IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 August 2024, accepted 21 August 2024, date of publication 27 August 2024, date of current version 4 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3450502

RESEARCH ARTICLE

Smart Pipe Inspection Robot With In-Chassis Motor Actuation Design and Integrated AI-Powered Defect Detection System

DARKHAN ZHOLTAYEV[1], DANIYAR DAULETIYA[1], AISULU TILEUKULOVA[2], DIAS AKIMBAY[3], MANAT NURSULTAN[3], YERSAIYN BUSHANOV[3], ASKAT KUZDEUOV[4], AND AZAMAT YESHMUKHAMETOV[3,4]

[1]Department of Computational and Data Science, Astana IT University, 020000 Astana, Kazakhstan
[2]Department of Physics and Technology, Al-Farabi Kazakh National University, 050040 Almaty, Kazakhstan
[3]Department of Robotics and Mechatronics, School of Engineering and Digital Sciences, Nazarbayev University, 010000 Astana, Kazakhstan
[4]Institute of Smart Systems and Artificial Intelligence, Nazarbayev University, 010000 Astana, Kazakhstan

Corresponding authors: Azamat Yeshmukhametov (azamat.yeshmukhametov@nu.edu.kz) and Aisulu Tileukulova (aisulu.tileukulova@gmail.com)

This work was supported by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan under Grant AP19679380.

ABSTRACT In the contemporary world, inspection operations have become a critical component of infrastructure maintenance. Over the years, the demand for comprehensive inspection of pipes, both internally and externally, has grown increasingly complex and challenging. Consequently, there is a pressing need for significant advancements in in-pipe robots, particularly in the areas of inspection speed, defect detection precision, and overall reliability. Recent developments in new devices and sensors have markedly improved our capability to inspect and diagnose defects within pipes with greater accuracy. Furthermore, the application of machine learning tools has optimized the inspection process, enhancing the detection and recognition of potential pipe defects, such as rust, blockages, and welding anomalies. This research introduces a novel mobile robot platform specifically designed for pipe inspection. It integrates an advanced machine learning model that effectively detects and identifies key pipe defects, including rust, compromised welding quality, and pipe deformation. Additionally, this platform offers enhancements in inspection speed. The integration of these technologies represents a significant stride in the field of infrastructure maintenance, setting a new standard for efficiency and precision in pipe inspection.

INDEX TERMS Inpipe robot, robot design, machine learning, defect detection, pipe inspection, SLAM.

I. INTRODUCTION

Pipe inspection robots have emerged as a pivotal element in ensuring high-quality maintenance and preventing emergency scenarios in daily life. Over the past few decades, researchers and engineers have developed a diverse range of pipe inspection robots, tailored to suit their specific applications. These designs include various types of in-pipe robots: wheel-driven [1], caterpillar-type [2], Pipeline Inspection Gauge (PIG) robots [3], screw-driven, walking robots [4], and inchworm-type robots [5]. Each type offers unique capabilities and advantages, reflecting the evolving landscape of robotic intervention in maintenance and safety [6], [7], [8].

The design and functionality of in-pipe robots are highly contingent upon their intended applications. For instance, Pipeline Inspection Gauge (PIG) and inchworm robots are particularly adept for cleaning tasks. These robots are driven by high pressure and are capable of inspecting pipe structures in offline mode. Conversely, for inspection purposes, wheel-based and track-based robots are more appropriate. These robots are equipped with actuators and can provide real-time online streaming. Furthermore, wheel and track-based robots can be enhanced with passive extension mechanisms. This adaptation allows them to achieve improved traction with

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

the pipe's surface and facilitates smoother movement within pipes of varying diameters.

A. PROBLEM STATEMENT

Pipelines are the primary method of transportation of oil and gas due to their cost and effectiveness [9]. Not only that but they are used in canalization and water and heat supply systems in the cities [10]. Kazakhstan is very rich in oil and gas reserves, it possesses 3 trillion cubic meters of gas reserves, with an estimated 5 trillion cubic meters remaining [11]. Several oil cites such as Kashagan and Tengiz had accidents over the past years [12]. This could have been prevented with regular and thorough inspection of the pipelines. In recent years, the development of IPIRs (in-pipe inspection robots) has become very popular [13]. We suggest developing a Smart In-Pipe Inspection Robot (SIPIR), which uses machine learning as well as SLAM methods in order to identify defects within the pipe.

B. RELATED RESEARCH PAPERS

Computer vision has become an integral part of various fields, leveraging prominent models to achieve diverse objectives. The application of computer vision in marketing involves the use of pre-trained models such as YOLOV2, Google Cloud Vision, and Clarifai to analyze brand-related user-generated content [14].

In the realm of industrial maintenance, the application of computer vision models has become pivotal in automating the inspection of pipe corrosion. This is traditionally achieved through the analysis of video streams captured by cameras. However, conventional computer vision techniques often encounter limitations in accurately classifying corrosion [15], [16], primarily due to its inherent variability in color, shape, and size, which further fluctuates based on the specific type of corrosion encountered. To address these challenges, deep learning-based approaches have emerged as a robust alternative to computer vision techniques [17], but first, machine learning-based algorithms were mostly implemented, as stated in [18]. Notably, CNN architectures have demonstrated exceptional efficacy across various domains, as substantiated by [19], who highlight the remarkable capabilities of CNNs in visual recognition tasks for external inspection pipes.

The potential of computer vision for defect detection in sewer pipelines was reviewed in [20] emphasizing different CNN-based algorithms like faster region-based convolutional neural networks (Faster-RCNN), conventional CNNs, and discussions on deterioration models for predicting sewer pipe conditions, highlighting computer vision's role in preventing corrosion, was presented [21], where mostly machine learning-based algorithms are discussed. On the other hand, a CV for pipe inspection was developed using architecture; you only look once the (YOLO)-V3 model is used in [22]. Other works on corrosion detection using magnetic-flux leakage, which can be considered a robust method but might struggle to detect in the early stages, were also mentioned [23], [24]. Survey and dataset benchmarks of different computer vision models are discussed as YOLO, Single Shot Multibox Detector (SSD), Faster R-CNN, PipeUNet models, and different ResNet modification models in sewer pipe inspection [25], [26].

MobileNet's efficiency on devices with limited processing power makes it useful for real-time applications and those involving mobile or embedded devices [27], [28], [29], [30]. Its adaptable architecture meets demands in human-interactive applications and is beneficial in healthcare, infrastructure, and environmental monitoring [31], [32], [33]. Furthermore, MobileNetv2's lightweight design is ideal for resource-constrained environments, balancing speed and accuracy in autonomous vehicles, surveillance, and industrial automation [29], [30], [32], [34]. Its effectiveness in object detection and classification is also acknowledged [35], [36].

Moreover, its effectiveness in object detection and classification is acknowledged in [35] and in [36]. Edge computing capability is essential for mobile robots, particularly for real-time detection. In this context, MobileNetv2, with its 3.4 million parameters [37], offers a significant advantage in terms of computational lightness compared to other reviewed models such as YOLO v5 (7.2 mln - 86.7 mln) [38], SSD (VGG16-based) (138 mln) [39], Faster R-CNN (19 mln - 41 mln) [40], ResNet50 (25 million) [41], and Pipe UNet (30 million) [42] in terms of parameter count.

Anomaly detection is another problem that was considered in this work. In the pipe, there may be sediments of soil or some instruments left during the pipe construction process. To prevent damage caused by the left instrument when liquid is released at high pressure, it is important to recognize it before exploitation. For this purpose, an anomaly detection solution is implemented using a Canny Edge detector [43] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [44].

The Canny edge detector is a widely used algorithm for edge detection due to its superior performance [43]. It is commonly employed in various applications such as bolt looseness detection [45], satellite component contour extraction [46], exterior wall hollowing detection [47], and ore and rock edge detection [48]. The Canny Edge detector is known for its ability to extract candidate contour points efficiently, making it suitable for subsequent processing with convolutional neural network models [49].

The Canny Edge detector has been the subject of research for optimization and improvement. For instance, a study aimed to enhance edge detection using SR-guided threshold maneuvering and window mapping, addressing broken edges and noisy structures in Canny edges [50]. Another piece of research focused on improving the robustness of edge detection against noise by preceding the Canny edge detector with a new type of denoising system [51]. Moreover, the Canny edge detector has been applied in the context of corner detection algorithms, where it was found to be a better method for extracting features in an image without disturbing its features [52].

DBSCAN is a widely used algorithm in various fields due to its ability to handle noise and arbitrary cluster shapes without the need to pre-determine the total clusters [44]. It is a density-based clustering algorithm that can discard points that are not in locally dense regions, making it suitable for identifying clusters with arbitrary shapes and sizes [53]. DBSCAN has been applied in diverse areas such as anomaly detection [54], indoor localization [55], maritime traffic pattern analysis [56], and risk assessment in railway investment [57].

The algorithm is particularly favored for its ability to identify clusters with any arbitrary shape and size, which is a limitation in other clustering algorithms such as K-means [58]. Furthermore, DBSCAN has been used in conjunction with other techniques, such as the use of RGB images in indoor localization [55] and the integration with three distances for Wi-Fi positioning algorithms [59]. In addition, the algorithm has been adapted and improved, leading to the development of variations such as Hierarchical DBSCAN (HDBSCAN) [60] and KR-DBSCAN [61], which further extend its applicability and effectiveness in different contexts.

In summary, DBSCAN is a powerful and versatile algorithm that has been widely adopted in various domains due to its ability to handle noise, identify clusters with arbitrary shapes and sizes, and adapt to different applications through enhancements and variations. The proposed pipe inspection robot has a wide range of applications, including inspection of oil and gas pipelines, as well as urban water distribution systems. Various solutions have been developed to address different inspection challenges. One notable approach is the collective gas detection technique, which effectively identifies sources of gas leakage or crack sources in pipelines, as demonstrated by Rohrich et al. [62]. However, this work specifically focuses on computer vision-based solutions for detecting and examining various types of defects in pipelines.

Simultaneous Localization and Mapping (SLAM) is crucial for autonomous inspection robots in water and sewer pipes, utilizing sensors like optical, acoustic, and inertial for navigation and fault detection. SLAM faces challenges such as the absence of GPS and environmental obstacles, but enhancements with prior map knowledge and data fusion techniques improve its robustness. Accurate defect localization is vital to minimize disruption and align with industry standards, employing visual odometry, visual SLAM, and Lidar to ensure precise mapping [63]. Hybrid maps and various sensors further optimize path planning and defect localization in pipe networks. Global climate change has heightened the need for resilient infrastructure, particularly for underground pipelines essential for drainage, sewage, and gas transport. Traditional manual inspections of these pipelines are laborious and risky, leading to the development of robotic systems that use advanced technologies like CNNs for defect detection. A proposed low-cost inspection platform uses infrared and depth cameras, a g-sensor, and deep learning to enhance defect recognition and positioning accuracy [64]. This platform demonstrates effective positioning, defect identification, and documentation, offering a safer and more cost-effective alternative to manual inspections, with future improvements aimed at enhancing sensor integration and training data diversity. The critical role of water mains in urban infrastructure necessitates efficient and non-disruptive inspection methods. The GAVIA AUV, equipped with nondestructive testing (NDT) sensors, uses a vision-based simultaneous localization and mapping (VSLAM) approach to enhance localization and mapping capabilities for pipe defect detection [65]. This VSLAM method integrates CCD cameras, inertial navigation sensors (INS), and range-finders to generate detailed, real-time mapping of pipe interiors. Experiments using a simulated pipe inspection setup demonstrated the advantages of combining camera data with INS measurements, highlighting the potential for improved AUV performance through optimized sensor fusion. Various SLAM algorithms have been developed to address the challenges of in-pipe robots, integrating computational geometry and computer vision concepts. Notable methods include PipeSLAM, which uses a Rao-Blackwellised particle filter and hydrophone-induced vibrations, and multi-sensor fusion systems that combine rangefinders, digital cameras, and inertial navigation for mapping and localization [66]. Other approaches incorporate depth cameras, infrared sensors, gyro sensors, and accelerometers to construct detailed 3D maps and enhance localization accuracy. Techniques such as artificial landmark generation and moving average filters have also been explored to improve SLAM performance in pipeline environments. Recent advancements in pipeline inspection robotics have predominantly targeted larger pipelines in industries such as gas and oil, leaving a notable gap for small pipelines under 15mm in diameter [67]. A semi-automatic inspection system using a SLAM algorithm has been designed for such small pipelines, featuring three specialized robot types validated through experimental testing. The system operates via a wire-driven mechanism propelled by compressed air, effectively mapping the pipeline using sensor fusion techniques. Experimental results demonstrate the efficacy of this system in accurately navigating and mapping pipelines with multiple elbows and branches. Advancements in SLAM technology are transforming pipeline inspections, improving mapping accuracy and defect detection capabilities across various infrastructure scales. These innovations integrate diverse sensor arrays and sophisticated algorithms, enhancing operational safety, cost-effectiveness, and the sustainability of urban infrastructure management amidst evolving environmental and regulatory demands.

C. PAPER CONTRIBUTION AND PAPER STRUCTURE

The main contribution of this research paper is the design of robot actuation mechanism design and implementation of anomaly detection method in pipe inspection applications. In contemporary inpipe robot actuating mechanism, motors were located outside of driving chassis which makes less maneuverable in case of driving over the welding seams. Moreover, to detect such kind of seams or anomaly obstacles inside of the pipe.

II. ROBOT DESIGN

The proposed in-pipe robot, SIPIR, is a track-based mobile robot featuring a unique three-legged configuration connected to a centralized block housing essential electronics and motor driver components (see Fig. 1). This research project specifically targets the optimization of SIPIR's design for pipes with diameters ranging from 350 mm to 400 mm. This diameter range has been selected due to its prevalence in industrial water pipelines and the oil and gas industry.

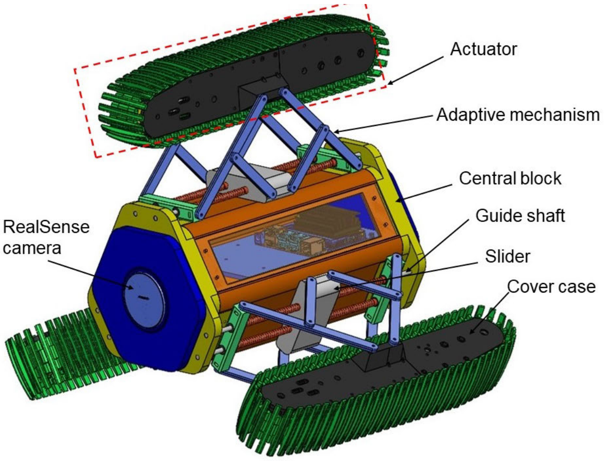


FIGURE 1. General inpipe robot CAD design.

A. ROBOT CENTRAL BLOCK DESIGN

Within the robot's central block resides both the robot control module and the computer vision module. Robot electronic board and batteries are located on a drawer board inside the robot central block (see Fig. 2). These include the Jetson board, Arduino module, and battery necessary to power the electronic boards effectively. Positioned at the forefront of the central block are the RealSense sensors, essential for computer vision tasks.

Furthermore, the central block serves a dual purpose, enhancing both stability and traction for the robot. This is achieved through the integration of a passive leg extension mechanism. This mechanism enables the robot to maintain constant traction force, facilitating smooth forward and backward motion while also adapting seamlessly to different pipe diameters.

B. ROBOT ACTUATION SYSTEM DESIGN

In this research project, our proposed advanced design is a motor placement inside of the chassis. Meanwhile, common track wheel based in-pipe mobile robots motors located in outer side of the chassis [68]. In chassis motor placement design allows to protect motor effectively and increase the robot index of protection. Moreover, such design accommodates the space for gearbox to convert motor speed and torque when it is necessary.

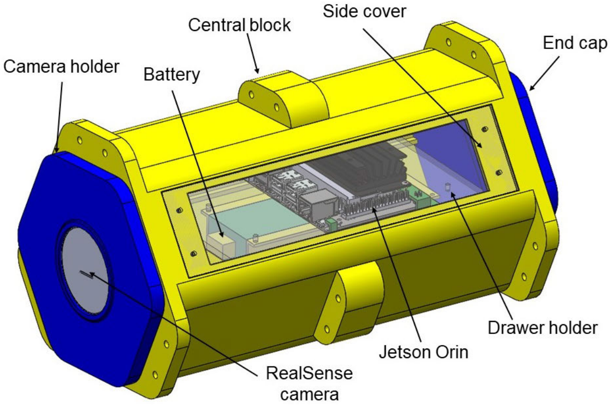


FIGURE 2. SIPIR central block.

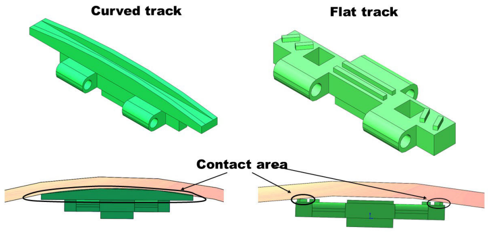


FIGURE 3. Inpipe robot track design comparison.

Furthermore, positioning the motor inside the chassis has significantly enhanced the robot's static balance and driving capabilities. Additionally, this design helps protect the motor from external forces and simplifies the implementation of waterproofing measures. (refer to Fig. 4). This strategic placement enhances the robot's stability and manoeuvrability during operation.

Additionally, we identified a critical need to modify the track design to suit the specific application environment. Commercially available tracks typically feature a flat shape, as depicted in Fig. 3. However, such designs often fail to provide sufficient traction on curved surfaces within pipes. To address this limitation, we innovated by designing curved tracks, enhancing the robot's traction and performance on varied pipe surfaces.

C. EXPERIMENTAL PROTOTYPE

In this research project, we developed a SIPIR prototype using rapid prototyping techniques. Most of the robot's components are constructed from PLA plastic, fabricated using an Ultimaker S5 3D printer (see the Fig. 7). The passive leg extension mechanism is actuated by helical compression

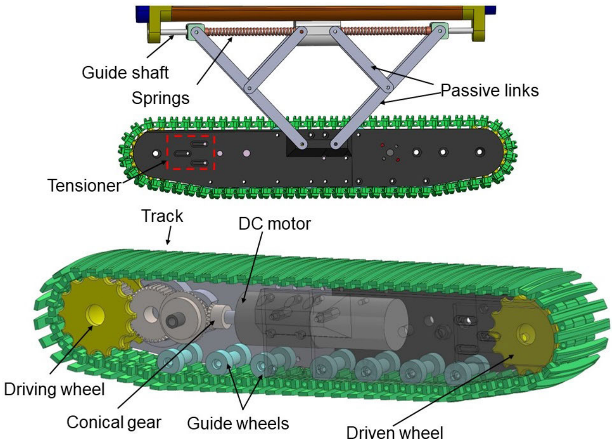


FIGURE 4. Inpipe robot actuator design.

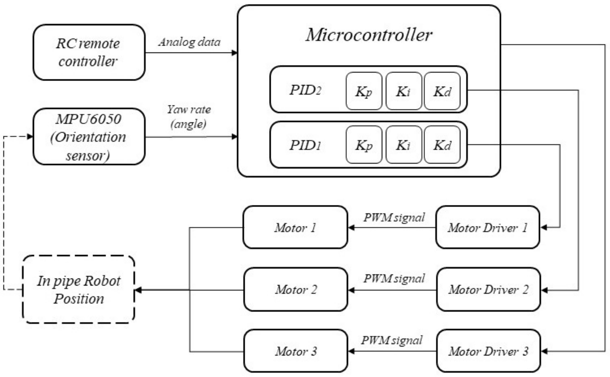


FIGURE 5. Robot block scheme.

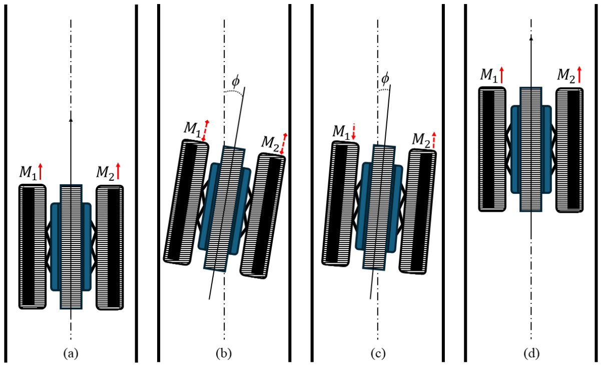


FIGURE 6. Robot orientation regulation.

springs. This setup enables the sliding linear bearing to move along a 5 mm diameter stainless steel guide shaft.

D. ROBOT CONTROL

In this research project, we applied a control method using a proportional-integral-derivative (PID) controller to move the robot inside a pipe. This control method allows precise adjustment of the motors, ensuring smooth and stable movement of the robot. The three motors are controlled using feedback from an inertial measurement unit (IMU), specifically the MPU6050 sensor, which measures the robot's angular velocity and acceleration, allowing precise control of its movement and orientation.

TABLE 1. Parameters.

Parameter	Value
Velocity	0.0333 m/s
Working Pipe Diameter	350-450 mm
Traction Force	93 N
Robot's Mass	4.8 kg
Robot's Height	0.36 m
Robot's Width	0.43 m

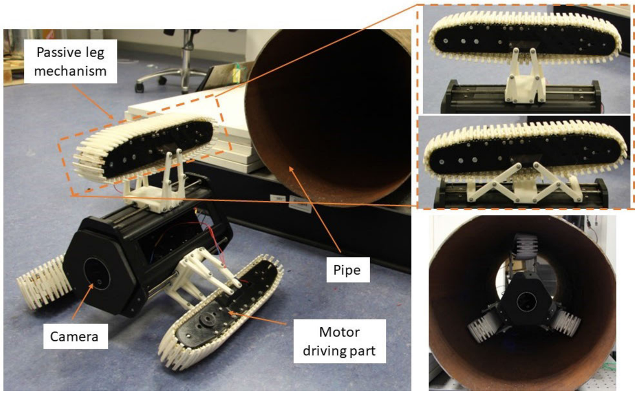


FIGURE 7. Experimental robot prototype SIPIR.

One of the significant problems in controlling the SIPIR robot is the slippage of the tracks inside the pipe. This phenomenon can cause the robot to get stuck or turn unintentionally, which negatively affects its performance. To solve this problem, the IMU sensor continuously monitors the orientation of the robot body relative to the pipe axis. The control system uses this data to adjust the speed and direction of rotation of the motors, keeping the robot body parallel to the axis of the pipe. This allows the motors to operate efficiently, preventing stalls and ensuring stable movement.

The Fig. 5 shows the robot control circuit. At the center of the control system is a microcontroller that receives data from the MPU6050 sensor about the angular velocity (yaw rate). The microcontroller uses two PID controllers to process this information and regulates the operation of three motors through motor drivers. Each PID controller consists of three components: proportional (Kp), integral (Ki) and differential (Kd). These components work together to minimize error in the robot's position and speed.

For additional control and flexibility in controlling the robot, an RC (radio-controlled controller) is used, which transmits analog data to the microcontroller. This allows the operator to remotely control the robot, making adjustments to its movement in real time. Such a control system provides not only automatic control, but also the possibility of manual control in difficult situations.

Thus, the developed SIPIR robot control system ensures high accuracy and stability of its movement. It allows the robot to adapt to different operating conditions inside the pipe, effectively coping with track slippage and unintended turns. The inclusion of PID controllers and remote control capabilities make the system flexible and reliable, capable of successfully completing assigned tasks in a wide variety of conditions.

The Fig. 6 illustrate the robot's behavior and correction mechanism:
Fig. 6(a): The robot moves straight along the central axis of the pipe with both motors M_1 and M_2 operating at equal speeds.
Fig. 6b): The robot deviates from the central axis by an angle ϕ.
Fig. 6(c): The control system adjusts the speeds of the motors M_1 and M_2 to generate a corrective moment that brings the robot back to the central axis.
Fig. 6(d): The robot returns to the central axis, and the motors resume operating at equal speeds.

Let:

- v_1 and v_2 be the linear velocities of the caterpillar tracks driven by motors M_1 and M_2.
- ϕ be the angle of deviation from the central axis.
- ω_y be the yaw rate measured by the MPU6050 sensor.
- t be the time over which correction occurs.
- k be a proportionality constant relating motor speed to track speed.
- L be the width of the robot (distance between the tracks).

Track Speeds: The linear velocities v_1 and v_2 are proportional to the motor speeds M_1 and M_2:

$$v_1 = k \cdot M_1, \quad v_2 = k \cdot M_2 \tag{1}$$

Angular Velocity and Moment: The robot's angular velocity ω is determined by the difference in track speeds:

$$\omega = \frac{v_2 - v_1}{L} \tag{2}$$

The yaw rate ω_y measured by the MPU6050 sensor is used for real-time correction:

$$\omega_y \approx \frac{v_2 - v_1}{L} \tag{3}$$

Deviation Correction: To return the robot to the central axis, the control system must adjust the speeds v_1 and v_2 such that the angular velocity ω corrects the deviation:

$$\omega_y \cdot t = \phi \Rightarrow \frac{v_2 - v_1}{L} \cdot t = \phi \tag{4}$$

Solving this equation for ϕ, we determine the necessary changes in track speeds:

$$v_2 - v_1 = \frac{\phi \cdot L}{t} \tag{5}$$

where t is the time required for the robot to return to its central position.

In practice, the microcontroller uses the yaw rate data ω_y from the MPU6050 sensor to dynamically adjust the motor speeds M_1 and M_2. The PID controllers use this information to minimize the deviation angle ϕ, ensuring that the robot maintains its intended path along the pipe.

By adjusting the speeds of motors M_1 and M_2, the robot's control system effectively corrects its orientation within the pipe. The mathematical model illustrates how the difference in track speeds, guided by the yaw rate data from the MPU6050 sensor, generates an angular velocity that corrects the robot's deviation, enabling stable and efficient movement along the pipe's central axis. This mechanism ensures the robot's ability to perform inspections without interruption or risk of jamming. The inclusion of PID controllers and remote control capabilities further enhance the system's flexibility and reliability, allowing it to adapt to a wide variety of conditions and successfully complete assigned tasks.

E. CALCULATION OF ROBOT CENTER OF MASS

The proposed SIPIR robot consists of three actuating units and a central block that houses the robot's electronic components. To ensure stability, the motors are positioned within the track wheel chassis.

To measure robot center of mass C_m

$$C_m = (X, Y, Z) \tag{6}$$

Robot total mass M can be calculated by following equation;

$$M = m_1 + m_2 + m_3 + m_4 \tag{7}$$

Here m_1 is a mass of central block, m_2, m_3 and m_4 are masses of the track wheel.

Then calculation of center of mass coordinates;

$$\begin{aligned} X &= \frac{(m_1 \cdot x_1) + (m_2 \cdot x_2) + (m_3 \cdot x_3) + (m_4 \cdot x_4)}{M} \\ Y &= \frac{(m_1 \cdot y_1) + (m_2 \cdot y_2) + (m_3 \cdot y_3) + (m_4 \cdot y_4)}{M} \\ Z &= \frac{(m_1 \cdot z_1) + (m_2 \cdot z_2) + (m_3 \cdot z_3) + (m_4 \cdot z_4)}{M} \end{aligned} \tag{8}$$

Here (x_1, y_1, z_1) is central block position coordinates, (x_2, y_2, z_2), (x_3, y_3, z_3) and (x_4, y_4, z_4) are track wheel position coordinates.

Then, (8) can be applied to the (6).

III. PIPE DEFECT INSPECTION SYSTEM

A. DATA COLLECTION AND PREPROCESSING

The uniqueness of our study lies in the approach to data collection and preprocessing. Given the challenge of sourcing internal images of pipes from the internet, we focused on creating a novel dataset using a custom-designed setup. This setup comprised an iron pipe stick equipped with a fisheye and web camera, along with a flashlight to illuminate the pipe's interior. This apparatus enabled us to capture comprehensive video footage from 53 different pipes, each exhibiting varying levels of corrosion.

Data collection was conducted under three distinct scenarios to simulate various inspection conditions: a straight

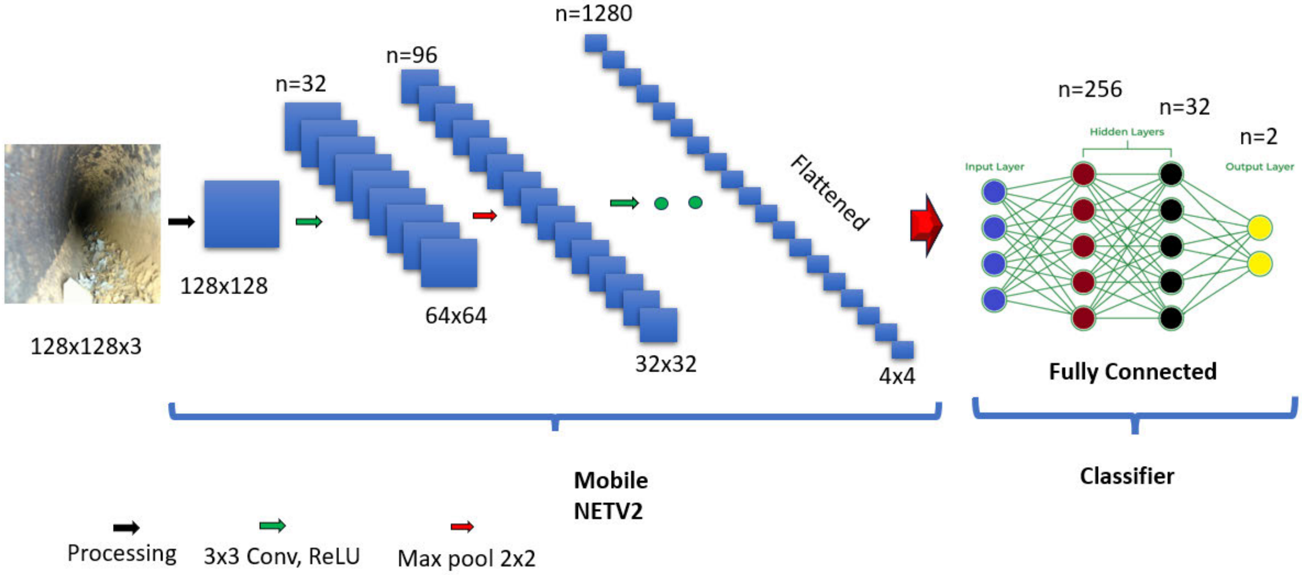


FIGURE 8. MobileNET computer vision model architecture.

insertion of the stick into the pipe, an axial rotation during insertion and retraction, and a spiral path to capture close-up images of the pipe walls. This approach enabled us to enhance our dataset by capturing images from diverse angles, ensuring the robustness of our data without compromising its quality. Consequently, 93 video sessions were conducted with an approximate total video length of 37.2 minutes, documenting the interior of the pipes. The videos were then categorized based on the presence or absence of corrosion and subsequently segmented into training, validation, and testing datasets. Image frames were extracted from these videos at a rate ranging from 8 to 12 frames per second (FPS). These frames were chosen to ensure that similar images were not fed several times during movement inside the pipe, on the other hand, to have enough dataset to make transfer learn the model, plus frames varied in the mentioned range to keep a ratio of 80-10-10 for train train-validation-test datasets. The final dataset comprised 4212 images for training, 370 for validation, and 512 for testing.

B. ML MODULE: COMPUTER VISION FOR REAL-TIME CLASSIFICATION CORROSION AND NO CORROSION CATEGORIES

1) MODEL DEVELOPMENT AND TRAINING

Our model's resilience and adaptability were significantly enhanced through a strategic data augmentation approach, employing the TensorFlow Keras preprocessing framework for images. Keras is a high-level neural network API written in Python and capable of running on top of TensorFlow. It enables fast experimentation with deep neural networks and is user-friendly, modular, and extensible. This augmentation method effectively expands the training dataset by introducing variations that simulate real-world scenarios. Specifically, the augmentation parameters included shear transformations with a range of 0.2 to mimic slanting effects caused by perspective shifts; zoom adjustments with a range of 0.2 to replicate varying focal distances; random shifts in the width and height of the images, each with a range of 0.2, to simulate off-center positioning; rotations up to 40 degrees to introduce orientation variance; horizontal and vertical flips to mitigate directional biases; and channel shifts with a range of 20 to emulate different lighting conditions. In addition, a custom augmentation function was implemented, which incorporated standard MobileNetV2 preprocessing and specific enhancements tailored to our project. These enhancement techniques were crucial in training our model to generalize effectively from a limited data set to a diverse array of practical inspection scenarios. Cameras inherently exhibit distortion that affects the shape and colour of captured objects, necessitating calibration or preprocessing. We initially collected data using both a fisheye camera and a Logitech C920 HD Pro web camera. Given the high-quality, low-distortion images produced by the Logitech C920 HD Pro web camera, we chose to focus exclusively on its images, thus obviating the need for additional distortion correction algorithms. This approach streamlined our workflow and ensured that our project objectives were met.

For the classification task, we used a pre-trained MobileNetV2 architecture with its top layers replaced by newly added fully connected layers sized 256, 32, and 2, as depicted in Fig. 8. We tested several computer vision algorithms, including ResNet50, Inception, EfficientNet, and VGG16, with thorough training and hyperparameter tuning. However, due to our small dataset, MobileNet consistently outperformed the others, which often underfit or overfit. Considering our robot's need for efficient power consumption to navigate pipes, we chose MobileNet for its optimal efficiency and suitability for this application. With our modification to the base model with approximate parameters equal to 3.4 million parameters and based on TABLE 2, the number of parameters in our model in total is approximately equal to 3,536,769 [37]. We incorporated global average pooling, batch normalization, and dropout layers to customize it for our specific needs. Transfer learning was implemented by freezing the backbone network's weights and training only the newly introduced layers.

We used an RMSprop optimizer with a learning rate of 1e-5 to optimize a binary cross-entropy loss function.

In our training process, we utilized Keras and incorporated callbacks such as ModelCheckpoint and ReduceLROnPlateau to enhance model performance and mitigate overfitting. ModelCheckpoint allows the saving of the model at specific intervals when it achieves optimal validation accuracy, ensuring the retention of the most effective model. On the other hand, ReduceLROnPlateau adjusts the learning rate when there is no improvement in performance for a predefined number of epochs, thus aiding in finer optimization during the training phase. These tools proved crucial in controlling the quality of the model training, particularly as signs of overfitting became evident starting at epoch 8, with a total of 13 epochs of training in general.

TABLE 2. model architecture.

Layer (type)	Output Shape	Param #
MobileNetV2 (Base Model)	(1, 1, 1280)	3.4 mln parameters
GlobalAveragePooling2D	(1280)	0
Dense (ReLU)	(256)	327936
BatchNormalization	(256)	512
Dropout (0.19)	(256)	0
Dense (ReLU)	(32)	8224
BatchNormalization	(32)	64
Dropout (0.5)	(32)	0
Dense (Sigmoid)	(1)	33

2) MODEL EVALUATION

Post-training, the model's performance was evaluated on the test dataset, which was processed with the same image preprocessing techniques as the training and validation sets. This step was crucial to assessing the model's ability to generalize to new, unseen data. The threshold for predictions was established at 0.7. This was determined through correct inference and multiple rounds of analysis using images we collected as well as images sourced from the internet. Our model demonstrated promising performance with an accuracy of 88.28% on the test dataset, indicating its effectiveness in classifying corrosion in pipes. The confusion matrix can be found in Fig. 9, The model exhibited some confusion in its predictions; it mislabeled cases as "Corrosion" when they were actually "Without Corrosion" about 17.96% of the time. Conversely, it misclassified "Without Corrosion" cases as "Corrosion" roughly 5.46% of the time. These figures indicate the areas where the model's predictive accuracy could be improved.

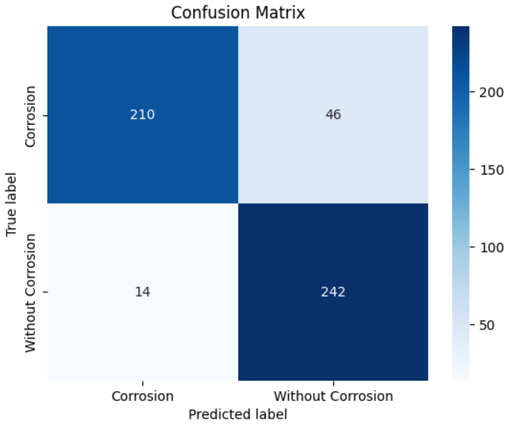


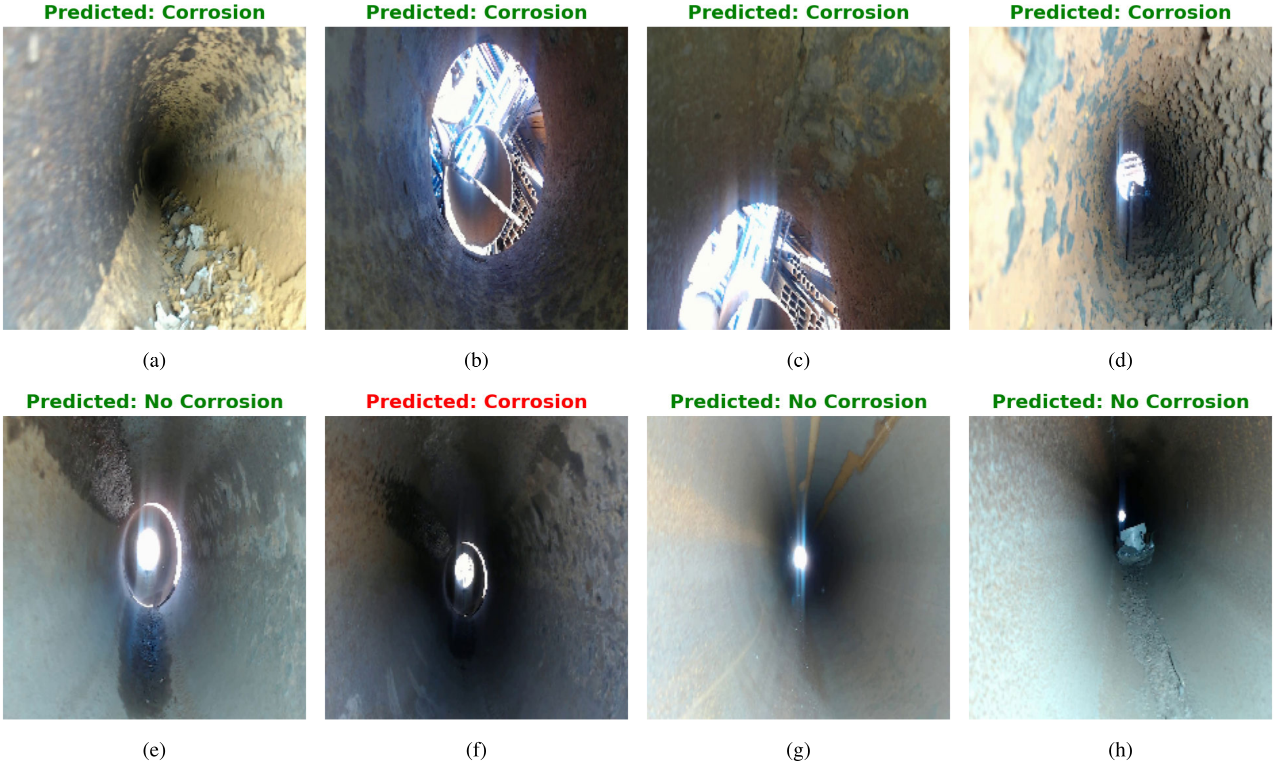
FIGURE 9. Confusion matrix of MobileNet model on test dataset.

The successful implementation of this model showcases the potential of using computer vision techniques in autonomous pipe inspection, particularly in scenarios where traditional methods might be challenging or infeasible.

Building on the discussion thus far, this study systematically analyzed data obtained from pipes exhibiting various degrees of corrosion. Fig. 10 displays a set of images from the test dataset showcasing different corrosion stages that the model mostly accurately identified and categorized in Figs. 10(a) - 10(d), and uncorroded pipes were also correctly classified in Figs. 10(e) - 10(h).

To evaluate the model's adaptability to diverse conditions, such as varying light exposure, corrosion types, and pipe materials, additional images were sourced from the internet 18 images and 5 images were misclassified. The model's classification capabilities, using these externally obtained images, are demonstrated in Fig. 11. Working with the prediction threshold of the trained model, MobileNetV2 performed very well on all the test datasets obtained from the internet and collected datasets. However, discrepancies were observed in the model predictions, specifically in Fig. 11(a) and Fig. 11(e), as well as in Fig. 10(f). For Fig. 11(a) and Fig. 11(e) displayed images markedly distinct from those in the collected dataset. Regarding Fig. 10(f), an ambiguity arises due to the presence of non-smooth surfaces on the right-side wall of the pipe, despite its classification as non-corrosive. This characteristic may have contributed to the model erroneously categorizing it as corrosive.

The implementation of the trained model is intended for a custom-designed mobile robot equipped with a Jetson microcontroller, specifically engineered for pipe inspection tasks. Considering the operational requirements of such a robot, particularly the necessity for swift decision-making, the model's inference speed is a critical factor. To assess this, we conducted tests involving 50 distinct images to determine the average inference time and its variability. These tests revealed that the streamlined MobileNet architecture



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 10. Predicted classes from different pipes with different corrosion levels and without corrosion (The label turns green when the model's prediction matches ground truth, and it turns red when there is a discrepancy).

achieved an average inference time of 0.1534 seconds, with a standard deviation of 0.2 seconds, indicating a generally consistent and rapid response suitable for real-time inspection scenarios.

3) ANOMALY DETECTION USING IMAGE PROCESSING TECHNIQUES

Anomaly detection in images is a challenging task, particularly in uncontrolled environments where anomalies can vary in size, shape, and intensity. Traditional methods often rely on predefined rules or manual inspection, which are not scalable or reliable for large datasets. Our approach automates the process using advanced image processing techniques such as Canny edge detector and DBSCAN clustering.

The proposed method consists of several stages:

- **Image Preprocessing**
  - **Process:** The methodology begins with the preprocessing of the image, converting it to the HSV (Hue, Saturation, Value) color space. This conversion is crucial as HSV is more effective in separating brightness from color information, which is essential for the accurate identification of anomalies based on intensity variations.
  - **Parameter and Effect:** Sensitivity, set at a value of 20, dictates the range of pixel intensities considered as potential anomalies, influencing the thresholding in the HSV space.
- **Anomaly Isolation**
  - **Process:** A mask is created to isolate bright white areas that are often indicative of anomalies. Using a specific sensitivity range, a binary mask is generated that selectively highlights these brightest areas in the image, focusing the anomaly detection process on regions of interest.
  - **Parameter and Effect:** Kernel Size, configured as 5 × 5, influences the dilation of the mask, connecting bright spots to nearby areas and thus expanding the focus areas where anomalies may be detected.
- **Mask Dilation and Inversion**
  - **Process:** After isolating bright areas, the mask is dilated using a kernel to cover surrounding pixels, ensuring comprehensive coverage of potential anomalies. The dilation process widens the area of interest around bright spots. Subsequently, the mask is inverted to shift focus to the darker, non-bright regions, aiding in distinguishing potential anomalies from the surrounding background.
  - **Parameter and Effect:** The 5 × 5 Kernel Size not only facilitates the expansion of identified bright regions but also ensures that the inversion highlights significant non-bright anomalies, enhancing the detection accuracy.

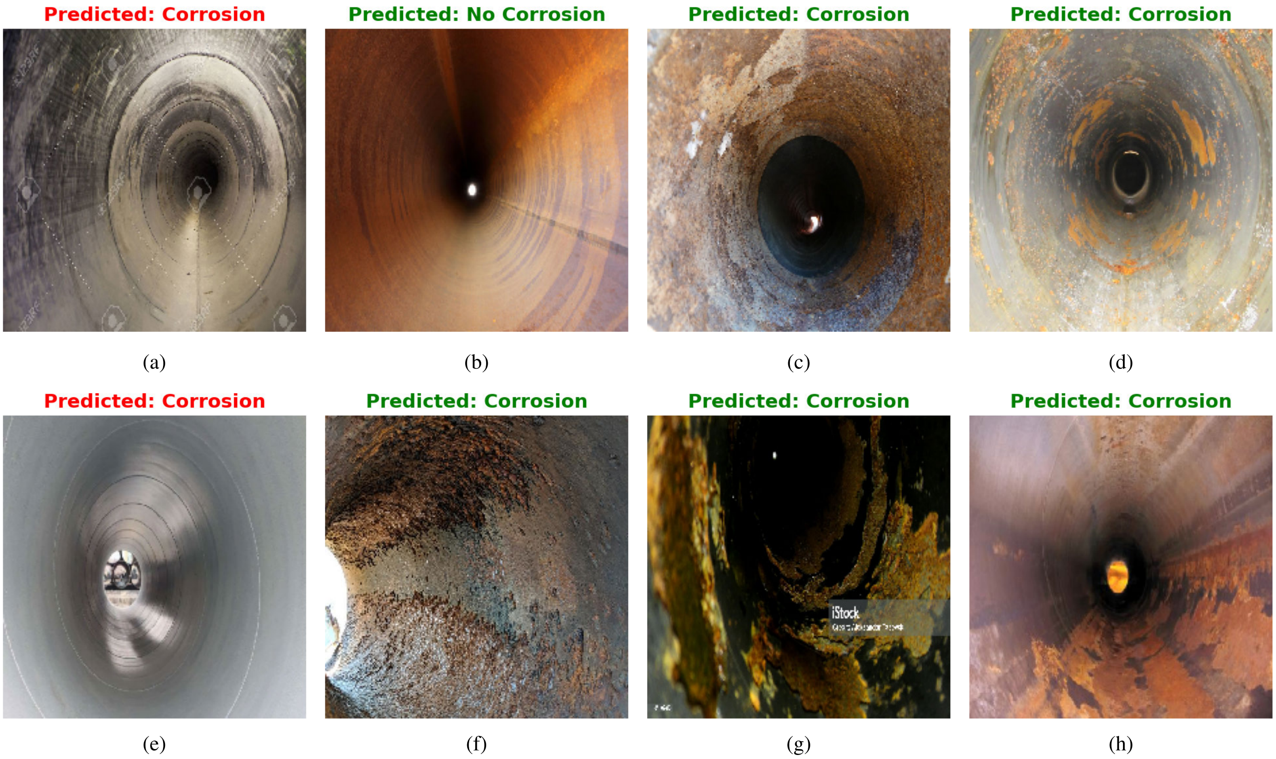


FIGURE 11. 8 × 2 Grid of Different Pipe Images with Corrosion Collected from the Internet (The label turns green when the model's prediction matches ground truth, and it turns red when there is a discrepancy).

- **Edge Detection and Contour Extraction**
  - **Process:** Post mask inversion, the image is transformed to grayscale and subjected to Gaussian Blur with a 7 × 7 parameter setting to reduce noise and irrelevant details. Following this, edge detection is performed using the Canny algorithm with thresholds set at 17 and 90, which identifies sharp changes in intensity that delineate potential anomalies.
  - **Parameter and Effect:** Gaussian Blur Parameters (7, 7) smooth the image to reduce noise before edge detection, enhancing clarity for more accurate edge detection. The Canny Edge Detection Thresholds (17, 90) determine the intensity gradients considered as edges, crucial for contour extraction.
- **Feature Extraction**
  - **Process:** Each extracted contour is analyzed to determine its geometric properties, such as area and centroid. Contours that exceed a predefined area threshold are flagged as potential anomalies, focusing analysis on significant irregularities.
  - **Parameter and Effect:** Minimum Area Threshold set at 5 filters out smaller, less significant contours, ensuring that only larger, more relevant anomalies are considered for further analysis.
- **Clustering Using DBSCAN**
  - **Process:** The final step involves applying the DBSCAN clustering algorithm to the features extracted from the contours. DBSCAN is known for its effectiveness in handling clusters of arbitrary shapes and sizes, grouping the identified features based on their proximity and density.
  - **Parameters and Effects:**
    - ∗ **DBSCAN 'eps':** Set at 50, it determines the spatial proximity required for points to be considered part of the same cluster, affecting how broadly anomalies are grouped.
    - ∗ **DBSCAN 'min_samples':** At a setting of 1, this parameter decides the minimum number of samples required in a neighbourhood to form a dense region, thus defining cluster density.
    - ∗ **Cluster Area Threshold:** Set at 10, it controls the significance of detected clusters to be labelled as anomalies, ensuring that only clusters with a substantial area are considered during the final analysis.

The whole process of image processing can be seen in Fig. 12. The performance of the algorithm can be seen in Fig. 13. Different objects that were left a long time ago and naturally covered with dust might cause difficulty in identifying them using image processing techniques; however, the proposed algorithm was able to identify all

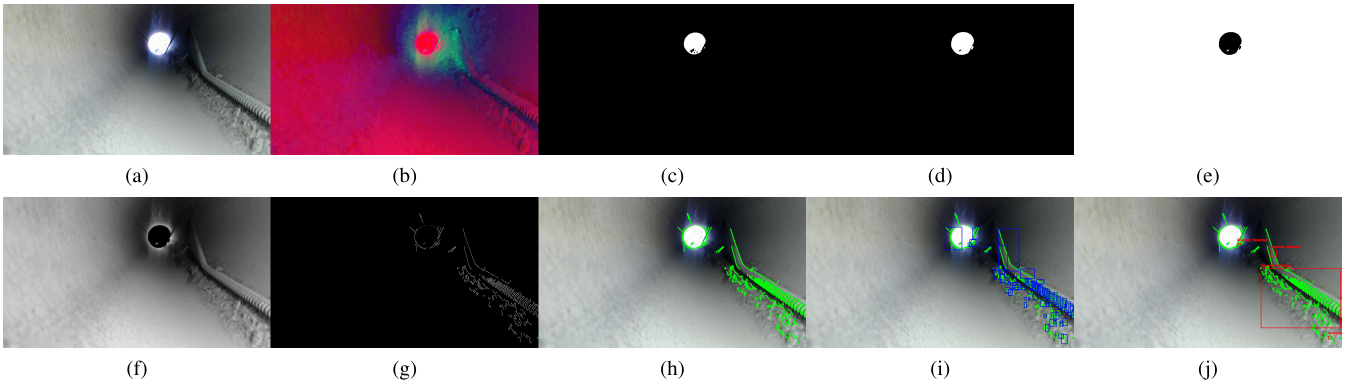


FIGURE 12. Automated Anomaly Detection pipeline using Canny edge detector and DBSCAN Clustering. The pipeline begins with the raw input (a), which is first converted to the HSV color space (b) to enhance color differentiation. Bright areas are then isolated (c) and expanded through dilation (d) to ensure complete coverage of significant features. These are inverted (e) to focus on darker regions. A Gaussian blur (f) is applied to smooth the image and reduce noise, preparing it for edge detection. The Canny edge detector (g) outlines prominent edges, and contours are traced (h) to delineate features clearly. Anomalies are detected by highlighting unusual patterns (i), and the process culminates with DBSCAN clustering (j), which groups similar data points to identify clusters of anomalies.

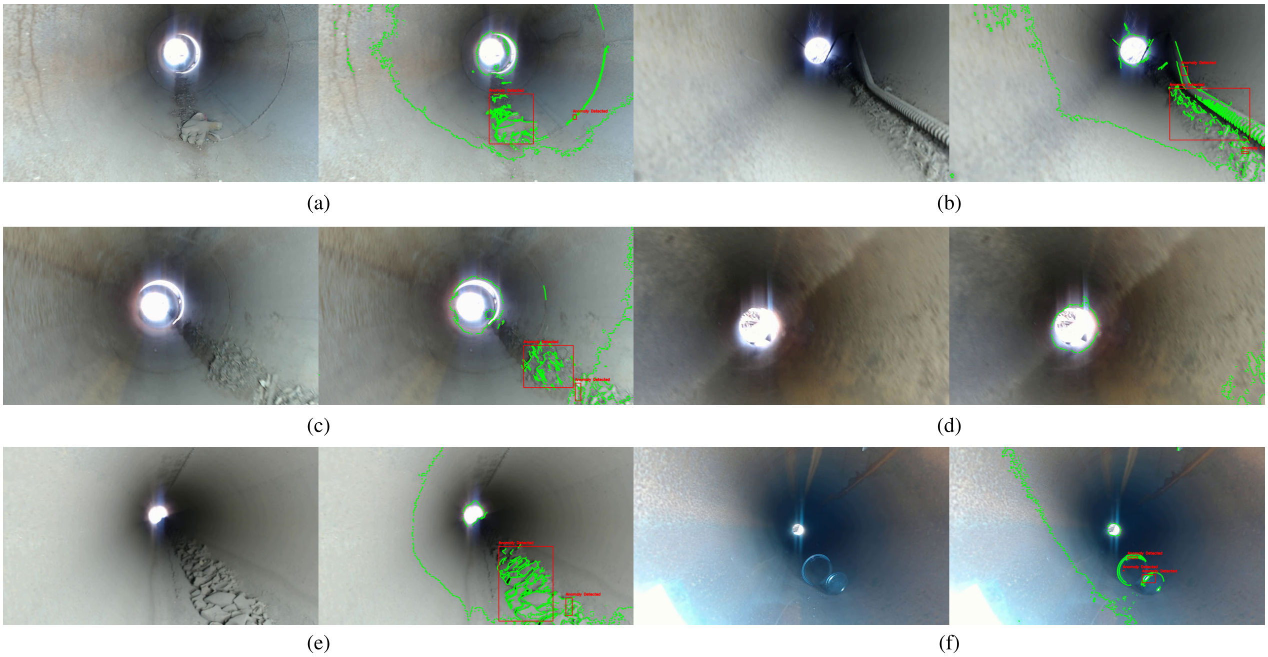


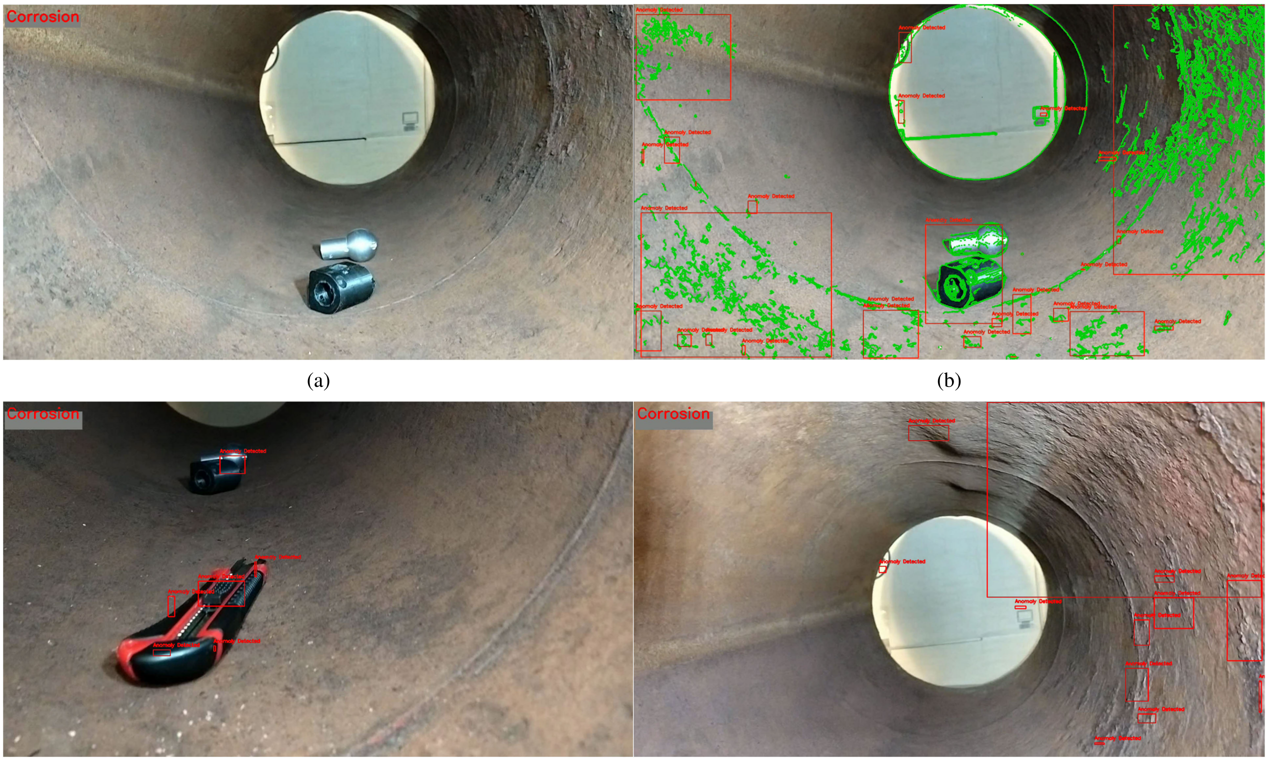
FIGURE 13. Automated anomaly detection in images using canny edge detector and DBSCAN clustering.

of them. The Fig. 13(d) demonstrates that algorithms do not react if there are no anomalies. The average inference time of the anomaly detection algorithm over 50 different images was 0.0232 seconds which is 6.6x faster compared to MobileNet inference time. As a consequence, we did such that MobileNet algorithms activate only when an anomaly is detected.

In Fig. 14, Fig. 14(a) showcases the MobileNet model's performance within a controlled laboratory setting, utilizing our test pipe. The presence of roughness within the pipe is unmistakable, and our model adeptly identifies these textural anomalies as indicators of rust, demonstrating its keen ability to detect signs of corrosion.

In Fig. 14(b), the anomaly detection algorithm-based image processing technique exhibits its precision in detecting all the rough patches along the right side of the internal wall of the pipe. This precision underscores the algorithm's capacity to identify even the subtlest signs of potential degradation.

For Fig. 14(c) and 14(d), the MobileNet and Anomaly detection algorithms operate in tandem. The absence of the green edge lines, typically highlighted by the Canny edge detector, indicates a heightened sensitivity of the edge detection process, allowing for a cleaner and more focused identification of anomalies. Fig. 14(d) is particularly noteworthy as it reveals the algorithms' selective response to foreign objects within the pipe, excluding the roughness



(a)

(b)

(c)

(d)

FIGURE 14. Automated Anomaly Detection in Images Using Canny Edge Detector and DBSCAN Clustering and corrosion detection using MobileNet.

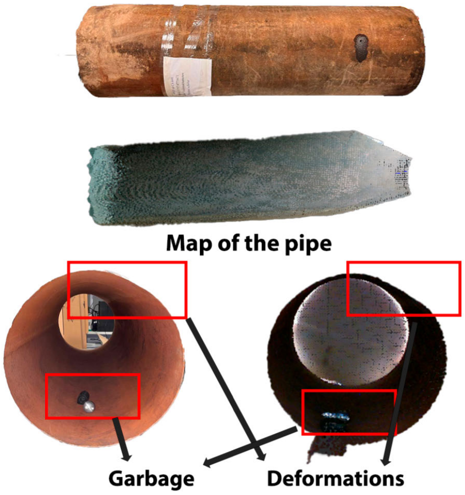


FIGURE 15. Deformation and garbage identification using SLAM.

of the internal surface. This specificity is beneficial in environments where only object detection is desired. Conversely, Fig. 14(c) demonstrates the system's responsiveness when the camera gets near the wall, where the algorithms begin to register the rough surface texture, which is indicative of the robustness of our integrated system in varying operational contexts.

C. INSPECTION BY SLAM

Inspection by SLAM (Simultaneous Localization and Mapping) is carried out using Intel Realsense L515, Jetson Orin NX that runs on Ubuntu 18.04. Intel Realsense depth cameras have internal IMU, however they are very unstable when performing SLAM. Firstly, if USB connected to the camera get touched, there are interruptions in data transferred, and if it is disrupted then the data will stop flowing. Secondly, if camera is moved rapidly, the IMU starts to drift or shake violently. Despite the unreliable setup, with slow movement speed and securing USB cable SLAM was successful. In this setup camera simultaneously creates map, and saves them, which later could be replayed. Map allows the user to identify the defects within pipe and accurately find where they are located due to the internal measurement unit. Moreover, creating an accurate map allows the user to see if the pipeline is deformed or not. However, achieving good results was very difficult, due to drifting issues, in the future, other methods in order to calculate the robot's position from the data should be used, either by using a simpler camera with separate IMU, and perhaps a GPS system.

In this research, the selection of the SLAM algorithm was based on its lightweight nature and customizable memory features, leading to the use of the GMapping algorithm for this application. Additionally, the Hector SLAM method, which operates without relying on odometry parameters, proves suitable for cluttered environments where odometry data may be unreliable. Although the TinySLAM algorithm is also lightweight, it is not suitable for dealing with the complex parameters essential for inspecting well-rusted pipes.

IV. EXPERIMENTAL VALIDATION ON PIPE INSPECTION

Experimental setup consisted of the SIPIR with Jetson Orin NX attached to the Intel Realsense L515, that traversed inside the pipe with 0,03 m/s. It travelled from start to the end, capturing the RGB and depth videos, as well as constructing the map. In some of the experiments, LED lights were used which made the map whiter, in other times map came out dark-brown.

V. CONCLUSION AND DISCUSSION

This research project introduced an inpipe mobile robot SIPIR equipped with machine learning (ML) and Simultaneous Localization and Mapping (SLAM) based inspection tools. The driving actuator of the proposed in-pipe robot exhibited remarkable adaptability and offered adequate traction, with the mobile platform achieving a pulling force of 100N. Additionally, the ML tool developed for detecting pipe defects effectively. Utilizing machine vision, the robot successfully identified anomalies, corrosion, and cracks. Furthermore, the integration of SLAM technology enabled the detection of pipe deformations arising from both external and internal impacts.

REFERENCES

[1] C. Ye, L. Liu, X. Xu, and J. Chen, "Development of an in-pipe robot with two steerable driving wheels," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2015, pp. 1955–1959.

[2] M. Z. A. Rashid, F. Yakub, S. A. S. Salim, S. A. Roslan, H. N. M. Shah, A. Yeshmukhametov, and Y. Yamamoto, "Recent trends of the in-pipe inspection robotic system from academia and industry perspectives," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1051, no. 1, Feb. 2021, Art. no. 012034.

[3] T. Mushiri, S. Ndlovu, and C. Mbohwa. (2016). *Design of a Mechanical Cleaning Device Pig (Pipeline Intervention Gadget) Connecting Two Transfer Lines in Zimbabwe*. [Online]. Available: https://api.semanticscholar.org/CorpusID:116446378

[4] T. Ren, Y. Zhang, Y. Li, Y. Chen, and Q. Liu, "Driving mechanisms, motion, and mechanics of screw drive in-pipe robots: A review," *Appl. Sci.*, vol. 9, no. 12, p. 2514, Jun. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/12/2514

[5] D. Fang, J. Shang, Z. Luo, P. Lv, and G. Wu, "Development of a novel self-locking mechanism for continuous propulsion inchworm in-pipe robot," *Adv. Mech. Eng.*, vol. 10, no. 1, Jan. 2018, Art. no. 168781401774940. [Online]. Available: https://api.semanticscholar.org/CorpusID:115960941

[6] Q. Ma, G. Tian, Y. Zeng, R. Li, H. Song, Z. Wang, B. Gao, and K. Zeng, "Pipeline in-line inspection method, instrumentation and data management," *Sensors*, vol. 21, no. 11, p. 3862, Jun. 2021.

[7] A. Verma, A. Kaiwart, N. D. Dubey, F. Naseer, and S. Pradhan, "A review on various types of in-pipe inspection robot," *Mater. Today, Proc.*, vol. 50, pp. 1425–1434, Jan. 2022.

[8] G. H. Jackson-Mills, B. A. Shead, J. R. Collett, M. Mphake, N. Fry, A. R. Barber, J. H. Boyle, R. C. Richardson, A. E. Jackson, and S. Whitehead, "Non-assembly walking mechanism for robotic in-pipe inspection," in *Robotics for Sustainable Future*. Springer, 2021, pp. 117–128.

[9] P. K. Dey, S. O. Ogunlana, and S. Naksuksakul, "Risk-based maintenance model for offshore oil and gas pipelines: A case study," *J. Quality Maintenance Eng.*, vol. 10, no. 3, pp. 169–183, Sep. 2004.

[10] S. Razvarz, R. Jafari, A. Gegov, S. Razvarz, R. Jafari, and A. Gegov, "The importance of pipeline transportation," in *Flow Modelling and Control in Pipeline Systems: A Formal Systematic Approach*, 2021, pp. 1–24.

[11] (2021). *Energy Resource Guide—Oil and Gas—Kazakhstan*. [Online]. Available: https://www.trade.gov/energy-resource-guide-oil-and-gas-kazakhstan

[12] S. Nadig. (Jul. 2022). *Oil Pipelines Burst in Tengiz Field, Two Casualties Reported News*. [Online]. Available: https://www.offshore-technology.com/news/tengiz-field-explosion-reports-two-casualties/

[13] I. N. Ismail, A. Anuar, K. S. M. Sahari, M. Z. Baharuddin, M. Fairuz, A. Jalal, and J. M. Saad, "Development of in-pipe inspection robot: A review," in *Proc. IEEE Conf. Sustain. Utilization Develop. Eng. Technol. (STUDENT)*, Oct. 2012, pp. 310–315.

[14] A. J. Nanne, M. L. Antheunis, C. G. Van Der Lee, E. O. Postma, S. Wubben, and G. Van Noort, "The use of computer vision to analyze brand-related user generated image content," *J. Interact. Marketing*, vol. 50, no. 1, pp. 156–167, May 2020.

[15] S. K. Sinha, P. W. Fieguth, and M. A. Polak, "Computer vision techniques for automatic structural assessment of underground pipes," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 18, no. 2, pp. 95–112, Mar. 2003.

[16] H.-S. Kim, B.-R. Lee, and R.-J. Kim, "Development of computer-vision-based pipe inspection system," in *Proc. Int. Forum Strategic Technol.*, Oct. 2006, pp. 403–406.

[17] M. Wang and J. C. Cheng, "Development and improvement of deep learning based automated defect detection for sewer pipe inspection using faster R-CNN," in *Proc. 25th EG-ICE Int. Workshop Adv. Comput. Strategies Eng.*, Lausanne, Switzerland. Springer, Jun. 2018, pp. 171–192.

[18] R. Rayhana, Y. Jiao, A. Zaji, and Z. Liu, "Automated vision systems for condition assessment of sewer and water pipelines," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1861–1878, Oct. 2021.

[19] B. T. Bastian, N Jaspreeth, S. K. Ranjith, and C. V. Jiji, "Visual inspection and characterization of external corrosion in pipelines using deep neural network," *NDT & E Int.*, vol. 107, Oct. 2019, Art. no. 102134.

[20] S. Moradi, T. Zayed, and F. Golkhoo, "Review on computer aided sewer pipeline defect detection and condition assessment," *Infrastructures*, vol. 4, no. 1, p. 10, Mar. 2019.

[21] M. Malek Mohammadi, M. Najafi, V. Kaushal, R. Serajiantehrani, N. Salehabadi, and T. Ashoori, "Sewer pipes condition prediction models: A state-of-the-art review," *Infrastructures*, vol. 4, no. 4, p. 64, Oct. 2019.

[22] O. P. Oluwatosin, S. A. Syed, O. Apis, and S. Kolawole, "Application of computer vision in pipeline inspection robot," in *Proc. Int. Conf. Ind. Eng. Oper. Manage.*, Singapore, Mar. 2021.

[23] X. Peng, U. Anyaoha, Z. Liu, and K. Tsukada, "Analysis of magnetic-flux leakage (MFL) data for pipeline corrosion assessment," *IEEE Trans. Magn.*, vol. 56, no. 6, pp. 1–15, Jun. 2020.

[24] A. Colvalkar, S. S. Pawar, and B. K. Patle, "In-pipe inspection robotic system for defect detection and identification using image processing," *Mater. Today, Proc.*, vol. 72, pp. 1735–1742, Jan. 2023.

[25] Y. Li, H. Wang, L. M. Dang, H.-K. Song, and H. Moon, "Vision-based defect inspection and condition assessment for sewer pipes: A comprehensive survey," *Sensors*, vol. 22, no. 7, p. 2722, Apr. 2022.

[26] J. B. Haurum and T. B. Moeslund, "Sewer-ML: A multi-label sewer defect classification dataset and benchmark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13451–13462.

[27] P. N. Srinivasu, J. G. Sivasai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, Apr. 2021.

[28] Z. N. I. Zailan, S. A. Mostafa, A. I. Abdulmaged, Z. Baharum, M. M. Jaber, and R. Hidayat, "Deep learning approach for prediction of brain tumor from small number of MRI images," *Int. J. Informat. Vis.*, vol. 6, no. 2, p. 581, Aug. 2022.

[29] M. E. Karar, O. Reyad, M. Abd-Elnaby, A.-H. Abdel-Aty, and M. A. Shouman, "Lightweight transfer learning models for ultrasound-guided classification of COVID-19 patients," *Comput., Mater. Continua*, vol. 69, no. 2, pp. 2295–2312, 2021.

[30] D. He, Z. Yao, Z. Jiang, Y. Chen, J. Deng, and W. Xiang, "Detection of foreign matter on high-speed train underbody based on deep learning," *IEEE Access*, vol. 7, pp. 183838–183846, 2019.

[31] B. Khasoggi, E. Ermatita, and S. Samsuryadi, "Efficient mobilenet architecture as image recognition on mobile and embedded devices," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, p. 389, Oct. 2019.

[32] Y. Yuan, "Computer vision and deep learning for precise agriculture: A case study of lemon leaf image classification," *J. Phys., Conf. Ser.*, vol. 2547, no. 1, Jul. 2023, Art. no. 012024.

[33] S. Jiang, Y. Cheng, and J. Zhang, "Vision-guided unmanned aerial system for rapid multiple-type damage detection and localization," *Struc Health Monit.*, vol. 22, no. 1, pp. 319–337, Jan. 2023.

[34] T. Feng, J. Liu, X. Fang, J. Wang, and L. Zhou, "A double-branch surface detection system for armatures in vibration motors with miniature volume based on ResNet-101 and FPN," *Sensors*, vol. 20, no. 8, p. 2360, Apr. 2020.

[35] L. Calton and Z. Wei, "Using artificial neural network models to assess hurricane damage through transfer learning," *Appl. Sci.*, vol. 12, no. 3, p. 1466, Jan. 2022.

[36] H. Jin Lim, S. Hwang, H. Kim, and H. Sohn, "Steel bridge corrosion inspection with combined vision and thermographic images," *Struct. Health Monit.*, vol. 20, no. 6, pp. 3424–3435, Nov. 2021.

[37] M. Hollemans. (2018). *MobileNet Version 2*. Accessed: Jan. 24, 2024. [Online]. Available: https://machinethink.net/blog/mobilenet-v2/

[38] M. Rajput. (2020). *YOLO V5—Explained and Demystified*. Towards AI. Accessed: Jan. 24, 2024.

[39] R. Thakur. (2023). *Beginner's Guide to VGG16 Implementation in Keras*. Built. Accessed: Jan. 24, 2024.

[40] *Torchvision Faster R-CNN*. Accessed: Jan. 24, 2024. [Online]. Available: https://paperswithcode.com/lib/torchvision/faster-r-cnn

[41] (2024). *ResNet-50 Trained on ImageNet Competition Data*. Accessed: Jan. 24, 2024. [Online]. Available: https://resources.wolframcloud.com/NeuralNetRepository/resources/ResNet-50-Trained-on-ImageNet-Competition-Data

[42] O. Ali, H. Ali, S. A. A. Shah, and A. Shahzad, "Implementation of a modified U-Net for medical image segmentation on edge devices," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 11, pp. 4593–4597, Nov. 2022.

[43] L. Guo and S. Wu, "FPGA implementation of a real-time edge detection system based on an improved Canny algorithm," Tech. Rep., 2022.

[44] S. Mustapha, "An alternative parameter free algorithm to DBSCAN method by using data point positioning analysis (DBSCAN-DPPA)," Tech. Rep., 2022.

[45] D. Song, X. Xu, X. Cui, Y. Ou, and W. Chen, "Bolt looseness detection based on Canny edge detection algorithm," *Concurrency Comput., Pract. Exp.*, vol. 35, no. 21, Sep. 2023.

[46] Q. Li, Z. Zhu, and J. Liang, "A satellite component contour extraction method for lightweight space mobile platforms," *Aircr. Eng. Aerosp. Technol.*, vol. 95, no. 8, pp. 1217–1226, Jul. 2023.

[47] Y. Lu, L. Duanmu, Z. Zhai, and Z. Wang, "Application and improvement of Canny edge-detection algorithm for exterior wall hollowing detection using infrared thermal images," *Energy Buildings*, vol. 274, Nov. 2022, Art. no. 112421.

[48] M. Huang, Y. Liu, and Y. Yang, "Edge detection of ore and rock on the surface of explosion pile based on improved Canny operator," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 10769–10777, Dec. 2022.

[49] Z. Su, W. Liu, Z. Yu, D. Hu, Q. Liao, Q. Tian, M. Pietikäinen, and L. Liu, "Pixel difference networks for efficient edge detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5097–5107.

[50] D. Dhillon and R. Chouhan, "Enhanced edge detection using SR-guided threshold maneuvering and window mapping: Handling broken edges and noisy structures in Canny edges," *IEEE Access*, vol. 10, pp. 11191–11205, 2022.

[51] A. Isar, C. Nafornita, and G. Magu, "Hyperanalytic wavelet-based robust edge detection," *Remote Sens.*, vol. 13, no. 15, p. 2888, Jul. 2021.

[52] N. E. Ramli, Z. R. Yahya, and N. A. Said, "Confusion matrix as performance measure for corner detectors," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 29, no. 1, pp. 256–265, Dec. 2022.

[53] C. Cammalleri and A. Toreti, "A generalized density-based algorithm for the spatiotemporal tracking of drought events," *J. Hydrometeorology*, vol. 24, no. 3, pp. 537–548, Mar. 2023.

[54] G. S. Fuhnwi, J. O. Agbaje, K. Oshinubi, and O. J. Peter, "An empirical study on anomaly detection using density-based and representative-based clustering algorithms," *J. Nigerian Soc. Phys. Sci.*, p. 1364, Apr. 2023.

[55] F. Cheng, G. Niu, Z. Zhang, and C. Hou, "Improved CNN-based indoor localization by using RGB images and DBSCAN algorithm," *Sensors*, vol. 22, no. 23, p. 9531, Dec. 2022.

[56] L. Liu, Y. Zhang, Y. Hu, Y. Wang, J. Sun, and X. Dong, "A hybrid-clustering model of ship trajectories for maritime traffic patterns analysis in port area," *J. Mar. Sci. Eng.*, vol. 10, no. 3, p. 342, Mar. 2022.

[57] J. Guo, J. Zhang, Y. Zhang, P. Xu, L. Li, Z. Xie, and Q. Li, "An improved density-based approach to risk assessment on railway investment," *Data Technol. Appl.*, vol. 56, no. 3, pp. 382–408, Jun. 2022.

[58] W. Zhao, Q. Li, and K. Jin, "An algorithm using DBSCAN to solve the velocity dealiasing problem," *Adv. Meteorol.*, vol. 2021, pp. 1–9, Dec. 2021.

[59] J. Bi, H. Cao, Y. Wang, G. Zheng, K. Liu, N. Cheng, and M. Zhao, "DBSCAN and TD integrated Wi-Fi positioning algorithm," *Remote Sens.*, vol. 14, no. 2, p. 297, Jan. 2022.

[60] E. L. Hunt and S. Reffert, "Improving the open cluster census," *Astron. Astrophys.*, vol. 686, Jun. 2024, Art. no. A104.

[61] L. Hu, H. Liu, J. Zhang, and A. Liu, "KR-DBSCAN: A density-based clustering algorithm based on reverse nearest neighbor and influence space," *Expert Syst. Appl.*, vol. 186, Dec. 2021, Art. no. 115763.

[62] R. F. Rohrich, M. A. S. Teixeira, J. Lima, and A. S. de Oliveira, "Collective gas sensing in a cyber-physical system," *IEEE Sensors J.*, vol. 21, no. 12, pp. 13761–13771, Jun. 2021.

[63] J. M. Aitken, M. H. Evans, R. Worley, S. Edwards, R. Zhang, T. Dodd, L. Mihaylova, and S. R. Anderson, "Simultaneous localization and mapping for inspection robots in water and sewer pipe networks: A review," *IEEE Access*, vol. 9, pp. 140173–140198, 2021.

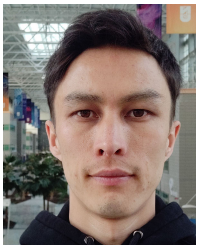
[64] T.-Y. Chuang and C.-C. Sung, "Learning and SLAM based decision support platform for sewer inspection," *Remote Sens.*, vol. 12, no. 6, p. 968, Mar. 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/6/968

[65] D. Krys and H. Najjaran, "Development of visual simultaneous localization and mapping (VSLAM) for a pipe inspection robot," in *Proc. Int. Symp. Comput. Intell. Robot. Autom.*, Jun. 2007, pp. 344–349.

[66] S. Kazeminasab, N. Sadeghi, V. Janfaza, M. Razavi, S. Ziyadidegan, and M. K. Banks, "Localization, mapping, navigation, and inspection methods in in-pipe robots: A review," *IEEE Access*, vol. 9, pp. 162035–162058, 2021.

[67] H. Lim, J. Youn Choi, Y. Sik Kwon, E.-J. Jung, and B.-J. Yi, "SLAM in indoor pipelines with 15 mm diameter," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 4005–4011.

[68] A. Kenzhekhan, A. Bakytzhanova, S. Omirbayev, Y. Tuieubayev, M. Daniyalov, and A. Yeshmukhametov, "Design and development of an in-pipe mobile robot for pipeline inspection with AI defect detection system," in *Proc. 23rd Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2023, pp. 579–584.



DARKHAN ZHOLTAYEV was born in Almaty, Kazakhstan, in 1992. He received the bachelor's and master's degrees in mathematics from Kazakh National University, in 2014 and 2017, respectively, with a focus on the mathematical underpinnings crucial to his later research in robotics and control systems, and the Ph.D. degree in robotics and mechatronics from Nazarbayev University, in 2024, specializing in model-free control methods for electric drives through advanced adaptive control techniques and deep reinforcement learning.

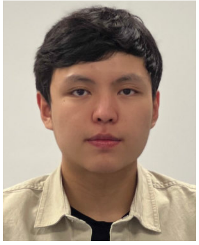
His academic foundation was built at Kazakh National University named after Al-Farabi. He has published extensively, including articles on adaptive super-twisting sliding mode control for maximum power point tracking in wind energy conversion systems in the *Renewable Energy* journal and on PMSG wind turbine control using deep reinforcement learning in *Optimal Control Applications and Methods*. His work not only advances the field of robotics but also contributes to the effective integration of renewable energy systems. His research interests include autonomous systems, computer vision, and robotics, with a particular emphasis on the application of deep reinforcement learning.

DANIYAR DAULETIYA was born in 1999. He received the Bachelor of Engineering and Technology degree majoring in instrument making (direction: mechatronics and robotics) from Satbayev University (KazNRTU named after Satpayev), in 2021, and the Master of Technical Sciences in computer engineering from KazNU named after Al-Farabi, in 2023.

He is currently the Head of the Laboratory "FabLab," Astana IT University. His main scientific works include eight publications, of which four are indexed in Scopus, and also four publications are in the list of publications recommended by the Committee for Education and Science of the Ministry of Education and Science of the Republic of Kazakhstan. He was the Winner of the Tauelsizdik Urpaktary Grant in the field of science, in 2023.

AISULU TILEUKULOVA was born in Taraz, Kazakhstan, in 1995. She received the B.S. and M.S. degrees in physics and astronomy from Al-Farabi Kazakh National University, in 2017 and 2019, respectively, where she is currently pursuing the Ph.D. degree in radio engineering, electronics, and telecommunications. She is a Research Assistant with the Advanced Robotics and Mechatronics Systems Laboratory, Nazarbayev University. Her current research interests include robotics, electrical engineering, and ML, with a focus on the development of parallel manipulators, robot design, and kinematics.

DIAS AKIMBAY was born in Kazakhstan, in 2003. He is currently pursuing the bachelor's degree in robotics engineering. He is a Research Assistant with the ARMS Laboratory. His focus is on construction, integration of in-pipe robot and manipulator for maintenance as well as controlling in distance.



MANAT NURSULTAN was born in Kazakhstan, in 2002. He received the bachelor's degree in industrial automation from Astana IT University, in 2023. He is currently pursuing the master's degree in biomedical engineering with Nazarbayev University. He is a promising Research Scholar specializing in robotics and biomedical engineering. His research interests include exoskeletons, remote control robots, bionic engineering, and biosensors, with a focus on advancing human capabilities through cutting-edge technology.

YERSAIYN BUSHANOV received the bachelor's degree in industrial automation from Astana IT University, in 2023. He is currently pursuing the master's degree in robotics with Nazarbayev University. He is a Research Scholar specializing in robotics. His research interests include mobile robots, electrical testing with tactile sensors, and the development of control systems in industrial robotics.

ASKAT KUZDEUOV received the B.S. degree in automation and control engineering from Almaty University of Power Engineering and Telecommunication, Almaty, Kazakhstan, in 2014, and the M.S. degree in robotics from Nazarbayev University, Astana, Kazakhstan, in 2019. He is currently a Senior Data Scientist with the Institute of Smart Systems and Artificial Intelligence, Nazarbayev University. Prior to that, he was a Software Engineer in the oil & gas industry, from 2014 to 2017, participating in more than 30 local and international projects. His current research interests include computer vision, speech recognition, NLP, and robotics. He served as a Technical Reviewer for IEEE Access, *Journal of Sensors*, the IEEE Conference on Decision and Control, the IEEE Industrial Electronics Society, and *Scientific Reports*.

AZAMAT YESHMUKHAMETOV was born in Kazakhstan, in 1990. He received the bachelor's degree in instrumentation and electronic engineering from Kazakh National Technical University, in 2012, and the master's and Ph.D. degrees from Tokai University, Japan, in 2016 and 2020, respectively. He is currently a Postdoctoral Scholar with the Department of Robotics Engineering, Nazarbayev University. He is a Research Scholar in the field of robotics. His research interests include wire-driven robots with hyper-redundant structures, the development of end-effector tools for inspection, and the design of agricultural machines.