

R7020E Final Project

Group 6

Jose Juan Pena Gomez (jospen-3)

Sushanta Mohapatra (susmoh-3)

31 October 2024

Index

1. Introduction
2. Object Detection
3. Decoy Filtering
4. Position Estimation
5. Discussion and Conclusion

Introduction

The aim of this project is to develop a system capable of detecting fire extinguishers in images, distinguishing between real and decoy objects, and estimating the object's relative distance using depth information. The pipeline is composed of three components:

- Object Detection
- Decoy Filtering
- Position Estimation

Object Detection

Models were trained on the Roboflow [FireExtinguisher](#) dataset, which includes images of fire extinguishers among other objects. The models were trained using the YOLOv11 and Faster R-CNN architectures over 100 epochs on a single NVIDIA RTX 2080 Ti GPU.

YOLOv11

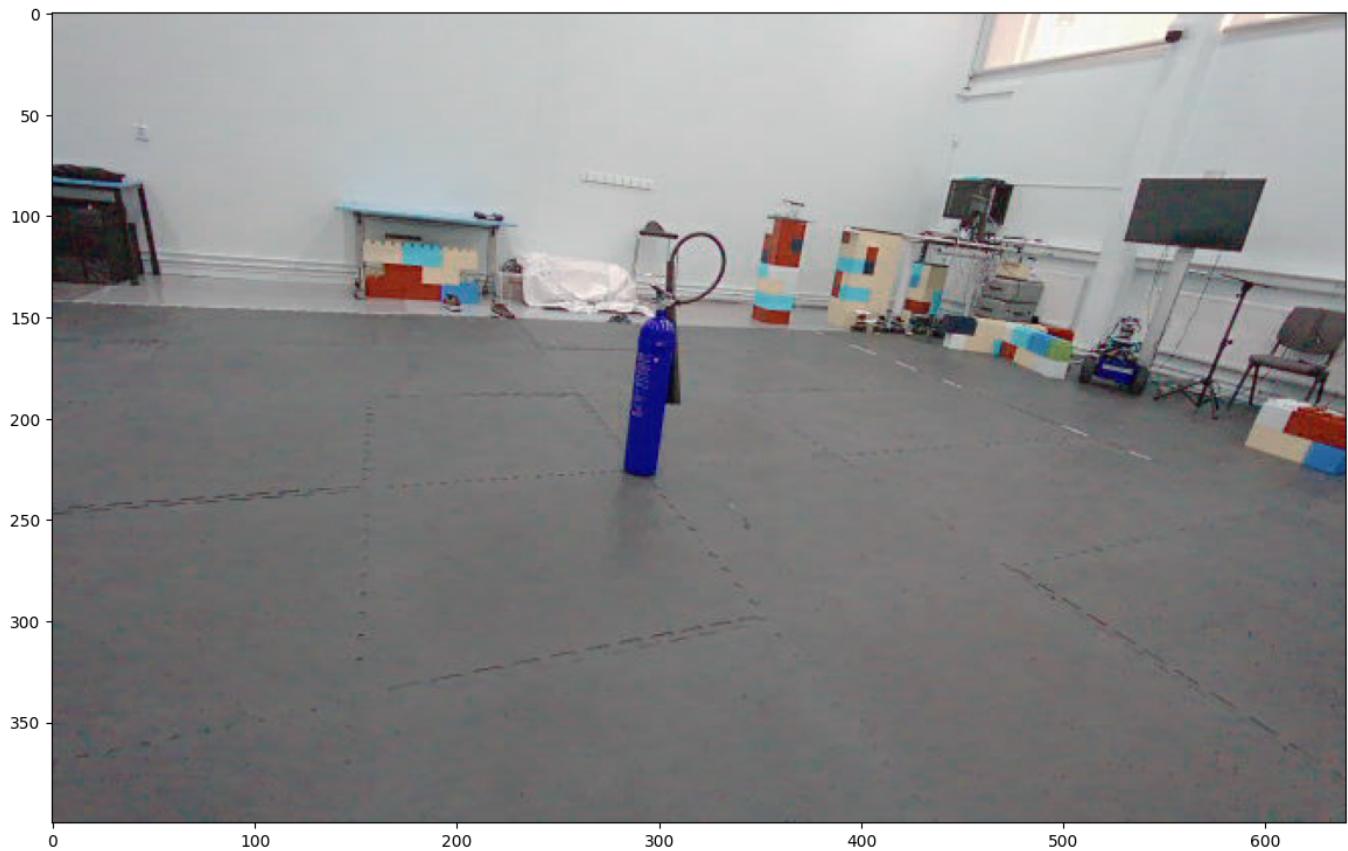


Figure 1: Original image in BGR Color Space

The YOLOv11 model was employed to detect objects within the dataset, as illustrated below.



Figure 2: Object Detection with YOLOv11

The model successfully detected the fire extinguisher with a confidence probability of 0.7.

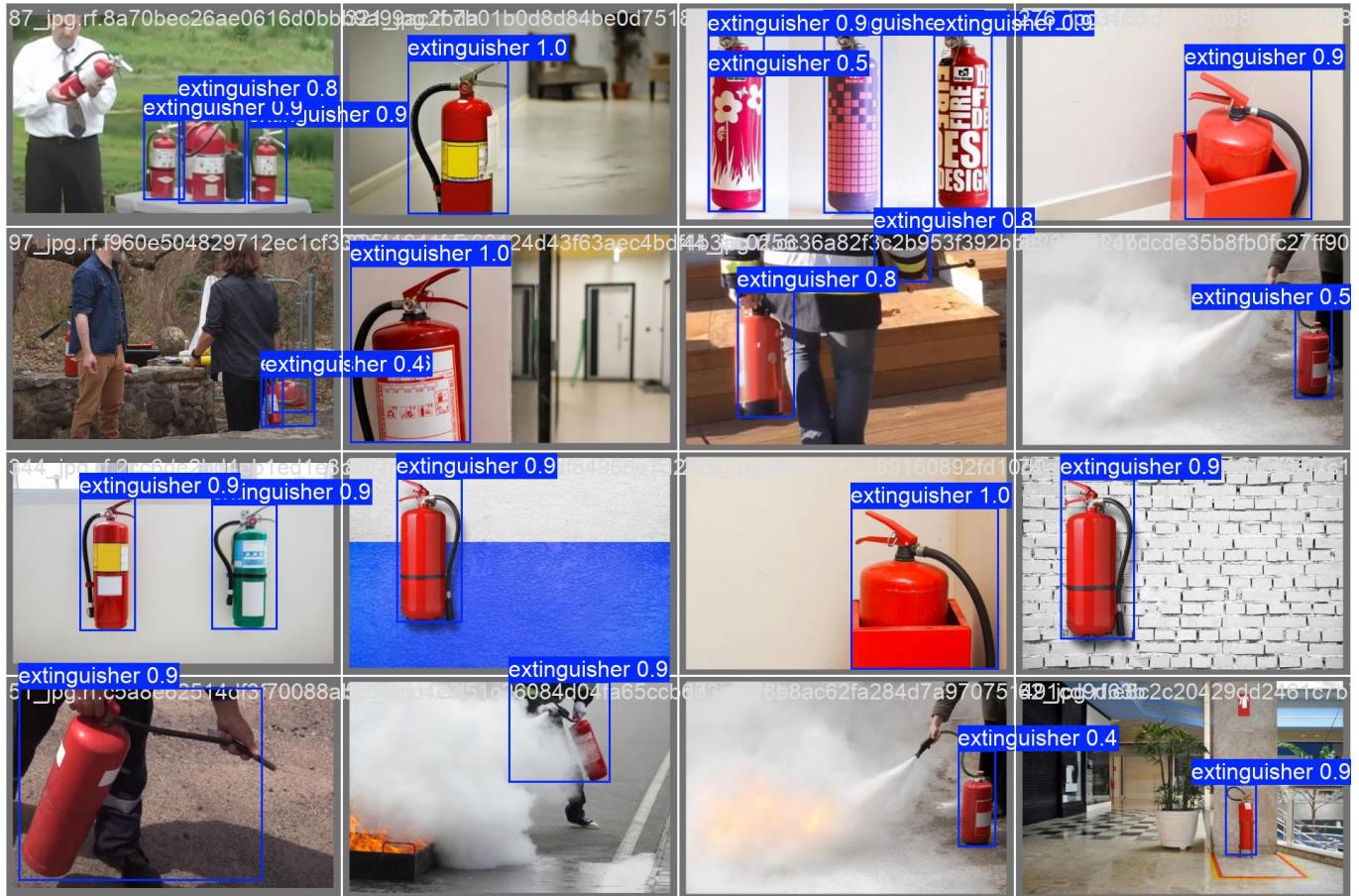


Figure 3: Fire Extinguisher Detection with YOLOv11

Validation dataset results showed a precision of 92% and a recall of 85%.

Metrics	Value
Precision	92.137%
Recall	85.076%

Table 1: YOLOv11 Results

Faster R-CNN



Figure 4: Fire Extinguisher Detection with Faster R-CNN

The Faster R-CNN model was used to detect objects within the dataset, as illustrated above. The model successfully detected the fire extinguishers.

Metrics	Value
Precision	61.55%
Recall	95.38%

Table 2: Faster R-CNN Results

Decoy Filtering

Decoy filtering is achieved through two methodologies:

- Feature Extraction using ResNet18 and K-Means Clustering
- Local Binary Pattern histogram for Texture Analysis

Feature Extraction using ResNet18 and K-Means Clustering

A pre-trained ResNet18 model extracts features from regions defined by bounding boxes in the images. These features are clustered using KMeans into two groups, effectively separating real fire extinguishers from decoys. The clustering results refine the bounding boxes to retain only those associated with real fire extinguishers.

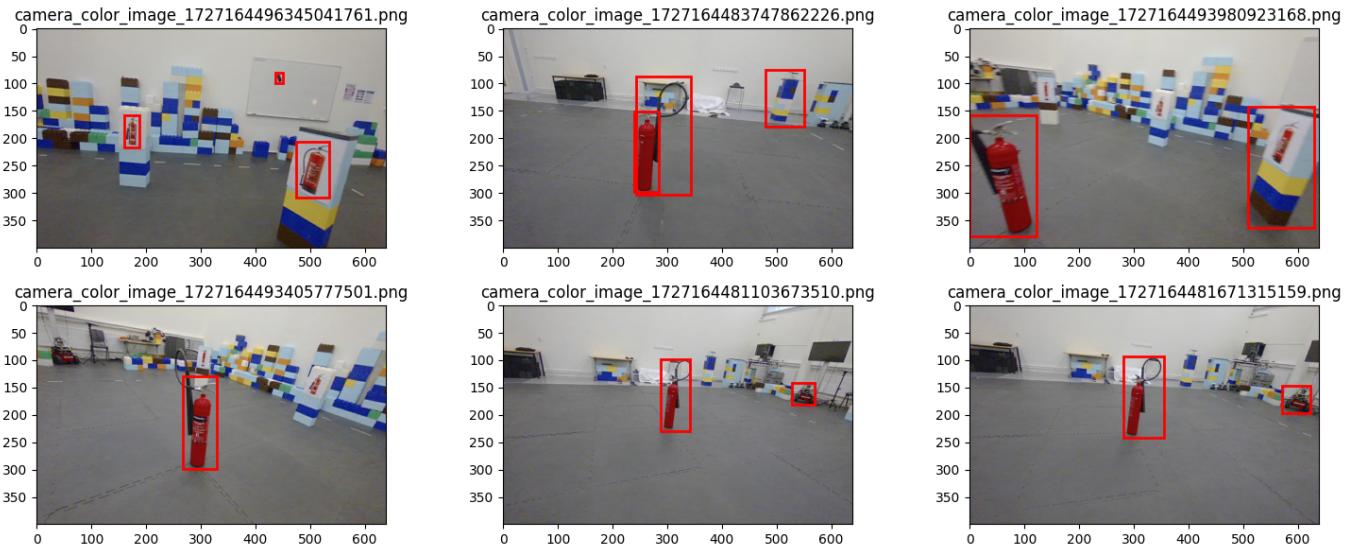


Figure 5: Object detection with Faster R-CNN before applying decoy filter

The image above shows the Faster R-CNN model's detection results before applying the decoy filter.



Figure 6: Object detection with Faster R-CNN after applying decoy filter using ResNet18 and K-Means

The image above shows the Faster R-CNN model's detection results after applying the decoy filter.

Local Binary Pattern histogram for Texture Analysis

The Local Binary Pattern (LBP) is a texture descriptor used to classify textures based on the patterns formed by pixel intensity values. The LBP histogram helps classify objects as real or decoys.

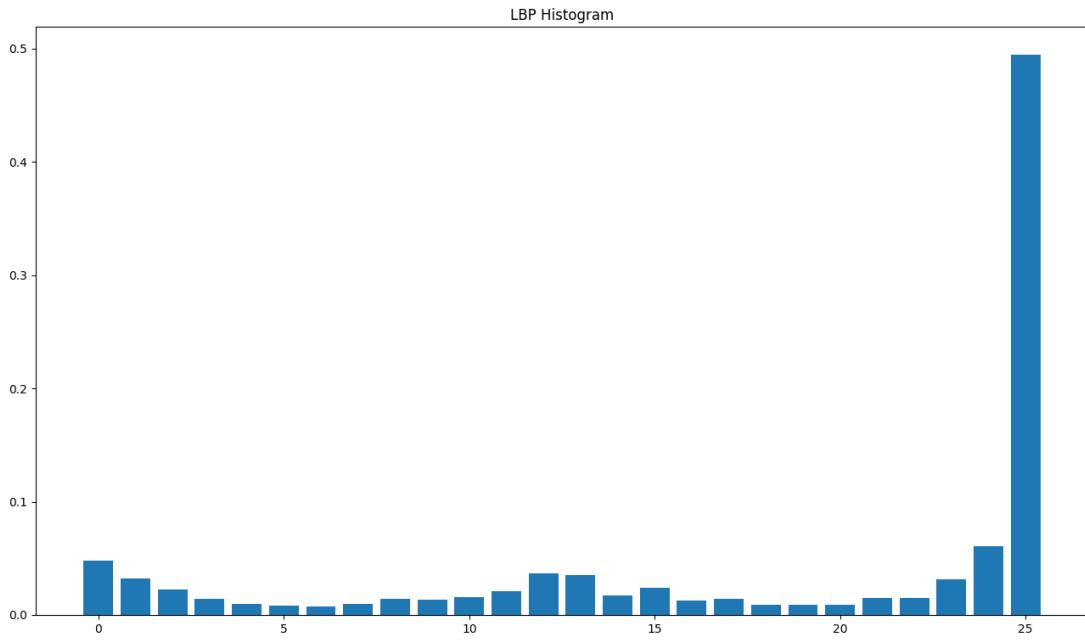


Figure 7: LBP Histogram for Texture Analysis

Position Estimation

Position estimation utilizes depth information from the image, obtained from a stereo camera's depth map. The depth map helps estimate the object's 3D position (X, Y, Z) based on its 2D position (x, y) and depth $D(x, y)$.

$$X = \frac{(x - c_x) * Z}{f_x}$$

$$Y = \frac{(y - c_y) * Z}{f_y}$$

$$Z = D(x, y)$$

where:

- (x, y) is the 2D position of the object in the image
- (c_x, c_y) is the principal point of the camera
- (f_x, f_y) is the focal length of the camera
- $D(x, y)$ is the depth at position (x, y)
- (X, Y, Z) is the 3D position of the object

Using Depth Map with YOLOv11

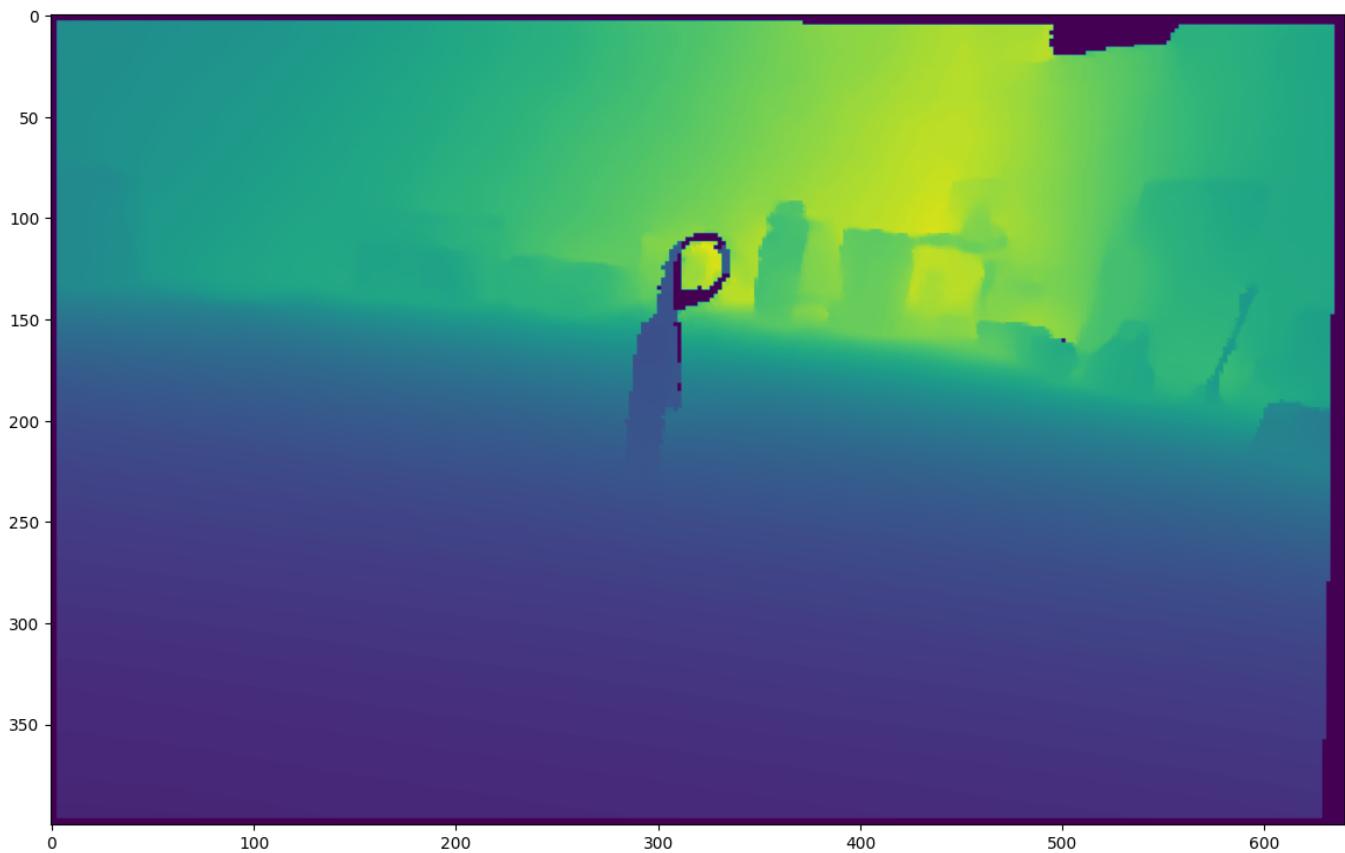


Figure 8: Depth Map

The depth map is used to estimate the 3D position of the object in the image.

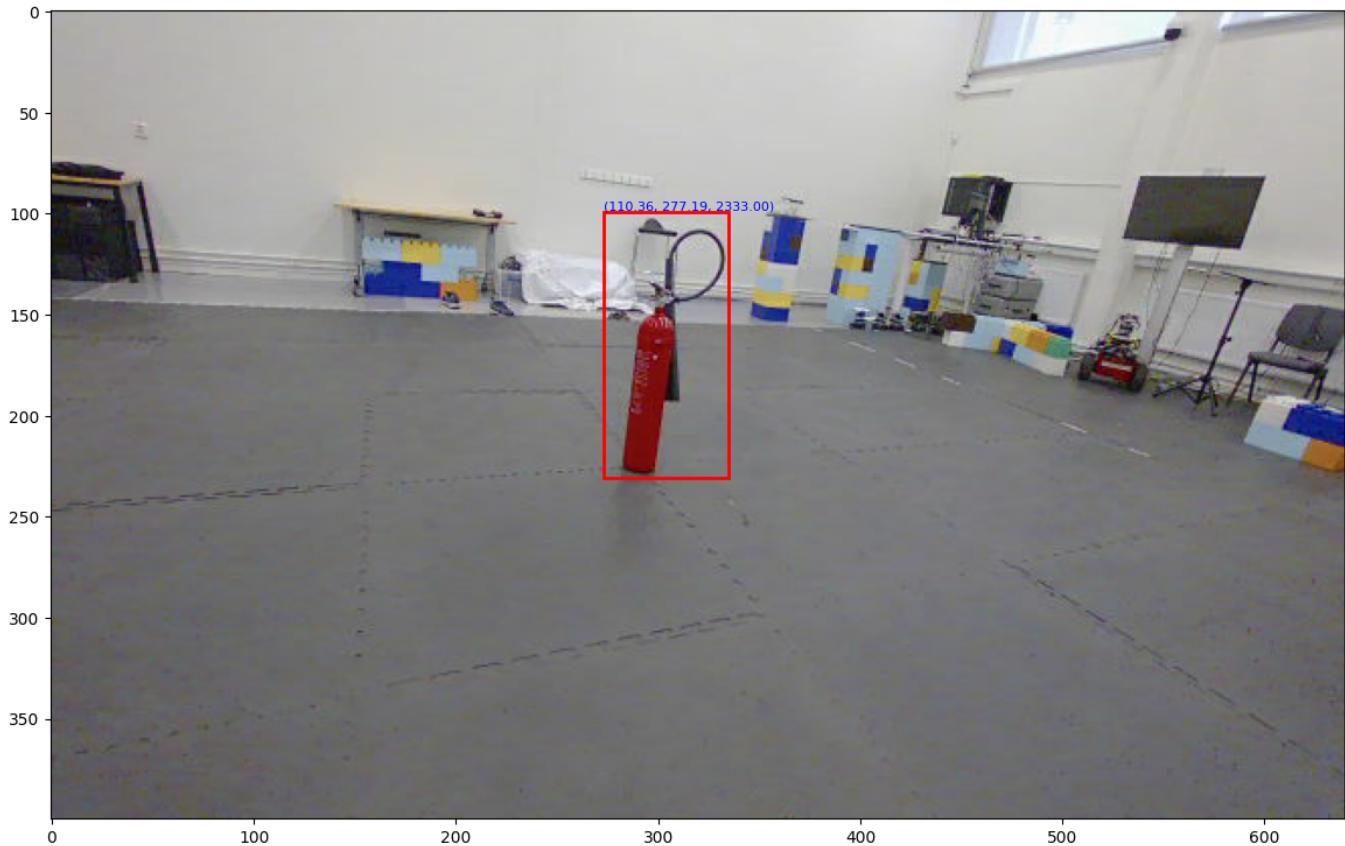


Figure 9: Object Detection with 3D Position Estimation

The 3D position of the object is estimated using the depth map.

Using Depth Map with Faster R-CNN

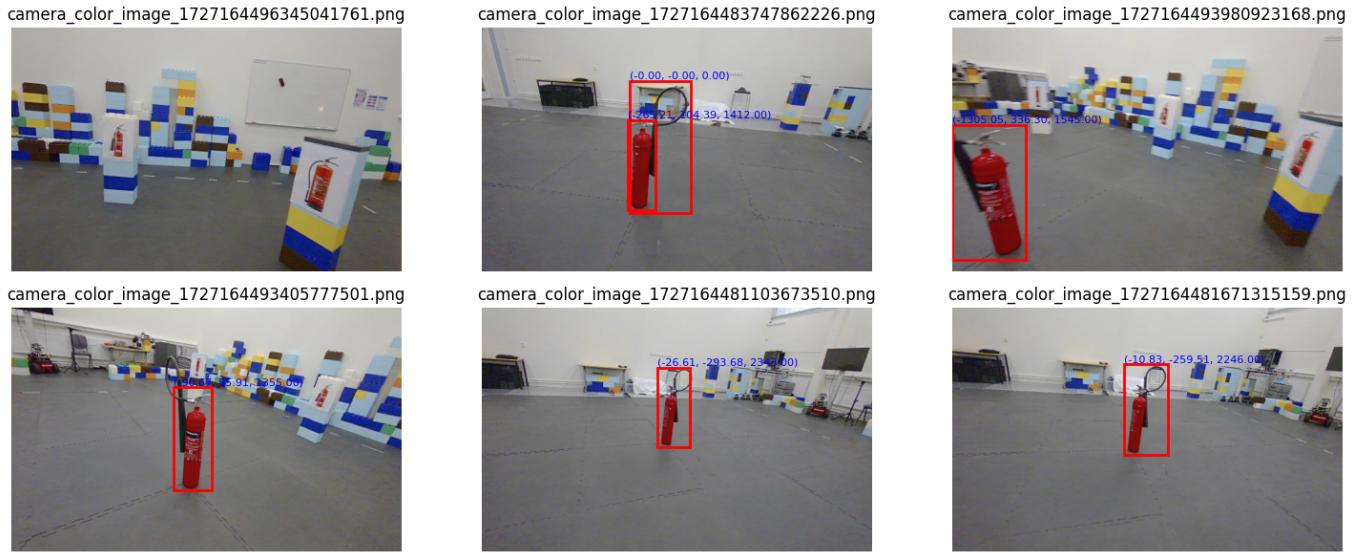


Figure 11: 3D position estimated with Depth images

Conclusion

- The trained YOLOv11 and Faster R-CNN models performed well on the validation dataset.
- YOLOv11 showed better precision, while Faster R-CNN had better recall.
- Decoy filtering using ResNet18 and KMeans clustering effectively distinguished real objects from decoys.
- Depth camera parameters and depth images were utilized to estimate the 3D position of detected objects.

Future improvements could include:

- Image rectification
- Fine-tuning texture analysis thresholds
- Investigating depth camera scale for better visualization.
- Integration with ROS could automate processes and enhance scene understanding.