

# 소형 UAV의 장애물 충돌 회피를 위한 YOLO 및 IR 센서 기반 장애물 크기 예측 방법

이의천<sup>1,2</sup> · 이종원<sup>1</sup> · 최의진<sup>1</sup> · 이선아<sup>1,2,†</sup>

<sup>1</sup>경상국립대학교 공과대학 항공우주 및 소프트웨어공학부

<sup>2</sup>경상국립대학교 공과대학 AI융합공학과

## The Obstacle Size Prediction Method Based on YOLO and IR Sensor for Avoiding Obstacle Collision of Small UAVs

Uicheon Lee<sup>1,2</sup> Jongwon Lee<sup>1</sup>, Euijin Choi<sup>1</sup> and Seonah Lee<sup>1,2,†</sup>

<sup>1</sup>Department of Aerospace and Software Eng., Gyeongsang National University

<sup>2</sup>Department of AI Convergence Eng., Gyeongsang National University

### Abstract

With the growing demand for unmanned aerial vehicles (UAVs), various collision avoidance methods have been proposed, mainly using LiDAR and stereo cameras. However, it is difficult to apply these sensors to small UAVs due to heavy weight or lack of space. The recently proposed methods use a combination of object recognition models and distance sensors, but they lack information on the obstacle size. This disadvantage makes distance determination and obstacle coordination complicated in an early-stage collision avoidance. We propose a method for estimating obstacle sizes using a monocular camera-YOLO and infrared sensor. Our experimental results confirmed that the accuracy was 86.39% within the distance of 40 cm. In addition, the proposed method was applied to a small UAV to confirm whether it was possible to avoid obstacle collisions.

### 초 록

UAV의 수요가 증가함에 따라 많은 충돌 회피 방법들이 제안됐다. 이러한 방법들은 LiDAR 및 스테레오 카메라를 주축으로 연구되었으나 무게거나 공간이 부족하여 소형 UAV에 접목이 어려웠기에, 최근에는 객체 인지 모델 및 거리 측정 센서를 복합적으로 사용한 방법들이 제안되고 있다. 하지만 이러한 객체 인지 복합 방법들은 인지한 장애물의 크기 정보를 도출하지 않아 인지 초기에 적정 회피 거리 도출 및 장애물의 좌표화가 어렵다는 단점이 존재한다. 본 논문에서는 단안 카메라-YOLO와 적외선 센서 기반의 장애물 크기 예측 방법을 제안하고, 실험을 통해 40cm의 거리 내에서 86.39%의 정확도를 보임을 확인했다. 또한, 제안한 방법을 적용하여 소형 UAV에 적용하여 장애물 충돌 회피가 가능한지를 확인하였다.

**Key Words** : Small UAV(소형 무인 항공기), Collision Avoidance(충돌 회피), Obstacle Detection(장애물 감지), Size Prediction(크기 예측)

## 1. 서 론

무인 항공기(Unmanned Aerial Vehicle, UAV)는 사람이 탑승하지 않고 원격 조종 등의 방법으로 비행하는 항공기로, 운용 및 비용의 측면에서 소형화의 장

점이 있기에 수요가 증가하는 추세이다. 무인 항공기의 비행 안전성 확보를 위해서는 충돌 회피(Collision Avoidance)가 필요한데, 충돌 회피의 첫 번째 단계는 장애물을 인지하는 것이다.

기존 장애물 인지 연구들은 타 센서들에 비해 높은 정확도를 보이는 LiDAR(Light Detection And Ranging) 및 스테레오 카메라(stereo camera)를 주축으로 진행되었다[1-6]. 하지만 이 센서들은 1 kg 이

하의 소형 UAV에는 적용이 어려우므로, 소형 UAV에는 일반적으로 단안 카메라와 객체 인지 딥러닝 모델인 YOLO(You Only Look Once)을 사용한 비전(vision) 기반 장애물 인지 방법들이 제안되고 있다 [7-11]. 그러나 이러한 방법들은 인지한 장애물의 크기 정보를 도출하지 못하고, 장애물로 다가가는 동안의 이미지 정보 및 거리 정보의 변화를 바탕으로 충돌을 예측한다는 단점이 있다.

본 논문에서는 충돌 회피에 필요한 장애물 크기 정보 예측을 위하여 단안 카메라-YOLO 및 적외선(InfraRed, IR) 센서 기반의 방법을 제안한다. 본 방법에 대한 기본 가정은 어떤 대상이 카메라로부터 멀어질수록 카메라 이미지 내 대상의 크기도 멀어진 거리와 비례하여 작아진다는 것이다. 이러한 비례를 바탕으로, IR 센서를 통해 장애물까지의 거리를 측정하고 YOLO를 통해 영상 내 장애물의 크기를 측정한다면 장애물의 실제 크기를 예측할 수 있다. 이때 사용되는 YOLO는 단안 카메라만을 요구하며, IR 센서는 타 센서에 비해 작고 가벼우므로 소형 UAV에 적합하다.

본 논문에서 제안하는 장애물 크기 예측 방법의 정확도를 평가하기 위해 총 3개의 연구 질문을 바탕으로 실험했으며, 질문 및 결과는 다음과 같다.

첫째, IR 센서의 거리 측정이 정확한지 확인하였다. 이는 크기 예측을 위한 전제사항이었으며, 이를 확인하기 위해, 장애물의 종류, 배치된 수평거리 및 수직거리에 변화를 주며 측정했으며, 40cm 범위 내에서 평균 91.54%의 정확도를 보임을 확인했다.

둘째, YOLO의 영상 내 장애물 크기 측정이 정확한지 확인하였다. 이도 크기 예측을 위한 전제사항으로, 이를 확인하기 위해, 장애물의 종류, 배치된 수평거리 및 수직거리에 변화를 주며 측정했으며, 40cm 범위 내에서 평균 95.04%의 정확도를 보임을 확인했다.

셋째, 제시된 방법의 장애물 크기 예측은 정확한지 확인하였다. 이는 본 연구의 중심 제안 사항을 평가하기 위한 것으로, 이를 확인하기 위해, 앞의 실험 결과들을 사용하여 계산했으며, 40cm 범위 내에서 평균 86.39%의 정확도를 보임을 확인했다.

본 논문의 방법은 객체 인지 기법 및 거리 측정 센서를 바탕으로 장애물의 크기를 예측하는 것을 최초로 제안한다. 또한 본 방법은 소형 UAV에 접목하기 쉬운

센서들을 사용하므로 소형 UAV의 장애물 인지에 공헌할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 제안 방법에 대해 설명한다. 3절에서 실험 계획을, 이어 4절에서는 실험 결과를 각각 제시한다. 5절에서는 제안한 방법에 관한 사례연구를 보이며, 6절에서는 논의를 진행한다. 그 뒤 7절에서는 논문을 결론짓는다.

## 2. 제안 방법

### 2.1 개요

본 논문에서 제안하는 장애물 크기 예측 방법은 단안 카메라, YOLO, IR 센서를 통해 이뤄지며, 아래 Fig. 1과 같이, 크게 준비 단계(Preparation Phase) 및 실행 단계(Execution Phase)로 구성된다.

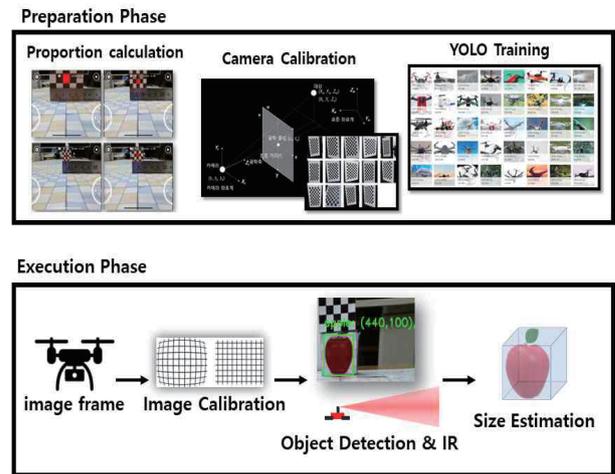


Fig. 1 Size Estimation Method of Obstacle

준비 단계에서는 장애물까지의 거리 및 장애물의 크기에 따른 비례를 계산하고, 카메라 캘리브레이션을 수행하여 사용될 카메라의 파라미터를 구하며, 또한 필요하다면 인지를 원하는 장애물에 대해 YOLO에 학습시키는 과정이 포함된다.

실행 단계에서는 준비 단계에서 구한 카메라 파라미터를 바탕으로 카메라로 입력되는 이미지를 보정하고 YOLO를 통해 이미지 내 장애물을 인지한다. 그다음, IR 센서를 통해 얻은 장애물까지의 거리와 이미지 내 인지된 장애물 크기 사이의 비례관계를 통해 장애물의

실제 크기를 예측한다.

## 2.2 준비 단계

### (1) 거리 및 크기에 따른 비례

장애물의 크기를 예측하기 위해선, 장애물까지의 거리 및 현재 위치에서 보이는 장애물의 크기가 주어졌을 때 이 정보와 장애물의 크기 사이의 비례를 알아야 한다. 고정된 특정 물체에 대해 카메라에서 해당 물체가 점차 멀어질수록 이미지 내 물체의 크기는 줄어들기므로, 물체와 카메라 사이의 거리는 물체의 이미지 내 크기와 반비례 관계에 있다. 이에, 카메라와 물체의 간격을 점차 벌리며 촬영하여, 이미지 내 물체의 픽셀(pixel)이 물체와 카메라와의 거리에 따라 얼마나 달라지는지를 측정하여 구체적인 비례를 구한다. 아래의 Fig. 2는 폭이 1 cm인 정사각형 형상에 대해 10 cm 단위로 멀어지며 거리를 예측한 것으로, 측정을 수행한 카메라의 렌즈는 960의 비례계수를 지니는 것을 확인할 수 있다.

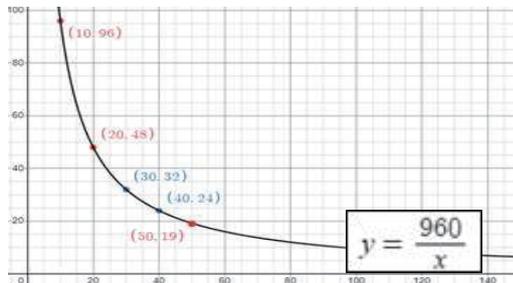


Fig. 2 Graph with Distance x and Width y

### (2) 카메라 캘리브레이션

카메라 캘리브레이션(Camera Calibration)은 카메라로 촬영한 이미지와 실제 이미지를 비교하여 이미지에 영향을 주는 카메라의 내부 및 외부 파라미터들을 파악하는 과정이다. 아래의 Fig. 3과 같이, 카메라 렌즈로 입력된 이미지는 실제 물체의 3차원 공간상 점들을 2차원 평면에 투사하여 얻은 것이므로, 카메라 렌즈의 구조적 특성으로 인해 실제 물체보다 휘어져 보일 수 있다. 따라서 촬영된 이미지들은 실제 비율로 보정하는 과정이 필요하며, 이때 카메라 캘리브레이션을 통해 구한 파라미터를 바탕으로 이미지를 보정한다. 일반적으로 카메라 캘리브레이션을 위해서는 다각도에서

촬영한 10~20개의 이미지가 요구된다.

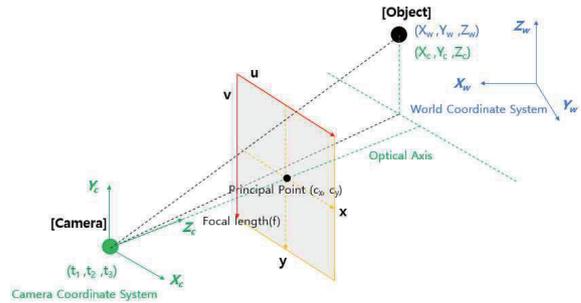


Fig. 3 The Process of Projecting 3D into 2D

### (3) YOLO의 학습

YOLO는 객체 인지를 위한 딥러닝 모델로, 주어진 이미지 내에 존재하는 장애물들의 크기 및 위치를 도출할 수 있기에 본 방법에서 필요하다. 하지만 학습된 장애물들만 인지할 수 있기에 사전에 장애물들에 대한 정보를 학습시켜두어야 한다. YOLO에게 장애물들을 학습시키기 위해서는 장애물이 포함된 이미지, 해당 장애물을 둘러싼 경계 상자(bounding box)의 좌표, 각 좌표에 해당하는 장애물의 이름 등의 3가지 정보가 포함된 학습 데이터 세트(Dataset)가 요구된다.

## 2.3 실행 단계

### (1) 이미지 보정

카메라로 촬영된 이미지는 카메라 렌즈의 굴절로 인해 휘어져 있으므로, 이미지 내에서 장애물의 크기를 예측하기 전에 보정을 해야 한다. 이미지의 보정은 준비 단계의 카메라 캘리브레이션에서 얻은 파라미터를 통해 수행된다. 실험에 사용된 카메라로 촬영한 이미지는 하단의 Fig. 4와 같이, 아치형으로 퍼지는 것을 확인할 수 있다. 이때, 사진이 변형되면서 발생한 공백은 검은색으로 이미지에 표현된다.

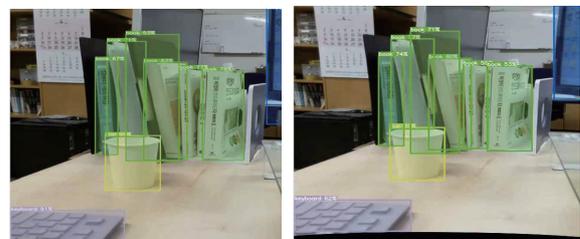
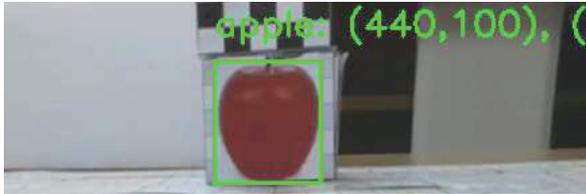


Fig. 4 (Left) Before (Right) After Calibration

**(2) 장애물 인지**

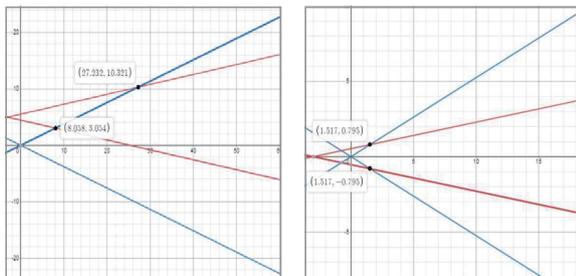
보정된 이미지는 YOLO에 의해 이미지 내 장애물들의 크기 및 위치가 도출된다. YOLO는 입력된 이미지에 있는 장애물 중 사전에 학습된 대상과 일정 이상의 일치도를 보이면 해당 장애물에 대한 정보를 반환한다. 하단의 Fig. 5와 같이, 인지된 장애물을 둘러싼 경계 상자 및 해당 경계 상자의 왼쪽 위와 오른쪽 아래의 이미지 내 좌표가 YOLO에 의해 도출된다. 경계 상자는 인지된 이미지와 접하고 있으므로, 경계 상자의 크기를 이미지 내 장애물의 크기로 간주할 수 있다.



**Fig. 5** Bounding Box and Coordinates

**(3) IR 범위 확인**

IR 센서는 장애물까지의 거리를 측정하므로, IR 센서의 측정 범위 내에 장애물이 있는 경우에만 크기를 예측할 수 있다. 하지만 단안 카메라의 시야각(Field of View, FoV)보다 IR 센서의 탐지각(이하 IR 범위)은 일반적으로 더 작기에, 카메라에는 보이더라도 IR 센서는 거리를 측정하지 못할 수 있다. 하단의 Fig. 6은 푸른 선으로 FoV, 붉은 선으로 IR 범위를 표현한 것으로, 좌측 그림은 측면, 우측 그림은 위에서 본 탐지 범위를 나타낸다. 서로 중첩되지 않는 영역이 존재함을 Fig. 6에서 확인할 수 있다. 따라서 이미지에서 인지된 장애물이 중첩 영역에 있는 경우에만 크기를 예측해야 한다.



**Fig. 6** FoV & Detection Angle

중첩 영역이 이미지 내에서 어떤 위치에 있는지 도출하는 방법으로는, FoV 범위 상단부를 기준으로 IR 범위의 상단부 및 하단부까지의 길이를 구하는 방법이 있다. IR 범위의 상단부 및 하단부의 길이를 안다면, 이미지의 한쪽 끝 면에서 각각의 길이를 추가하여 이미지 내 IR 범위, 즉 중첩 영역의 위치를 도출할 수 있다. 이때, 장애물까지의 거리에 따라 FoV 및 IR 범위는 달라지므로 상단부 및 하단부까지의 길이를 FoV 범위 길이 전체로 나눠 각각에 대한 비율을 도출 후 이미지의 폭을 곱하여 중첩 영역의 위치를 구할 수 있다. 각 범위를 직선화하여 이를 적용한 수식은 Eq. 1에서 확인할 수 있으며,  $P_{start}$  및  $P_{end}$ 는 각각 상단부 및 하단부까지의 길이비를 의미한다.

$$P_{start} = \frac{\tan(d_{fov}/2)x - \{\tan(d_{ir}/2)(x + C_{horizontal}) + C_{vertical}\}}{\tan(d_{fov}/2)x - \{-\tan(d_{fov}/2)x\}}$$

$$P_{end} = \frac{\tan(d_{fov}/2)x - \{-\tan(d_{ir}/2)(x + C_{horizontal}) + C_{vertical}\}}{\tan(d_{fov}/2)x - \{-\tan(d_{fov}/2)x\}}$$

- $P_{start}$ : Length ratio to lower IR based on upper FoV
- $P_{end}$ : Length ratio to upper IR based on upper FoV
- $x$ : Distance to the obstacle[cm]
- $d_{fov}$ : Angle of view of the camera[deg]
- $d_{ir}$ : Angle of view of the IR sensor[deg]
- $C_{horizontal}$ : Horizontal distance between monocular camera and IR sensor[cm]
- $C_{vertical}$ : Vertical distance between monocular camera and IR sensor[cm]

**Eq. 1** Calculation of Overlapping Detection Area of Monocular Camera and IR Sensor

이를 바탕으로, 이미지 내 장애물의 크기를 예측할 수 있는 영역의 경계에 대한 좌표를 도출할 수 있으며, 해당 영역 내 장애물의 좌표가 존재하는 경우에만 이후의 단계로 진행된다.

**(4) 복수의 장애물 처리**

YOLO는 이미지에서 다수의 장애물 정보를 출력할 수 있지만, IR 센서는 주변 장애물들에서 반사된 적외선의 양을 바탕으로 거리를 측정하기에 단 하나의 값만을 출력한다. 따라서 만약 FoV 및 IR 범위의 중첩

영역에 장애물이 다수 존재할 경우, 이들의 경계 상자를 하나로 합쳐 하나의 장애물로 간주하는 과정이 필요하다. 하단의 Fig. 7과 같이, 만약 인지된 장애물이 3개(주황, 노랑, 초록 경계 상자) 존재하고, IR 센서의 범위는 붉은 선의 경계 상자인 경우, 중첩 영역 밖에 있는 노랑 경계 상자는 배제되며 주황 및 초록 경계 상자를 병합하여 하나의 장애물로 간주한다.

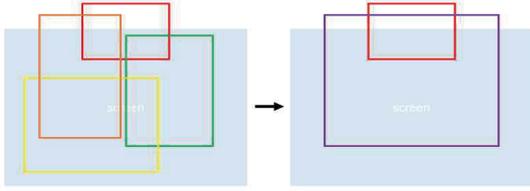


Fig. 7 Merge Bounding Boxes into One

#### (5) 크기 예측

준비 단계에서 계산된 거리 및 크기에 따른 비례( $P$ )와, 실행 단계에서 얻은 영상 내 장애물의 크기( $L_{img}$ )를 하단의 Eq. 2에 대입하여 물체의 가로 길이 및 세로 길이를 예측할 수 있다. 이때, 카메라와 IR 센서의 위치는 서로 다를 것이므로, 이를 바로잡기 위해 수평 거리에 대한 보정값( $C_{horizontal}$ )을 추가해준다. 이는 장애물에 대한 거리를 바로잡는 것이기 때문에 단안 카메라와 IR 센서 사이의 높이는 보정할 필요가 없다. 장애물의 실제 크기 예측은 IR 센서와 YOLO에서 얻은 값을 동시에 사용하므로, 실제 크기 예측에 대한 정확성 또한 IR 센서의 정확성 및 YOLO의 정확성에 종속된다.

$$L_{real} = \frac{L_{img}}{P} (Dist + C_{horizontal})$$

- $Dist$ : Distance from IR sensor to target[cm]
- $L_{real}$ : Actual length of the target[cm]
- $L_{img}$ : Length of the target in the image frame[px]
- $P$ : Proportional coefficient for size - distance
- $C_{horizontal}$ : Horizontal distance between monocular camera and IR sensor[cm]

Eq. 2 Equation for Actual Length Estimation

## 3. 실험 계획

### 3.1 연구질문

본 논문에서 제안하는 방법은 YOLO 및 IR 센서의 출력을 결합하여 크기를 예측하므로, 크기 예측의 정확도는 YOLO 및 IR 센서 각각의 정확도에 종속된다. 그러므로 장애물의 종류 및 측정 거리가 달라질 때 YOLO 및 IR 센서가 어느 정도의 정확도를 보이는지, 또한 종합적으로 본 방법은 어느 정도의 정확도를 보이는지 확인할 필요가 있다. 따라서 본 논문은 이에 따른 3개의 연구 질문을 제시한다.

- (1) RQ1. IR 센서의 거리 측정은 정확한가?
- (2) RQ2. YOLO의 영상 내 장애물 크기 측정은 정확한가?
- (3) RQ3. 제시된 방법의 장애물 크기 예측은 정확한가?

### 3.2 실험 환경

실험을 위해 사용할 단안 카메라 및 IR 센서는 DJI사의 소형 드론인 RMTT(Robomaster Tello Talent)에 내장된 센서들을 사용한다. RMTT의 규격은 다음의 Table 1에서 확인할 수 있다.

Table 1 Specification of RMTT

| Category              | Content  |
|-----------------------|----------|
| Image transfer        | HD720P30 |
| UAV Max Speed         | 8 m/s    |
| IR detection distance | 50 cm    |
| $d_{fov}$             | 41.5 deg |
| $d_{ir}$              | 20 deg   |
| $C_{horizontal}$      | -3 cm    |
| $C_{vartical}$        | 7 cm     |

실험을 위해 준비 단계에서 수행한 비례계수, 카메라 캘리브레이션, YOLO 및 기타 수행 환경에 대한 정보는 다음의 Table 2에서 확인할 수 있다[12].

Table 2 Specification of Environment

| Category           | Content              |
|--------------------|----------------------|
| <i>P</i>           | 960                  |
| Camera calibration | Zhengyou Zhang       |
| YOLO version       | Ultralytics YOLOv8   |
| model              | YOLOv8s              |
| CPU                | Intel i5-10210U      |
| GPU                | NVIDIA GeForce MX250 |
| CUDA               | 11.6                 |
| RAM                | DDR4 16384MB         |
| OS                 | Windows 10           |
| Language           | Python 3.9           |
| Image codec        | H.264                |
| Protocol           | UDP/IP - WiFi        |

### 3.3 실험 절차

YOLO 및 IR 센서의 정확도 측정은 장애물의 종류, 카메라 정면 기준 장애물까지의 거리(수직거리), 정면에서 수평으로 벗어난 거리(수평거리)를 변화시키며 수행한다. IR 센서 혹은 카메라를 중심으로 하는 원의 둘레를 따라 장애물이 이동해야 측정 센서와 장애물의 거리가 일정할 것이므로 수직거리에서 직각 방향이 아닌 호 모양을 그리며 이동시켰고, 해당 위치에 장애물 정면부의 중심이 오도록 배치하여 측정한다. 수직거리 및 수평거리에 대한 예시는 아래의 Fig. 8에서 확인할 수 있다.

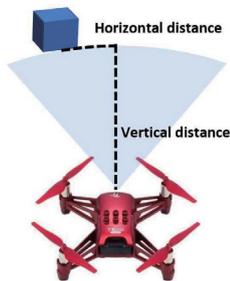


Fig. 8 Vertical and Horizontal Distances

장애물의 종류는 하단의 Fig. 9와 같이, 사과, 사람, 컵 3가지 종류로, 각 대상의 이미지를 출력 후 측정용 정육면체의 한 면에 부착하여 측정한다. 실제 장애물을 관측하는 방식이든 장애물의 상을 출력하여 관측하는 방식이든 둘 다 거리가 멀어짐에 따라 관측되는 상이 줄어드는 비율은 같으므로 동일한 관측 결과를 얻

을 수 있기에, 측정 대상의 크기 제어를 위하여 출력한 이미지를 통해 실험하였다. 측정용 정육면체는 5 x 5 x 5 cm 각 대상의 가로-세로 길이는, 사과는 4 x 5 cm, 사람은 2 x 5 cm, 컵은 5 x 5 cm이다.

장애물 및 센서에 따른 정확도 측정은 각 장애물에 대해 20cm, 30cm, 40cm의 수직거리의 변화 및 중심부에서 0~5cm의 좌측 수평거리 변화를 주며 수행하였으며 정확성을 위해 각각 위치마다 10회씩 반복 측정하였다. 정리하면, 장애물의 종류는 3개이고, 수직거리의 변화는 3개, 수평거리의 변화는 6개이며 각 위치마다 10회씩 측정했으므로, 총 540회 측정하였다.

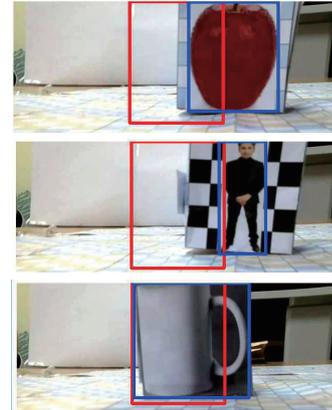


Fig. 9 Obstacles to Measure

RQ1의 IR 센서의 값은 거리 측정과 관련되며, 거리 측정의 참값은 곧 실험의 독립변수이므로 확인할 수 있다. 이에 반해, RQ2의 YOLO가 측정한 이미지 내 장애물 크기에 대한 참값은 이미지 내 장애물이 차지하는 픽셀(Pixel)의 개수가 되므로 정확한 참값을 알기 위해선 이를 일일이 측정해봐야만 한다. 이는 노동 집약적 작업이므로, 그 대신 RQ2에서는 IR 센서의 값을 실제값으로 둔 상태에서 YOLO가 측정한 크기를 Eq. 2에 대입하여 예측한 장애물의 크기와 실제 장애물의 크기를 비교한다. 또한 RQ3의 장애물 크기 예측의 정확도는 RQ1과 RQ2에서 구한 거리값 및 이미지 내 장애물 크기를 통해 도출한다.

### 3.4 실험 척도

RQ1은 IR 센서의 거리값에 대한 정확도를, RQ2는 거리값을 참으로 두었을 때의 크기 예측에 대한 정확

도를, RQ3은 이전 두 실험에서 얻은 결과를 종합하여 구한 크기 예측 정확도를 평가한다.

각 측정에 대한 정확도는 오차율에서 1을 감한 값으로, 오차율은 추정된 값과 실제값(참값) 사이의 차이를 백분율로 표현한 값이다. 추정된 값은 값들의 평균을 사용한다. 정확도에 대한 수식은 아래의 Eq. 3에서 확인할 수 있다.

$$Rate_{accuracy} = \left\{ 1 - \left| \frac{val_{estimate} - val_{actual}}{val_{actual}} \right| \right\} \times 100 [\%]$$

-  $Rate_{accuracy}$  : Accuracy  
 -  $val_{estimate}$  : Estimated values, measured values  
 -  $val_{actual}$  : Actual value (true value)

Eq. 3 Equation for Error Rate

## 4. 실험 결과

### 4.1 RQ1 결과

RQ1은 IR 센서의 거리 측정이 정확한지에 대한 연구 질문으로, 장애물 종류, 수직거리, 수평거리의 변화에 따라 평균적으로 91.54%의 정확도를 보였다. 약 20 cm에서 40 cm 사이의 각 장애물에 따른 최소, 최대, 평균 거리 측정 정확도는 하단의 Table 3에 표현되어 있다.

Table 3 Minimum, Maximum, Average Accuracy of IR Sensor

| Target          | Min[%] | Max[%] | Avg[%] |
|-----------------|--------|--------|--------|
| Apple           | 80.00  | 100.00 | 91.96  |
| Person          | 77.50  | 100.00 | 89.98  |
| Cup             | 80.00  | 100.00 | 92.68  |
| Overall average |        |        | 91.54  |

실험 결과에 대한 전반적인 결과는 하단의 Fig. 10과 같이 거리 단위가 cm인 평면 위에 표현했다. 평면은 실제 실험에 사용된 배치를 묘사했으며, 각 위치에 따른 정확도는 점을 통해 표현했다. 점의 크기가 클수록 정확도가 높음을 뜻한다. 단, 가시성의 극대화를 위해 각 정확도는 최대-최소 정규화(Min-Max Scaling)를 통해 범위를 [0,1]로 조정하였으며 실제 측정된 결과

인 좌측의 붉은 점을 세로축에 대해 대칭이동하여 회색 점으로 표현했다. 각 점의 투명도는 90%로 설정 후 3회의 측정 결과들을 누적하여 표현했으므로, 점의 가장자리가 선명할수록 측정값들의 표준편차가 적음을 의미한다. 하단의 Fig. 10에서 가장 작은 점이 Table 3에서의 최소 정확도에 해당하며, 가장 큰 점이 Table 3에서의 최대 정확도에 해당한다.

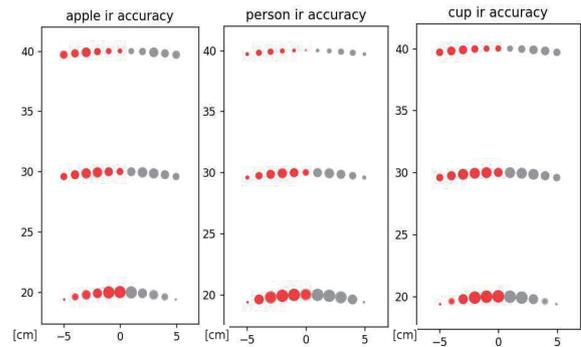


Fig. 10 Distance Measurement Accuracy of IR Sensor

Figure 10의 결과를 보면, 전반적으로, 수직거리가 멀어질수록 정확도가 떨어지며 실제 거리보다 더 먼 거리를 출력하는 경향이 있다. 하지만 특이하게도, 수평거리에 따른 정확도는 수직거리마다 달라짐을 보인다. 수직거리가 20 cm일 때는 수평거리가 멀어질수록 정확도가 떨어지는 경향을 보이나, 수직거리가 30 cm일 때는 수평거리 3 cm 부근에서 가장 정확하고, 수직거리가 40 cm일 때는 수평거리 4 cm 부근에서 가장 정확함을 확인할 수 있다.

장애물에 대해선, 빨간 사과와 흰색 컵 형상에 비해 검은색 복장의 사람 형상이 거리에 따른 정확도 차이가 두드러지나, 이는 최대-최소 정규화로 인한 사람 형상의 측정값 내 상대적 차이이며, 다른 장애물들과의 실제 정확도 차는 3% 미만으로 미미하다.

정리하면, IR 센서의 거리 측정은 평균적으로 91.54%의 정확도를 보이며, 실제 거리보다 더 먼 거리를 출력한다. 정확도에 영향을 주는 주요 변수는 수직거리로, 수직거리가 멀어질수록 정확도가 감소한다. 수평거리 및 장애물의 종류는 큰 영향을 주지 않았다.

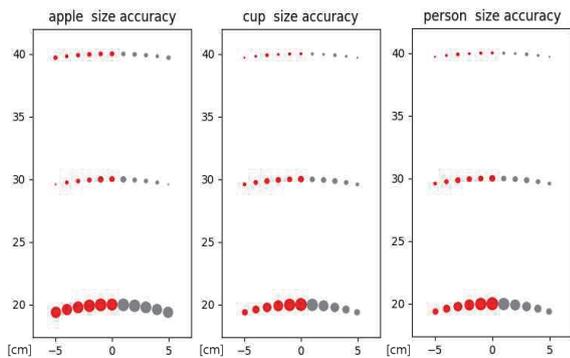
### 4.2 RQ2 결과

RQ2는 YOLO의 영상 내 장애물의 크기 측정이 정확 한지에 대한 질문으로, 장애물 종류, 수직거리, 수평거 리의 변화에 따라 평균적으로 95.04.%의 정확도를 보 였다. 약 20 cm에서 40 cm 사이의 각 장애물에 따른 최소, 최대, 평균 정확도는 하단의 Table 4에 표현되 어 있다.

**Table 4** Minimum, Maximum, Average Accuracy of YOLO

| Target          | Min[%] | Max[%] | Avg[%] |
|-----------------|--------|--------|--------|
| Apple           | 92.19  | 100.00 | 98.55  |
| Person          | 89.69  | 98.44  | 93.41  |
| Cup             | 90.00  | 95.83  | 93.15  |
| Overall average |        |        | 95.04  |

실험 결과에 대한 전반적인 결과는 하단의 Fig. 11과 같이 거리 단위가 cm인 평면 위에 표현했다. 이는 거 리를 참값으로 두고 YOLO의 데이터만 측정하여 장애 물의 크기를 예측했을 때의 정확도를 의미한다. 측정 한 높이 및 너비의 평균값으로 정확도를 산출했으며, 그 외의 표현 방법은 RQ1과 같다. 마찬가지로, 하단의 Fig. 11에서 가장 작은 점이 Table 4에서의 최소 정 확도에 해당하며, 가장 큰 점이 Table 4에서의 최대 정확도에 해당한다.



**Fig. 11** Size Measurement Accuracy of YOLO

전반적으로, 수직거리가 멀수록 정확도가 감소하는 경향이 보인다. YOLO의 영상 내 장애물의 측정은 평균 적으로 95.04.%의 정확도를 보인다. 정확도에 영향을 주는 변수는 장애물의 종류, 특히 장애물의 색상으로,

밝은색의 계열은 정확도가 감소할 것으로 추정되나 이 변수 부분이 많은 영향을 주는 것은 아니다.

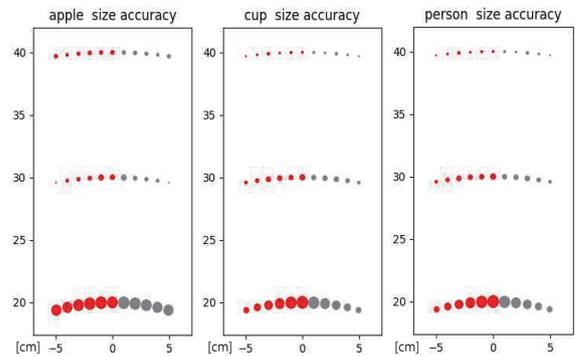
### 4.3 RQ3 결과

RQ3은 제시된 방법의 장애물의 크기 예측이 정확 한지에 대한 연구 질문으로, 장애물 종류, 수직거리, 수 평거리의 변화에 따라 평균적으로 86.39%의 정확도를 보였다. 약 20cm에서 40cm 사이의 각 장애물에 따른 최소, 평균 정확도는 하단의 Table 5에 표현되어 있 다.

**Table 5** Minimum, Maximum, Average Accuracy of the Proposal Method

| Target          | Min[%] | Max[%] | Avg[%] |
|-----------------|--------|--------|--------|
| Apple           | 76.02  | 99.79  | 91.19  |
| Person          | 69.13  | 98.23  | 82.70  |
| Cup             | 68.75  | 95.83  | 85.28  |
| Overall average |        |        | 86.39  |

실험 결과에 대한 전반적인 결과는 하단의 Fig. 12와 같이 거리 단위가 cm인 평면 위에 표현했다. 이는 RQ1에서의 거리 측정값과 RQ2에서의 이미지 내 크기 측정값을 사용하여 장애물의 크기를 예측했을 때의 정 확도를 의미한다. 예측한 높이와 너비의 평균값으로 정확도를 산출했으며, 그 외의 표현 방법은 RQ1과 같 다. 마찬가지로, 하단의 Fig. 12에서 가장 작은 점이 Table 5에서의 최소 정확도에 해당하며, 가장 큰 점이 Table 5에서의 최대 정확도에 해당한다.



**Fig. 12** Size Measurement Accuracy of Proposal Method

전반적으로, RQ2의 결과와 유사한 형태를 보이며, IR 센서가 실제 거리보다 더 큰 값을 반환하는 경향이 있었기에 본 결과 또한 실제 크기보다 좀 더 크게 예측하는 경향이 존재한다.

이러한 원인은 크기를 예측하는 수식인 Eq. 2에서 확인할 수 있다. IR 센서가 반환하는 거리와 YOLO가 반환하는 영상 내 장애물 크기는 서로 곱해져 있으나, 본 실험에서 사용된 거리의 범위가 영상 내 장애물 크기보다 범위보다 훨씬 작기에, 거리의 변화가 결과에 더 많은 영향을 미친다. 같은 이유로, YOLO의 정확도가 높음에도 IR 센서의 부정확함이 더 크게 증폭되어 크기 예측 결과의 정확도가 두 센서의 정확도에 비해 더 낮아졌다.

정리하면, 제시된 방법의 영상 내 장애물의 측정은 평균적으로 86.39%의 정확도를 보이며 실제 크기보다 더 크게 예측하는 경향이 있다. 정확도에 영향을 주는 주요 변수는 수직거리로, 수직거리가 멀어질수록 정확도가 감소한다.

## 5. 사례 연구

본 논문에서 제시한 장애물 크기 예측 방법은 소형 UAV의 장애물 충돌 회피를 목표로 한다. 따라서 본 방법을 기반으로 한 소형 UAV의 충돌 회피 시스템을 실제로 구현 후 회피 성능을 확인해보았다. 시스템의 구현은 인지한 장애물의 위치 및 크기를 바탕으로 장애물을 최단 거리로 회피할 수 있는 방향을 결정 후 안전거리를 확보하도록 설정했으며, 구현 환경은 3절의 실험과 같다. 이때, 사용한 UAV인 RMTT는 내장 프로세서에 로직의 삽입이 불가능하기에 외부 컴퓨터에서 WiFi를 통해 UAV를 조작하였다.

시나리오는 호버링 중인 UAV에 장애물이 다가오는 경우와 정적 장애물에 UAV가 다가가는 경우를 선정했고, 이에 관해 충돌 회피를 성공적으로 수행하는지 확인했다. 수행 영상은 참조된 유튜브 링크<sup>1)</sup>에서 확인할 수 있다. 수행 결과, 각 시나리오에서 장애물의 크기를 인지 후 가장 빠르게 회피가 가능한 방향으로 성공적으로 회피하였다. 이는 단순히 단안 카메라만 존

재하거나 IR 센서만 존재하는 UAV에서는 도출되기 어려운 결과로, 보다 많은 정보량을 토대로 소형 UAV의 장애물 인지 정확성을 높일 수 있는 가능성을 보여준다.

하지만 구현에서 장애물의 위치를 추적 후 장애물의 운동방향을 도출하는 부분이 없었기에 동적 장애물의 운동을 예측 후 회피할 수는 없었으며, YOLO의 특성상 원뿔형과 같이 정면과 측면의 길이 차이가 큰 대상은 크기를 명확히 예측하기 어렵다는 한계가 존재했다. 따라서 향후 연구에서는 장애물의 크기 예측을 이미지 프레임 단위로 수행하는 것이 아닌, 일정량의 이미지 프레임 단위로 수행함을 통해 장애물의 운동 정보 또한 도출하는 과정을 추가할 예정이다. 또한, 단순히 장애물의 종류 및 외곽선을 인지하는 것을 넘어 장애물 내 명암을 통해 입체적인 크기를 도출하는 모델을 연구하여 상기된 사항들을 보완할 예정이다.

## 6. 논 의

소형 UAV는 페이로드 및 장치 부착을 위한 공간이 부족하기에, 전력 소모 및 중량을 최대한 줄여야 하므로 기존에 많이 사용되는 센서나 장치들을 활용하기에 어려움이 따른다. 본 제안 방법은 일반적으로 작고 가벼우며 전력 소모가 적다고 알려진 IR 센서 및 단안 카메라만을 측정 센서로 사용하여 소형 UAV에 대한 접목성을 높였다.

본 논문에서 제시한 방식은 시각 기반의 정보를 활용하는 방식에 관한 구체적인 사례 및 이에 대한 이점과 가능성을 보여준다. 궁극적으로는 곤충과 같은 초소형의 에이전트에도 도입할 수 있는 시각 정보 기반 탐지 기법의 개발로 이어질 수 있다. 그 외에도 기존에 존재하는 방법들과 복합적으로 사용한다면 중복 탐제를 통한 결함 감내(Fault-Tolerance)에 기여할 수 있다.

본 방법의 제약 사항은 IR 센서의 낮은 정확도에 있다. YOLO가 상당히 높은 정확도를 보여줌에도 불구하고, IR 센서는 가볍다는 이점 외에는 제한된 환경에서도 낮은 정확도를 지닐 수밖에 없다는 한계점이 존재하므로 이를 보정하기 위한 후속 연구가 필요하다. 다만 그럼에도 86.39%의 정확도는 20 cm 크기의 장애물을 23 cm 정도로 예측하는 정도의 수준으로, 소형

1) <https://www.youtube.com/watch?v=plu39oBKIGY>

UAV의 크기와 기동성을 고려했을 때 유용하게 사용될 수 있을 것이다. 또한 기존의 소형 UAV에 사용하던 단안 카메라 기반 방식과는 달리, 장애물의 크기 정보를 도출하므로 장애물의 좌표화가 가능하여 보다 다양한 회피 알고리즘 및 작업을 위해 사용될 수 있다.

YOLO는 본 연구에서 정확성이 높게 측정이 되었으나, 딥러닝 모델은 블랙박스의 성질을 지니고 있기에 언제 어떤 대상을 인지하지 못할지, 못할 경우 왜 못했는지 밝혀내는 것이 어렵다는 위험성이 존재한다. 또한 학습되지 않은 장애물에 대해서는 인지가 불가하다. 이에 따라 설명 가능한 인공지능(eXplainable AI, XAI)과 같이 AI가 도출한 이미지에 대한 설명을 통해 학습을 보다 강화할 수 있는 방법, zero-shot과 같이 기존에 학습하지 않았음에도 이미지를 식별하는 방법 등이 연구되고 있으므로, 이에 관한 연구 및 접목 또한 고려해볼 수 있다[13-18].

## 7. 결 론

본 논문에서는 단안 카메라-YOLO 객체 검출 모델 및 IR 센서를 결합하여 장애물의 크기를 예측하는 방법을 제안하였다. 제안된 방법의 구현은 장애물의 종류 및 위치의 변화에도 크기 예측에 있어 평균적으로 86.39%의 정확도를 보였으며, IR 센서가 거리를 실제보다 크게 측정하는 경향이 있어 크기 예측 또한 실제보다 크게 예측하는 경향이 있었다. 실험을 통해, 크기 예측의 정확도에는 장애물까지의 거리가 가장 큰 영향을 미침을 확인했다.

제시한 방법은 작고 경량화되어 소형 UAV에 접목이 용이한 센서를 사용하기에, 소형 UAV의 충돌 회피에 기여한다. 향후 연구에서는 더 정확한 거리 측정 기술 및 더욱 발전된 객체 검출 모델과 기타 보정 기술들을 활용하여 더욱 향상된 장애물 인지 방법을 연구할 수 있을 것으로 기대된다.

## 후 기

본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 3단계 산학협력 선도대학 육성사업(LINC 3.0) 및 중견연구과제(No. NRF-

2021R1A2C1094167)의 연구결과입니다.

## Reference

- [1] L. Lu, G. Fasano, A. Carrio, M. Lei, H. Bavle, and P. Campoy, "A comprehensive survey on non-cooperative collision avoidance for Micro Aerial Vehicles: Sensing and obstacle detection," *Journal of Field Robotics*, vol. 40, no. 6, pp. 1697-1720, 2023.
- [2] B. Khmel, H. Ghadia, and S. Bhandari, "Collision avoidance system for a multicopter using stereoscopic vision with target detection and tracking capabilities." *AIAA SCITECH 2023 Forum*, pp.1147, 2023.
- [3] M. Likhita Nagendra Sai Sumanth, Advait Ashwin Harish, Remidi Rohith Reddy, K. A. Nethravathi & M. Uttara Kumari, "Obstacle detection in autonomous vehicles using 3D lidar point cloud data." *Data Intelligence and Cognitive Informatics*, , pp. 745-757, 2022.
- [4] Z. Shang and Z. Shen. "Topology-based UAV path planning for multi-view stereo 3D reconstruction of complex structures." *Complex & Intelligent Systems*, vol. 9, no. 1, pp. 909-926, 2022.
- [5] M. Y. Arafaty, M. M. Alam, and S. Moh, "Vision-based navigation techniques for unmanned aerial vehicles: Review and Challenges." *Drones*, vol. 7, no. 2, p. 89, 2023.
- [6] G. Petrakis, Angelos Antonopoulos, Achilles Tripolitsiotis, Dimitris Trigkakis & Panagiotis Partsinevelos, "Precision mapping through the stereo vision and geometric transformations in unknown environments." *Earth Science Informatics*, vol. 16, no. 2, pp. 1849-1865, 2023.
- [7] J. Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You only look once: Unified, real-time object detection." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [8] Chiang, M. L., Tsai, S. H., Huang, C. M., & Tao, K. T, "Adaptive visual servoing for obstacle

- avoidance of micro unmanned aerial vehicle with optical flow and switched system model". *Processes*, 9.12: 2126, 2021.
- [9] Badrloo, S., Varshosaz, M., Pirasteh, S., & Li, J. "Image-based obstacle detection methods for the safe navigation of unmanned vehicles: A review", *Remote Sensing*, 14.15: 3824, 2022.
- [10] Karlsson, S., Kanellakis, C., Mansouri, S. S., & Nikolakopoulos, G. "Monocular vision-based obstacle avoidance scheme for micro aerial vehicle navigation" In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, p. 1321-1327, 2021.
- [11] Lee, H. Y., Ho, H. W., & Zhou, Y, "Deep Learning-based monocular obstacle avoidance for unmanned aerial vehicle navigation in tree plantations: Faster region-based convolutional neural network approach". *Journal of Intelligent & Robotic Systems*, 101: 1-18, 2021.
- [12] Y. J. Zhang, "Camera calibration." *3-D Computer Vision*, pp. 37-65, 2023.
- [13] D. Gunning and D. Aha, "DARPA's Explainable Artificial Intelligence (XAI) Program." *AI Magazine*, vol. 40, no. 2, pp. 44-58, 2019.
- [14] S. H. Han, Min-Su Kwon, Ho-Jin Choi, "Explainable AI (XAI) approach to image captioning." *The Journal of Engineering*, vol. 2020, no. 13, pp. 589-594, 2020.
- [15] B. H. M. Van der Velden, Hugo J. Kuijf, Kenneth G.A. Gilhuijs, Max A. Viergever, "Explainable artificial intelligence (XAI) in deep learning-based medical image analysis," *Medical Image Analysis*, vol. 79, pp. 102470, 2022.
- [16] B. Li, Y. Gou, H. Zhu, and X. Peng, "Zero-shot image Dehazing." *IEEE Transactions on Image Processing*, vol. 29, pp. 8457-8466, 2020.
- [17] G. Parmar, "Zero-shot image-to-image translation." *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, 2023.
- [18] Z. Novack, J. McAuley, and S. Garg, "Chils: Zero-shot image classification with hierarchical label sets." *International Conference on Machine Learning*, pp. 26342-26362, July 2023.