APPLICATION OF DEEP NEURAL NETWORK ON AUTONOMOUS UNDERWATER VEHICLES (AUV) IN SUBSEA PIPELINE INSPECTION

Ong Yi Kai[1], Siow Chee Loon[2]*, Jaswar Koto[3], Istas Nusyirwan[4]

[1]Faculty of Mechanical Engineering Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor
[2]Marine Technology Center, Institute for Vehicle Systems and Engineering (IVeSE), Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor
[3]Universitas Insan Cita, Indonesia.
[4]Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor

Article history
Received
8th December 2022
Received in revised form
7th December 2023
Accepted
8th December 2023
Published
28th December 2023

*Corresponding author
scheeloon@utm.my

ABSTRACT

This study aims to investigate the application of a deep neural network for the Automated Underwater Vehicle in the subsea pipeline inspection. In today's modern world, we see more sophisticated and precise computer vision object detection technology being implemented in our daily lives. To name a few, security cameras, self-driving cars, drones and more. This research suggests that computer vision pipeline defect detection is an attractive solution for future subsea pipeline defect detection as it relies less on human intervention and is more reliable. The machine learning algorithm of our focus, Faster RCNN, is studied. We explained the methods of our experimentation and the training process and validated the custom dataset. The outcomes are separated into different sections: training curve, visual data representation, and model accuracy in different operating environments. In conclusion, our model can detect the underwater pipeline with a leak size of up to 1mm. However, the low accuracy due to the insufficient dataset is recognized as a bottleneck, and some recommendations are suggested for future improvement.

KEYWORDS

Machine learning, subsea pipeline leak, defect detection, computer vision, AUV

INTRODUCTION

It is well-known that offshore leaks have been and still are a problem humanity face. Whenever the mass media reports any leak, giant corporations are the ones to be criticized and blamed. According to GESAMP, in 2007, the number of offshore spills increased from an average of 47 per annum since 1968 to 188 ruptures and 228 leaks, respectively. Amongst the offshore leaks, pipeline ruptures are one of the main causes of the offshore disaster. Statistics have shown that from the 1970s to the 2000s, pipeline ruptures increased at an alarming rate and contributed to almost 37% of the spills in number and 26% in volume at their peak (Burgherr, 2007), as shown in Figure 1. Meanwhile, in 2010, a report released by the U.S. Minerals Management Services revealed that the number has quadrupled since the 90s. In the first decade of the millennium, 350 pipeline spills happened worldwide.

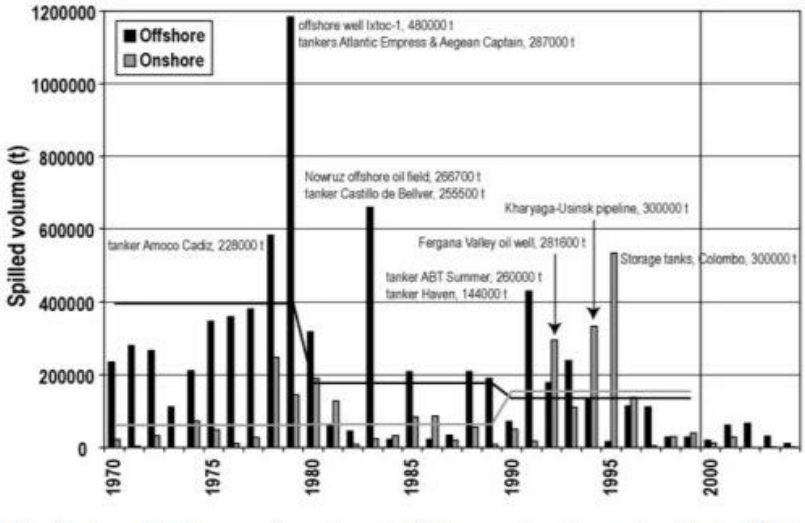

Figure 1: Onshore and offshore spill volume from 1970 to 2000 (Burgherr,2007)

It is important to point out that the rise of offshore leaks is due to the sharp increase in the total number and length of offshore pipelines deployed due to more offshore activities and the increased demand for gas and petrol. On the other hand, it has also been pointed out that the poor maintenance of the pipelines has also been identified as the cause behind the leaks. On one occasion, in tropical West Africa, the poor maintenance and corrosive environment have led to several ruptures and spills. Besides, during the reign of the Soviet Union, this was especially true when numerous damaged pipes were idled for years with minimal effort to dig out ditches to contain the contamination.

The literature covers offshore visual inspection methods, emphasizing in-line inspection with Non-Destructive Testing (NDT) methods like magnetic flux leakage (MFL) testing, ultrasonic testing (UT), closed circuit television camera (CCTV), eddy current testing (EC) and electromagnetic acoustic technology (EMAT) for defect detection. CCTV, predominantly via robots, is common. Computer vision aids data analysis, though limitations exist in capturing subsurface defects. Safizadeh (2012) proposed an optical system for corrosion detection, with documented limitations (Ma et al., 2021). Thermal imaging by Vrana et al. (2008) and Oswald et al. (2007) detects defect cracks effectively for various applications. Chen et al. (2019) introduced eddy current pulse thermal imaging.

Sewage pipelines use machine learning with CCTV for defect detection, suggesting cross-industry insights. In today's modern world, we see more sophisticated and precise computer vision object detection technology being implemented in our daily lives. To name a few, security cameras, self-driving cars, drones and more. This technology advancements suggest that computer vision pipeline defect detection can be an attractive solution for future subsea pipeline defect detection as it relies less on human intervention and is more reliable. Therefore, as a first approximation, we study the adoption of deep neural network on an experimental inspection data set obtained under controlled environment for pipeline defect detection.

METHODOLOGY

Different PVC pipe sizes were prepared beforehand with diameters of 0.5 in, 1.0 in, and 1.5 in. The pipes were then submerged into a 40 cm × 40 cm × 70 cm dimension of a water tank. The illumination of the surroundings was measured with an infrared sensor and was recorded. As a first approximation, the underwater camera was then submerged to capture the data. A measuring tape measured the distance between the camera and the pipe to ensure the distance as per requirement. On the other hand, the defect of the pipe is made in different leak sizes, such as 1 mm, 2 mm, and 3 mm, as illustrated in the table below. The defect size is also measured with the measuring tape. In this study, two data sets were taken for each training and validation use. The training data are random, while the validation data are required to follow specific fixed criteria. The validation criteria are given in Table 1. Overall, 289 data are taken, of which 244 are for training and 45 for validation.

Table 1: Validation Criteria

No.	Lux	Diameter (in)	Height (cm)	Joint	Leak Size (mm)
1	<100	5	20	yes	
2	<100	5	20	yes	
3	<100	5	20	no	
4	<100	5	20	no	
5	<100	5	20	no	
6	>100	5	40	no	20
7	>100	5	40	no	20
8	>100	5	40	no	20
9	>100	5	40	no	20
10	>100	5	40	no	20
11	>100	5	60	no	10
12	>100	5	60	no	10
13	>100	5	60	no	10
14	>100	5	60	no	10
15	>100	5	60	no	10
16	<100	10	20	no	10
17	<100	10	20	no	10
18	<100	10	20	no	10
19	<100	10	20	no	10
20	<100	10	20	no	10
21	>100	10	40	no	
22	>100	10	40	no	
23	>100	10	40	no	
24	>100	10	40	no	
25	>100	10	40	no	
26	<100	10	60	no	30
27	<100	10	60	no	30
28	<100	10	60	no	30
29	<100	10	60	no	30
30	<100	10	60	no	30
31	<100	15	20	no	10
32	<100	15	20	no	10
33	<100	15	20	no	10

No.	Lux	Diameter (in)	Height (cm)	Joint	Leak Size (mm)
34	<100	15	20	no	10
35	<100	15	20	no	10
36	>100	15	40	no	20
37	>100	15	40	no	20
38	>100	15	40	no	20
39	>100	15	40	no	20
40	>100	15	40	no	20
41	>100	15	60	no	
42	>100	15	60	no	
43	>100	15	60	no	
44	>100	15	60	no	
45	>100	15	60	no	

The collected data shall be fed into the codes for training and validation. Before this, labeling and normalizing are required. This is done in the Linux bash shell. Firstly, a heif-convert function is used to convert the images into jpgs. Then, exiftool is used to remove the exif data in the image data, which will cause trouble during the training stage. Next, the processed images are labeled manually using the coco-annotator provided by J.Brooks from the University of Guelph, Canada. Manual data labeling is essential in every part of object detection custom dataset training. A handy tool can save much time and effort for such a process. The labeled data will have their json file containing the annotation information in the COCO dataset format. This information will be useful as it will tell the machine learning algorithm where the ground-truth object is and segment them from their background. The validation data also requires these annotation data to cross-check its prediction with the json data. It is important to mention that there are two classes of data labeled during the annotation: the “Pipe” and the “Defect.” Two classes are labeled with bounding box and polygon segmentation.

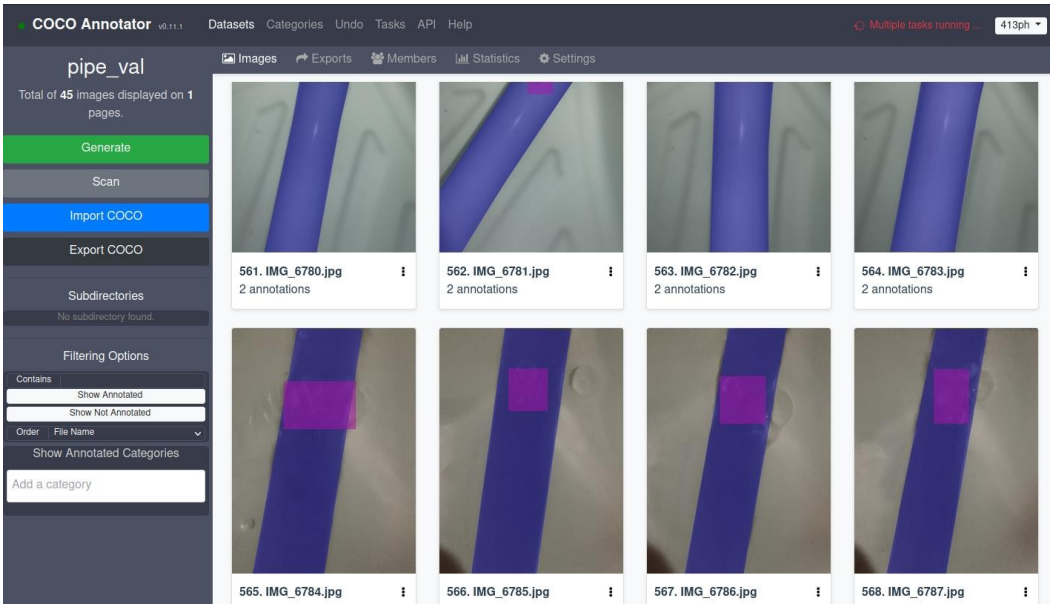

Figure 2: Data Annotation

2.0 Training and Validation

The data is trained in the Google COLAB, an interactive Python shell with dedicated GPU, CPU, and RAM for users to run their codes, specifically for machine learning model training. The dataset is separated into 2 for training and validation purposes. The program will then give the output by using the trained memory. The results are done by answer-checking with the validation dataset to see how many scores the model gets. Python uses an extensive library provided by Facebook Research and Detectron2 by Y.X. Wu et al. (2019). We will use its pre-trained model baseline for our custom dataset to improve the model accuracy. The selected baseline in this case will be Faster RCNN DC5 3x. All the annotation files are produced beforehand, and the images are referred to as datasets.

The number of iterations and batch size are adjusted at 300 and 128 separately. This is done by trial and error until the training curve is optimized. Generally, an iteration indicates the number of times a batch of data is passed through the algorithm, while a batch size is the number of data passed to the algorithm per iteration. In addition, an epoch describes the entire dataset passed through the algorithm. As discussed before, the under-fitting and over-fitting of a model shall be avoided. Hence, the trial-and-error method is used to find the optimum point for allocating the iteration and batch size. The validation results will then assess the trained model. The tensorboard and the image visualization function visualize the results.

RESULTS AND DISCUSSION

.

3.0 Tensorboard

The training and validation results are displayed on the tensorboard, a training visualization tool provided by Google, TensorFlow. We will discuss several parameters, such as class accuracy, false negative, foreground class accuracy, total loss, and class loss. Class accuracy indicates the model's accuracy in determining the different classes in the validation process. In contrast, foreground class accuracy refers to how well the model segments the object from its background. The false negative refers to when the output is expected to be positive, but the model gives a negative result. From the tensorboard, in Figure 3, we observe the trend where both accuracies of the model increase as the iteration increases until its peak at iteration 300 (class accuracy is 0.96 while foreground class accuracy is 0.7). On the other hand, the false negative drops to 0 at iteration 280. Looking at the loss curve in Figure 4, the losses of the model reduce as expected with the increase of iteration from 1.9 to 0.4 and from 1.1 to 0.1 for total loss and class loss, respectively.

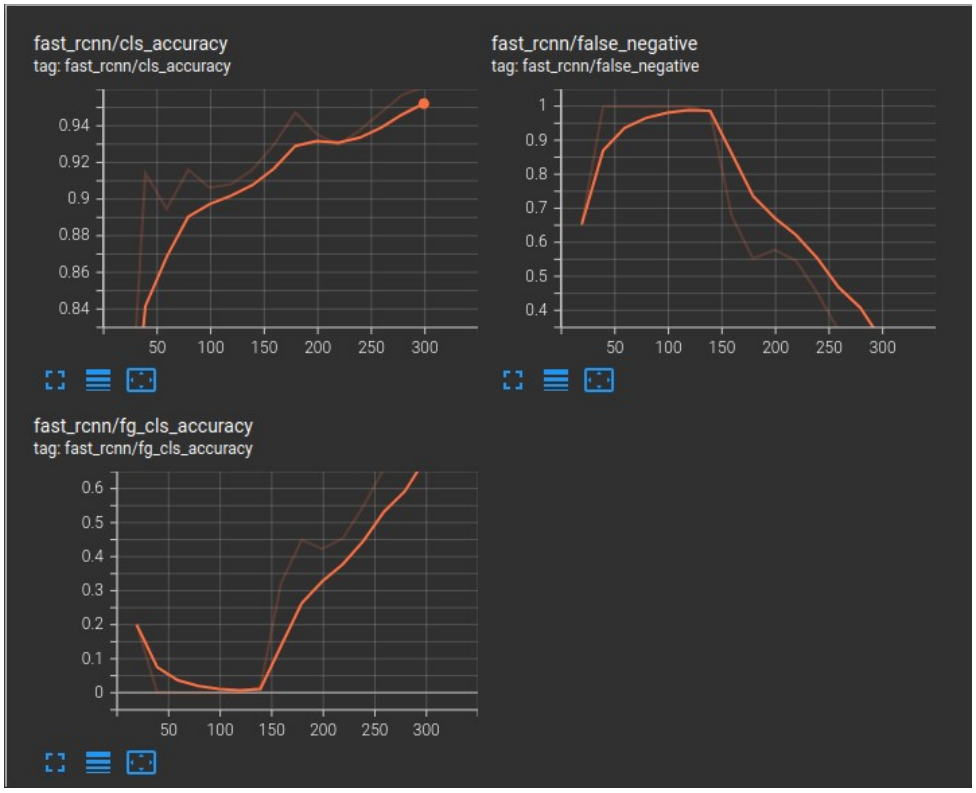

Figure 3: Accuracy of model

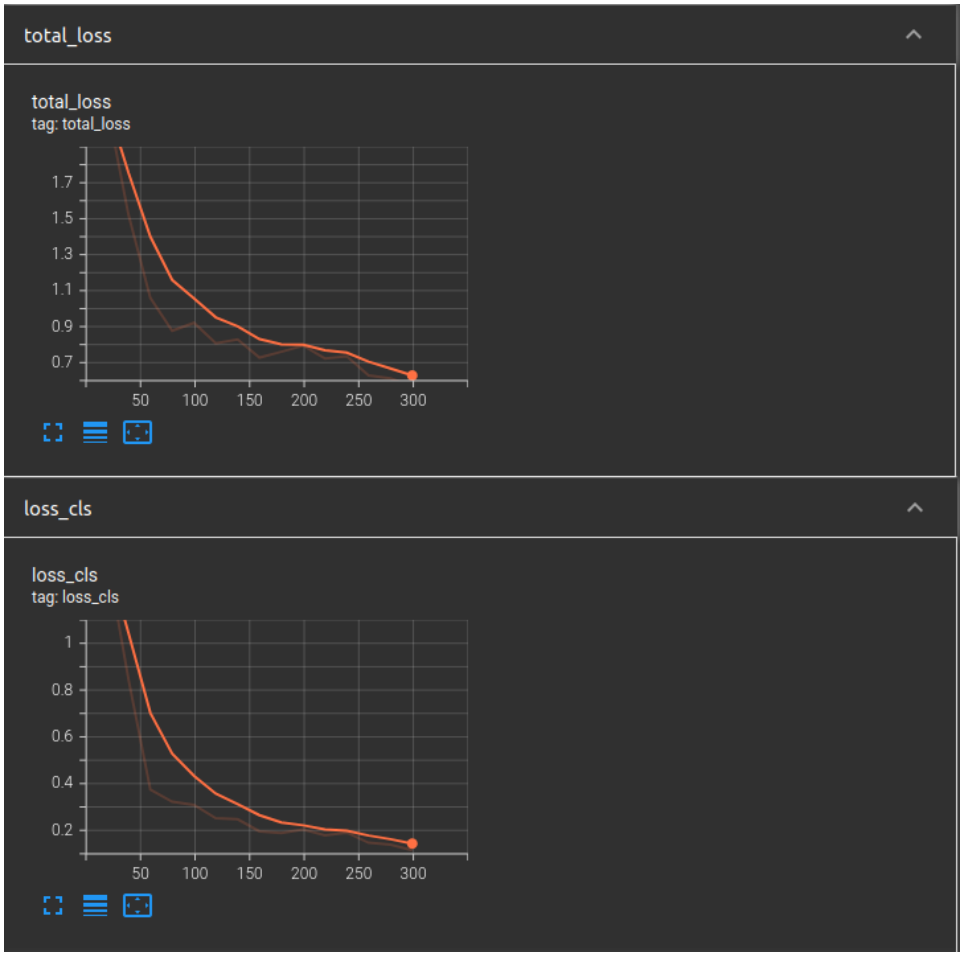

Figure 4: Losses of model

3.1 Visualization

From the validation process, we can output the segregated data with labels done by the algorithm. The bounding box will highlight the selected object and indicate the class it belongs to in the top left corner. There are some outputs to share as in Figure 5. Do note that the color of the bounding box does not carry any representation meaning and is random.

(a) Output a (b) Output b (c) Output c

Figure 5: Validation image

3.3 Comparison in Different Environments

This section will assemble the validation results and display them graphically according to their groups. From the results, we know that the bigger the pipe diameter, the easier it is for object detection to identify the leaks in the pipe as shown in Figure 6. It shall be noted that the low accuracy on the 1.5 pipe diameter is due to the different heights, which will be discussed later. Also, from the evidence gathered, we can confirm that with greater illuminance (see Figure 7), object detection works better in identifying the leaks in the pipe. Besides, the graph in Figure 8 shows that for the camera closer to the pipe up to 20cm, the detection accuracy increases to 0.8 but drops to 0.67 when the height increases to 40cm. When the camera height is 80cm, the accuracy fails to 0.20 due to the difficulty in finding the leaks as highlighted in Figure 9. This may also be caused by the lower resolutions of the pixels captured by the camera.

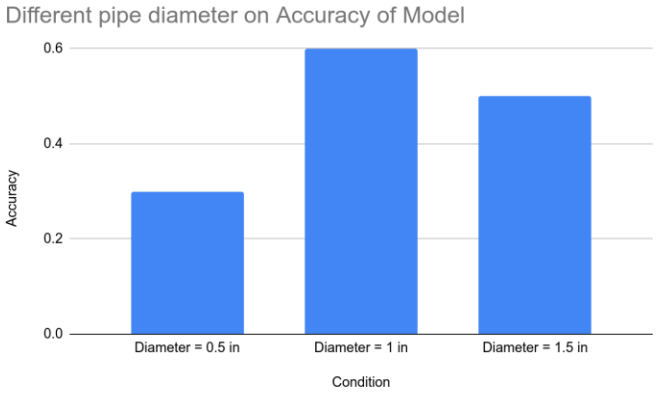

Figure 6: Accuracy for different pipe diameter

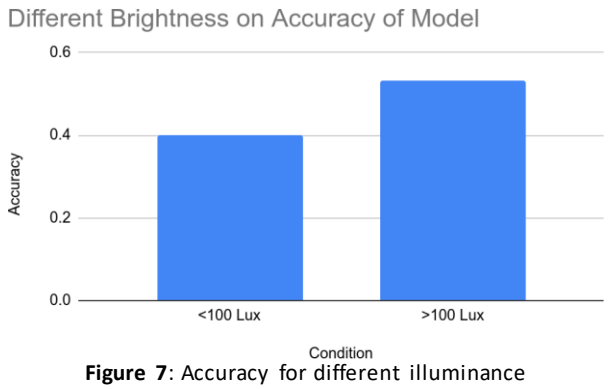

Figure 7: Accuracy for different illuminance

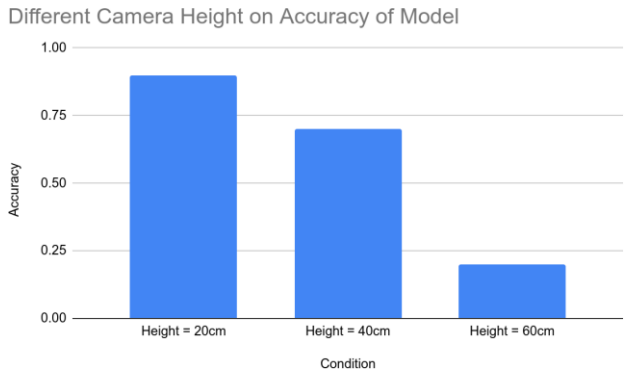

Figure 8: Accuracy for different camera heights

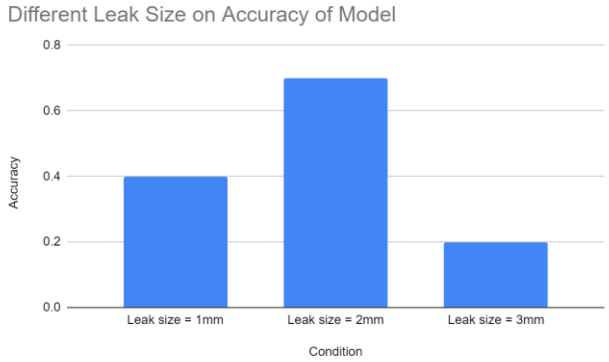

Figure 9: Accuracy for different leak sizes

Lastly, the leak size is found to be best detected at 2mm with an accuracy of 0.7. The 1 mm leak size is harder to detect due to its smaller dimensions. The reason behind the low accuracy for the 3mm leak size is the absence of data on the leak portrayed on the pipe. The different defect patterns may also cause this low accuracy compared to the 1mm and 2mm leaks. The defect pattern on the 3mm pipe is noticed to look like a punctuated shape, while the other ones look like cracks. This also indicates that more data with different defect patterns is required for the training to improve object detection accuracy.

CONCLUSION

In conclusion, our research demonstrates the successful training of a machine learning model for deep water pipeline defect inspection, showcasing its proficiency in capturing defects as small as 1mm. However, challenges arise with a 3mm leak size due to a distinct crack pattern not covered in the training dataset. To enhance overall accuracy, diversifying the training dataset becomes imperative, allowing the model to adapt to various scenarios and mitigating the risk of overfitting.

Furthermore, environmental factors significantly impact the model's accuracy, revealing that higher illuminance positively influences object detection. Proximity of the camera to the pipeline and larger pipe sizes contribute to better precision in object detection.

As we conclude, we would like to advocate for further exploration in this field. Future research should encompass a broader range of datasets to mirror real-world conditions, incorporating elements such as buried pipes, natural gas leaks, crude oil spills, and varying ambient light conditions. Attention to meticulous dataset labeling is essential, as it directly influences the model's performance.

The current training, conducted with 244 datasets on Google's dedicated GPU, achieved a commendable speed of 30 minutes. However, as datasets increase, investing in more sophisticated hardware with CUDA support is essential for efficiency improvement. Additionally, future researchers should dedicate efforts to comparing different pre-trained models to optimize defect detection algorithms.

ACKNOWLEDGEMENT

We extend our sincere gratitude to the Faculty of Mechanical Engineering, Universiti Teknologi Malaysia in supporting this research endeavor.

REFERENCES

[1] Burgherr, P. (2007). In-depth analysis of accidental oil spills from tankers in the context of global spill trends from all sources. Journal of hazardous materials, 140(1-2), 245-256.
[2] Chen, X., Tian, G., Wu, J., Tang, C., & Li, K. (2019). Feature-based registration for 3D eddy current pulsed thermography. ieee sensors journal, 19(16), 6998-7004.
[3] Ma, Q., Tian, G., Zeng, Y., Li, R., Song, H., Wang, Z., ... & Zeng, K. (2021). Pipeline in-line inspection method, instrumentation and data management. Sensors, 21(11), 3862.
[4] Oswald-Tranta, B. (2007). Thermo-inductive crack detection. Nondestructive Testing and Evaluation, 22(2-3), 137-153.
[5] Safizadeh, M. S., & Azizzadeh, T. (2012). Corrosion detection of internal pipeline using NDT optical inspection system. NDT & E International, 52, 144-148.
[6] Vrana, J., Goldammer, M., Baumann, J., Rothenfusser, M., & Arnold, W. (2008, February). Mechanisms and models for crack detection with induction thermography. In AIP conference Proceedings (Vol. 975, No. 1, pp. 475-482). American Institute of Physics.