RESEARCH & DEVELOPMENT

Technology of Mapping and NDT for Pipes Inspection

Sameer Hamoush, PhD, PE, GC Professor of Civil Engineering
Sun Yi, Professor of Mechanical Engineering
Ahmed Megri, Professor of Architectural Engineering
Younho Seong, professor of Industrial Engineering

North Carolina A and T State University
Greensboro, NC 27411

NCDOT Project No. 2021-055
January 1, 2021 to June 30, 2023

Disclaimer

The contents of this report reflect the views of the author(s) and not necessarily the views of the University. The author(s) are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of either the North Carolina Department of Transportation or the Federal Highway Administration at the time of publication. This report does not constitute a standard, specification, or regulation.

Acknowledgments

The research team acknowledge the support of NCDOT Steering and Implementation Committee for their guidance and support through the process of the work.

Cabell Garbee	**Chair**
John Kirby	**Research Engineer**
Stephen Morgan	**Co-Chair**
David Holmes	**Member**
Joseph Barbour	**Member**
Erik Seiler	**Member**
Andy Jordan	**Member**

Also, the Pi and all Co-PIs acknowledge the hard work by the students to reach the final line of this projects. This includes the following students:

HossamEldin ElSherif	Graduate Student
M A Muktadir	Graduate Student
Selorm Garfo	Graduate Student
Xingguang Li	Graduate Student
Babatunde Keshinro	Graduate Student
Hadi Khoury	Undergraduate and Graduate Student
Marvin Burke	Undergraduate and Graduate Student

Executive Summary

This project aimed to implement the advances in new technologies to develop a robotic-based autonomous inspection system for underground pipelines. The new technology is based on the combination of intelligent powerful portable software and the newly advancement in mapping techniques. The system can scan and reconstruct the 3D profile of a pipeline in real-time using a networked system of ground robots. LiDAR (Light Detection And Ranging), Infrared (IR) cameras and 3D cameras with other NDT (Non-Destructive Testing) sensors used to generate the 3D profiles of the pipes as the robots with the advanced sensors autonomously move inside the pipes. During the pipe condition assessment, video files, pictures, and other scans like LiDAR and 3D are used to acquire a lot of data. This becomes cumbersome to interpret and make meaning out of. In the context of human decision-making, we aim at reducing and reordering the data to show to the human inspectors to enhance their decision performance. The proposed method is based on computer vision and pattern recognition algorithms that have proven to improve the proficiency of the inspection process in identifying and locating important features and defects. A complete technology transfer system is developed and proposed to NCDOT for implementation.

Table of contents

List of Tables

List of Figures

Introduction

According to the pipeline incidents data provided by PHMSA for nearly ten years, there were 280 total pipeline-related incidents during that time. In addition, there were nearly 538 injuries and 112 fatalities. The data revealed the always-present dangers that pipe inspectors face on a daily basis see Table 1.

Table 1 PHMSA Pipeline Incidents: (2012-2021)

Calendar year	Number	Fatalities	Injuries
2012	28	10	54
2013	24	8	42
2014	27	19	94
2015	26	9	48
2016	38	16	86
2017	23	7	30
2018	36	6	78
2019	25	11	35
2020	27	15	40
2021	26	11	32
Grand Total	280	112	538

To reduce incidents related to pipelines and underground pipe networks, it is necessary to inspect and monitor the condition of said networks. The currently used methods mostly depend on manual visual inspection using CCTV cameras. This process is time consuming and depends mainly on the inspectors' experience and skill level. This project aims to develop a full inspection system which includes a fully developed defect detection algorithm which automatically analyzes the video footage and notes what the defects are and where they exist. It also aims to utilize unmanned ground vehicles which automatically navigate and map the inspection area which saves a lot of time and cost due to it being automated. Finally, the system also aims to present summarized data at the end which would be easy to read and interpret by any future inspector.

Organization of report:

The project has been split into four main tasks and each task has a chapter explaining the work that has been done in it. This report is organized as follows:

1) Literature review: Contains the state-of-the art technologies I the field and previous research information that has been done on the area.
2) Equipment: Information on the equipment used in this project.
3) Task 1: Integration of Ground robots, Arms and Sensors: the purpose of this task is to show the work done on the integrations of the different robots and how they are connected to each other. This task is directly related to task 3 and therefor, some work is shared between them.

4) Task 2: Detection and Classification: shows the algorithms that were tested and developed for defect detection.
5) Task 3: Collecting Data from Sensor Networks: Mainly focuses on the mapping that has been done by the robots and the algorithms used to produce said maps. Directly related to task one since mapping and navigation are working simultaneously.
6) Task 4: Data Reduction: shows the algorithm used to reduce the data gathered and analyzed.

Literature Review

A city's drainage system is a crucial part of its infrastructure. Its pipes deteriorate gradually as the drainage system ages and depend on factors such as usage conditions and extrinsic factors [1]. Corrosion of the pipe surface must be detected quickly and accurately. However, manual surveying by human inspectors is notoriously time-consuming and labor-intensive. The task of detecting pipe corrosion can be automated with image processing by using an image processing-based method [2].

Various technologies have been applied in pipe inspection. A most recent survey conducted a thorough investigation of these methods and categorized them into visual methods, electromagnetic methods and acoustic methods, and ultrasound [3]. Among them, CCTV (visual method) is one of the most widely applied sewer inspection methods due to its relative ease of use and the municipalities and contractors' familiarity with this technology [4]. The CCTV method is subjective and lacks image analysis to support the operator reviewing the video. Inspectors have to stop and check the area of interest during the inspection.

To assist decision-makers in evaluating culvert conditions, a variety of inspection technologies and procedures are available. These inspection methods range from simple end-of-pipe visual inspections to complex culvert condition profiles generated by laser profiling. Inspection procedures are usually classified into three main categories:

There is the video recorded inspection, end-of-pipe inspection, and measurement-based inspection. This video-recorded inspection is straightforward. It simply uses advanced imaging technology to record and capture the condition of the culvert or pipe. The end-of-pipe inspection entails visually inspecting and recording the condition of a culvert. A regular part of an end-of-pipe examination is measuring the diameter at the inlet and exit of a culvert. For the measurement-based inspection, the internal diameter, ovality, slope, debris quantity, location/extent of holes in the pipe wall, and wall strength are all different types of measurements taken. This is done by using inspection equipment to measure and assess the wall integrity from within the culvert.

End-of-pipe inspection is a low-cost inspection approach that is also quick and allows for a full-system assessment. However, when it is not complemented with manned access, it only provides a limited view into the culvert. Inspection data is only as exact as the inlet's reach/access, and accuracy decreases as lighting decreases.

Closed-circuit television (CCTV) inspection techniques are still the most widely used approach for assessing the condition and operability of sewer systems. Despite the undeniable improvement in video quality over the last few decades, the CCTV concept for sewer inspection has been consistently criticized [5]. Two major concerns have been identified:

The first drawback is that only a snapshot of the condition is obtained, which means there is no information on the degradation process or reason. Another issue is the lack of accuracy and consistency. Due to the reliance on human observation of images, the result has a high rate of false negatives and positives in defect identification. Furthermore, individual inspectors' numerical classifications, which range from good to poor, can differ by up to two steps.

Efforts to reduce the human factor, such as recent research on applying automated image processing techniques using pattern recognition and machine learning (e.g [4], [6], [7], have shown promising results (accuracies of around 90%), especially in their ability to reduce the percentage of false negatives. Correct classification, on the other hand, remains a challenge. This is because the standards' classification approach is based on semantic descriptions rather than precisely defined measurable metrics.

Manual detection is widely regarded as being less accurate and time-consuming. Machine detection methods based on an ultrasonic, microwave, and other signals have advanced fast in recent [8]–[11]. There are several crack detection methods available, each of which can be classified into one of several categories. The first type includes an excitation device at one end of the structural member and a receiving device at the opposite end. The position and depth of fractures can be determined without causing any structural damage by analyzing the properties of the waveform amplitude and frequency. However, the recognition accuracy of this method is low. The second kind is based on the minimum path [12]–[14], which takes into account the image brightness and geometric properties. The path is made up of a succession of contiguous pixel values, the intensity is determined by the sum of the adjacent pixel values, and the crack is the shortest path. This method necessitates a significant amount of computational effort. The third technique relies on image processing, which entails applying filters to the gathered crack images, such as median and mean filtering, and then recognizing the cracks using the Hough transform, binarization, and tensor voting [15]–[20]. The fourth method relies on classic machine-learning algorithms [21], in which crack images are preprocessed using algorithms like random forest and AdaBoost [22]. Deep learning is another way for crack detection and recognition. This method is distinguished by the use of convolutional neural networks and segmentation algorithms to recognize, segment, and extract cracks, as well as the augmentation of the [23]–[26].

To recognize crack images, three neural networks are used: AlexNet [27], VGGNet13 [28], and ResNet18 [29], and they are tested against each other based on the evaluation indications. Second, to classify crack photos, the crack area must be detected. Currently, two types of detection algorithms are widely used: The first is a two-stage approach based on regional suggestions, such as Fast R-CNN [30] and Faster R-CNN [31], and the second type is one-stage methods, such as SSD [32] and YOLO v3 [33]. Because the YOLOv3 model has the advantage of faster training speed over other models, it is frequently utilized

in the industry, particularly in cases where high precision is not required, such as crack detection. We discovered that YOLOv3 can accurately distinguish crack photos and detect the crack area in real-time in this experiment.

As a result, using a computerized process to detect and analyze surface cracks is extremely valuable in replacing the tedious and subjective inspection of human inspectors [34]. Recent evaluations found a growing trend of using image processing techniques to improve the productivity of crack detection in [34]–[37].

Many fields rely on the quantitative evaluation of surface defects in materials. Accurately estimating the actual sur-face defects of some equipment can directly affect the assessment of that equipment's service life and performance [38]. A study has proposed creating a real-time automated defect detection system using a deep-learning algorithm and CCTV footage. As deep learning techniques develop, a state-of-the-art convolutional neural network (CNN) based object detector, the YOLOv3 network, has been used in this study. The model used in this study has been trained using a data set of 4056 samples containing six types of defects and one type of construction features. Average precision is used to validate the model. A proposed system output includes labeled CCTV videos, frames con-training defects, and associated defect information [1].

A diagnostic system by applying machine learning approaches to CCTV inspection images is proposed by Yang et al. to help general staffs diagnose pipe defects more efficiently. In this study, three neural network approaches were adopted to classify leaky pipe patterns-back propagation neural network (BPN), radial basis network (RBN), and support vector machine (SVM), and their performance was compared [39].

Researchers sought to extract pipe surface features using image texture, gray-level co-occurrence matrix, and gray-level run-length data. A decision boundary can be built to recognize corroded and intact pipe surfaces using the support vector machine. Furthermore, experimental results supported by Wilcoxon signed-rank testing support the validity of the proposed method, with an accuracy rate of greater than 90% [2]. In a paper published in 1999, the basic components and architecture of the system are described; there are four major modules in this product that automate image acquisition, image processing, feature extraction, and defect classification [40]. Using neural networks to classify defects in underground sewer pipes, they published another article the following year. To classify four defect categories, a three-layer (i.e., one hidden layer) neural network was developed and trained using a back-propagation algorithm [41].

Based on defect detection and metric learning, a framework is introduced for tracking multiple sewer defects in CCTV videos to identify the defect across consecutive CCTV frames and the number of defects in the video to determine the pipe condition. The team developed a defect detection model based on deep learning and a metric-learning algorithm

[42]. An article on sewer CCTV inspection videos presents a new approach to automated anomaly detection and localization. Authors exploited 3D Scale Invariant Feature Transforms (SIFTs) to extract spatiotemporal features from sewer CCTV videos [43].

The YOLO model has shown promise in the field of object detection by decreasing computational load and time at the expense of accuracy. This has led to the development of other models such as the Single Shot Multibox Detector (SSD) [44]. The SSD model performs multiple objects detection in a single shot unlike the two-stage counterparts such as the Faster R-CNN model which makes it much faster without sacrificing much accuracy [45]. SSD is mainly comprised of two steps which are extracting feature maps and object detection using convolution filters and it has proven successful [46]. In addition to SSD, the Faster R-CNN model also has been used in object detection [47] and is also used in this project.

In addition to the defect detection algorithms, the fields of ground robot mapping and navigation are crucial in the world of robotics. They involve creating an accurate depiction of a robot's surroundings and providing the capability for it to autonomously navigate through its environment. These abilities are indispensable for numerous operations such as exploration, surveillance, search and rescue, and automated delivery systems.

Sensor-Based Mapping:

In the realm of sensor-based mapping, data gathered from an array of sensors are used to create a depiction of the environment. A common method employed for this is Simultaneous Localization and Mapping (SLAM), which aids robots in developing maps while concurrently determining their location within these maps. Initially, these methods depended on cameras and range finders, but current technological developments have ushered in the use of more advanced tools like LiDAR, depth sensors, and RGB-D cameras, providing more precise and comprehensive mapping.

Mapping in Dynamic Environments:

Dynamic environments, characterized by the presence of moving objects, pose challenges to effective mapping. Multiple techniques have been proposed to tackle the issues of identifying, tracking, and integrating dynamic objects into the map-making process. Probabilistic model-based methods, such as occupancy grids and particle filters, have proven successful in these fluctuating environments.

Path Planning and Navigation:

Following the creation of a map, the robot must navigate through the environment, requiring path planning. Algorithms for path planning strive to discover optimal or near-optimal paths, taking into account elements like obstacles, terrain, and robot dynamics.

Conventional methods, like potential fields and graph-based algorithms, have seen widespread use. Contemporary approaches have begun to leverage machine learning techniques, including reinforcement learning and deep learning, to enhance path planning and navigation performance.

Localization and Sensor Fusion:

Precise robot localization is essential for efficient mapping and navigation. Techniques involving sensor fusion, which amalgamate data from multiple sources like odometry, GPS, IMU, and visual sensors, are typically employed to improve localization precision. Advanced methods of localization, such as feature-based, landmark-based, and map matching, are used to cater to different scenarios and increase robustness.

Multi-Robot Systems:

Ground robot mapping and navigation can significantly benefit from multi-robot systems, offering benefits such as expanded coverage, enhanced collaboration, and improved fault tolerance. Techniques fostering cooperative mapping and navigation have been studied, allowing robots to exchange information and synchronize their actions to achieve shared objectives. Research in this field is centered around communication protocols, task allocation, and coordination strategies.

Equipment

We used two concrete pipes in the structural lab on the A&T campus to test the equipment. The pipes are shown in Figure 1.

Figure 1 Structural lab concrete pipes specimens

For this project, the following 3 different robots carrying multiple sensors were used. These robots are the Husarion ROSbot, the Jackal unmanned ground vehicle (UGV), and the Husky UGV shown in Figure 2.

(a) Husarion ROSbot

(b) Jackal UGV

(c) Husky UGV

Figure 2 Project robot vehicles

The technical details for the robot vehicles are as follows:

The ground robots' details are shown in Table 2.

Table 2 Ground robots' details

Attribute	Husarion ROSbot	Jackal	Husky
External dimensions	200 x 235 x 220 mm	508 x 430 x 250 mm	990 x 670 x 390 mm
Weight	2.84 kg	17Kg	50Kg
Max payload	5 kg	20Kg	75Kg
Max speed	1.0 m/s	2.0 m/s	1.0 m/s
Runtime	1.5 – 5 hours	Basic usage: 8 Hours Heavy usage: 2 Hours	Standby:8 Hours Nominal usage: 3 Hours

As for the LiDARs used, we used two Velodyne Lidars, the VLP-16 and the HDL-32E. They are shown in Figure 3.

(a) VLP-16 (b) HDL-32E

Figure 3 Velodyne LiDARs

The specifications of the Velodyne LiDARs are shown in Table 3.

Table 3 Velodyne LiDARs specifications

	VLP-16	**HDL-32E**
Channels	16	16
Range	Up to 100 meters	Up to 100 meters
Accuracy	+/- 3 cm (typical)	+/- 3 cm (typical)
Field of view	Vertical: 30° (+15° to -15°) Horizontal: 360°	Vertical: 41.33° (+10.67° to -30.67°) Horizontal: 360°
Vertical angular resolution	2°	1.33°
Horizontal angular resolution	0.1° – 0.4°	0.1° – 0.4°
Rotation rate	5-20 Hz	5-20 Hz

In addition to the Velodyne LiDARs, we also used the Intel RealSense LiDAR camera L515 attached to the Husarion ROSbot for use in defect detection. This camera LiDAR uses laser scanning technology and has a field of view (FOV) of 70° and a range of 0.25m - 9m which is affected by reflectivity. Figure 4.

Figure 4 Intel RealSense LiDAR camera L515

Task 1: Integration of Ground Robots, Arms, and Sensors

In this task, the focus is on the integration of robots. This is demonstrated best in the mapping which is technical part of task 3 but is also presented here due to it being highly relevant and it running simultaneously with the navigation algorithm.

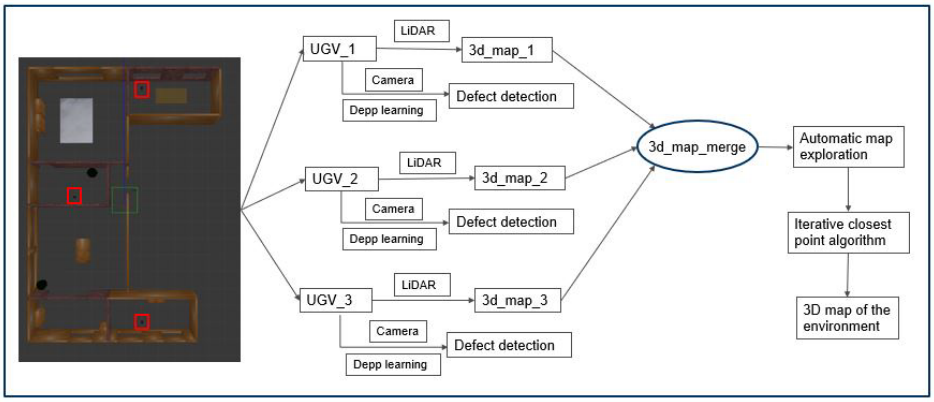


Figure 5 Autonomous 3D mapping and inspection using multiple ground robots

We propose a 3D mapping framework to automatically generate the 3D point cloud of the environment in real time that can be used for determining pipe geometry and navigation. This also allows us to enable the robot to navigate for a specific defect or a specific point for further evaluation. Figure 5.

Each robot will use the LiDAR sensor to generate its own 3D point cloud. Based on the location and orientation of each robot, the 3D map merge algorithm will combine all the 3D point clouds together. The automatic map exploration algorithm will then select navigation point for each robot to obtain a complete 2D map of the environment. While robots are moving, the iterative closest point algorithm will calculate the transformation between the point clouds to build the 3D map of the environment.

Cooperative mapping by multiple robots

Cooperative mapping by multiple robots is a highly advantageous approach, particularly in large-scale environments, where the benefits of collaboration become evident. By distributing the task of exploration among several robots, we can achieve increased coverage and persistence in mapping the surroundings.

The main objective of this cooperative mapping approach is to optimize exploration and mapping efficiency. Through information sharing, robots can exchange their individual maps to prevent redundant efforts. This collaborative strategy enables the environment to be effectively partitioned among the robots, minimizing overlap and maximizing overall exploration efficiency.

By working together, the robots can navigate through the environment in a coordinated manner, efficiently covering more ground and ensuring that each area is explored thoroughly. This cooperative exploration and mapping approach is particularly beneficial in scenarios where time is a constraint or when the environment is vast, complex, or potentially hazardous for a single robot to handle alone.

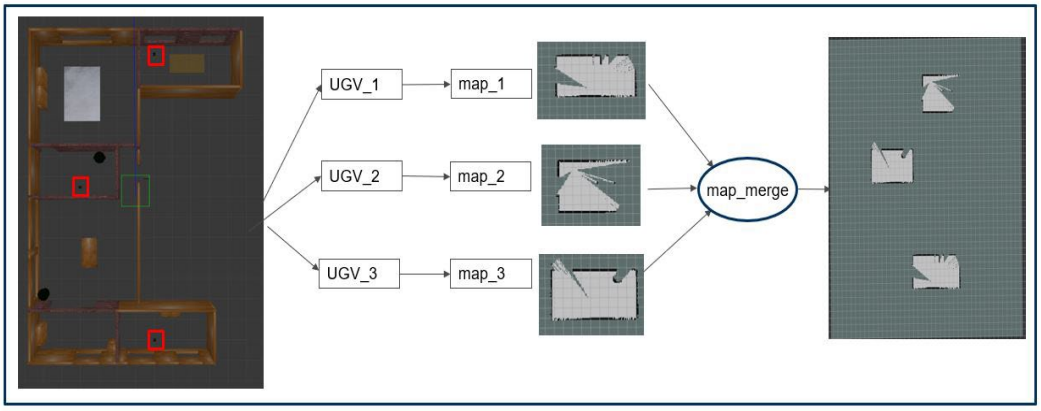
Moreover, the collaborative aspect of the mapping process allows for continuous mapping even if one or more robots encounter obstacles or malfunctions. The other robots can compensate for the affected areas, ensuring the mapping process remains uninterrupted.

2D map merge

Each robot in the team leverages its own set of sensor mechanisms, such as LiDAR, to generate an occupancy grid map specific to its immediate environment. This map is divided into cells, each representing whether the space is occupied, free, or unknown. The process of merging individual 2D maps into a comprehensive map involves determining the real-world distances between the robots and using this information to align the maps accurately. By transposing these distances into Rviz which is a visualization tool, the maps can be combined seamlessly. The algorithm steps and outline are shown in Figure .

Algorithm Multiple maps merge algorithm
Input: maps and initial positions and direction of the robots
Output: merged map
1: **procedure**
2: calculate the distance between robots
3: convert the real distance to the distance in the Rviz
4: combine the maps with the direction of the robots
5: merge the maps
6: **end procedure**

(a) Map merge algorithm steps



(b) Map merge algorithm outline

Figure 6 2D map merge algorithm

Autonomous map exploration

Certain robotics applications necessitate robots to autonomously generate a map of their environment for efficient navigation. A group of robots can initially explore the area and generate a map for subsequent robots to utilize. In this project we utilize two algorithms for this purpose. First, we have the frontier-based map exploration algorithm to detect the edges of the known area. The robots then use the A* algorithm to navigate the known area.

Frontier-based map exploration:

A commonly employed method in autonomous robot navigation is frontier-based map exploration. This algorithm seeks to explore an unknown environment in an organized and efficient manner, thus reducing the resources required for complete exploration. Figure 7.

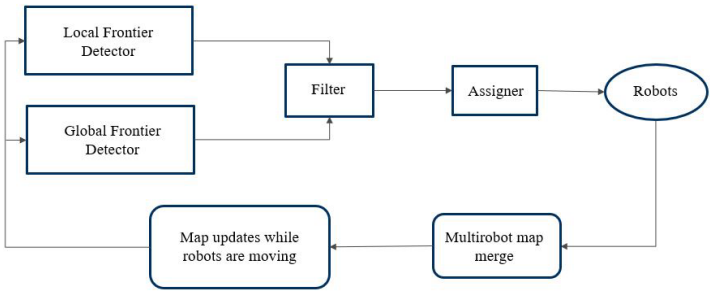


Figure 7 Framework of frontier-based map exploration algorithm

Frontier edges, or boundaries between known and unknown spaces, are extracted from the occupancy grid map. The center of each edge is marked as an exploration target. The mission planner then uses the robot's current location and the available map data to devise the most effective route to the destination point.

A algorithm*

The A* algorithm is frequently used for pathfinding. It's designed to find the shortest path between two points on a graph or grid, blending the strengths of uniform-cost search and greedy best-first search with a heuristic to guide the search. The A* algorithm's process involves initialization, looping, goal-checking, generating successors, and, finally, path construction. Figure 8.

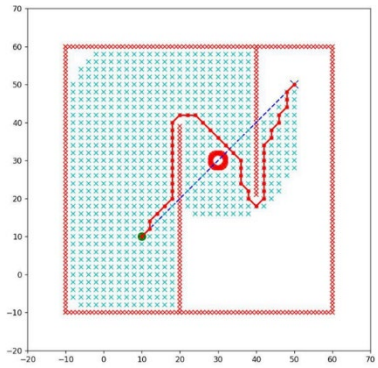


Figure 8 A* algorithm route planning

Here's a step-by-step explanation of the A* algorithm:

Initialization:

- Create an open set that will store the nodes to be explored. Initially, only the starting node is in the open set.
- Create a closed set to keep track of the nodes that have already been explored. Initially, the closed set is empty.
- Assign a "g-score" value of 0 to the starting node. The g-score represents the cost of the path from the start node to the current node.
- Calculate the "h-score" for the starting node using a heuristic function. The h-score estimates the cost of the cheapest path from the current node to the goal node.
- Calculate the "f-score" for the starting node by adding the g-score and the h-score. The f-score is an estimate of the total cost of the path passing through the current node.
- Associate each node with its parent node to keep track of the path.

Loop:

- While the open set is not empty, continue the loop.
- Select the node with the lowest f-score from the open set. This node is considered the current node.
- Move the current node from the open set to the closed set.

Goal check:

If the current node is the goal node, the path has been found. Construct the path by following the parent pointers from the goal node to the start node.

Generate successors:

- Generate the neighboring nodes of the current node that have not already been explored.
- For each neighbor:
  1- Calculate the tentative g-score by adding the cost of moving from the current node to the neighbor.
  2- If the neighbor is already in the closed set and the tentative g-score is greater than the neighbor's g-score, skip to the next neighbor.
  3- If the neighbor is not in the open set or the tentative g-score is lower than the neighbor's g-score:
    a- Update the neighbor's parent to the current node.
    b- Calculate the neighbor's h-score using the heuristic function.
    c- Update the neighbor's g-score with the tentative g-score.
    d- Calculate the neighbor's f-score by adding the g-score and the h-score.
    e- If the neighbor is not in the open set, add it to the open set.

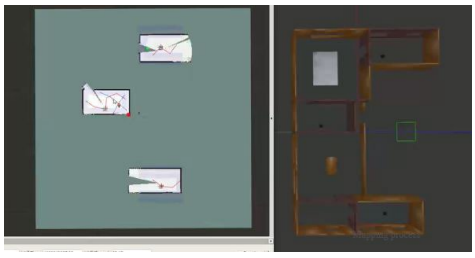
Loop continuation:

If the loop ends without finding the goal node, there is no path from the start node to the goal node, indicating that the goal is unreachable.

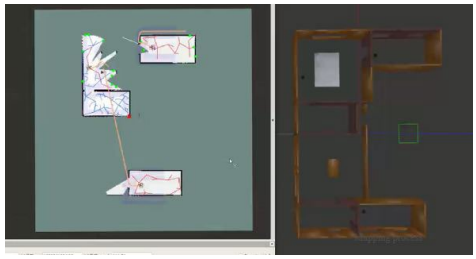
Path construction**:**

Once the goal node has been reached, construct the path by following the parent pointers from the goal node to the start node. Upon reaching a frontier, the robot refreshes its map with new sensor data, identifies new frontiers, and repeats the process until the environment is thoroughly mapped.

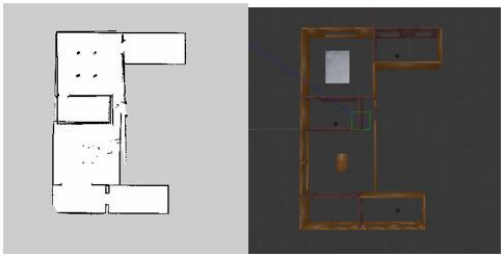
Upon reaching a frontier, the robot refreshes its map with new sensor data, identifies new frontiers, and repeats the process until the environment is thoroughly mapped. The process is shown in Figure 9 .



(a) Map merging with known initial locations



(b) Map updates while robots are moving



(c) Mapping result

Figure 9 Multi-robot automatic map exploration process

The iterative closest point (ICP) algorithm:

The iterative closest point (ICP) algorithm plays a crucial role in the field of point cloud processing, particularly in applications involving LiDAR sensors. It is used in this project to construct one a unified 3D point cloud from multiple smaller ones which will give us a 3D map from multiple robots. Point clouds are three-dimensional data representations consisting of numerous points in space, typically captured by LiDAR sensors. The ICP algorithm serves as a powerful tool for aligning and registering two point clouds, thereby minimizing the differences between them.

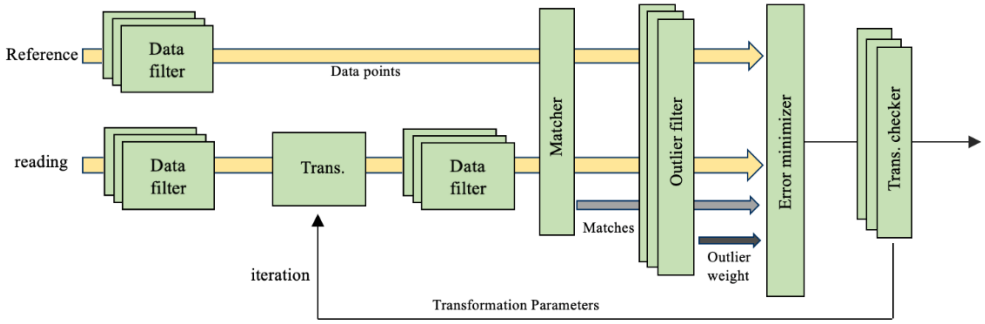


Figure 10 The Framework of ICP algorithm

The primary objective of the ICP algorithm is to estimate the transformation that aligns two point clouds together. By iteratively refining this transformation, the algorithm ensures that corresponding points in the real world align with each other through the computed transformation. Consequently, a coherent and accurate model of the environment can be incrementally constructed. Figure 10.

To achieve this alignment, the ICP algorithm goes through a series of iterations. In each iteration, it associates points from one cloud (the "source" cloud) with their nearest neighbors in the other cloud (the "target" cloud). It then calculates the transformation that minimizes the distance between the corresponding point pairs. This process is repeated iteratively, refining the transformation estimation until convergence is achieved or a predetermined stopping criterion is met.

During each iteration, the ICP algorithm updates the transformation by minimizing the distance metric, typically the Euclidean distance, between corresponding point pairs. This optimization process aims to find the transformation that best aligns the two point clouds. The algorithm continues iterating until it reaches a satisfactory alignment or until the maximum number of iterations is reached. Figure 11.

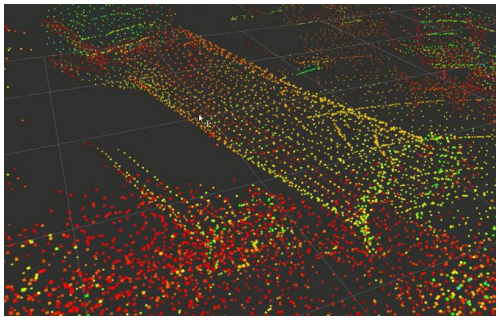
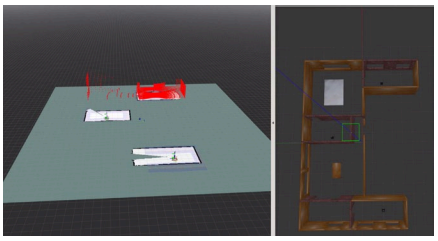
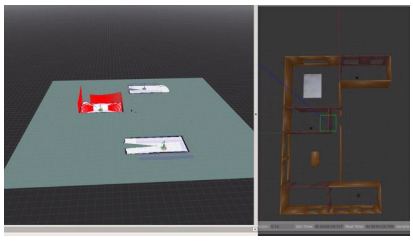


Figure 11 Pipe 3D map generated by ICP algorithm

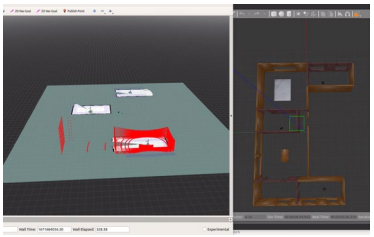
Overall, by combining the ICP algorithm, cooperative mapping, and autonomous exploration techniques, the result is an efficient and comprehensive exploration and mapping framework. The robots can collaboratively generate accurate maps of the environment, optimize exploration efforts, and autonomously navigate and update their maps. Figure 12.



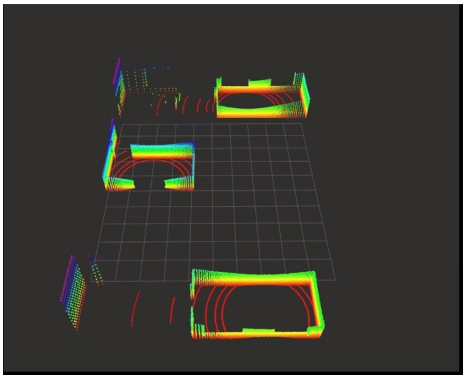
(a) First 3D map



(b) Second 3D map



(c) Third 3D map



(d) Merged 3D point cloud

Figure 12 3D point cloud merging

Task 2: Detection and Classification

In this task, multiple models and technique were developed and tested. The purpose of these models being detecting the defects in the pipe networks. The following sections cover each of the developed models.

Machine learning models

Machine learning has been in common use for years now for several fields. A few models here were developed to be used for the detection of defects. Development of the machine learning models starts first with the dataset. An initial model was made using a few images captured in the concrete pipe in the structural lab to show how big of a dataset we can get from only a few images. The main pint with these models is that each sample is only a small patch of the image where our target class exists instead of using the whole image. This results in one image giving us a lot of samples. The outline of how the trained program works is as follows:

- Crop crack from the image.
- Loop over the object to get patches of square size (1/18 of vertical height of the image)
- Calculate descriptive statistics per patch in the crack.
- Arrange the descriptive statistics into an array per patch.
- Construct a dataset for training the classifier.
- Apply the trained classifier on each patch of the image and highlight if a crack is detected.

An example of patches extraction for training is demonstrated Figure 13.

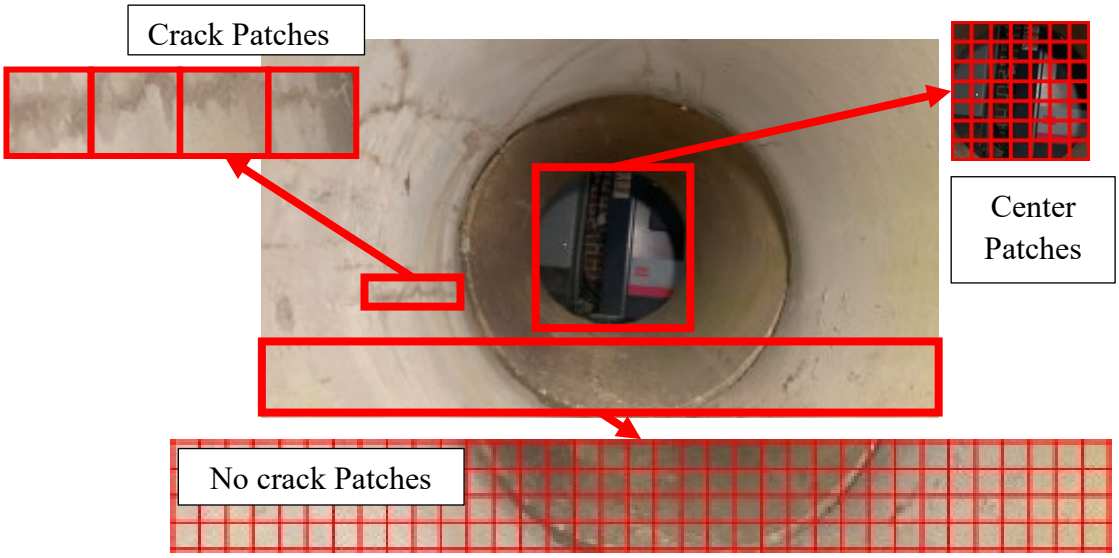


Figure 13 Training sample extraction for machine learning

The data for training is extracted from small parts of the image as shown in Figure 4 and this allows us to extract a large dataset from a small number of images. For each patch, a number of statistics are then calculated for each color channels (once for each of the red, green, and blue color channels) which will constitute our training features. The statistics are as follows:

$$Mean: \quad \mu = \frac{\sum x_i}{n}$$

$$Standard\ deviation: \quad \sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{n}}$$

$$Variance: \quad s^2 = \frac{\sum (x_i - \mu)^2}{n}$$

$$Kurtosis = \frac{\sum (x_i - \mu)^4}{n(s^2)^2}$$

Where x_i is the pixel value in the patch and n is the total number of pixels in the patch. In addition to these statistics, the x and y positions of the patch are added as features resulting in a total of 14 training features per patch. The total number of samples extracted in the first model was 618 from 4 images. Preliminary results using a random forest classifier indicated that the model is very accurate. The detection results marking the position of cracks is shown in Figure 5.

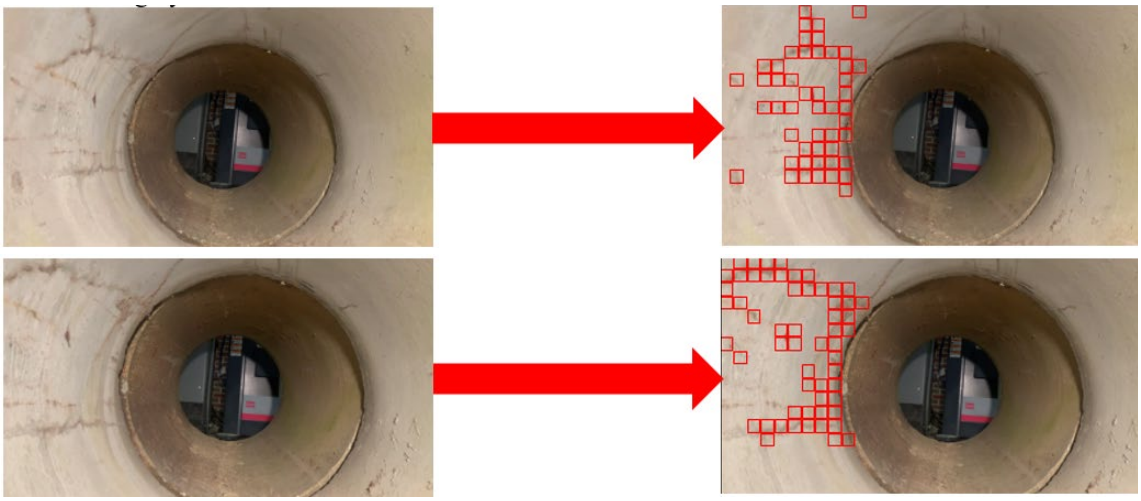


Figure 14 Machine learning preliminary results

The model was further improved to include more attributes bringing the total number of features per patch to 26 and retrained using NCDOT inspection footage. The total number of samples for training was 1162 and four different classifiers were utilized to test the model. These classifiers were random forest (RF), support vector machine (SVM), logistic regression (LR), and K-nearest neighbor (KNN). The performance metrics and an example of the detection result are shown in Table and Figure , respectively.

Table 4 Machine learning models performance metrics

Model	**5-Fold Cross Validation**		**Accuracy**	**Precision**	**Recall**	**F1-score**
	Accuracy	Std. Dev.				
SVM	0.81	0.02	0.84	0.84	0.84	0.83
RF	0.82	0.03	0.82	0.82	0.82	0.81
LR	0.69	0.03	0.66	0.65	0.66	0.65
KNN	0.78	0.01	0.77	0.80	0.77	0.74

The results indicate good accuracy when validated against known data but applying the model to some of the footage results in low detection accuracy. Reasons for this include the different lighting and conditions of each inspection location. To account for this, additional development time would be required to account for these factors. This can be done by incorporating depth data from the LiDAR into the training data.

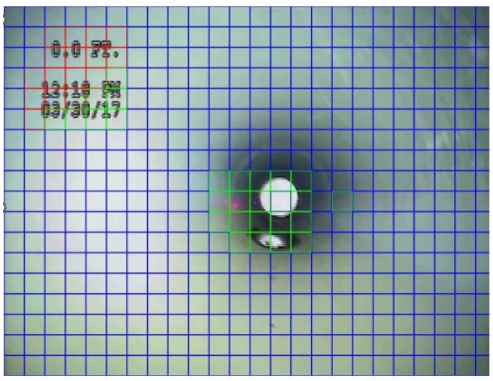


Figure 15 Machine leaning model example results

YOLOv5 model

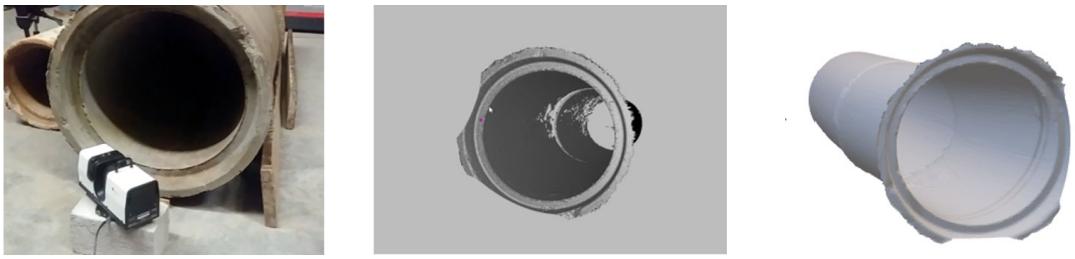
According to the National Association of Sewer Service Companies (NASSCO) documents and the Pipeline Assessment Certification Program, major pipeline defects can be classified under structural defects and operational and maintenance defects. For this model, six types of defects have been considered considering the availability of data as follows:

- Crack
- Fracture.
- Surface damage.
- Blockage.
- Joint.

In this study, two types of data have been used. Firstly, images have been collected from the prerecorded video of the sewer piping system. Data was provided by the funding agency, the North Carolina Department of Transportation (NCDOT).

From 111.2 GB of video files, files have been selected of 31.8 GB (51 recorded videos) considering the two factors, pipe types (only RC pipes) and the considered defects type for that study (Crack, Fracture, Surface Damage, and Deposit (left) and obstacle (right)). From the selected recorded videos using Python, the images were captured every 3 seconds. Initial images were 3246 nos., which was further shortened to 154 nos. by removing the bad images that may reduce the efficiency of the further preprocessing.

Secondly, 2D images have been collected from a 3D scan; in that case, a high-resolution 3D scanner has been used. In this data collection process, an Artec3D scanner was used. The joint data was extracted using this method and is considered synthetic data. Figure 16.



(a) 3D scanner setup (b) Post processing (c) 3D model

Figure 16 Data collection from 3D scanner

The total number of images was not much so more data was needed. To address that issue, the extracted images (181) were processed to augment the dataset, Figure 17. The augmentation was performed as follows:

a) Flip the images: In that case, the original data set images have been flipped only vertically direction.
b) RGB to Gray: All the images have been converted to grayscale.
c) Rotation 1: All the initial data set has been rotated to 45 degrees.
d) Rotation 2: All the initial data set has been rotated to 90 degrees.
e) Rotation 3: All the initial data set has been rotated to 180 degrees.

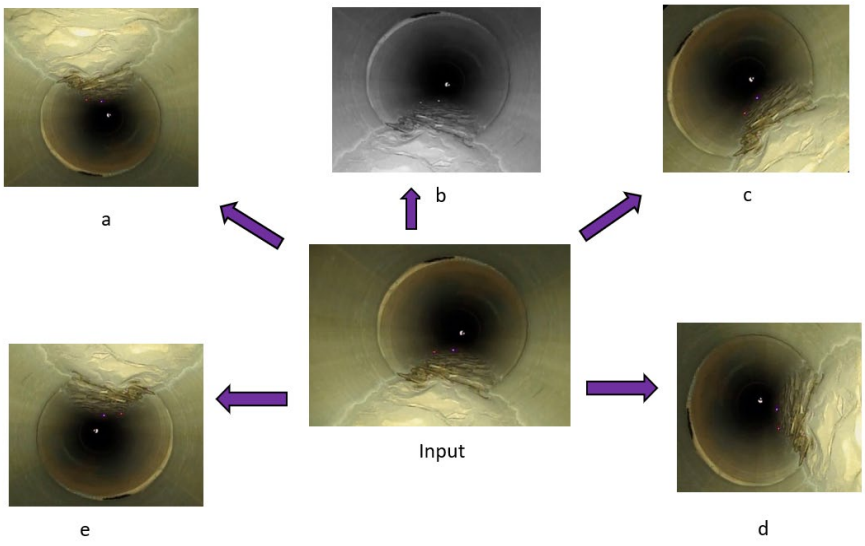


Figure 17 Dataset Augmentation

The dataset size is shown in Table for each considered defect class before and after augmentation:

Table 5 YOLOv5 model dataset size per class

Types of defects	Number of images before augmentation	Number of images after augmentation
Crack	39	234
Fracture	39	234
Surface Damage	39	234
Blockage	39	234
Joint (synthetic)	25	150
Total	181	1086

The state-state-of-the-art model YOLOv5 has been trained and tested in this project and the architecture of the model is shown in Figure . The backbone network convolves the input image and converts it into feature maps of different sizes. The neck network fuses high-level and low-level feature maps, and the fused feature maps are input into the detect network to detect objects in the picture [48].

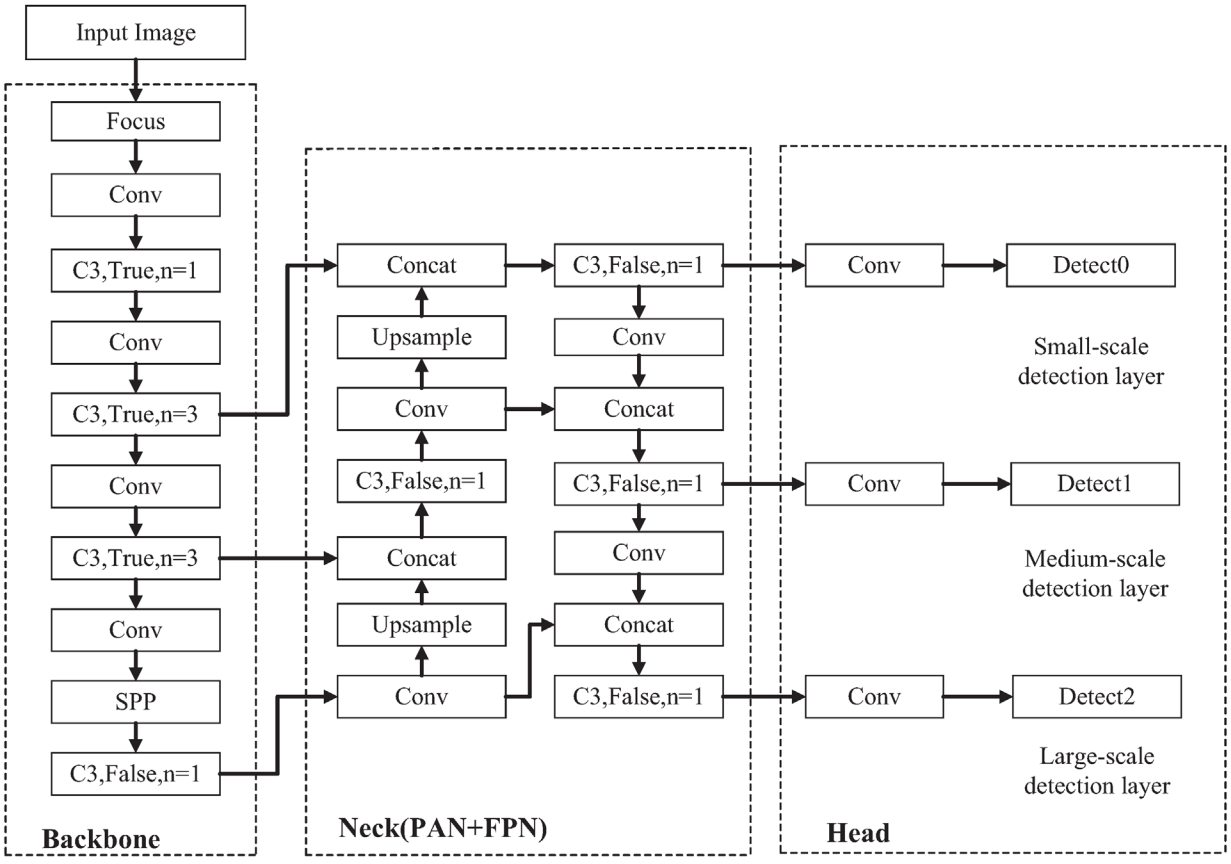


Figure 18 Overall network structure of the YOLOv5s model

The initial training hyper parameters were set as follows in Table 6:

Table 6 YOLOv5 Training hyper parameters

Parameter	Value
Learning rate	0.01
Weight decay	0.0005
Momentum	0.937
Batch size	32
Epoch	100

Three performance metrics were used to compare the models' performance, these metrics are as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{FN + TP}$$

$$F1\ Score = \frac{2 \times P \times R}{P + R}$$

Where, TP = True Positive, FP= False Positive, FN= False Negative

The model was retrained with different parameters and Table shows the results of the different models, where model 9 shows the maximum F1 score.

Table 7 YOLOv5 Models results

Model	Batch Size	Epoch	Precision (P)	Recall (R)	F1
1	32	50	0.679	0.67	0.67
2	16	10	0.385	0.565	0.46
3	16	10	0.489	0.544	0.52
4	16	20	0.567	0.616	0.59
5	16	30	0.631	0.628	0.63
6	16	40	0.715	0.655	0.68
7	16	50	0.648	0.682	0.66
8	**32**	**50**	**0.674**	**0.681**	**0.68**
9	**32**	**100**	**0.714**	**0.662**	**0.69**
10	32	200	0.71	0.648	0.68
11	8	20	0.58	0.63	0.60
12	8	50	0.711	0.662	0.69
13	8	20	0.594	0.638	0.62
14	8	50	0.683	0.69	0.69
15	8	100	0.664	0.666	0.66

The following confusion matrix in Figure 19 shows the predicted results of the validation data of all the classes, crack (CR), joint (JO), fracture (FR), surface damage (DA), and Deposits and obstacles/Obstructions (BL). The performance metrics for model 9 are also shown in Figure 20 for each defect class vs. the confidence level which shows the joints having the maximum confidence level and cracks having the lowest.

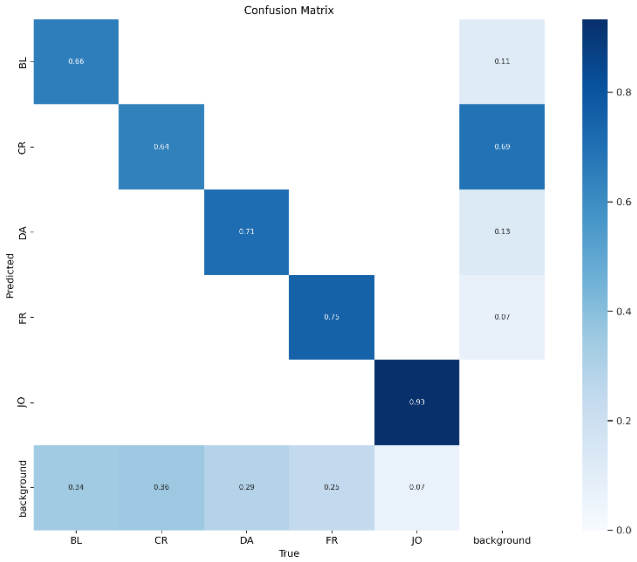
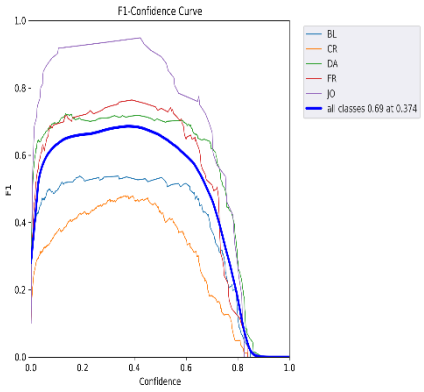
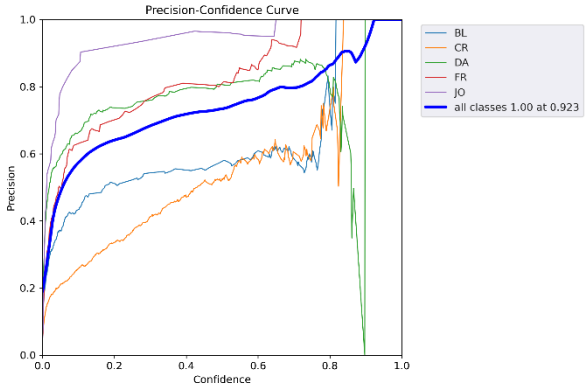


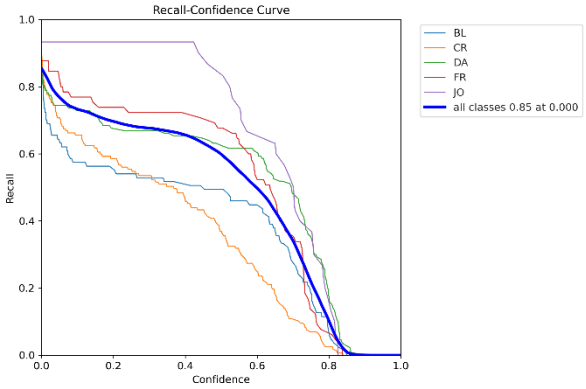
Figure 19 Confusion matrix of model 9



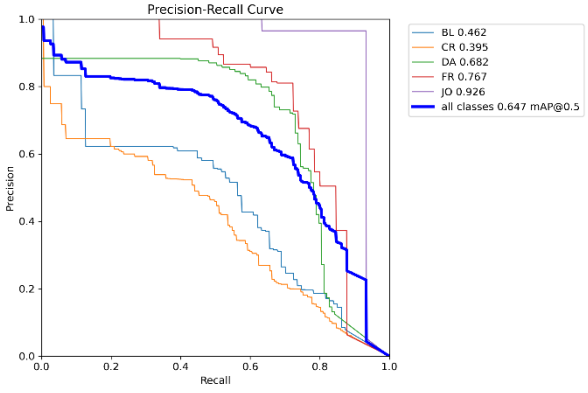
(a) F1 score vs. confidence curve



(b) Precision vs. confidence curve



(c) Recall vs. confidence curve



(d) Precision vs. confidence curve

Figure 20 Performance metrics vs. confidence curves for model 9

YOLOv7 model

Following on the same deep learning models, three different deep learning models were developed and tested on the NCDOT dataset. These models were the SSD MobileNET, the YOLOv5, and the Faster R-CNN. The difference being they were trained and tested on 9 different defect classes of a total of 16 defined classes. The idea being that the best model would then be further improved and retrained with all 16 classes. These classes are shown in Figure .

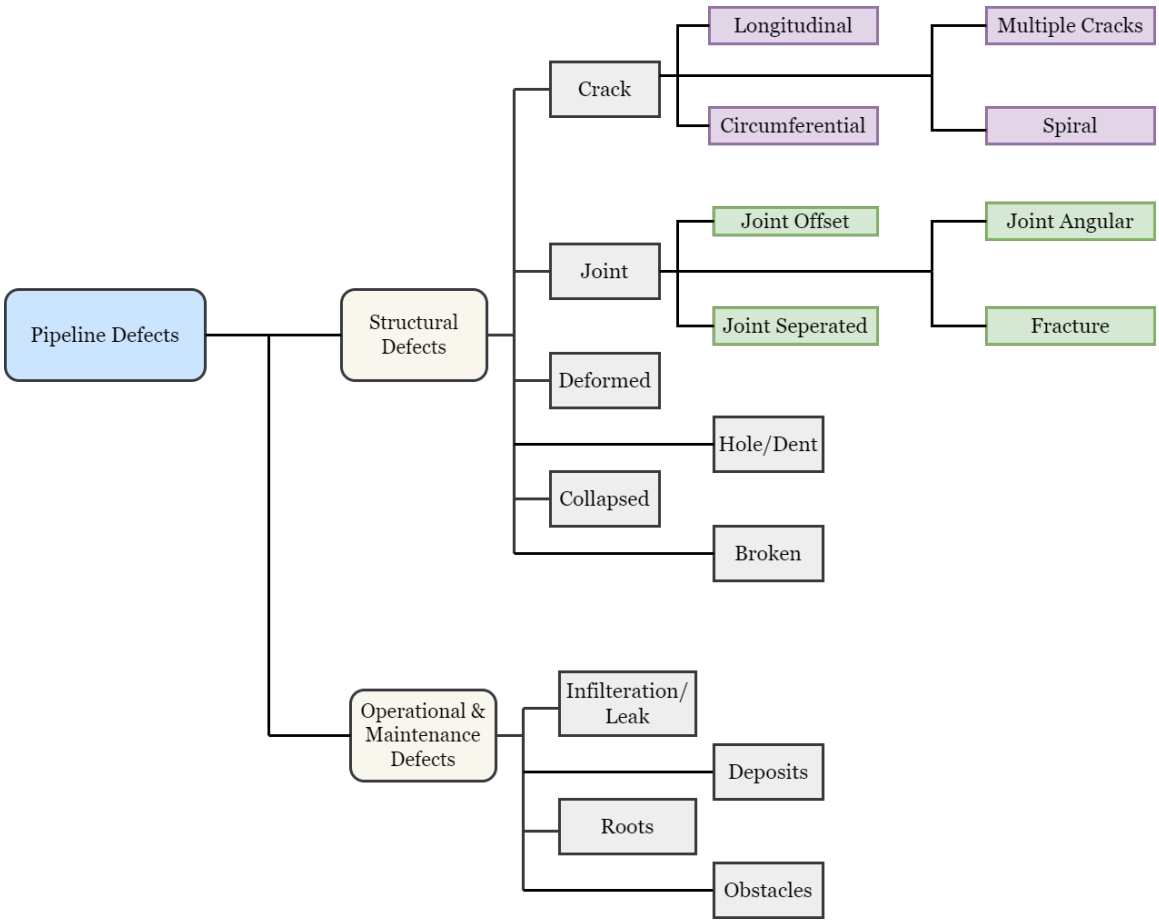


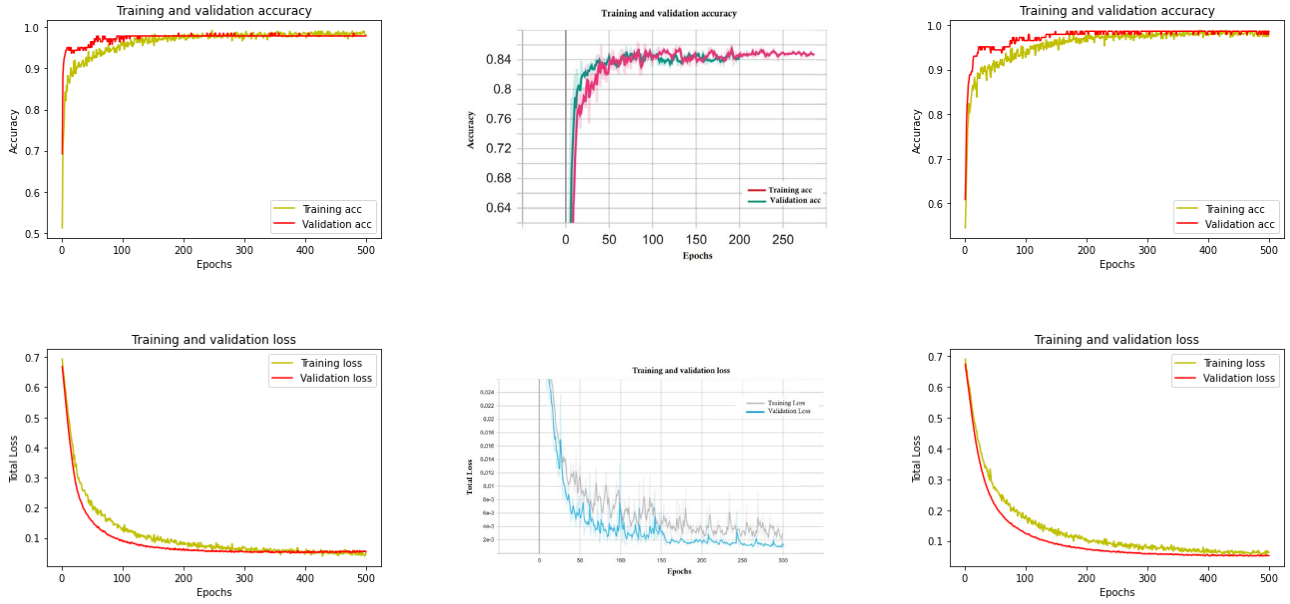
Figure 21 Defect classes

The models were trained with different backbones and tested using a computer with an NVIDIA GeForce RTX 3080 GPU. The performance of these models indicates that the YOLOv5 model while having the least mean average precision (mAP), would be the most appropriate. This is due to speed of the processing per frame coupled with the difference in mAP being small. The model was chosen to be further improved and include more classes.

The performance of the three initial models is shown in Table 8 and Figure 22..

Table 8 Initial models speed vs. accuracy

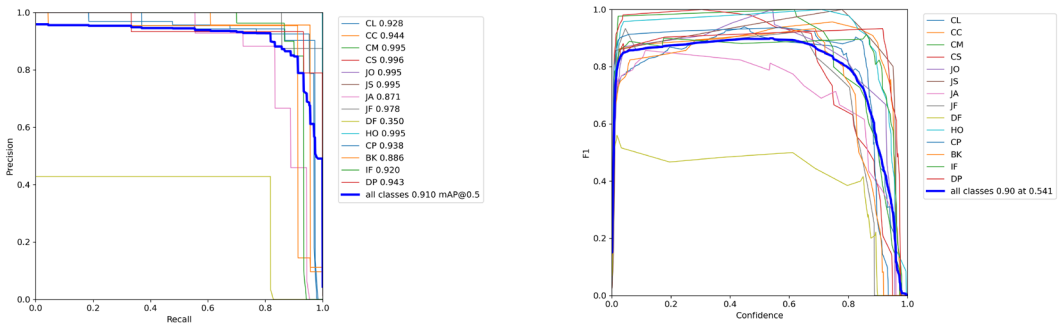
Model	Dataset	Backbone	Mean Average Precision, mAP (%)	Seconds per frame
SSD MobileNet	NCDOT 2017-2020	MobileNet	89.6	12
YOLOv5	NCDOT 2017-2020	DarkNet	87.5	0.89
Faster R-CNN	NCDOT 2017-2020	Regional Proposal Network	91.2	90



(a) SSD MobileNet (b) YOLOv5 (c)Faster R-CNN

Figure 22 Initial models accuracy and total loss curves

The YOLOv5 model was chosen and further developed. It was later swapped with YOLOv7 model which was then trained with all 16 classes and presents the best performance. The performance curves and the confusion matrix of the final model are shown in the following Figures 23 and 24.



(a) Mean average precision (mAP) (b) F1 score curve

Figure 23 Final YOLOv7 model performance curves

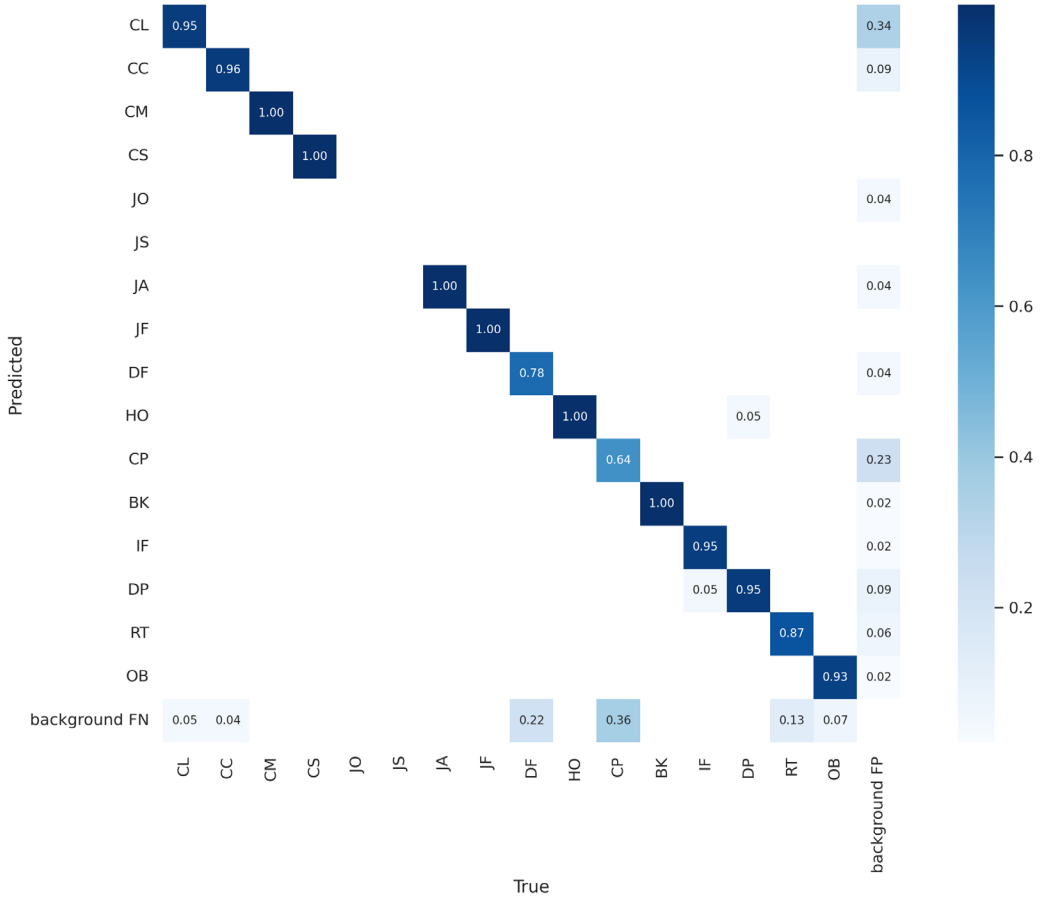
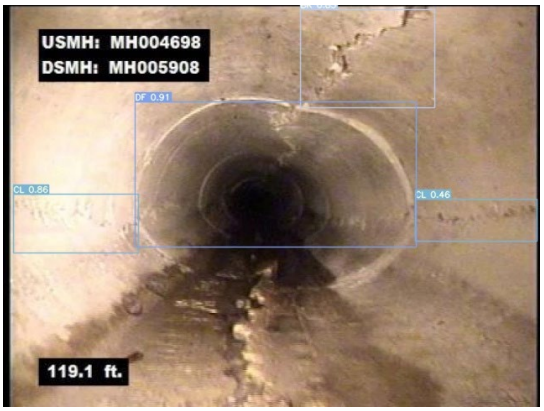


Figure 24 Final YOLOv7 model confusion matrix

Applying the model on images taken from the NDOT inspection videos also shows the accuracy of the detection in Figure .



(a) Longitudinal cracks and deformation

(b) Joint offset

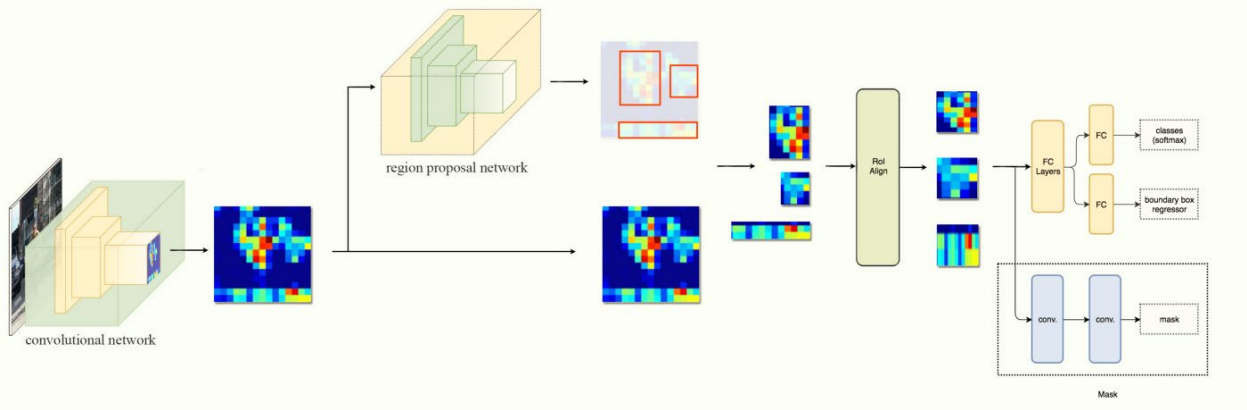
(c) deposits

(d) deposits and holes/dents

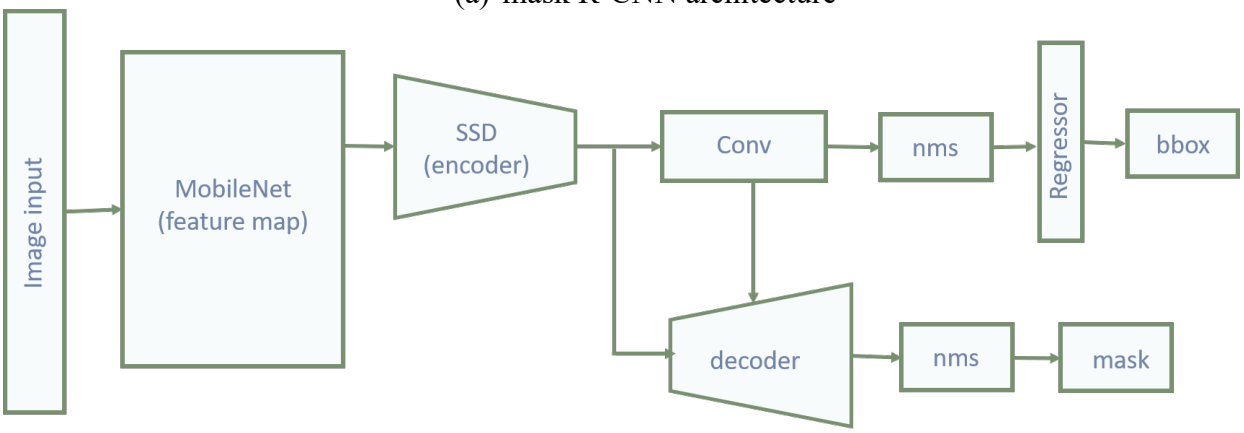
Figure 25 Structural defects detected by final YOLOv7 model

Defect quantification

After locating the defects, the next step is to quantify them. For this purpose, two deep learning models were developed and compared. The first being Mask R-CNN with a region proposed network (RPN) backbone and the second being Mask SSD with a MobileNet backbone. The architecture of both models is shown below in Figure 26.



(a) mask R-CNN architecture



(b) mask SSD architecture

Figure 26 Segmentation models architecture

The mask R-CNN model uses ROI align, which makes it ideal for close-up image analysis and quantification of defects. The masks resulting from the ROI align are shown below in Figure 27.

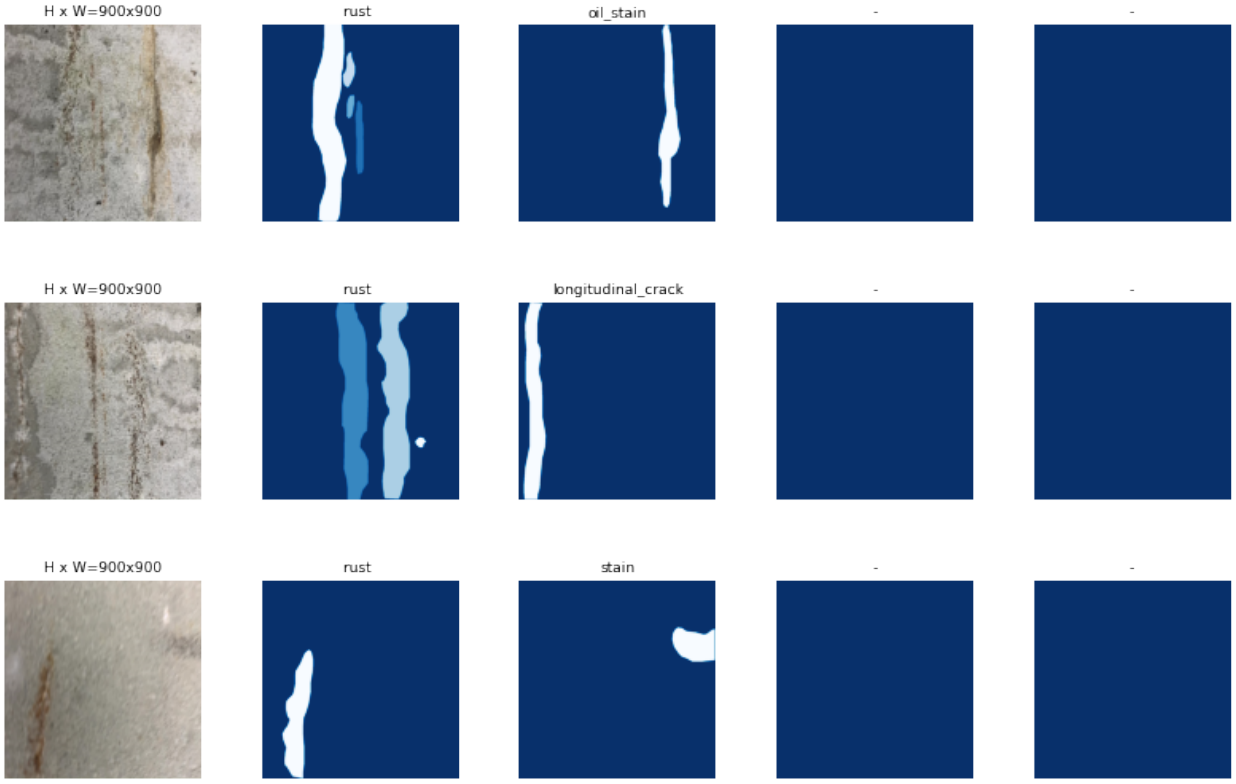


Figure 27 Generated mask obtained from ROI align

Both models were compares and the difference between them is shown in Table .

Table 9 Comparison of performance of two different instance segmentation techniques

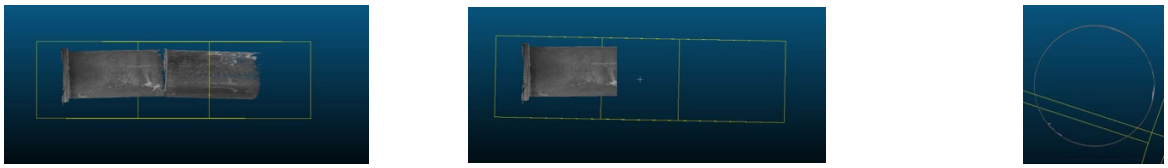
Model	**Mean Average Precision (mAP)**	**Inference time in milliseconds (ms)**
Mask SSD	0.193	~167
Mask R-CNN	0.320	~400

Task 3: Collecting Data from Sensor Networks

In this task, the high-quality 3D point cloud is presented with the pipe circularity calculation and the SLAM model which is sued as another method for mapping the pipe and tracking the robot location.

3D point cloud segmentation

The Artec Ray 3D scanning technology enables the generation of highly detailed 3D point clouds. With this rich dataset, we can perform segmentation techniques to extract specific features from the point cloud, such as obtaining cross-sectional views of pipes for circularity analysis. By segmenting the point cloud, we isolate the region representing the pipe, allowing us to extract a cross-sectional profile. This profile provides valuable information about the shape and geometry of the pipe at a specific location. We have used the Artec Ray 3D scanner to get a detailed 3D point cloud for the pipe we have in the structural lab to get the cross section. Figure 28.



(a) 3D point cloud mdoel (b) Point cloud segmentation (c) Cross-section

Figure 28 3D point cloud cross-section extraction

Pipe circularity

To assess circularity, we fit ellipses to the extracted cross-sectional contour of the pipe. The ellipses provide an approximation of the circular shape, and their parameters can be used to derive important data for determining the circularity of the pipe. The process is shown in Figure .

(a) Pipe cross-section (b) Filtering the binary image (c) Fitting curves to the detected shapes

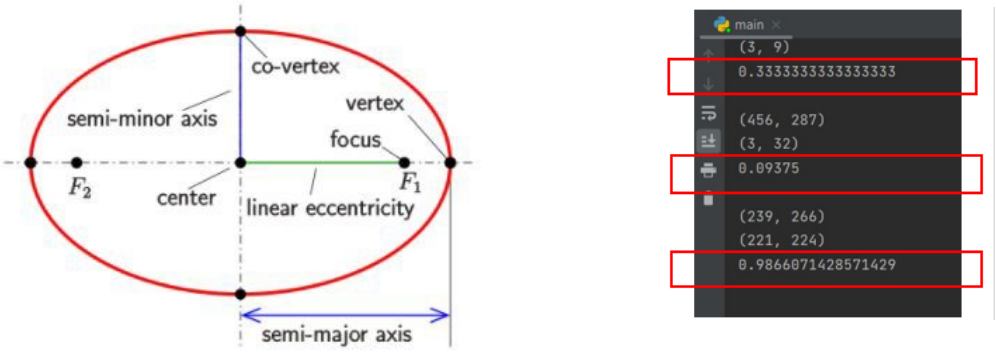
Figure 29 The process of fitting the ellipse of the cross section

By analyzing the ellipse data, we can extract relevant measurements, such as the major and minor axes lengths, eccentricity, and position of the ellipse center. These parameters allow us to quantitatively evaluate the circularity of the pipe, providing insights into its deviation

from a perfect circle. An ellipse is fitted on every shape in the final figure shown in Figure c and then we choose the most appropriate on from the list based on the circularity value calculated for every fitted ellipse. The ellipse circularity is calculated by dividing the length of the semi-minor axis and the semi-major axis. The equation is as follows:

$$Circularity: \qquad 0 \leq \frac{semi-minor\ axis}{semi-major\ axis} \leq 1$$

The calculated circularity values are then shown and we choose the one most appropriate. The process is shown in Figure .



(a) Ellipse elements (b) Calculated circularities

Figure 30 Circularity calculation

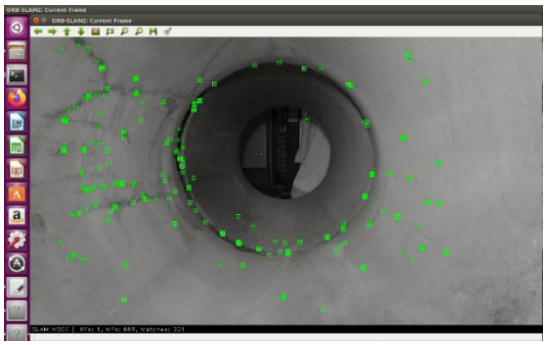
Pipeline SLAM

In addition to the previously presented mapping algorithm in Task 1, another model was developed. Which uses the same lidar for defect detection (LiDAR Camera L515) to map the pipe. This SLAM model is deployed using the Robot Operating System (ROS) of Ubuntu 16.04 operating system. The developed SLAM model is compared to the existing ORB-SLAM2 to evaluate effectiveness in terms of relative pose error (RPE). SLAM models are used to construct a 3D map of the pipe and to detect features in it. Both the existing and the proposed model are compared in Table 10.

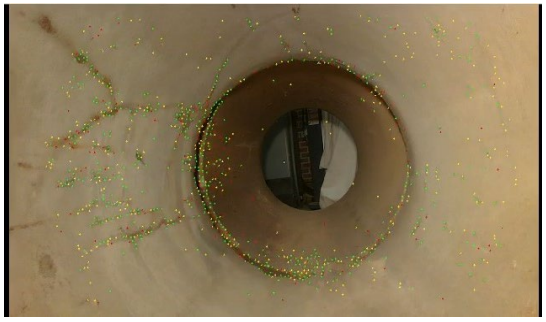
Table 10 Absolute trajectory error comparison of proposed and existing SLAM models

	Error Estimation			
Framework	**RMSE (m)**	**Mean (m)**	**RPE Translation (m)**	**RPE Rotation (m)**
ORB-SLAM2	0.151	0.155	0.056	0.089
Proposed System	0.137	0.123	0.042	0.071

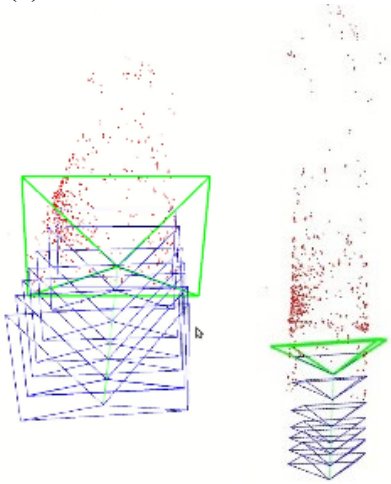
The proposed system has lower error and is better performing. The difference is further demonstrated in the results shown in Figure and the graphs in Figure .



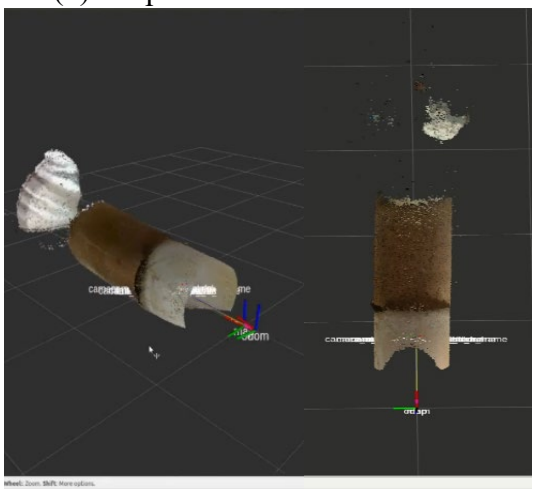
(a) ORB SLAM2 feature detection



(b) Proposed model feature detection

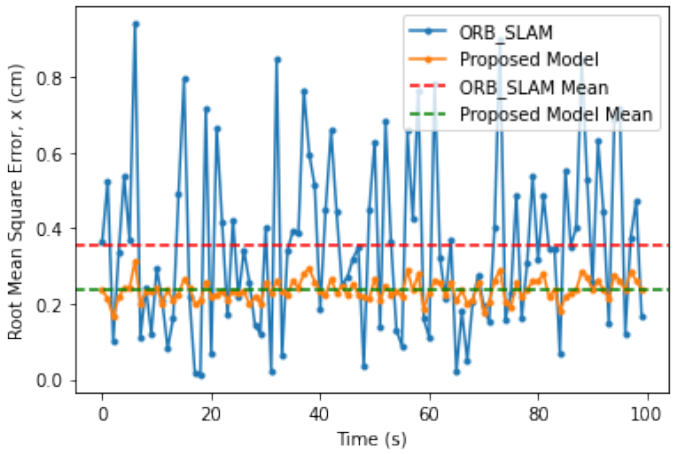


(c) Map and trajectory estimation of ORB SLAM2

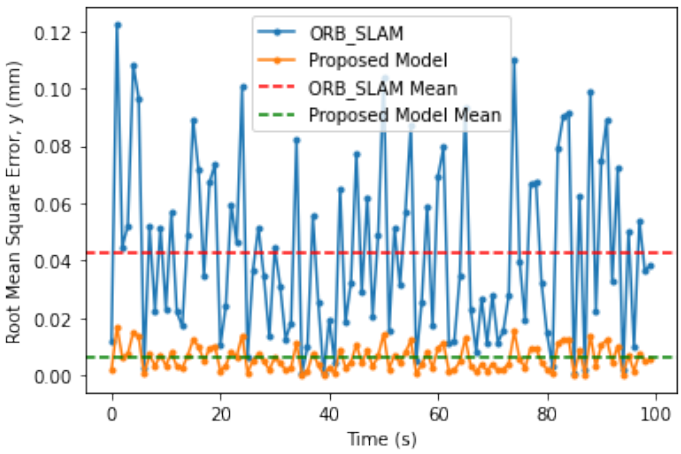


(d) Map and trajectory estimation of proposed model

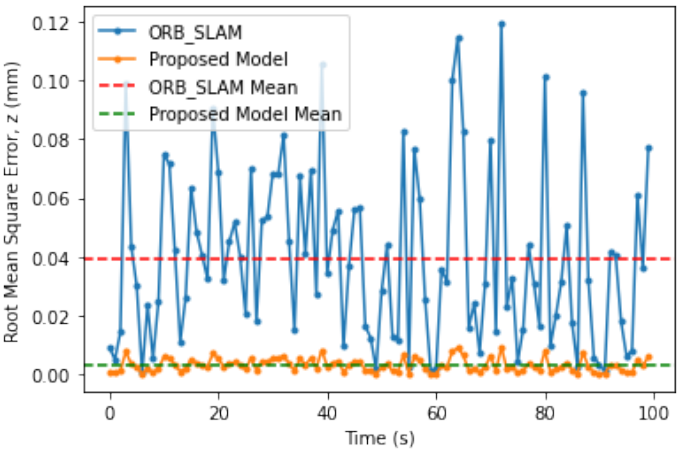
Figure 31 comparison of results from ORB SLAM2 and proposed SLAM models



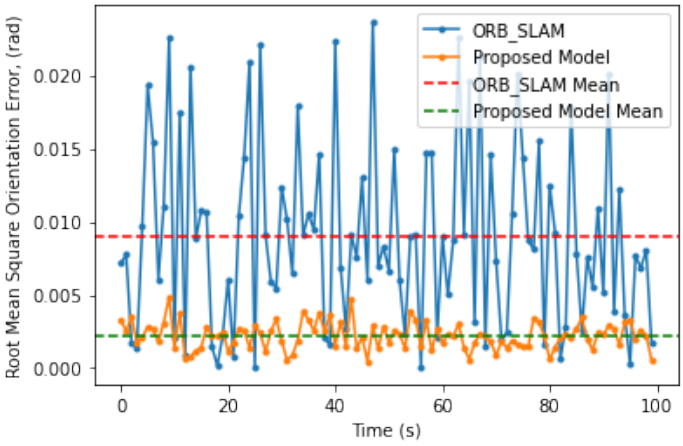
(a) Position x error



(b) Position y error



(c) Position z error



(d) Orientation error

Figure 32 ORB SLAM2 and proposed model errors comparison

Task 4: Data Reduction

Crack damage detection in industrial and civil constructions has long been an issue. Physical and operational inspections of sewer pipelines are essential for maintaining system serviceability. These networks are too old and nearing the end of their design life; in the meantime, rising environmental and health standards, rising demands, and constrained budgets have all made the problem more difficult to solve. Both manual and machine detection methods are used in traditional crack detecting technology. The manual detection method is widely regarded as less accurate and time-consuming. Closed-circuit television (CCTV) inspections of municipal sewer pipelines are known to be time-consuming, costly, and prone to errors, largely owing to operator fatigue or low-level experience. Automated defect detection can be a useful tool for maintaining the quality, accuracy, and consistency of condition data while also lowering inspection time and expense. New methods of autonomous inspection are proposed. This technology uses smart robots and sensors equipped with artificial intelligence-based techniques to access areas that are difficult for humans to reach, reduce decision makers discrepancy and the stress of analyzing the video and focus on more critical analysis. During the pipe condition assessment, video files, pictures, and other scans like LiDAR and 3D are used to acquire a lot of data. This becomes cumbersome to interpret and make meaning out of. In the context of human decision-making, we aim at reducing and reordering the data to show to the human inspectors to enhance their decision performance. The proposed method is based on computer vision and pattern recognition algorithms that have proven to improve the proficiency of the inspection process in identifying and locating important features and defects. This part of the project aims to extract the key information from this huge data and present the data to decision-makers or inspectors to make the decision-making process seamless and cognitively less challenging.

INTRODUCTION

Sewer condition evaluation is crucial for urban asset management since system failure has a major impact on both municipalities and users. With a limited recurring budget, the assessment offers the foundation for optimizing the restoration plan. A complete structural inspection of the sewer pipes is required to provide a thorough condition evaluation [49]. Detecting crack damage in industrial and civil structures has long been a problem. Inspections of sewer pipelines, both physical and operational, are required to keep the system working. Because the presence of a crack reduces the value of civil infrastructure, it is vital to determine the severity of the crack. Crack detection and classification techniques combined with quantitative analysis are essential for determining the severity of a crack. The length, width, and area are the different quantitative measures. The number of images acquired for analysis is rapidly increasing due to rapid technological

advancements. As a result, civil infrastructure requires autonomous crack detection and classification algorithms.

A crack is the whole or partial separation of two or more concrete components caused by fracturing or breaking. Cracks can form on surfaces such as buildings, bridges, roads, pavements, railway tracks, automobiles, tunnels, and aircraft. The two types of cracks that might be identified are active and dormant cracks. The direction, width, and depth of active cracks change with time, whereas the direction, width, and depth of dormant cracks remain constant. Both active and dormant fractures allow moisture to enter, which can cause further damage if not fixed. Active cracks include longitudinal cracks, transverse cracks, crocodile cracks, and reflection cracks.

There are different codes to categorize these cracks depending on the features and properties. According to the pipeline assessment certification program (PACP) guidelines, once a crack is deemed critical it is categorized into five distinct classes. These categories are Longitudinal, Circumferential, Multiple, Spiral, and Hinge crack. The category is majorly based on the crack run along or across the length of the pipe, the position, or a combination of these cases. For example, Crack Longitudinal (CL) runs along the length of the pipe while Crack Circumferential (CC) is transverse to the length of the pipe.

Crack detection is an image processing technique for automatically detecting a crack in an image. Image processing techniques include segmentation, morphological operations, Sobel edge detection, cannel edge detection, Otsu's method, gradient method, clustering method, least square method, histogram equalization method, particle filter, maximum entropy method, wiener filter, and wavelet transform.

The main goal is to make visual inspection more objective and also promote data-driven asset management. However, there are several obstacles to achieving this goal. Data collection and processing is a huge problem. For every inspection, video data is collected which is cumbersome and time-consuming to analyze. Inspection video data analysis is challenging. This data can be categorized as Big data because of the volume, velocity, variety, and veracity. A standard RGB data collects up to 108k images per hour, and with a robotic inspection system, data collection is done with fast speed. The video data also has different variations, cluttered background, and noise which can affect the data quality. Letting inspectors watch inspection videos for hours and days to locate the desired regions of interest is low in efficiency and easily triggers the development of fatigue. Therefore, it is important to employ data reduction techniques to make the decision-making process seamless.

In the present work, an image processing model that automatically detects cracks and other types of defects such as rust and void in concrete pipes from inspection video obtained from autonomous robot navigation. The proposed model also extracts key features like the

width, length, depth, and type of the crack and presents the information visually in an understandable manner. Further, post-processing is also done to reduce the data being processed and stored by identifying the location and type of crack on the pipe geometry in a simplified manner.

METHODS

The proposed method aims to enhance the proficiency of the inspection process for identifying and locating important features and defects in underground sewer pipes, leveraging computer vision and pattern recognition algorithms. Specifically, the YOLOv3 (You Only Look Once version 3) model is employed for crack detection due to its proven effectiveness in object detection tasks. This section provides a detailed overview of the chosen algorithm, its pre-processing requirements, and the key steps involved in the detection process.

YOLOv3 Model Selection:

The YOLOv3 algorithm is a deep learning-based approach that has demonstrated remarkable performance in object detection tasks. Its ability to detect multiple objects in real-time with high accuracy makes it well-suited for sewer pipe inspection. By selecting the YOLOv3 model as the core algorithm, our method aims to leverage its powerful capabilities to identify and locate cracks within the sewer pipe system.

Pre-processing of Data:

Prior to feeding the data into the YOLOv3 model, certain pre-processing steps are required to ensure optimal performance. These steps typically involve image preparation and normalization. The input images are typically resized to a standard size to ensure consistency and facilitate efficient processing. Additionally, normalization techniques such as mean subtraction or scaling may be applied to enhance the model's ability to detect cracks under varying lighting conditions.

YOLOv3 Architecture:

The YOLOv3 algorithm adopts a deep convolutional architecture known as darknet as its feature extraction network. The darknet architecture comprises multiple layers of convolutional neural networks (CNNs) that learn hierarchical features from the input image. These learned features are then used to predict bounding boxes and class probabilities. Notably, YOLOv3 employs anchor boxes, which are pre-defined dimension clusters, to predict the bounding boxes of objects within the image.

Bounding Box Regression and Confidence Scores:

The YOLOv3 algorithm outputs four coordinate values (tx, ty, tw, th) for each predicted bounding box, representing the coordinates of the box's top-left corner, its width, and its height, respectively. These values are obtained through regression operations. Additionally, confidence scores are generated to indicate the precision of the predicted

bounding box. A higher confidence score suggests a higher likelihood of the grid containing an object. Furthermore, class probabilities are assigned to each predicted bounding box, enabling the identification of different types of cracks.

Object Detection and Localization:
The YOLOv3 algorithm employs a grid-based approach to detect and localize objects. By dividing the input image into a grid, the algorithm predicts bounding boxes for each grid cell. The size of the predicted bounding box is adjusted based on the anchor boxes, which provide prior knowledge about the expected dimensions of objects. The YOLOv3 algorithm then applies non-maximum suppression to remove redundant bounding box predictions and retain the most accurate and relevant ones.

In summary, the proposed method utilizes the YOLOv3 algorithm as a computer vision and pattern recognition tool for crack detection in underground sewer pipes. Through appropriate pre-processing of the data, the YOLOv3 model is trained to identify cracks by predicting bounding boxes and assigning confidence scores. By leveraging the power of deep convolutional networks and anchor boxes, the algorithm exhibits remarkable performance in accurately detecting and localizing cracks, thereby improving the efficiency and effectiveness of sewer pipe inspection processes. The architecture of the proposed method is shown in Figure .

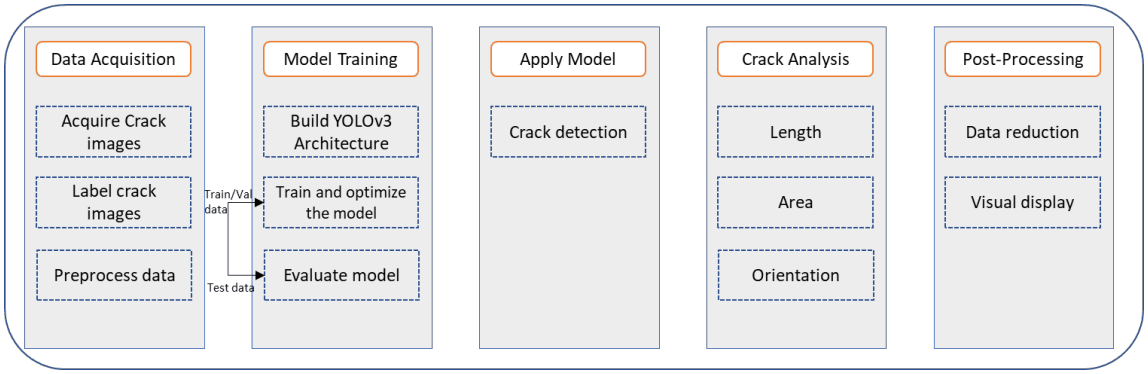


Figure 33 Overall architecture of the proposed method

Data Acquisition:
The data in this experiment are based on images collected from a test concrete pipe. The system proposed is a custom system, hence it requires using custom images peculiar to the inspection situation to reduce the complexity of modeling and training. The concrete pipe was scanned to inspect the internal condition and identify the presence of a crack. The video from the inspection was split into image frames. A total of 205 images were obtained. Three defect types (crack, void, and rust) were identified and correctly annotated using the LabelIMG software as part of the image pre-processing step in the deep learning modeling. The picture resolution is (480 * 848) pixels in the format of JPG. A sample of the ground truth data is shown in Figure .

The original collected defect images are only 205. For deep-learning tasks, to obtain a model with good generalization performance and high accuracy, this number of images data is not enough. The resolution of the original pictures is 480× 848 pixels, and the pixels are large and not suitable for direct input to the neural network. The input image size of the neural network is too large, which will cause too much memory overhead. The number of layers of the neural network is bound to increase. The parameters of the neural network also increase exponentially, which leads to an increase in video memory and training time and a decrease in training batches, making the model show poor performance. To solve these problems, the experiments were performed with the sliding window cropping technique, which uses 416× 416-pixel window sliding on the original picture and cropping line by line without overlapping. The resolution of the cropped picture is 416× 416 pixels. To improve the generalization ability of the model, it is necessary to perform data augmentation of the crack pictures, including horizontal flip, vertical flip, rotation 90/180◦, random zoom aspect ratio, random cut, brightness, and saturation change. After augmentation, the total number of images increased to 348 images.

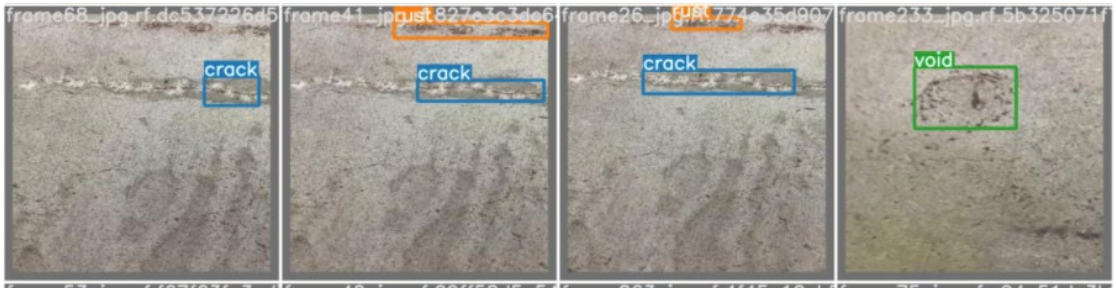


Figure 34 Ground truth data based on initial annotation

RESULTS & DISCUSSION

The proposed method was evaluated using a dataset comprising 348 original images of underground sewer pipes. Following the widely accepted 70/20/10% data splitting rule, the dataset was divided into training, validation, and test sets. Specifically, 244 samples were used for training, 70 samples were allocated for validation to assess the model's generalization performance, and an additional 34 samples were reserved as an independent test set. The training process was conducted for a duration of 0.484 hours over 200 epochs. During this training period, the model underwent iterative updates to improve its performance. Upon completion, the trained model achieved a precision rate of approximately 97.8% and a recall rate of 85.8%. The mean average precision, which measures the model's accuracy in detecting objects, reached a value of 78.6%. The graphs in Figure depicts the training process and the achieved model precision and recall.

The training process was carried out on the high-performance Google Colab computing platform. It has 1XTesla T4 GPU, 2496 CUDA cores, and 12GB GDDR5 VRAM.

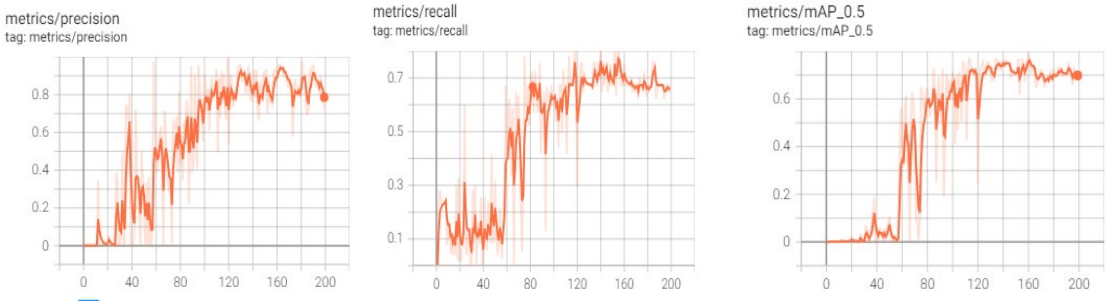


Figure 35 Model Precision and Recall

To evaluate the model's performance on previously unseen data, it was tested on images that were not included in the training or validation sets. The results of this testing phase are presented in Figure . Notably, the detection model demonstrated an impressive ability to correctly classify approximately 70% of the previously unseen data. In terms of precision and recall for specific classes, the model exhibited high performance in identifying voids and cracks. A summary of the results is provided in Table 11. The precision and recall rates for these classes were consistently robust. However, the individual performance for the rust class was comparatively lower. This discrepancy can be attributed to the limited number of rust images available for training. Increasing the quantity of rust images in the training dataset may enhance the model's performance on this class.

Table 11 Model Accuracy and Performance

Class	Images	targets	Precision	Recall	mAP@.5	mAP@.5:.95
All	31	34	0.723	0.673	0.693	0.349
Crack	31	13	0.769	0.769	0.742	0.267
Rust	31	4	0.435	0.25	0.342	0.212
Void	31	17	0.966	1	0.995	0.567

In summary, the trained detection model showcased its effectiveness in underground sewer pipe inspection. It achieved high precision and recall rates, particularly for voids and cracks, while the performance on the rust class indicated room for improvement due to the limited training data available. These findings underscore the potential of the proposed method for accurate and efficient identification of defects in sewer pipe systems, with the scope for further enhancements through the inclusion of more diverse training samples.

Figure 36 Predicted defect of test data

The model provides the percentage of confidence for the defect class it's predicting. The threshold is 40%, so if the prediction confidence is less than 40%, the model is not going to capture. We want the model to give a level of confidence that inspectors can rely on.

The recall is a very relevant metric for our application. We don't want to miss defects of any type because it could mean a catastrophic failure or even a sanitary overflow or even worse. The key is to train the algorithm to achieve a balance between recall and precision by not missing defects, but also not generating an inordinate amount of false positives. The model has an acceptable level of precision and recall to detect defects when present. Figure 36.

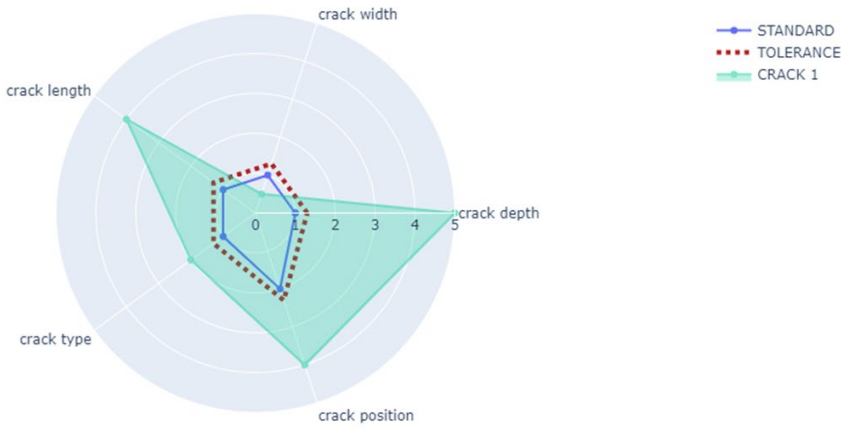


Figure 37 Radar chart for decision making

The next step is to extract information about the individual crack. The properties of the crack are represented on the radar chart with reference to a default standard. The crack depth, length, width, type, and position are important features according to the PACP guidelines. To make the decision-making easy for whoever is inspecting the pipe. The radar

chart is proposed because of its ability to present more than two attributes at a time in a meaningful way. A visual representation of the radar chart is shown in Figure 38.

Further, post-processing is also in works to reduce the data being processed and stored by identifying the location and type of crack on the pipe geometry in a simplified manner. So this way, the required details are captured without necessarily storing all the data collected. Figure 38 explains the idea of the post-processing phase in a basic manner. It shows a section of the pipe with the corresponding length and the defect type with the respective location in 2D. Any inspector can re-visit this, and easily pinpoint and make conclusions about the section. This will also help to track the history of the pipe, for example when it is required to see how the pipe has been functioning for the past months or years, it can be easily compared and reviewed.

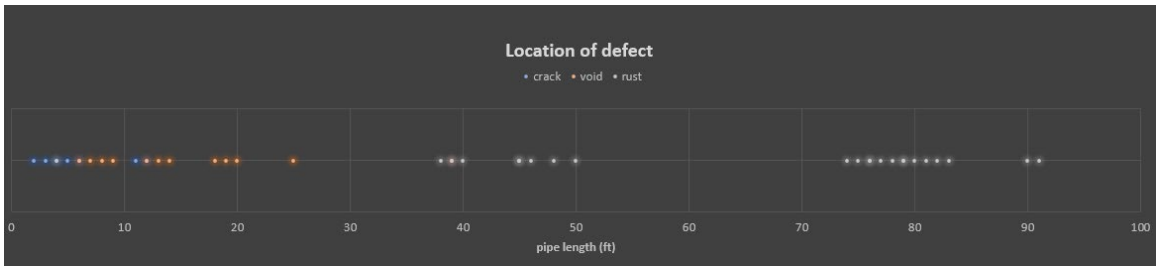


Figure 38 Post-processing representation of pipe profile

Field Demo

In this section, the field demo we performed is presented. The field demo was performed in 4253 Camp Burton Road, McLeansville, 27301-9255 on June 16th, 2023. The site is shown below in Figure 39.

Figure 39 Field demo site

In order to use the developed system, the computers were setup next to the inspection point, Figure 40.

Figure 40 Computers setup

At the time of the demo, a laptop had not yet been setup for using the inspection system so the main computers were used. The robot used for the demo was the husarion ROSbot and it was deployed in an 18 inch pipe. See Figure 41.

Figure 41 Husarion ROSbot positioned at pipe entrance with tether being connected

At that time, the wireless connection between the robot and the computers had not yet been setup so it was dependent on the tether. The results of the defect detection algorithm was shown to the NCDOT engineers present at the time as shown in Figure 42.

Figure 42 NCDOT engineers examining the results during the field day.

It was concluded from the field demo that the focus needed to be on setting up a laptop with the required algorithms and the wireless communication with the robots. This would greatly enhance the usability of the developed system.

Conclusions

In conclusion, this research has presented a holistic approach for automated detection and classification of sewer pipe defects using a deep convolutional neural network coupled with advanced navigation and mapping algorithms. While it is important to note that an automated visual inspection system cannot fully replace human inspection, it serves as a powerful tool to enhance the quality, objectivity, and feedback of pipe inspections. The system developed in this study leverages deep learning methods to accurately label defects and features in pipe inspection videos, thereby accelerating the production of condition reports and critical decision-making processes related to state infrastructure and assets. It also provides the framework for using robots that automatically maneuver and map the area inspected.

By automating the detection and classification of sewer pipe defects, the developed system offers several notable benefits. Firstly, it improves the accuracy of defect identification, reducing the likelihood of escapes and ensuring a higher level of quality in inspection outcomes. Secondly, the system enhances objectivity by removing human bias from the detection process, leading to more consistent and reliable results. Moreover, the system provides valuable data visualization tools, facilitating intuitive decision-making processes for infrastructure management professionals.

The implementation of this system is expected to yield significant long-term advantages. By reducing the reliance on manual inspection methods, it minimizes overall inspection costs and relieves stress on human inspectors. The increased efficiency in producing condition reports and making critical decisions enables timely maintenance interventions, contributing to the longevity and sustainability of sewer pipe systems.

Implementation and Technology Transfer Plan

A system developed for hydraulic concrete pipe inspection can be transferred to support NC DOT Material and Test Unit for underground concrete pipes inspection.

This includes training materials; hands-on training; field demonstration and software development and training. The training also will include robotic programing control, and image processing of 2D and 3D scans.

The system will include the following items as shown in Figure 43:

1. Pipe circularity evaluation; done from the 3D Scan of the Pipe.
2. LiDAR 3D point cloud, done with the 3D Lidar on a ground robot.
3. Defect classification; surface defect detection and capability; 3D Mapping.
4. The following hardware and software are transferred:
    a. A main control computer,
    b. Python-based Algorithm,
    c. A ground robot and 3D sensors including Intel RealSense D435 camera, Intel RealSense L515 camera, other necessary Lidar and position sensors,
    d. Robot control and data collection/processing algorithms.

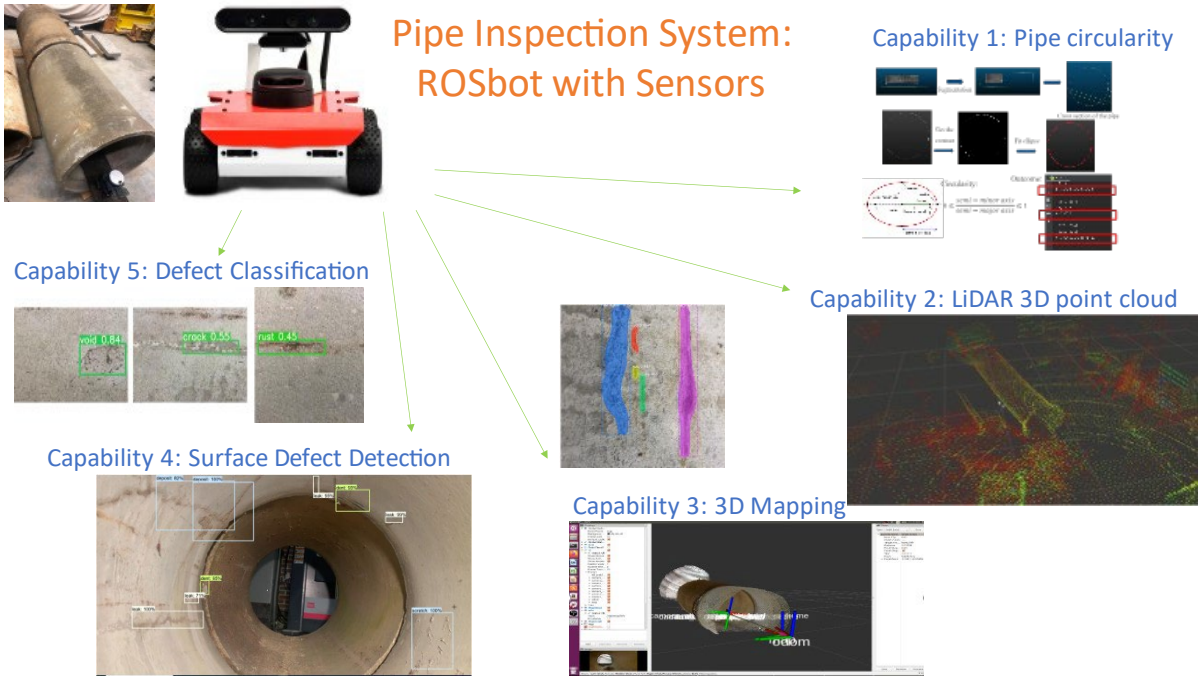


Figure 43 The proposed technology transfer system.

References

[1] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, "A deep learning-based framework for an automated defect detection system for sewer pipes," *Autom Constr*, vol. 109, Jan. 2020, doi: 10.1016/j.autcon.2019.102967.

[2] N.-D. Hoang and V.-D. Tran, "Image Processing-Based Detection of Pipe Corrosion Using Texture Analysis and Metaheuristic-Optimized Machine Learning Approach," *Comput Intell Neurosci*, vol. 2019, 2019, doi: 10.1155/2019/8097213.

[3] Z. Liu and Y. Kleiner, "State of the art review of inspection technologies for condition assessment of water pipes," *Measurement*, vol. 46, no. 1, pp. 1–15, 2013, doi: 10.1016/j.measurement.2012.05.032.

[4] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr, "Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks," *Autom Constr*, vol. 91, pp. 273–283, Jul. 2018, doi: 10.1016/j.autcon.2018.03.028.

[5] J. Dirksen *et al.*, "The consistency of visual sewer inspection data," *Structure and Infrastructure Engineering*, vol. 9, no. 3, pp. 214–228, Mar. 2013, doi: 10.1080/15732479.2010.541265.

[6] D. Meijer, L. Scholten, F. Clemens, and A. Knobbe, "A defect classification methodology for sewer image sets with convolutional neural networks," *Autom Constr*, vol. 104, pp. 281–298, Aug. 2019, doi: 10.1016/j.autcon.2019.04.013.

[7] J. Myrans, R. Everson, and Z. Kapelan, "Automated detection of faults in sewers using CCTV image sequences," *Autom Constr*, vol. 95, pp. 64–71, Nov. 2018, doi: 10.1016/j.autcon.2018.08.005.

[8] A. M. Albishi and O. M. Ramahi, "Surface crack detection in metallic materials using sensitive microwave-based sensors," in *2016 IEEE 17th Annual Wireless and Microwave Technology Conference (WAMICON)*, 2016, pp. 1–3. doi: 10.1109/WAMICON.2016.7483842.

[9] G. Lacidogna, G. Piana, F. Accornero, and A. Carpinteri, "Multi-technique damage monitoring of concrete beams: Acoustic Emission, Digital Image Correlation, Dynamic Identification," *Constr Build Mater*, vol. 242, May 2020, doi: 10.1016/j.conbuildmat.2020.118114.

[10] P. Liu *et al.*, "Development of a 'stick-and-detect' wireless sensor node for fatigue crack detection," *Struct Health Monit*, vol. 16, no. 2, pp. 153–163, Mar. 2017, doi: 10.1177/1475921716666532.

[11] S. Zhao, L. Sun, J. Gao, J. Wang, and Y. Shen, “Uniaxial ACFM detection system for metal crack size estimation using magnetic signature waveform analysis,” *Measurement (Lond)*, vol. 164, Nov. 2020, doi: 10.1016/j.measurement.2020.108090.

[12] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, “Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2718–2729, Oct. 2016, doi: 10.1109/TITS.2015.2477675.

[13] X.-Z. Yang and Z.-Q. Zhou, “Design of crack detection system,” in *Proceedings - 2017 International Conference on Network and Information Systems for Computers, ICNISC 2017*, Institute of Electrical and Electronics Engineers Inc., Apr. 2017, pp. 147–152. doi: 10.1109/ICNISC.2017.00039.

[14] X. Zhang, K. Wang, Y. Wang, Y. Shen, and H. Hu, “Rail crack detection using acoustic emission technique by joint optimization noise clustering and time window feature detection,” *Applied Acoustics*, vol. 160, Mar. 2020, doi: 10.1016/j.apacoust.2019.107141.

[15] M.-H. Cheon, D.-G. Hong, and D. Lee, “Surface Crack Detection in Concrete Structures Using Image Processing,” in *Robot Intelligence Technology and Applications 5: Results from the 5th International Conference on Robot Intelligence Technology and Applications*, Springer Verlag, 2019, pp. 559–564. doi: 10.1007/978-3-319-78452-6_45.

[16] W. Li, J. Huyan, S. L. Tighe, Q. Ren, and Z. Sun, “Three-Dimensional Pavement Crack Detection Algorithm Based on Two-Dimensional Empirical Mode Decomposition,” *Journal of Transportation Engineering, Part B: Pavements*, vol. 143, no. 2, p. 04017005, Jun. 2017, doi: 10.1061/jpeodx.0000006.

[17] H. Sun, Q. Liu, and L. Fang, “Research on Fatigue Crack Growth Detection of M (T) Specimen Based on Image Processing Technology,” *Journal of Failure Analysis and Prevention*, vol. 18, pp. 1010–1016, Aug. 2018, doi: 10.1007/s11668-018-0493-6.

[18] A. Tedeschi and F. Benedetto, “A real-time automatic pavement crack and pothole recognition system for mobile Android-based devices,” *Advanced Engineering Informatics*, vol. 32, pp. 11–25, Apr. 2017, doi: 10.1016/j.aei.2016.12.004.

[19] Y. Wang, Y. Huang, and W. Huang, “Crack junction detection in pavement image using correlation structure analysis and iterative tensor voting,” *IEEE Access*, vol. 7, pp. 138094–138109, 2019, doi: 10.1109/ACCESS.2019.2942318.

[20] X. Yuling and Z. Haoran, “Research on Surface Crack Detection Technology Based on Digital Image Processing,” in *Journal of Physics: Conference Series 1550*, IOP Publishing, Jun. 2020, p. 032012. doi: 10.1088/1742-6596/1550/3/032012.

[21] W. Shibing, Y. Fan, C. Yunmei, Y. Yuanyuan, and W. Yuzeng, “Adaboost-based Crack Detection Method for Pavement,” in *IOP Conf. Series: Earth and Environmental Science 189*, IOP Publishing, 2018, p. 022005. doi: 10.1088/1755-1315/189/2/022005.

[22] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J Comput Syst Sci*, vol. 55, pp. 119–139, 1997.

[23] M. Gaith, M. E. H. Assad, A. Sedaghat, M. Hiyasat, and S. Alkhatib, “Neural Network Usage in Structural Crack Detection,” in *Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management*, 2015.

[24] Z. Liu, Y. Cao, Y. Wang, and W. Wang, “Computer vision-based concrete crack detection using U-net fully convolutional networks,” *Autom Constr*, vol. 104, pp. 129–139, Aug. 2019, doi: 10.1016/j.autcon.2019.04.005.

[25] S. Sharma and N. K. Gupta, “A Genetic Approach to Segment and Detect Crack in Rail Track,” in *Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2019, pp. 707–712. doi: 10.1109/ICCMC.2019.8819757.

[26] N. A. M. Yusof *et al.*, “Automated Asphalt Pavement Crack Detection and Classification using Deep Convolution Neural Network,” in *Proceedings, 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2019): PARKROYAL Penang Resort Batu Ferringhi, Penang, Malaysia, 29 Nov 2019 -1 Dec 2019*, Penang, Malaysia, 2019, pp. 215–220.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25 (NIPS 2012)* , 2012. [Online]. Available: http://code.google.com/p/cuda-convnet/

[28] K. Simonyan and A. Zisserman, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION,” 2015. [Online]. Available: http://www.robots.ox.ac.uk/

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), 2016, pp. 770–778. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[30] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448. [Online]. Available: https://github.com/rbgirshick/

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)* , 2015. [Online]. Available: https://github.com/

[32] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science, vol. 9905. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-46448-0.

[33] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018. [Online]. Available: https://pjreddie.com/yolo/.

[34] H. Kim, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, "Comparative analysis of image binarization methods for crack identification in concrete structures," *Cem Concr Res*, vol. 99, pp. 53–61, Sep. 2017, doi: 10.1016/j.cemconres.2017.04.018.

[35] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, Apr. 2015, doi: 10.1016/j.aei.2015.01.008.

[36] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, Jun. 2018, doi: 10.1016/j.aej.2017.01.020.

[37] H. Zakeri, F. M. Nejad, and A. Fahimifar, "Image Based Techniques for Crack Detection, Classification and Quantification in Asphalt Pavement: A Review," *Archives of Computational Methods in Engineering*, vol. 24, pp. 935–977, Nov. 2017, doi: 10.1007/s11831-016-9194-z.

[38] Y. Zong *et al.*, "An intelligent and automated 3D surface defect detection system for quantitative 3D estimation and feature classification of material surface defects," *Opt Lasers Eng*, vol. 144, Sep. 2021, doi: 10.1016/j.optlaseng.2021.106633.

[39] M. Der Yang and T. C. Su, "Automated diagnosis of sewer pipe defects based on machine learning approaches," *Expert Syst Appl*, vol. 35, no. 3, pp. 1327–1337, Oct. 2008, doi: 10.1016/j.eswa.2007.08.013.

[40] O. Moselhi and T. Shehab-Eldeen, “Automated detection of surface defects in water and sewer pipes,” *Autom Constr*, vol. 8, no. 5, pp. 581–588, 1999, doi: 10.1016/S0926-5805(99)00007-2.

[41] O. Moselhi and T. Shehab-Eldeen, “CLASSIFICATION OF DEFECTS IN SEWER PIPES USING NEURAL NETWORKS,” *Journal of Infrastructure Systems*, vol. 6, no. 3, 2000, doi: 10.1061/(ASCE)1076-0342(2000)6:3(97).

[42] M. Wang, S. S. Kumar, and J. C. P. Cheng, “Automated sewer pipe defect tracking in CCTV videos based on defect detection and metric learning,” *Autom Constr*, vol. 121, p. 103438, Jan. 2021, doi: 10.1016/j.autcon.2020.103438.

[43] S. Moradi, T. Zayed, F. Nasiri, and F. Golkhoo, “Automated Anomaly Detection and Localization in Sewer Inspection Videos Using Proportional Data Modeling and Deep Learning–Based Text Recognition,” *Journal of Infrastructure Systems*, vol. 26, no. 3, p. 04020018, Sep. 2020, doi: 10.1061/(asce)is.1943-555x.0000553.

[44] M. J. Chae and D. M. Abraham, “NEURO-FUZZY APPROACHES FOR SANITARY SEWER PIPELINE CONDITION ASSESSMENT,” *Journal of Computing in Civil Engineering*, vol. 15, no. 1, pp. 4–14, 2001, doi: 10.1061/(ASCE)0887-3801(2001)15:1(4).

[45] H. Noh, S. Hong, and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” in *IEEE International Conference on Computer Vision (ICCV)*, IEEE, Dec. 2015, pp. 1520–1528. doi: 10.1109/ICCV.2015.178.

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[47] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[48] G. Wen *et al.*, “YOLOv5s-CA: A Modified YOLOv5s Network with Coordinate Attention for Underwater Target Detection,” *Sensors*, vol. 23, no. 7, p. 3367, Apr. 2023, doi: 10.3390/s23073367.

[49] D.-H. Koo and S. T. Ariaratnam, “Innovative method for assessment of underground sewer pipe condition,” *Autom Constr*, vol. 15, no. 4, pp. 479–488, Jul. 2006, doi: 10.1016/j.autcon.2005.06.007.