UNDERWATER SINGLE AND MULTIPLE OBJECTS DETECTION BASED ON THE COMBINATION YOLOV7-tiny AND VISUAL FEATURE ENHANCEMENT

Dewi Mutiara Sari[1], Bayu Sandi Marta[2], R. Haryo Dwito Armono[3], Alfan Rizaldy Pratama[4], Firmansyah Putra Pratama[5]

[1, 2, 5)] Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya

[3)] Department of Ocean Engineering, Faculty of Marine Technology, Institut Teknologi Sepuluh Nopember Surabaya

[4)] Data Science Department, Universitas Pembangunan Nasional

email : dewi.mutiara@pens.ac.id[1], bayu@pens.ac.id[2], alfan.fasilkom@upnjatim.ac.id[3], armono@its.ac.id[4], firmansyah@ce.student.ac.id[5]

Abstract - Breakwater construction in Indonesia frequently employs tetrapods to dissipate wave energy. However, the placement process remains manual, relying on divers to guide crane operators. This approach not only poses safety risks but also limits visibility due to underwater turbidity. While prior research has focused on underwater image enhancement, the integration of tetrapod object detection remains unexplored. This study proposes a combined method of underwater image enhancement and tetrapod object detection to support land-based operator visualization. Auto-Level Filtering and Histogram Equalization techniques were applied to enhance image clarity, followed by object detection using the YOLOv7-tiny model. Tetrapod models at a 1:20 scale were used for training and testing. The proposed system achieved a mean average precision (mAP) of 0.95. Evaluation was conducted across 12 scenarios, involving four lighting levels and two water conditions: clear and 45.8% turbidity. The object detection confidence scores were 0.80 without enhancement, 0.85 with Histogram Equalization, and 0.84 with Auto-Level Filtering. Multiple object detection achieved an accuracy of 88.75%, outperforming previous approaches using YOLOv4-tiny. The results demonstrate the potential of integrating image enhancement and deep learning-based object detection for improving underwater operational safety and placement precision in breakwater construction.

Keywords – Breakwater construction, Tetrapod, Underwater Image, Object Detection, Turbidity, Computer Vision.

I. INTRODUCTION

In coastal engineering, tetrapods are commonly employed as armor units in breakwater structures to dissipate wave energy and prevent scouring at the base. Their distinctive geometric configuration facilitates interlocking, providing structural stability and long-term durability against hydrodynamic forces. Accurate placement of tetrapods is critical; misalignment or improper orientation can lead to displacement or structural failure under high wave impact [1]. In Indonesia, the conventional method for tetrapod placement involves land-based or pontoon-mounted hydraulic cranes, guided manually by divers who ensure that each unit is positioned according to the interlocking grid plan [2], [3]. However, this manual approach poses significant safety risks for divers and suffers from operational inefficiencies, particularly in high-current or low-visibility environments. The need for remote or semi-automated monitoring systems has led to interest in vision-based technologies to replace or augment diver-based guidance. Yet, object detection in underwater environments is inherently challenging due to complex optical distortions. Underwater imagery is subject to light attenuation, scattering, and absorption, resulting in poor contrast, colour distortion, and noise. These effects vary significantly with depth, water turbidity, and particle concentration, severely degrading visual quality [4].

Prior research has explored various underwater image enhancement techniques such as Multi-Scale Retinex with Colour Restoration (MSRCR), Contrast-Limited Adaptive Histogram Equalization (CLAHE), and basic Histogram Equalization (HE) [5]. While effective to some extent, these approaches often overlook dynamic environmental factors such as variable lighting conditions.

Some studies have integrated enhancement with object detection; for instance, YOLOv4-tiny combined with Histogram Equalization has achieved a confidence score of 0.76 in underwater scenarios [6]. Nonetheless, most existing research focuses on small, colourful marine organisms (e.g., scallops, starfish), and is based on datasets such as TrashCan or URPC, which may not generalise to large, low-contrast man-made structures like tetrapods [7]–[10]. Furthermore, the effectiveness of detection algorithms under varying turbidity and lighting levels remains underexplored. Large objects with non-distinctive textures and neutral colours, such as tetrapods, present unique challenges for detection systems. These gaps underscore the need for robust, real-time object detection frameworks tailored to underwater structural applications. Despite significant progress in underwater vision and deep learning-based detection, few studies have specifically addressed the detection of large-scale, low-texture, and low-contrast structures essential to coastal infrastructure. Prior works have largely neglected performance under realistic conditions of turbidity and lighting variability, which are common in field deployment scenarios. This study addresses these challenges by proposing an integrated vision-based framework that combines visual enhancement methods (Histogram Equalisation and Auto-Level Filtering) with the YOLOv7-tiny detection model. The proposed approach is evaluated under controlled variations in lighting and turbidity, using tetrapod objects modelled at a 1:20 scale. The main contribution of this work is the application and evaluation of a lightweight deep learning model optimised for real-time underwater object detection in the context of civil marine engineering and the demonstration of improved detection performance over YOLOv4-tiny in challenging visual environments.

II. SIGNIFICANCE OF THE STUDY

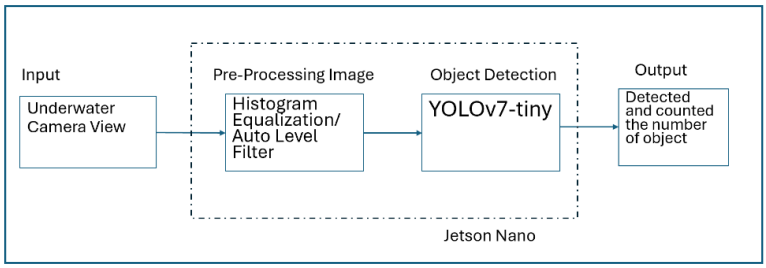


Fig. 1: Design System of Research

Based on the system design in Figure 1, the input of this system is the results of underwater camera captures, then processed at the image pre-processing stage using the Histogram Equalization/Auto Level Filter method. After that, it is processed using YOLOv7-tiny to detect and count the number of tetrapod objects.

A. Visual Feature Enhancement

Visual Feature Enhancement plays a crucial role in improving the interpretability and discriminative quality of image data in various computer vision and image analysis tasks. The primary objective of visual feature enhancement is to amplify relevant visual information while suppressing noise or irrelevant background details, thus facilitating more accurate detection, recognition, and classification processes. This research applied Histogram Equalization and Auto Level Filtering for this step.

1. Auto Level Filter

The Auto Levels Filter is a simplified automatic adjustment tool that remaps the tonal range of an image by setting new black and white points. It adjusts the darkest and lightest pixels in an image so that the darkest pixels are pushed closer to pure black (0), the lightest pixels are pushed closer to pure white (255), and the midtones are redistributed proportionally across the range. This remapping is performed independently for each of the three RGB colour channels (Red, Green, and

Blue). As a result, Auto Levels not only increases contrast but can also shift the colour balance slightly, depending on the tonal distribution of each channel. The equation of Auto Level Filtering is as follows

$$y = \frac{255}{i_{max} - i_{min}} (i - i_{min}) \quad (1)$$

With i is the original image's pixel value, i_{min} is the minimum image pixel value, i_{max} is the maximum image pixel value, and y is the final pixel value. The algorithm operates by continuously capturing frames and converting them from the RGB to the YUV colour space to isolate luminance information (Y channel). It calculates the minimum and maximum luminance values to determine a scaling factor, which is then used to normalize and adjust the brightness across the image. After modifying the luminance channel, it recombines it with the chrominance channels (U and V) and converts the image back to RGB. This enhanced frame is subsequently passed to a YOLO-based detection function for object recognition. The process iterates for each incoming frame, effectively improving contrast and detection reliability in low-visibility or underwater environments.

2. Histogram Equalization

Histogram Equalization is one of the most popular image processing methods in the spatial domain based on intensity transformation because it is very efficient, computationally effective, and simple to implement. Currently, modifications of the method are widely used to enhance images by increasing contrast [8]. The definition of equalization is often defined as follows,

$$s = T(r) = \int_0^r p_r(\omega) d\omega \quad (2)$$

With $p_r(\omega)$ is the PDF (Probability Density Function) of intensity values in the input image, $T(r)$ is the CDF (Cumulative Distribution Function), $s \epsilon [0,1]$ is the output normalized intensity? In discrete form, for an image with L are possible intensity levels $(0,1, \dots, L-1)$ the transformation is given by,

$$s_k = T(r_k) = \frac{L-1}{MN} \sum_{j=0}^{k} n_j \quad (3)$$

Where s_k is the output intensity, r_k is the input intensity, n_j is the number of pixels with intensity r_j, $Mx\ N$ is the total number of pixels in the image, and L is the number of intensity levels (typically 256 for 8-bit images). This transformation redistributes the pixel intensities so that the histogram becomes approximately flat, thus enhancing contrast in areas where intensities are concentrated within a narrow range. The algorithm of Histogram Equalization describes a visual preprocessing pipeline based on Histogram Equalization to enhance the quality of input frames before passing them to an object detection model, particularly YOLO. The enhancement focuses on improving the contrast of the image by redistributing the intensity values in the luminance channel. This is especially useful in conditions with poor visibility, such as underwater imaging or low-light environments.

B. Deep Learning

CNNs are designed to mimic the human visual cortex, leveraging the concept of local receptive fields and hierarchical feature extraction. Each layer in a CNN can be interpreted as a feature map, progressively extracting higher-level features from the raw input image. The input layer of a CNN typically accepts a three-dimensional matrix that represents different colour channels (e.g., RGB). As the signal propagates through the network, each layer performs specific transformations: Convolutional layers apply learnable filters to detect local patterns such as edges, textures, or shapes; activation functions (such as ReLU) introduce non-linearity, enabling the network to model complex relationships; and pooling layers (e.g., max pooling) reduce the spatial dimensions,

preserving essential features while minimizing computational cost and mitigating overfitting. Internally, the resulting representations take the form of multichannel feature maps, where each channel encodes a learned abstraction. Through deep stacking of these layers, CNNs are capable of learning highly abstract and semantically rich features. To perform specific tasks such as classification or regression, the convolutional and pooling layers are followed by one or more fully connected layers. These layers integrate the spatially extracted features and enable task-specific learning. Finally, the output layer typically incorporates an activation function such as SoftMax (for multi-class classification) or sigmoid (for binary classification), which converts the network's raw scores into probabilistic output values. The architectural flexibility and hierarchical structure of CNNs make them particularly effective across various domains, including medical imaging, autonomous driving, and facial recognition. Furthermore, CNNs form the backbone of many modern architectures such as ResNet, VGG, and Inception, each introducing innovations in depth, connectivity, and performance [11]. This paradigm shift marks a significant departure from the limitations of traditional machine learning approaches, positioning deep learning as the dominant methodology in modern AI research and deployment [12]. The primary goal of object detection is to determine both the location and the category of objects present in a given image.

1. *YOLOv7*

YOLOv7 introduces a compound model scaling mechanism tailored for concatenated network architectures. The objective of this scaling strategy is to dynamically adjust architectural attributes in order to produce model variants of different computational complexity and inference speed, aligning with diverse deployment scenarios. The proposed compound scaling technique is designed to preserve the essential architectural characteristics of the base model, thereby maintaining structural efficiency and detection performance.

2. *YOLOv7-tiny*

YOLOv7-Tiny is a compact and computationally efficient variant of the YOLOv7 object detection architecture, developed to meet the increasing demand for high-speed, real-time detection systems that can run on edge devices and resource-limited hardware. As an evolution of the well-established YOLOv3 and YOLOv4 models, YOLOv7-Tiny incorporates critical improvements in backbone design, feature aggregation, and optimization techniques while preserving the fundamental real-time detection principles of the YOLO family. YOLOv7-Tiny is particularly advantageous for deployment in domains requiring fast and efficient object detection, such as autonomous vehicles, real-time surveillance systems, drones, and mobile computing platforms. Despite its compact size and reduced computational complexity, it achieves robust performance across various benchmark datasets, demonstrating that architectural simplification does not necessarily entail a significant loss of accuracy [13]. Table 1 is the configuration of YOLOv7-tiny.

Table 1: YOLOv7-tiny Configuration

Layer	***In***	***Out***	***Kernel***	***Stride***	***Pad***	***Activation***
Conv	-	32	(3, 3)	(2, 2)	(1, 1)	*LeakyReLU*
Conv	32	64	(3, 3)	(2, 2)	(1, 1)	*LeakyReLU*
Conv	64	32	(1, 1)	(1, 1)	(0, 0)	*LeakyReLU*
Conv	64	32	(1, 1)	(1, 1)	(0, 0)	*LeakyReLU*
Conv	32	32	(3, 3)	(1, 1)	(1, 1)	*LeakyReLU*
Conv	32	32	(3, 3)	(1, 1)	(1, 1)	*LeakyReLU*
Concat	-	-	-	-	-	-
Conv	-	64	(1, 1)	(1, 1)	(0, 0)	*LeakyReLU*
MP	64	64	(2, 2)	(2, 2)	(0, 0)	-
Conv	64	64	(1, 1)	(1, 1)	(0, 0)	*LeakyReLU*
Conv	64	64	(1, 1)	(1, 1)	(0, 0)	*LeakyReLU*
Conv	64	64	(3, 3)	(1, 1)	(1, 1)	*LeakyReLU*
Conv	64	64	(3, 3)	(1, 1)	(1, 1)	*LeakyReLU*
Concat	-	-	-	-	-	-

Layer	In	Out	Kernel	Stride	Pad	Activation
Conv	-	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
MP	128	128	(2, 2)	(2, 2)	(0, 0)	-
Conv	128	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	128	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	128	128	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	128	128	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
MP	256	256	(2, 2)	(2, 2)	(0, 0)	-
Conv	256	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	256	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	256	256	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	256	256	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	512	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	512	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	512	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
SP	-	-	-	-	-	-
SP	-	-	-	-	-	-
SP	-	-	-	-	-	-
Concat	-	-	-	-	-	-
Conv	-	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	256	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Upsample	-	-	-	-	-	-
Conv	512	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	64	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	64	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	256	64	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	64	64	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	64	(3, 3)	(2, 2)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	64	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	128	64	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	64	64	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	32	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	128	32	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	32	32	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	64	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	128	(3, 3)	(2, 2)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	64	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	64	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	128	64	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	64	64	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	128	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	256	128	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	128	128	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Concat	-	-	-	-	-	-
Conv	-	256	(1, 1)	(1, 1)	(0, 0)	LeakyReLU
Conv	-	128	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	256	256	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Conv	256	512	(3, 3)	(1, 1)	(1, 1)	LeakyReLU
Detect	-	-	-	-	-	-

Based on Table 1, there are several parts that affect its performance: Using concatenation and convolution intensively to merge features, Using the Leaky ReLU activation function, Increasing accuracy by using concatenation and up-sampling to support multi-scale detection, and not using Spatial Pyramid Pooling with Fusions but overcoming with convolution and concatenation. With these parts, YOLOv7-tiny is designed to be more complex than YOLOv3-tiny and YOLOv4-tiny. YOLOv7-tiny provides excellent accuracy performance thanks to extensive feature merging and the use of multi-scale detection. This study contributes both practical and academic significance. Practically, the proposed system has the potential to transform tetrapod placement operations by enabling remote object detection, thus reducing dependence on divers and improving worker safety in high-risk underwater environments. The system's capability to maintain high detection accuracy under varying turbidity and lighting levels aligns with real-world deployment challenges commonly encountered in Indonesia's coastal infrastructure projects. This can support decision-making for crane operators, reduce human error, and shorten placement time in the field. Scientifically, this research introduces a novel application of YOLOv7-tiny combined with visual enhancement techniques tailored for large, low-contrast man-made underwater objects. Unlike prior works, which focus on small biological targets or simulated environments, this study provides a new benchmark for detecting industrial-scale objects under real aquatic conditions. Furthermore, the curated dataset and controlled evaluation scenarios (involving water turbidity, lighting, and multi-object setups) offer a valuable reference for future studies in underwater machine vision and civil marine engineering. By addressing real-world constraints while advancing lightweight deep learning frameworks for submerged detection tasks, the findings serve both coastal engineering practitioners and the broader research community seeking robust vision-based solutions in marine contexts.

III. RESULT AND DISCUSSION

This part discusses the experimental setup and results with the analysis.

A. Experimental Setup

In this session, the author explains the testing stages on the system built to determine the level of performance and success of the system implementation. The process of implementation, compilation, and system creation in this final project was carried out with the specifications in Table.

Table 2: Device Specification

Testing Environment	Part	Specification
Laptop Hardware	*Processor*	Intel Core i3
	Storage	*SSD* 250*GB*
	RAM	*8GB*
	GPU	*Intel UHD*
Jetson Nano Hardware	*Processor*	Quad-core ARM A57 @ 1.43 GHz
	Storage	*128GB memory card*
	RAM	*4GB*
	GPU	128-core Maxwell
Software	Sistem Operasi	NVIDIA JetPack 4.6.4 (Ubuntu 18.04 LTS)
	Software Build	Visual Studio Code dan Text Editor
	Library	*OpenCV 4.8 with CUDA*

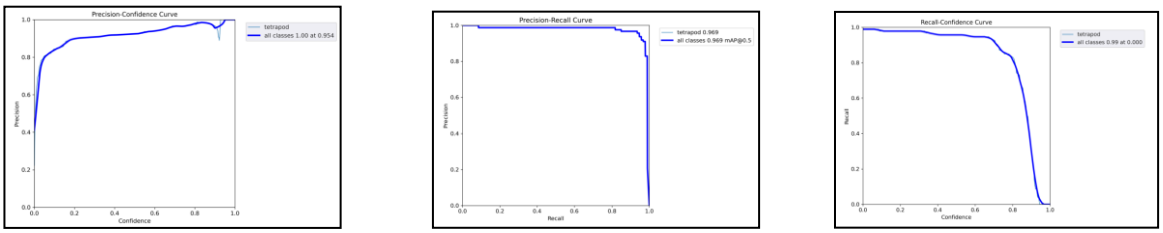
The camera distance with respect to the object is ±15.8 cm. The dataset utilized in this study was obtained through controlled laboratory experiments conducted in a custom-built underwater test tank. Tetrapod models at a 1:20 scale were submerged under varying lighting intensities and turbidity levels to emulate realistic underwater conditions. Image sequences were captured using fixed underwater cameras positioned at calculated distances to encompass both single and multiple object configurations. While the dataset closely approximates real-world underwater scenarios, it is important to note that all data were collected in a simulated environment rather than in open-sea conditions.

B. Testing Result

This section discusses the result of training for the object detection and testing of object detection, which divided into two parts, single and multiple objects.

1. Training of Object Detection

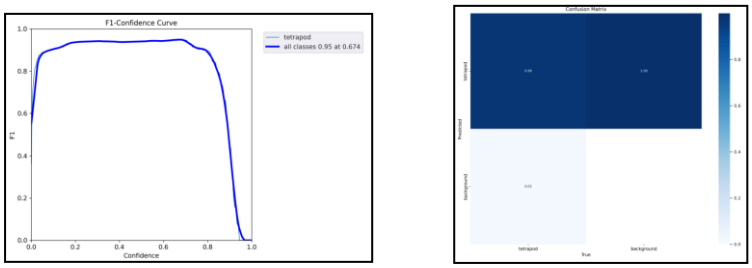
The training dataset consisted of labelled frames extracted from video sequences that featured floating and submerged objects in various positions and backgrounds. The objects were annotated with bounding boxes using the YOLO format, each accompanied by class labels. The dataset was augmented using techniques such as random cropping, horizontal flipping, colour jittering, and Gaussian blur to improve generalization. The YOLOv7-tiny model was initialised with pretrained weights on the COCO dataset and fine-tuned using a custom dataset of 2597 images, image size 640, batch size 16, and learning rate 0,01 with 50 epochs. The result of the training process is discussed as follows,



(a) (b) (c)

Fig. 2: (a) Precision Confidence (b) Precision Recall Curve (c) Recall Confidence Curve

Based on Figure 2, the curve shows the change in precision (accuracy of correct detection) against the confidence threshold. The results show stability in the range of 0.8 to 1. This indicates that the model is confident that the object recognized is indeed a tetrapod object. The Precision-Recall Curve shows the relationship between precision and recall. The curve results show a high precision value throughout the recall range. This can be interpreted as meaning that the model has very high accuracy in detecting tetrapods with an IoU threshold of 0.5. It can also be interpreted that the model can detect almost all tetrapod objects that appear. The curve above is the Re-call-Confidence Curve, which shows the relationship between recall (the model's ability to find all relevant objects) and confidence. The curve results show a high recall value when confidence is low and decrease slightly when confidence approaches 1.0. This can be interpreted, if confidence is low, then the model recognizes almost all objects but is at risk of error, conversely, with high confidence, the model is more selective in detecting objects but is at risk of missing some objects.



(a) (b)

Fig. 3: (a) F1-Confidence Curve (b) Confusion Matrix

The curve in Figure 3 (a) is the F1-Confidence Curve which shows the relationship between F1 (a combination of precision and recall) to confidence. The F1 score is at a value close to 1 in the confidence range of 0 to 0.8 and then decreases significantly approaching confidence 1. This shows that the model's ability to detect can be said to be quite robust to variations in confidence threshold, but becomes too selective if confidence is too high. Figure 6 (b) is a Confusion Matrix that shows the proportion of correct and incorrect predictions between two classes, tetrapod and background (no object). The upper left corner shows that the tetrapod was successfully detected 0.98 or 98% correctly. The lower left corner shows that there were 0.02 or 2% that were not detected. The upper right corner shows that there was no background that was predicted incorrectly or can be interpreted as no background that was detected as a tetrapod.

2. Testing of Object Detection

The system detection test was conducted in two water conditions and two tetrapod object count conditions under different lighting conditions. In more detail, the test scenario is written as in Table 3.

Table 3: Testing Scenario

Water Turdibity	**Lighting**	**Number of Tetrapod**
Clear Water	25%	1
	50%	
	75%	
	100%	
Turdibity 45,8%	25%	
	50%	
	75%	
	100%	
Clear Water	25%	5
	50%	
	75%	
	100%	
Turdibity 45,8%	25%	
	50%	
	75%	
	100%	

To measure water turbidity, the Secchi disk is immersed in the water at two specific points. The first point (D_1) is when the Secchi disk is no longer visible to the eye, and the second point (D_2) is when the Secchi disk becomes visible to the eye again. The value obtained for D_1 is 48 cm and D_2 is 26 cm from the water surface. The values D_1 and D_2 are entered into the following Equation 7.

$$s = T(r) = \int_0^r p_r(\omega) d\omega \quad (7)$$

Then the measurement results are obtained as follows.

$$P = \frac{D_1 - D_2}{D_1} \times 100\% = \frac{48 - 26}{48} \times 100\% \quad (8)$$

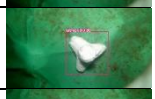
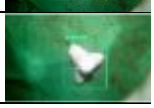
$$P = \frac{22}{48} \times 100\% = 45.8\% \quad (9)$$

Thus, the water used in the test has a turbidity level (P) is 45.8%.

1) Single Object

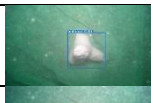
The test continued by testing the model's performance in detecting objects in the form of a tetrapod. These visual results have applied image preprocessing using Histogram Equalization, Auto Levels Filter, and without any method of image preprocessing method conducted in the clear water condition. So in this test, the total test scenarios is 12. In each scenario, a 10 second test video was taken. while the total frames of each scenario are 134. The result is summarized in Table 4.

Table 4: Single Object Detection in Clear Water

Pre-Processing Image	**Lighting Level**	**Confidence Value Average**	***Confidence Value***
No Method	25%	0,84	
	50%	0,86	
	75%	0,86	
	100%	0,88	
Histogram Equalization	25%	0,8	
	50%	0,87	
	75%	0,86	
	100%	0,88	
Auto Level Filter	25%	0,78	
	50%	0,84	
	75%	0,87	
	100%	0,88	

Based on the test results in 12 test scenarios, it was found that in clear water conditions, all models showed performance with confidence of more than 0.78 using YOLOv7-tiny. The average confidence value of YOLOv7-tiny performance with the combination of Histogram Equalization method is 0,85. Meanwhile the average confidence value of YOLOv7-tiny performance with the combination of Auto Level Filter is 0,84. And the average confidence value of YOLOv7-tiny performance only is 0,86. The test is continued by testing the model's performance in detecting tetrapod object in the turbidity water 45,8% with 4 various lighting levels. The result is summarized in Table 5.

Table 5: Single Object Detection in 45,5% Turbidity Water

Pre-Processing Image	**Lighting Level**	***Confidence Value***	***Confidence Value***
No Method	25%	0,74	
	50%	0,73	
	75%	0,76	
	100%	0,78	
	25%	0,83	

Histogram Equalization	50%	0,83	
	75%	0,86	
	100%	0,87	
	25%	0,84	

Auto Level Filtering	50%	0,85	
	75%	0,84	
	100%	0,78	

The combination of external lighting levels and the application of image pre-processing influence detection performance. Lower lighting causes a decrease in confidence, especially in turbid water conditions. However, YOLOv7-tiny can maintain the ability to maintain confidence so that the confidence value is almost the same as when the water condition is clear, which is above 073. The average confidence value of YOLOv7-tiny performance with the combination of Histogram Equalization method is 0,84. Meanwhile, the average confidence value of YOLOv7-tiny performance with the combination of Auto Level Filter is 0,82. And the average confidence value of YOLOv7-tiny performance only is 0,75.

In both clear and turbid water, detection confidence improves with lighting intensity, confirming the model's sensitivity to illumination. However, the use of Histogram Equalization consistently boosts detection across all lighting levels, especially under 45.8% turbidity. This confirms its effectiveness in restoring contrast under poor visibility. In contrast, Auto Level Filtering showed fluctuating performance, particularly at 25% and 100% lighting, possibly due to local channel overcompensation or colour imbalance. The inconsistent results from Auto Level Filtering may stem from its per-channel adjustment strategy, which can unintentionally distort colour channels and reduce edge sharpness, two critical cues in object localization. This is especially problematic under extreme lighting or turbidity, where noise amplification may occur.

2) Multiple Objects

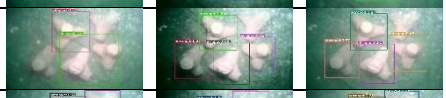
In this session is to test the accuracy of the system in detecting 5 tetrapod objects using YOLOv7-tiny. The testing was carried out in 2 water conditions (namely clear water and water turbidity of 45.8%) and 4 lighting levels (25%, 50%, 75%, and 100%). In each test scenario, a 10-second video was taken. The total frames for the clear water scenario were 134 frames (for each lighting level) and 126 frames for the 45.8% water turbidity scenario (at each lighting level). The result is represented in Table 6.

Table 6: YOLOv7-tiny Multiple Object Test Results

Water Turbidity	**Lighting**	**Number of Tetrapod (Refference)**	**Number of Detected Tetrapod with Pre-Processing Image Method**		
			No Method	**Histogram Equalization**	**Auto Level Filtering**
Clear Water	25%	5	4	5	4
	50%		4	5	5
	75%		4	5	4
	100%		5	5	4
45,8% Turbidity Water	25%		2	4	4
	50%		3	5	4
	75%		3	5	4
	100%		4	4	4

Based on Table 6, the performance of the YOLOv7-tiny method in detecting multiple objects also depends on the application of the pre-processing image method. Without the image pre-processing method, the worst result is when in the water turbidity test scenario of 45.8% with an external lighting level of 25%, only 2 objects are detected when there should be 5. When using the Histogram Equalization and Auto Level Filtering Methods as image pre-processing, the worst result is that it can detect 4 objects from the supposed 5 objects in several scenarios (especially when the water turbidity is 45.8%). Meanwhile for the image result for this test is presented in Table 7.

Table 7: Visual Results of YOLOv7-tiny in Multiple Objects Testing

Lighting	Pre Processing Image Method			Lighting	Pre Processing Image Method		
	No Method	Histogram Equalization	Auto Level Filter		No Method	Histogram Equalization	Auto Level Filter
	Clear Water				45,8% Turbidity Water		
25%				25%			
50%				50%			
75%				75%			
100%				100%			

Based the 8 test scenarios, without using the image pre-processing method, only 1 time successfully detected all objects. When using Histogram Equalization, 6 times the system was able to detect all objects in their entirety. When the combination of the YOLOv7-tiny and Auto Level Filtering methods was applied to 8 test scenarios, 1 time the system was able to detect all objects (in the clear water scenario with the help of external lighting level 50%), and the rest detected 4 objects (from what should have been 5).

IV. CONCLUSION

This study utilized YOLOv7-tiny combined with visual feature enhancement methods, Histogram Equalization and Auto Level Filtering for underwater tetrapod detection. The model was trained on a custom dataset of 2,597 images and achieved a mean average precision (mAP) of 0.95. In testing across 12 scenarios, each involving 10-second video sequences under various lighting and turbidity levels, the system produced an average object detection confidence of 0.83, outperforming a previous YOLOv4-tiny implementation (0.76). For multiple-object detection, the system achieved an accuracy of 88.75%, reflecting a 2.5% improvement over prior work. These results demonstrate the system's potential for practical deployment in coastal construction by reducing diver dependency, improving safety, and enhancing placement precision. Nevertheless, the experiments were conducted in a controlled laboratory environment using scaled models, which may not fully reflect real underwater conditions. Future research should evaluate the system in open-sea scenarios, include non-tetrapod distractor objects, and explore advanced detection architectures to further improve robustness and generalizability.

REFFERENCE

[1] J.-P. Latham et al., “Stability and hydraulic performance of interlocking concrete armour units: A review,” Coastal Engineering, vol. 79, pp. 1–14, 2013.
[2] T. Uda, “Application of UAV and GIS Technologies for Coastal Structure Construction Monitoring,” Journal of Coastal Research, vol. 95, pp. 812–816, 2020.
[3] Kementerian PUPR, "Panduan Teknis Pemasangan Struktur Pemecah Gelombang," 2021.
[4] C. Li, J. Guo, R. Cong, Y. Pang, and B. Wang, “An underwater image enhancement benchmark dataset and beyond,” IEEE Transactions on Image Processing, vol. 29, pp. 4376–4389, 2020.
[5] B. S. Marta, M. A. Amin, D. M. Sari, and H. D. Armono, “The Comparison of Image Enhancement Methods for A Realtime Under Water Vision System of The Break Water Implementation,” 2022 International Electronics Symposium (IES), Surabaya, Indonesia, pp. 361–367, 2022, doi: 10.1109/IES55876.2022.9888686.
[6] F. P. Pratama, A. R. Pratama, D. M. Sari, B. S. Marta, and R. H. Dwito Armono, “Performance Evaluation of YOLO-Based Deep Learning Models for Real-Time Armour Unit Detection with Image Pre-processing Method,” 2024 International Electronics Symposium (IES), Denpasar, Indonesia, pp. 541–546, 2024, doi: 10.1109/IES63037.2024.10665816.
[7] M. J. Islam, Y. Xia, and J. Sattar, “Fast underwater image enhancement for improved visual perception,” IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 3227–3234, 2020.
[8] P. Pachiyappan et al., “Enhancing Underwater Object Detection and Classification Using Advanced Imaging Techniques: A Novel Approach with Diffusion Models,” MDPI, Sustainability, vol. 16, no. 17, p. 7488, 2024.
[9] H. Zhou et al., “Real-time underwater object detection technology for complex underwater environments based on deep learning,” Ecological Informatics, vol. 82, 2024.
[10] A. Mathia et al., “Underwater object detection based on bi-dimensional empirical mode decomposition and Gaussian Mixture Model approach,” Ecological Informatics, vol. 66, 2021.
[11] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 11, pp. 3212–3232, 2019.
[12] L. Alzubaidi et al., “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” Journal of Big Data, vol. 8, p. 53, 2021.
[13] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” arXiv preprint arXiv:2207.02696, 2022.
[14] S. Yelmanov, O. Hranovska, and Y. Romanyshyn, “A new approach to the implementation of histogram equalization in image processing,” IEEE, pp. 288–293, 2019.
[15] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. AlShamma, J. Santamar´ıa, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” Journal of Big Data, vol. 8, p. 53, 3, 2021.
[16] Wang, C.Y., Bochkovskiy, A., & Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696. https://arxiv.org/abs/2207.02696, 2022.

ACKNOWLEDGEMENT

This research is provided by Penelitian Lokal (PENLOK) Politeknik Elektronika Negeri Surabaya (PENS).