리즘 실행 시 미리 설정해둔 이웃해의 개수만큼 해당 이웃해 방식으로 이웃해를 생성한다. 이렇게 생성된 이웃해 집합은 다음 절에서 설명할 국소 탐색의 입력값으로 사용되어 해의 개선이 이루어진다.

## 5) 국소 탐색(Local search)

국소 탐색은 UAM 혹은 드론의 배송 및 방문순서의 변경으로 고려하였다. UAM이 방문하는 배송지의 순서를 변경하거나 각 vertiport에서 드론이 방문하는 고객들의 순서를 변경하는 방식으로 국소 탐색을 진행하였다. 자세한 방법은 아래와 같다.

### [Pseudocode]

**Step 1.** iteration  $\leq$  Max\_Iteration인 경우 Step 2로 간다. 아니면 Step 6으로 간다.

**Step 2.** UAM과 드론 중 어떤 경로를 변경할지 임의로 선택한다. UAM이 선택되었다면 Step 3으로, 아니라면 Step 4로 간다.

**Step 3.** UAM 경로에 있는 배송지들 중 임의로 2개를 선택하여 방문순서를 변경하고 소요시간을 계산한다. 만약, 새로운 경로가 기존 경로보다 소요시간이 짧다면 기존 경로를 새로운 경로로 변경한다. Step 5로 이동한다.

**Step 4.** 고객이 2명 이상 할당된 vertiport 중 하나를 임의로 선택하여, 고객들 중 임의로 2명을 선택하여 방문순서를 변경하고 소요시간을 계산한다. 만약, 새로운 방문순서가 기존 방문순서보다 소요시간이 짧다면 기존 방문순서를 새로운 방문순서로 변경한다. Step 5로 이동한다.

**Step 5.** iteration을 iteration+1로 변경하고 Step 1로 간다.

**Step 6.** 현재 경로를 최종 경로로 설정한 후 종료한다.

## 4. 실험 및 분석

본 절에서는 수리최적화모형과 휴리스틱 알고리즘의 실험 환경과 실험 결과에 대해 설명하고 이를 분석한다. UAM을 통해 배송을 진행할 때 vertiport와 vertistop으로 나누어 구분하며, UAM과 드론의 협업배송을 고려하는 경로 문제는 지금까지 연구된 경우가 없다. 이로 인해, 제안 모델의 성능을 검증하기 위해 비교할 대상이 없다. 따라서 본 논문에서는 제안한 수리최적화모형과 휴리스틱 알고리즘의 성능을 비교하여 검증을 수행하였고, 동일한 문제를 기본적인 TSP로 풀었을 때의 결과를 함께 비교하였다.

### 4.1 실험 설정

실험에 사용된 데이터는 Solomon 데이터로 이는 Solomon(1987)이 제안한 경로 문제에 주로 사용되는 데이터이다. Solomon 데이터는 크게 C (Clustered), R (Random), RC (Random Clusterd) 형식으로 나누어진다. C 형식 데이터는 고객들이 여러 군집을 이루어 존재하며, R 형식 데이터는 고객들이 군집을 이루지 않고 임의의 위치에 분포되어 존재한다. 마지막 RC 형식 데이터는 C 형식 데이터와 R 형식 데이터를 합친 형태로, 군집을 이루어 존재하는 고객들과 임의의 위치에 분포하는 고객들이 혼재되어 있다. Solomon 데이터의 크기는 25, 50, 100으로 나누어진다.

수리최적화모형의 경우 문제의 크기가 커지면 해를 구하기 매우 어려워지는 특성 때문에, 수리최적화모형은 Solomon 데이터 중 고객의 수가 25, 50인 데이터를 재구성한 작은 문제에 대해 실험을 진행하였다.

Solomon 데이터에는 출발점인 depot과 고객 노드들만이 존재하기 때문에 UAM이 방문하고, 고객들이 배정될 배송지 노드가 존재하지 않는다. 이를 해결하기 위해 배송지 중 vertiport는 임의의 위치에 생성하였고, vertistop은 기존 고객 노드들 중 일부를 선택하여 사용하여 Solomon 데이터의 성질과 형태를 최대한 유지하고자 하였다.

제안 모델에서는 UAM과 드론을 사용하며, vertistop에 배정된 고객의 경우에는 고객이 직접 방문하여 물품을 수령하기 때문에 UAM, 드론, 고객의 속도를 지정해 주어야 한다. Straubinger et al.(2020)에서는 UAM의 속도에 대해 150km/h에서 200km/h, 240km/h 등 다양하게 제시하였다. 본 논문에서는 가장 보수적인 기준을 적용하여 UAM의 속도를 150km/h로 가정하였다. 전기 드론의 속도는 Restas et al.(2021)을 참고하여 50km/h로 가정하였으며, 고객의 이동 속도는 다양한 이동수단의 활용을 고려한 Shin et al.(2022)의 연구를 참고해 10km/h로 가정하여 매개변수를 산정해주었다. 최종적으로 UAM의 속도는 3, 드론의 속도는 1, 사람의 속도는 0.2로 설정하였다.

실험은 AMD Ryzen 9 5900X 12-Core Processor, 64GB RAM에서 수행하였으며, 수리최적화모형을 풀기 위한 solver로 Gurobi 9.5.2 버전을 사용하였다. 휴리스틱 알고리즘은 Python 3.8.11 환경에서 개발하였다.

### 4.2 실험 결과

본 절에서는 Solomon 데이터를 활용한 수리최적화모형과 휴리스틱 알고리즘의 실험 결과에 대한 설명과 분석을 진행한다.

먼저, 수리최적화모형의 결과에 대해 기술한다. 최적해를 도출할 수 있도록 총 노드 수를 10, 11, 13개로 줄여서 실험을 진행하였다. 이때, 고객의 수를 7, 8, 9명, vertiport의 수를 1, 1, 1개, vertistop의 수를 2, 2, 3개로 설정하였으며, Table 3을 통해 그 결과를 확인할 수 있다. 제안 모델의 성능을 평가하기 위해 경로 문제의 가장 기본인 TSP의 해와 함께 비교하였다. TSP의 경우 Google(2022)의 OR-Tools를 사용하였고, 실험을 진행할 때 도심의 평균 차량의 속도는 Shin et al.(2022)의 연구를 참고하여 30km/h로 가정하였다. 앞서 50km/h로 가정한 드론의 속도 파라미터를 1로 지정하였기 때문에 30km/h로 가정한 차량의 속도 파라미터는 0.6으로 지정하였다. 수리최적화모형은 3600초(1시간)의 연산시간 제한을 두어 최적해를 구하지 못한 경우 최선해의 값을 표기하였다.



Table 3. Experiment result of the MILP and OR-Tools TSP

Problem	Number of Nodes				MILP		TSP	
	Customer	Verti port	Vert istop	Total	Solution (gap)	Run Time	Solution	Run Time
C101	8	2	3	13	20.00 (0%)	0.11	173	0.003
C101	9	2	4	15	15.00 (0%)	0.02	182	0.004
C101	10	3	5	18	25.00 (0%)	0.08	179	0.006
R101	8	2	3	13	35.00 (0%)	21.06	295	0.003
R101	9	2	4	15	58.62 (42.5%)	3600	276	0.003
R101	10	3	5	18	49.52 (9.1%)	3600	338	0.005
RC101	8	2	3	13	30.00 (0%)	5.27	299	0.006
RC101	9	2	4	15	31.15 (0%)	10.32	300	0.004
RC101	10	3	5	18	45.00 (14.3%)	3600	329	0.004

Table 3을 살펴보면 최적해와 Google OR-Tools의 TSP 모델과 비교하기 위해 작은 크기로 데이터를 조절하였지만, R101 데이터의 경우 총 노드 수가 15개, 18개일 때, RC101 데이터 경우 총 노드의 수가 15개일 때 3600초 동안의 연산에도 최적해를 구하지 못한 것을 확인할 수 있다. 또한, R101 데이터의 경우 15개 노드 수의 문제에 대해 42.5%, 18개 노드 수의 문제에 대해 9.1%의 gap을 보였고, RC101 데이터의 경우 18개 노드 수의 문제에 대해 14.3%의 gap을 보였다. 결과를 보면 대체로 군집화 되어있는 경우에 더 좋은 해를 도출하는 것을 볼 수 있다. 이는 제안 모델이 가지는 성질 때문으로 보이며, 이는 Table 4의 결과와 함께 다시 설명하도록 하겠다. 노드의 수를 더 줄여서 실험을 진행할 수도 있지만, 고려하는 vertiport와 vertistop의 수가 더 줄어들면 고정적인 해를 도출할 위험이 있어 위와 같은 설정으로 실험을 진행하였다.

제시한 수리최적화모형이 최적해를 도출한 경우 TSP와 비교하였을 때, 80% 이상 개선된 해를 도출한 것을 확인할 수 있다. 이는 제안 모델의 목적식의 비용 고려를 위한 파라미터 설정에 의한 차이와 더불어, UAM과 드론의 속도가 차량의 속도보다 훨씬 빠르다는 점이 크게 작용한 것으로 보인다.

다음으로 휴리스틱 알고리즘이다. 앞서 살펴본 것처럼 수리최적화모형은 문제가 조금만 커지더라도 최적해를 찾는 데 매우 시간이 오래 걸리게 된다. 이에 따라 큰 문제에 대해서도 적절한 해를 제공할 수 있는 2단계 휴리스틱 알고리즘을 개발하였다. 휴리스틱 알고리즘의 경우 Solomon 데이터 중 노드 수가 25개, 50개, 100개인 모든 경우에 대해서 실험을 진행하였고, 생성할 이웃해는 50개, 국소 탐색 100회, 총 반복 100회로 설정하여 실험을 진행하였다. Table 4를 통해 그 결과를 확인할 수 있다. 마찬가지로 수리최적화모형의 경우 3600초의 연산시간 제한을 두고 실험을 진행하였다.

Table 4. Experiment result of the MILP and the heuristic algorithm for proposed model

Problem	Number of Nodes				MILP		Heuristic	
	Customer	Verti port	Verti stop	Total	Solution (gap)	Run Time	Solution	Run Time
C101	18	2	4	25	30.00 (0%)	0.23	35.81	87.34
C101	35	4	11	50	60.00 (0%)	6.22	108.85	105.80
C101	71	9	20	100	251.90 (48.4%)	3600	200.90	127.29
R101	18	2	4	25	202.23 (85.2%)	3600	218.63	93.21
R101	35	4	11	50	677.20 (91.8%)	3600	175.83	90.00
R101	71	9	20	100	2956.76 (96.4%)	3600	214.82	117.51
RC101	18	2	4	25	376.49 (89.4%)	3600	218.63	105.57
RC101	35	4	11	50	793.90 (90.6%)	3600	223.17	92.38
RC101	71	9	20	100	6026.57 (97.9%)	3600	331.07	138.27

휴리스틱의 경우 수리모형이 최적해를 구한 경우에는 근소한 차이로 유효해를 찾은 경우도 있지만 최적해와 차이를 보이는 경우도 있다. 수리최적화모형이 3600초 동안 최적해를 구하지 못한 경우에 대해서는 한 경우를 제외하고 모두 수리최적화모형보다 더 좋은 해를 도출하였다.

Table 4의 실험 결과 역시 Table 3에서와 비슷한 성향을 보인다. 고객들이 군집을 이루는 형태인 C101 데이터의 경우 문제를 푸는 속도가 R101, RC101 데이터에 비해 빠르고, 더 낮은 해를 도출하는 것을 볼 수 있다. 총 노드 50개를 고려하는 문제의 경우 C101 데이터는 6.22초 만에 최적해를 구한 반면, R101과 RC101 데이터는 3600초 동안 최적해를 찾지 못한 모습이다.

이러한 이유가 발생하는 제안 모델의 특성은 다음과 같다고 판단된다. 제안 모델의 경우 UAM이 방문하는 배송지의 위치와 고객의 위치에 따라 해의 값이 크게 영향을 받는다. 배송지들과 고객들이 가깝게 위치한다면, 고객에게 물품 배송까지의 시간이 크게 줄어들게 되고 이 경우 UAM의 이동 거리가 주요 비용으로 작용하게 된다. 반면에 배송지들과 고객들 사이의 거리가 멀다면, UAM이 아무리 빠르게 배송을 완료하여도 고객이 직접 물품을 수령 하는 시간은 늦어지게 된다. 이런 경우에는 UAM의 이동 거리와 더불어 고객이 배송받기를 원하는 시간과 실제 배송을 받은 시간의 차이가 주요 비용으로 작용한다. 이러한 특성으로 인해 제안된 모델은 문제를 풀기 위한 네트워크의 상태에 따라 해가 매우 크게 변동할 수 있음을 알 수 있다.

## 5. 결론

본 논문에서는 UAM과 드론의 협업 배송모형을 제시한다. 본 연구는 UAM을 배송에 활용하며, UAM의 방문 노드를 UAM 운영에 필수적인 기반 시설인 vertiport와 vertistop로 나누어 구분한다. 또한 vertiport와 vertistop은 각각 다른 방식으로 고객에게 최종 물품을 전달한다. 이러한 문제 상황을 반영한 수리최적화모형을 제시하였다. 문제 크기가 커짐에 따라 수리모형이 유효해를 구하기 어려워지는 문제로 인해 수리최적화와 메타-휴리스틱을 함께 사용하는 2단계의 휴리스틱 알고리즘을 제시하였다. 휴리스틱 알고리즘은 1단계에서는 수리최적화모형인 상자 채우기 문제를 풀어 메타-휴리스틱에서 사용할 초기해를 찾는다. 2단계에서는 VNS 기반의 휴리스틱 알고리즘을 사용하여 1단계에서 찾은 초기해를 바탕으로 보다 개선된 해를 도출한다.

제시한 수리최적화모형과 휴리스틱 알고리즘의 성능평가를 위해 Solomon 데이터를 활용한 실험을 진행하였다. 실험 결과를 통해 제시한 수리최적화모형이 UAM이나 드론을 사용하지 않는 방식보다 우수함을 확인할 수 있었으며, 작은 문제 크기에 대해 휴리스틱 알고리즘이 수리최적화모형과 근접한 해를 도출하는 것을 확인하였다. 이를 통해 본 논문에서 제시한 수리최적화모형과 휴리스틱 알고리즘이 기존 방식을 이동거리 및 시간 면에서 크게 발전시킬 수 있다는 것을 확인하였다.

본 연구에서는 UAM과 드론의 협업 배송 시스템이라는 새로운 문제를 제안하였기 때문에 이를 발전시킬 추후 연구 방향은 다양할 것으로 판단된다. Vertiport와 Vertistop에서의 배송 방식을 확장하여 vertiport에서는 차량을 통해 하나의 경로를 형성하여 배송을 하고, vertistop에서는 하나의 드론을 통해 고객에게 배송을 진행하는 방식의 연구가 다음 연구 주제가 될 수 있다. 또한, 본 연구에서는 vertiport에서 하나의 드론만을 사용하지만, 다수의 드론을 활용하여 배송을 진행하는 방식의 연구도 추후 연구 방향으로서 제시한다.

## 참고문헌

- Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60(6), 831-842.
- Bennaceur, M., Delmas, R., & Hamadi, Y. (2022). Passenger-centric Urban Air Mobility: Fairness trade-offs and operational efficiency. *Transportation Research Part C: Emerging Technologies*, 136, 103519.
- Cheon Seunghoon., Kim, Seongmin., & Lee, Chaeyoung. (2020). Paradigm Change in Traffic Congestion Cost Estimation and Results of Traffic Congestion Cost Estimation in 2017. *Monthly KOTI Magazine on Transport*, (), 11-16.
- Choi, J-S., Lee, S-H., Baek, J-S., Hwang, H-W. (2021). A Study on Vertiport Installation Standard of Drone Taxis(UAM). *Journal of the Korean Society for Aviation and Aeronautics*, 29(1), 74-81.
- Fadhil, D. N. (2018). A GIS-based analysis for selecting ground infrastructure locations for urban air mobility. inlangen]. Master's Thesis, Technical University of Munich.
- Geem, Z. W., Tseng, C. L., & Park, Y. (2005, August). Harmony search for generalized orienteering problem: best touring in China. In *International conference on natural computation* (pp. 741-750). Springer, Berlin, Heidelberg.
- Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, 34(3), 307-318.
- Google OR-Tools (2022). [https://developers.google.com/optimization/routing/tsp#cpp\\_8](https://developers.google.com/optimization/routing/tsp#cpp_8). (Accessed on 10/07/2022)
- Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2), 315-332.
- Hassanzadeh, A., Mohseninezhad, L., Tirdad, A., Dadgostari, F., & Zolfagharinia, H. (2009). Location-routing problem. In *Facility Location* (pp. 395-417). Physica, Heidelberg.
- Hong, J., Lee, M., Cheong, T., & Lee, H. C. (2019). Routing for an on-demand logistics service. *Transportation Research Part C: Emerging Technologies*, 103, 328-351.
- Hyundai Motor Group. (2022), <https://tech.hyundaimotorgroup.com/kr/mobility-device/uam/>. (Accessed on 10/07/2022).
- Ilhan, T., Iravani, S. M., & Daskin, M. S. (2008). The orienteering problem with stochastic profits. *IIE Transactions*, 40(4), 406-421.
- Jeong, J., So, M., & Hwang, H. Y. (2021). Selection of Vertiports Using K-Means Algorithm and Noise Analyses for Urban Air Mobility (UAM) in the Seoul Metropolitan Area. *Applied Sciences*, 11(12), 5729.
- Kim, S., & Moon, I. (2018). Traveling salesman problem with a drone station. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1), 42-52.
- Labadie, N., Mansini, R., Melechovský, J., & Calvo, R. W. (2012). The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research*, 220(1), 15-27.
- Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European journal of operational research*, 177(2), 649-672.

- Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86-109.
- Restás, Á., Szalkai, I., & Óvári, G. (2021). Drone Application for Spraying Disinfection Liquid Fighting against the COVID-19 Pandemic—Examining Drone-Related Parameters Influencing Effectiveness. *Drones*, 5(3), 58.
- Shin, H., Lee, T., & Lee, H. R. (2022). Skyport location problem for urban air mobility system. *Computers & Operations Research*, 138, 105611.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35 , 254-265.
- Straubinger, A., Rothfeld, R., Shamiyeh, M., Büchter, K. D., Kaiser, J., & Plötner, K. O. (2020). An overview of current research and developments in urban air mobility—Setting the scene for UAM introduction. *Journal of Air Transport Management*, 87, 101852.
- Verbeeck, C., Sörensen, K., Aghezzaf, E. H., & Vansteenwegen, P. (2014). A fast solution method for the time-dependent orienteering problem. *European Journal of Operational Research*, 236(2), 419-432.
- Wu, Z., & Zhang, Y. (2021). Integrated Network Design and Demand Forecast for On-Demand Urban Air Mobility. *Engineering*, 7(4), 473-487.

# 추가학습이 불필요한 이미지 특징 유사도 기반 상품 식별 시스템

유영재<sup>1,3</sup>, 윤혜정<sup>2</sup>, 김준오<sup>2</sup>, 박예솔<sup>2</sup>, 장병탁<sup>1,2,3†</sup>

<sup>1</sup>서울대학교 컴퓨터공학부

<sup>2</sup>서울대학교 협동과정 인공지능전공

<sup>3</sup>투모로 로보틱스

## Product Identification System based on Image Feature Similarity with Learning-free Model

Youngjae Yoo<sup>1,3</sup>, HyeJung Yoon<sup>2</sup>, Juno Kim<sup>2</sup>, Yesol Park<sup>2</sup>, Byoung-Tak Zhang<sup>1,2,3†</sup>

<sup>1</sup>Department of Computer Science and Engineering, Seoul National University

<sup>2</sup>Interdisciplinary Program in Artificial Intelligence, Seoul National University

<sup>3</sup>Tommoro Robotics

Recently, logistics centers attempts to get help from artificial intelligence robots with hard labor. To generally perform picking task for robots, it is essential to detect and identify the product through the camera. Supervised learning-based deep learning technology is suitable for recognizing objects with high accuracy. But it has a disadvantage that requires a lot of time because a human must manually label the answer of the train image. In this paper, we propose a method that minimizes manual labor and makes it easy to add products. Our algorithm identifies the product with the highest similarity by calculating the similarity between the features of the input image and the features of the images in the product database. It does not require learning and labeling to identify a new product. To verify it, we test the algorithm by photographing a test product images in an environment that simulates a logistics site.

**Keywords:** Object Recognition, Image Feature Matching, Learning-free Model, Logistics Automation

---

논문접수일 : 2022.10.07.

심사완료일 : 2022.12.17.

게재확정일 : 2022.12.19.

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원(2021-0-02068-AIHub/25%, 2021-0-01343-GSAI/20%, 2022-0-00951-LBA/20%, 2022-0-00166-PICA/25%, NO.2021-0-01343, 인공지능대학원지원(서울대학교)/10%)의 지원을 받아 수행되었음.

† Corresponding Author: btzhang@snu.ac.kr

## 1. 서론

### 1.1 연구배경

코로나19 및 디지털 전환으로 인해 온라인 쇼핑과 같은 전자상 거래가 폭발적으로 증가하며, 물류 센터의 필요 인력도 함께 늘고 있다(Singhdong et al., 2021). 하지만 포장, 분류, 상하차 등 물류센터의 업무는 높은 노동 강도가 요구되기에 최근 인공지능 기술을 갖춘 로봇에게 도움을 받으려는 시도가 활발히 이루어지고 있다(Karabegović et al., 2015). 고강도의 물류 작업에 적용하기 적절한 로봇은 작업대에 팔이 고정된 형태의 암 로봇이 일반적이다. <Figure 1>과 같이 로봇이 물체를 탐지하기 위해 필수적인 기술은 카메라를 통해 상품을 식별하는 것이다. 이는 로봇이 물체를 적절한 위치에 옮기거나, 분류하는 데 필수적이다.

물체의 종류를 식별하기 위해 현재 물류센터에서 사용하는 대표적인 방법은 바코드를 리더기로 읽어 파악하는 것이다. 바코드라벨은 주로 상품의 로케이션 관리, 납품처 식별, 재고관리 등에 활용되고 있다. 하지만 바코드는 정형화된 박스의 형태가 아닌 경우 배치된 면이 일관되지 않기에 위치를 파악하는 것이 쉽지 않아 사람의 도움 없이 정확한 식별을 하는 데 한계가 있다 (Grover et al., 2010).

단순 이미지만을 사용하는 경우, 이미지 해쉬와 같은 이미지 검색을 통해 부분적으로 해결할 수 있다 (Wang et al., 2015). 이미지 해쉬는 이를 이미지를 해쉬함수를 거쳐 해쉬값으로 변환한 뒤 유사도를 분석하는 방법이다. 하지만 이 방법은 동일 이미지를 찾는 데 특화되어 있기에, 물류 환경과 같이 배경이 복잡하고 구도와 조명, 촬영한 카메라가 다른 경우 유사성을 탐지하기 어렵다.

인공지능 분야에선 이를 위해 감독학습 기반 물체 탐지 연구가 활발히 이루어졌다(Cunningham et al., 2008). 감독학습 기반 모델 물체의 사진 데이터와 물체 위치 정보, 물체 종류를 데이터화하는 라벨링 작업을 거쳐 학습을 통해 개발한다. 이들은 보다 복잡한 이미지에 대해서도 높은 성능을 보이는 장점이 있지만, 학습을 위한 라벨링은 모두 사람이 수작업으로 표기하는 작업이 필요하여 많은 시간과 노력이 필요한 단점이 있다.

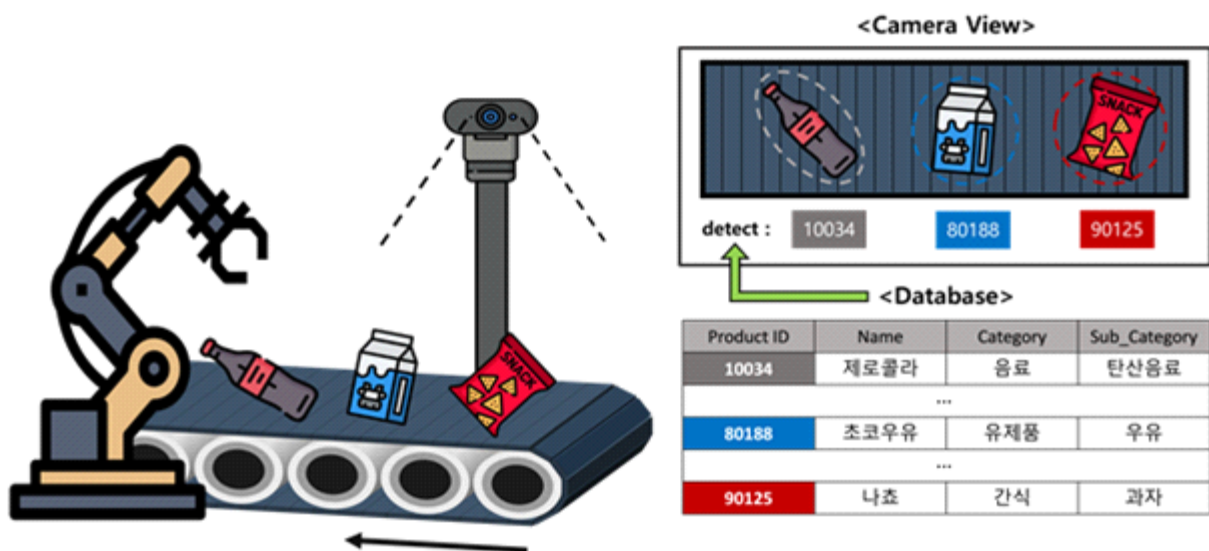


Figure 1. Object recognition for robot picking tasks

## 1.2 문제정의 및 논문구성

물류의 자동화를 위해 상품 이미지를 통해 종류를 식별하는 것은 필수적이거나, 많은 데이터를 수집한 뒤 라벨링을 수작업으로 표기하는 작업에 많은 노력이 필요한 한계점이 있다. 또한, 새로운 물체를 추가하려면 다시 데이터 수집 및 라벨링, 학습을 수행해야 하므로 상품 추가 및 수정이 용이하지 않다. 그리고 다뤄야 하는 물체의 개수가 많아지면 이 수작업은 기하급수적으로 늘어난다. 하지만 대부분의 물류 상품의 경우 일반적으로 공업적으로 생산되어 외관이 같으며, 카메라를 통한 이미지 수집 시 작업 환경에 따라 회전, 빛 반사, 밝기 등의 제한된 변수만이 존재한다. 따라서 이와 같은 제한된 변수를 고려하면 학습을 거치지 않고, 이미지의 유사도를 비교해 동일 상품을 데이터베이스 내에서 검색하는 방식으로 식별을 수행하면 많은 양의 데이터 수집 및 라벨링이 불필요할 것이라는 발상에서 본 연구를 시작하였다.

이처럼 상품 종류의 변동이 용이하며 사람의 수작업을 최소화하는 방안으로 본 논문에선 학습 및 라벨링이 필요하지 않고, 물체의 앞뒷면 사진만을 필요로 하는 이미지 특징 비교를 통한 상품 식별 알고리즘을 제안한다. 먼저 2장 관련 연구에선 로봇의 물체 파지를 위한 물체 탐지, 물체를 잘 탐지하기 위한 이미지 전처리, 특징 추출, 그리고 특징 간 유사도 비교를 위해 사용할 수 있는 지표 등을 설명한다. 3장에서는 제시하는 알고리즘을 설명한다. 먼저 입력 이미지를 배경 제거 및 이미지 전처리를 거쳐 특징을 추출한 뒤, 상품 데이터베이스 내 상품 이미지의 특징과 유사도를 비교하여 가장 유사한 상품을 찾아낸다. 4장에서는 이를 검증하기 위해 물류 현장을 묘사한 환경에서 상품을 촬영해 수집한 테스트 셋을 소개하고, 구축한 상품 데이터베이스를 소개한다. 그리고 실험의 결과를 보이며 제시한 알고리즘을 사례와 함께 분석한다. 5장에서는 결론을 정리하고 후속 연구 계획을 소개한다.

## 2. 관련 연구

### 2.1 로봇의 파지 작업을 위한 물체 탐지

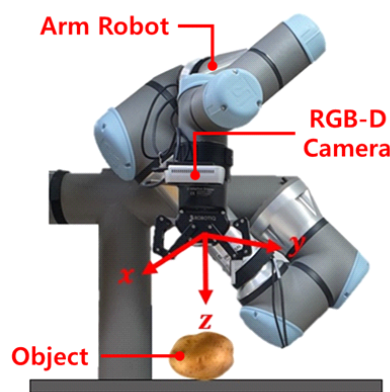


Figure 2. Arm Robot for object grasp manipulation

작업대에 고정된 암 로봇은 무거운 물체들을 파지하여 목표 지점까지 옮기는 작업을 수행할 수 있다. 이때 인공지능을 기반으로 파지할 목표 물체를 탐지하기 위해선 <Figure 2> 에서와 같이 천장이나 작업대에 고정된 rgb-d 카메라를 일반적으로 사용한다. 인공지능 기반의 목표 물체 탐지는 먼저 RGB 카메라로 물체의



2차원 이미지상 위치를 파악하고, 심도(Depth) 카메라를 활용해 3차원의 실세계 거리를 파악한다. 그리고 카메라와 로봇팔의 위치를 고려해 로봇팔이 손을 뻗을 거리를 계산한다(Sarabu et al., 2019). Zhuang et al.(2021)은 물체 위치 탐지를 위해 물체 탐지 및 분할을 수행하였다. Semantic PPF이라는 이 방법은, pointcloud 데이터를 기반으로 object-part 객체 분할을 수행한다. 그리고 현실 데이터를 다량 수집하는데 많은 시간과 제약이 있음을 고려해 가상 물리엔진을 이용해 객체 분할 데이터를 효율적으로 생성하여 데이터 생성의 비용을 줄였다. Rennie et al.(2016)은 물류창고에서 상품 파지를 위해 특화된 데이터 셋을 공개하기도 했다. 해당 데이터 셋은 약 10000장의 RGB, Depth 이미지와 3D 물체 위치를 함께 제공하였다.

## 2.2 이미지 유사도 분석

이미지 유사도는 이미지 데이터 간의 같은 위치의 픽셀값, 인접한 픽셀값의 변화량, 밝기, 대비, 색분포 등 다양한 기준을 비교하여 계산할 수 있다. 대표적인 이미지 유사도 분석법으로는 이미지 히스토그램과 이미지 해쉬 방법 등이 있다. 이미지 히스토그램은 가로축에 이미지 픽셀값을, 세로축에 이미지 픽셀 수를 나타내어 이미지의 특성을 비교하는 방법이다(Chapelle et al., 1999; Jia et al., 2006). 하지만 유사한 색 분포만을 가진 상품이 있거나, 상품의 방향이 바뀌는 등의 경우엔 정확성이 떨어지는 한계가 있다.

이미지 해쉬는 <Figure 3> 과 같이 이미지를 해쉬함수를 거쳐 고유한 해쉬값으로 만든 뒤 비교하는 방법으로, 크게 average hash, perceptive hash, difference hash 등이 있다(Wang et al., 2015; Zauner, 2010). 이미지 해쉬는 이미지를 낮은 차원으로 압축한 뒤 색 단순화를 거쳐 2차원 배열 혹은 해쉬값을 얻고 hamming 거리를 구해 유사도를 비교할 수 있다. 해쉬값을 사용하여 빠른 시간이 소요되는 장점이 있지만, 정해진 사이즈로 압축해야 하거나 단일 해쉬값으로 변환해야 하기에 충분한 특징을 추출하는 데 한계가 있기에 상품의 촬영 환경에 따라 정확도가 떨어지는 한계가 있다.

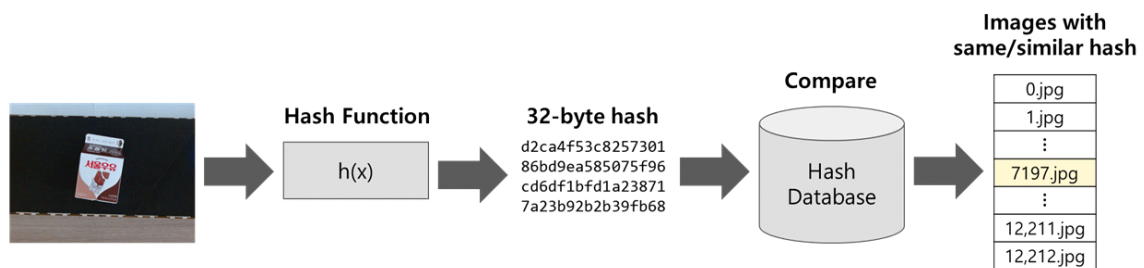


Figure 3. Similarity analysis using Image Hash

## 2.3 이미지 전처리 및 특징 추출

### 1) 이미지 전처리

이미지에서 특징을 추출하기 위해 모델에 입력하기 이전에 보다 이미지를 명확하게 가공하는 과정을 전처리라 하며, <Figure 4> 와 같이 크게 두 가지로 나눌 수 있다(Shorten and Khoshgoflaar, 2019). 첫 번째는 주어진 이미지의 픽셀 단위로 처리하는 픽셀 단계 변환(Pixel-Level Transform)으로, 흐림(Blur), 대비(Contrast), 양각(Emboss), 선명도(Sharpen) 등의 기법이 존재한다(Buslaev et al., 2020; Li et al., 2017; Singh and Kapoor, 2014). 두 번째는 이미지 공간 자체에 변형을 주는 공간 단계 변환(Spatial-Level Transform)이다. 대표적으로 뒤집기(Flip), 회전(Rotation), 그리고 이미지의 일부 영역만 이용하는 잘라내기(Crop) 등이 이에 해당한다.

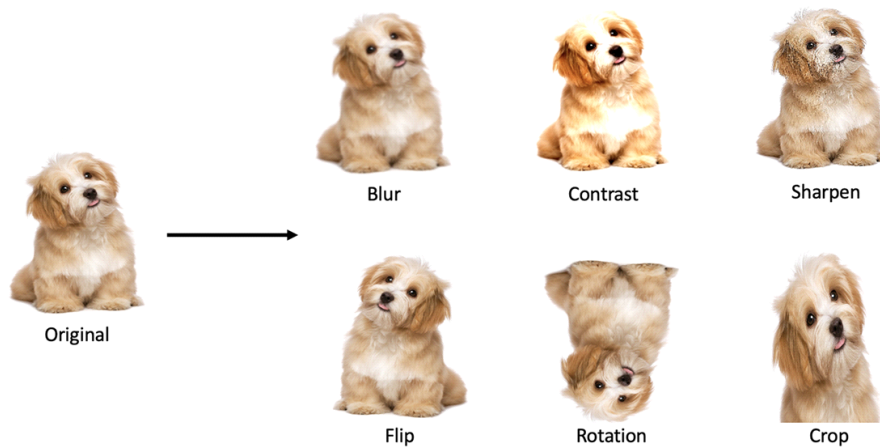


Figure 4. Image preprocessing

## 2) 이미지 객체 분할

이미지 객체 분할(Instance Segmentation)이란, 이미지 내에서 픽셀 단위로 영역을 분리해 객체를 추출하는 방법을 말한다(Zaitoun and Aqel, 2015). U-Net은 객체 분할 모델 중 하나로, 모델의 모양이 U자를 띠고 있다(Ronneberger et al., 2015). <Figure 5> (a)에서와 같이 U-Net은 크게 수축 경로(Contracting path)와 전환 구간(Bottle Neck), 그리고 확장 경로(Expansive path)로 이루어져 있다. 수축 경로를 통해 점진적으로 넓은 범위의 이미지 픽셀을 보며 의미 정보를 추출하고, 전환 구간에서 확장 경로로 바뀐다. 확장 경로에서 해당 정보를 위치 정보와 결합해 각 픽셀이 어떤 객체에 속하는지 구분한다. 이 과정을 거쳐 입력된 넓은 범위의 이미지에서 의미 있는 객체를 가져온다. Qin et al., (2020)에서 소개한 U2-Net (U-Square Net)은 Ronneberger et al.,(2015)의 U-Net 구조를 가진 블록(residual U-block)들이 또 하나의 U-Net을 이루고 있는 모델이다. Rembg는 이러한 U2-Net을 기반으로 사전 학습한 모델을 제공하는 배경 제거 도구로, <Figure 5> (b)처럼 입력된 이미지에서 배경을 제거하고 객체에 해당하는 부분만 <Figure 5> (c)와 같이 얻어낼 수 있다.

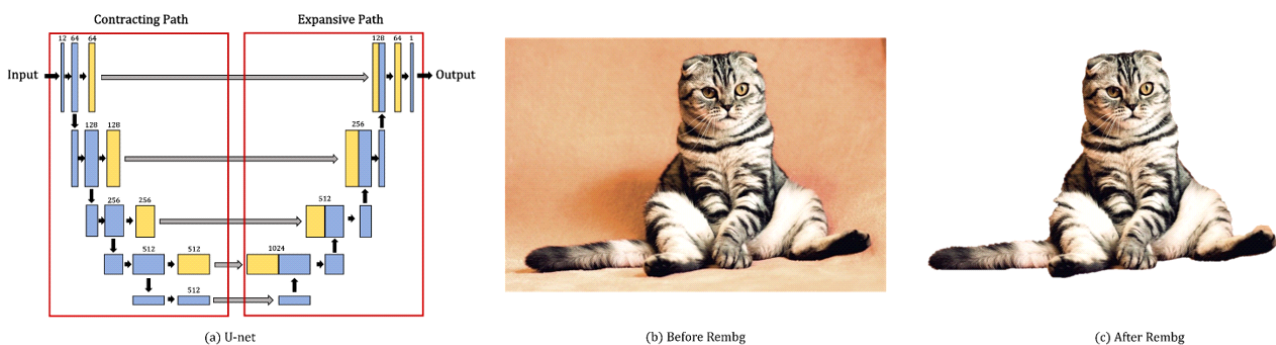


Figure 5. U2-Net based remove background(rembg) example

## 3) 이미지 특징 추출

이미지 데이터는 RGB의 3차원 정보를 저장한 픽셀로 이루어져 있다. 이미지의 특징 추출에 특화된 인공지능 모델인 합성곱 신경망은 이미지의 공간정보를 유지한 채 학습을 진행한다. 합성곱 층을 더 깊이 쌓아 향상된 성능을 제공하는 모델로, <Figure 6>와 같이 ResNet과 EfficientNet이 있다(He et al., 2016; Tan and Le, 2019).

ResNet은 모델의 깊이(depth)를 키울수록 과적합이 발생하는 문제를 Residual learning이라는 개념을 도입하며, 입력값과 출력값의 차이(Residual)를 학습에 이용해 늘어나는 레이어의 수에 따라 성능을 개선할 수 있게 하였다. EfficientNet은 최근 높은 성능을 보이는 모델 중 하나이며 모델의 깊이뿐만 아니라 너비(width), 해상도(resolution) 간 관계를 효율적으로 조절할 수 있는 Compounding scaling 방법을 제안하여 뛰어난 성능 향상을 이뤄내었다.

딥러닝 라이브러리인 PyTorch에서는 위의 ResNet 및 EfficientNet의 사전 학습 모델을 제공한다. 사전 학습 모델이란, Deng et al.(2009)의 ImageNet과 같은 방대한 양의 데이터를 이용해 학습을 사전에 완료한 모델이다. 따라서 사전 학습 모델을 활용할 경우, 정교한 특징 추출 모형을 처음부터 직접 구축하는 것보다 효율적으로 사용할 수 있다.

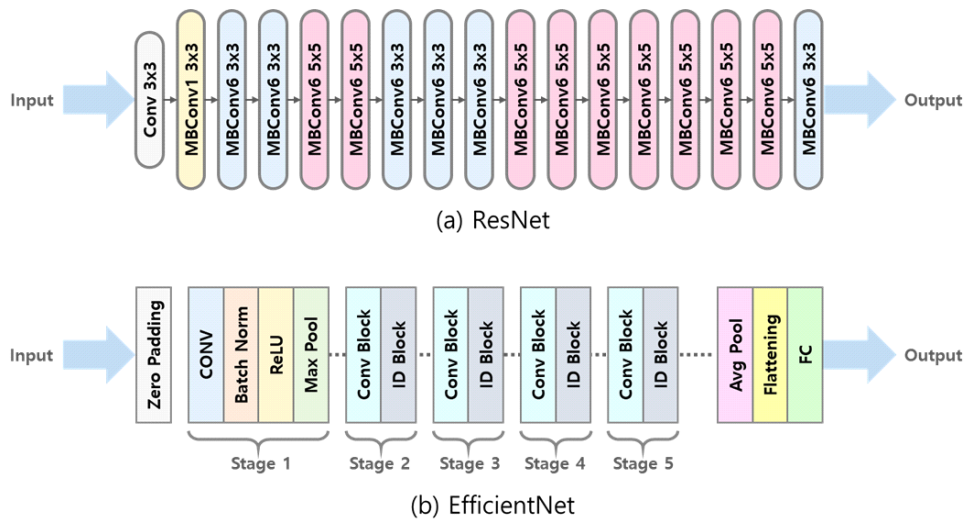


Figure 6. ResNet & EfficientNet Architecture

## 2.4 이미지 특징벡터 간 유사도 분석을 위한 지표

2.3의 이미지 특징 추출을 이용해 이미지 특징 벡터 간 유사도 분석을 위해선 인공지능 특징 추출 모델을 통과하여 나온 벡터들을 비교하는 유사도 비교 기법이 필요하다(Choi et al., 2010). 벡터 유사도 비교 기법에는 크게 두 가지 유형이 있다. 첫 번째는 벡터 간의 거리를 비교하는 유사도 기법으로 이에 유클리디언 유사도, 맨해튼 유사도, 민코스키 유사도가 해당한다. 두 번째로는 벡터 간의 각도를 비교하는 유사도 기법으로 이에 코사인 유사도가 해당한다.

## 1) 거리 기반 유사도 비교 기법

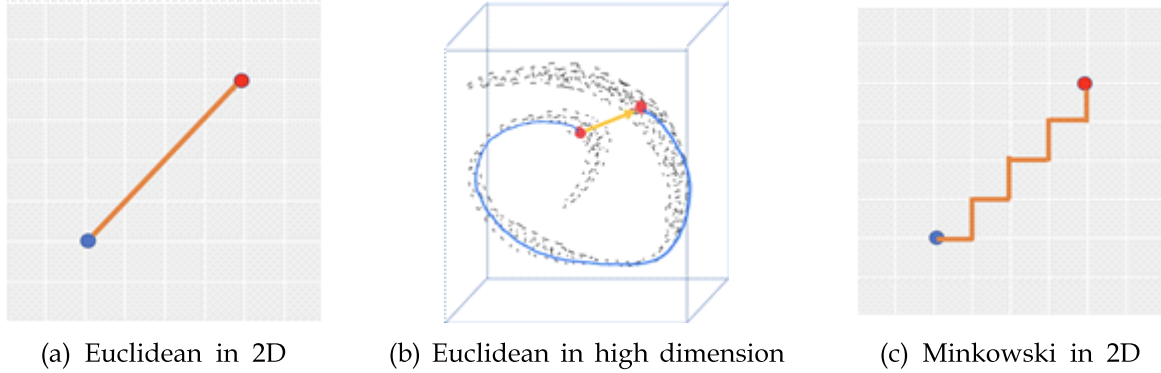


Figure 7. Visualization of distance based similarity matching metrics

벡터 간의 거리를 비교하는 유사도 비교 기법 중 대표적인 방법은 유클리디안 거리로 식 (1)과 같다. 결과 값이 작을수록 두 벡터 간의 거리가 가까운 것이므로 유사하다고 판단한다. 하지만, 유클리디안 거리는 벡터 차원에 대해 독립적이지 못하므로, 벡터 차원의 크기에 따라 거리의 계산값이 왜곡될 수 있다. 따라서 유클리디안 거리는 차원의 저주 이론에 따라 벡터의 차원이 커질수록 부정확한 유사도를 제공한다(Keogh and Mueen, 2017). <Figure 7> (a)와 같은 2차원 공간의 경우 두 점 간의 거리가 최단 거리로 계산되어 유사도가 정확하게 비교되는 것을 볼 수 있다. 하지만, 높은 차원을 시각화한 <Figure 7> (b)를 보면 실제 거리는 직선이 아닌 왜곡된 거리이다. 이처럼 유클리디안 거리는 높은 차원에서는 신뢰성이 떨어지기 쉽다. 따라서 유클리디안 거리는 낮은 차원에서 보다 신뢰성 있는 결과를 제공한다.

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

높은 차원의 벡터를 비교하는 경우 맨해튼 거리는 유클리디안 거리보다 더 신뢰 있는 유사도를 도출할 수 있다(Aggarwal et al., 2001). 맨해튼 거리 기법은 식 (2)와 같이 두 벡터 간의 1차원적인 정보를 비교하여 거리를 계산하는 기법이다. 유클리디안 거리와 유사하지만, 차이점은 2차원 정보를 사용하지 않고 1차원 정보를 사용한다는 것이다. 유클리디안 거리는 <Figure 7> (a)와 같이 체스판에서 대각선으로 움직여 거리를 계산한다면 맨해튼 거리는 <Figure 7> (c)와 같이 대각선으로는 움직이지 않고 거리를 계산한다. 따라서, 맨해튼 거리는 최단 거리를 계산하지 않기 때문에 유클리디안 거리와 달리 높은 차원에서 비교적 신뢰도 높은 유사도를 보일 수 있다.

$$D(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

민코스키 거리는 식 (3)과 같이 거리 기반 유사도를 매개변수에 따라 일반화한 유사도 기법이다. 식 3의 매개변수  $p$ 에 따라  $p$ 를 1로 설정하면 식 (1)과 같고, 2로 설정하면 식 (2)와 같다.  $p$ 를 큰 값으로 설정하여  $n$  차원의 거리를 계산할 수 있다. 하지만, Aggarwal et al.(2001)에서 소개된 것과 같이 높은 차원에서  $p$  값을 크게 설정할수록 정확한 거리를 얻기 힘들고 신뢰 있는 유사도를 얻기 힘들다. 이 경향은 벡터 간에 가장 멀리 떨어져 있는 차원만 보기 때문에 관찰된다. 고차원 공간에서 이는 가까운 벡터에 대해 의미 있는 거리 표현을 보장하지 못한다. 따라서 고차원에서 노름 공간을 바탕으로 하는 거리 계산 기법은 정확한 유사도를 도출하기 힘들다. Christian(2019)과 Hennig(2020)의 연구에서는 높은 차원에서  $p$  값이 1일 경우의 가능성에 더 주목해 볼 필요가 있음을 주장하였다.

$$D(x, y) = \sum_{i=1}^n (|x_i - y_i|^p)^{\frac{1}{p}} \quad (3)$$

## 2) 각도 기반 유사도 비교 기법

벡터 간의 각도를 비교하는 유사도 기법인 코사인 유사도는 식 (4)와 같다. 코사인 유사도는 유클리디안 거리의 단점인 차원의 저주를 보완하기 위해 사용할 수 있다. 코사인 유사도는 두 벡터 간의 코사인 각도를 계산하고 내적 값으로 정규화를 하여 두 벡터 간의 각도를 비교하는 유사도 기법이다. 각도 계산의 결과값은 -1부터 1의 값을 출력하고 1에 가까울수록 서로 유사한 벡터이며 -1에 가까우면 서로 유사하지 않은 벡터라 판단한다. 위의 유클리디안 거리와 달리 코사인 유사도는 벡터의 크기에 대한 값은 정규화하므로, 만약 벡터 간의 크기가 중요한 특징이라면 코사인 유사도는 적용이 부적절하다. 즉, <Figure 8> (b)와 같이 벡터 간의 방향성이 유사하지만 크기가 다르다면 잘못된 결과를 도출할 수 있다.

$$D(x, y) = \cos \theta = \frac{x \cdot y}{|x||y|} \quad (4)$$

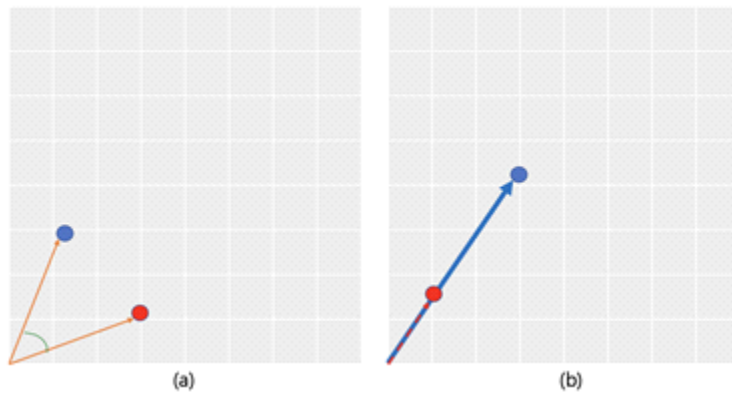


Figure 8. Cosine similarity

하지만, 코사인 유사도 기법은 두 벡터 간의 각도를 바탕으로 계산하기 때문에 차원의 저주를 받지 않아 높은 차원에서 정확한 정보를 도출할 수 있다는 장점이 있다. 특히 이미지 특징 벡터와 같이 차원이 높지만, 벡터들의 크기 정보가 중요하지 않은 경우 코사인 유사도 기법이 사용 가능하다.



### 3. 이미지 특징 유사도 기반 상품 식별 시스템

#### 3.1 시스템 개요

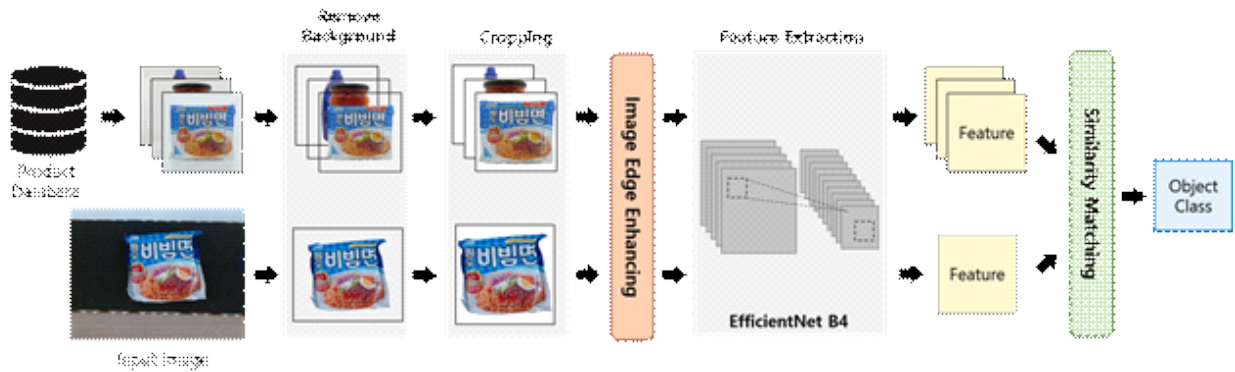


Figure 9. System architecture

본 연구의 시스템 구조는 <Figure 9>와 같다. 먼저 상품 데이터베이스 내 모든 물체의 사진을 데이터베이스에서 순차적으로 열어 3차원 배열화한다. 그리고 U2-Net 기반 배경 제거 모델을 활용해 이미지들의 배경을 제거한 뒤 흰 배경으로 채우는 이미지 전처리를 수행한다. 그리고 카메라를 통해 찍은 이미지를 동일하게 배경 제거한 뒤, 이미지 윤곽선 강화(Image Edge Enhancing)를 거쳐 윤곽선이 두드러지도록 전처리를 적용한다. 전처리를 거친 각 이미지를 특징 추출기인 EfficientNet B4를 거쳐 크기 1,792인 특징을 추출한다. 그리고 유사도 매칭 방법인 코사인 유사도를 통해 입력 상품의 특징과 상품 데이터베이스의 상품별 특징과 유사도를 구한 뒤, 가장 높은 유사도의 상품 번호를 찾아낸다.

#### 3.2 이미지 전처리

상품 데이터베이스 이미지 및 입력 이미지에서 상품의 특징을 추출하기 이전, 이미지 내에서 상품에 보다 집중하기 위한 전처리 과정을 거친다. 먼저 2.3.2절에서 소개한 배경 제거 모델인 Rembg를 사용해 원본 이미지에서 배경을 제거하고 상품 객체만을 얻기 위해 잘라내기(cropping)를 수행한다. 배경 제거 시엔 배경을 투명하게 표현하기 위해 투명도(alpha) 값을 포함하는 RGBA 이미지로 변환된다. RGBA 이미지는 이미지 윤곽선 강화를 적용하기에 용이하지 않으므로, 투명한 픽셀은 흰색 값(255, 255, 255)으로 채우며 RGB 이미지로 다시 변환한다. 다음으로 잘라낸 이미지에 대표적인 윤곽선 강화 처리 기법인 Embossing과 Sharpening을 <Figure 10>과 같이 적용한다. 또한 비교실험을 위해 RandomBrightnessContrast도 함께 적용해본다. 이미지 전처리 과정을 모두 거친 이미지는 크기를 600 x 600으로 조정한 뒤 사전학습된 특징 추출 모델인 EfficientNet B4에 입력해 특징을 추출한다.



Figure 10. Image preprocessing type of our system

### 3.3 특징 유사도 측정

본 연구에서는 총 4가지 특징 벡터 유사도 측정 기법을 적용한다. EfficientNet B4를 통해 추출된 특징 벡터의 크기는 1,792로, 차원이 크다. 2.4장에서 소개한 항목들 중, 높은 차원에 적절한 유사도 기법인 코사인 유사도와 맨해튼 거리 유사도를 우선 적용한다. 또한 비교실험을 위해 유클리디안 거리와 민코스키 거리 지표도 함께 적용해 본다. 이때 민코스키의 경우 매개변수  $p$ 를 3으로 설정한다.

## 4. 시스템 도입을 위한 실험

### 4.1 실험 데이터 수집 환경

실험 데이터 수집을 위한 촬영은 물류센터를 모사한 <Figure 11>의 실험 환경에서 진행하였다. 환경은 상품이 혼재되어 있지 않고, 개별 상품이 분리되어 컨베이어 벨트를 통해 이동하는 특정 상황임을 가정 하였다. 촬영엔 컨베이어벨트 위에 거치대를 통해 바닥면과 수평으로 고정된 로지텍 C920 카메라를 사용하였으며, 생활 물체가 모두 화면에 담기도록 카메라 높이를 40cm로 고정해 주었다.



Figure 11. Experimental environment for data collection

## 4.2 실험 데이터 구성

Table 1. Data statistics for AIHub product image dataset

Category	Amount	Heights(3) + Angles(24)	
		Singular	Plural
과자	1,693	121,896	121,896
디저트	77	5,544	5,544
면류	208	14,976	14,976
상온HMR	1,093	78,696	78,696
생활용품	1,112	80,064	80,064
소스	736	52,992	52,992
유제품	291	20,952	20,952
음료	1,130	81,360	81,360
의약외품	203	14,616	14,616
이/미용	2,019	145,368	145,368
주류	496	35,712	35,712
커피차	508	36,576	36,576
통조림/안주	322	23,184	23,184
홈클린	392	28,224	28,224
<b>Total</b>	<b>10,280</b>	<b>740,160</b>	<b>740,160</b>



실험을 위한 상품 데이터베이스는 한국지능정보사회진흥원이 운영하는 AI 통합 플랫폼인 AIHub의 ‘상품 이미지’ 데이터 셋(<Table 1>)을 활용하여 구축하였다. 상품 이미지 데이터는 10,280개의 상품을 하나씩 촬영한 단수 상품 이미지와 2개 묶음을 촬영한 복수 상품 이미지로 나누어 제공한다. 한 상품에 대해 다양한 구조에서 촬영하였으며 3가지의 높이(0도, 30도, 60도)와 15도씩 24번의 회전된 이미지를 제공한다. 이처럼 상품 한 개에 144개, 전체 상품에 대해 총 약 144만 장의 이미지를 제공한다. 제공되는 이미지 예시는 <Figure 12>과 같다.

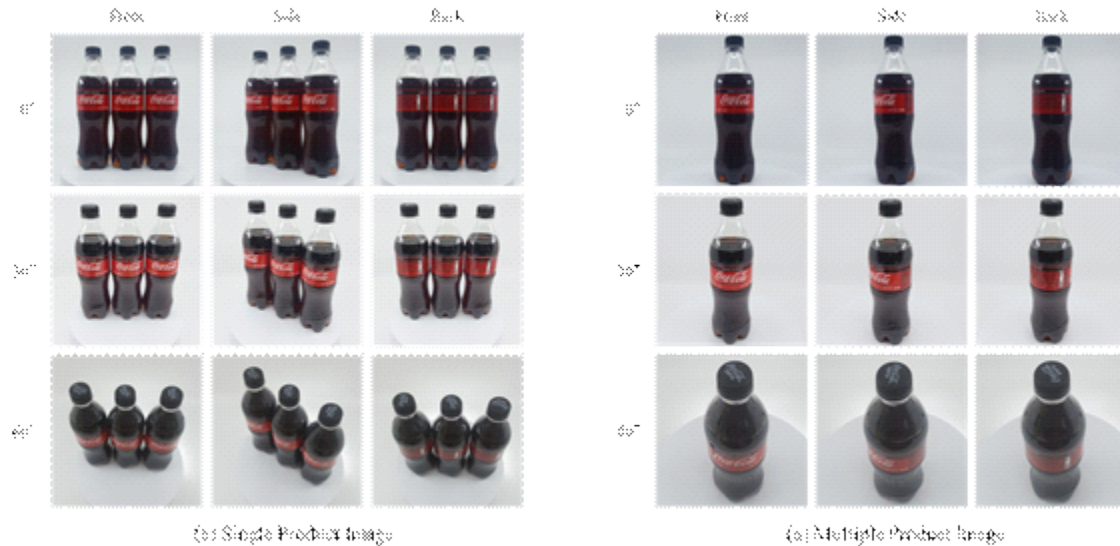


Figure 12. Example of AIHub product image dataset

본 논문에선 위의 데이터 셋에서 단수 상품 이미지만을 사용하였으며 상품당 0도 높이에서 앞면과 뒷면을 찍은 이미지 2장만을 사용하여 자체 상품 데이터베이스를 구축하였다. 이는 총 14개 카테고리의 6,106개 상품에 대해 상품당 앞, 뒷면 이미지를 가져와 총 12,212장을 저장하였다. AIHub의 상품 이미지 데이터 셋은 사람이 수작업으로 제작하여 이미지가 깨진 채 저장된 오류가 있거나 파일명이 일관되지 않아 앞, 뒷면 이미지가 존재하지 않는 상품이 다수 존재하였다. 따라서 일관된 상품 데이터베이스 구축을 위해 이처럼 오류가 발생할 수 있는 상품은 제외하여 저장해 주었다.



Figure 13. Product database structure

상품 데이터베이스(Product DB) 구축을 위한 구조는 <Figure 13>과 같다. 먼저 이미지 저장 시에 0번부터 순차적으로 숫자를 이름으로 붙여주어 기존의 복잡한 폴더 구조를 단순화하였다. 파일명(id)과 함께 앞서 데이터를 선별하며 파싱해온 정보인 상품 이름(name), 상품 고유번호(product\_id), 이미지의 앞/뒷면 여부(fb), 카테고리(category) 등의 메타 데이터를 스프레드시트 파일에 함께 저장하였다. 최종적으로 생성된 데이터베이스의 구조와 데이터베이스의 메타정보를 저장한 메타 데이터 구조는 <Table 2>와 같다.

Table 2. Structure of meta data for product database (data.csv)

id	name	product_id	fb	category
0	오뚜기도나스믹스500G	10232	0	소스
1	오뚜기도나스믹스500G	10232	1	소스
2	청정원12나트륨솔트	10243	0	소스
3	청정원12나트륨솔트	10243	1	소스
4	대상청정원명란크림파스타소스350G	10293	0	소스
5	대상청정원명란크림파스타소스350G	10293	1	소스
6	폰타나카르니아베이컨앤머쉬룸크림430G	15329	0	소스
7	폰타나카르니아베이컨앤머쉬룸크림430G	15329	1	소스
8	청정원)연탄불고기양념140G	25214	0	소스
9	청정원)연탄불고기양념140G	25214	1	소스
⋮	⋮	⋮	⋮	⋮

또한 실시간으로 물류 환경에서 식별이 필요한 상품 데이터 셋은 다음과 같이 구축하였다. 먼저, 데이터베이스에 있는 6,106개의 상품 중 시중에서 쉽게 구할 수 있는 상품을 50개 선정하였다. 그리고 각 상품을 실험 환경의 컨베이어벨트에 올려 앞면과 뒷면 이미지를 직접 촬영하여 100개의 테스트 이미지 데이터 셋을 구축하였다. 촬영 시 각 물체 당 이미지 촬영 시 3초 이내로 제한을 두어, 최대한 실제 산업 현장과 비

숫하게 정제되지 않은 이미지 데이터를 구축하도록 하였다. 테스트 이미지는 800x600 해상도로 저장하였다. 최종적으로 생성된 테스트용 이미지 예시는 <Figure 14>와 같고 상품 정보를 저장한 메타 데이터 구조는 <Table 3>과 같다.

Table 3. Structure of meta data for test product dataset (test\_label.csv)

id	name	product_id
0	롯데)제크오리지날100G	30100
1	롯데)제크오리지날100G	30100
2	영남코프레이션레이저레몬주스(노란통)	65349
3	영남코프레이션레이저레몬주스(노란통)	65349
4	오뚜기스위트콘340G	15037
5	오뚜기스위트콘340G	15037
6	오뚜기옛날볶음참깨100G	20298
7	오뚜기옛날볶음참깨100G	20298
8	금강깨끗한표백제(바르는타입)	80213
9	금강깨끗한표백제(바르는타입)	80213
⋮	⋮	⋮



Figure 14. Example of image in test dataset

### 4.3 실험 결과 및 분석

#### 1) 이미지 윤곽선 강화, 밝기 조절 및 특징 추출 모델에 따른 비교 실험 결과

Table 4. Experimental results by image edging enhancing filter

	ResNet		ResNet + EfficientNet		EfficientNet (Ours)	
	Top-1 Acc.	Top-3 Acc.	Top-1 Acc.	Top-3 Acc.	Top-1 Acc.	Top-3 Acc.
No Filter	51%	71%	60%	<b>78%</b>	53%	77%
Emboss Filter	<b>59%</b>	<b>75%</b>	54%	74%	58%	74%
Sharpen Filter	52%	69%	<b>62%</b>	77%	54%	72%
Random Brightness Contrast Filter	43%	62%	59%	75%	<b>60%</b>	76%
<b>Emboss + Sharpen Filter (Ours)</b>	53%	66%	54%	72%	58%	<b>80%</b>

<Table 4>는 이미지 윤곽선 강화 및 이미지 특징 추출 모델의 비교실험 결과이다. 본 연구의 시스템과의 비교를 위한 특징 추출 모델은 크게 ResNet과 ResNet + EfficientNet 두 개의 모델을 동시에 사용하는 앙상블 모델로 선정하여 비교실험을 하였다. 사용한 성능 지표는 Top-1 Accuracy와 Top-3 Accuracy로, Top-1 Accuracy는 100장의 테스트 이미지에 대해 특징 유사도 기법을 이용해 상품 이미지 데이터베이스 내 상품을 검색했을 때, 가장 유사한 상품으로 식별되는 상품이 실제 테스트 이미지의 상품과 일치하는 정도를 수치화 한 것이고, Top-3 Accuracy는 가장 유사한 상품으로 판단되는 3가지 상품 중에 실제 상품이 존재하는 지를 수치화 한 것이다. 먼저 본 논문의 시스템에서 사용한 특징 추출 모델인 EfficientNet에서, 이미지 강화 처리하지 않은 경우보다 Emboss와 Sharpen을 모두 적용한 경우 더 높은 성능을 보였다. 또한 이미지 강화 처리를 적용한 경우, 다른 특징 추출 비교모델보다 EfficientNet이 가장 높은 성능을 보였다.

ResNet과 EfficientNet의 특징을 연결한 앙상블 모델에서는 전반적으로 좋은 성능을 보였으나, 윤곽선 강화를 적용한 EfficientNet에 비해 높은 성능을 보이진 않았다. 이미지 전처리와 특징 추출모델을 복합적으로 분석해 보면, 일반적으로 ResNet의 경우 색 정보를, EfficientNet의 경우 윤곽선 정보를 중시하는 경향을 보였다. 동일한 테스트 이미지에 대해서 <Figure 15>과 같이 전자의 경우 색이 유사한 물체를, 후자의 경우 모양이 비슷한 물체를 우선하여 선택하는 경향을 보임을 확인할 수 있다.



Figure 15. Example of results by ResNet and EfficientNet

## 2) 이미지 특징 유사도 비교 지표에 따른 결과

Table 5. Experimental results by similarity comparison metric

Emboss + Sharpen Filter (Ours)	ResNet		ResNet + EfficientNet		EfficientNet (Ours)	
	Top-1 Acc.	Top-3 Acc.	Top-1 Acc.	Top-3 Acc.	Top-1 Acc.	Top-3 Acc.
Cosine Similarity (Ours)	<b>53%</b>	<b>66%</b>	51%	<b>72%</b>	<b>58%</b>	<b>80%</b>
Euclidean Distance	48%	60%	52%	71%	56%	77%
Manhattan Distance	50%	61%	<b>54%</b>	71%	55%	72%
Minkowski Distance	20%	30%	32%	46%	36%	51%

이미지 특징 유사도 비교 지표에 따른 실험 결과는 <Table 5> 와 같다. 먼저 대부분의 수치에서 각도 유사도 기법인 코사인 유사도가 거리 유사도 기법들에 비해 높은 정확도를 보여주었다. 그리고 유클리디안 거리와 맨해튼 거리가 비슷한 수치로 그다음의 성능을 보였다. 본 연구의 특징 추출모델을 통해 얻은 특징 벡터들은 1,000 이상의 고차원이므로 코사인 유사도가 가장 높은 정확도를 보여주는 것으로 분석된다. 반면, Christian(2019)의 연구를 근거로 거리 기반 유사도 기법 중 맨해튼 거리 방법이 가장 높은 정확도를 보일 것으로 추측하였지만, 유클리디안 거리와 유사하거나 오히려 좋지 못한 결과를 보였다. 민코프스키 유사도는 매개변수  $p$ 가 3일 경우 Hennig(2020)의 연구에서와 같이 좋지 못한 정확도를 보여주었다.

## 3) 이미지 해쉬 검색 방법과의 비교 실험

Table 6. Comparison experiment with Image hash

Model	Top-1 Acc.	Top-3 Acc.
Ours	62%	80%
Average Hash	0%	0%
Perceptual Hash	0%	3%
Differential Hash	1%	1%

<Table 6>은 본 논문의 시스템과 이미지 검색 기법인 이미지 해쉬와의 비교실험 결과이다. Ours는 <Table 4>의 Top-1 정확도와 Top-3 정확도의 최댓값을 기준으로 작성하였다. 이미지 해쉬의 경우 전반적으로 상당히 낮은 정확도를 보였다. Top-3 정확도에 대해 Perceptual hash 사용 시 3%를, Differential hash 사용 시 1%를 보였다. 이는 테스트 이미지가 상품 데이터베이스의 이미지와 비교하여 회전되었거나, 초점이 맞지 않아 지문 문자열을 유사하게 찾기 어려웠을 것이라 분석된다. 또한 이미지 해싱 기술 적용을 위해 이미지 해상도를 낮게 조정하여 특징 정보가 많이 소실되고, 회색조 이미지로 색 단순화 변환을 거치며 색 정보가 고려되지 않아 더욱 유사 패턴을 찾기 어려웠을 것으로 분석된다. 따라서 물류환경과 같은 동적인 환경에선 본 논문의 이미지 특징 유사도 비교 방법이 더욱 적합함을 확인하였다.



	Test Image	Image Preprocessing	Top 1	Top 2	Top 3
(a) Light Reflection					
(b) Out of Focus					
(c) Similar Object					
(d) Angular Distortion					
(e) Hard to Recognize - Success					
(f) Hard to Recognize - Failure					
(g) Different Packaging Shapes					

Figure 16. Experimental result for test images

#### 4) 테스트 이미지의 상품 식별 결과 분석

<Figure 16>은 테스트 상품의 다양한 상황에 따른 상품 식별 결과 예시이다. <Figure 16> (a) 상품의 경우 과자 봉지 재질의 특성상 빛 반사가 잘 발생하였다. 빛 반사로 일부 픽셀이 소실되지만, 특징적인 로고나 색감 등이 뚜렷하여 식별에 성공하였다. <Figure 16> (b)는 움직이는 물체를 촬영해 초점이 정확히 맞지

않았다. 이미지의 특징이 왜곡되기 쉬움에도 형태와 주요 색감 특징이 드러나 성공적으로 식별되었다. <Figure 16> (c)는 같은 회사의 같은 상호인데 디자인이 달라진 경우이다. 이에 따라 동일 상품을 2번째로 식별하였지만, 가장 높은 유사도의 상품도 동일 상품군으로 식별하였다. 이는 사람과 유사한 방식으로 물체를 찾는 것을 보여주었다. 또한 투명한 용기 재질로 인해 내부 물체가 차 있는 정도가 미세하게 다른 것도 변수로 작용했을 것으로 보인다. <Figure 16> (d)는 상품의 방향이 기울어졌지만, 상품 데이터베이스 내의 정방향 이미지를 잘 식별하였다. 이는 이미지 특징 추출 시 합성곱 연산이 다양한 방향에서 특징을 추출하기에 동적 환경에서 적응적인 활용에 용이함을 보여준다. <Figure 16> (e)는 상품이 이동하며 찍히는 와중에 초점이 맞지 않았고 상품명 및 로고가 보이지 않아 쉽게 구분하기 어려운 경우이다. 하지만 상품 데이터베이스에서 해당 상품의 옆면 디자인이 테스트 이미지와 유사하며 용기의 모양이 특징적이었기 때문에 잘 식별된 것으로 분석된다. 이를 통해 상품의 이미지를 다양한 구도에서 촬영하여 상품 데이터베이스를 구축하면 연산량이 증가하지만 보다 동적인 환경의 상품에 대해 강건한 식별을 수행할 수 있을 것으로 분석된다. <Figure 16> (f)는 앞면에 상품의 주요 정보가 집중되어 있지만 뒷면이 입력되어 마치 사람도 정확한 식별을 수행하기 어려운 경우이다. 이처럼 상품의 특징이 매우 적게 드러나는 경우엔 식별의 한계가 있어 추가적인 메타정보와 함께 고려되는 것이 필요해 보인다. <Figure 16> (g)는 유사 상품 간 혼동되기 쉬운 상황에서 동일한 재질 및 형태의 포장을 상호보다 더 높은 유사도로 탐지한 결과이다. 사람의 경우 상호와 내부 내용물의 유사성을 우선으로 판단하나 모델은 미미하게 포장 형태 및 포장 재질을 우선으로 탐지하였다.

본 연구의 시스템은 색 정보와 윤곽선 정보를 복합적으로 고려해 상품을 인식한다. 그런데, 위의 <Figure 16> (e)에서 분석했듯, 실험 결과에서는 색 정보보다 윤곽선(모양) 정보를 더 중시하여 상품을 식별하는 경향을 볼 수 있다. 이는 <Figure 16> (b), <Figure 16> (e) 등 다른 제품들에서도 유사한 경향성을 확인할 수 있다. <Figure 16> (b)의 상위 1 유사도는 색과 모양이 동일하여 식별하였지만 뒤이어 오는 상위 2, 3 유사도의 상품들은 색은 다르지만, 모양이 비슷한 제품들로 탐지하였다. 이는 이미지 윤곽선 강화 적용으로 상품의 모양 정보가 두드러지게 강조되었기 때문으로 분석된다.

## 5. 결론

본 논문에선 학습 및 라벨링이 필요하지 않고 물체의 앞뒷면 사진을 사용해 데이터베이스 내 이미지와 특징 비교를 하여 상품을 식별해내는 알고리즘을 제안하였다. 제안한 알고리즘은 배경 제거 및 이미지 전처리를 거쳐 특징을 추출한 뒤 특징 간 유사도를 비교하는 방식으로 이미지 검색알고리즘 중 하나인 이미지 해쉬와 비교해 더 우수한 성능을 보였다. 그뿐만 아니라 초점이 잘 맞추어지지 않았거나, 빛 반사가 있는 경우 등 촬영 환경이 동적인 상황에도 상품을 잘 식별해냄을 결과로 확인하였다.

하지만 상품이 특정 부분이 가려져 있지 않고 모두 보여야 하는 시스템의 제약이 있어, 상품이 겹쳐 있거나 혼재된 상황이 아닌 개별상품이 분리되어 촬영된 특정상황에서만 실험을 적용한 한계점이 있다. 또한 실험 결과에서 보였듯, 일부 유사품이 있거나 같은 표지의 상품임에도 다른 용량이거나 디자인이 바뀐 경우에는 잘 잡아내지 못한 한계점도 있었다. 이의 경우엔 물체의 크기 및 무게와 같은 메타 정보를 함께 고려하면 성능이 향상될 것으로 보인다.

본 연구의 기술은 물류센터, 마트, 제조업 등에서 사용 가능할 것으로 기대되며 로봇팔을 통한 동적 환경에서의 자율 작업뿐만 아니라 상품 분류 및 검수 작업에도 활용 가능할 것이다. 향후 후속 연구에선 가로, 세로, 높이 정보, 무게 정보 등을 함께 수집해 유사도 분석 시 함께 고려하여 성능을 향상하고, 이를 실제 로봇의 픽-앤-플레이스 작업에 적용하여 작업 수행 효율을 최적화해보려 한다.

## 참고문헌

- Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001), On the surprising behavior of distance metrics in high dimensional space, In International conference on database theory ., pp, 420~434, Springer, Berlin, Heidelberg.
- Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. (2020), Albumentations: fast and flexible image augmentations. Information, 11(2), 125.
- Chapelle, O., Haffner, P., and Vapnik, V. N. (1999), Support vector machines for histogram-based image classification, IEEE transactions on Neural Networks, 10(5), 1055-1064.
- Choi, S. S., Cha, S. H., and Tappert, C. C. (2010), A survey of binary similarity and distance measures, Journal of systemics, cybernetics and informatics, 8(1), 43-48.
- Cunningham, P., Cord, M., and Delany, S. J. (2008), Supervised learning, In Machine learning techniques for multimedia., pp, 21~49, Springer, Berlin, Heidelberg.
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009), Imagenet: A large-scale hierarchical image database, In 2009 IEEE conference on computer vision and pattern recognition., pp, 248~255, Ieee.
- Grover, A., Braeckel, P., Lindgren, K., Berghel, H., and Cobb, D. (2010), Parameters effecting 2D barcode scanning reliability, In Advances in Computers ., Vol, 80, pp, 209~235, Elsevier.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016), Deep residual learning for image recognition, In Proceedings of the IEEE conference on computer vision and pattern recognition., pp, 770~778.
- Hennig, C. (2020), Minkowski Distances and Standardisation for Clustering and Classification on High-Dimensional Data, In Advanced Studies in Behaviormetrics and Data Science ., pp, 103~118, Springer, Singapore.
- Jia, W., Zhang, H., He, X., and Wu, Q. (2006), A comparison on histogram based image matching methods, In Proceedings-IEEE International Conference on Video and Signal Based Surveillance 2006, AVSS 2006.
- Karabegović, I., Karabegović, E., Mahmić, M., and Husak, E. (2015), The application of service robots for logistics in manufacturing processes, Advances in Production Engineering & Management, 10(4).
- Keogh, E. J., and Mueen, A. (2017), Curse of dimensionality, Encyclopedia of machine learning and data mining, 314-315.
- Li, S., Kang, X., Fang, L., Hu, J., and Yin, H. (2017), Pixel-level image fusion: A survey of the state of the art, information Fusion, 33, 100-112.
- Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., and Jagersand, M. (2020), U2-Net: Going deeper with nested U-structure for salient object detection, Pattern recognition, 106, 107404.
- Rennie, C., Shome, R., Bekris, K. E., and De Souza, A. F. (2016), A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place, IEEE Robotics and Automation Letters, 1(2), 1179-1185.



- Ronneberger, O., Fischer, P., and Brox, T. (2015), U-net: Convolutional networks for biomedical image segmentation, In International Conference on Medical image computing and computer-assisted intervention., pp. 234~241, Springer, Cham.
- Sarabu, H., Ahlin, K., and Hu, A. P. (2019), Leveraging deep learning and rgb-d cameras for cooperative apple-picking robot arms. In 2019 ASABE Annual International Meeting ., pp. 1, American Society of Agricultural and Biological Engineers.
- Shorten, C., and Khoshgoftaar, T. M. (2019), A survey on image data augmentation for deep learning, Journal of big data, 6(1), 1-48.
- Singh, K., and Kapoor, R. (2014), Image enhancement using exposure based sub image histogram equalization, Pattern Recognition Letters, 36, 10-14.
- Singhdong, P., Suthiwartnarueput, K., and Pornchaiwiseskul, P. (2021), Factors Influencing Digital Transformation of Logistics Service Providers: A Case Study in Thailand, The Journal of Asian Finance, Economics and Business, 8(5), 241-251.
- Tan, M., and Le, Q. (2019), Efficientnet: Rethinking model scaling for convolutional neural networks, In International conference on machine learning., pp. 6105~6114, PMLR.
- Wang, X., Pang, K., Zhou, X., Zhou, Y., Li, L., and Xue, J. (2015), A visual model-based perceptual image hash for content authentication, IEEE Transactions on Information Forensics and Security, 10(7), 1336-1349.
- Zaitoun, N. M., and Aqel, M. J. (2015), Survey on image segmentation techniques, Procedia Computer Science, 65, 797-806.
- Zauner, C. (2010), Implementation and benchmarking of perceptual image hash functions.
- Zhuang, C., Wang, Z., Zhao, H., and Ding, H. (2021), Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking. Robotics and Computer-Integrate d Manufacturing, 68, 102086.

# 횡단보도 보행자 인지 향상을 위한 ByteTrack 기반 다중객체추적

노유진, 이유진, 이상민<sup>†</sup>  
광운대학교 소프트웨어융합대학

## ByteTrack-Based Multi-Object Tracking for Recognition of Pedestrian in Crosswalk

Yujin Roh, Yujin Lee, Sangmin Lee<sup>†</sup>  
College of Software and Convergence, Kwangwoon University

According to traffic accident statistics, 42% of traffic accidents between 2017 and 2021 were crossing accidents. To reduce these accidents, the research on the collision avoidance assistance for pedestrian safety is in the spotlight in machine learning field. However, there are few studies to improve the accuracy of recognizing the crossing multiple pedestrians and estimating the residual time to the crossing. We here propose using hybrid approach combining Yolo-X and ByteTrack to obtain the accurate detector of pedestrians. We gathered the actual data set of the pedestrian crossing for the accurate multi-pedestrian tracking model. The proposed approach can detect and track the crossing pedestrians in real time. Furthermore, we can estimate the pedestrian's walking direction, speed, acceleration, and distance to the crosswalk for inferring the motion of pedestrians entering the crosswalk. The experimental results present that the proposed approach outperforms the alternatives in terms of tracking accuracy and computation cost.

**Keywords:** Pedestrian Recognition, Multi-Object Tracking, Autonomous Vehicle, ByteTrack, CDNet

---

논문접수일 : 2022.10.08.

심사완료일 : 2022.12.26.

게재확정일 : 2022.12.27.

<sup>†</sup> Corresponding Author: smlee5679@gmail.com

## 1. 서론

도시 물류에서 자동화 물류 개념을 도입할 경우 물류시스템의 효율성 못지않게 중요한 가치로 보행자의 안전과 편의 증진을 들 수 있다. 이와 더불어 자율주행의 발전은 물류산업 영역에 적용되어 물류시스템의 효율성과 안전성 연구에 기여하고 있다.(김도영 et al., 2018; 김태선 et al., 2021; 박상미와 김정우, 2020; 웨이치 et al., 2018). 교통사고 분석시스템에서 발표한 최근 5개년 통계 보고에 따르면 2017년부터 2021년까지 전체 차대 사람 간의 교통사고 209,616건 중 횡단 중 보행자 충돌사고 건은 87,786건으로 42%를 차지했다(Traffic Accident Analysis System., 2021). 그중 사망자 수는 3,385명, 부상자 수는 90,045명으로 횡단 중인 보행자에 대해 대응할 수 있는 충돌방지보조(collision avoidance assistance: CAA) 시스템에 관한 연구가 필요하다. 이러한 안전사고를 방지하기 위해 차량에는 다양한 센서들이 장착되고 있으며, CAA 시스템에 관한 연구가 활발하게 진행되고 있다(Akhauri et al., 2020; Althoff et al., 2009; Cho et al., 2019; Chu et al., 2012; Grigorescu et al., 2020; Noh and Han, 2014; Park et al., 2018; Shalev-Shwartz et al., 2016). 최근 각광받고 있는 자율주행 자동차에서는 이러한 CAA 시스템이 필수적이며 카메라 센서, 레이더(radio detection and ranging: RADAR) 센서, 라이다(light detection and ranging: LIDAR) 센서 등을 이용하고 있다. 하지만 RADAR는 작은 크기의 물체 식별이 어려우며, 또한 LIDAR는 원거리 측정이 어렵다는 문제점을 지닌다(Lee et al., 2021; Jin et al., 2018). 많은 CAA 시스템의 발전에도 불구하고 RADAR, LIDAR 센서에 의존적인 경향이 있는 CAA 시스템은 두 센서의 문제점을 가지게 된다. 이 문제점을 해결하기 위해 본 논문에서는 작은 크기의 물체 식별과 종합적인 상황 인식에 적합한 카메라 센서를 이용하여 다중객체 추적(multi object tracking: MOT) 기반으로 두 문제점을 해결하고자 한다.

본 논문은 횡단보도와 횡단보도 인근 보행자를 탐지하여 1) 횡단하려는 다중 보행자를 정확하게 인식하고, 2) 다중 보행자 중 횡단보도까지의 잔여 시간을 추정하는 알고리즘을 개발한다. 자율주행차량의 운행에 있어 이 두 가지 문제를 풀어야 교차로 상황에서 갑자기 뛰어들거나 RADAR 센서 사각지대에 놓여진 보행자에 대한 인식율을 높일 수 있다. 특히, 정확도 성능을 높이기 위해서 제안 기법은 객체 탐지 부문과 객체 추적 부문에 있어서 가장 성능이 높게 나온 두 가지 최신 기법을 활용하였다. 횡단보도 인식에는 도로 교통 객체탐지에 특화된 CDNet(crosswalk detection network) 모델(Zhang Z, et al., 2022)을 활용하고, 인근 보행자의 움직임을 정확히 추적하기 위해 MOT에 특화된 ByteTrack(Zhang Y, et al., 2021)을 학습시켜 활용한다. 제안하는 알고리즘은 2단계로 구성된다. 첫째, 카메라 센서로부터 수집된 데이터인 동영상 데이터를 인공지능 알고리즘이 학습하기 위해 전처리하고 MOT 알고리즘인 ByteTrack을 통해 보행자를 추적한다. 둘째, 보행자의 진행방향, 속도, 가속도, 보행자와 횡단보도 간 거리를 계산한다. 추정된 수치를 이용해서 보행자의 진행방향이 횡단보도를 향하고, 횡단보도까지 남은 거리가 계속 줄어들며 가속도가 감소하지 않는다면 횡단할 가능성이 있는 보행자로 추정한다. 이후 탐지된 보행자 중 가장 먼저 횡단보도에 도착할 보행자를 탐지한 후 해당 보행자의 보행 속도를 반영하여 횡단보도까지의 도착시간을 예측한다. 본 논문에서 제안하는 알고리즘을 통해 보행자의 횡단보도 진입시간을 유추할 수 있으며, 이는 횡단 중 차량과 보행자 간의 충돌사고 빈도를 최소화와 더불어 보행자의 충돌에 대응하는 자율주행 CAA 시스템의 기반기술이 되리라 기대한다.

본 논문의 구성은 다음과 같다. 2장은 본 연구의 관련 연구 및 선행 연구들에 대해 기술하였으며, 3장은 본 연구를 수행하기 위한 알고리즘에 대해 기술한다. 4장은 제안하는 알고리즘의 성능지표와 연구 결과에 대해 기술하고, 5장에서는 연구 결과에 대한 분석과 그 의미에 대해 기술한다. 마지막으로 6장에서 결론 및 향후 연구과제에 대해 기술한다.

## 2. 선행연구

Figure 1에서 볼 수 있듯 2017년부터 2021년까지 보행자 횡단 중 차량에 의한 충돌사고가 매해 약 20,000 건가량 발생하고 있다. 또한, 매년 약 15,000명 이상이 횡단 중 교통 사상사고를 당하고 있다(Traffic Accident Analysis System., 2021). 이러한 교통사고를 줄이기 위해 횡단 중인 보행자에 대해 대응할 수 있는 충돌방지보조(collision avoidance assistance: CAA) 시스템에 관한 연구가 활발히 이루어지고 있으며, CAA 시스템은 차량에 부착된 여러 센서 데이터들을 수집하고 처리하여 교통사고와 같은 충돌을 미연에 방지하도록 보조한다(Akhauri et al., 2020; Althoff et al., 2009; Cho et al., 2019; Chu et al., 2012; Grigorescu et al., 2020; Noh and Han, 2014; Park et al., 2018; Shalev-Shwartz et al., 2016).

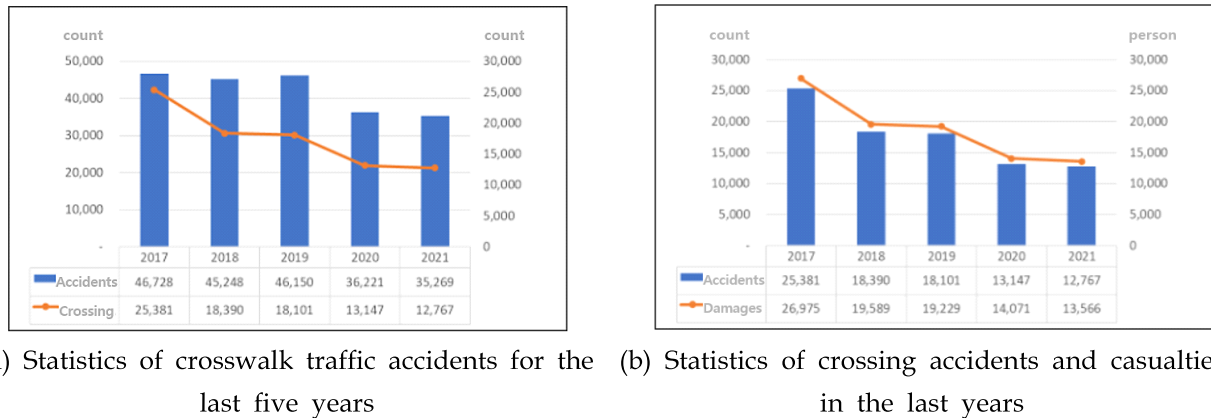


Figure 1. Statistics of traffic accidents in South Korea between 2017 and 2021  
(Traffic Accident Analysis System, 2022)

최근에는 정보기술의 발달과 센서들의 정밀도가 크게 향상되었으며, 다수 센서 데이터를 활용하는 기술 개발이 많은 관심을 받고 있다. 또한, 이 데이터들을 효율적으로 분석 및 처리하기 위해 다양한 인공지능 기술들이 연구되고 있다(Almalioglu et al., 2022; Al-Qizwini et al., 2017; Farag et al., 2018; Kwon et al., 2021; Li et al., 2021; Okuyama et al., 2018; Pak A et al., 2022; Prabhakar et al., 2017). 특히, 이미지 데이터에 특화된 합성곱신경망(convolution neural networks: CNNs)은 고사양 컴퓨팅 자원의 등장으로 인해 의료, 산업, 자율주행 등의 분야에서 두드러지는 성과를 보이고 있다(He et al., 2016; Krizhevsky et al., 2017; LeCun et al., 1998; Simonyan et al., 2014; Szegedy et al., 2015). CNNs 기반 이미지 인식 분야에서는 분류(classification), 객체분할(object segmentation), 객체탐지(object detection), 다중객체추적(multi object tracking: MOT) 등으로 구분된다(Girshick, 2014; Redmon et al., 2016). 이 중 CAA 시스템을 포함한 자율주행 자동차 기술에서는 주로 MOT를 활용한 연구가 진행 중이다.

MOT는 동영상에서 객체탐지 및 객체분할과 동시에 인식하고자 하는 객체가 움직이는 경로상 이전 프레임(frame)에서 탐지된 객체와 동일 객체인지 여부를 인식하는 추적 기술을 지칭한다. 최근까지도 MOT를 수행하기 위한 많은 알고리즘들이 개발되고 있는데, 대부분 임계값 보다 높은 탐지박스와 경로조각(tracklet)을 연결하여 동일 객체를 추적하게 된다. 객체추적의 정확도를 높이기 위해서는 조명 및 조도(brightness and contrast) 변화 문제 외 객체 간 중첩(occlusion)되거나 자체 중첩(self-occlusion)되는 경우가려진 객체에 대한 경계박스(bounding box) 추정이 어려워지는 문제 등을 해결할 필요가 있다(Bochinski et al., 2017; Wang et al., 2020; Xu et al., 2022; Zeng et al., 2021; Zhanget al., 2021).

최근 제안된 ByteTrack은 기존 MOT 알고리즘에서 발생하는 가려진 물체의 경계박스를 제거하는 문제를 tracklet의 유사점을 활용하여 높은 경계박스뿐만 아니라 낮은 경계박스도 연결하여 객체를 추적하는 방법으로 해결했다(Zhang Y, et al., 2021). Tracklet는 객체의 ID 값을 유지하는데 구하는 짧은 구간에서의 경

로를 의미한다. ByteTrack은 YOLO(You Only Look Once) 기반의 YOLO-X 아키텍처를 사용하였다. YOLO는 객체탐지를 수행하는 인공지능 알고리즘 중의 하나이다. YOLO 이전의 객체탐지 알고리즘은 분류와 지역화(localization)의 두 가지 일을 순차적으로 학습하기 때문에 학습 수렴(convergence) 속도가 느린 단점이 있었다. YOLO는 기존 알고리즘들과 달리 이 두 가지 일을 동시에 처리하는 one-stage-detection 방법을 채택하여 실시간 객체탐지가 가능하게 만들었다. 따라서 YOLO는 추론 속도 면에서 빠른 실시간 처리가 필요한 자율주행에서 각광 받고 있다. YOLO는 YOLOv1, YOLOv2, YOLOv3, YOLOv4, 그리고 YOLOv5로 발전해왔다. 하지만 해당 논문은 YOLOv4, YOLOv5는 파이프라인을 anchor 위주로 최적화가 진행되어있기 때문에, 일반적인 성능이 오히려 떨어질 수 있다고 판단하여서 YOLOv3를 YOLO-X의 베이스로 삼았다.

YOLO-X는 YOLOv3를 기반으로 사용하지만, head 부분에서는 기존 YOLOv3와 달리 decoupled head를 사용한다(Ge Z, et al., 2021). 분류와 경계박스 회귀(bounding box regression)는 서로 다른 특성을 갖는다. 분류에는 완전연결레이어(fully connected layer)가 효과적이지만, 반면에 지역화에는 convolution head가 효과적이다. 따라서 기존 연구에서 이러한 head 부분을 double-head 방식으로 변경하여 분류에는 완전연결 head를, 지역화에는 convolution head를 적용함으로써 성능을 향상시켰다. 분류에는 이진교차엔트로피(binary cross entropy) 손실함수(loss function)를 사용하고 지역화에는 IoU(intersection over union) 손실함수를 사용하여 학습을 진행한다. 실험 결과, 해당 방식이 기존 coupled head 방식보다 학습 수렴 속도가 빠르고 AP(average precision)가 향상되었다고 한다. 또한, YOLOv3-ultralytics 알고리즘의 성능(AP: 44.3)보다 YOLO-X의 성능(AP: 51.2)이 향상된 성능을 보여준다.

더불어 자율주행을 위해 실시간 횡단보도 검출 알고리즘에 대한 연구 또한 활발히 진행되고 있다(Chen et al., 2019; Uddin and Shioyama., 2005; Zhang Z, et al., 2022). 최근 제안된 CDNet(crosswalk detection network)은 자동차에 카메라 센서와 Jetson nano device를 부착하고, 카메라 센서로부터 수집된 데이터를 Jetson nano device에서 실시간으로 횡단보도를 검출하기 위한 프레임워크를 제안한다(Zhang Z, et al., 2022). CDNet은 실시간 횡단보도 검출을 위해 YOLOv5 모델을 기반으로 실제 기상환경(구름, 맑음, 비, 안개 및 밤)과 같은 실제 복잡한 시나리오에서 적용하기 위해 다양한 데이터 증강 방법이 적용되었다. CDNet은 Jetson nano device에서도 33.1 fps로 처리 속도가 빠르고 F1 score 94.83%의 높은 성능을 보였다.

기존연구에서는 보행자의 인식 및 보행자 횡단 예측이 자율주행에서 중요한 과제로 인식되었으며, 다양한 연구가 진행되고 있다. 하지만 보행자 신호가 있는 횡단보도 또는 단순히 보행자의 횡단 유무를 추정하는 연구에 그쳤다. 이는 무단횡단이나 보행자 신호등이 없는 경우 적용이 어려우며 단순히 횡단 유무를 예측하는 것은 보행자가 횡단보도까지 도달하는 시간을 알기 어렵기 때문에 차량이 보행자보다 먼저 횡단 가능한지 여부를 알기 어렵다. 이는 교통 흐름을 방해하는 요소가 된다. 최근 차량이 우회전시 보행자가 횡단 여부가 있다면 무조건 정지 해야 하는 법규가 시행되었다. 하지만 보행자의 횡단 예측 어려움으로 운전자가 보행자 신호 동안 정지해서 출발지연이 발생하는 교통 흐름 방해 문제가 야기되었다. 따라서 무신호 횡단보도에서 실시간으로 보행자를 인지 및 추적해서 보행자 횡단시간을 예측할 수 있는 연구가 필요하다(Hashimoto et al., 2015).

본 논문에서는 자율주행의 특성에 필수요소인 실시간 처리를 위해 YOLO를 백본모델(backbone)로 사용하는 ByteTrack과 CDNet을 활용하여 횡단보도 인근의 보행자 탐지 및 횡단 예측 연구를 진행하였다. 횡단보도 검출에는 CDNet 모델과 공개되어 있는 가중치(weight)를 활용하였으며, 보행자 검출을 위해 ByteTrack 알고리즘을 학습시켰다. CDNet과 ByteTrack의 출력값(output)들을 이용해 횡단보도 인근의 보행자 탐지 및 횡단 예측 연구 방법을 제안한다.

### 3. 본론

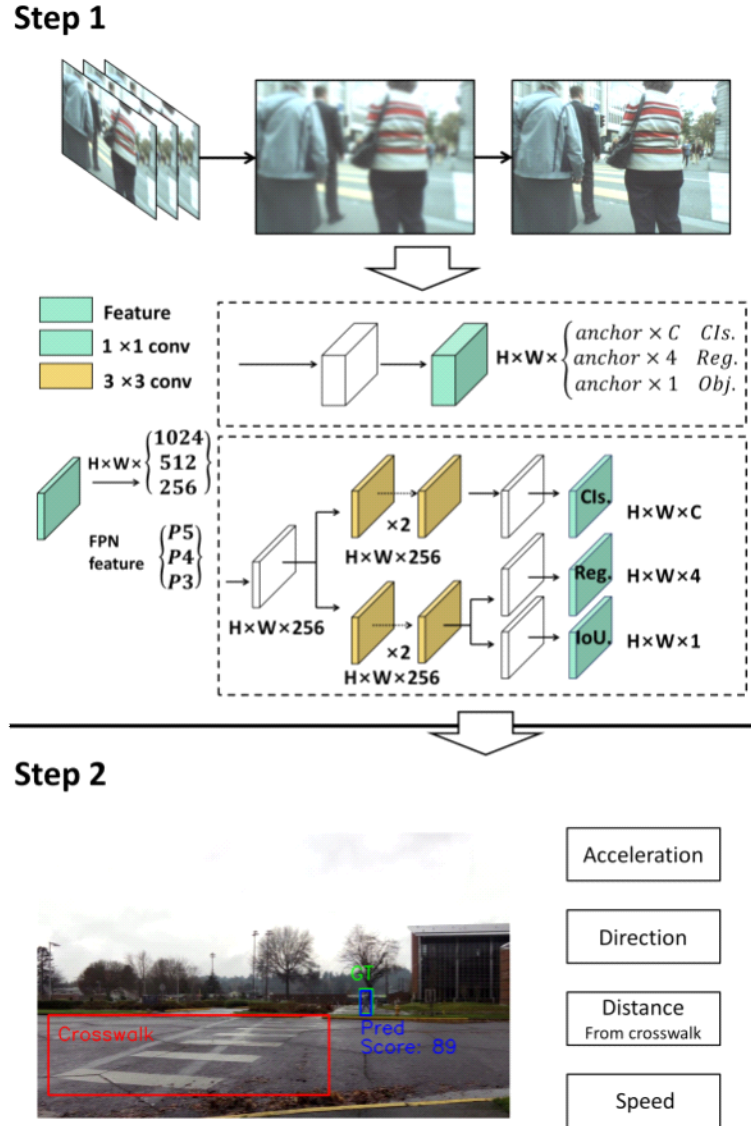


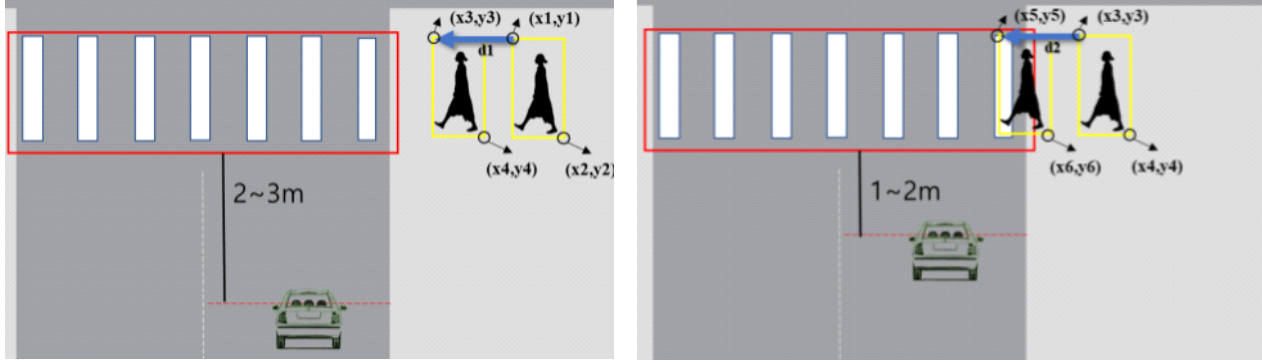
Figure 2. Overall procedure of the proposed approach

3장에서는 횡단보도 인근 보행자 탐지 및 횡단 예측 알고리즘에 대해 기술한다. 제안하는 알고리즘은 Figure 2와 같이 도식화할 수 있으며 총 2단계로 구성된다. 첫째, 카메라 센서로부터 수집된 데이터인 동영상 데이터를 인공지능 알고리즘이 학습할 수 있도록 전처리하는 과정이다. 전처리기(pre-processor)에서는 동영상을 단일 이미지로 만들고 이미지 강조 처리(image enhancement)를 한다. 이후 ByteTrack을 통해 보행자를 인식 및 추적한다. 두 번째 단계에서는 추정된 보행자 객체를 인지하여 진행방향, 속도, 가속도, 보행자와 횡단보도 간 거리를 계산한다.

첫 번째 단계에서는 인공지능 알고리즘에 입력하기 위한 데이터를 전처리한다. 데이터 전처리에는 동영상을 프레임(frame) 단위로 추출하여 단일 이미지로 만들고, 이미지의 밝기(brightness)와 대비(contrast), 선명화(sharpening) 등의 처리를 거쳐 특징을 강조한다. ByteTrack을 통해 실시간으로 보행자를 추적한다. ByteTrack 알고리즘에 보행자를 학습(fine-tuning)하였다. 또한, ByteTrack 알고리즘의 강건한 예측성능을 확보하기 위해 데이터증강(data augmentation) 기법을 적용하여 학습데이터를 구성하였다. 이미지에 대한 데이터증강 시 좌우반전(horizontal flip), 회전(rotation), 이동(translation) 등의 실제 데이터와의 이질성

(heterogeneity)이 적은 증강 기법만을 사용하였다.

각 프레임은 초당 30장이 수집된다고 가정할 때, 우선 ByteTrack으로 탐지한 보행자의 bounding box로부터 추적하며 이동 방향 및 속도에 대한 정확한 추정을 위해 초기 프레임  $f_0$ 과  $\tau$ 시점 이후의 두번째 프레임  $f_{+\tau}$ 을 샘플링한다.



(a) An example schematically illustrating a method of the proposed system. It detects crosswalks (red box) and navigates pedestrians towards crosswalks (yellow box). Additionally, the proposed system performs object tracking on a frame-by-frame basis to identify movement for the pedestrian.

(b) An example illustrating a method of estimating the remaining time until the pedestrians arrive at a crosswalk. The speed of the pedestrian and the remaining time of arrival at the crosswalk are estimated according to Equations (1) to (9).

Figure 3. Scenarios of the proposed system to detect the crosswalk, track the pedestrians, and estimate the remaining time to the crosswalk.

두 번째 단계에서는 보행자의 속도, 가속도, 진행방향, 횡단보도와 남은 거리를 계산한다. Figure 3은 첫 번째 단계에서 탐지한 결과의 예를 도식화한 것이다. Figure 3 (a)의 가장 우측에 있는 경계박스는 첫 번째 프레임에서 탐지된 보행자이며, 좌측 경계박스는 두 번째 프레임에서 탐지된 보행자를 의미한다. Figure 3 (b)의 경계박스는 두 번째 프레임에서 탐지된 보행자(우측)와 세 번째 프레임에서 탐지된 보행자(좌측)를 표시한 것이다.

보행자의 진행방향, 속도, 가속도, 이동거리, 횡단보도와 보행자 간의 거리를 계산하기 위해 경계박스의 좌표를 이용한다. Figure 3 (a)를 예를 들면, 첫 번째 프레임에서의 보행자 경계박스의 좌표를  $(x_1, y_1)$ ,  $(x_2, y_2)$ 로 두고, 보행자가 움직인 다음 프레임에서의 좌표를  $(x_3, y_3)$ ,  $(x_4, y_4)$ 라고 가정한다. 이때, 보행자의 진행방향( $\vec{v}_2$ ), 속도( $s_2$ ), 그리고 이동거리( $d_2$ )는 각각 식 (1)~(3)과 같다. 이동거리는 유클리디안 거리(Euclidean distance)를 이용하며, 속도는 이동거리를 시간(frame per second: fps)으로 나누어 계산한다.

$$\vec{v}_2 = \begin{cases} LEFT & (x_3 - x_1 < 0) \\ RIGHT & (x_3 - x_1 > 0) \\ FORWARD & (y_3 - y_2 > 0) \text{ and } (y_4 - y_2 < 0) \\ BACKWARD & (y_3 - y_1 < 0) \text{ and } (y_4 - y_2 > 0) \\ STOP & (x_3 - x_1 \leq \epsilon) \text{ and } (y_3 - y_1 \leq \epsilon) \end{cases} \quad (1)$$

$$d_2 = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} \quad (2)$$

$$s_2 = \frac{d_2}{\tau} \quad (3)$$

식 (2)와 (3)의 방식으로 Figure 3 (b)의 보행자 이동거리 및 속도는 (4), (5)과 같이 유도할 수 있다.

$$d_3 = \sqrt{(x_5 - x_3)^2 + (y_5 - y_3)^2} \quad (4)$$

$$s_3 = \frac{d_3 - d_2}{\tau} \quad (5)$$

위 수식들을 일반화하면 아래 식 (6)~(8)과 같다. 이때,  $f$ 는 프레임을 의미하며,  $x_{(f,lu)}$ ,  $y_{(f,rd)}$ 는 각각 해당 프레임에서의 좌측상단(left-up), 우측하단(right-down)에서의  $x$ 와  $y$ 좌표로 정의한다.

$$d_f = \sqrt{(x_{(f,lu)} - x_{(f-1,lu)})^2 + (y_{(f,lu)} - y_{(f-1,lu)})^2} (f \geq 1) \quad (6)$$

$$s_f = \begin{cases} \frac{d_f}{t_f} & (f = 1) \\ \frac{d_f - d_{f-1}}{t_f - t_{f-1}} & (f \geq 2) \end{cases} \quad (7)$$

$$a_f = \begin{cases} s_f & (f \leq 2) \\ s_f - s_{f-1} & (f \geq 3) \end{cases} \quad (8)$$

또한, 위 수식들의 결과를 바탕으로 보행자가 횡단보도에 도착하기까지의 잔여시간(remaining time)  $t_{(f,cp)}$ 은 식 (9)와 같다.  $d_{(f,cp)}$ 는 횡단보도와 보행자 간의 거리로 정의한다. 식 (3)에서 측정된  $\tau$ 시점 이후 움직인 거리  $s_2$ 은 픽셀로 환산할 수 있는데, 예를 들어  $\tau=30(\text{frame})$ 에서 180 픽셀만큼 이동했다면, 등속도를 가정할 때 프레임 당 6 픽셀만큼 이동했음을 추정할 수 있다. 그러므로, 보행자의 위치로부터 횡단보도까지의 거리인  $d_2$ 에 도달하는데 잔여시간  $t_{(f,cp)}$ 은 식 (9)와 같다. 예를 들어,  $\tau$ 시간 내 180픽셀만큼 이동했는데 횡단보도까지 360 픽셀이 남은 경우, 잔여시간은  $\frac{360}{180}\tau$  이다.

$$t_{(f,cp)} = \frac{d_{(f,cp)}}{d_f} \times fps \quad (f \geq 1) \quad (9)$$



## 4. 실험 및 결과

### 4.1 연구배경 및 목적

인공지능 알고리즘을 학습시키기 위해서는 고성능의 그래픽 처리 장치(graphics processing unit: GPU)가 필요하다. 본 연구에서는 3장에서 제안한 인공지능 알고리즘 학습을 위해 GPU는 TITAN Xp를 사용하였고, 소프트웨어는 python 3.8.13, openCV 4.6.0.66, torch 1.12.1를 사용하였다.

ByteTrack 알고리즘을 학습시키기 위한 데이터셋으로 MOT17, MOT20, ETHZ를 이용하였다(Dendorfer., 2020; Ess et al., 2018; Milan., 2016). 해당 데이터들은 보행자와 관련한 빅데이터로 여러 카테고리들을 분할(segmentation)하거나 탐지(detection), 추적(tracking)하는 것을 목표로 한다. 데이터셋은 총 50,426장의 이미지이며, 해상도는 640x480~1920x1080, 그리고 fps는 14~30로 이루어져 있다. 전체 데이터셋 중 43,311장을 학습(train) 데이터셋으로, 7,115장을 검증(validation) 데이터셋으로 사용하였다.

또한, 본 논문에서 제안한 횡단보도 인근에서의 보행자 검출과 보행자 상태를 계산하기 위해 추가적으로 pedestrian dataset을 사용하였다(Meschke., 2017). pedestrian dataset은 횡단보도(crosswalk), 사거리(fourway), 밤(night)에 촬영한 3개의 동영상으로 구성된다. 해상도는 1920x1080이며, 30 fps로 이루어져 있다. 3장의 전처리기를 이용하여 동영상 데이터를 횡단보도 378장, 사거리 1,659장, 그리고 밤 565장의 단일 이미지로 변환하여 사용하였다.

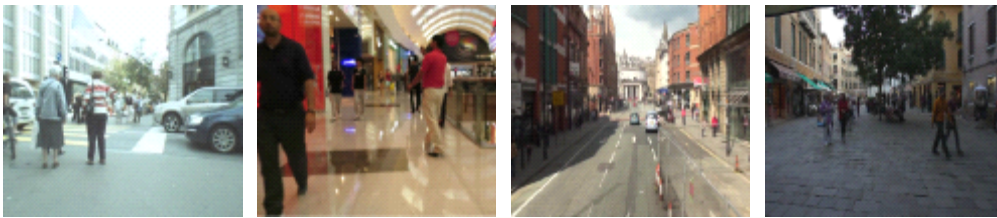


Figure 4. Sample frames of tracking people in MOT17 dataset

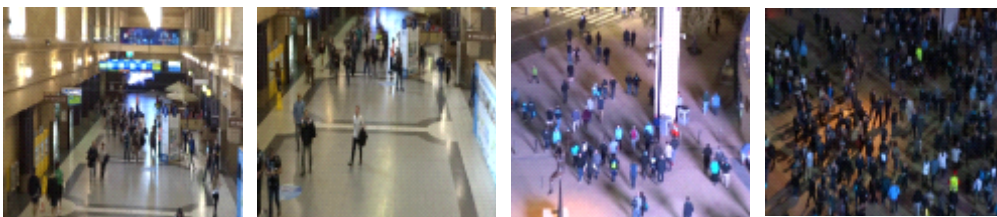


Figure 5. Sample frames of tracking people in MOT20 dataset

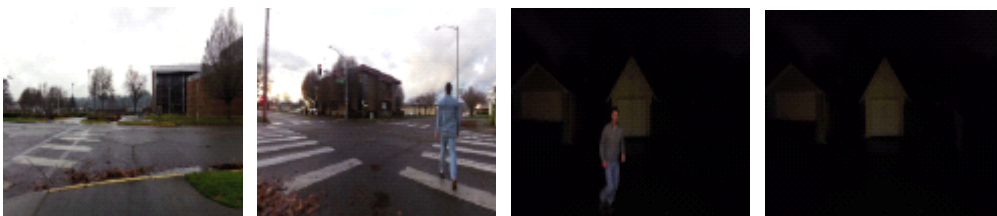


Figure 6. Sample frames in Pedestrian dataset

## 4.2 성능평가

### 1) 성능지표

본 논문에서는 성능 평가를 위한 주요 성능지표로 MOTA(multi-object tracking accuracy)와 IDF1(identification F1 score), 그리고 IDs(ID switches)를 사용하였다. MOTA는 Bernardin K et al(2006)이 MOT 알고리즘에서의 정확도를 측정하기 위해 제안한 성능지표이다. MOTA의 식은 (10)과 같다. 식 (10)에서  $t$ 는 시간을 의미하며,  $m$ 은 인식하지 못한 객체의 개수를 의미한다.  $fp$ 와  $mme$ 는 찾은 객체들 중 잘못된 찾은 것을 의미하며 각각 위양성(false positive: FP)과 위음성(false negative: FN)에 해당하는 개수를 의미한다.  $g$ 는 존재하는 모든 물체의 개수를 의미한다.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (10)$$

IDF1은 MOT 알고리즘이 추적하고 있는 ID가 정답군(ground truth)과 얼마나 일치하는지를 평가하기 위한 지표로 ID에 대한 F1 점수이다. IDF1은 식 (13)과 같으며, IDR와 IDP는 각각 ID에 대한 재현율(identification recall)과 정밀도(identification precision)로 식 (11), (12)와 같다.

$$IDR = \frac{IDTP}{IDTP + IDFN} \quad (11)$$

$$IDP = \frac{IDTP}{IDTP + IDFP} \quad (12)$$

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \quad (13)$$

IDs는 탐지한 결과와 실제 객체에 할당되어 있는 ID가 다른 것을 의미한다. 그 외 추가적인 성능지표로 MT(number of mostly tracked trajectories), ML(number of mostly lost trajectories), FP, 그리고 FN을 사용하여 성능을 평가하였다.

### 2) 결과

<Table 1> Performance evaluation of ByteTrack regarding MOT17 and MOT20 datasets. The experiments were conducted on ByteTrack without additional training phase.

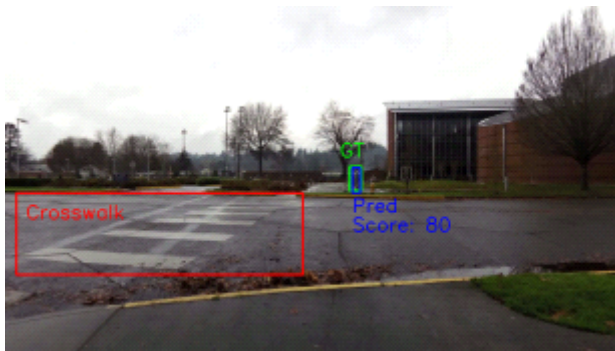
Dataset (N)	MOTA ↑	IDF1 ↑	IDs ↓	MT ↑	ML ↓	FP ↓	FN ↓
MOT17 (2,652)	75.3%	77.1%	134	186	55	2,101	11,093
MOT20 (4,463)	63.5%	66.2%	1,320	642	201	65,462	157,580

ByteTrack 알고리즘의 성능을 평가하기 위해 학습되지 않은 MOT17와 MOT20의 데이터를 이용하였다. 성능 평가 결과는 아래 <Table 1>과 같다. MOT17에서는 MOTA가 75.3%로 꽤 높은 성능을 보였으나 MOT20 데이터셋에서는 다소 떨어진 63.5%의 성능을 나타냈다. 이는 MOT17 데이터셋의 객체(objects) 개수는 53,890개로 MOT20의 615,137개에 비해 현저히 적기 때문으로 판단된다. 특히, MOT20 데이터셋은 군집되어 있는 객체가 다수 포함되어 있어 MOT17과 비교하여 상대적으로 낮은 성능을 보였다.

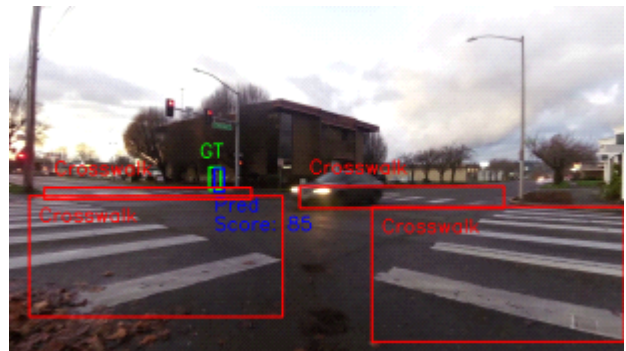
<Table 2> Performance evaluation of Pedestrian dataset  
in terms of detection and tracking accuracy.

Dataset (N)	MOTA ↑	IDF1 ↑	IDs ↓	MT ↑	ML ↓	FP ↓	FN ↓
Crosswalk (378)	100.0%	100.0%	0	1	0	0	0
Fourway (1,659)	90.9%	28.8%	5	1	0	52	59
Night (565)	49.9%	55.5%	2	0	0	13	268
Overall (2,224)	82.1%	47.3%	7	2	0	65	327

<Table 2>는 Pedestrian dataset에서의 성능 평가 결과이다. 전체적인(overall) 성능은 MOTA가 82.1%로 각각의 데이터셋에서 100%, 90.9%, 49.9%의 성능을 보였다. 이 중 밤에 해당하는 데이터셋은 객체가 잘 보이지 않는 이미지의 포함으로 인해 FN이 증가하여 상대적으로 낮은 성능을 보였다. Figure 7는 MOT 알고리즘이 탐지한 경계박스와 정답군을 그림으로 나타낸 것이다. 녹색 경계박스는 정답군이며, 파랑은 탐지한 경계박스, 그리고 빨강은 횡단보도의 경계박스이다. (단, 여기서 빨강 object는 횡단보도에 대한 CDNet과 추론 결과를 토대로 별도 표기하였다. Figure 7 (a), (b)는 탐지점수(pred score)가 높고(80 이상), 경계박스도 정답군과 어느 정도 일치함을 확인할 수 있다. 그러나 Figure 7 (c)와 같이 탐지점수가 낮거나 Figure 7 (d)처럼 이미지가 너무 어두운 경우 탐지하지 못하는 경우가 발생하였다.



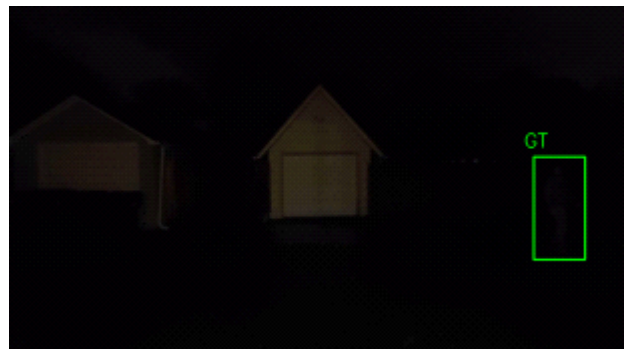
(a) Example of detecting crosswalk and a pedestrian



(b) Example of detecting multiple crossroads and a pedestrian



(c) Example of detecting a pedestrian in dark situations



(d) Example of in-screen detection of insufficient light, not even pedestrians

Figure 7. Examples of pedestrian detection and tracking results

Figure 8과 Figure 9은 Pedestrian dataset의 횡단보도 데이터셋을 이용해 본 논문에서 제안하는 알고리즘의 결과를 계산하여 그림에 표기한 것이다. Figure 8은 1 프레임 ~ 8 프레임까지의 결과이며, Figure 9은 보행자가 횡단보도에 도착하여 건널 때까지(62 프레임 ~ 69 프레임)의 결과이다. 경계박스가 보행자를 잘



탐지하고 있음을 확인할 수 있다. 제안하는 알고리즘을 이용한 보행자의 진행방향, 속도, 가속도, 횡단보도에 도달하기까지의 시간 또한 시간의 흐름에 따라 보행자의 상태를 잘 반영하고 있음을 확인할 수 있다. 다만, 프레임마다 예측하다 보니 Figure 8의 8번째 프레임에서 횡단보도 도착 시간 결과( $t_{(f, cp)}$ )처럼 어느 정도 튀는 값이 존재한다. 이는 추후 보행자의 보행 속도를 프레임 단위로 표준화한 후 표준화시킨 프레임마다 슬라이딩 윈도우 방식을 적용하여 튀는 값을 조정할 수 있을 것이다. 또한, 횡단보도의 경계박스와 보행자의 경계박스가 교차하는 지점부터는 보행자의 상태를 '건너고 있음(crossing)'으로 정의하였다. 이는 Figure 9의 63 프레임 ~ 69 프레임에서 확인할 수 있다.

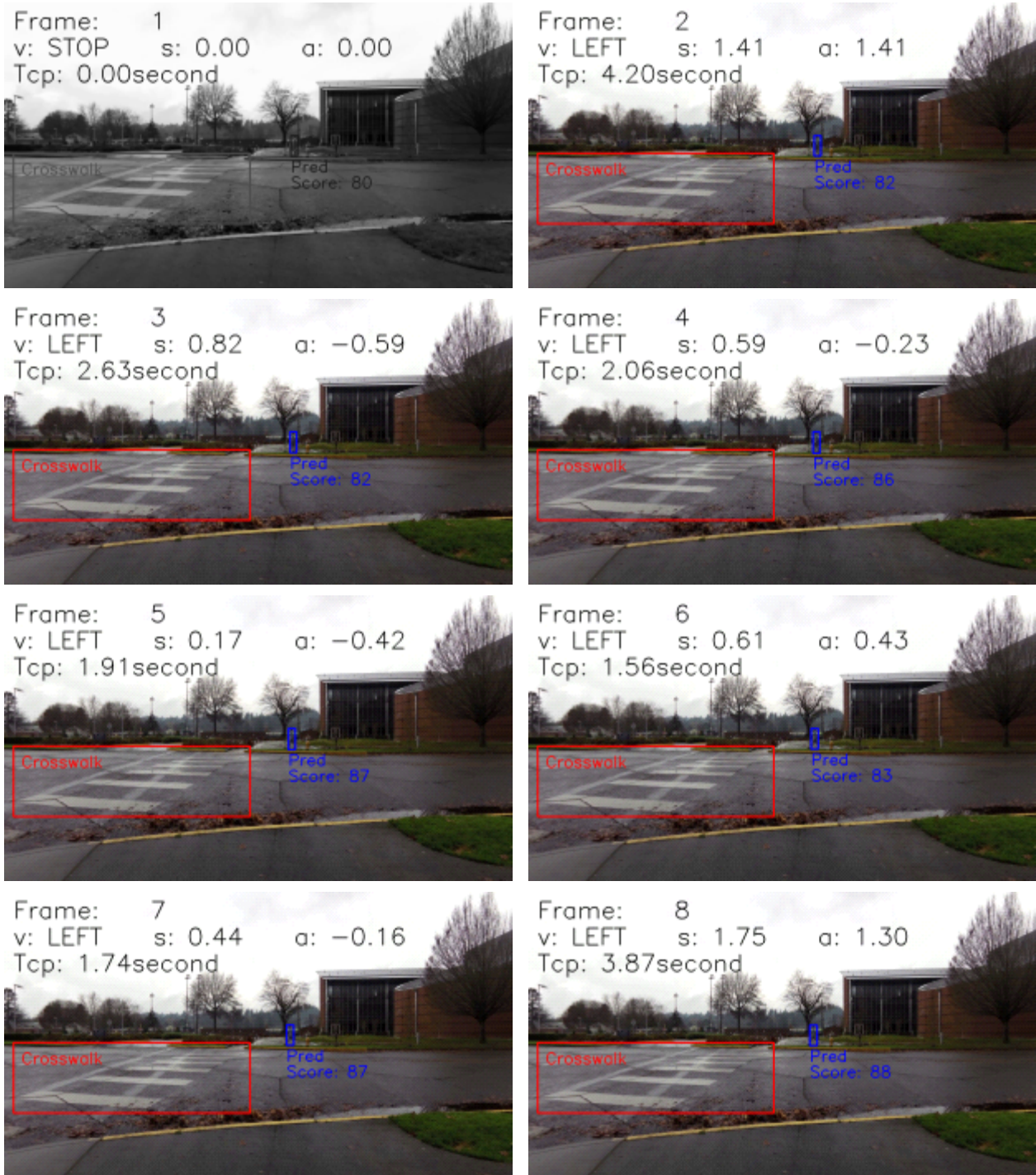


Figure 8. Examples of tracking a pedestrian at crosswalks and estimating the remaining time through eight sequential frames(1-8) (direction: v, speed: s, acceleration: a, remaining time until arrival at crosswalk: Tcp, status: crossing)



Figure 9. Examples of tracking a pedestrian at crosswalks and estimating the remaining time through eight sequential frames(62-69) (direction: v, speed: s, acceleration: a, remaining time until arrival at crosswalk: Tcp, status: crossing)



### 4.3 적용사례

본 절에서는 직접 촬영한 영상을 제안한 알고리즘으로 처리한 결과에 대해 기술한다. 영상 촬영 장소는 광운대학교 앞 인근 횡단보도이며, 촬영기기는 아이폰 14 프로를 이용하여 HD(1280x720), 30 fps로 촬영하였다. 실제 상황과 유사한 환경에서 촬영하기 위해 보도의 연석 부분에서 촬영을 진행하였으며, 안전을 위해 신호등은 보행가능 신호일 때만 촬영을 진행하였다. 다양한 상황에서 알고리즘을 평가하기 위해 다수의 보행자가 있는 환경에서 촬영하였으며 보행자가 횡단보도까지 걸어서 지나가는 경우, 횡단보도까지 걷다가 멈추는 경우, 그리고 횡단보도까지 걷다가 뛰는 경우를 추가 실험하였다.

#### 1) 여러 보행자 인지



Figure 10. Example of detecting multiple pedestrians around crosswalks

Figure 10은 다수의 보행자들이 있을 때 알고리즘이 인지한 보행자를 표시한 것이다. 결과를 보면 다수의 보행자들에 대해서도 잘 인지하고 있음을 확인할 수 있었으며, Figure 10의 상단 그림과 같이 보행자가 겹쳐있거나, Figure 10의 하단 그림처럼 보행자 영역 일부가 잘린 경우에도 보행자로 인지하고 있음을 알 수 있다.

## 2) 횡단보도까지 걸어서 이동하는 경우

Figure 11은 특정 보행자(ID: 1)가 걸어서 횡단보도까지 이동하여 건너는 경우를 제안 알고리즘으로 처리한 결과이다. 보행자의 경계박스와 횡단보도의 경계박스가 겹치는 경우 상태를 '건너고 있음(crossing)'으로 처리하였다. Figure 11을 보면 보행자가 횡단보도까지 걸어가는 동안 남은 시간이 줄어들고 있음을 알 수 있다.



Figure 11. An example of detecting the ID#1 pedestrian approaching the crosswalk (direction: v, speed: s, acceleration: a, time remaining until arrival at crosswalk: Tcp, status: crossing)



### 3) 횡단보도까지 걸다가 멈추는 경우와 뛰는 경우

Figure 12는 특정 보행자(ID: 1)가 걸어서 횡단보도까지 이동하여 건너는 경우를 제안 알고리즘으로 처리한 결과이다. Figure 12를 보면 보행자가 횡단보도까지 걸어가는 동안 남은 시간이 줄어들다가 보행자가 멈추는 경우 상태는 '멈춤(stop)'으로 바뀌고 남은 시간이 0으로 변화하였음을 알 수 있다.



Figure 12. An example of detecting multiple pedestrians approaching a crosswalk (direction: v, speed: s, acceleration: a, time remaining until arrival at crosswalk: Tcp, status: crossing)



Figure 13은 특정 보행자(ID: 11)가 걸어서 횡단보도까지 뛰어서 이동하여 건너는 경우를 제안 알고리즘으로 처리한 결과이다. Figure 13을 보면 보행자가 횡단보도까지 뛰어가는 동안 남은 시간이 빠르게 줄어듦을 알 수 있다. 또한, 보행자와 횡단보도의 경계박스가 겹치는 경우 상태가 '건너고 있음(crossing)'으로 변화한 것을 확인할 수 있다.

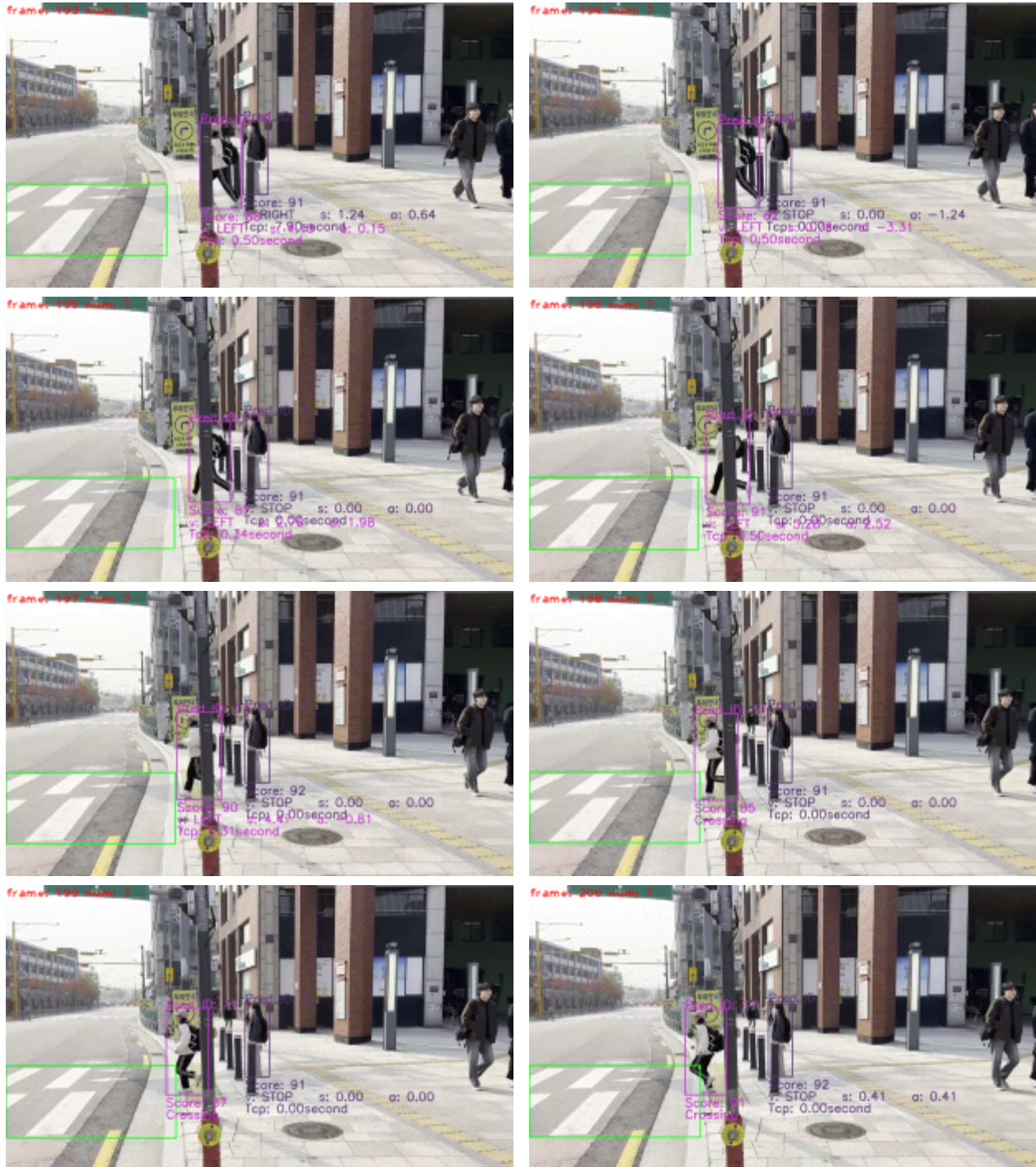
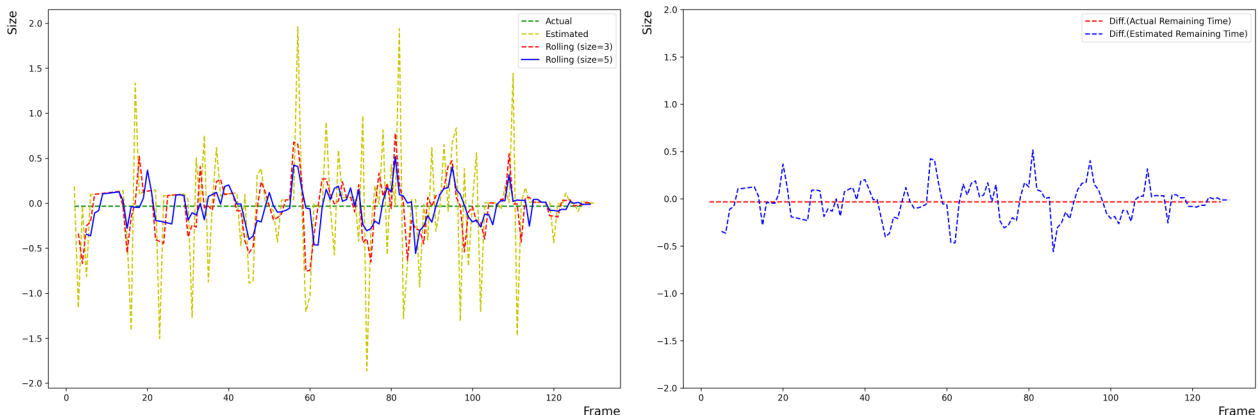


Figure 13. An example of detecting the ID#11 pedestrian running into a crosswalk (direction: v, speed: s, acceleration: a, time remaining until arrival at crosswalk: Tcp, status: crossing)

## 5. 결과분석 및 토론

5장에서는 4장에서의 결과에 대해 분석하고, 그 의미에 대해 토론한다. 먼저, 보행자 인지 모델의 성능은 MOT17 데이터셋에서 75.3%, MOT20에서 63.5%의 MOTA를 보였으며, 테스트 데이터셋에서 82.1%의 성능을 보여주었다. 장당 처리 속도는 GPU TITAN Xp 환경에서 약 31.2 fps를 보여 실시간 처리에도 적용이 가능함을 보여주었다.

또한, 인공지능 모델들에서의 결과를 기반으로 제안 알고리즘을 실제 환경에 적용하여 보행자의 방향, 속도, 그리고 횡단보도까지 남은 시간 등을 계산하였다. Figure 14는 보행자가 횡단보도에 도달하기까지의 실제 남은 시간과 제안 알고리즘의 예측결과를 그래프로 비교한 것이다. Figure 14에서 녹색은 실제시간, 노랑은 예측시간, 그리고 빨강, 파랑은 후처리된 예측시간 의미한다. 프레임 간 추정 오차를 보정하기 위해서 이동평균(moving average; MA) 필터를 적용하였으며, 윈도우 크기(time window size)는 각 3(빨강)과 5(파랑)으로 표시하였다. Figure 14 (b)에서 확인할 수 있듯 일정한 패턴을 그리며 점차 줄어들고 있음을 알 수 있다. Figure 14를 통해 제안하는 알고리즘이 보행자가 횡단보도에 도달하기까지의 시간을 반영하고 있음을 알 수 있었다. 이를 활용하여 차량 운전자의 보행자가 횡단보도에 진입하기까지의 시간을 알림 서비스 형태로 제공하여 안전사고를 예방하고, 나아가 자율주행자동차의 차량제어 시스템에 활용될 수 있음을 확인하였다.



(a) Experimental results showing the difference between the actual and estimated remaining time of pedestrians crosswalk arrival. Green dotted line: actual remaining time, yellow dotted line: estimated remaining time, red dotted line: MA(3), blue solid line: MA(5)

(b) Experimental results showing the difference between the actual remaining time and the estimated remaining time with MA(5). Overall, the predicted values of the remaining time is accurately estimated to be within 0.5 seconds of those of the actual value.

Figure 14. Experimental results of estimating the time the pedestrian arrives at the crosswalk

## 6. 결론

본 논문에서는 도시 물류에서 있어 자동차에 의한 보행 안전사고의 위험을 줄이기 위해 횡단보도 인근 보행자 탐지 및 횡단 예측 시스템을 제안하였다. 횡단 예측 시스템에서는 횡단보도 인지와 보행자 인지 및 추적을 위한 동작을 분리함으로써 부하 분산 및 추론시간의 단축을 꾀하였다. 또한, 횡단보도 인지를 위해서는 CDNet을 활용하였고 다수 보행자를 정확하게 추적하여 횡단까지 잔여시간을 추정하기 위해 ByteTrack을 사용하였다. CDNet와 ByteTrack에서 추출한 값을 가지고 보행자의 횡단 여부를 예측하기 위한 값인 보행자의 진행 방향, 속도, 가속도, 횡단보도와의 거리를 계산해서 횡단보도에 도착까지 잔여시간을 추정하였다.

본 연구에서는 ByteTrack이 실제 활용 목적에 적합한지 여부를 판단하기 위해 MOT17와 MOT20의 데이터를 활용하여 검증하였다. 실험 결과, MOT17에서는 MOTA가 75.3%, MOT20에서 63.5%의 성능을 보였다. MOT20은 보행자의 밀집도가 크다 보니 MOT17 데이터셋에 비해 상대적으로 낮은 성능을 보였다. 또한, Pedestrian 데이터셋에서 성능 평가 결과는 MOTA 82.1% 성능을 나타냈다. 추가로 다양한 상황에서 알고리즘을 탐지 및 추적 성능을 평가하기 위해 실제 환경에서 직접 촬영을 통해 시나리오 검증을 수행하였다. 특히, 보행자가 횡단보도까지 걸어서 지나가는 경우와 횡단보도까지 건다가 멈추는 경우, 그리고 횡단보도까지 건다가 뛰는 경우를 추가로 실험하여 제안 방식의 정확도 성능을 보였다. 제안 알고리즘은 다양한 상황에 적용 가능성을 보여주며 전반적으로 높은 MOTA 성능을 보여주었다.

제안 알고리즘은 교차로 교통사고를 줄이기 위해 다양한 방식으로 실제 서비스 상용화에 기반기술로 활용될 수 있을 것으로 기대된다. 첫째, 보행자가 횡단보도에 진입하기까지의 시간을 차량 운전자에 알려주는 알림 서비스로 활용될 수 있다. 둘째, 대한민국을 포함하여 다수 나라의 교차로 법규에 따라 범용적으로 자율주행차량에 응용될 수 있다. 셋째, 비전 기반의 보행자 충돌에 대응하는 충돌방지도조시스템의 핵심 기술의 기반이 될 것으로 기대한다.

다만, 본 논문에서 제안한 알고리즘에서의 횡단 여부 예측 정확도의 경우 프레임마다 예측하기 때문에 이상치(outlier)가 존재하여 현시점에서 자율주행자동차에 접목하기에는 강건함이 떨어진다. 따라서 추후 논문에서는 본 논문에서의 제안 알고리즘에 칼만 필터(Kalman filter) 등을 이용해 보정(calibration)하고, 보행자의 속도를 프레임 단위로 표준화하여 정확도를 높일 계획이다. 또한, 슬라이딩 윈도우 방식을 적용하여 잡음(noise)을 최소화할 계획이다. 더불어 시뮬레이터를 사용해서 복잡한 시나리오에서 학습을 진행하여 보행자 횡단 여부 예측 정확도를 강화하고, 보행자 신호등과 같은 도로 인프라 데이터셋을 수집하여 더욱 강건한 시스템을 연구할 생각이다.

## 참고문헌

- Akhauri, S., Zheng, L. Y., and Lin, M. C. (2020). Enhanced transfer learning for autonomous driving with systematic accident simulation. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (pp. 5986-5993). IEEE.
- Almalioglu, Y., Turan, M., Trigoni, N., and Markham, A. (2022). Deep learning-based robust positioning for all-weather autonomous driving. *Nature Machine Intelligence*, 1-12.
- Al-Qizwini, M., Barjasteh, I., Al-Qassab, H., and Radha, H. (2017). Deep learning algorithm for autonomous driving using googlenet. In 2017 IEEE Intelligent Vehicles Symposium (IV), (pp. 89-96). IEEE.
- Althoff, M., Stursberg, O., and Buss, M. (2009). Model-based probabilistic collision detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 10(2), 299-310.
- Bernardin, K., Elbs, A., and Stiefelwagen, R. (2006). Multiple object tracking performance metrics and evaluation in a smart room environment. In Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV, (Vol. 90, No. 91). Citeseer.
- Bochinski, E., Eiselein, V., and Sikora, T. (2017). High-speed tracking-by-detection without using image information. In 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS), (pp. 1-6). IEEE.
- Chen, N., Hong, F., & Bai, B. (2019). Zebra crossing recognition method based on edge feature and Hough transform. *Journal of Zhejiang University of science and technology*, 31(6), 476-483.
- Cho, M. G. (2019). A study on the obstacle recognition for autonomous driving RC car using lidar and thermal infrared camera. In 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), (pp. 544-546). IEEE.
- Chu, K., Lee, M., and Sunwoo, M. (2012). Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE transactions on intelligent transportation systems*, 13(4), 1599-1616.
- Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., and Leal-Taixé, L. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*.
- Ess, A., Leibe, B., Schindler, K., and Van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, (pp. 1-8). IEEE.
- Farag, W., and Saleh, Z. (2018, April). Traffic signs identification by deep learning for autonomous driving. In Smart Cities Symposium 2018, (pp. 1-6). IET.
- Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 580-587).

- Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3), 362-386.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770-778).
- Jin, Y., Jeon, H., Shin, Y. N., and Hyun, E. (2018). Data Analysis for Sensor Fusion of AVM Camera and Automotive Radar. *Korean Society of Automotive Engineers Spring Conference*, 581-583.
- Kim, D., Kim, Y., Go, S., Kim, Y., Lee, Y., Kim, C., (2018). A Study on the Application of Autonomous Vehicles to the Logistics Industry : Focusing on the Change of the Logistics Hub. *Journal of the Korean Society of Supply Chain Management* 18(2), (pp. 81-90).
- Kim, T., Kim, S., Kim, K., Oh, Y., Lee, J., Cho, W., Kim, K., (2021). Logistics Sorting System using Autonomous Driving Robot. *Journal of The Korea Socceity of Computer and Information*, 29(2), (pp. 491-492).
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.
- Kwon, T. W., Lee, S. P., Kim, D., Jang J., Lee, M., Kang, S. U., Kim, H., Oh, K., On, J., Kim, Y. J., Yun, S. J., Jin, K. W., Kim, E. Y., Kim, and K. G. (2021). Diagnostic performance of artificial intelligence model for pneumonia from chest radiography. *Plos one*. 16.4: e0249399.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lee, S., Baek, S., and Lee, S. (2021). A Study on the Collision Avoidance Algorithm Using FMCW Radar. *Proceedings of the Korean Telecommunications Society Conference*, 668-669.
- Li, G., Yang, Y., Qu, X., Cao, D., and Li, K. (2021). A deep learning based image enhancement approach for autonomous driving at night. *Knowledge-Based Systems*, 213, 106617.
- Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., and Leibe, B. (2021). Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2), 548-578.
- Meschke, S. (2017). Pedestrian Dataset. Retrieved 2022 from <https://www.kaggle.com/datasets/smeschke/pedestrian-dataset>.
- Milan, A., Leal-Taixé, L., Reid, I., Roth, S., and Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.
- Noh, S., and Han, W. Y. (2014). Collision avoidance in on-road environment for autonomous driving. In *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, (pp. 884-889). IEEE.
- Okuyama, T., Gonsalves, T., and Upadhyay, J. (2018). Autonomous driving system based on deep q learnig. In *2018 International conference on intelligent autonomous systems (ICoIAS)*, (pp. 201-205). IEEE.
- Pak, A., Manjunatha, H., Filev, D., and Tsiotras, P. (2022). CARNet: A Dynamic Autoencoder for Learning Latent Dynamics in Autonomous Driving Tasks. *arXiv preprint arXiv:2205.08712*.
- Park, S., Kim, J., (2020). Development and Validation of a Picker Following Automated Guided Vehicle Design Process for Pickers in a Logistics Environment. *Korean Journal of Logistics*, 28(5), (pp. 11-26).

- Park, Y., Yang, J. H., and Lim, S. (2018). Development of complexity index and predictions of accident risks for mixed autonomous driving levels. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), (pp. 1181-1188). IEEE.
- Prabhakar, G., Kailath, B., Natarajan, S., and Kumar, R. (2017). Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. In 2017 IEEE region 10 symposium (TENSYP), (pp. 1-6). IEEE.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 779-788).
- Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In European conference on computer vision, (pp. 17-35). Springer, Cham.
- Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295.
- Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 1-9).
- Traffic Accident Analysis System (2021). Traffic Accidents by Accident Type. Retrieved 2022 from [http://taas.koroad.or.kr/sta/acs/gus/selectAcCntTyTfCacd.do?menuId=WEB\\_KMP\\_OVT\\_MVT\\_TAG\\_ATT](http://taas.koroad.or.kr/sta/acs/gus/selectAcCntTyTfCacd.do?menuId=WEB_KMP_OVT_MVT_TAG_ATT).
- Uddin, M. S., & Shioyama, T. (2005). Detection of pedestrian crossing using bipolarity feature-an image-based technique. IEEE Transactions on Intelligent transportation systems, 6(4), 439-445.
- Wang, Z., Zheng, L., Liu, Y., Li, Y., and Wang, S. (2020). Towards real-time multi-object tracking. In European Conference on Computer Vision, (pp. 107-122). Springer, Cham.
- Wei, Q., Won S., (2018). Conceptual Design and Importance Analysis of Industry 4.0 Based Products and Services Applied in Container Terminals, Korean Journal of Logistics, 26(3), (pp. 91-110).
- Xu, S., Wang, X., Lv, W., Chang, Q., Cui, C., Deng, K., Wang, G., Dang, Q., Wei, S., Du, Y., and Lai, B. (2022). PP-YOLOE: An evolved version of YOLO. arXiv preprint arXiv:2203.16250.
- Y. Hashimoto, Y. Gu, L. -T. Hsu and S. Kamijo, "Probability estimation for pedestrian crossing intention at signalized crosswalks," 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2015, pp. 114-119, doi: 10.1109/ICVES.2015.7396904.
- Zeng, F., Dong, B., Wang, T., Zhang, X., and Wei, Y. (2021). Motr: End-to-end multiple-object tracking with transformer. arXiv preprint arXiv:2105.03247.
- Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2021). Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864.
- Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision, 129(11), 3069-3087.

Zhang, Z. D., Tan, M. L., Lan, Z. C., Liu, H. C., Pei, L., & Yu, W. X. (2022). CDNet: a real-time and robust crosswalk detection network on Jetson nano based on YOLOv5. *Neural Computing and Applications*, 1-12.



# 도시철도 물류 시스템의 통합관제시스템 설계 및 수평이송장치 경로 생성 알고리즘 연구

황선우<sup>1</sup>, 김주욱<sup>2+</sup>, 김영민<sup>1</sup>, 최준우<sup>3</sup>, 한수민<sup>3</sup>, 박상원<sup>3</sup>, 장경호<sup>4</sup>

<sup>1</sup>아주대학교 시스템공학과

<sup>2</sup>한국철도기술연구원 첨단물류시스템연구실

<sup>3</sup>(주)노바 기업부설연구소

<sup>4</sup>서울교통공사 도시철도연구원

## On the Path Generation Algorithm for AGV and Design of Integrated Control System for Urban Railway Logistics System

Sunwoo Hwang<sup>1</sup>, Joouk Kim<sup>2+</sup>, Youngmin Kim<sup>1</sup>, Junwoo Choi<sup>3</sup>, Sumin Han<sup>3</sup>,

Sangwon Park<sup>3</sup>, Gyungho Jang<sup>4</sup>

<sup>1</sup>Department of Systems Engineering, Ajou University /

<sup>2</sup>Advanced Logistics System Research Department, Korea Railroad Research Institute /

<sup>3</sup>Research Institute Global Nova / <sup>4</sup>Research Institute, Seoul Metro

Due to the revitalization of the e-commerce market and the development of Internet communication technology, the demand for logistics is increasing worldwide. Due to the increase in logistics demand, the inflow of cargo trucks in large cities has increased, causing traffic congestion and air environment pollution. We presented an urban railway logistics system to solve these problems. The urban railway logistics system is a new concept including vertical/horizontal transport devices for logistics. In this study, a system for integrated control of the vertical/horizontal transportation device of the urban railway logistics system was proposed. In addition, route generation research for logistics automation was conducted by using the Automated Guided Vehicle as a horizontal transfer device. The integrated control system derived as a result of this study can contribute to the operational reliability and stability of the urban railway logistics system.

**Keywords:** Urban Railway Logistics System, Integrated Control System, Route Generation, Automated Guided Vehicle, Logistics Automation

논문접수일 : 2022.09.30.

심사완료일 : 2022.12.20.

게재확정일 : 2022.12.20.

이 논문은 국토교통과학기술진흥원의 고부가가치 융복합 물류 배송·인프라 혁신기술개발 사업의 지원을 받아 수행되었음. (21HCLP-C163194-01)

† Corresponding Author: jookim@krri.re.kr



## 1. 서론

### 1.1 연구배경 및 목적

이커머스 시장의 활성화 및 인터넷 통신 기술의 발전으로 인하여, 전세계적으로 물류에 대한 수요가 증가하고 있다. 물류 수요 증가로 인해, 대도시의 화물차 유입이 증가하였으며, 교통체증과 대기 환경오염을 유발하고 있다. 이러한 문제는 대도시에서 큰 사회적 비용을 일으키는 문제로 간주되고 있다. 이에 따라, 우리는 이러한 문제를 해결하기 위해 대도시에 적용할 수 있는 도시철도 물류 시스템을 제시하였다. 도시철도 물류 시스템은 수직/수평이송장치를 포함하고있는 신개념 물류 시스템이다. 도시철도 물류 시스템은 수직/수평이송장치를 서브 시스템으로 포함하고 있으며, 이외에, 화물열차, 운영시스템 등의 서브 시스템도 포함하고 있다. 도시철도 물류 시스템은 기존에 사용되고 있는 철도 인프라를 활용하는 신개념의 물류 시스템이다. 또한, 도시철도 물류 시스템은 시스템은 규모가 방대한 만큼 정밀한 시뮬레이션을 수행하기가 어렵다. 이러한 문제로 인하여, 시스템 운영상의 안전성과 신뢰성을 확보하기 위해서는 동적 행위를 하는 서브 시스템과 컴포넌트들을 제어할 수 있는 통합관제시스템이 필요하다. 본 연구에서는 도시철도 물류 시스템의 수직/수평이송장치를 제어할 수 있는 통합관제시스템을 제안하였다.

### 1.2 관련 선행연구

본 절에서는 도시철도 물류 시스템의 수직/수평이송장치 통합관제시스템 설계를 위한 선행연구 분석을 수행하였다. 이용환 등은 AGV(Automated Guided Vehicle) 그리드 단위의 제어에 기반한 자동화 터미널의 AGV 시스템 운영 방안 연구를 수행하였다(Lee et al., 2003). 본 연구는 AGV의 생산성을 고려한 자동화 터미널의 구조를 제시하고, 운행의 유연성을 위해 그리드 기반의 제어 방식을 제안하였다. 최종적으로, 실제 현장과 유사한 작업을 할 수 있고, 운영규칙의 수정 개선이 용이한 시뮬레이션 환경을 개발하였다. 김정민 등은 퍼지와 비례 제어를 이용한 지게차 AGV의 주행제어 연구를 수행하였다(Kim et al., 2009). 본 논문은 팔레트를 하역, 운송해야 하는 지게차 AGV의 주행제어를 위해 팔레트와 지게차 AGV 사이의 거리 차와 각도 차를 바탕으로 퍼지 제어 및 비례 제어를 이용한 주행제어기를 설계하였다. 실험 결과, 시뮬레이션에 의해 생성된 경로와 실제 주행 경로의 최대 평균 오차가 87.77mm를 가지는 것으로 도출되었다. 우승범 등은 중량물 운송을 위한 AGV의 주행 제어 방법 연구를 수행하였다(Woo et al., 2010). 본 논문에서 제안한 주행 제어 방법의 실험은 팔레트 앞 4m 지점부터 직선 주행 후에 팔레트 하역 작업을 수행하도록 하였고, 총 10회 실험 후에 이들의 정밀도를 분석하였다. 실험 결과, 팔레트 하역 시 목표 지점에 대한 정지 정밀도의 최대 오차가 18.64mm 이내로 높은 정지 정밀도의 안정적인 주행 제어가 가능함을 확인할 수 있었다. 전해경 등은 이중 랜드마크 인식 기반 AGV 이동 제어 연구를 수행하였다(Jeon and Hong, 2012). 본 논문은 바닥에 유도라인을 매립하는 방식은 특수한 환경에 적합하지 않기 때문에, 적외선 센서 기반 AGV 이동 궤적 제어 방식을 제안했다. 본 논문에서 제안한 방식은 모의 차량에 적용하여 이에 대한 타당성을 검증하였으며, 실내 특수한 환경에서 적용한 결과, 오차 범위 내에서 작동함을 확인하였다. 탁한호 등은 무게중심법을 이용한 무인 운반차(AGV)의 운전제어 연구를 수행하였다(Tack and Kwon, 2014). 본 연구에서는 AGV 주행 흔들림 개선과 공장의 특성에 따라 자유롭게 변경이 가능한 PLC 연동형 AGV 전용 제어 시스템의 설계 방법을 제안하였다. 실험을 통하여 여러 가지 주행 성능을 분석한 결과, 제안된 AGV 시스템이 운영상에 효율적임을 알 수 있었다. 이주원 등은 PLC 내장형 무인 반송차(AGV) 제어기 설계 연구를 수행하였다(Lee and Lee, 2019). 본 연구에서는 소형화, 용이한 재구성을 만족시키기 위하여, PLC 내장형 AGV 제어기의 설계법을 제안하고, 그 성능을 평가하였다. 실험 결과, 속도제어 오차 0.021 m/s와 주행 자세 제어의 평균오차 2.1mm의 성능을 검증하였다. 또한, 이주원 등은 퍼지 추론 시스템에 기반한 AGV 주행 제어기 설계 연구를 수행했다(Lee and Lee, 2020). 본 논문에서는 주행 오차의 누적값과 속도값을 이용한 PD 제어기와 퍼지 제어기 설계법을 제안하고, 산업현장에서 사용되고 있는 AGV에 적용하여 성능을 평가하였다. 실험 결과, 직선주행과 곡선주행에 있어서, PD 제어기보다 10배 이상의 주행 안전성을 보였으며,

이 설계법을 엔코더리스형 AGV 제어에 사용하면 높은 주행 안전성을 나타낼 것이라는 결론을 도출하였다.

경로 생성과 관련된 선행연구에서, 정초희 등은 AR(Augmented Reality) 기반의 증강현실 도보 내비게이션 앱 개발에서 Dijkstra Algorithm 적용 연구를 수행 하였으며(Jeong and Lee, 2017), 최중해 등은 홀 센서와 Dijkstra Algorithm을 이용한 로봇의 실내 주행과 구현 연구에서 최단거리 알고리즘의 종류로, 경험 및 휴리스틱 추정값을 활용하는 A\* 알고리즘과 모든 경로를 계산한 후 최단 경로를 선택하는 Dijkstra 알고리즘을 고려하였고, 무수히 많은 중간지점이 존재하기 때문에, Dijkstra 알고리즘을 활용하였다. 본 논문에서도 무수히 많은 경우의 수를 가진 중간지점 특성을 고려하여 Dijkstra 알고리즘을 선정하였다. 본 연구에서 활용하고자 하는 수평이송장치는 기존의 AGV를 화물운송 표준용기 및 화물열차 출입문의 물리적 형상에 맞게 개량한 기계장치로써, 관련된 선행연구는 제시한 주행 제어 설계 기법, 경로 인식 방안 등의 관점에서 검토되었다. 검토 결과, 본 연구에서 활용하고자 하는 수평이송장치는 물리적 환경, 특성에 맞게 새로운 개념으로 제시되는 기계장치이기 때문에, 경로 생성에 대한 직접적인 비교는 어려웠으며, 상용화하기 위해 제작되는 AGV 개발의 핵심 사항에 대한 사안을 확인하였다. 본 연구에서는 경로 생성에 Dijkstra Algorithm을 활용하였으며, 경로 인식 신뢰성이 높은 QR 방식의 경로 인식 방안을 채택하였다.

### 1.3 논문의 구성 및 흐름

본 논문은 1장 서론에서 통신 기술의 발전으로 인한 이커머스 시장의 활성화와 물류 시장의 수요 증가를 언급하며, 이에 따른 사회적 비용 문제에 대하여, 본 연구의 배경 및 필요성을 기술하였다. 또한, 본 논문에서 궁극적으로 개발하고자하는 수평이송장치의 주행 제어 설계 기법, 경로 인식 방안 등의 관점에서 관련 선행연구 분석을 수행하였다. 2장에서는 도시철도 물류 시스템에 대한 개념과 이를 구성하고 있는 수직/수평이송장치에 대한 핵심 기능을 도출하였다. 3장에서는 통합관제시스템의 구조화 및 기능 정의를 통하여, 통합관제시스템의 전반적인 개념을 설명하고, 통합관제시스템의 운용에 필요한 인터페이스 사항을 식별하였다. 도시철도 물류 시스템은 화물운송 표준용기와 체결된 수평이송장치의 화물 운송 기능이 핵심 기능으로 분류될 수 있다. 즉, 수평이송장치의 경로가 제대로 생성되지 않을 경우, 도시철도 물류 시스템의 화물 운송 기능이 성공적으로 구현될 수 없다. 따라서, 수평이송장치의 경로 생성 알고리즘 연구는 핵심 기능으로 분류될 수 있다. 따라서, 4장에서는 수평이송장치 경로 생성 알고리즘의 핵심 요소인 수평이송장치의 이동 속도 예측, 선정 계획의 상세, 경로 계획의 상세 부분을 기술하였다. 마지막으로, 5장에서는 도시철도 물류 시스템의 수직/수평이송장치 통합관제시스템 설계와 수평이송장치 경로 생성 알고리즘 연구 수행 내용을 정리하였으며, 추후, 본 연구 내용이 가지는 의의를 기술하였다. 본 논문은 도시철도 물류 시스템의 운용상 필요한 수직/수평이송장치의 통합관제시스템의 설계와 수평이송장치의 경로 생성 알고리즘 연구를 목적으로 수행되었다. <Figure 1>은 본 논문의 구성 및 전개 흐름을 도시한다.

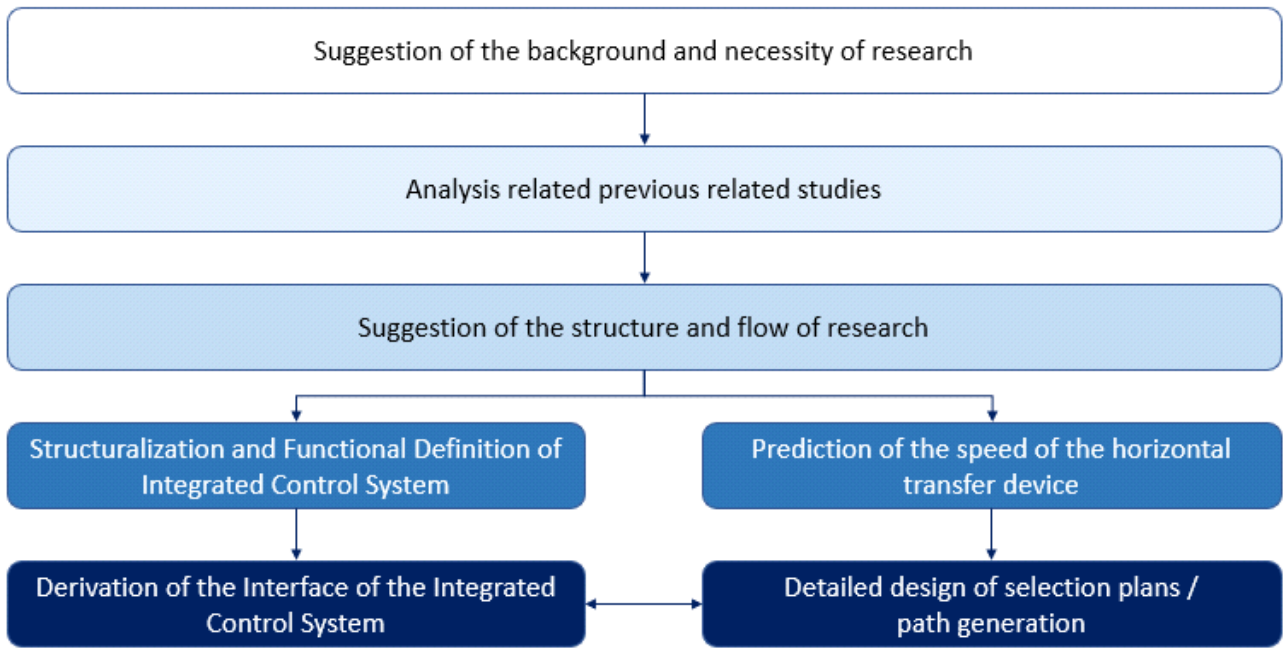


Figure 1 : Structure and flow of research

## 2. 도시철도 물류 시스템

### 2.1 수직/수평이송장치의 개념과 필요기능 정의

도시철도 물류 시스템은 수직/수평이송장치를 포함하는 신개념의 물류 시스템이다(Hwang et al., 2021). 또한, 수직/수평이송장치는 도시철도 물류 시스템에 있어서, 화물을 운송할 수 있는 핵심 시스템이다. 먼저, 도시철도 물류 시스템의 특성에 맞는 수직이송장치의 요구사항과 개념을 정리하였다. 수직이송장치의 주요기능은 12가지로 도출되었다. 수직이송장치는 다수의 화물이 담긴 화물운송 표준용기를 이송하는 수평이송장치를 수직으로 층간 이송하기 위한 수직 방향 작동이 가능해야 한다. 이를 위해, 발생하는 하중을 견딜 수 있는 설계가 적용된 구조적 안전성을 가져야 한다. 또한, 수직 이송을 위해 구동부 및 각 장치에 전력을 공급하는 기능을 갖춰야 하며, 수직이송장치의 장비와 기능들의 제어 기능이 있어야 한다. 이를 위해서는 원격 제어 외 실제 장비 제어가 가능한 장비제어 패널이 필요하다는 것을 도출하였다. 수평이송장치의 진입 순서, 목적지 정보, 수직이송장치의 상태 정보 등을 통신하기 위하여, 정보의 통신 기능을 가져야 한다. <Table 1>은 이에 대한 내용을 나타낸다.

Table 1. Function of vertical transfer device

No	Key function	Contents
1	Vertical drive function	<ul style="list-style-type: none"> <li>- Vertical operation for vertical interlayer transfer of horizontal transport</li> <li>- Operates via power from the drive, maximum speed of 1.0 m/s</li> </ul>
2	Cargo transfer function	<ul style="list-style-type: none"> <li>- Transfer plate function for vertical interlayer transfer of horizontal transfer unit</li> <li>- Maximum speed of the drive for interlayer transfer is 1.0 m/s</li> </ul>
3	Power control function	<ul style="list-style-type: none"> <li>- Ability to supply power to the drive unit and each unit for vertical transfer</li> <li>- Ability to control the electrical energy connected to the power system for the operation of the device</li> </ul>
4	Load support function	<ul style="list-style-type: none"> <li>- Structural performance features designed to withstand loads generated by vertical interlayer transfer of horizontal transfer devices</li> </ul>
5	Equipment control function	<ul style="list-style-type: none"> <li>- Control functions of vertical transport devices and functions</li> <li>- Equipment control panel with actual equipment control in addition to remote control is required</li> </ul>
6	Communication function	<ul style="list-style-type: none"> <li>- Transmission and reception of information required to vertically transfer horizontal transfer devices between floors</li> <li>- Entry order, destination (floor) information, etc. of horizontal transport devices</li> <li>- Ability to transmit status information and receive relevant information from the vertical transport device</li> </ul>
7	Recognition function	<ul style="list-style-type: none"> <li>- Ability to locate mounting when transferring horizontal transport vertically between floors</li> </ul>
8	Guideline function	<ul style="list-style-type: none"> <li>- Ability to guide the boarding path for horizontal transport to be brought into the vertical transport</li> </ul>
9	Horizontal Transfer Device Binding Function	<ul style="list-style-type: none"> <li>- Function of securing the horizontal transport on the transport plate of the vertical transport device so that it does not deviate</li> <li>- Function of preventing the horizontal transfer device from being dislodged is binding upon boarding the horizontal transfer device, and the binding must be released upon disembarkation</li> </ul>
10	Emergency stop function	<ul style="list-style-type: none"> <li>- Ability to stop the vertical transfer device urgently with the emergency stop button located at the entrance of the unit if it is abnormal or if an emergency stop is required</li> <li>- Emergency stop buttons are located in the upper and lower entrances</li> </ul>
11	Operation display function	<ul style="list-style-type: none"> <li>- Function of displaying the operation of the device through a visual display(SCREEN) or indicator at the upper and lower input ports of the vertical transfer device</li> </ul>
12	Connection function	<ul style="list-style-type: none"> <li>- Function to facilitate smooth movement of horizontal transport device when entering and exiting vertical transport device</li> <li>- Automatic or fixed application to assist horizontal transfer device movement</li> </ul>

수평이송장치의 주요기능은 12가지로 도출되었다. 수평이송장치는 목적지로의 이동 및 다수의 화물이 담긴 화물운송 표준용기를 이송하기 위한 수평 방향 작동이 가능해야 한다. 이는 구동부의 동력과 4개의 휠을 통한 전진, 후진 및 회전 기능이다. 또한, 수평 이송을 위해 구동부 및 각 장치에 전력을 공급하는 기능을 가져야 한다. 장치의 작동을 위해 자체 배터리에 충전된 전기 에너지를 제어하는 기능이 포함된다. 또한, 고중량의 하중을 견딜 수 있는 설계가 적용된 구조적 안전성을 가져야 한다. 수직이송장치와 마찬가지로, 원격 제어 외 실제 장비 제어가 가능한 장비 제어기능이 필요하다. 추가로, 수평이송장치의 상태 정보 송신 및 관련 정보 수신 기능 등 수평이송장치의 상태 정보 등을 통신하기 위하여, 정보의 통신 기능을 가져야 한다. <Table 2>는 이에 대한 내용을 나타낸다.

Table 2. Function of vertical transfer device

No	Key function	Contents
1	Vertical drive function	<ul style="list-style-type: none"> <li>- Horizontal operation for moving to a destination and carrying standard cargo containers containing multiple cargoes</li> <li>- Power of the drive and the ability to move forward, reverse and rotate through the four wheels</li> </ul>
2	Cargo transfer function	<ul style="list-style-type: none"> <li>- Ability to transport standard cargo containers containing multiple cargoes to the destination</li> </ul>
3	Power control function	<ul style="list-style-type: none"> <li>- Ability to supply power to the drive unit and each unit for horizontal transfer</li> <li>- Ability to control the electrical energy charged in its own battery for the operation of the unit</li> </ul>
4	Load support function	<ul style="list-style-type: none"> <li>- Structural performance features designed to withstand heavy loads to transport standard containers containing a large number of cargoes</li> </ul>
5	Equipment control function	<ul style="list-style-type: none"> <li>- Control functions of horizontal transport devices and functions</li> <li>- In addition to remote control, equipment control function that allows actual equipment control is required</li> </ul>
6	Communication function	<ul style="list-style-type: none"> <li>- Ability to transmit and receive information necessary to transport standard containers containing multiple cargoes</li> <li>- Ability to transmit status information and receive relevant information from the horizontal transport device</li> </ul>
7	Recognition function	<ul style="list-style-type: none"> <li>- Horizontal transfer device carrying standard cargo containers containing a large number of cargoes is capable of recognizing guidelines on the floor to move to a designated destination</li> </ul>
8	Rolltainer binding function	<ul style="list-style-type: none"> <li>- Function of fastening and securing the standard container for freight transport containing 500 kg of cargo to the horizontal transport device so that it does not deviate during transport</li> </ul>
9	Train binding function	<ul style="list-style-type: none"> <li>- Horizontal transfer on a rotary transfer plate or freight train of a vertical transfer device</li> <li>- Ability to hold the device so that it does not deviate</li> <li>- Function of preventing the horizontal transfer device from being dislodged is binding upon boarding the horizontal transfer device, and the binding must be released upon disembarkation</li> </ul>
10	Emergency stop function	<ul style="list-style-type: none"> <li>- Function of stopping the horizontal transfer device urgently with the emergency stop button located on the surface of the device when the operation of the horizontal transfer device is abnormal or emergency stop is required</li> <li>- Emergency stop buttons are located one per unit or per surface</li> </ul>
11	Operation display function	<ul style="list-style-type: none"> <li>- Ability to display relevant information through a display or indicator that visually displays the operation of the device during horizontal transfer of the horizontal transfer device</li> </ul>
12	Lifting function	<ul style="list-style-type: none"> <li>- Function of lifting the standard cargo container on the ground through its own lifting function to prevent movement problems due to the support of the container when transferring the standard cargo container of the horizontal transport device</li> </ul>

도시철도 물류 시스템의 수직/수평이송장치는 제한된 환경에 설치되어 기능을 수행해야 한다는 한계성이 존재하며, 개별 화물에 대한 이송이 아닌 화물과 장치의 결합체를 수직/수평으로 이송해야 한다는 특수성이 존재한다. 즉, 기능적으로 수직/수평 이송을 통한 화물을 이송한다는 점은 동일하지만, 그 외 장치에 요구되는 부가기능, 운영기술, 환경 적합성 설계 등이 다르기에, 관련 특허 조사 및 국내/외 사례 분석 결과를 바탕으로 도시철도 물류 시스템에서 제안하고자하는 수직/수평이송장치는 기존의 수직/수평이송장치와 차별성과 특수성을 가진다는 점을 도출하였다. 이에 대한 내용은 <Table 3>에 나타내었다.

Table 3. Deriving the specificity of vertical/horizontal transfer device development  
for urban railway logistics system

Functional Specificity	Technical specificity	Environmental specificity
Raise and lower transfer must be possible	Control and control of the device's up and down transfer shall be possible	Must be installed in a space within a limited station
Vertical/horizontal transport device must be fixed when transported	Vertically/horizontally transport device must be capable of recognizing boarding and controlling the binding function	Should not be affected by vibrations generated by train transfers in stations
Weight of the cargo, the standard container for freight transport, and the vertical/horizontal transport device must be all covered	Device control linked to entry and boarding of vertical/horizontal transport devices shall be possible	Designed and installed to avoid structural strain on the installation environment
Multiple vertical/horizontal transfer devices must be transported in either ascending or descending direction	Information related to the location of the vertical/horizontal transfer device must be transmitted and received	Noise and vibration should be minimized when transporting vertical/horizontal transport devices
Position of the vertical/horizontal transport device to be transferred shall be recognized	Radio control and control through the control system shall be possible	Impact of operating the device should be minimized

### 3. 통합관제시스템 설계

#### 3.1 통합관제시스템의 구조화 및 기능 정의

통합관제시스템은 수직/수평이송장치의 동작을 감지하고, 상황에 따라, 동작을 하계끔 제어하는 시스템이다. 4차 산업혁명으로 무인 통신 기반의 로봇을 활용하는 통합 시스템이 급증하고 있는 추세이다. 이러한 시점에서 통합관제시스템은 어떠한 목적에 맞게, 수행하는 로봇들을 제어하는 중요한 관제탑 역할을 수행한다. 통합관제시스템을 개발한 사례로써, 재난환경에서 자율탐색을 위한 ROS(Robot Operating System) 기반의 다중로봇 통합관제시스템이 제시된 바 있으며(Yang et al., 2019), 불안정한 통신 문제를 해소하기 위하여, 다중 ROS 코어 기반의 통합관제시스템을 제시하였다(Yang et al., 2021). 이러한 사례를 검토함으로써, 본 논문에서는 도시철도 물류 시스템의 특성에 맞는 새로운 개념의 수직/수평이송장치 통합관제시스템을 제시하였다.

본 논문에서는 S/W 구성항목에 대한 구조를 분류하여, 통합관제시스템 설계에 기준을 정의하였다. 통합관제시스템의 설계 기준은 다음과 같다. 1. 요구사항 분석활동에서 정의 및 산출되었던 산출물과 추가로 작성되거나 보충되는 자료를 바탕으로, 분할되어 구성된 시스템 요소들, 품질속성을 만족하는 아키텍처를 기술하였다. 2. 추가 연구개발 활동에서 수행하는 요구사항 분석 활동과 검증활동을 고려하여 수행하였다. 3. 분석결과에서 제시된 품질속성을 기반으로 시스템을 분할하여 아키텍처를 설계하였다. 4. 설계된 산출물을 기반으로 아키텍처 검증 활동을 수행하였다. 통합관제시스템의 S/W는 AMQP(Advanced Message Queuing Protocol) 처리기를 통하여, 기본적으로, 수직/수평이송장치와의 정보를 통신해야 한다. 또한, 운영시스템에 데이터를 전달하기 위해 운영시스템 API 처리기를 포함한다. 이에 대한 모니터링은 웹/앱 인터페이스 처리기를 통하여 명령을 수행할 수 있다. 본 연구에서는 수평이송장치의 경로 생성과 그에 따른 통합관제시스템과의 데이터 통신을 주 목적으로 수행되었다. 수평이송장치와 데이터 통신을 수행하는 통합관제시스템의 개념을 <Figure 2>에 나타내었다.

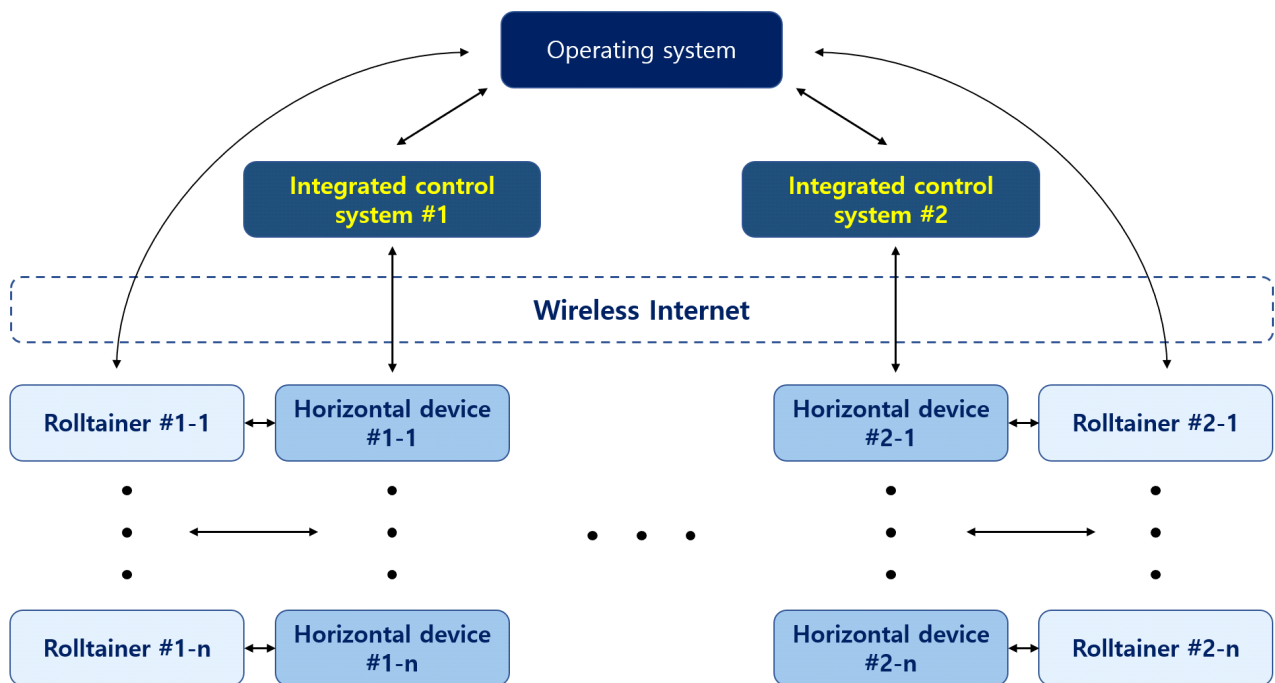


Figure 2. Configuration diagram of operating system and integrated control systems

도시철도 물류 시스템의 수직/수평이송장치를 제어하기 위해서는 통합관제시스템이 필수적이다. 수직/수평이송장치 제어 메시지 생성 및 전송, 제어 S/W 원격조정 등 통합관제시스템은 원격을 통한 관제서버 데이터 상태 정보 모니터링 및 수직/수평이송장치를 제어할 수 있게끔 설계되어야 한다. <Table 4>는 통합관제시스템의 구조화와 요구사항 도출 결과이다. <Figure 3>은 구조화 및 요구사항 도출 결과에 따른 수평이송장치 관제 모니터링 및 제어 아키텍처를 나타낸다.

Table 4. Integrated control systems structure and requirements

Classification	Contents
Control Server Monitoring Handler	<ul style="list-style-type: none"> <li>- Process periodic updates of data received by the control server</li> <li>- Updated data is visualized in the UI (User Interface)</li> </ul>
Horizontal Transfer Device Control Message Handler	<ul style="list-style-type: none"> <li>- Forward horizontal transfer device control messages through the control server</li> <li>- Run control of horizontal transfer device</li> <li>- Processing of real-time control messages through AMQP</li> </ul>
Rolltainer Position/QR Recognizer	<ul style="list-style-type: none"> <li>- Communicate information through location markers (QR codes)</li> </ul>

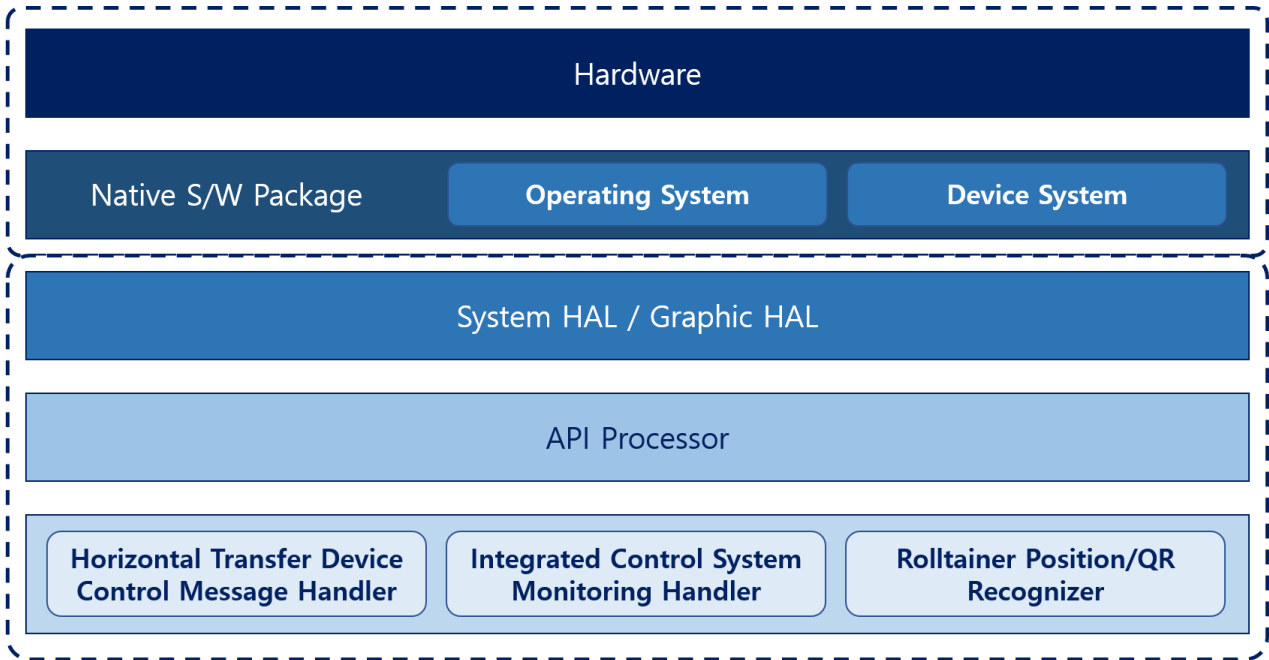


Figure 3. Vertical/horizontal transfer device control monitoring and architecture

### 3.2 통합관제시스템의 인터페이스 도출

통합관제시스템은 운영시스템에서 필요한 정보를 추출하여 서버에 전송하고, 모든 수직/수평이송장치의 데이터는 상태가 변경될 때마다 실시간으로 통신되어야 한다. 이를 수행하기 위해서는 AMQP 처리기, 운영시스템 API(Application Programming Interface) 처리기, 수직/수평이송장치 연결 처리기 및 상태 정보 처리기, 라우팅 처리기, 웹/앱 인터페이스 처리기 등으로 구성된다. AMQP 처리기는 통합관제시스템에서 수평/수직이송장치에 제어 메시지를 효율적으로 전달하기 전달체이다. 통합관제시스템과 수직/수평이송장치 간의 모든 데이터는 AMQP를 이용하여 통신한다. 중점적으로는 이동 중 변경되는 위치정보, 배터리 잔량 등을 실시간/양방향으로 통신이 가능해야 한다. 운영시스템 API 처리기는 이동준비가 완료된 화물운송 표준용기의 정보를 수신하고, 통합관제시스템에서 수평이송장치를 연결하는 과정을 처리한다. 이를 수행하기 위해서는 화물운송 표준용기의 정보(화물운송 표준용기 ID, 목적지, 차량번호)가 필요하다. 운영시스템 API 처리기는 이동 준비가 완료된 화물운송 표준용기의 정보를 수신할 수 있어야 하며, 해당 차량기지에 준비된 화물운송 표준용기의 정보만을 수신해야 한다. 예를 들어, 어떠한 차량기지에 대한 요청인지를 구분하여, 해당 차량기지의 준비된 화물운송 표준용기의 정보만을 수신해야 한다. 화물역사 내에서 관리해야할 모든 수직/수평이송장치에 대한 모니터링 및 제어 처리가 가능해야 한다. <Figure 4>는 통합관제시스템과 관련 시스템의 인터페이스 관계도를 나타낸다.



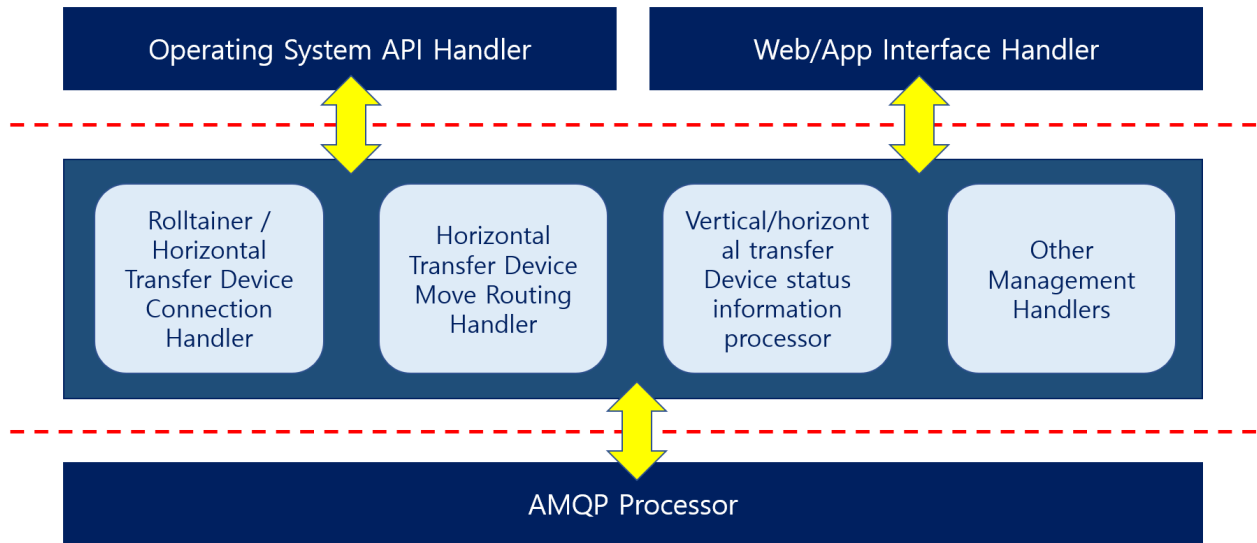


Figure 4. Configuring the integrated control systems interface

본 논문에서는 도시철도 물류 시스템의 수직/수평이송장치 통합관제시스템의 요구사항을 분석하고, 이에 대한 기능을 정의하였다. 또한, 설계에 필요한 부분을 정의하여, 성공적으로 통합제어시스템이 개발될 수 있도록, 정형화된 설계 기준을 제시하였다. 도시철도 물류 시스템의 수직/수평이송장치 통합관제시스템은 각각의 차량기지별, 화물역사별로 설치되어야 하며, 수직/수평이송장치의 상태 모니터링 및 제어 기능을 처리할 수 있어야 한다. 또한, 상태 모니터링에 대한 결과를 운영시스템 API를 통하여 운영시스템과 실시간으로 통신할 수 있어야 한다. 이를 수행하기 위하여, CPU: Intel CORE i5-11세대(11400F) 이상, Memory: 32GB 이상, 스토리지: 4TB(RAID 구성) 이상의 H/W 스펙이 필요하다고 판단된다.

#### 4. 수평이송장치 경로 생성 알고리즘

도시철도 물류 시스템은 화물운송 표준용기와 체결된 수평이송장치의 화물 운송 기능이 핵심 기능으로 분류될 수 있다. 즉, 수평이송장치의 경로가 제대로 생성되지 않을 경우, 도시철도 물류 시스템의 화물 운송 기능이 성공적으로 구현될 수 없다. 따라서, 수평이송장치의 경로 생성 알고리즘 연구는 핵심 기능으로 분류될 수 있다. 본 논문에서는 도시철도 물류 시스템의 화물 운송 기능 확보를 위하여, 서브 기능이라고 판단할 수 있는 수평이송장치의 경로 생성 알고리즘 연구를 목적으로 수행되었다. 수평이송장치의 표준 용기 체결, 열차 탑승 및 하차, 표준 용기의 반환 및 충전 대기까지의 과정에서 발생하는 모든 이동은 경로 계획 알고리즘에서 정해진 경로 계획을 따른다. 경로 계획 알고리즘은 다음 <Figure 5> 및 <Figure 6>에 의해 설명되는 과정을 통해 화물의 적재 및 탑승 과정과 하차 과정에 필요한 계획을 도출하였다.

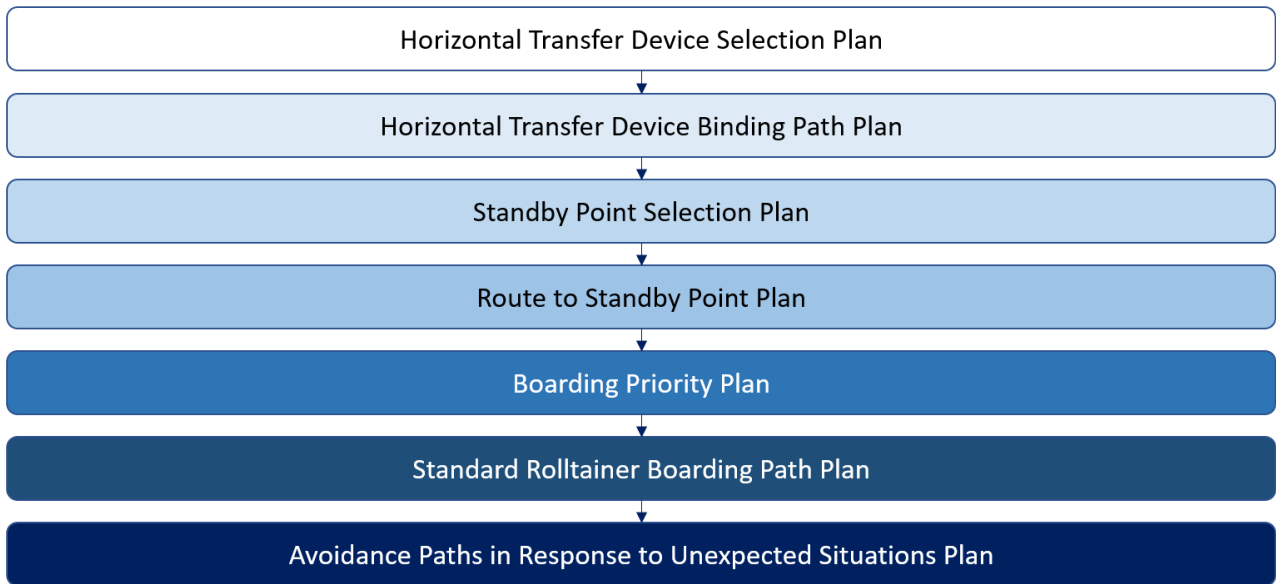


Figure 5. Flow of route planning used in the loading and boarding process

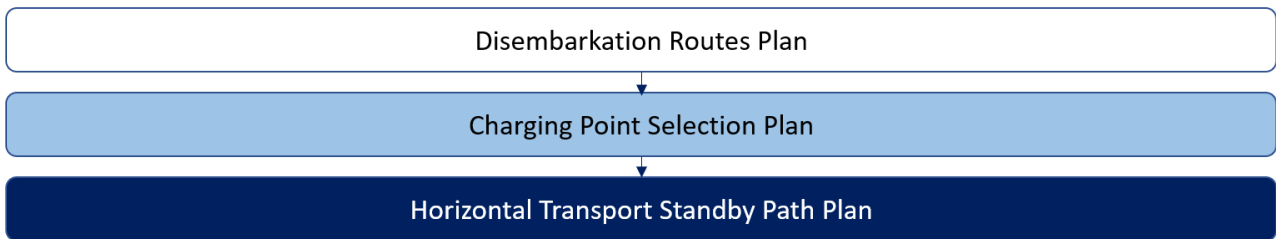


Figure 6. Flow of drop-off process route plan

적재, 탑승 및 하차 프로세스에서 수립되는 계획의 상세는 다음 <Table 5>와 같다.

Table 5. List of plans in the loading, boarding and unloading processes

Establishment Plan	Contents
Horizontal Transfer Device Selection Plan	- Plan to select horizontal transfer devices to be engaged when a new cargo container is introduced
Horizontal Transfer Device Binding Path Plan	- Path planning for horizontal transfer device movement in accordance with the tightening direction of cargo container
Standby Point Selection Plan	- Plan to select a point at which horizontal transfer devices engaged with a cargo container and loaded with cargo will wait in the vehicle base before boarding the train
Route to Standby Point Plan	- Route planning for moving horizontal transport to selected waiting points
Boarding Priority Plan	- Ranking plan to determine the order in which a train travels to the destination of a freight container arrives

Standard Rolltainer Boarding Path Plan	- Train boarding route plan to ensure that the train arrives at a fixed rank and to complete its boarding quickly
Avoidance Paths in Response to Unexpected Situations Plan	- Path planning to achieve the original goal in accordance with the changed circumstances, if various unexpected situations occur during the movement of the horizontal transfer device, do not follow the original travel plan
Disembarkation Routes Plan	- Planning the horizontal transfer device movement path after the train arrives and when getting off
Charging Point Selection Plan	- Plan to select the point at which the horizontal transfer device will stand by and be charged after the horizontal transfer device and the freight container are disengaged
Horizontal Transport Standby Path Plan	- Path planning for horizontal transport device to the selected low point

#### 4.1. 수평이송장치의 이동 속도 예측

계획 과정에서는 차량의 속도 및 이동에 걸리는 시간이 중요 의사결정 요소로 활용된다. 이동 시간은 배경이 되는 차량기지 및 역사의 상태와 표준 용기에 실린 화물의 총무게 및 무게 중심 등에 따라 달라지므로, 이를 실제 이동 전에 예측할 필요가 있다. 본 경로 계획에서는 다음과 같은 과정을 통하여 차량의 이동에 걸리는 시간을 예측한다. 우선 각 수평이송장치가 표준 용기와 결합한 직후의 이동 시간과 지점별 이동 시간을 기록하여 수평이송장치의 이동 시간 감소 비율을 측정하고, 평균하여 지점의 이동 시간 감소 비율을 산출한다. 이후, 새로운 수평이송장치의 경로 계획을 수립할 때는 기존의 이동 시간 감소 비율을 통해 이동 시간을 추측하였다.

#### 4.2. 선정 계획의 상세 설계

수평이송장치의 표준 용기 체결, 열차 탑승 및 하차, 표준 용기의 반환 및 충전 대기까지의 과정에서 발생하는 모든 이동은 경로 계획 알고리즘에서 정해진 경로 계획을 따르기에 무척 중요하다. 또한, 경로 계획을 세우기 위하여, 선정 계획의 상세 설계가 진행되어야 한다. 본 경로 계획 프로세스의 계획 알고리즘은 크게 우선순위 선정 알고리즘과 경로 생성 알고리즘으로 분류할 수 있다. 이 중, 선정 알고리즘은 다음 <Table 6>과 같은 요소를 고려하여 수행하였다. 선정 계획의 고려 요소의 수치를 0, 1 사이의 값으로 정규화하고, 이를 가중 합하는 과정을 거쳐 산출된 최종 평가치를 비교하는 과정을 통해서 이루어진다. 가중치는 고려 요소의 중요도를 고려하여 0과 1 사이의 값으로 정해지며, 해당 요소의 중요도를 고려해 정해진다.

Table 6. Requirements by algorithms

Algorithms	Requirements
Horizontal feed supplies, plan selection of algorithms	<ul style="list-style-type: none"> <li>- Distance between horizontal transfer unit and standard container</li> <li>- Amount of battery remaining on the horizontal transfer device</li> <li>- Fragmentation level of space predicted when using the point</li> <li>- Distance traveled to the waiting point</li> </ul>
Holding area for planning algorithm	<ul style="list-style-type: none"> <li>- Fragmentation level of space predicted when using the point</li> <li>- Distance traveled to the waiting point</li> </ul>
Train boarding priority planning algorithm	<ul style="list-style-type: none"> <li>- Disembarkation priorities on the train</li> <li>- Distance traveled by horizontal transfer device</li> <li>- the speed of movement of the horizontal transfer device</li> </ul>
Horizontal transfer device charging point selection planning algorithm	<ul style="list-style-type: none"> <li>- Amount of battery remaining on the horizontal transfer device</li> <li>- Fragmentation level of space predicted when using the point</li> <li>- Distance traveled to that point</li> </ul>

#### 4.3. 경로 생성 계획의 상세 설계

경로 계획은 가중치 기반 변형 Dijkstra 알고리즘을 통하여 작성된다. 본 연구의 경로 계획 알고리즘의 원본인 Dijkstra 알고리즘은 효율적으로 모든 가능한 경로를 비교하는 알고리즘으로써, 본 연구의 차량기지 와 같은 공간에 적용되기에 적합한 알고리즘이다. 해당 알고리즘은 다음 <Table 7>과 같은 과정을 통해 작동한다. 해당 의사 코드는 배경이 되는 경로 네트워크가 Graph, 출발지는 Source의 형태로 전달하여, Source에서부터 모든 지점까지의 최단 경로 및 거리를 도출하는 알고리즘이다. 데익스트라 알고리즘은 기본적으로 모든 가능한 경로를 모두 검토한다. 이 과정에서 비효율적인 경로는 빠르게 목록에서 제거되고, 효율적인 경로만이 지속해서 재사용 되도록 하여, 경로의 검토 과정을 최대한 효율적으로 수행하도록 한다. 해당 알고리즘은 가능한 모든 경로를 검토한다는 특성상, 배경 경로 네트워크의 복잡도가 커지면 사용이 힘들어진다는 단점이 있으나, 본 연구의 배경은 일반적인 도로가 아닌, 한정된 차량기지의 사각 격자 형태이므로, 구동에 필요한 시간이 매우 짧아 실제 수평이송장치의 구동 시 적용이 가능하다.

Table 7. Process of Dijkstra algorithm

```

function Dijkstra(Graph, Source):
    dist[source] ← 0
    prev[source] ← undefined
    create vertex set Q
    for each vertex v in Graph:
        if v ≠ source:
            dist[v] ← INFINITY
            prev[v] ← UNDEFINED
        add v to Q

    while Q is not empty:
        u ← vertex in Q with min dist[u]
        remove u from Q
        for each neighbor v of u:
            alt ← dist[u] + length(u, v)
            if alt < dist[v]:
                dist[v] ← alt
                prev[v] ← u
    return dist[], prev[]

```

본 경로 계획 프로세스 중 경로 생성 알고리즘 알고리즘은 다음 <Table 8>과 같은 특성을 가진다.

Table 8. Characteristics of path generation algorithms

Path Generation Algorithm	Characteristics
Horizontal Transfer Device Binding Path Plan	- Consideration of the tightening direction of the horizontal transfer device and the standard transfer container
Standby Point Selection Plan	- General Path Planning
Standard Rolltainer Boarding Path Plan	- Compliance with boarding priorities - Multiple vehicles moving simultaneously - Multiple Path Overlapping Occurred - Time taken is critical
Avoidance Paths in Response to Unexpected Situations Plan	- Plans to respond to environmental changes caused by unpredictable and unexpected situations - Consider existing boarding priorities as far as much as possible
Disembarkation Routes Plan	- Planning of the drop-off route considering the delay in the vertical transfer unit - Time taken is critical
Horizontal Transport Standby Path Plan	- General Path Planning

해당 알고리즘 중, 열차 탑승 경로 계획을 제외한 나머지 계획은 차량의 경로 겹침의 발생이 매우 드물게 발생하는 계획으로, 기존의 경로 계획을 적용하는 것이 가능하여 변형하지 않은 원본 Dijkstra 알고리즘을 적용하여 수행되었다. 반면 다수의 차량이 좁은 열차 탑승구를 통해 이동하므로 경로 겹침이 수시로 발생하며 탑승 우선순위를 반드시 지켜야 하는 열차 탑승 경로 계획과 이미 겹침이 발생하여 대응해야 하는 상황인 돌발 상황 대응 회피 경로 계획 알고리즘에는 변형된 Dijkstra 알고리즘을 적용한다. 변형된 Dijkstra 알고리즘은 다음 <Figure 7>과 같은 과정을 거쳐 동작한다. 해당 알고리즘에서는 충돌이 자주 발생할 것으로 예측되는 지점의 통과 비용을 일반적인 물리적 비용보다 더욱 비싸게 책정하는 방법을 통하여 꼭 해당 지점을 통과하여야 하는 차량이 아니라면 해당 지점을 피하는 경로를 택하도록 유도하는 방식이다. 비용의 변경은 해당 지점의 통과 횟수에 기반한 보정치를 곱하는 형태이다.

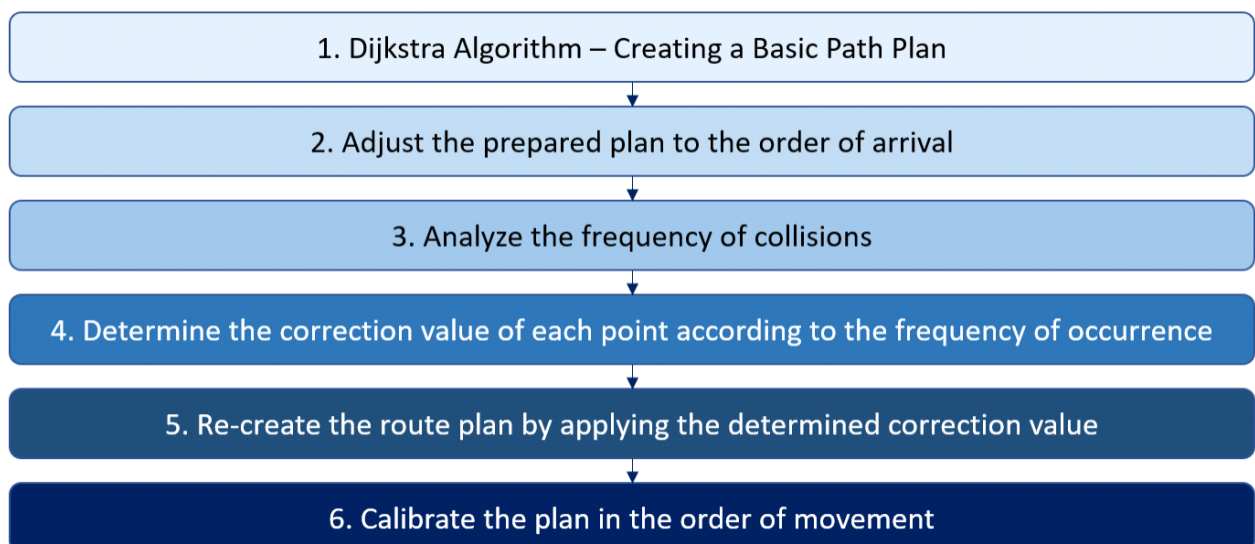


Figure 7 : Modified Dijkstra Algorithm

돌발 상황 대응 회피 경로 계획 알고리즘의 경우, 이미 충돌이 발생한 상황에서 이루어지므로, 남은 경로 계획을 분석하여 보정치를 재산정하되, 더욱더 강하게 바로잡도록 하여 더욱더 충돌 회피적인 경로 계획을 작성하도록 하였다. 해당 알고리즘은 시뮬레이터를 통해 구현되었으며, 이후 다양한 환경에서 검증을 거쳐 실제 관제 시스템에 탑재될 예정이다. 실제 구현 장면은 다음 <Figure 8>과 같다.

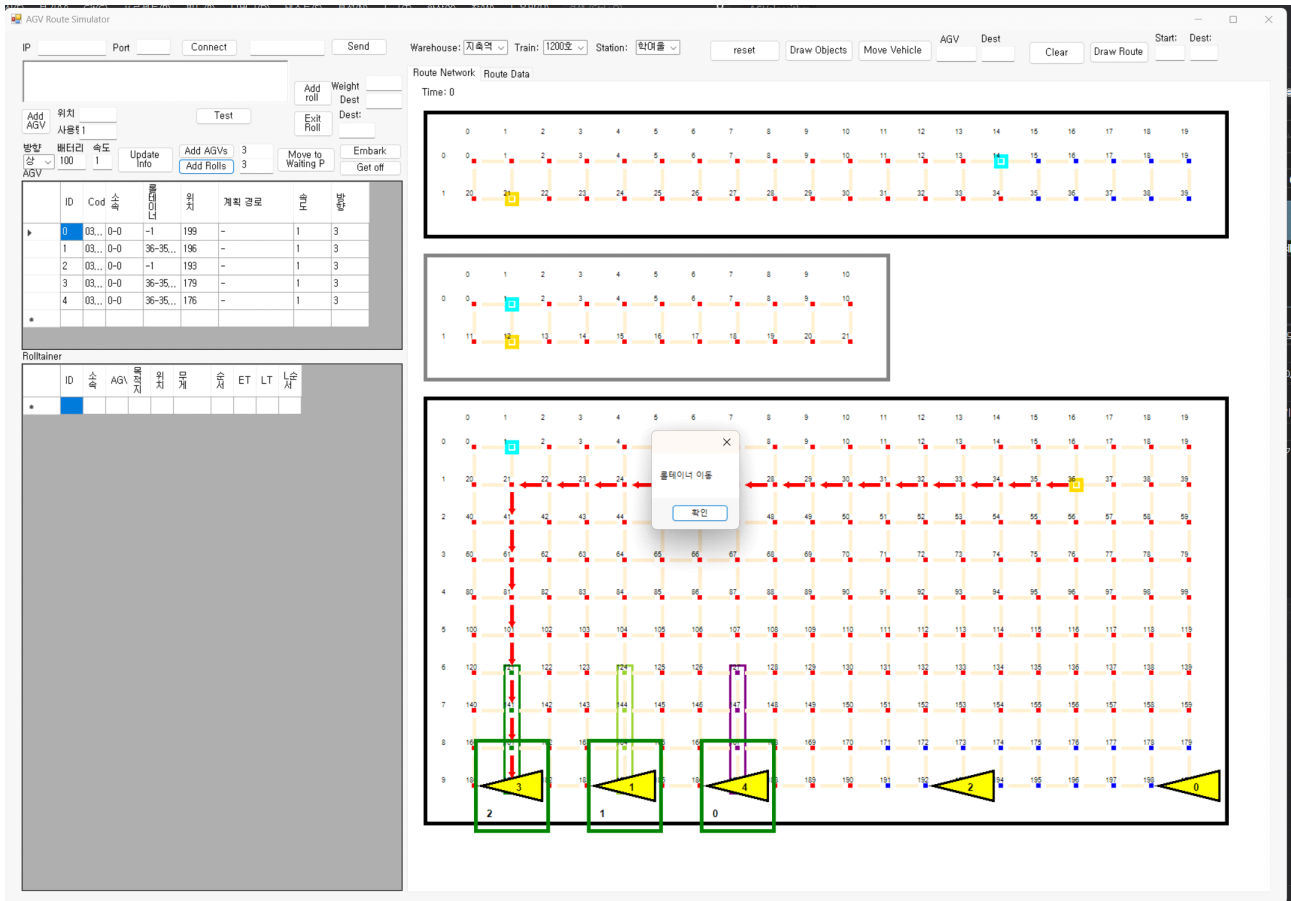


Figure 8. Implementation of algorithms through simulators

## 5. 결론

본 연구에서는 도시철도 물류 시스템의 수직/수평이송장치를 통합관제하는 시스템을 제안하였다. 도시철도 물류 시스템의 수직/수평이송장치를 제어하기 위해서는 통합관제시스템이 필수적이다. 본 연구에서 활용하고자 하는 수평이송장치는 기존의 AGV를 화물운송 표준용기 및 화물열차 출입문의 물리적 형상에 맞게 개량한 기계장치로써, 관련된 선행연구는 제시한 주행 제어 설계 기법, 경로 인식 방안 등의 관점에서 검토되었다. 검토 결과, 본 연구에서 활용하고자 하는 수평이송장치는 물리적 환경, 특성에 맞게 새로운 개념으로 제시되는 기계장치이기 때문에, 경로 생성에 대한 직접적인 비교는 어려웠으며, 상용화하기 위해 제작되는 AGV 개발의 핵심 사항에 대한 사안을 확인하였다. 본 연구에서는 기존의 사례와는 다르게, 경로 생성에 Dijkstra Algorithm을 활용하였으며, 경로 인식 신뢰성이 높은 QR 방식의 경로 인식 방안을 채택하였다.

수직/수평이송장치 제어 메시지 생성 및 전송, 제어 S/W 원격조정 등 통합관제시스템은 원격을 통한 관제서버 데이터 상태 정보 모니터링 및 수직/수평이송장치를 제어할 수 있게끔 설계되어야 한다. 이를 수행하기 위해서는 통합관제시스템은 AMQP 처리기, 운영시스템 API 처리기, 수직/수평이송장치 연결 처리기

및 상태 정보 처리기, 라우팅 처리기, 웹/앱 인터페이스 처리기 등으로 구성된다. 통합관제시스템과 수직/수평이송장치 간의 모든 데이터는 AMQP를 이용하여 통신한다. 중점적으로는 이동 중 변경되는 위치정보, 배터리 잔량 등을 실시간/양방향으로 통신이 가능해야 한다. 이를 수행하기 위해서는 화물운송 표준용기의 정보(화물운송 표준용기 ID, 목적지, 차량번호)가 필요하다. 본 논문에서는 도시철도 물류 시스템 운영에 필요한 수직/수평이송장치의 개념과 필요 기능을 정의하였고, 수평이송장치를 제어하기 위한 통합관제시스템에 대한 개념과 설계 기준, 기능을 정의하였다. 또한, 본 논문에서는 도시철도 물류 시스템의 화물운송 기능 확보를 위하여, 서브 기능이라고 판단할 수 있는 수평이송장치의 경로 생성 알고리즘 연구를 목적으로 수행되었다. 이를 통해, 또한, 본 연구의 결과로 도출된 통합관제시스템 및 수평이송장치 경로 생성 알고리즘은 도시철도 물류 시스템의 효율적인 측면에서의 운용 목적에 기여될 수 있다.

본 논문은 도시철도 물류 시스템 물류 자동화의 핵심 구성요소로써, 수평이송장치의 경로 생성 알고리즘을 제시하였으나, 통합관제시스템과 수평이송장치 경로생성에 대한 통신 문제만을 다루고 있기에 경로생성 알고리즘 최적화에 대한 한계점을 가진다. 추후, 이를 보완하기 위한 향후 연구로써, 경로생성 알고리즘 자체의 성능을 평가하는 성능평가의 기준을 마련하여, 더욱 최적화된 경로생성 알고리즘 연구를 진행할 예정이다.

## 참고문헌

- Choi J. H. and Choi B. J. (2019), "Indoor Moving and Implementation of a Mobile Robot Using Hall Sensor and Dijkstra Algorithm", IEMEK Journal of Embedded Systems and Applications (IEMEK J. Embed. Sys. Appl.), vol. 14, no. 3, 151-156.
- Hwang S. W., Kim H. S. and Kim J. U. (2021), "Improvement Interface Requirements through Operation Scenario Analysis of Urban Railway Logistics System", Journal of Logistics Science & Technology, vol. 2, no. 2, 38-51.
- Jeon H. G. and Hong Y. S. (2012), "A Moving Control of an Automatic Guided Vehicle Based on the Recognition of Double Landmarks", The Journal of Korean Institute of Communications and Information Sciences", vol. 37, no. 8, 721-730.
- Jung C. H. and Lee M. S. (2017), "Development of Augmented Reality Walking Navigation App using Dijkstra Algorithm", Journal of The Korea Society of Computer and Information (JKSCI), vol. 22, no. 2, 89-95.
- Kim J. M., Park J. J., Jeon T. R. and Kim S. S. (2009), "Fuzzy and Proportional Controls for Driving Control of Forklift AGV", Journal of Korean Institute of Intelligent Systems, vol. 19, no. 5, 699-705.
- Lee J. W. and Lee B. R. (2019), "Design of Automatic Guided Vehicle Controller with Built-in Programmable Logic Controller", The Journal of Korea Institute of Convergence Signal Processing, vol. 20, no. 3, 118-124.
- Lee J. W. and Lee B. R. (2020), "Design of the Driving Controller of Automatic Guided Vehicle based on Fuzzy Inference System", Journal of Korean Institute of Information Technology, vol 18, no. 6, 11-16.
- Lee Y. H., Park E. K., Park T. J., Ryu K. R. and Kim K. H. (2003), "AGV System Operating Scheme based on Grid Level Control in Automated Terminal", Journal of Korean Navigation and Port Research, vol. 27, no. 2, 223-231.
- Tack H. H. and Kwon S. G. (2014), "Driving Control of Automated Guided Vehicle Using Centroid of Gravity Method", Journal of the Korea Society Industrial Information System, vol. 19, no. 2, 59-66.
- Woo S. B., Jung K. H., Kim J. M., Park J. J. and Kim S. S. (2010), "Velocity Control Method of AGV for Heavy Material Transport", Journal of Korean Institute of Intelligent Systems, vol. 20, no. 3, 394-399.
- Yang K. M., Han J. B., Sin H. S., Lee J. I., Lee S. J. (2019), "An Integrated Multi-robot Control System based on ROS for Autonomous Exploring Disaster Environment", Journal of Institute of Control, Robotics and Systems (J Inst Contr Robot Syst), vol. 25, no. 6, 534-540.
- Yang K. M., Suh J. H., Lee J. W., Noh J. H., Kim M. G., Seo K. H. (2021), "An Integrated Control System for Disaster Response Robot based on Multiple ROS Core considering Network Instability", Journal of the Korean Society for Precision Engineering (JKSPE), vol. 38, no. 10, 741-748.



## 물류과학기술연구 제3권 제2호

- 
- 인 쇄 : 2022년 12월 30일
  - 발 행 : 2022년 12월 31일
  - 발행인 : 하 현 구
  - 편집인 : 채 준 재
  - 발행처 : (사)한국물류과학기술학회(<https://www.klst.or.kr>)  
경기도 의왕시 철도박물관로 176  
전화 : 031-460-5921
- 

(비매품)

□ 이 학술지의 저작권은 (사)한국물류과학기술학회가 갖고 있습니다.

Copyright 2022. The Society of Logistics Science and Technology all rights reserved.