©Copyright 2024
Apoorva Patil

Characterizing Robot Vision Solutions for Anomaly Detection in Confined Spaces

Apoorva Patil

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Mechanical Engineering

University of Washington

2024

Committee:
Ashis G. Banerjee
Santosh Devasia
Xu Chen

Program Authorized to Offer Degree:
Department of Mechanical Engineering

University of Washington

Abstract

Characterizing Robot Vision Solutions for Anomaly Detection
in Confined Spaces

Apoorva Patil

Chair of the Supervisory Committee:
Ashis G. Banerjee
Department of Industrial and Systems Engineering and Department of Mechanical Engineering

Industrial structures are often susceptible to damage due to environmental and human factors and need to be inspected to determine safety, longevity, damage repair requirements and strength of the structure. Confined spaces in mechanical structures contain featureless surfaces hindering visual recognition, and are conventionally inspected manually implementing a labor and cost intensive process, subjecting the inspectors to a prolonged exposure to dark, hazardous, stifling, tight spaces. I have developed a robotic framework for a ground robot enabled with vision sensors to perform scanning, and inspection of confined spaces in order to perform anomaly detection and anomaly classification. This research addresses unique anomalies such as rust patches and unconventional objects like industrial maintenance tools, and proposes a novel approach to evaluating object detection and classification by fusing RGB and Depth information in a computationally efficient technique. This study highlights the effect of vision sensor performance, controlled illumination, robot pose with respect to anomaly, integration of color and depth input, and the use of images of sections in a confined space, for anomaly recognition. The proposed system employs transfer learning, utilizing a partially pre-trained YOLOv8 model on a custom dataset of anomaly images and getting a result of mean Average Precision (at IoU50) = 0.83 for industrial tools and a mean average precision (at Iou50) = 0.54 for rust anomalies.

TABLE OF CONTENTS

Page

List of Figures iii

List of Tables v

Glossary vi

Chapter 1: Introduction 1
1.1 Background and Motivation 1
1.2 Scope of Work 2

Chapter 2: Literature Review 4
2.1 Point Cloud Based Anomaly Detection 4
2.2 RGB-D Image Based Object Detection 5

Chapter 3: Methodology 7
3.1 Hardware Integration 7
3.2 Prototype Environment Setup 13
3.3 Classification of Anomalies 14
3.3.1 Foreign Object Debris 14
3.3.2 Rust 15
3.4 Design of Experiments 16
3.5 Data Acquisition 21
3.6 Deep Learning based Object Detection Model 23
3.7 Pipeline 24

Chapter 4: Experimental Results 26
4.1 Software Implementation 26
4.2 Data Pre-processing 27

4.3 Data Augmentation 28
4.4 Results: FOD Detection 31
4.5 Results: Rust Detection 36

Chapter 5: Conclusions 43
5.1 Contributions 43
5.2 Future Work 44

Bibliography 47

LIST OF FIGURES

Figure Number Page

3.1 Modified TurtleBot3 used as a ground robot 7
3.2 Custom 3D mount for attaching vision sensors 9
3.3 (A1) Intel RealSense D435i camera, (A2) Color image (RealSense), (A3) Depth image (RealSense), (B1) OAK-D camera, (B2) Color image (OAK-D) and (B3) Depth image (OAK-D) 12
3.4 Graphical representation (CAD model) of prototype tank, (Left) and Bird-eye-view of actual prototype tank model (Right) used for data collection. . . 13
3.5 Classification of Foreign Object Debris (FODs) 14
3.6 Classification of Rust Anomalies 15
3.7 Documenting all the FOD and rust types and classes 16
3.8 Establishing the 4 poses for collecting images of each anomaly 18
3.9 Demonstrating one sample scenario containing 10 anomalies 19
3.10 Displaying 3 sample scenarios with various specific FODs and rust patches selected 20
3.11 Image of robot inspecting a tool in a sample scenario 21
3.12 Image of robot inspecting a tool in a sample scenario 22
3.13 Ground robot demonstrating an experiment of collecting images of anomalies from the prototype confined space 23
3.14 Figure displaying the framework setup for developing and testing a system for anomaly detection in confined spaces. 25

4.1 Figure displaying a sample image or a rust patch anomaly. On the left is the raw image and the figure on the right contains a Gaussian blur of 0.5 pixels added to it. 27
4.2 Sample image – raw (pre-processed), with vertical flip and with horizontal flip. 29
4.3 Sample image – raw (pre-processed), with clockwise (+90°) rotation, counterclockwise (-90°) rotation and with upside down (180°) rotation. 30
4.4 Images observed in one training batch by the YOLOv8 machine learning model white training on FOD detection and classification data. 32

4.5 YOLOv8 results for anomaly detection and classification of FODs with OAK-D. 33
4.6 YOLOv8 results for anomaly detection and classification of FODs with OAK-D with cosine learning rate scheduler. 34
4.7 YOLOv8 results for anomaly detection and classification of FODs with RealSense. 34
4.8 YOLOv8 results for anomaly detection and classification of FODs with RealSense with cosine learning rate scheduler. 35
4.9 Comparison between OAK-D (Left) and RealSense (Right) of YOLOv8 results for anomaly detection and classification of FODs. 36
4.10 Images observed in one validation batch by the YOLOv8 machine learning model white training on rust detection and classification data. 37
4.11 YOLOv8 results for anomaly detection and classification of rust with OAK-D. 38
4.12 YOLOv8 results for anomaly detection and classification of rust with OAK-D with cosine learning rate scheduler. 38
4.13 YOLOv8 results for anomaly detection and classification of rust with RealSense. 39
4.14 YOLOv8 results for anomaly detection and classification of rust with RealSense with cosine learning rate scheduler. 40
4.15 Comparison between OAK-D metrics and RealSense of YOLOv8 results metrics of precision and recall for anomaly detection and classification of rust. 41
4.16 Comparison between OAK-D (Left) and RealSense (Right) of YOLOv8 results of confusion matrices for anomaly detection and classification of rust. 41

5.1 Camera performance comparison based on color and contours generated from depth images of a sample scenario. RealSense color image (Top left), RealSense contour plot (Top right), OAK-D color image (Bottom left), OAK-D contour plot (Bottom right) 45

LIST OF TABLES

Table Number Page

3.1 OAK-D vs. RealSense: comparison of key features 11

GLOSSARY

FOD: Foreign Object Debris; This can be any object that is not native to the environment that we are considering.

IOU: Intersection Over Union; In object detection or classification, objects are identified by establishing a bounding box around them during the labeling phase. Machine learning or deep learning models produce a similar output bounding box during validation or testing. When evaluating model performance metrics, the intersection of the area of the two bounding boxes over the union of the area of the two bounding boxes is called the IoU metric. This indicates how closely the predicted area of the anomaly overlaps with the true area of the anomaly.

SLAM: Simultaneous Localization And Mapping; This technique used by autonomous vehicles lets you build a map and localize your vehicle in that map at the same time.

YOLO: You Only Look Once; A deep learning model used for object detection based on image or video data of the objects.

ROS: Robot Operating Software; This is a robotic middleware platform that allows for software - hardware integration, robot software development and robot control.

APRILTAGS: A system of visual tags developed to provide high accuracy localization for different applications. This research uses the tags for establishing the ground truth information of anomalies.

FASTER RCNN: Faster Region-based Convolutional Neural Network; This is an object detection model that is faster and more accurate and can produce good results. It is often used as a comparison model when analyzing the performance of YOLO.

EXIF: Exchangeable Image File Format; This is a standardized way for digital cameras to store metadata in image files. This data includes technical information about how the image was created, such as the camera and lens used, the shooting settings, and the time and date it was taken.

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to my research advisor and Master's thesis committee chair, Professor Ashis G. Banerjee for his unconditional support and guidance. I would like to thank my thesis committee members, Professor Santosh Devasia and Professor Xu Chen for their mentorship. I would like to express my sincere appreciation to the staff and faculty at the Department of Mechanical Engineering. I would like to thank the members of the SMARTS Lab at the University of Washington, in particular Ekta Samani, Ryan Lee, Shankruth Balasubramaniyam, Benjamin Wong, Tommy Zhang, Marina Ruediger and Vamsi Jonnalagadda for their invaluable insights, support and assistance throughout my research endeavor. I would also like to express profound gratitude to thank my parents, brother, family and friends for their continued support and encouragement throughout my academic journey.

DEDICATION

to all those whose unconditional love and guidance fuelled my journey

Chapter 1

INTRODUCTION

1.1 Background and Motivation

Industrial structures such as warehouses, factories, large scale vehicular bodies, and aircraft and marine platforms often contain confined spaces. According to OSHA (Occupational Safety and Health Administration), confined spaces such as tanks, vessels, silos, vaults and tunnels can be large enough for people to enter and perform certain tasks in, but often have limited or restricted means for entry or exit and are not designed for continuous occupancy. Permit required confined spaces are those that have a potentially hazardous atmosphere [16], and these especially pose to be obstacles for tasks of inspection and maintenance.

Manual inspection in such surroundings often encompasses implementing safety checks, providing for equipment that can be carried and used by humans outside of protective gear, time limitations based on human ability to access these spaces, limited speed of inspection owing to human fatigue and exhaustion, involving a multi-member team of humans to coordinate inspection and requiring the application of comprehensive and detailed inspection procedures and techniques by all personnel who are specifically trained for that purpose. This creates a need for pre-inspection planning and administrative arrangements, identification of areas to be inspected, the relevant standards that apply, the possible damage that may be found or sample results that could constitute of a mild or significant structural risk and the documentation that must be reviewed before, during and after the inspection process [14].

Implementing a pipeline for remote inspection using robots can provide significant advantages including lower risk of human life, minimal labor requirement, reduced inspection costs, uninterrupted inspection cycles, scope of executing multiple inspections in parallel,

scalability, robust inspection and improved inspection efficiency. Robotic inspection can be carried out around the clock and the robots can be programmed to learn inspection rules, regulations and constrains without needing individual training for the same. Robotic inspection of confined spaces has the added advantage of being able to be performed on multiple areas synchronously.

The marine industry has a plethora of areas and enclosures that can be categorized as confined spaces. Marine vessels which operate in open-ocean conditions are very prone to weathering and degradation, thus generating the need for frequent inspection and repair which proves to be a very expensive endeavour. Some of the most challenging areas to inspect in such environments are confined spaces [22].

1.2 Scope of Work

Object detection and classification is a prominent topic in the field of computer vision. Deep learning technology has become a keyword in recent times due to the state-of-the-art results obtained in the domain of object detection, natural language processing and image classification [15]. At a time when mechatronics and automation are redefining the face of every industry, developing robust robotic systems that can support high throughput and increase efficiency has become imperative. In the field of deep learning, object detection techniques and models that use distance information as input are actively studied for utilizing automated locomotion. However, the conventional object detectors often have large network structures, and their real-time detection property is impaired [19]. The use of RGB-D images for anomaly and object detection is an approach that can improve instantaneous object identification and classification using depth information of the anomalies.

In this research, I explore the development of a framework for setting up an autonomous robot to perform computer vision enabling anomaly detection and anomaly classification to identify surface discoloration anomalies such as rust patches and industrial FODs (Foreign Object Debris) in confined spaces within large marine vessels. Our novel technique specifically focuses on object detection and classification in confined spaces where computer vision

can be hampered by unknown structural geometry, minimal or absent illumination, irregularity of structure, unpredictability of location and orientation of anomalies and obstruction of field of view. This study aims to establish the effect of sensors, on the ability of the system and deep learning model, to perform anomaly detection.

The study proposes the use of custom modified ground robots equipped with vision sensors to collect images of sections inside confined spaces containing anomalies. This procedure involves setting up of a prototype confined space environment for the purpose of data collection and using this environment to perform experiments and obtain images of anomalies. The pipeline utilizes these images to train a deep learning model to learn the identification and classification of these images. The same model is then deployed on a set of new images and used for validating the identification and classification of anomalies as has been learnt by the model. The deliverable product of this research is the development, trial and execution of a robust framework that can be utilized for performing anomaly detection in confined spaces.

Chapter 2
LITERATURE REVIEW

This research focuses on anomaly detection in confined spaces. The following sections study various approaches that have been used in the past to address the same problem statement. In recent years, the field of computer vision has significantly evolved with cutting edge vision sensors relaying high resolution color and depth information. New sensors using active stereo promise to offer improved depth-sensing results and different characteristics for robotics applications [11]. Section 2.1 dives into the specifics of generating point clouds of the entire scope of the confined spaces in question and then using deep learning models to identify and classify the anomalies. in section 2.2 we can take a closer look at the various techniques that utilize depth images, containing color and depth information, of scenarios to perform object detection.

2.1 Point Cloud Based Anomaly Detection

Point cloud based anomaly detection is a common approach in many object detection tasks as proposed in [22] where a confined space anomaly detection and classification technique is put forth, that uses offline pre-processing and learning covariance of comparing the discrepancy of each point in the point cloud to a known nominal map of the environment. Although a nominal map is either created using the mesh of the CAD model or through post processing of the collected sample point clouds. This technique prevents misclassification due to noise. Efficient three dimensional object detection can be implemented by adding a multi-scale feature fusion network to improve the algorithm's capability to detect varying-size objects [18].

A new Markov chain Monte Carlo method is proposed in [23], for discrete, global planning

through ergodic traversal of a directed graph that represents the regions that need to be inspected. This technique utilizes point clouds of confined spaces obtained by the locomotion of a ground robot in a confined space with the use of a technique called SLAM. SLAM stands for Simultaneous Localization And Mapping and it helps us build a map and localize the robot vehicle in that map at the same time. This work proposes a hierarchical, dual estimation-planning framework with an estimation module that will include the proposed anomaly detector at the top level, and adapt SLAM solutions, from various SLAM implementation options, for the lower level pose optimization and robot localization tasks. This framework targets the identification and classification of anomalies alongside the pose estimation of the robot. The scope is larger in this case as the objective is to establish a robot vision pipeline to enable the ground robot to navigate a confined space, travel within the regions of the confined space to locate, observe, identify and classify the anomalies. I document the use of an alternative approach that inculcates the use of RGB-D images of sections within confined spaces to establish a framework for anomaly detection and classification in confined spaces. Our research studies an approach that can substitute the technique of anomaly identification and classification in the broader spectrum of the work in [23].

2.2 RGB-D Image Based Object Detection

RGB-D images, as the name suggests, comprise of color information as well as depth information. These images provide a way to obtain 3D information from a scene but in individual instances as compared to the whole space at a time. We can study related work that has constructed a light-weight object detector that inputs a depth and a color image from a stereo camera. In [19], by modifying and extending the network architecture of YOLOv3 to 3D, it is observed to be possible to output in the depth direction. [19] also establishes a concept of 3D IoU, which is Intersection over Union (IoU) in 3D space, to confirm the accuracy of the results. The proposed model is able to output 3D bounding boxes and detect partially hidden objects at improved detection speeds as compared to point cloud based detection models.

A new deep neural network system is proposed in [21] called Yolo+FPN, which fuses both 2D and 3D object detection algorithms to achieve better real-time object detection results and faster inference speed, and which can be used on real robots. [21] focuses on finding an optimized fusion strategy to efficiently combine 3D object detection with 2D detection and to see its effects on environments that are challenging for both indoor and outdoor robots. [21] uses an RGB-D camera Intel RealSense D435 to realize real-time high-accurate 2D and 3D detection. The hardware GPU acceleration module of Nvidia Jetson TX2 Development Kit is used. This was tested on a ground robot and can easily be moved into a drone or any other robotic platform. In this thesis I have used an Intel RealSense D435i camera as one of the vision sensors and used an Nvidia Jetson AGX Orin as the processor for the robotic framework. This allows us to draw inspiration, comparison and inference from [21] in contrast to the results obtained in this research.

The literature and works studied in this section document the various approaches that can be used to address an anomaly detection problem, a problem of object detection in confined spaces, a problem of object detection using depth information and a problem of using various deep learning models for object detection. The works inspire us to consider even nuanced approaches focusing on one or more avenues of the issues, solutions and techniques posed and addressed by these works.

Most recently, research and updates in deep learning models such as YOLO equip us with the ability to test our approach on faster, more efficient and more robust models. In the course of the past several years, the architecture of deep learning model YOLO has developed immensely with many new versions launching sequentially, each having unique features and advantages. This research analyzes the use of various YOLO models for object detection and chooses the optimal YOLOv8 model for performing anomaly detection and classification.

Chapter 3

METHODOLOGY

3.1 Hardware Integration

Autonomous ground robots are commonly found in warehouses, agricultural applications, automobile and manufacturing industries and military applications [17, 3, 7]. Considering the specific case of using ground robots for anomaly detection and classification inside confined spaces located inside marine vessels, we consider a ground robot capable of entering and navigating closed tight spaces. In order to make the pipeline more robust, we have selected an off-the-shelf ground robot, the 'TurtleBot3 Waffle Pi', and customized it to best fit the requirements of this research project.

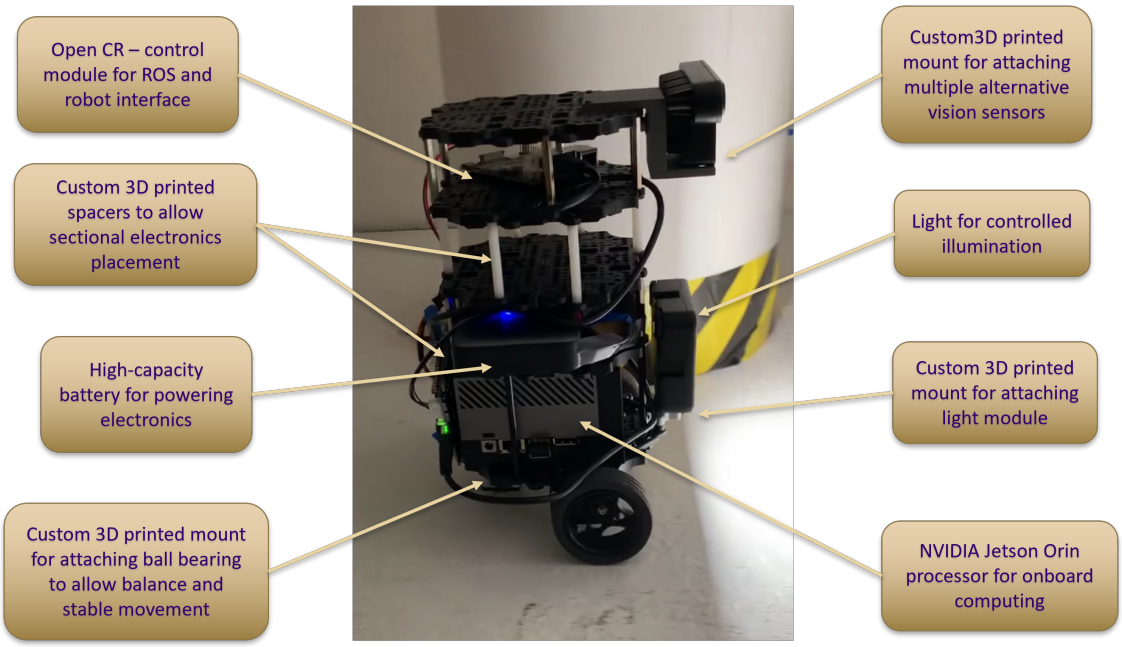

Figure 3.1: Modified TurtleBot3 used as a ground robot

Assembling various electronic and structural components was a crucial stage in developing this ground robot. The TurtleBot3 comes equipped with a set of electronic components and hardware specification. The notables ones are the OpenCR control board, a Raspberry Pi processor, 3 platform architecture, lithium polymer 11.1V 1800mAh / 19.98Wh 5C battery and a Raspberry Pi Camera Module v2.1. The components each have specific dimensions that the off-the-shelf robot is designed with. For the purposes of this research, we have modified the structure of certain components, as elaborated on in this section.

From Fig. 3.1 we can observe the architecture and assembly of the modified ground robot. The waffle configuration, that the robot is manufactured in, includes 3 wide platforms for mounting various components. This study addresses robot vision as applied to confined spaces. Some of these environments include very narrow spaces and small entryways. The same observations can be made when studying the prototype tank at a later stage in this report. In order to make the robot most compatible with navigating narrow spaces, we have modified the layers in the originally wide configuration. The width and breadth of each platform has been halved to create a modified platform 0.25 times the size of the original. In order to still accommodate for all the electronics that need to be mounted on the robot, the number of platforms used have been increased. The customized robot has 4 platforms, as compared to 3 platforms on the original robot, each having approximately a quarter of the area of the original platforms. The height offset between platforms has been modified by using custom 3D printed standoffs. From Fig. 3.1 we can observe the white standoffs of height 75mm and black standoffs of height 80mm as compared to the default TurtleBot3 standoffs of height 45mm.

The OpenCR has been retained as the control board used to interface the various electronic devices. The 2 motors and 2 wheels attached to the motors have been retained. The motor and wheel position has been modified to mount on the underside of the lower-most platform. This is a significant change from the motors originally mounting on top of the lower-most platform. This modification is done to provide more ground clearance to the robot. This will allow the robot to travel over uneven surfaces and inclined surfaces with a

higher degree of success than the original structure accounts for.

The TurtleBot3 comes equipped with a ball bearing as a third point of contact with the ground aside from the two wheels discussed earlier. After modifying the TurtleBot3 to mount the wheels on the underside of the lower-most platform, the ball bearing does not come in contact with the ground. I have designed a custom 3D mount for the ball bearing to level the robot. The lithium polymer battery, rated 11.1v 1800mAh / 19.98Wh, provided with the TurtleBot3 has been substituted by a battery pack rated 25000 mAh / 145 Wh to power the modified set of electronics and to provide multi point discharge and long duration charge sustainability.

Figure 3.2: Custom 3D mount for attaching vision sensors

The Raspberry Pi Camera Module v2.1 that has been provided, contains a Sony IMX219 8-megapixel sensor. This has been replaced by more advanced vision sensors that will be elaborated upon further in this report. We have selected two vision sensors or cameras to use for the experiments. The modified ground robot is fixed with a custom 3D printed mount for attaching either of the two cameras as can be observed from Fig. 3.2

A 2000mAh led video light has been attached to the front of the ground robot, to provide controlled illumination inside the dark confined space with no natural light. One of the key factors that we wish to consider is the provision of a controlled light setting throughout the experiments. The light is mounted on a custom 3D printed mount that is attached to the lower-most platform of the robot. The Raspberry Pi processor has been substituted by the Nvidia Jetson AGX Orin Developer Kit in order to obtain enhanced compute capabilities.

One of the key focus points for this research is observing the effect of vision sensors on the ability of the robot and deep learning model to perform anomaly detection. Another principal factor is the approach of using RGB-D images containing color and depth information. This generated a need for commodity hardware. Subsequently, the selection of cameras as potential vision sensors was a crucial step warranting a closer look at the specific hardware selections and metrics that were considered for the vision sensors.

RGB-D sensors have become an essential component in many applications and methods that use machine vision, especially in the field of robotics. RGB-D sensors are used in robotics applications such as three-dimensional simultaneous localization and mapping (SLAM) and navigation, reconstruction, object recognition and tracking, human recognition and following, and hand gesture analysis [11]. Based on a study of prominent cameras used for object detection tasks for applications similar to or closely aligning to the task of confined space anomaly detection, we have selected the OAK-D and the Intel RealSense D435i cameras.

OAK-D stands for 'OpenCV AI Kit with Depth'. It is purely stereo and has a built-in object detection module with an RGB frame resolution of 12MP (4056x3040). This camera can distinguish objects and humans, real-time, based on point cloud information. The camera module consists of two monochrome cameras on the side (stereo cameras) one RGB camera in the center which enables it to generate high quality depth images.

One significant factor for choosing OAK-D is that OAK-D's own native object detector works well. OAK-D also works very well underwater For the application of ship hull inspection, the ground robot may encounter damp, wet or waterlogged confined spaces. Additionally, as a part of the future scope of this research, the framework can be integrated

with a system developed for identifying anomalies on marine surfaces underwater. Hence, in terms of inspection teams that have underwater applications and would encompass scanning and imaging in damp or wet conditions, the OAK-D camera can prove to be advantageous.

The Intel RealSense cameras are very popular for robot vision applications. The Intel RealSense D435i contains the Intel RealSense Module D430 + RGB Camera. The RGB frame resolution is 1920 × 1080 pixels. They provide depth images as 3D output and have both a conventional sensor and secondary infrared sensors, for precise depth of field. These robust system properties, combined with advanced configuration and calibration tools and the presence of multiple models, opens up new avenues in a wide range of applications such as 3D detection, facial recognition, object recognition, and so on. The Intel RealSense cameras use the open-source SDK 2.0, which is compatible with several development platforms, programming languages and operating systems.

OAK-D	**RealSense**
Stereo	Stereo, LiDAR, and infrared (for noise correction)
RGB resolution: 4056x3040 pixels	RGB resolution 1920x1080 pixels
Depth field of View: 81° × 55°	Depth field of view: 87° × 58°
Intended use with PyTorch	Only compatible with TensorFlow
Positive depth residual error	Negative depth residual error
Simple and adaptable API	More challenging to adapt code

Table 3.1: OAK-D vs. RealSense: comparison of key features

The OAK-D and RealSense have certain advantages and disadvantages over each other. One of the objectives of this research is to identify and document the effect of these qualities on the performance of the anomaly detection model. Table. 3.1 documents the numerical and architectural specification differences between the two cameras.

Fig. 3.3 illustrates the view of the OAK-D and the RealSense depth camera along with a sample color and depth image taken from each camera in a scene inside a prototype confined

space.

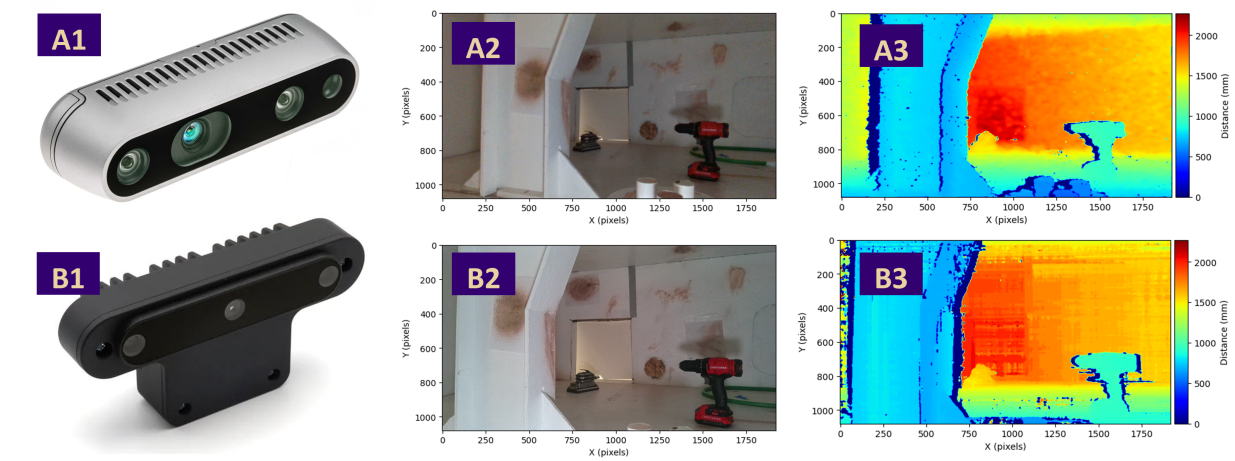

Figure 3.3: (A1) Intel RealSense D435i camera, (A2) Color image (RealSense), (A3) Depth image (RealSense), (B1) OAK-D camera, (B2) Color image (OAK-D) and (B3) Depth image (OAK-D)

3.2 Prototype Environment Setup

In an endeavor to study confined spaces inside marine vessels and to understand the kind of anomalies there, a prototype scaled down recreation of a confine marine space has been constructed in the SMARTS Lab at the University of Washington.

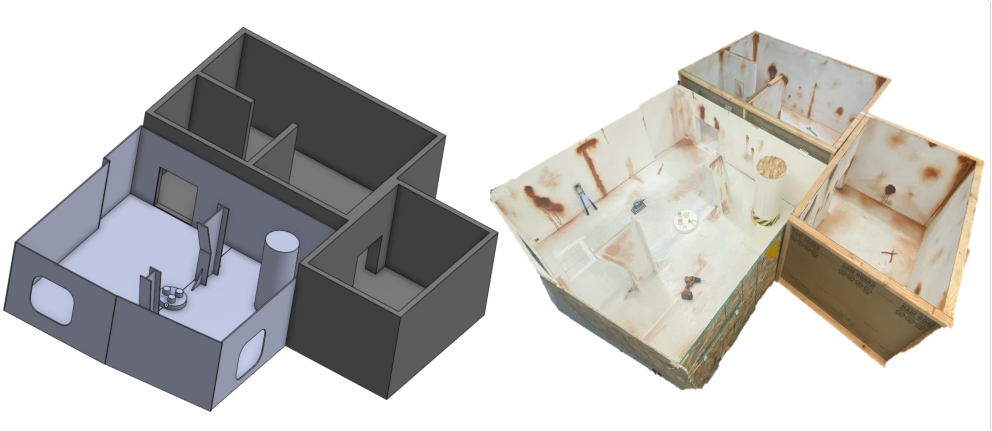

Figure 3.4: Graphical representation (CAD model) of prototype tank, (Left) and Bird-eye-view of actual prototype tank model (Right) used for data collection.

Fig. 3.4 illustrates the graphical representation of the CAD model and the pictorial representation of the actual prototype tank that has been constructed and used for performing the experiments related to this research project. In the figure we can observe the simulation of anomalies such as rust and FODs (industrial tools) that can be found inside confined spaces in ship hulls and ballast tanks inside marine environments and marine vessels. We can also observe the presence of I-beams, pillars, pipes and other structural components that can also be found inside confined environments. We can further focus on the presence of rust and tools adjacent to each other, or, on certain occasions, overlapping each other. In this prototype environment, rust is simulated using a combination of acrylic, oil spray and water paints of various colors and shades. The tools are obtained for the purpose of the experiments in this research study. This carefully designed custom environment has been created to obtain RGB-D images in scenarios where multiple types of anomalies, each of

varying classes, are found to coexist or appear together. There are also additional walls in the prototype tank equipped with small entryways and narrow gaps between them. These tight pathways enforce the principal of confined spaces and the walls create shadows as well as obstacles. Identifying anomalies alongside these factors proves to be a very challenging case and that is the niche situation that this research focuses on.

3.3 Classification of Anomalies

3.3.1 Foreign Object Debris

FOD stands for Foreign Object Debris, and this is defined as any object that is not native to the environment that is being considered. For the purpose of this research study, we are considering confined spaces in ship vessels which are often undergoing repair and maintenance operations. During these operations, industrial tools get left behind in the confined spaces. Hence these are established as the FODs for the length of this study.

Figure 3.5: Classification of Foreign Object Debris (FODs)

These objects can be observed or found in various poses in the confined space environ-

ment. We have considered a few options of poses in which these FODs could be found left behind or unattended, in the confined spaces. At a later stage we have spoken about using data augmentation. This technique helps perform modifications on the poses of the anomalies to create more poses from various views and angles. Fig. 3.5 displays some images from our FOD dataset appearing in various poses in the confined space. We have considered 6 classes, namely hammer, wrench, drill, screwdriver, sander and crimper.

3.3.2 Rust

We can take a close took at the rust patches that we have established as anomalies for the purpose of the project.

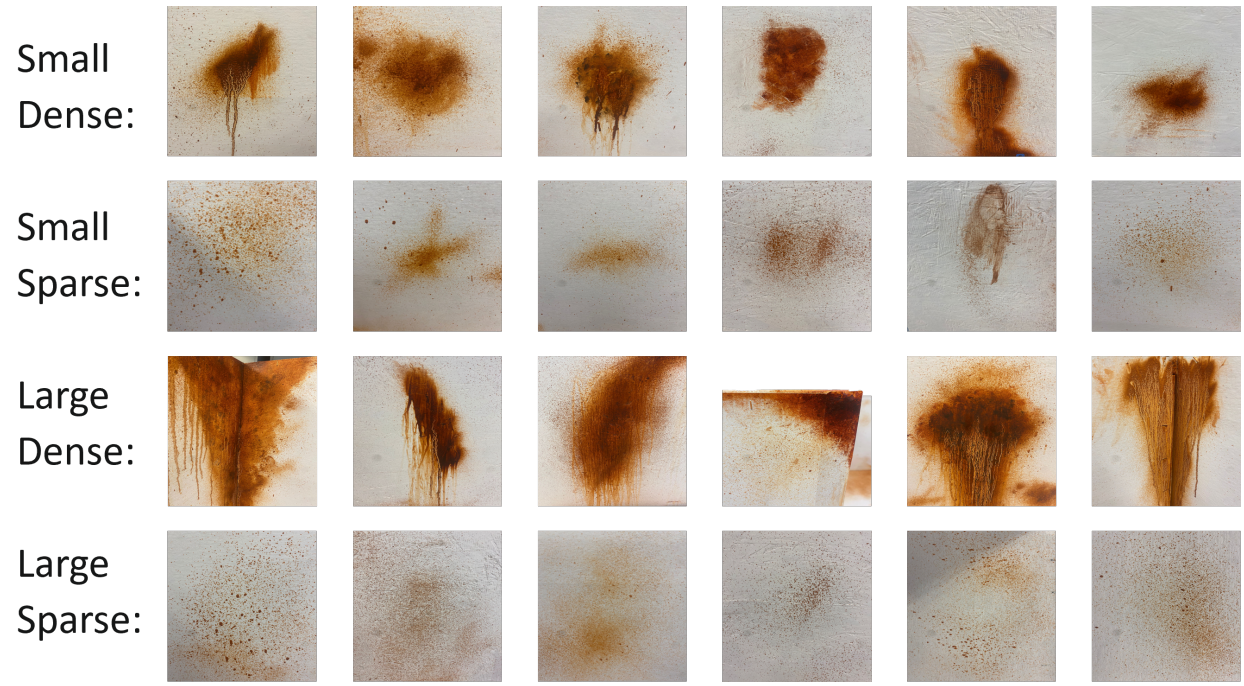

Figure 3.6: Classification of Rust Anomalies

For the purposes of this research, rust has been sub-classified based on size and density of the rust patch. All rust patches smaller than 25 cms x 25 cms will be labeled as small and all rust patches larger than 25 cms x 25 cms will be labeled as large. This distinction holds true

even if one of the dimensions goes outside the 25 cm limit. The density gradient is another sub-classification for the rust patches, further creating a dense and a sparse categorization. Fig. 3.6 demonstrates some sample rust patches as can be observed from the prototype tank. The final rust classes are 'Small Dense', 'Small Sparse', 'Large Dense' and 'Large Sparse' as displayed in the Fig. 3.6

3.4 Design of Experiments

Given an unknown confined space with anomalies and foreign object debris, we would like to execute an algorithm that detects images containing anomalies based on pre-trained information. This design of experiments section is to define a technique to collect the images to be used for pre-training as well as programming such an algorithm to perform image-based anomaly detection and classification. In this study we try to understand the relationship between various factors and variables in our system.

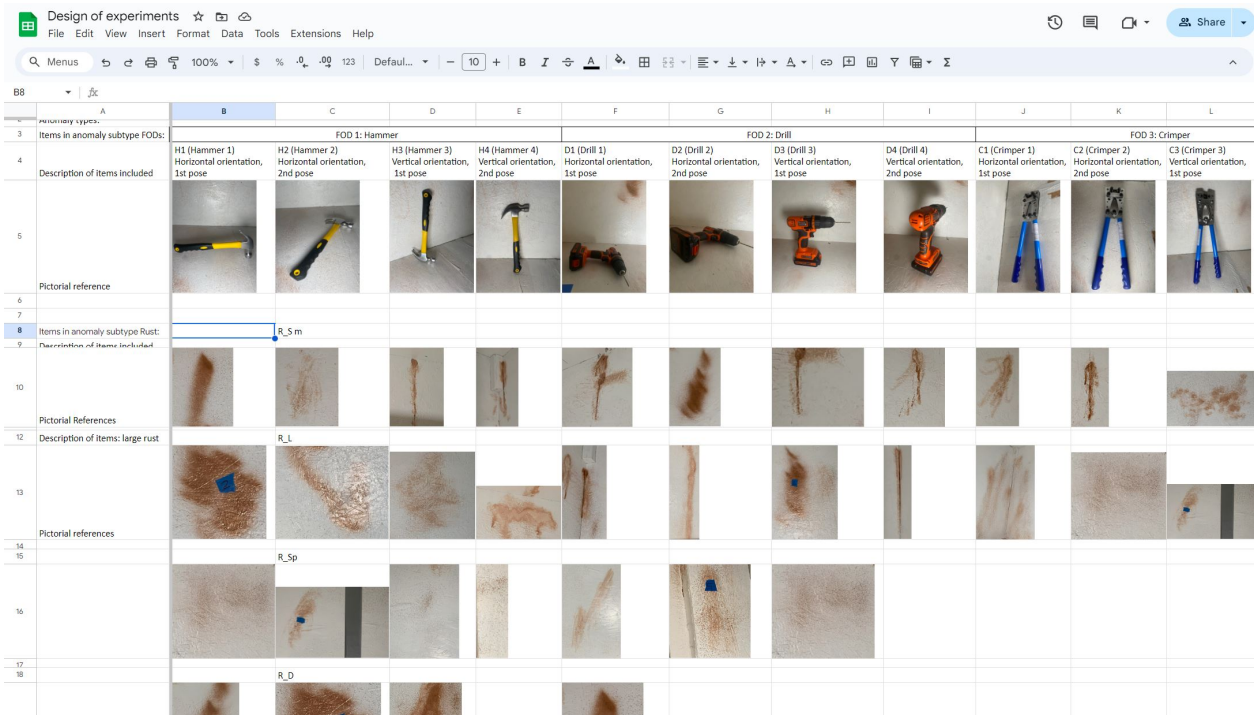

Figure 3.7: Documenting all the FOD and rust types and classes

The goal is detection and classification of FODs and rust while analyzing if the quality of detection depends on where the FODs and rust are, or the occurance of FODs and rust in the same scene, the relative size between the FODs and rust, the relative size between the anomalies and the structures in the prototype tank, the angle and brightness with which an anomaly is visible, and so on. Given these important questions and two different sensing platforms, the study aims to see how sensitive the anomaly detection performance is based on the variables.

Some of the constants that can be observed during the design of experiments are the specifications of the ground robot, the camera location and mounting with respect to the robot, the focal length of the camera lens, which is constant and is is achieved by adjusting the intrinsic parameters of the camera during the camera calibration phase. The architectural layout, dimensions and structure of the confined space remain constant throughout the experimental stage. Based on this, we can document the instances and occurrences of FODs and anomalies as they are observed inside a tank to create a design of experiments document as displayed in Fig 3.7.

Now let us consider some variables that will affect our experiments. We can talk about brightness. The prototype tank is a dark environment with the robot light being the only light source. At this stage, we have set the light to a fixed amount of brightness, saturation and hue and kept these values constant throughout the experiments. The elevation of the tank floor is a small variable term with surface roughness and unevenness observed on the floor of the prototype tank, especially when transitioning from one section of the confined space to another. To avoid bias that can be brought around by a small set of images taken at certain brightness and elevation specifications, we have decided to use a 4 pose technique for capturing depth images of each anomaly. Fig. 3.8 illustrates how the ground robot will be placed at 4 locations with respect to the anomaly. Thus setting up a standardized experiment for data collection comprising of 4 poses per anomaly: + 45° and 100 cms, - 45° and 100 cms, + 30° and 125 cms, and - 30° and 125 cms. This gives us images of the anomalies and the background, from various angles.

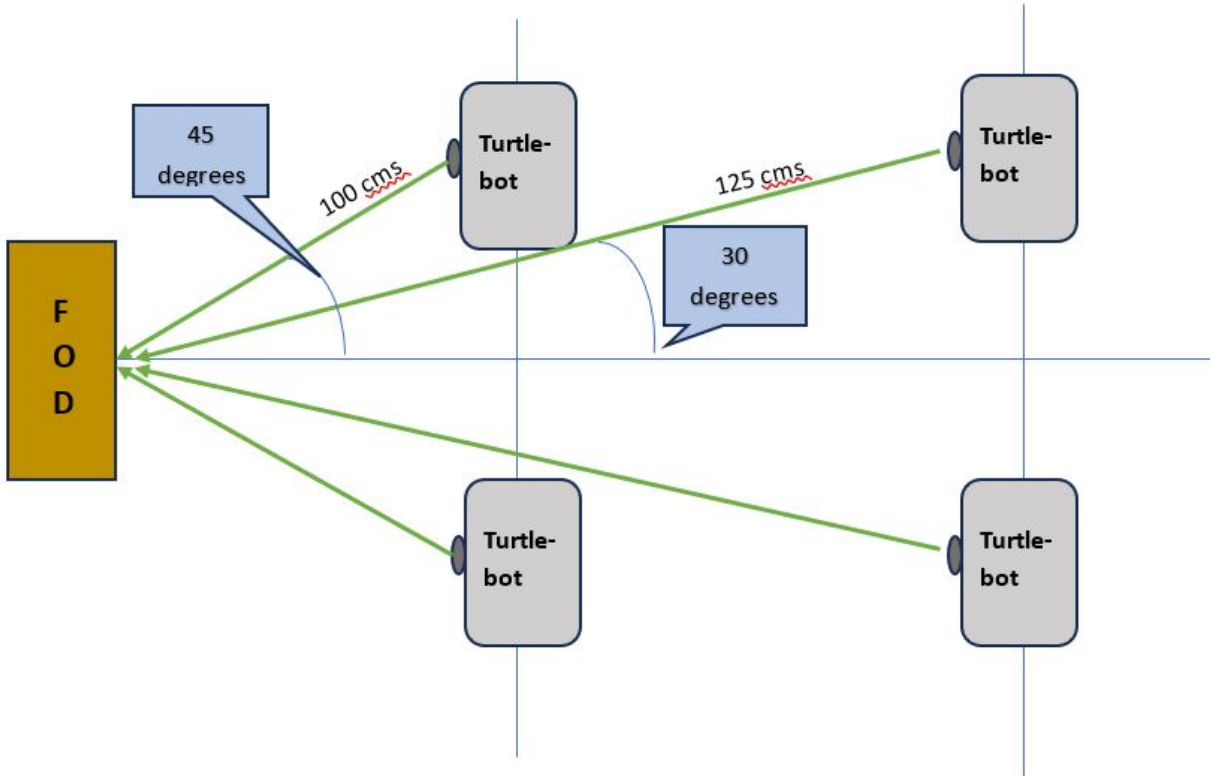

Figure 3.8: Establishing the 4 poses for collecting images of each anomaly

Considering the dependable variables of the system, these are affected by the dynamics caused due to a change in the location of the robot and camera. The pose of the camera with respect to the prototype structure will change with every movement. Similarly the pose of the camera with respect to the anomalies will change. To encompass these variables across the space of the prototype tank, we have established a scenario comprising of 10 anomalies as seen in Fig. 3.9. Each anomaly is located in a different section in the prototype tank, but most are often in the vicinity of other anomalies that are not in focus during the current scenario. The design of experiments is set such that the ground robot will travel to each anomaly and execute the 4 poses as explained in Fig. 3.8. While capturing images at these 4 poses per anomaly, the robot will capture a large amount of images of the anomalies and structure surrounding the anomalies in focus. One sample scenario demonstrating the use

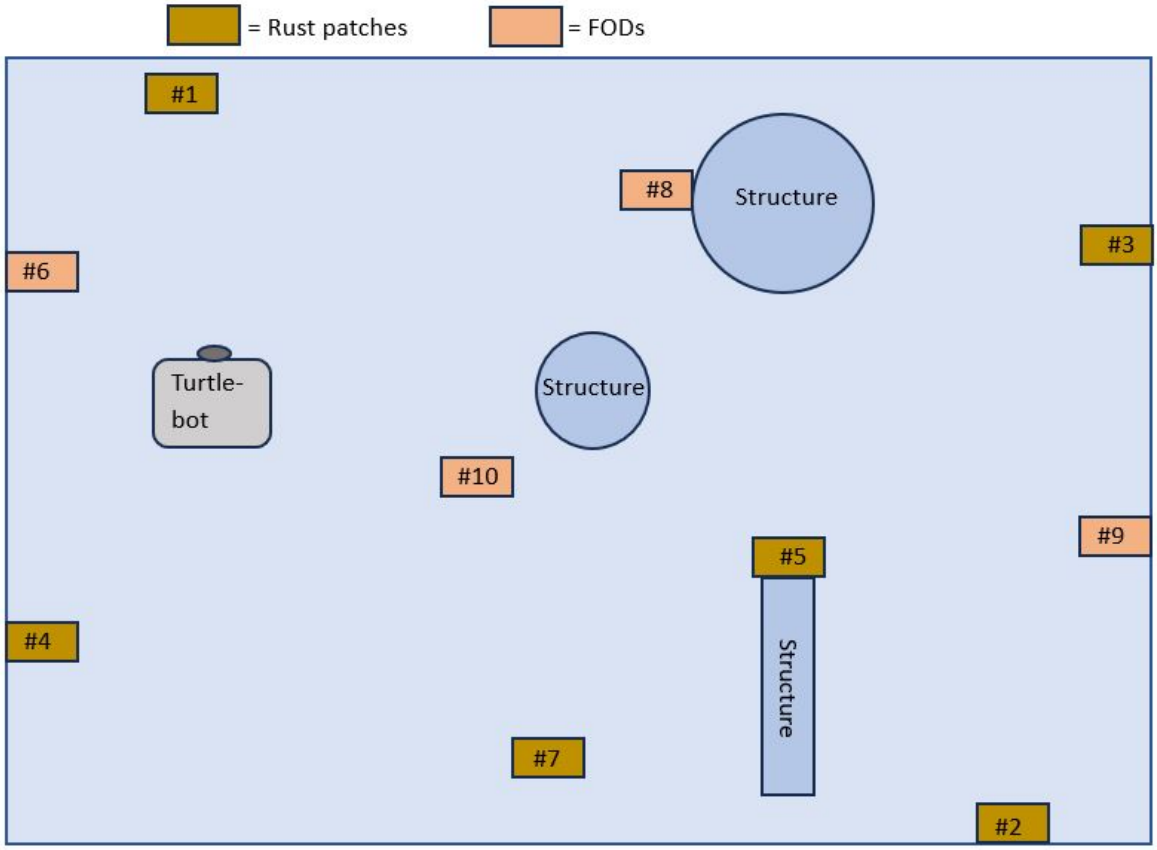

Figure 3.9: Demonstrating one sample scenario containing 10 anomalies

of 10 anomalies is displayed in Fig. 3.9.

We can replicate the procedure explained using Fig. 3.7, Fig. 3.8,and Fig. 3.9 to create multiple scenarios each with 10 anomalies. The 10 anomalies can comprise of 6 or 7 rust patches and 4 or 3 FODs each. Each anomaly selected is assigned the pose which it will appear in, in the scenario. Fig. 3.10 displays a sample of 3 such scenarios that are used for conducting the experiments.

Addressing cracks and debris: This research does not consider cracks or debris found in the structure as anomalies. These objects would, most likely, be the same color as the structure they will not be as easy to distinguish. Ship hulls are composed of sturdy and

	Type of anomaly	Anomaly classification	Anomaly label	Pose 1	Pose 2	Pose 3	Pose 4
Scenario no 1	FOD 1	SD3	SD3	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 2	W2	W2	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 3	H1	H1	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 4	C2	C2	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 1	L_Sp	27	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 2	SM_Sp	2	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 3	L_D	1	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 4	SM_D	3	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 5	L_Sp	25	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 6	SM_Sp	12	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
Scenario no 2	FOD 1	S2	S2	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 2	C3	C3	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 3	H4	H4	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 1	L_D	15	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 2	SM_D	4	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 3	L_Sp	23	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 4	SM_Sp	11	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 5	L_D	13	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 6	SM_D	16	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 7	L_Sp	22	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
Scenario no 3	FOD 1	W2	W2	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 2	C4	C4	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 3	SD3	SD3	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	FOD 4	S2	S2	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 1	SM_Sp	9	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 2	L_D	10	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 3	SM_D	14	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 4	L_Sp	18	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 5	SM_Sp	32	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg
	Rust 6	L_D	31	100 cms / + 45 deg	100 cms / - 45 deg	125 csm + 30 deg	125 cms / - 30 deg

Figure 3.10: Displaying 3 sample scenarios with various specific FODs and rust patches selected

tough structures, where cracks or debris are very rare and so focusing on these objects as anomalies would prove to be peripheral and unnecessary. In the future scope of this research, we can use some images containing cracks or debris as a part of the test dataset and observe how the anomaly detection model reacts to these rare anomalies.

It is observed that the time taken to capture the images of one anomaly with one pose is one minute. Considering four poses per anomaly, ten anomalies per scenario, twenty scenarios in total and factoring in two cameras, indicating that each image will be taken four times, once with each camera with AprilTags and without AprilTags, it is estimated to take around 4800 minutes to perform all the experiments needed. Using this Design of Experiments, we have now established a pipeline for performing the experiments to collect RGB-D images from the prototype tank using both the OAK-D and the RealSense camera. A sample scenario of the robot inspecting the confined space prototype tank and collecting

Figure 3.11: Image of robot inspecting a tool in a sample scenario

images can be found in Fig. 3.11.

3.5 Data Acquisition

The data acquisition phase of the research involves executing the design of experiments and running the scenarios as they have been developed. The ground robot with modified hardware and electronic components, custom camera mounts and specifically selected vision sensors, the prototype tank that have been designed and constructed for the experiments and the anomalies that have been created inside the prototype tank, all come together with the help of the design of experiments and lead to the data collection phase, executed by the ground robot, which is being teleoperated with the help of ROS.

It is important to note that during the data collection phase, we have also inculcated ground truth collection into the pipeline as can be observed from Fig. 3.12. This includes the

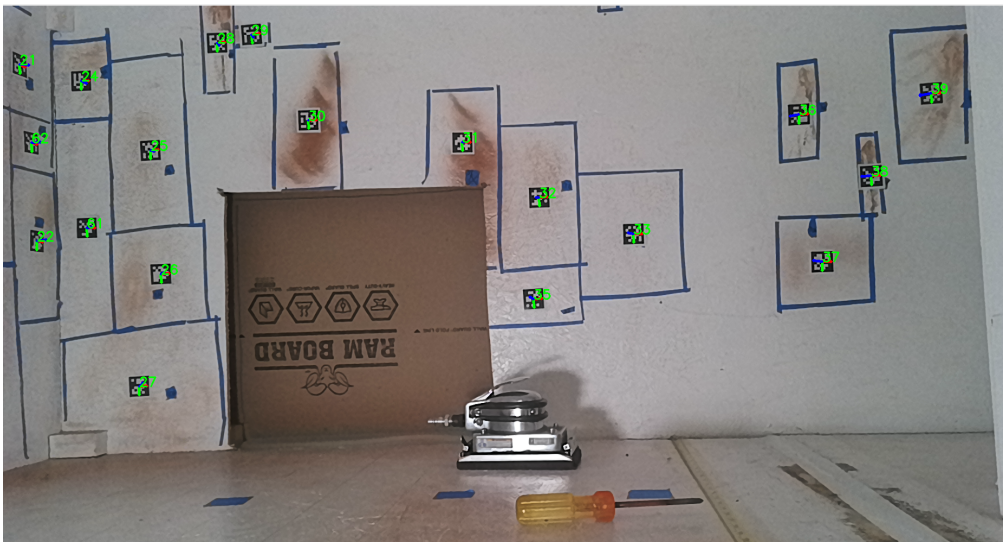

Figure 3.12: Image of robot inspecting a tool in a sample scenario

use of AprilTags, fiducial distance and pose markers, to obtain the ground truth information. The AprilTags are placed at the center of the predicted bounding box or predicted area of the anomaly. RGB-D images are now taken with both cameras with the AprilTag. This will give us the exact pose and distance of the camera with respect to the anomaly's center. Images are then captured without the AprilTag as described in the previous section.

Fig. 3.13 demonstrates a pictorial representation of the ground robot inside the prototype confined space tank, navigating columns and structural components to look at rust and a FOD present in the scenario that it is executing.

The experiments are performed sequentially. First each scenario is set up with the rust anomalies (permanently located in the prototype tank at fixed locations) and the FODs (located at arbitrary locations with pre-determined poses) as defined by the design of experiments document. The ground robot captures images focusing on one anomaly at a time and executing the experiments by obtaining images at four poses per anomaly. The images will capture many rust anomalies and FODs in the background, beyond the anomaly in focus.

Figure 3.13: Ground robot demonstrating an experiment of collecting images of anomalies from the prototype confined space

3.6 Deep Learning based Object Detection Model

Visual recognition systems that are at the core of deep learning applications, which encompass image classification, localization and detection. These systems have achieved great research momentum. Due to significant development in neural networks especially deep learning, these visual recognition systems have attained remarkable performance [15].

One example of a state-of-the-art deep learning object recognition system is YOLO. YOLO stands for 'You Only Look Once' and this is a real-time object detection algorithm that processes images very fast. It is one of the most popular algorithms in the machine learning world. It has many versions and is known for several features as noted below.

Rationale behind selecting YOLO as the deep learning model to perform anomaly detection and classification:

- YOLO is robust and easy to customize to suit our specific needs.
- The YOLO community is very active, resulting in sequential new versions, modifica-

tions and updates as compared to some strong models like Faster RCNN which have a disadvantage as they have not been updated in a long time.

- YOLO affords us the freedom to run multiple models in parallel and integrate the pipelines at various stages. This allows us to run FOD detection and rust detection as separate models with possible avenues of fusing the results.
- The YOLO architecture can be modified for implementing specific hand-tuned weights.
- In YOLO it is very easy to execute transfer learning to integrate our custom dataset into the pre-trained model.

At the time of this report presentation, a newer version of YOLO called YOLOv9 had been launched. After doing a comprehensive comparison, it was concluded that YOLOv9 has improved real-time object detection for continuous processing of videos, but YOLOv8 still performs better on still images. Due to this reasoning YOLOv8 has been selected and used throughout this study.

3.7 Pipeline

Fig. 3.14 demonstrates the complete framework depicting RGBD image processing via the YOLOv8 neural network model. This pipeline involves creating a custom dataset of RGB images and Depth images for each anomaly (FODs and rust), setup in various scenarios. We then label the images obtained, pre-process them, perform data augmentation and then use a partially pre-trained off-the-shelf YOLOv8 machine learning model for detection and classification. The model is combined with transfer learning and fine tuning from the custom dataset of images collected in this study.

The pipeline explains how we collect the custom dataset for training and validation and the pipeline is setup for validation or testing new images of anomalies in confined spaces. It is important to note that we have executed this pipeline separately for rust and for FODs.

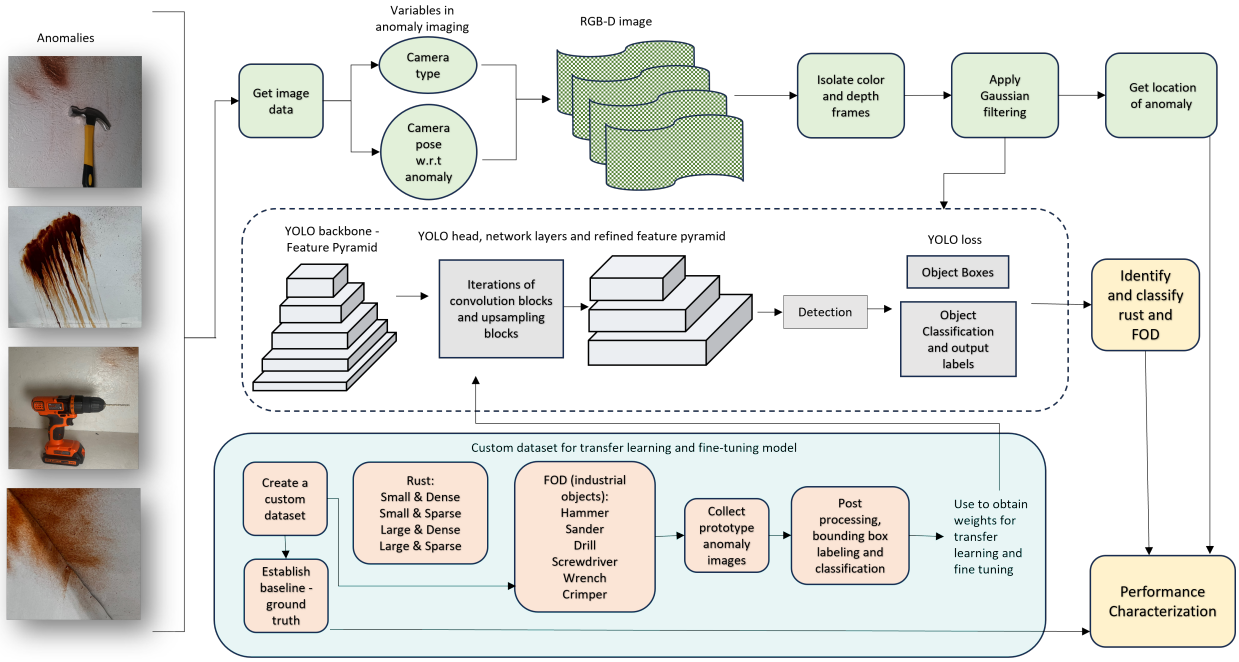

Figure 3.14: Figure displaying the framework setup for developing and testing a system for anomaly detection in confined spaces.

Chapter 4
EXPERIMENTAL RESULTS

The next step after collecting the data is using the YOLOv8 model discussed above, to perform machine learning on the images that have been collected. However, data usually needs some pre-processing and some augmentation before it can be passed to a machine learning model. In this section, we take a closer look at the software programs used for collecting and processing data, the additional data processing steps and then study the results obtained from the YOLOv8 machine learning model.

4.1 Software Implementation

This study uses a combination of various software packages for developing the framework of robot vision for anomaly detection and classification in confined spaces. The Nvidia Jetson AGX Orin Development Kit has been used as the main processor to integrate all the electronic components and sensors. The processor has GPU memory of 106M, and has been installed with a python 3.8.10 environment, with CUDA version 11.8.89, ROS noetic version1.16.0 and OpenCV version 4.5.4. Each of these software packages and programs have been utilized in operating the ground robot, interfacing with the cameras and running machine learning models for anomaly identification and classification.

For the purposes of deploying the OAK-D camera, I have used the Depth AI Software Development Kit with the Depth AI python API library [9]. For deploying the RealSense camera, I have used the Intel RealSense SDk 2.0 and the Github RealSense library called librealsense [8].

Roboflow [10] is an online computer vision platform and database that allows users to upload and modify caches of data. I have used this software for creating custom datasets,

integrating various internet datasets into our custom datasets to perform transfer learning, generating and assigning detection and classification labels to the datasets and performing data pre-processing and data augmentation.

For running the machine learning model of YOLOv8, I have used Ultralytics [12]. Ultralytics is an intuitive AI platform for creating, training, and deploying machine learning models where users can choose to interface with the code or with the deep learning framework. This tool allows for hyperparameter tuning, customization of model type, size and parameters and allows for the use of advanced troubleshooting techniques for YOLOv8.

4.2 Data Pre-processing

In this research we have used 3 types of pre-processing techniques. Auto orient pre-processing, Gaussian blur pre-processing and resize pre-processing.

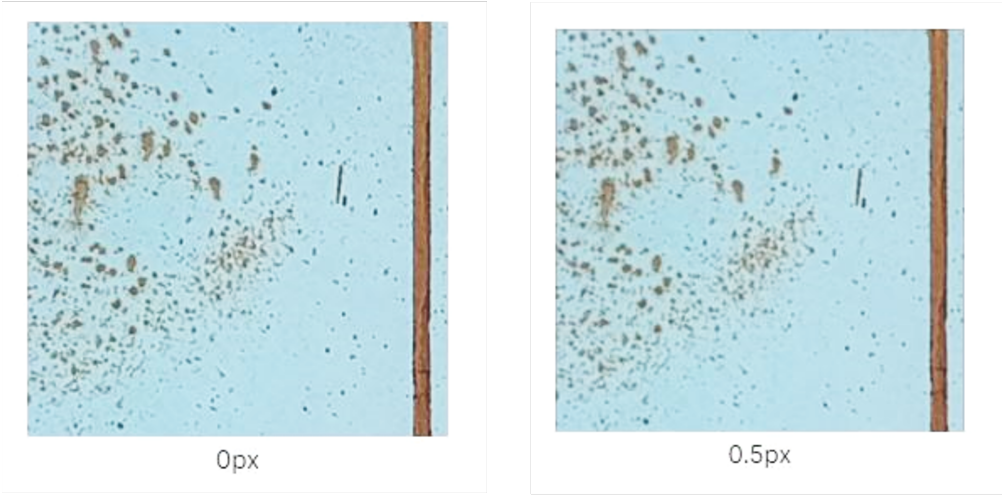

Figure 4.1: Figure displaying a sample image or a rust patch anomaly. On the left is the raw image and the figure on the right contains a Gaussian blur of 0.5 pixels added to it.

A brief description of the various pre-processing steps and their functions:

- Auto orient: This allows us to discard EXIF rotations and standardize pixel ordering

- Gaussian blurring: This option lets us add random Gaussian blur of up to 0.5 pixels to the entire dataset to help the model be more resilient to camera focus.
- Resize pre-processing: Stretches or compresses the image to 384 x 360 pixels to match the pixel size of the custom dataset that we have collected through the experiments.

Using a pre-processing step like Gaussian blurring applies the Gaussian function to a random number of pixels varying between 0 and 0.5 pixels across the dataset of images. We can observe a sample of this in Fig. 4.1. This helps combat any differences in images due to camera focus and helps make the model more immune to learning features based on focus.

4.3 Data Augmentation

Primarily dataset augmentation is a great tool for creating big datasets. In this technique, we can find any data that is available to the public and is relevant to our type and classification of data, and we can use the images to enhance the size, and hence the ability of our dataset.

Through thorough searches on the internet, I was able to find one pre-labelled dataset consisting of industrial tools. This dataset was very large and could add 3000 images to our custom dataset of 1200 images which was a great advantage, This dataset contained 5 classes, namely hammer, wrench, drill, screwdriver, and pliers. In order to match our custom dataset based on our experimental data, I have renamed the classes of crimper and the sander to a single class called 'miscellaneous tools' before processing it through the machine learning model.

Similarly, we found and curated a dataset for rust based on images from the internet. This external dataset was 200 images and our custom dataset had 520 images for each camera. However upon running the machine learning model of YOLOv8, the model performed very poorly on the internet based images and on the combined dataset of internet images and custom dataset images. In retrospect, the images from the internet are very harsh in terms of coloration, shape of rust carrying objects, background color of the walls and the nature of the water accumulation causing rust. For the prototype tank, we have created rust patches to

Figure 4.2: Sample image – raw (pre-processed), with vertical flip and with horizontal flip.

simulate the confined spaces inside marine vessels, with plain monochromatic walls, specific areas of maximum water accumulation, and water damage patterns. Hence the internet rust images may have been too varied to be able to provide any useful or distinct features and weights to the model. The point to note is that we have not used the internet rust images to obtain any of the results listed in this research. Rust detection is based on our custom dataset alone.

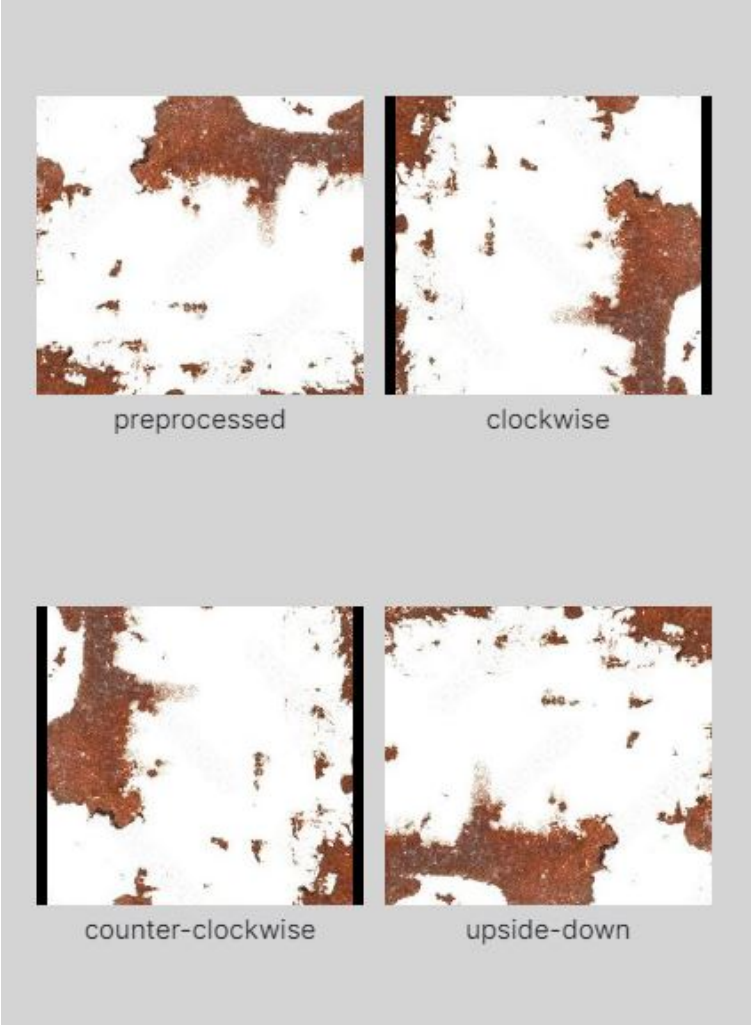

Figure 4.3: Sample image – raw (pre-processed), with clockwise (+90°) rotation, counter-clockwise (-90°) rotation and with upside down (180°) rotation.

Here is a brief overview of the data augmentation steps executed:

- Using other, public and related datasets to enhance the strength of the custom dataset.
- Adding horizontal and vertical flip helps the model be insensitive to subject orientation.
- Adding 90° rotation, clockwise, counterclockwise and upside down helps the model be insensitive to camera orientation.

- Saturation, brightness and exposure can be randomly varied by a user defined percentage. This will make the model insensitive to light and camera settings. We have not varied saturation, brightness or hue as of yet, but we plan to execute this as a future step.

Fig. 4.2 demonstrates a sample image undergoing a vertical and horizontal flip as a data augmentation step. Fig. 4.3 demonstrates the same sample image undergoing a clockwise rotation, anticlockwise rotation and upside down rotation. The data augmentation techniques help make our model more robust and improve results.

4.4 Results: FOD Detection

Before diving into the visualization of the results, it is important to note the hyperparameters used while training and validating the machine learning model:

- Model = yolov8.m
- Batch size = 16
- Learning rate = 0.01
- Total FOD images = 4130
- Total rust images = 1344
- Training : validation split = 80:20
- Cosine learning rate scheduler = True or False (both variations tried for all models)

Now let us take a quick look at a sample training batch that the YOLOv8 model processes when performing FOD detection. We can observe in Fig. 4.4 which displays a collection of images randomly assigned to the training batch of the model. In these images, we can observe the presence of some images from the internet-sourced dataset, and some images from our custom trained dataset. It is interesting to note that a large portion of the images contain

magnified views of the FODs. The model picks up the bounding boxes assigned to these complete or partial images of the FODs and learns what FOD anomalies are and how to distinguish them.

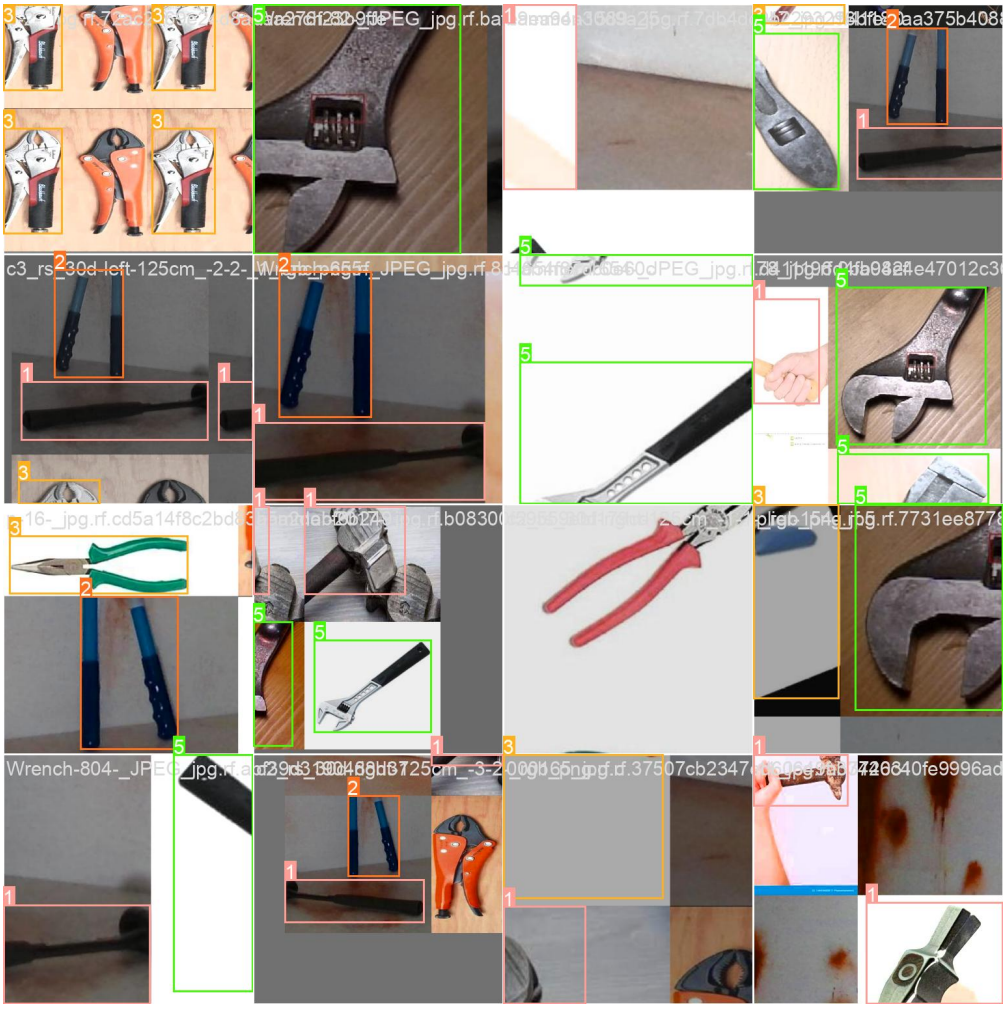

Figure 4.4: Images observed in one training batch by the YOLOv8 machine learning model white training on FOD detection and classification data.

Studying the results obtained by the YOLOv8 model when performing anomaly detection and classification on FODs, we come across the validation result plots and metrics obtained by the images from each camera. Fig. 4.5 displays the results obtained by the YOLOv8 model on the FOD dataset obtained by the OAK-D camera. We observe that the model

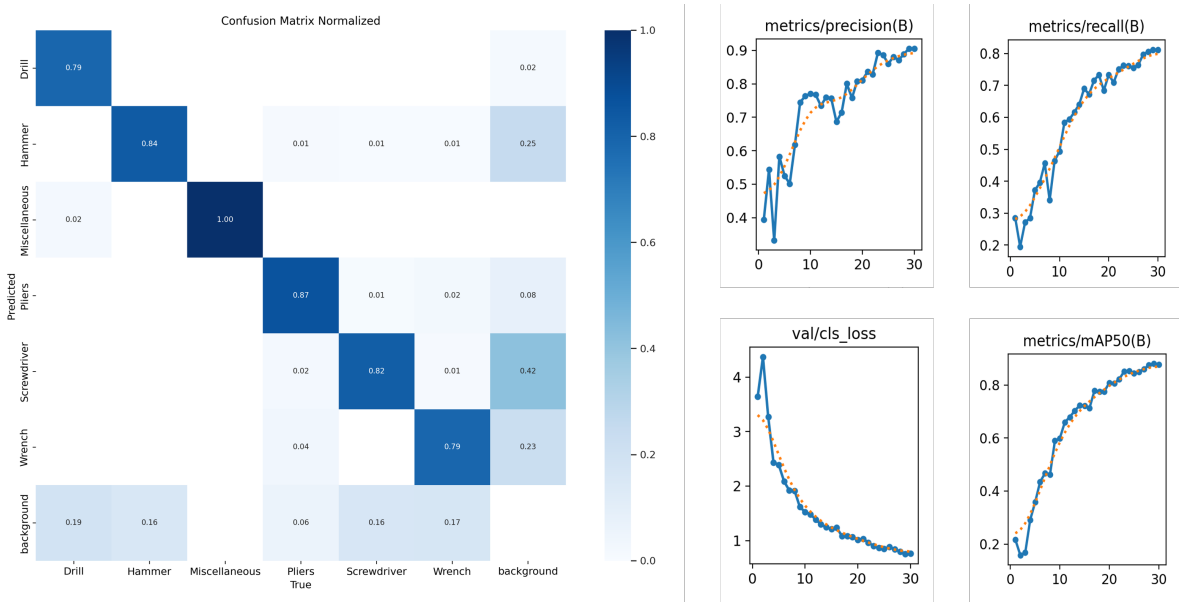

Figure 4.5: YOLOv8 results for anomaly detection and classification of FODs with OAK-D.

performs considerably well with a precision of 0.9, a recall of 0.8, a validation/class loss that settles to 0.2 and a mAP at IoU50 of 0.88.

Next we can study the effect of using a cosine scheduler for adaptively varying the learning rate. This can be interpreted from the term 'with cos lr'. Using the cosine scheduler for learning rate helps the model get out of a local minima. From Fig. 4.6 we can observe that the validation performance metrics change very marginally when we run the model for FOD images obtained by using the OAK-D camera with the cosine scheduler learning rate hyperparameter set to true. Overall, there is no observable and distinct change in the metrics. Both models perform well.

Now let us take a look at the validation performance metrics for running the YOLOv8 object detection model using FOD images obtained from the RealSense camera. Fig. 4.7 demonstrates that this model performs considerably well with a precision of 0.9, a recall of 0.8, a validation/class loss that settles to 0.1 and a mAP at IoU50 of 0.87.

Next we can look into the effect of using cosine scheduler for adaptive learning rate

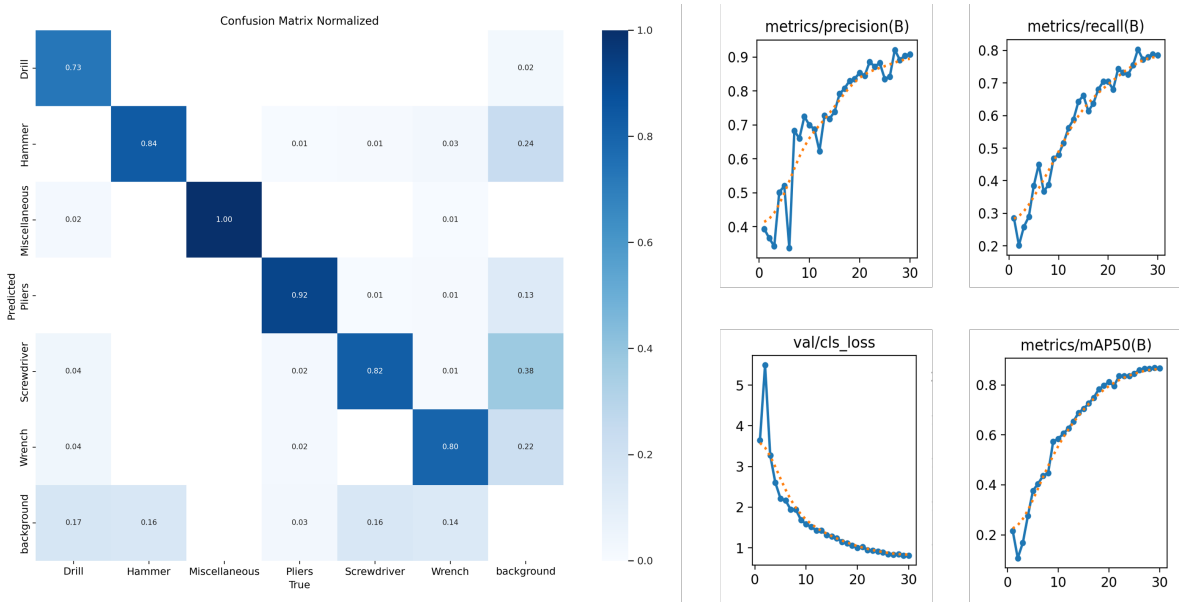

Figure 4.6: YOLOv8 results for anomaly detection and classification of FODs with OAK-D with cosine learning rate scheduler.

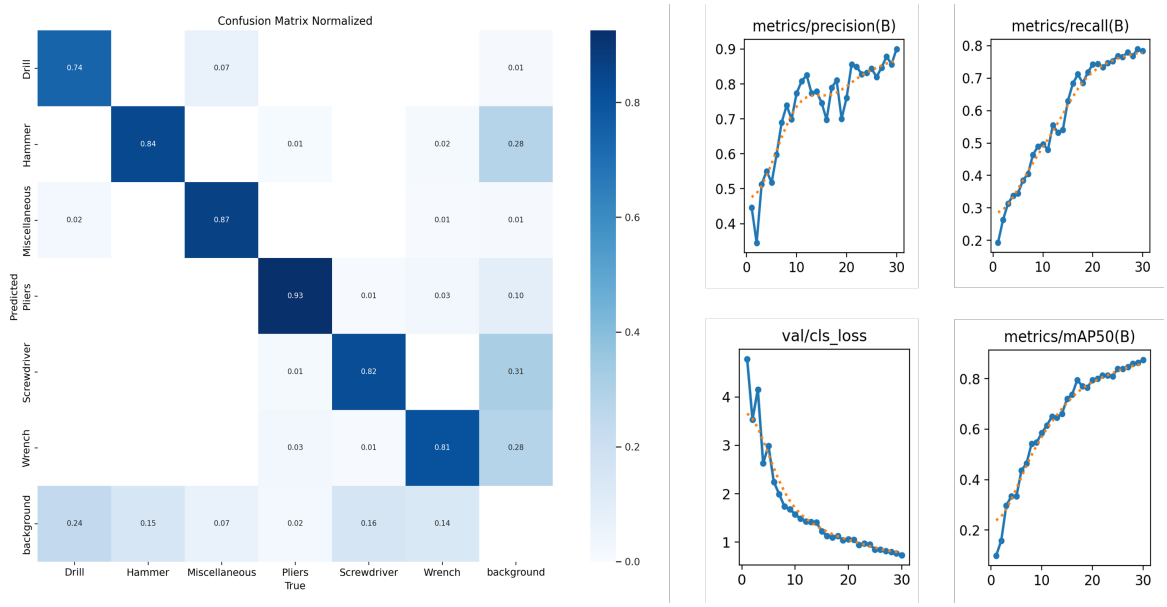

Figure 4.7: YOLOv8 results for anomaly detection and classification of FODs with RealSense.

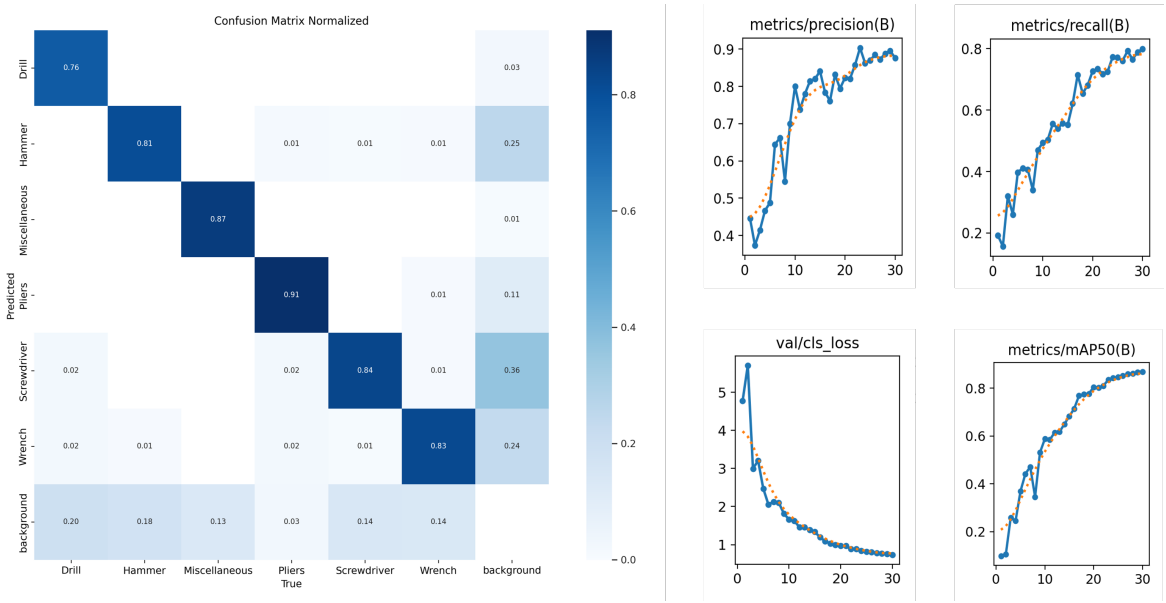

Figure 4.8: YOLOv8 results for anomaly detection and classification of FODs with RealSense with cosine learning rate scheduler.

variation with the FOD images obtained from the RealSense camera. Fig. 4.8 shows that the validation/class loss spikes (one specific outlying value that can be ignored) before settling down to 0.1, which is a very marginal change. Overall this model performs well similar to the model with RealSense without 'cos lr'. There is no observable and distinct change in the models.

Analysing the various validation results obtained from the various FOD detection models, we can draw a comparison between the OAK-D and the RealSense. The confusion matrices of the OAK-D FOD detection and the RealSense FOD detection, displayed in Fig. 4.9 show that the OAK-D is able to identify 100% of instances where miscellaneous tools (containing the sander tool and/or the crimper tool) occurred in the validation dataset. Whereas the RealSense is able to identify 87% of the instances where miscellaneous tools occurred in the validation dataset which is a significant result.

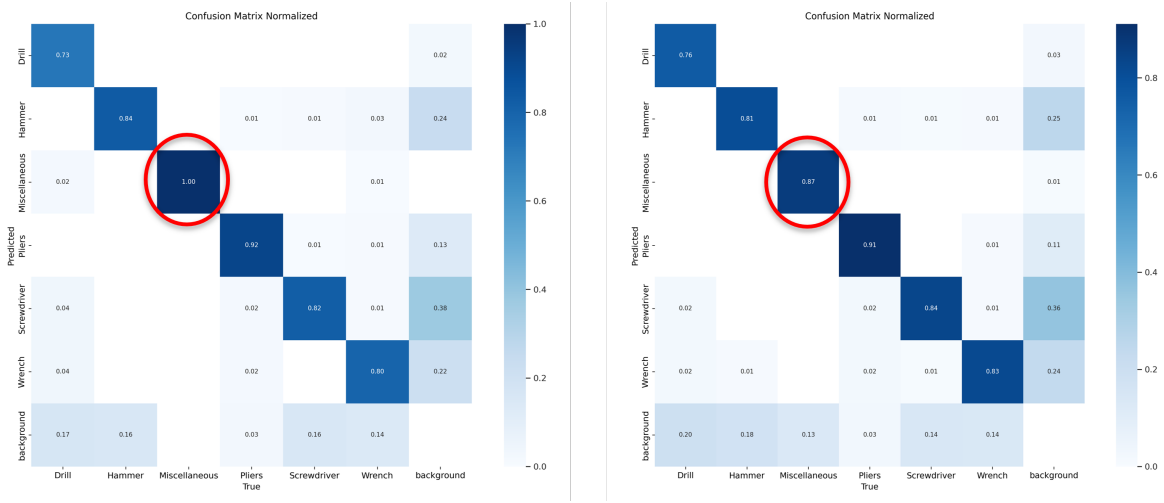

Figure 4.9: Comparison between OAK-D (Left) and RealSense (Right) of YOLOv8 results for anomaly detection and classification of FODs.

4.5 Results: Rust Detection

Now let us take a quick look at a sample validation batch that the YOLOv8 model processes when performing rust detection. Fig. 4.10 demonstrates a sample set of images randomly assigned to one of the validation batches of the model. In this figure we can see multiple images containing various sections of the walls in the prototype confined space that we have created. Some images display narrow spaces, additional walls, structural components and other anomalies present in the background of the images. Since this image is taken from a validation batch, we can observe the predicted bounding boxes on the rust patches. This indicates that the model trained on 80% of the entire dataset and learnt how to identify bounding boxes. Subsequently when the model was exposed to the validation dataset, containing 20% of the images in the entire dataset, Fig. 4.10 demonstrates the model's ability to identify rust anomalies and detect bounding boxes to encompass the anomalies. The figure also displays the confidence score which indicates the model's surety of the predicted bounding box and classification label.

Now let us take a closer look at the validation result metrics obtained when taking a

Figure 4.10: Images observed in one validation batch by the YOLOv8 machine learning model white training on rust detection and classification data.

dataset of rust images collected by using the OAK-D camera, performing data pre-processing and augmentation to it, and running a YOLOv8 object detection model on those images.

Fig. 4.11 we observe that the model performs moderately well on rust images taken with the OAK-D camera, with a precision of 0.6, a recall of 0.5, a validation/class loss that settles to 1 and a mAP at IoU50 of 0.5.

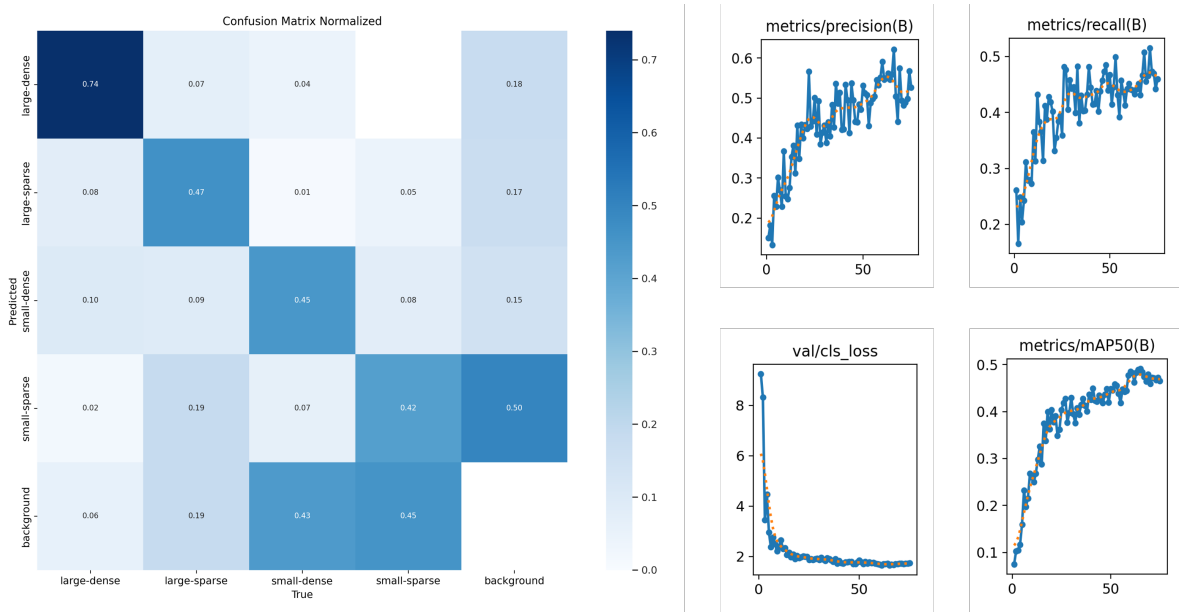

Figure 4.11: YOLOv8 results for anomaly detection and classification of rust with OAK-D.

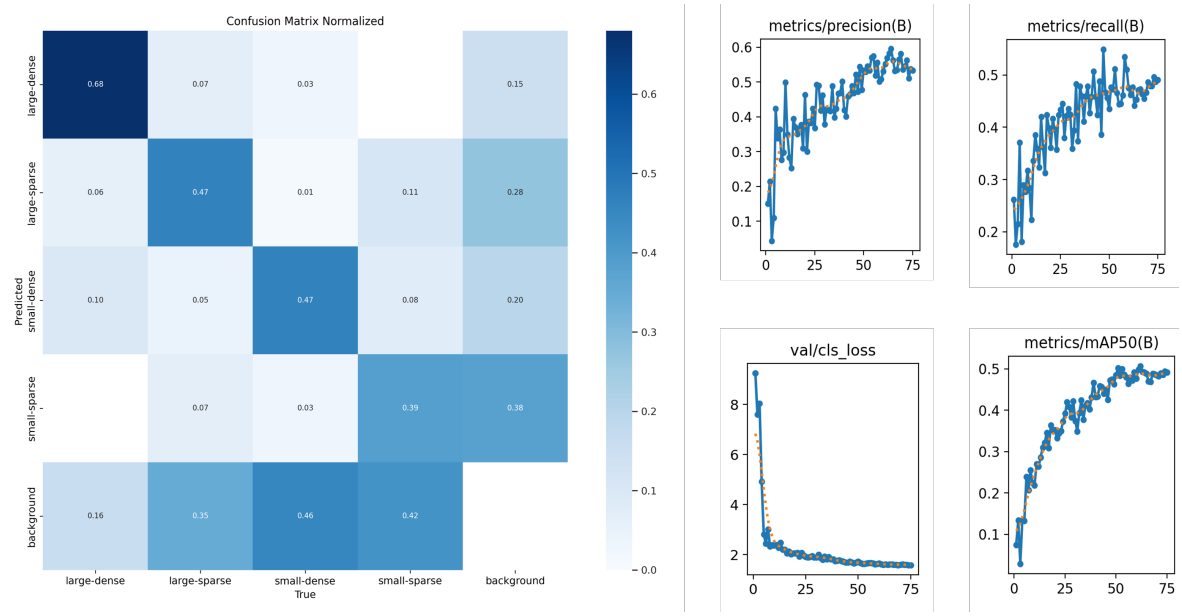

Figure 4.12: YOLOv8 results for anomaly detection and classification of rust with OAK-D with cosine learning rate scheduler.

Next we can take a look at the validation performance metrics for the YOLOv8 model run on our custom rust dataset collected with the OAK-D camera. This iteration of the machine learning model utilizes a hyperparameter modification of cosine scheduler to adaptively modify the learning rate. Fig. 4.12 demonstrates that the model performs slightly better than the OAK-D rust detection model without cosine learning rate. The OAK-D rust detection model with cosine performs with a precision of 0.6, a recall of 0.57, a validation/class loss that settles to 1 and a mAP at IoU50 of 0.5.

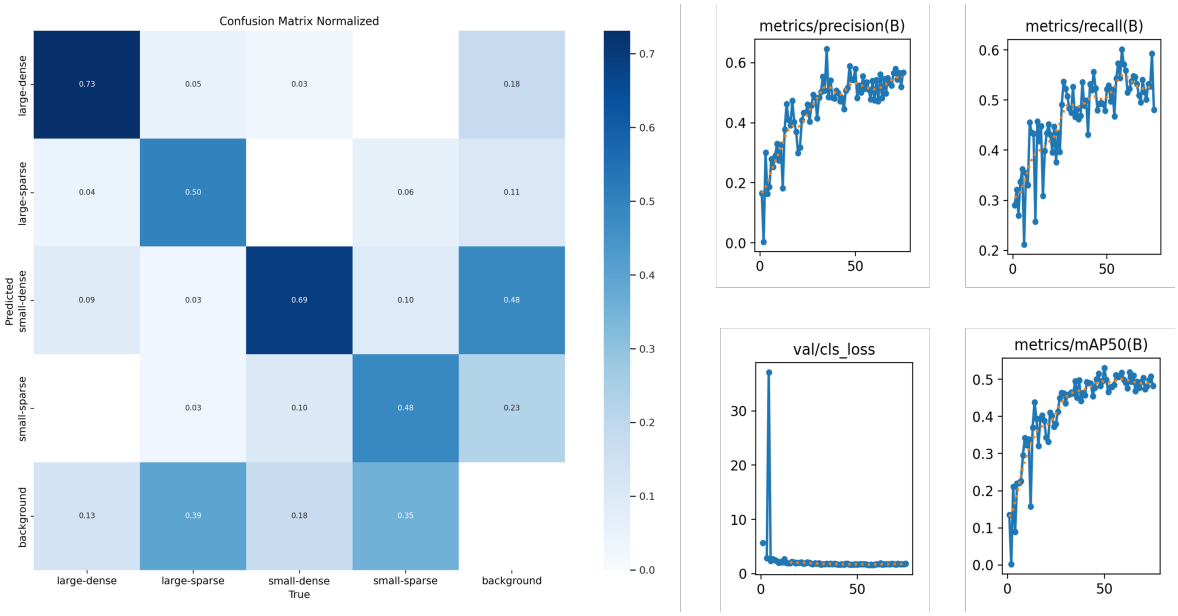

Figure 4.13: YOLOv8 results for anomaly detection and classification of rust with RealSense.

Now let us consider the performance of the model when processing images of rust anomalies obtained using the RealSense camera. Fig. 4.13 shows the validation performance metrics for this model. The model performs moderately well with a precision of 0.6, recall of 0.6, a validation/class loss that settles to 1 and a mAP at IoU50 of 0.5

We can also take a look at how the YOLOv8 performs on a rust images dataset obtained with the RealSense camera after we use the cosine scheduler to adaptively modify the learn-

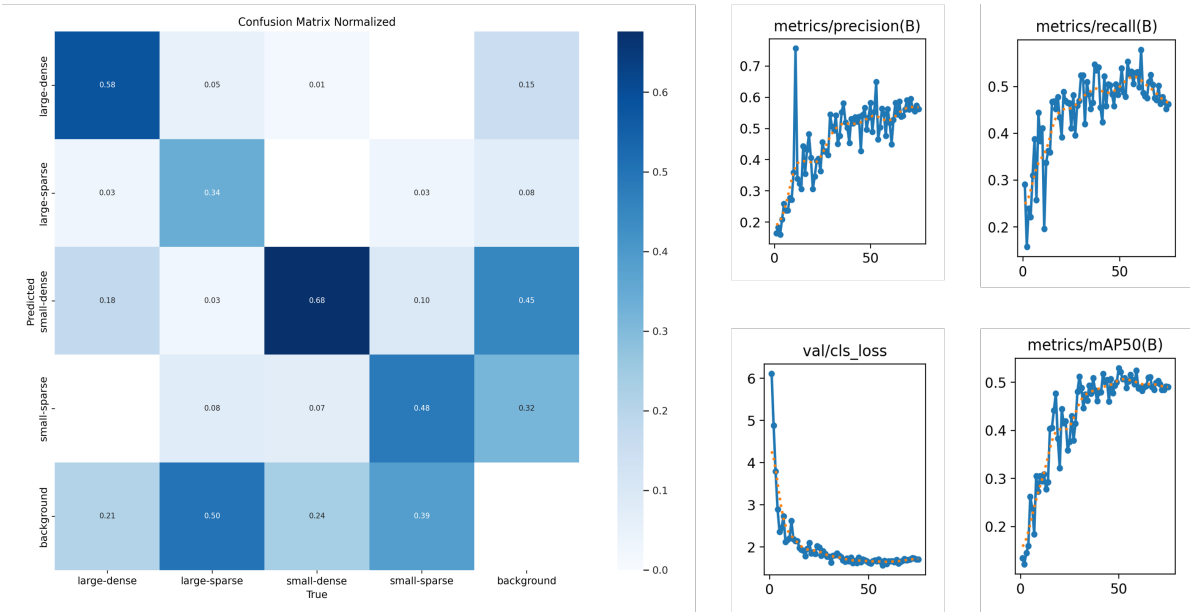

Figure 4.14: YOLOv8 results for anomaly detection and classification of rust with RealSense with cosine learning rate scheduler.

ing rate. The model performs moderately well, the RealSense model with cosine learning rate scheduler performs slightly better due to a spike in the precision metric, but overall the RealSense rust detection model without 'cosine lr' works better. Fig. 4.14 shows the validation results with a precision of 0.7, a recall of 0.58, a validation/class loss that settles to 1 and a mAP at IoU50 of 0.5.

Now we can analyse the results that we have just listed, and study the differences to understand the effect of sensors on the performance of the framework of using robot vision for rust detection and classification. For the case of rust anomaly identification, the OAK-D model with 'cos lr' performs better than the OAK-D model without 'cos lr' and hence the OAK-D model with 'cos lr' is taken under consideration. The RealSense model without 'cos lr' performs better than the RealSense model with 'cos lr' and hence the RealSense model without 'cos lr' is taken under consideration. From Fig. 4.15 we can observe that the OAK-D model with 'cos lr' has better mAp at IoU 50 than the RealSense model without 'cos lr'.

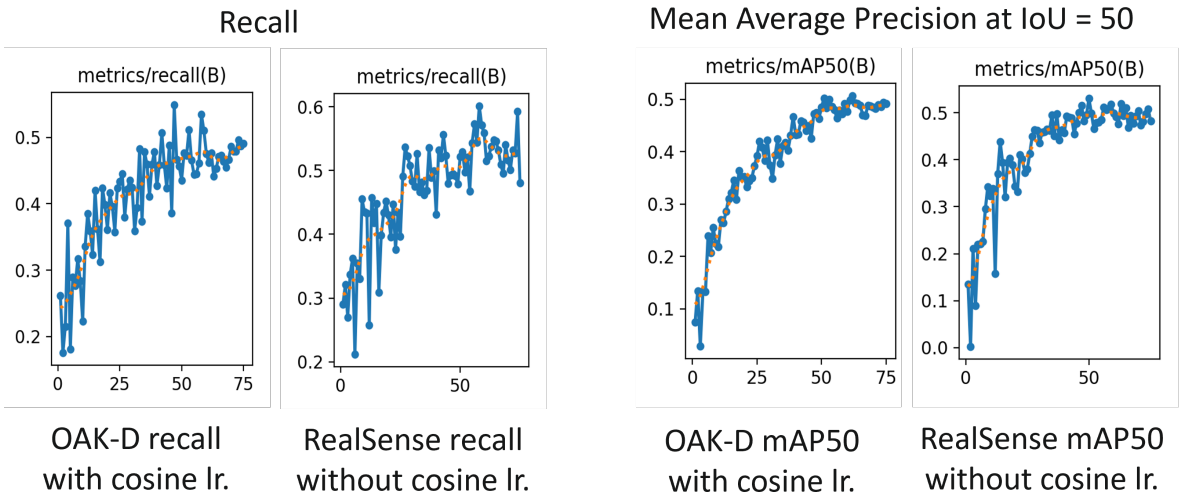

Figure 4.15: Comparison between OAK-D metrics and RealSense of YOLOv8 results metrics of precision and recall for anomaly detection and classification of rust.

However the RealSense model without 'cos lr' has better recall than the OAK-D model with 'cos lr'.

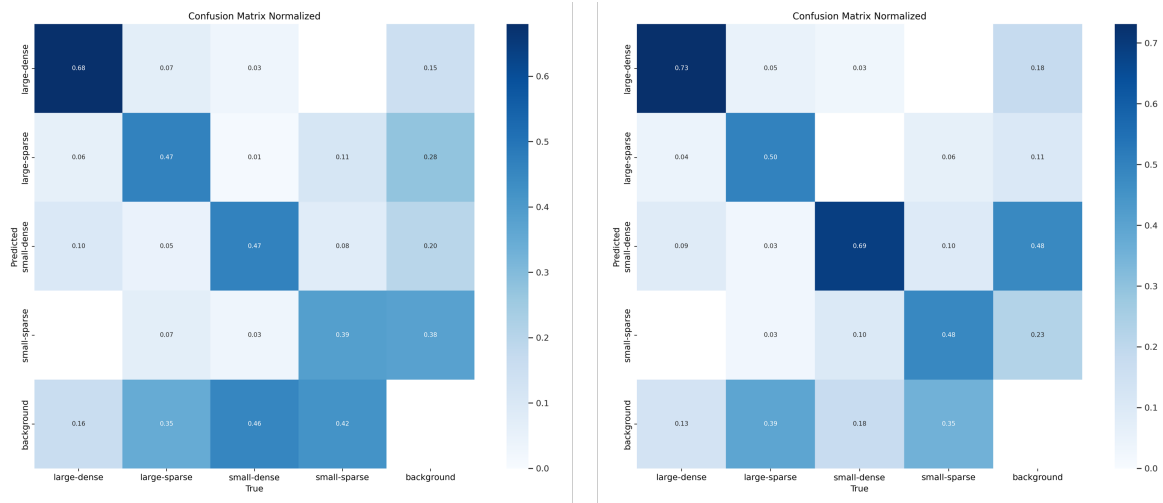

Figure 4.16: Comparison between OAK-D (Left) and RealSense (Right) of YOLOv8 results of confusion matrices for anomaly detection and classification of rust.

Furthermore we can analyze the confusion matrices of the OAK-D model with 'cos lr' and the RealSense model without 'cos lr'. Fig.4.16 shows that the OAK-D misidentifies small sparse rust patches as the background, which aligns with expected behavior. RealSense has unexpected behavior in terms of class misidentification.

Chapter 5

CONCLUSIONS

5.1 Contributions

This research study has successfully built and tested the framework to characterize vision solutions for uncommon environments and anomalies. I have focused on confined spaces as a specific environment to study anomaly detection and recognition. The research has also tested the ability of depth cameras to recognize rust classification which is an unconventional anomaly in comparison to standard objects considered by object detection models.

Some of my key contributions are:

- Designed and developed custom components for modifying and assembling a ground robot.
- Performed hardware and software integration, created a software pipeline for extracting RGB-D images from different vision sensors.
- Designed experiments and collected data for unique anomalies such as rust and FOD appearing in confined spaces, thus contributing to creating a dataset in a domain where data and previous work is limited.
- Tested the effect of various data pre-processing and data augmentation steps on the size and quality of a rust or FOD dataset.
- Integrated a state-of-the-art deep learning object detection YOLOv8 model into the pipeline to perform anomaly detection and classification.
- Explored the modifications of various hyperparameters in the YOLOv8 model and the subsequent effect on anomaly detection and classification metrics.

I have developed a pipeline for rust identification and classification as well as for FOD identification and classification in the same framework from the same environment. This involves diving into the depths of analysing different types of anomalies and the specific classes for each type of anomaly and studying the effect of vision sensors on each of those anomaly types and classes.

The robot framework pipeline that is produced in this research is scalable, adaptable, robust, customizable, fast and generates significant results in terms of anomaly detection and classification.

5.2 Future Work

As a scope of the future work of this project, one of the main areas to focus on will be to integrate depth information into the anomaly detection and classification pipeline. This research focuses on the proposal of using RGB-D images of sections of a confined space environment, as an alternative to using point clouds of the entire confined space environment. So far, we have established a pipeline for collecting RGB-D images. However, in the machine learning phase, we have only studied the model's performance based on color images for the two anomaly categories, rust and FODs, considering the images obtained by two different vision sensors. A natural progression is to now run the model on the depth images, and study the model performance, sensor performance and the effect of depth data on the anomaly detection and classification results.

In order to understand if depth will have a significant impact on the results of anomaly detection and classification, we can refer to Fig. 5.1 where I have generated contour plots based on the depth data received by each camera. The figure demonstrates how a sample scenario looks as perceived by the RGB lens of each camera and the depth sensor. The contour plots are generated solely based on the depth information obtained by each camera. As we can observe, each camera captures different depth information related to boundaries and geometric outlines of objects, structural components and anomalies. It is a crucial observation that the contours are observed to be formed around the rust patches as well,

Figure 5.1: Camera performance comparison based on color and contours generated from depth images of a sample scenario. RealSense color image (Top left), RealSense contour plot (Top right), OAK-D color image (Bottom left), OAK-D contour plot (Bottom right)

which is very interesting as rust as an object has very little depth and is a flat 2D object that occurs on walls. This proves that the depth cameras pick up some influential information in the depth data and it was be very compelling to look into how this depth data affects the performance of the model in terms of anomaly detection and classification and to analyse the same for different vision sensors.

This brings us to the question of how to fuse RGB and depth data. There are two approaches to this problem, early fusion and late fusion. As the name suggests, early fusion is when RGB and depth information is fused before passing it through a machine learning model. Late fusion is when RGB and depth information is individually passed to a machine learning model and the weights extracted from those are fused after model training and are used for validation and testing.

Another avenue that I would like to explore is to combine the RGB-D images received from the OAK-D and the RealSense into one large dataset containing images from both vision sensors. From the results and conclusions in Fig. 4.9, Fig. 4.15, and Fig. 4.16, we

have observed that one camera has the advantage in some metrics and the other camera performs better in other metrics. It will be very interesting to see if combining the datasets from both the cameras yield better performance metrics overall.

Further exploration can be done on the data augmentation front by using variations of hue, saturation and brightness to create a more robust dataset. We can also find more relevant images from the internet to augment it with the current dataset to make it more versatile and deployable on real-world confined space environments.

Furthermore, the findings of this research and the future steps can be integrated with a broader framework for autonomous robot inspection of confined spaces, and be used to correlate anomaly detection metrics and classes to object recognition and pose estimation. This research has explored the avenues of rust and FOD detection and classification in confined spaces based on RGB-D images obtained from two cameras and using a YOLOv8 machine learning model. This framework can be explored and developed further to enhance the results obtained for anomaly detection.

BIBLIOGRAPHY

[1] Ahmad Ali Imran Mohd Ali, Shahrizan Jamaludin, Md Mahadi Hasan Imran, Ahmad Faisal Mohamad Ayob, Sayyid Zainal Abidin Syed Ahmad, Mohd Faizal Ali Akhbar, Mohammed Ismail Russtam Suhrab, and Mohamad Riduan Ramli. Computer vision and image processing approaches for corrosion detection. *Journal of Marine Science and Engineering*, 11(10), 2023.

[2] Ahmad Alsayed, Mostafa R. A. Nabawy, Akilu Yunusa-Kaltungo, Mark K. Quinn, and Farshad Arvin. An autonomous mapping approach for confined spaces using flying robots. In *Towards Autonomous Robotic Systems: 22nd Annual Conference, TAROS 2021, Lincoln, UK, September 8–10, 2021, Proceedings*, page 326–336, Berlin, Heidelberg, 2021. Springer-Verlag.

[3] Héctor Azpúrua, Adriano Rezende, Guilherme Potje, Gilmar Júnior, Rafael Fernandes, Victor Miranda, Levi Filho, Jaco Domingues, Filipe Rocha, Frederico Sousa, Luiz Guilherme Barros, Erickson Nascimento, Douglas Macharet, Gustavo Pessin, and Gustavo Freitas. Towards semi-autonomous robotic inspection and mapping in confined spaces with the espeleorobô. *Journal of Intelligent Robotic Systems*, 101, 04 2021.

[4] Saifullahi Aminu Bello, Shangshu Yu, and Cheng Wang. Review: deep learning on 3d point clouds, 2020.

[5] Prahar M. Bhatt, Rishi K. Malhan, Pradeep Rajendran, Brual C. Shah, Shantanu Thakar, Yeo Jung Yoon, and Satyandra K. Gupta. Image-Based Surface Defect Detection Using Deep Learning: A Review. *Journal of Computing and Information Science in Engineering*, 21(4):040801, 02 2021.

[6] Rune Y. Brogaard, Ole Ravn, and Evangelos Boukas. Absolute localisation in confined spaces using deep geometric features. *Electronics Letters*, 57(16):621–623, 2021.

[7] Dimitrios Chatziparaschis, Michail G Lagoudakis, and Panagiotis Partsinevelos. Aerial and ground robot collaboration for autonomous mapping in search and rescue missions. *Drones*, 4(4):79, 2020.

[8] Intel Coorporation. Intel realsense d435i. 2013.

[9] Luxonis Holding Coorporation. Usb3c-ob oak-d. 2023.

[10] Nelson J. Hansen T. et. al. (2024). Dwyer, B. Roboflow (version 1.0) [software].

[11] Georg Halmetschlager-Funek, Markus Suchi, Martin Kampel, and Markus Vincze. An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments. *IEEE Robotics Automation Magazine*, 26(1):67–77, 2019.

[12] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8. 2023.

[13] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, and Shilei Wen. Pp-yolo: An effective and efficient implementation of object detector, 2020.

[14] International Health Regulations Organization. *Handbook for Inspection of Ships and Issuance of Ship Sanitation Certif cates, ISBN 978 92 4 154819 9*. World Health Organization, 2005.

[15] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Application of deep learning for object detection. *Procedia Computer Science*, 132:1706–1717, 2018. International Conference on Computational Intelligence and Data Science.

[16] Occupation Safety and U.S. Department of Labor Health Administration. Safety and health topics / confined spaces. https://www.osha.gov/confined-spaces.

[17] Luís C Santos, Filipe N Santos, EJ Solteiro Pires, António Valente, Pedro Costa, and Sandro Magalhães. Path planning for ground robots in agriculture: A short review. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 61–66. IEEE, 2020.

[18] Yongxin Shao, Zhetao Sun, Aihong Tan, and Tianhong Yan. Efficient three-dimensional point cloud object detection based on improved complex-yolo. *Frontiers in Neurorobotics*, 17, 2023.

[19] Masahiro Takahashi, Yonghoon Ji, Kazunori Umeda, and Alessandro Moro. Expandable yolo: 3d object detection from rgb-d images. In *2020 21st International Conference on Research and Education in Mechatronics (REM)*, pages 1–5. IEEE, 2020.

[20] Georgios Tziafas and Hamidreza Kasaei. Early or late fusion matters: Efficient rgb-d fusion in vision transformers for 3d object recognition. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9558–9565, 2023.

[21] Ya Wang and Andreas Zell. Yolo+ fpn: 2d and 3d fused object detection with an rgb-d camera. In *2020 25th International conference on pattern recognition (ICPR)*, pages 4657–4664. IEEE, 2021.

[22] Benjamin Wong, Wade Marquette, Nikolay Bykov, Tyler M Paine, and Ashis G Banerjee. Human-assisted robotic detection of foreign object debris inside confined spaces of marine vessels using probabilistic mapping. *Robotics and Autonomous Systems*, 161:104349, 2023.

[23] Benjamin Wong, Tyler Paine, Santosh Devasia, and Ashis Banerjee. Active anomaly detection in confined spaces using ergodic traversal of directed region graphs. *arXiv preprint arXiv:2310.00588*, 10 2023.