

# 딥러닝과 스테레오 카메라를 이용한 객체 거리 추출 방법

박민홍<sup>1</sup>, 원종운<sup>2</sup>, 조재훈<sup>3</sup>, 김용태<sup>1</sup>

<sup>1</sup>한경대학교 ICT로봇기계공학부

<sup>2</sup>한국철도기술연구원

<sup>3</sup>한경대학교 스마트물류기술연구센터

## Object distance extraction method using deep learning and stereo camera

Min-Hong Park<sup>1</sup>, Jong-Un Won<sup>2</sup>, Jae-hoon Cho<sup>3</sup>, Yong-Tae Kim<sup>1</sup>

<sup>1</sup>Dept. of ICT, Robot and Mechanical Engineering, Hankyong National University

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

<sup>3</sup>Smart Logistics Technology Institute, Hankyong National University, Korea

In this paper, we proposed a method to calculate the linear distance value according to the location of the delivery box and the stereo camera. The proposed method recognizes individual delivery boxes using YOLOv4, and calculates the vertical plane position of the camera and the linear distance between the boxes using the IMU sensor mounted on the stereo camera. In order to evaluate the performance of the proposed algorithm, a laboratory-scale experimental space was constructed, and the measured distance of the actual delivery box and the output distance of the proposed algorithm were compared.

**Keywords:** YOLOv4, IMU sensor, delivery boxes, stereo camera.

### 1. 서론

최근 코로나 19 사태는 사람과 화물의 이동패턴에 큰 영향을 미치고 있다. 사람과 화물의 이동에 있어서 검역과 보안기능이 강화되어 과거보다 이동의 자유가 제한되고 더 많은 시간과 비용 소요될 것으로 전망된다. 한편 유통의 온라인화는 더 빠른 속도로 진행되면서 택배와 같은 도시 물류수요가 크게 증가하고 있다(Lee, 2020).

온라인 쇼핑시장은 2017년 78조2000억 원으로 집계 되었으며, 5년마다 약 2배 이상 성장하고 있다. 2022년에는 176조2000억 원에서 189조8000억 원으로 늘어날 전망이다. 또한, 전자상거래 택배 물량도 2배 성장해 10억 박스가 넘을 것으로 예상된다(Soh et al., 2020). 이러한 택배 시장의 변화는 택배 회사들로 물동량 처리 효율을 높이기 위해서 인공지능, 자동화 시설의 필요성을 증가 시키고 있다. 인공지능 및 딥러닝을 이용한 창고관리는 물론 상품의 인식 및 분류 등 다양하게 적용되어지고 있다.

최근 택배 물류 기술 분야에서는 노동력 비중에 큰 상·하역 작업에 자동화 시스템을 도입하는 연구 개발들이 활발히 진행되어지고 있다. 특히 자동 하역장비에서 카메라와 딥러닝 기술을 이용한 박스와 화물을

인식하는 기술들은 택배 물류의 자동화를 위한 중요한 기술 중의 하나이다(Won et al., 2019).

딥러닝 기술은 합성곱 신경망을 이용한 다양한 구조의 딥러닝 기술들이 개발되어 지고 있다. 합성곱 신경망이 제안된 이후로 객체가 존재하는 영역을 먼저 구분하여 성능을 개선한 R-CNN이 개발되었으며 R-CNN의 연산시간을 개선시킨, Fast R-CNN(Girshick, 2015), Faster R-CNN(Ren, 2015) 등의 구조들이 개발되어 졌다. 또한 정확성을 높이기 위한 Masked R-CNN(He, 2017)의 개발은 산업현장에 적용할 만한 성능을 보였다. 이후 다양한 대기업들이 딥러닝 기술 연구에 참여하면서 산업현장에서의 딥러닝 기술이 적용이 빠르게 확산되어지고 있다. 특히 구글에서 개발하고 오픈소스로 제공되는 YOLO모델은 객체 인식의 속도와 성능은 다양한 산업분야에서 성공적인 적용이 이루어지고 있다(Choi et al., 2019).

한편 딥러닝 기술과 함께 카메라 기술 또한 빠르게 발전하고 있으며 최근에는 카메라를 이용한 객체의 거리값을 추출하는 저가형 스테레오 카메라 시장이 빠르게 발전하면서 딥러닝 기술과 카메라를 이용한 기술들이 다양하게 연구되어지고 있다. 스테레오 카메라는 객체와 카메라 사이의 거리를 알 수 있는 장점이 있다. 그러나 카메라의 높이와 기울기에 따라 거리값이 달라지기 때문에 적용분야에 따라 적절한 보상 알고리즘 개발이 필요하다. 자동 택배하역 장비의 경우 하역장비의 초기 위치에서 목표 박스 및 화물로 이동하여 성공적인 작업을 수행하기 위해서는 하역장비의 초기 좌표와 카메라의 설치 위치, 높이 그리고 기울기에 따른 객체의 직선거리를 계산할 필요가 있다.

본 논문에서는 택배박스과 카메라 위치(높이 및 기울기)에 따른 직선 거리값을 계산하는 방법을 제안한다. 제안한 방법은 딥러닝을 기술 중 YOLO-v4(Alexey, 2020)를 이용하여 개별 택배 박스들을 인식하고 인식된 박스들을 스테레오 카메라에 장착된 IMU(Inertial Measurement Units)센서를 이용하여 카메라의 수직 평면 위치와 박스들 간의 직선거리를 계산한다. 제안된 알고리즘의 성능을 평가하기 위하여 연구실 규모의 택배 화물 공간을 구축하여 실험을 진행하였으며 실제 택배 박스의 실측거리와 제안된 알고리즘의 출력 거리값을 비교하였다. 실험결과 제안된 알고리즘의 출력 거리값이 만족할만한 성능을 보임을 확인할 수 있었다.

## 2. 딥러닝 기반 객체 인식

이미지나 영상에서 객체인식을 기술들은 다양한 방법으로 연구되어져 왔다. 초기의 객체인식 알고리즘들은 히스토그램이나 푸리에 변환과 같은 방법들에 의해서 특징점들을 추출하고 각각의 특징점들과 가장 유사한 객체들을 판별하는 기술이었으며 신경망 및 SVM(Support Vector machine)등과 같은 분류기 등을 사용하였다. 2012년 ILSVRC(Large Scale Visual Recognition Challenge)에서 딥러닝에 기반한 AlexNet이 기존의 방법들에 비해 우수한 성능을 보임으로써 딥러닝에 의한 객체인식 기술이 큰 관심을 받기 시작하였다(Krizhevsky et al., 2012). 이후 2013의 ZF Net(Zeiler et al., 2014), 2014년에 GoogLeNet(Szegedy, et al., 2015), 2015년 ResNet(He et al., 2016) 등의 딥러닝 기반의 기술들이 ILSVRC에서 우승하면서 딥러닝 기술의 성능을 입증하였다.

최근 구글은 오픈 딥러닝 기술인 YOLO모델들을 연달아 발표하면서 지속적인 성능 개선이 이루어지고 있다. 본 논문에서는 YOLO-v4를 이용한 객체인식 기술을 기반으로 객체를 인식하고 제안된 알고리즘을 적용하였다.

### 2.1 객체 인식을 위한 데이터셋 구축

본 연구에서는 다양한 택배 화물 중 박스를 대상 객체로 선정하였으며 딥러닝 기술을 이용하여 박스 인식을 수행하기 위하여 5,102장의 학습 이미지 수집하여 이용하였다. 다양한 택배환경에서 박스들은 단일 박스로 존재하기 보다는 하나의 이미지에 다양한 위치와 형태로 적재되어진 경우가 많다. 또한 외부의 힘에 의해 박스의 일그러짐 등의 왜곡들에 대해서 정확한 위치와 인식을 위하여 이미지의 라벨링을 다각형 형태로 표시하였다. 사용된 데이터셋의 예시는 아래와 같다.



(a) example of box data



(b) example of box data

Figure 1. example of box data

객체에 대한 영역을 표시하기 위하여 라벨링 기법의 하나인 VIA(Vgg Image Annotator)를 이용하였다. 박스에 대한 데이터셋은 하나의 객체(박스)의 영역을 세부적으로 표현하기 위하여 다각형의 점들로 표현하였으며 하나의 이미지에 박스의 개수와 하나의 객체가 갖는 이미지상의 위치 및 하나의 객체를 표현하는 사각형의 경계박스(Bounding Box)로 구성하였다.



Number of object : 6

Bounding Box (x, y, width, height)

Each area of object

Figure 2. Information of dataset of box image

## 2.2 객체 인식을 위한 딥러닝 모델

객체 인식을 위한 모델은 YOLOv4 모델을 이용하여 학습을 진행하였다. YOLOv4의 전체적인 구조는 아래와 같은 구조를 보인다.

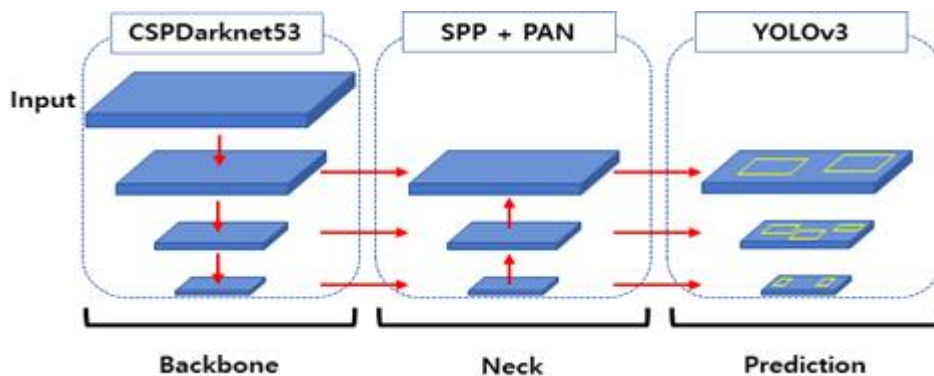


Figure 3. Structure of learning model

YOLO는 객체 인식의 대한 문제를 회귀 문제로 접근하여 객체가 있는 위치 및 크기에 대하여 추출하는 네트워크를 따로 갖지 않는다. 입력 크기에 따른 최종 출력은 격자 형태에 정보를 포함한 형태를 갖고 있는 것이 특징이다. 객체가 있는 위치를 사전에 학습하거나 먼저 연산하는 과정이 제외되어 RCNN 계열의 모델보다 연산 속도가 빠르다는 장점을 갖는다. 또한 YOLOv4에서는 CSPNet을 추가하여 기존의 ResNet과 같은 깊이가 100층 이상의 네트워크에서도 학습 시 전체 연산량을 줄였다. 연산량을 줄임으로 출력되는 속도가 향상되는데 이는 카메라 영상의 FPS 성능을 올렸다. YOLO는 다른 객체 인식 방법들의 모델들 보

다 빠른 속도를 가지며 최종출력을 통하여 각 격자의 정보를 통해 해당 객체의 위치나 크기를 알 수 있다. 따라서 객체의 정보를 직관적으로 알 수 있기에 객체의 실제 위치 추정에 적합하다.

### 3. 제안된 깊이 영상에 따른 객체의 위치 추정 방법

거리에 대한 정보를 나타내는 영상이라고 할 수 있는 깊이 영상은 3차원 공간상의 깊이 정보(Depth Map) 즉 3차원 공간상의 피사체의 거리 값을 갖는 정보이다. 이러한 3차원 깊이 정보는 스테레오 카메라를 이용하거나 레이저 스캐너, TOF(Time of Flight)를 이용하는 방법들이 있다. 본 논문에서는 객체를 인식하고 인식된 객체의 3차원 공간상의 거리와 높이를 추정하기 위하여 스테레오 카메라를 사용하였다.

스테레오 카메라는 사람의 눈과 유사한 방식으로 카메라에 입력된 영상을 통해 관측된 공간상의 거리 및 관측 물체의 형태를 감지 할 수 있다. 스테레오 카메라는 영상에서의 임의의 패턴을 다른 카메라 영상에서 찾아낸다. 이러한 패턴의 관계를 바탕으로 삼각기법을 이용하여 3차원 공간상의 위치를 도출 할 수 있다. 원리는 아래 그림과 같이 임의의 base line의 거리를 두고 위치한 카메라의 영상을 통해 피사체의 변위를 구하고 렌즈의 초점 거리를 통하여 스테레오 카메라로부터 피사체의 거리를 다음 식(1)과 같이 추출한다.

$$1) \text{ distance} = \text{Focal} \frac{b}{\text{diparity}} \quad (1)$$

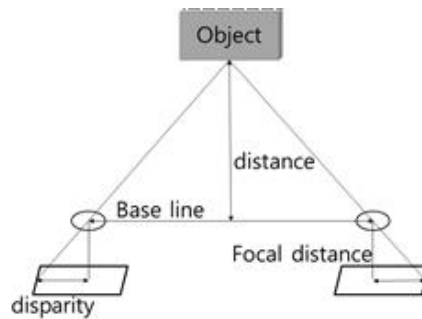


Figure 4. Geometry of the stereo matching system

Distance는 피사체와 카메라의 거리, Base line은 두 카메라 렌즈의 거리를 뜻하며, disparity는 두 개의 이미지에서 피사체 간 거리 차, Focal distance는 렌즈의 초점 길이이다. 이를 통해 카메라 영상을 통해 피사체를 아래 그림과 같이 거리 정보로 표현할 수 있는 것이다.

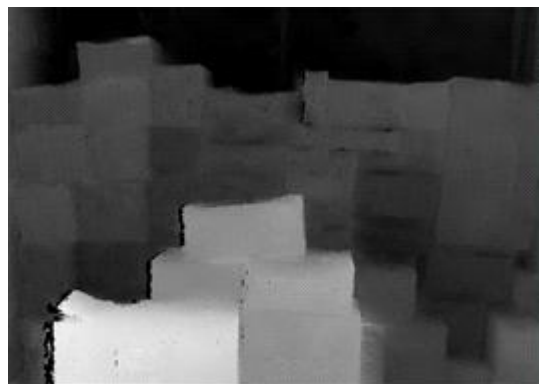


Figure 5. Example of depth map

#### 3.1 깊이 영상 추출 및 위치 계산

본 논문에서는 인식된 박스의 바닥면에서 높이와 카메라가 위치한 평면에서의 직선거리를 추출한다. 스

스테레오 카메라는 지면 또는 이동하는 면에서 높이가 존재하고 카메라를 통해 나타나는 영상도 정확한 수평을 이룰 수 없다. 이러한 요소로 인하여 스테레오 카메라는 영상에 입력된 피사체의 직선거리를 계산하고 도출하게 된다. 도출된 직선거리는 카메라의 위치에서부터 떨어진 거리를 나타내므로 인식된 피사체의 바닥 평면에서 높이, 카메라가 위치한 평면에서 피사체까지의 직선거리는 표현될 수 없다. 본 논문에서는 3차원 공간에서 카메라의 각도와 무관하게 피사체의 수직, 수평 거리를 계산하는 방법에 대해 제안한다. 피사체의 수직, 수평거리를 계산하게 된다면 인식된 객체를 적재하거나 하역 시 기준을 제시할 수 있다. 제안된 방법은 다음 아래 그림과 같이 카메라의 위치를 직교좌표의 원점으로 가정한다.

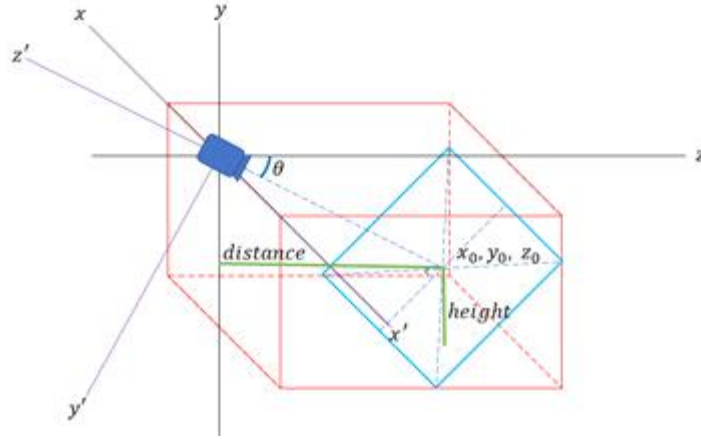


Figure 6. Distance and height extraction using depth information

수식 (2)과 (3) 같이 카메라의 영상의 중심점을 기준으로 카메라의 roll( $\phi$ ), yaw( $\psi$ ) 회전각을 계산하여 카메라 영상의 중심점을 회전 변환을 한다. 회전 변환된 중심점과 pitch( $\theta$ ) 회전각을 이용하여 카메라가 위치한 평면의 법선 벡터를 구한다.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3)$$

도출된 법선 벡터를 통하여 기준좌표계의 평면의 방정식을 다음 식(4)과 같이 나타낼 수 있다. 평면의 방정식을 통하여 스테레오 카메라를 통해 얻은 공간상의 한 점을 바탕으로 평면의 수직인 직선거리를 다음 식(5)과 같이 나타낼 수 있다.

$$t(x-x_0)+p(y-y_0)+q(z-z_0)=0 \quad (4)$$

$$distance = \frac{|tx_0+py_0+qz_0+d|}{\sqrt{t^2+p^2+q^2}} \quad (5)$$

위 과정과 같이 카메라가 위치한 기준 좌표의 z축으로부터 직선거리를 도출하여 지면에서의 높이 또한 측정가능 하며 높이는 바닥면에서 카메라의 높이의 기준이 있어야한다. 본 논문에서는 이러한 과정을 통해 인식된 객체의 중심점을 기준으로 해당 객체의 위치가 직교좌표계에서 어떤 위치를 갖는지 추정할 수 있다.

### 3.2 인식된 객체의 작업 영역

본 논문에서는 박스를 딥러닝을 통해 인식하여 해당 박스의 위치를 스테레오 카메라 영상의 정보를 통

해 좌표를 계산한다. 이러한 좌표들은 많은 박스가 쌓여있는 물류창고나 택배하역장과 같은 곳에서는 박스가 적층된 위치에 따라 순서대로 하역이 요구되는 자동 하역장비에 적용될 수 있으며 적층된 깊이에 따라 박스를 하역하는데 있어서 작업하고자 하는 위치의 우선순위가 변동되는 시스템에도 적용되어질 수 있다. 또한 제안된 알고리즘의 특징은 딥러닝을 통해 인식된 객체의 거리 및 높이 정보를 이용하여 하역하는 기준에 따라 작업 영역을 구분할 수 있다. 하역하는 기준이라 함은 작업 시 박스를 하역할 수 있는 면적이다. 기준에 따라 박스를 작업하는 공간 영역이 달라지므로 하역 면적에 따라 구분할 수 있는 알고리즘은 다음 흐름도와 같다.



Figure 7. Flowchart of workspace division

연산 순서는 첫째로 딥러닝을 통해 객체인식을 하여 인식된 객체의 이미지상 좌표정보를 통해 카메라에서 직선거리를 추출한다. 추출된 직선거리와 카메라에서 보는 객체의 좌표값을 통하여 객체 중심점의 3.1절에서 언급한 방법으로 객체의 높이와 거리를 추출한다. 다음으로 추출된 거리와 높이 및 이미지상 좌표를 통해 각각의 객체가 갖는 정보를 정렬한다. 객체의 각 거리 즉 카메라가 위치한 평면에서의 직선거리에 따라 내림차순 정렬한다. 정렬된 객체의 거리값을 통해 해당 객체간의 차이를 통해 적재된 위치가 다른 곳을 추출한다. 마지막으로 적재 깊이가 유사한 부분은 산출하여 유사한 부분의 객체들의 높이를 통해 작업할 수 있는 공간 안에 들어오는 높이를 산출하여 최종적으로 작업영역을 산출한다.

## 4. 실험 결과

제안된 방법의 성능을 평가하기 위하여 연구실 규모의 테스트 베드를 구축하였으며 딥러닝 모델의 정확도 평가 및 실험, 인식된 객체의 중심점의 연산된 높이와 거리의 실제 거리와의 오차를 실험 및 계산된 작업영역과 실제 작업영역을 비교하여 진행하였다. 실험에 사용된 박스는 실제 우체국 택배 박스 및 다양한 업체의 박스를 이용했으며 실험실 또는 컨테이너 내부에서 촬영된 이미지가 사용되었으며, 스테레오 카메라 모델은 ZED 스테레오 카메라를 사용하였다.

### 4.1 객체 인식 학습

객체인식 및 분류를 회귀문제로 접근하는 YOLO는 합성곱 형태의 출력을 나타낸다. 이러한 형태는 평균 30FPS(Frame Per Second) 이상의 카메라 영상을 표현할 수 있는 장점을 갖고 있다. 본 논문에는 5,102장의 박스 사진에 대하여 학습을 진행하였다. 학습을 위하여 이미지 데이터셋은 직접 촬영과 구글 이미지를 통해 구축하였다. 데이터셋은 단일 박스 또는 하나의 이미지에 복수의 박스가 존재하는 이미지들을 사용하였다.



Figure 8. Box image used in the experiment

학습 데이터는 전체 데이터셋의 90%(4,592장)를 사용하였으며 평가 데이터셋은 10%(510장)를 사용하였다. 또한 전체 epoch 수는 2,000, 메모리 크기를 고려하여 batch size는 2로 설정하였다. 활성화함수는 Mish를 이용하여 기존의 활성화함수보다 가중치를 갱신하고 Mosaic augmentation 기법을 통해 성능을 개선하였다. 그림 9는 제안된 방법의 박스 인식 정확도를 나타낸다. 실험결과 박스에 대한 단일 객체의 정확도는 약 97%의 성능을 보여주고 있다.

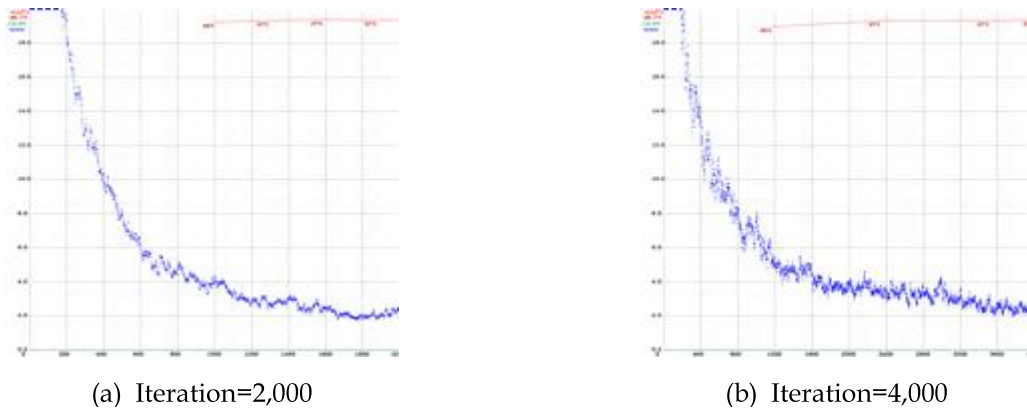


Figure 9. Learning result of box dataset

#### 4.2 스테레오 카메라를 통한 인식된 객체 거리 비교 실험

그림 10과 11은 실제 실험에서 박스를 인식을 결과 영상이며 표1~4는 앞서 학습된 결과를 바탕으로 입력된 영상에서 해당 객체의 중심점을 계산하여 추출된 거리와 높이와 실제 거리와 높이를 비교 하였다. 또한 결과값의 신뢰성을 위해서 6번의 다른 환경에서 각각 8회씩 48회의 실험을 진행하였으며 오차율은 mm 단위까지 인식하도록 하였다. 실험결과 거리에 대한 평균오차는  $\pm 13.6\text{mm}$ , 높이에 대한 평균 오차는  $\pm 12.6\text{mm}$ 로 만족할만한 성능을 보였다.



(a) Pitch angle = 10



(b) Pitch angle = 15

Figure 10. Result of object detection at deferent angle

Table 1. Result of figure 10.(a)

지점	예측 거리	실측 거리	거리 오차	예측 높이	실측 높이	높이 오차
1	2760	2780	-20	1119	1140	-21
2	2770	2755	+15	1709	1698	+11
3	2776	2760	+16	228	245	-17
4	2719	2705	+14	1130	1148	-18
5	2754	2735	+19	685	695	-10
6	2756	2740	+16	216	225	-09
7	2750	2735	+15	773	795	-22
8	2876	2890	-14	1793	1782	+11

Table 2. Result of figure 10.(b)

지점	예측 거리	실측 거리	거리 오차	예측 높이	실측 높이	높이 오차
1	2761	2780	-19	1123	1140	-17
2	2770	2755	+15	1707	1698	+09
3	2775	2760	+15	225	245	-20
4	2715	2705	+10	1131	1148	-17
5	2750	2735	+15	686	695	-09
6	2751	2740	+11	215	225	-10
7	2748	2735	+13	780	795	-15
8	2878	2890	-12	1792	1782	+10



(a) Result of move right to 1m



(b) Result of move left to 1m

Figure 11. Result of object detection at deferent position

Table 3. Result of figure 11.(a)

지점	예측 거리	실측 거리	거리 오차	예측 높이	실측 높이	높이 오차
1	2685	2701	-14	226	240	-14
2	2676	2658	+18	206	218	-12
3	2712	2691	+11	330	311	+19
4	2691	2681	+10	1004	1010	-06
5	2676	2670	+06	883	891	-08
6	2693	2705	-12	1175	1163	+12
7	2701	2708	-09	1431	1442	-11
8	2707	2685	+22	1640	1622	+18

Table 4. Result of figure 11.(b)

지점	예측 거리	실측 거리	오차 거리	예측 높이	실측 높이	오차 거리
1	2688	2701	-13	229	240	-11
2	2673	2658	+15	210	218	-08
3	2683	2691	-08	327	311	+16
4	2665	2681	-16	1001	1010	-09
5	2682	2670	+12	888	891	-03
6	2690	2705	-25	1172	1163	+09
7	2700	2708	-08	1429	1442	-13
8	2704	2685	+19	1638	1622	+16



Figure 12. Result of object detection

Table 5. Result of figure 12.

지점	예측 거리	실측 거리	거리 오차	예측 높이	실측 높이	높이 오차
1	2684	2665	+19	602	615	-13
2	2694	2685	+09	230	245	-15
3	2688	2693	-05	718	735	-17
4	2690	2679	+11	328	340	-12
5	2691	2688	+03	394	400	-06
6	2685	2677	+08	189	198	-09
7	2691	2680	+11	187	197	-10
8	2685	2691	-06	828	833	-05

Table 6. Result of figure 13.

지점	예측 거리	실측 거리	거리 오차	예측 높이	실측 높이	높이 오차
1	2781	2765	16	1418	1415	+03
2	2784	2785	1	1395	1397	-02
3	2798	2777	22	555	576	-21
4	2799	2780	19	739	753	-14
5	2792	2775	17	306	323	+17
6	2795	2775	20	1050	1071	-21
7	2798	2785	13	1062	1085	-23
8	2793	2780	13	1613	1608	+05

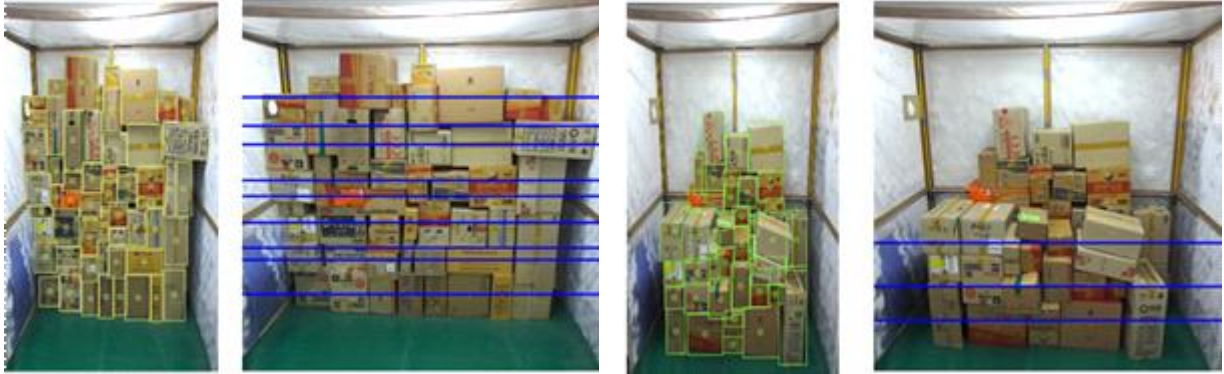
#### 4.3 작업영역 비교 실험

본 논문에서는 인식된 객체를 바탕으로 400mm\*2,000mm의 로봇 그리퍼의 작업영역을 기준으로 해당 객체들을 하역하는데 필요한 작업 횟수와 중복 영역의 차이를 기준으로 실험을 진행하였다. 실제 박스의 전체 영역을 계산하여 필요한 작업 횟수와 중복 영역을 산출하였다. 딥러닝을 이용한 객체인식을 통하여 도출된 작업영역과 비교를 진행하였다. 아래 그림은 실제 로봇을 작업하기 위한 로봇의 그리퍼 이다.



Figure 14. Real robot gripper shape

컨테이너에서 작업하기 위하여 해당 그리퍼와 동일한 공간을 제작하였으며 제작된 공간과 컨테이너 내부에서의 작업영역을 다음 그림과 같다.



(a) One section

(b) Two sections

Figure 15. Separating the workspace in different situations

## 5. 결론

본 논문에서는 택배박스과 카메라 위치에 따른 직선 거리값을 계산하는 방법을 제안하였다. 제안한 방법은 YOLOv4를 이용하여 개별 택배 박스들을 인식하고 인식된 박스들을 스테레오 카메라에 장착된 IMU 센서를 이용하여 카메라의 수직 평면 위치와 박스들 간의 직선거리를 계산한다. 제안된 알고리즘의 성능을 평가하기 위하여 연구실 규모의 택배 화물 공간을 구축하여 실험을 진행하였으며 실제 택배 박스의 실측 거리와 제안된 알고리즘의 출력 거리값을 비교하였다. 또한 결과값의 신뢰성을 위해서 6번의 다른 환경에서 각각 8회씩 48회의 실험을 진행하였으며 오차율은 mm단위까지 인식하도록 하였다. 실험결과 거리에 대한 평균오차는  $\pm 13.6\text{mm}$ , 높이에 대한 평균 오차는  $\pm 12.6\text{mm}$ 로 만족할만한 성능을 보였다.

## 참고문헌

- Lee, W-H (2020), Strategy for Overcoming Environmental Change Risks Facing the Distribution and Logistics Industry Caused by COVID-19 Pandemic, *Journal&Article Management System*, 23(3), 81-95.
- Soh, S-B, Kim, S-B & Lee, J-P (2020), A Study on the Impact of Product Category on the Expectation of Delivery Service Quality in Online Shopping Malls, *Korean society of supply chain management*, 20(1), 52-66.
- Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. (2019), Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving. *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 502-511.
- Cheng-Yang Fu, Mykhailo Shvets, and Alexander C Berg. (2019), RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv preprint arXiv:1901.03353*
- Girshick, R. (2015), Fast r-cnn, *In Proceedings of the IEEE international conference on computer vision*, 1440-1448.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015), Faster r-cnn, *Towards real-time object detection with region proposal networks. In Advances in neural information processing systems*, 91-99.
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017), *Mask r-cnn. In Proceedings of the IEEE international conference on computer vision*, 2961-2969.
- Won, J-U., Park, M-H., Park, S-W., Cho, J-H., and Kim, Y-T. (2019), Deep Learning Based Cargo Recognition Algorithm for Automatic Cargo Unloading System, *International Journal of Fuzzy Logic and Intelligent Systems*, 29(6), 430-436.
- Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan and Mark Liao. (2020), YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934v1 [cs.CV] 23 Apr*
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, 1097-1105.
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *In European conference on computer vision*, 818-833.
- 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*, 1-9.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.