Development of Underwater Pipe Crack Detection System for Low-Cost Underwater Vehicle using Raspberry Pi and Canny Edge Detection Method

Mohd Aliff[1], Nur Farah Hanisah[2], Muhammad Shafique Ashroff[3], Sallaudin Hassan[4]
Quality Engineering Research Cluster (QEREC), Universiti Kuala Lumpur, 81750, Johor, Malaysia

Siti Fairuz Nurr[5]
Faculty of Plantation and Agrotechnology
Universiti Teknologi MARA, Cawangan Melaka Kampus
Jasin, 77300 Merlimau, Melaka, Malaysia

Nor Samsiah Sani[6]
Center for Artificial Intelligence Technology
Universiti Kebangsaan Malaysia
Selangor, Malaysia

***Abstract*—The effective loading area decreases because of cracking, leading to a rise in stress and eventual structural failure. Monitoring for cracks is an important part of keeping any pipeline or building in excellent working order. There are several obstacles that make manual inspection and monitoring of subsea pipes challenging. The fundamental objective of this study is to create a relatively inexpensive underwater vehicle that can use an image processing technique to reliably spot cracks on the exteriors of industrial pipes. The tasks involved in this project include the planning, development, and testing of an underwater vehicle that can approach the circular pipes, take pictures, and determine whether there are fractures. In this project, we will utilize the Canny edge detection technique to identify the crack. The system could function in either an online or offline mode. Using a Raspberry Pi and a camera, the paper will discuss the procedures followed to locate the pipe cracks that activate the underwater vehicle. While Python is used for image processing to capture photographs, analyze images, and expose flaws in particular images, the underwater vehicle's movement will be controlled via a connected remote control. When the physical model has been built and tested, the results are recorded, and the system's benefits and shortcomings are discussed.**

Keywords*—*Crack detection; pipeline; underwater vehicle: image processing; Raspberry Pi; canny edge detection

I. Introduction

As certain countries start an economic growth period, a subsequent increase in energy demand is anticipated. The infrastructure that supports energy production expands to meet the growing needs of the world's energy consumers. Subsea natural gas production is gaining popularity as a means of accommodating this growth. As a result, there will be an even higher need for resources like human workers, capital, and expertise. These days, robots are being utilized extensively in a wide range of fields [1-6]. Even in the oil and gas industry, robotic assistance is becoming commonplace. By decreasing the need for human intervention while simultaneously boosting operational efficiency and safety, inspection robots are being employed to carry out inspection and maintenance activities on industrial property. Technology advancements in the field of inspection robotics have made the oil and gas industry more productive, secure, and dependable.

Extraction of petroleum from subterranean sources and transportation to the surface is a complex process that requires a wide range of structures and systems to work together. When conducting such tasks, it is essential to do so while minimizing costs and environmental damage as much as feasible. Transmission of oil and gas across great distances is essential, and subsea pipelines play a key role in this process [7].

From an economic, safety, and environmental perspective, pipelines could be regarded as the most ideal means of transporting petroleum fluids from underwater structures to floating manufacturing plants during exploration and from these sites to land oil refineries. The diameter and length of these pipelines, as well as operating pressures, subsea terrains, undersea conditions, and fluid properties, all have a significant impact on the price of subsea pipelines. The price of building a subsea pipeline varies depending on these factors, from several hundred dollars per kilometer to many millions of dollars per kilometer [7].

Several obstacles make it hard to inspect and monitor underwater pipelines. The area in which a pipeline may be situated for decades is complex and full of potential hazards, and problems can occur at any time during the pipe's lifetime. Damage to the pipeline might be caused by debris that falls from the topside. The pipeline might get caught on ship anchors or dragged by fishing gear that drags the ocean floor. Erosion of the soil beneath the pipeline by ocean currents can lead to free spanning, in which the pipeline is no longer supported anywhere but at its beginning and terminus. Corrosion and erosion can be caused by the inner fluids, which are often acidic and contain abrasive sand particles travelling at high speeds. Cracks in the pipe wall can occur because of such circumstances. Significant damage can be avoided if the crack or fracture is monitored on a regular basis, and the pipe's life can be extended with good treatment. Even seemingly tiny cracks can expand and finally lead to serious structural failure [8, 9]. Therefore, inspection and monitoring

activities must be carried out to ensure that no cracks or defects develop, as this might lead to serious implications for the industry, such as explosions and fires caused by the discharge of toxic gases and liquids.

In consideration of all these issues, it's crucial to constantly check pipelines and install monitoring equipment. Once the system is in place, it might be difficult to inspect the subsea pipeline due to the often-considerable water depths. Many useful inspection technologies cannot be delivered to the pipeline because they rely on costly and potentially risky equipment and methods. It would be impractical to strip away coating layers for routine inspection if the pipeline was multilayered and buried a few thousand feet below the ocean's surface. Pipelines can be inspected internally using inline inspection, but the benefit of doing so must be evaluated against the cost of shutting the pipeline down. In most cases, it would be preferable and more cost-effective to use a less invasive screening inspection or monitoring tool that provides a more comprehensive overview of the pipeline and recommends which regions may require additional attention.

Several studies on remotely operated vehicles (ROVs) for maritime research have recently been published [10-14]. In offshore sectors, such as oil and gas, marine structures, marine sciences, naval security, marine renewable energy, and research reasons, ROVs have been employed for underwater intervention, repair, and maintenance activities. ROVs are used in submarine applications to track mines and are programmed to do high-risk activities by executing algorithms for prediction, diagnosis, and classification. Other research initiatives have concentrated on underwater surveillance, and they are set up to manage continuous tasks with defined goals [10]. ROVs can also be used to capture underwater photographs, for which there are now open research lines [10]. Several detection techniques utilizing image processing with deep learning have previously been deployed by mounting cameras on ROVs to detect fractures in infrastructure such as walls and pipelines, but results can still vary due to ambient factors such as illumination [15]. As a result, researchers are becoming more interested in crack detection investigations to improve detection approaches [16-20]. Despite the tremendous improvements made in the various fields of use and development of ROVs, the wide range of specialties that come into play in obtaining maximum performance makes it a rich topic for research. The accurate sensing of underwater applications is currently a hot issue in academia.

The primary aim of this research is to develop a low-cost underwater vehicle that can accurately detect cracks on industrial pipe surfaces by using a Raspberry Pi as an embedded controller, which can be programmed using open-source software, is inexpensive, has multiprocessing capabilities, and can be programmed in Python. Several parameters, such as the optimal distance and angle between the camera and the object, will be examined in the experimental section. At the end of the paper, two offline and real-time online monitoring systems will be presented and discussed.

II. Literature Review

Multiple mode nonlinear guided waves have already been investigated and published by several academics for the purpose of detecting fatigue cracks in pipes [21]. Guan R et al. provided in-depth analyses of guided wave propagation and interaction with microcrack in a pipe structure using numerical and experimental methods. The simulation model incorporated a third-order elastic constant and a seam crack to account for material nonlinearity and CAN. Piezoelectric transducers and a nonlinear signal collection system were employed in the acquisition of wave nonlinearity experiments. Both theoretical and empirical evidence demonstrated the nonlinearity produced by a 'breathing' fracture, and the identification of several second harmonic waves was distinct from the findings of a previous investigation in a plate structure. In order to quantify nonlinearity in pipe structures, a new nonlinear index has been introduced [21].

Concerns about the stability, longevity, and functionality of a structure are often prompted by the appearance of cracks. The effective loading area decreases as fractures develop and spread, leading to a rise in stress and, finally, structural failure. Nhat-Duc Hoang proposed a model [22] for employing image processing to spot fracture faults on the exterior of structures. When dealing with digital images for fracture analysis, common problems such as low contrast, uneven illumination, and noise pollution during image processing render the conventional Otsu approach useless. Min-Max Gray Level Discrimination is an image enhancement algorithm used in the new model to enhance the Otsu method (M2GLD). The newly developed model can recognize crack objects and assess their attributes like area, perimeter, width, length, and orientation. Experiments indicate that flaws in testing images were appropriately spotted. When used in conjunction with the Otsu technique, the M2GLD can significantly improve performance [22].

Image processing techniques such as Canny edge detection can find lines suggesting fractures within the pipe and split the images to create a new image that only shows the detected fractures, allowing the fractures to be studied. The Canny edge detection method will be used to detect the fracture in this project. The Raspberry Pi will be used to control the system, which will be attached to the underwater vehicle. When compared to the slow and subjective human inspection, the image processing method for detecting cracks and flaws can offer fast and trustworthy results [23-25]. Pipe inspection has traditionally been done visually, and there are a few drawbacks that may be highlighted, including the fact that it is difficult, time-consuming, subjective to the inspector, and only provides a qualitative conclusion [26, 27].

III. Methodology

The areas of technology and science, as well as the world of computer programming, can all benefit from the use of robots. Developers and researchers are working to make robots useful in as many fields as possible [28-30]. Modern technology has led to a huge increase in the use of image processing tools in the engineering community for a variety of purposes, including measurement, robot navigation, and others. The research community and industry alike can benefit

from this technology. Robotics processing steps are shown in Fig. 1, whereas image processing steps are shown in Fig. 2. When the camera is activated, the robot will proceed through the pipe. If a camera spots a crack, it will take a picture and send it to the raspberry pi so it can be analyzed. If a fault or crack is found, it will be identified by a comparison of the two images. If the system detects a fracture, it will take a picture and record the relevant data. If no damage is found, the operation will restart with the camera. To begin the process of crack detection in Fig. 2, a camera image of the structure was first obtained. Prior to data saving, the captured image will be uploaded and processed with the Canny edge technique.

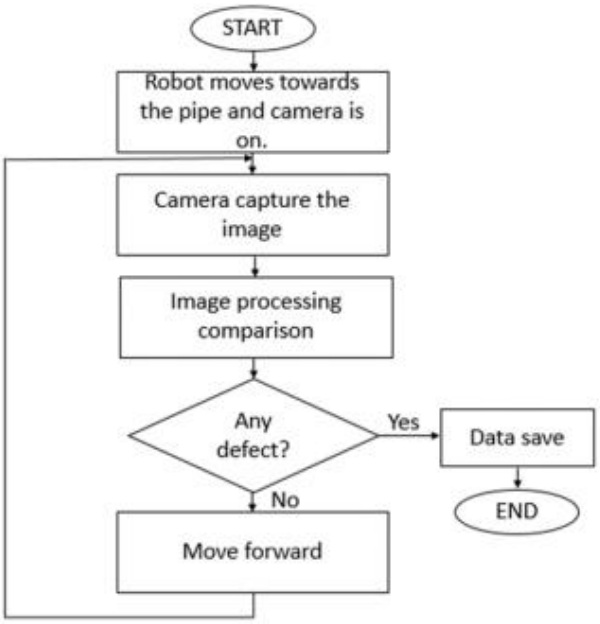

Fig. 1. Process Flow for the Robot.

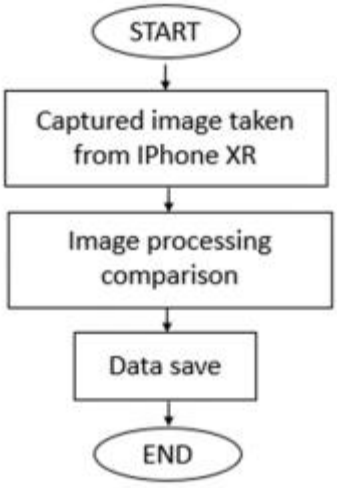

Fig. 2. Image Processing Process Flow.

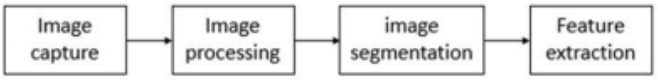

Fig. 3. Process Block Diagram.

Fig. 3 depicts the process block diagram used to identify pipe cracks. Capturing an image is the initial stage of the procedure. The online mode image is captured by the Raspberry Pi V2 NoIR camera, while the offline mode image is captured by the iPhone XR camera. The procedures will thereafter be used both online and offline. There won't be many actions taking place in the image processing phase. At first, the image will shift from its original RGB color mode to a more conventional grayscale. This brings us to our second topic: the logarithmic scale. During this procedure, the image quality improves as the number of pixels is increased by a factor of logarithm. Image smoothing comes last. The term "bilateral filter" or "blurring process" can be used to describe this technique as well. The edge of the break will be maintained during this procedure.

Moving on to the image segmentation stage, where Canny edge detection takes place. Canny edge detection typically has three goals, but the researcher here added a fourth one to boost the overall image quality. The first thing to look at is the image's intensity gradient. Each pixel's gradient vector is computed at this step. Second, the suppression that is less than maximal. To achieve a width of one pixel, the edge must be narrow. Thresholding comes next. First, the morphological operator will be reduced, and then the artificial edge. It serves to patch up the hiccups to make the main crack look more complete. To conclude, we will extract features. SIFT, SURF, and ORB are the three primary methods of feature extraction. The Oriented Fast and Rotated Brief (ORB) format was employed for this study. The object recognition relies on this.

The Raspberry Pi and ultrasonic sensor connection is shown in Fig. 4. The module can be powered by the Pi's 5V and Ground pins. In this project, the Pi's GPIO header's pins 2 and 6 were utilized. The transmission of the ultrasonic pulse is started by pressing the "trigger" input pin on the module. The general-purpose digital input/output (GPIO) uses a 3.3V signal; therefore Pin 16 is directly connected to the trigger (GPIO23). A bit more thought must be given to the module's "echo" output. The output pin is low before the module does its distance measurement (0V). It then maintains this pin high (+5V) for the same amount of time that it took for the pulse to return.

This pin's value must therefore be monitored by the script. The Pi's inputs prefer 3.3V, but the module uses a +5V level for a "high," which is too high. To make sure that the Pi is only exposed to 3.3V in this project, a straightforward voltage divider can be utilized. This is made with two resistors. If R1 and R2 are identical, the voltage is divided in half. 2.5V would be the voltage as a result. If R2 is twice as large as R1, 3.33V will result. The circuit used resistors with resistance values of 470 and 330 ohms.

- Pin 2 is directly connected to the connection for +5V.
- Pin 6 is included into the voltage divider for the Echo pin but connects directly to the GND connection.
- Pin 18 is connected to the voltage divider's center.
- Pin 16 is used to establish a direct connection to the Trig connector.

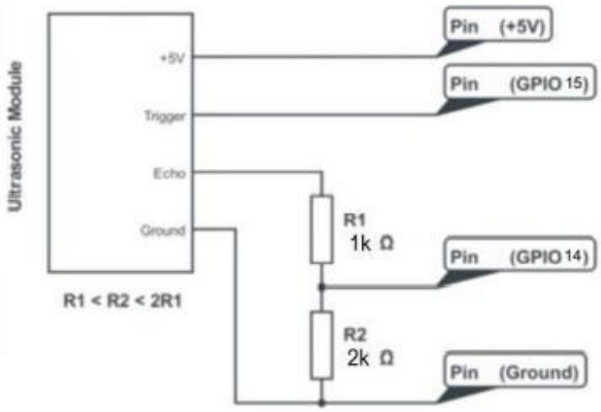

Fig. 4. Ultrasonic Sensor Connection.

For analyzing digital logic signals, the Raspberry Pi has GPIO pins for general-purpose digital input/output. The GPIO pins will receive the signal from the 'Echo' output pin. The output voltage of the Echo pin is 5V, which is unsafe for use with the Raspberry Pi's GPIO pins, which operate at 3.3V. A voltage divider is used to divide the signal between two connected resistors in a ratio equal to the resistance of the circuit, which is then fed to the Pi to generate the required voltage. As can be seen in Fig. 5, when a 5V signal is supplied from the JSN-SR04T, 1.7V is dropped across the 1k Ohm resistor, and 3.3V is dropped across the 2k Ohm resistor. Fixes and maintenance performed on the prototype of the underwater vehicle are depicted in Fig. 6.

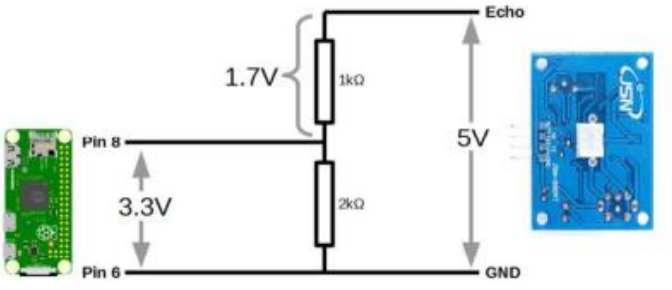

Fig. 5. Voltage Divider.

Fig. 6. Underwater Vehicle Prototype.

A. Canny Edge Detection

The basic method of edge detection has been improved using Canny edge detection. Edge tracking employing hysteresis, smoothing, gradient discovery, non-maximum suppression, and double thresholding are some of the stages that go into obtaining Canny edge detection. To remove camera noise, smoothing will inevitably damage the image. Gaussian filter application triggered the smoothing process. The fundamental idea behind the Gaussian filter was to use a standard deviation using the following equation.

$$B = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (1)$$

The standard deviation is a multiplier used to determine how many pixels in a raw image the Gaussian filter will change. This matrix filter is used to convert matrix pixels contained inside a raw image to a raw image. By looking at the region with the most change in intensity, the Canny algorithm seeks to identify the edges of the grayscale image. The gradient of each pixel was handled by the Sobel-operator. Equations following provide illustrations of the x and y gradients.

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2)$$

$$K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3)$$

Finite differences $f(x + 1, y)$ $f(x, y)$ and $f(x, y + 1)$ $f(x, y)$ were used to simplify the finite differences used to calculate the row and column values inside the matrix, where x and y were the raw image's pixel coordinates. The gradient's magnitude is defined in (3) as the angle from Euclidean that Pythagoras' law specifies. The Manhattan distance, which is derived by comparing the distances between the starting and finishing points/blocks, simplifies this calculation (4).

$$|G| = \sqrt{G\frac{2}{x} + G\frac{2}{y}} \quad (4)$$

$$|G| = |G_x| + |G_y| \quad (5)$$

The most obvious limits are shown by the magnitudes of G_y and G_x. But, at times the margins were excessively large, making it difficult to make out exactly where the borders were. To get around this limitation, it is necessary to determine and remember the edges' orientation using the equation below.

$$\theta = \tan^{-1}\left(\frac{|G_y|}{|G_x|}\right) \quad (6)$$

The raw picture was first blurred by smoothing, and then sharpened by using non-maximum suppression. The only part of the gradient picture that was kept was the local maximum. As a second step after non-maximal suppression, we used the improved pixels to label the remaining edge pixels in a

process called double thresholding. Rough surfaces can generate noise and color shifts, which could result in a loss of pixels. We performed a double thresholding method to extract the strongest edge from the image. In this case, only the edge pixels with a value higher than the thresholding value were kept, while the rest were discarded. Weak pixels were those with a threshold value between 0 and 1. In the final stage of edge detection, the strong edge was treated as the real thing. Only edges that were found to have some sort of connection to the real edge were weak. The idea behind it is that noise and other color fluctuations render a clear advantage impossible to achieve. After this point, Canny edge detection will be finished.

IV. Result and Discussion

The system may function both online and offline. Consequently, the user can opt to either upload an existing image or take a new one and process it in real time. Location of the camera used in the experiment is depicted in Fig. 7. Online mode will produce three photos, one from each of the three specified perspectives and distances. The offline option also limits users to uploading and processing just a single image at a time. Fig. 8 through Fig. 16 depicts the original iPhone XR image and the final product after The Canny edge detection was applied. Fig. 17 to 19 demonstrate what happened when we applied the Canny edge detection to data, we got from a Raspberry Pi V2 NoIR camera.

A. Distance of the Camera to the Pipe (cm) and the Angle Taken to the Pipe

The camera's optimal placement and distance from the pipe were measured. In both online and offline settings, the camera's angle relative to the pipe has been set at 45 degrees, 90 degrees, and 135 degrees, as illustrated in Fig. 7. However, in both online and offline modes, the camera distance to the pipe is fixed at 18cm, 13cm, and 7cm, respectively.

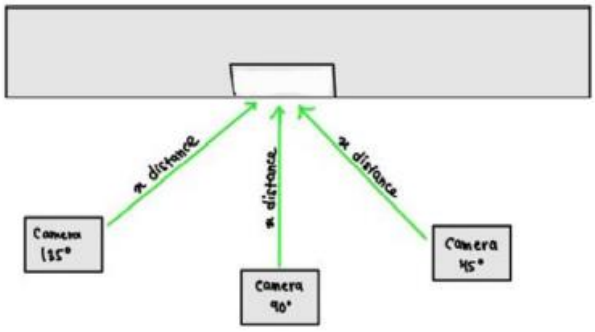

Fig. 7. Camera Angle for Testing.

TABLE I. Detection for Offline Mode

Angle (°)	Distance of the camera to the pipe (cm)		
	7	13	18
45	Detect	Detect with noise	Detect with noise
90	Detect	Detect	Detect
135	Detect with noise	Detect	Detect with noise

Table I displays the results of the offline testing conducted with the various factors discussed in the introduction. Table I demonstrates that the outcome can be detected across a wide range of parameter settings. However, when testing at 7cm with a camera at an angle of 135 degrees, 13cm with an angle of 45 degrees, 18cm with an angle of 45 degrees, and 135 degrees with an angle of 45 degrees, noise from the raw image is observed. A crack can be detected without any background noise from any distance and any camera angle. Testing in offline mode at a camera distance of 18cm at an angle of 90 degrees revealed that this was the optimal configuration for detecting pipeline cracks.

TABLE II. Detection for Online Mode

Angle (°)	Distance of the camera to the pipe (cm)		
	7	13	18
45	Detect with noise	Detect with noise	Detect with noise
90	Detect	Detect with noise	Detect with noise
135	Detect with noise	Detect with noise	Detect with noise

Table II displays the outcomes of the tests conducted in the online mode. Based on the testing, a camera positioned at a 90-degree angle and 7 cm away produces the best results. The alternative setting demonstrates that the crack is still detected, although with background noise. The camera on the Raspberry Pi V2 NoIR may have needed bright light to concentrate on the crack, which could explain why this happened.

B. Image Capturing with a Regular Camera for Monitoring in Offline Mode

The offline mode will enable computers input and output data quickly. As a result, relatively slow input devices are no longer required. Instead, the information is kept as files on a fast data storage system. The primary processing computer does not immediately take control of and read the data from its input devices. The data is prepared, kept on a high-speed storage device separate from the computer, and then made available as required. In this mode, the user can select the preferred camera and still apply the same algorithm. From Fig. 8 to Fig. 16, the result may be observed in detail as indicated in Table I.

Distance between the camera and the pipe (7 cm) (45°).

The outcome for a camera to pipe distance of 7 cm and a 45° angle is shown in Fig. 8. The results clearly demonstrate crack detection without any noise.

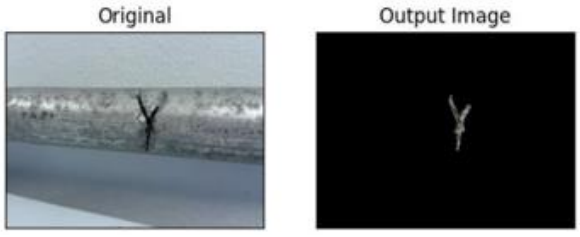

Fig. 8. Camera to Pipe Distance is 7cm and Angle is 45°.

1) Distance between the camera and the pipe (7 cm) (90°): The result is shown in Fig. 9 for a camera-to-pipe distance of 7 cm at a 90° angle. Crack detection is plainly demonstrated by the results.

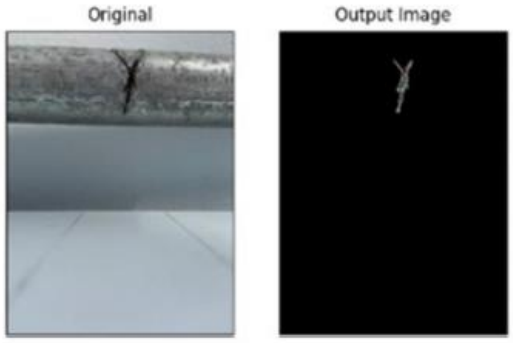

Fig. 9. Camera to Pipe Distance is 7cm and Angle is 90°.

2) Distance between the camera and the pipe (7 cm) (135°): Fig. 10 depicts the outcome with a camera-to-pipe distance of 7 cm and an angle of 135 degrees. The results indicate the detection of cracks with some background noise.

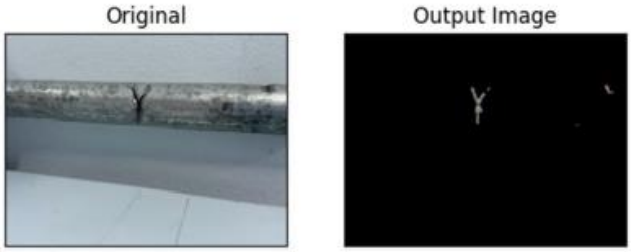

Fig. 10. Camera to Pipe Distance is 7cm and Angle is 135°.

3) Distance between the camera and the pipe (13 cm) (45°): The outcome for a 13 cm pipe to camera distance with a 45° angle is shown in Fig. 11. The results demonstrate the crack's detection with some noise.

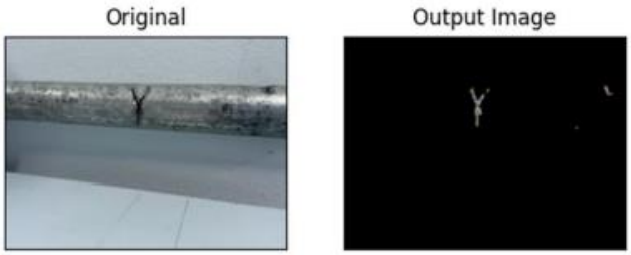

Fig. 11. Camera to Pipe Distance is 13cm and Angle is 45°.

4) Distance between the camera and the pipe (13 cm) (90°): The outcome for a 13 cm camera to pipe distance at a 90° angle is shown in Fig. 12. The outcome demonstrates crack detection.

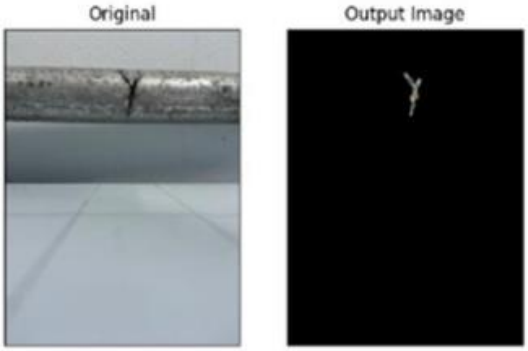

Fig. 12. Camera to Pipe Distance is 13cm and Angle is 90°.

5) Distance between the camera and the pipe (13 cm) (135°): The outcome with a camera to pipe distance of 13 cm and a 135° angle is shown in Fig. 13. The outcome demonstrates crack detection.

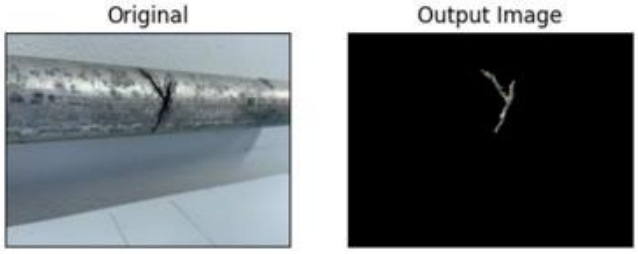

Fig. 13. Camera to Pipe Distance is 13cm and Angle is 135°.

6) Distance between the camera and the pipe (18 cm) (45°): The outcome for a camera to pipe distance of 18 cm and a 45° angle is shown in Fig. 14. The results demonstrate the crack's detection with some noise.

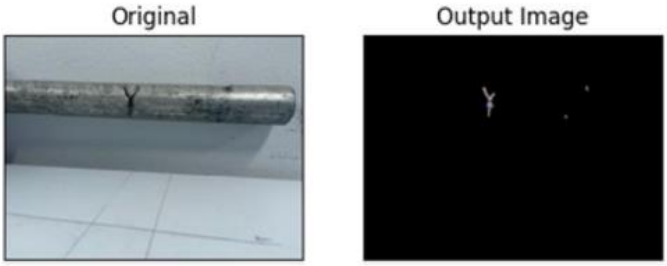

Fig. 14. Camera to Pipe Distance is 18cm and Angle is 45°.

7) Distance between the camera and the pipe (18 cm) (90°): Fig. 15 displays the outcome for an 18 cm camera to pipe distance at a 90° angle. The outcome demonstrates crack detection.

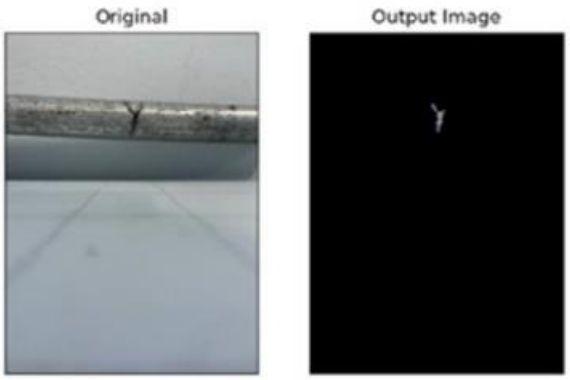

Fig. 15. Camera to Pipe Distance is 18cm and Angle is 90°.

8) Distance between the camera and the pipe (18 cm) (135°): Fig. 16 displays the outcome for an 18 cm camera to pipe distance at a 135° angle. The results demonstrate the crack's detection with some noise.

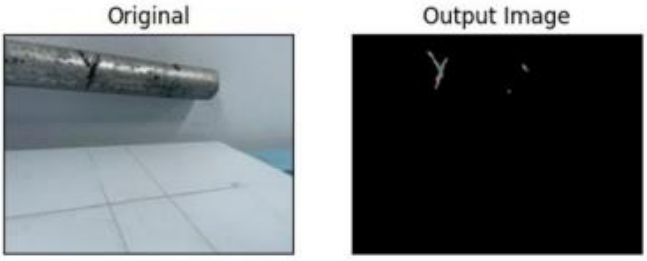

Fig. 16. Camera to Pipe Distance is 18cm and Angle is 135°.

C. Image Capturing with the Raspberry Pi V2 Module NoIR Camera for Monitoring in Real-Time Online Mode

When a system is connected to a computer and processing data files while using input, output, and storage devices, the process is referred to as real-time "online" processing. The system will take three photographs, giving each image taken a five-second pause period. With the same distance, the three photos have different angles.

The results of an online test for a 7 cm camera distance from the sample are shown in Fig. 17. When the camera is at a 90° angle, feature #2 displays the result without any noise, however testing at 45° and 135° results in some noise in the processed image.

1) Distance between the camera and the pipe (7 cm) (45°), (90°) and (135°): Fig. 18 displays the outcomes of an online test for a 13 cm camera distance from the sample. The crack is visible in the photograph after evaluating it from three different angles, but the results have significant noise. Regarding Fig. 19, the testing is done at an 18 cm camera-to-pipe distance. The crack is still detectable in the results, but there is significantly more noise than at a 13 cm distance. To compare to this chart, use the simplified data from Table II as a guide.

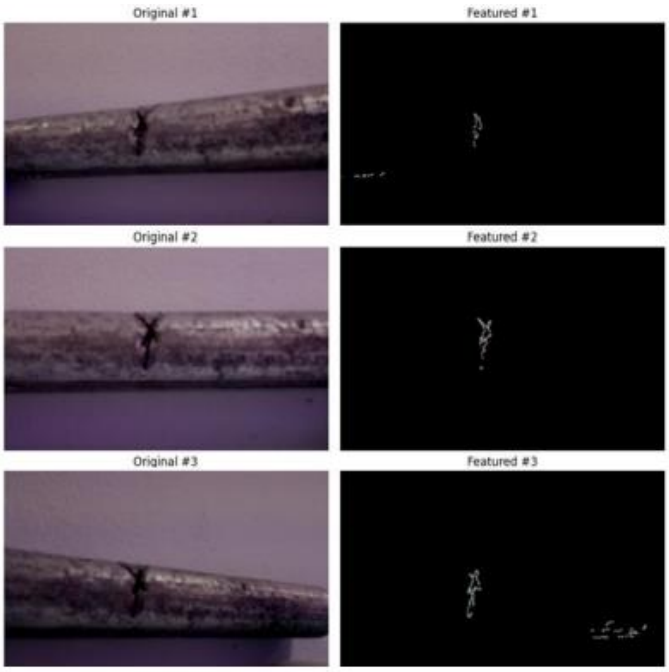

Fig. 17. Camera to Pipe Distance is 7cm and Angle #1 45°, #2 90° and #3 135°.

2) Distance between the camera and the pipe (13 cm) (45°), (90°) and (135°)

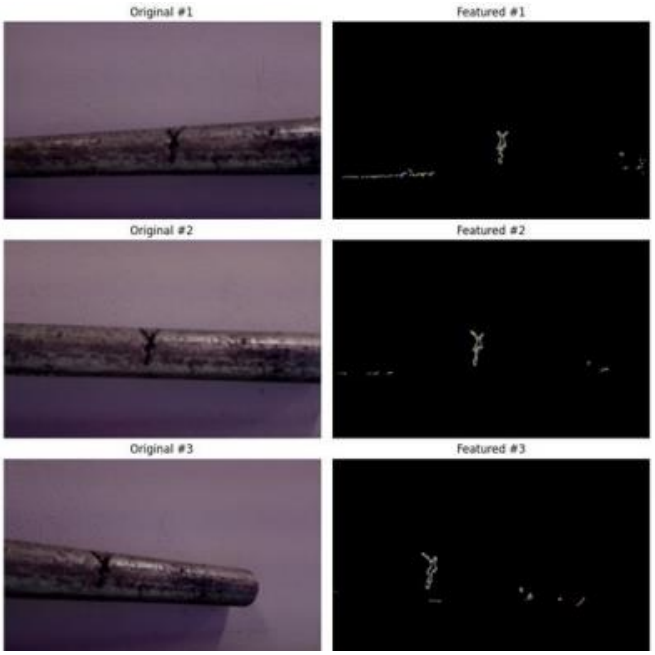

Fig. 18. Camera to Pipe Distance is 13cm and Angle #1 45°, #2 90° and #3 135°.

3) Distance between the camera and the pipe (18 cm) (45°), (90°) and (135°)

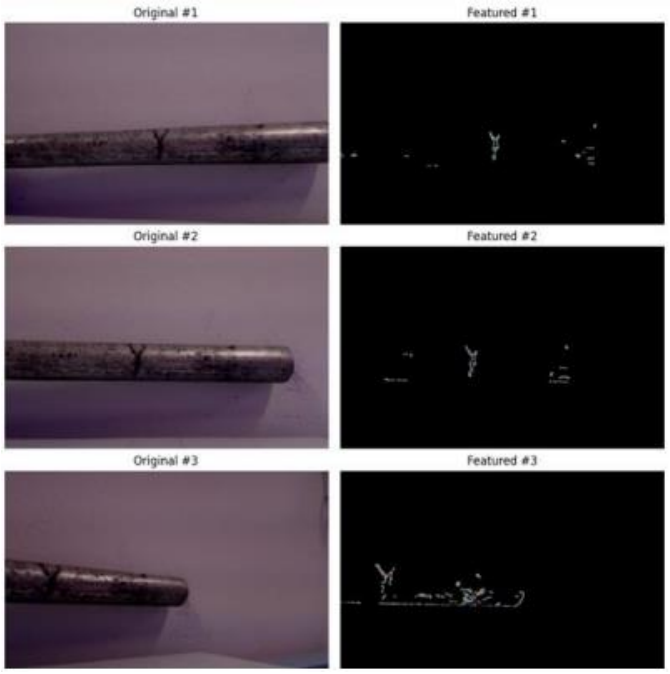

Fig. 19. Camera to Pipe Distance is 18cm and Angle #1 45°, #2 90° and #3 135°.

V. Conclusion

The project's purpose is to help the user detect pipeline faults under water by developing an underwater pipe crack detection system for a low-cost underwater vehicle using a Raspberry Pi and Canny edge detection. Online and offline modes are the two possible techniques. The cracks on the pipe are detected by both modes using the same method, the Canny edge detection.

The primary goal of this project is to use Python on a Raspberry Pi to implement the Canny edge detection crack technique. The algorithm Canny edge detection is employed in coding. Images that are captured online and those that are uploaded offline follow the approach depicted in Fig. 8 to 16. The picture segmentation step three includes the Canny edge detection. Canny edge detection has three fundamental steps, but for this project, one more step is included to make sure that the pipe fracture can be seen clearly. Finding the image's intensity gradient is the first step in the Canny edge detection procedure. Each pixel's gradient vector is computed throughout this process. The second process is called non-maximum suppression. The edge will become narrow thanks to this technique, giving it a one-pixel width. Thresholding is the final step. The artificial edge will be reduced in this stage, and the next step, the morphological operator, will fill in the little gaps to give the main crack more character.

The project's second goal is to create a low-cost underwater vehicle Remotely Operated Vehicle (ROV) prototype that can find cracks in industrial pipelines. Prototypes for vehicles are created with inexpensive but capable materials. The vehicle's mobility, which uses its propulsion system, is one of the precautions. The vehicle's mobility is controlled by the motor or turbine propulsion system. As a result, it is difficult for the user to control.

The project's final goal is to compare the two types of offline and online crack detection methods utilizing image processing. In the system, there are two operating modes: online mode and offline mode. Both modes use the same algorithm but a distinct concept, as is mentioned below. When a user uploads a picture into the system via the online mode, the system analyses the image as described in Fig. 17 through Fig. 19. The system will take three photographs, giving each image taken a five-second pause period. The three photos are at various distances or angles. The user can submit an image into the system for offline mode, and the system will immediately analyze the image. However, only one image may be posted at a time.

The camera utilized for the online approach to find pipe cracks must be improved due to the project's constraints. The camera cannot concentrate on the crack since the Raspberry Pi V2 NoIR camera required the proper lighting. The ultrasonic sensor is up next. It is impossible to determine the precise distance between the camera and the pipe with this ultrasonic sensor. This is due to the UT sensor's high quality. Consequently, it was necessary to purchase a high-quality UT sensor.

Acknowledgment

The authors gratefully acknowledge to the Ministry of Higher Education (MoHE) Malaysia for financial supports given under the Fundamental Research Grant Scheme (FRGS/1/2019/TK04/UNIKL/02/11).

References

[1] M. Aliff, S. Dohta, and T. Akagi, “Trajectory controls and its analysis for robot arm using flexible pneumatic cylinders,” 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 2015, pp. 48–54.

[2] M. Aliff, M. A. Dinie, I. Yusof, and N. Samsiah, “Development of Smart Glove Rehabilitation Device (SGRD) for Parkinson's Disease,” International Journal of Innovative Technology and Exploring Engineering, 9(2), 2019, pp. 4512–4518.

[3] K. Kusunose, T. Akagi, S. Dohta, W. Kobayashi, T. Shinohara, Y. Hane, K. Hayashi, and M. Aliff, “Development of Inchworm Type Pipe Inspection Robot using Extension Type Flexible Pneumatic Actuators,” International Journal of Automotive and Mechanical Engineering, 17 (2), 2020, pp. 8019–8028.

[4] N. S. Sani, I. Shlash, M. Hassan, A. Hadi, and M. Aliff, “Enhancing Malaysia Rainfall Prediction Using Classification Techniques,” Journal of Applied Environmental and Biological Sciences, 7 (2s), 2017, pp. 20–29.

[5] A. Syamim, M. Aliff, M. Ismail, S. Izwan, N. Samsiah, and M. Usairy Syafiq, “Application of Fuzzy Logic in Mobile Robots With Arduino and IoT,” 2022 7th International Conference on Automation, Control and Robotics Engineering (CACRE), 2022, pp. 82–86.

[6] M. Aliff, M. Imran, S. Izwan, M. Ismail, N. Samsiah, T. Akagi, S. Dohta, W. Tian, S. Shimooka, and A. Athif, “Development of Pipe Inspection Robot using Soft Actuators, Microcontroller and LabVIEW,” International Journal of Advanced Computer Science and Applications, 13(3), 2022, pp. 349–354.

[7] M. Ho, S. El-Borgi, D. Patil, and G. Song, “Inspection and monitoring systems subsea pipelines: A review paper,” Structural Health Monitoring, 19(2), 2020, pp. 606–645.

[8] A. M. A. Talab, Z. Huang, F. Xi, and L. HaiMing, “Detection crack in image using Otsu method and multiple filtering in image processing techniques,” Optik, 127(3), 2016, pp. 1030–1033.

[9] D. Dhital, and J. R. Lee, "A Fully Non-Contact Ultrasonic Propagation Imaging System for Closed Surface Crack Evaluation," Exp Mech, 52, 2012, pp. 1111–1122.

[10] O. A. Aguirre-Castro, E. Inzunza-González, E. E. García-Guerrero, E. Tlelo-Cuautle, O. R. López-Bonilla, J. E. Olguín-Tiznado, and J. Cárdenas-Valdez, "Design and Construction of an ROV for Underwater Exploration," Sensors, 19(24), 2019, pp. 5387.

[11] M. Aliff, N. Raihan, I. Yusof, and N. Samsiah, "Development of Remote Operated Vehicle (ROV) Control System using Twincat at Main Control Pod (MCP)," International Journal of Innovative Technology and Exploring Engineering, 8(12), 2019, pp. 5606–5610.

[12] M. Aliff, N. Firdaus, N. Rosli, M. I. Yusof, N. Samsiah, and S. Effendy, "Remotely Operated Unmanned Underwater Vehicle for Inspection," International Journal of Innovative Technology and Exploring Engineering, 9(2), 2019, pp. 4644–4649.

[13] M. S. A. M. Nor, M. Aliff, and N. Samsiah, "A Review of a Biomimicry Swimming Robot using Smart Actuator," International Journal of Advanced Computer Science and Applications, 12(11), 2021, pp. 395–405.

[14] M. Aliff, A. R. Mirza, M. Ismail, and N. Samsiah, "Development of a Low-Cost Bio-Inspired Swimming Robot (SRob) with IoT," International Journal of Advanced Computer Science and Applications, 12(7), 2021, pp. 452–457.

[15] Y. Hamishebahar, H. Guan, S. So, and J. Jo, "A Comprehensive Review of Deep Learning-Based Crack Detection Approaches," Applied Sciences, 12(3), 2022, pp. 1374.

[16] Z. Wang, M. Liu, Y. Qu, Q. Wei, Z. Zhou, Y. Tan, L. Hong, and H. Song, "The Detection of the Pipe Crack Utilizing the Operational Modal Strain Identified from Fiber Bragg Grating," Sensors, 19(11), 2019, pp. 2556.

[17] D. Rifai, A. N. Abdalla, R. Razali, K. Ali, and M. A. Faraj, "An Eddy Current Testing Platform System for Pipe Defect Inspection Based on an Optimized Eddy Current Technique Probe Design," Sensors, 17(3), 2017, pp. 579.

[18] X. Xu, M. Zhao, P. Shi, R. Ren, X. He, X. Wei, and H. Yang, "Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN," Sensors, 22(3), 2022, pp. 1215.

[19] Y. Zhao, Y. Han, C. Chen, and H. Seo, "Crack Detection in Frozen Soils Using Infrared Thermographic Camera," Sensors, 22(3), 2022, pp. 885.

[20] Z. Shi, X. Xu, J. Ma, D. Zhen, and H. Zhang, "Quantitative Detection of Cracks in Steel Using Eddy Current Pulsed Thermography," Sensors, 18(4), 2018, pp. 1070.

[21] R. Guan, Y. Lu, K. Wang, and Z. Su, "Fatigue crack detection in pipes with multiple mode nonlinear guided waves," Structural Health Monitoring, 18(1), 2019, pp. 180–192.

[22] N. D. Hoang, "Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding," Advances in Civil Engineering, 2018, pp. 1–10.

[23] N. D. Hoang and Q. L. Nguyen, "Metaheuristic Optimized Edge Detection for Recognition of Concrete Wall Cracks: A Comparative Study on the Performances of Roberts, Prewitt, Canny, and Sobel Algorithms," Advances in Civil Engineering, 2018, pp. 1–16.

[24] N. D. Hoang, Q. L. Nguyen, and D. T. Bui, "Image Processing-Based Classification of Asphalt Pavement Cracks Using Support Vector Machine Optimized by Artificial Bee Colony," Journal of Computing in Civil Engineering, 32 (5), 2018.

[25] R. S. Adhikari, O. Moselhi, and A. Bagchi, "Image-based retrieval of concrete crack properties for bridge inspection," Automation in Construction, 39, 2014, pp. 180–194.

[26] N. M. Syahrian, P. Risma, and T. Dewi, "Vision-Based Pipe Monitoring Robot for Crack Detection Using Canny Edge Detection Method as an Image Processing Technique," Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 2(4), 2017, pp. 243–250.

[27] X. Wu, Y. Jiang, K. Masaya, T. Taniguchi, and T. Yamato, "Study on the Correlation of Vibration Properties and Crack Index in the Health Assessment of Tunnel Lining," Shock and Vibration, 2017, pp. 1–9.

[28] T. Morimoto, M. Aliff, T. Akagi, and S. Dohta, "Development of Flexible Pneumatic Cylinder with Backdrivability and Its Application," International Journal of Materials Science and Engineering, 3(1), 2015, pp. 7–11.

[29] K. Hayashi, T. Akagi, S. Dohta, W. Kobayashi, T. Shinohara, K. Kusunose, and M. Aliff, "Improvement of Pipe Holding Mechanism and Inchworm Type Flexible Pipe Inspection Robot," International Journal of Mechanical Engineering and Robotics Research, 9(6), 2020, pp. 894 – 899.

[30] M. Aliff, S. Dohta, T. Akagi, T. Morimoto, "Control of Flexible Pneumatic Robot Arm Using Master Device with Pneumatic Brake Mechanism," JFPS International Journal of Fluid Power System, 8(1), 2014, pp. 38–43.