Lightweight WaSR-T Network for Detection Boat Approaching a Tsunami Early Warning System

Wayan Wira Yogantara[1,3*], Suprijanto[2], A. A. N. Ananda Kusuma[4], Yuki Istianto[5]

[1]Master's Program in Instrumentation and Control, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia
[2]Instrumentation, Control and Automation Research Group, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia
[3] Research Center for Electronics, National Research and Innovation Agency, Jakarta, Indonesia
[4] Research Center for Telecommunications, National Research and Innovation Agency, Jakarta, Indonesia
[5]Research Center for for Artificial Intelligence and Cyber Security, National Research and Innovation Agency, Jakarta, Indonesia

Abstract. A tsunami early warning system using buoys is vital for early warning of tsunami waves. Its vulnerability to tampering and even vandalism emphasizes the need for an object detection vision system for tsunami buoys. Moreover, researchers typically position these buoys at considerable distances from the seashore. The current tsunami early warning system lacks an object detection system capable of providing warnings about the presence of other disturbing objects. Hence, any system vision must incorporate object detection with energy efficiency. This research studies various efficient object detection network models that support object detection systems for these tsunami buoys. WaSR-T model network with temporal context was developed and equipped with a lightweight encoder MobileNetV3 to run on a single board computer. Although the test results show less than optimal performance than the original network model, the experiments highlight that the lightweight WaSR-T remains the most promising for object monitoring on tsunami buoys, given its low memory requirements. Researchers can also implement it in other mid-ocean monitoring applications, such as rigs and marine platforms.

Keywords: *tsunami early warning; bouy; system vision; object detection;WaSR-T; MobileNetV3.*

1 Introduction

The Tsunami Early Warning System (TEWS) [1] relies on a buoy on the sea surface connected to a water pressure sensor on the seabed via acoustic communication to detect the potential for tsunami waves. However, a significant obstacle to this system is that ship propellers approaching the buoy create noise interference, which can exceed the noise level tolerated by data transmission with acoustic waves. The absence of an intelligent

Received _____, Revised _____, Accepted for publication _____

computer vision device on the tsunami buoy that can provide information about approaching ships and warn the ship to stay away is an obstacle for the tsunami buoy. This research developed intelligent computer vision to detect the presence of ships that could interfere with the performance of acoustic communication. This research focuses on developing an object recognition network model that does not require significant computational processes or complex algorithms that require much computational effort while maintaining a balance between accuracy. The limited space on the buoy and a battery pack restricts the recognition and computing process. The system must minimize power consumption while computing to enable object recognition on embedded devices or computer.

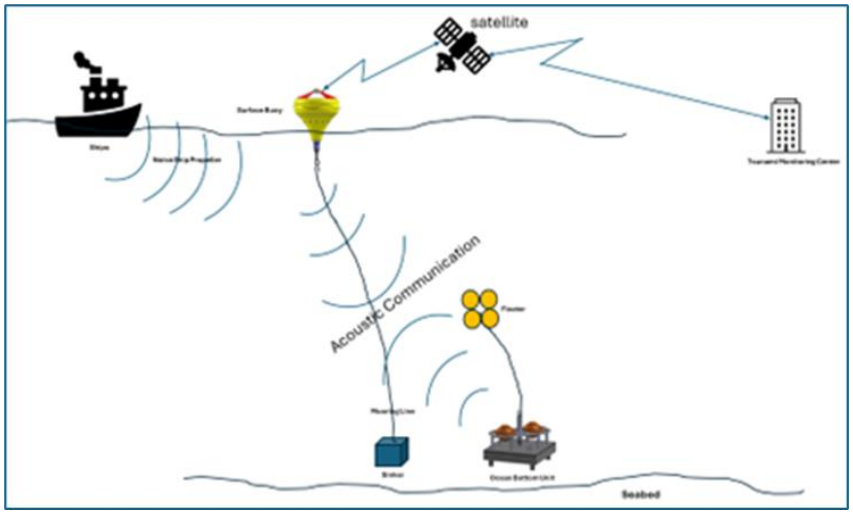

Figure 1 Acoustic communication failure due to ship propeller noise

Current intelligent computer vision methods detect objects by classifying each pixel in the image using convolutional neural networks, providing valuable information for understanding the environmental conditions and detecting objects. Sophisticated convolutional neural networks for intelligent computer vision object detection have become a well-established approach in ground-based autonomous vehicles[2],[3],[4],[5]. However, researchers must improve existing networks developed on land for use in maritime areas.

So, various object recognition network models were developed for unmanned vessels, such as marine surveys in limited areas[6]. The most reliable network model currently is the Temporal Water-obstacle

Separation and Refinement Network or WaSR-T network with an endocer and decoder architecture consisting of several information fusion and feature scaling blocks that extract temporal context from a series of image frames to distinguish objects from sea surface reflections[7].

Despite the excellent performance of unmanned vessels in recognizing surface objects for navigation purposes, WaSR-T networks still require high computing devices[8]. For tsunami buoys, a lighter network is needed with a balance between accuracy and minimal computational processing to detect ships approaching the buoy as an integral part of an intelligent computer vision system in the open sea.

2 Method

Developing a network model for an intelligent vision system consists of two stages: a dataset creation process and a network model development process.

A. Dataset Preparation

Detecting surface objects on the high seas with convolutional neural networks has various challenges, such as object complexity, marine environmental conditions that are very different from conditions on land, availability of image data sets with annotations [9], class imbalance to be detected, anomaly detection, detection constraints for objects of small size, and model reliability for all sea conditions. In addition, special knowledge and expertise in the field of maritime imagery can further improve the effectiveness of convolutional neural network-based object detection in the open sea.

All images used as training data were taken simultaneously at the tsunami buoy installation location as part of the maintenance process. This image was taken from the research ship Baruna Jaya. All images recorded depict all sea conditions from morning to dusk, both in clear and lousy weather conditions, so that, as far as possible, all sea conditions can be used as training data.

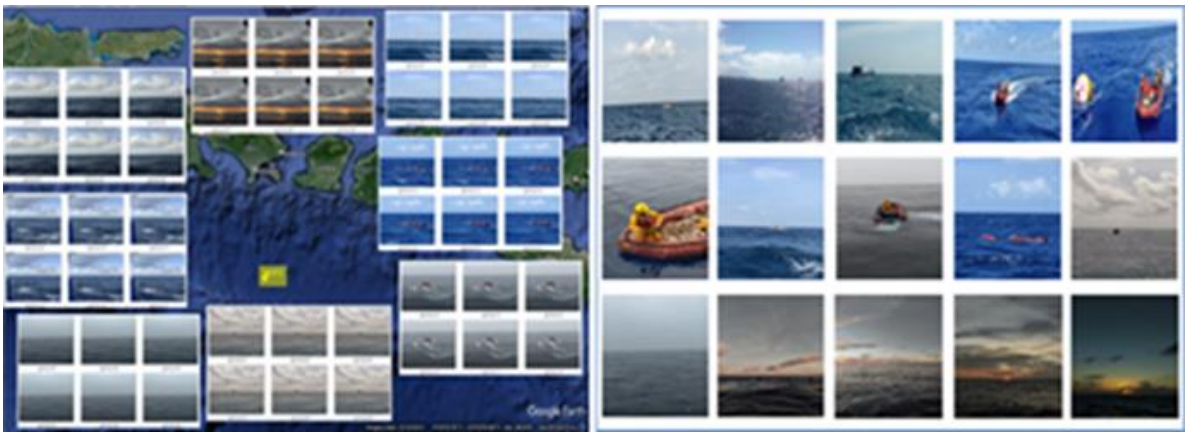

Figure 2 Image recording process at the Tsunami Buoy location

After all the images have been collected, a data-cleaning process is carried out. Here, data selection is carried out where inappropriate or duplicate data is discarded. Next, resizing is done to adjust the image dimensions according to the data format. The image annotation process is carried out after cleaning using the Labelme application. Each image is labeled with clouds, water, ships, and the associated pixels.

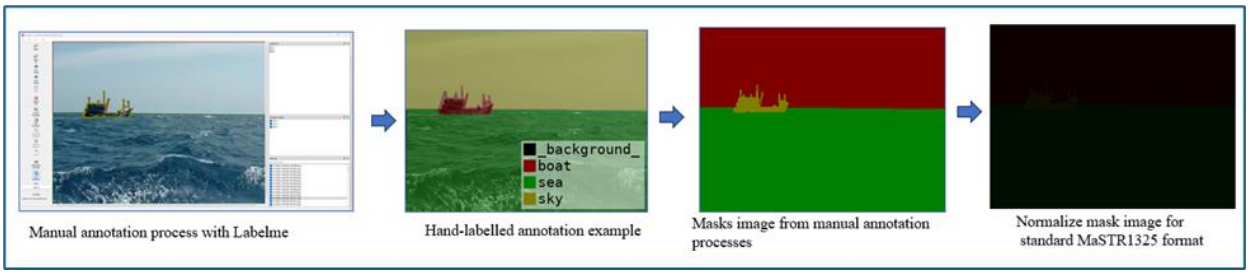

Figure 3 Manual image annotation process

This dataset is divided into 80% training data, 10% validation data and 10% test data.

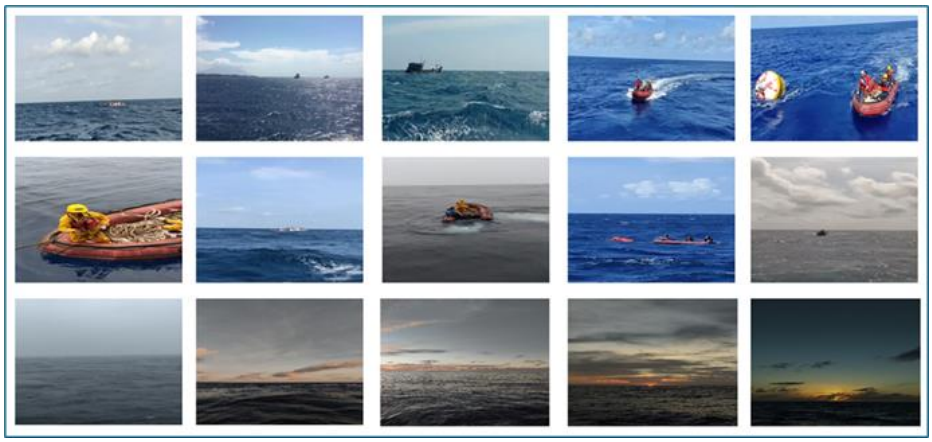

Figure 4 Sample Training Dataset.

B. Lightweight WaSR-T Network Development

The most up-to-date object detection network model on the water surface is the Temporal context Water-obstacle Separation and Refinement Network or WaSR-T[7], where this network model is used in water surface autonomous vessel to recognize all objects in front of them so that this vesel can avoid these objects so as not to be hit, this is the feature we will use in tsunamis buoy as a network model to recognize objects with a slight difference that the tsunami buoy will not move away because the position of the tsunami buoy is tied to the mooring line but will issue an alarm if the object approaches. By analyzing the WaSR-T network model, we will make modifications to get a lighter network model.

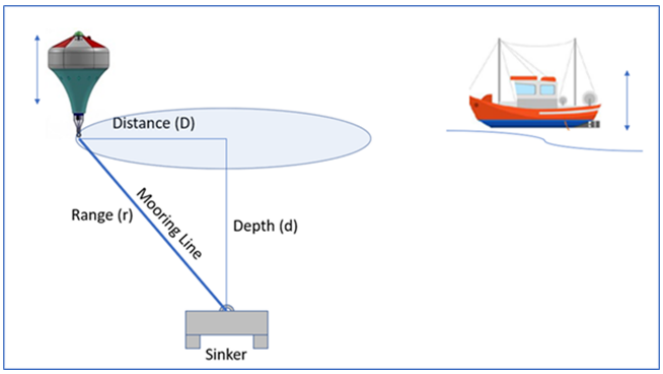

Figure 5 Tsunami Buoy with mooring line and anker

To get a lightweight network model that can run on a single-board computer, the encoder blocks in the WaSR-T network model, which previously used ResNet-101[10], were replaced with MobileNetV3[11].

ResNet-101 requires a high computational process to extract features in the image. MobileNetV3 has the same function but only requires a low computing process. MobileNetV3 was developed to run on computer systems with low memory resources, such as cellular devices.

Unlike standard convolution in ResNet, depth separable convolution breaks the computation into two distinct steps:

- Deep convolution applies a single convolutional filter to each input channel, and
- Unidirectional convolution combines the outputs of the deep convolution into a linear form

Deep convolutional kernels are learnable parameters applied to each input channel separately to improve model efficiency and reduce computational processes in MobileNets[12] networks. The system also shares this across all input channels. Next, researchers use neural architecture search (NAS) to find the optimal kernel size for deep convolution and to determine the best architecture that suits low-resource hardware platforms in terms of size, performance, and latency. Additionally, MobileNetV3 uses squeeze and excitation (SE)[13] blocks, which pay more attention to relevant features in each channel during training and improve feature representation with low memory usage. MobileNet is structured using several units known as bottleneck blocks (bneck).

The overall architecture of MobileNet is depicted in Figure 6. (a), while Figure (b) provides a closer look at the internal configuration of the neck block. MobileNetV1[12] innovatively replaces standard convolutional operations with depth-separable convolutions in each block, significantly reducing the number of parameters. Additionally, a residual connection is introduced between the input and output tensors, as shown in the Figure below. MobileNetV2[14], the design is further improved by incorporating expansion and compression steps at the beginning and end of each neck block, forming what is known as an Inverted Residual Block (IRB). This configuration is inverted because it connects narrow input and output tensors (i.e., low channel count) through residual connections, unlike the

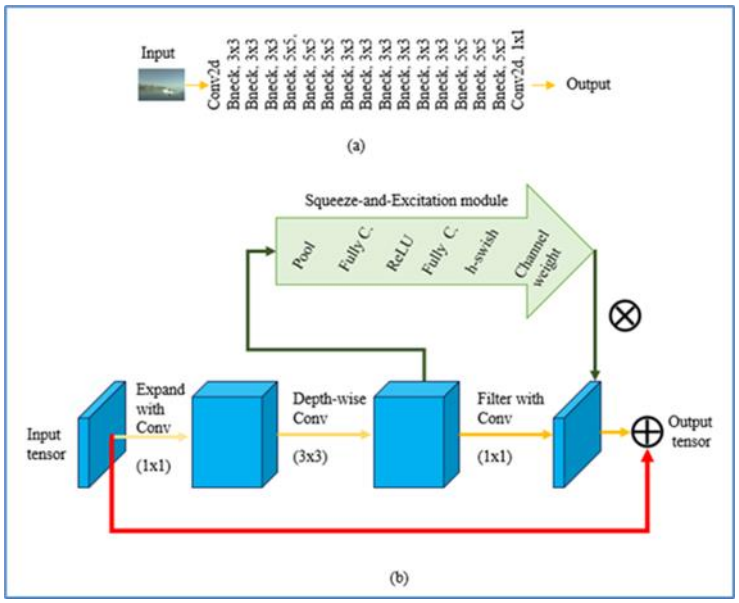

Figure 6 The MobileNet model consists of a set of merged bneck blocks. (a) High-level overview. (b) Illustration of bneck block

original ResNet CNN model, which connects extended tensors through residuals.

Introducing an Inverted Residual Block (IRB)[14] in MobileNetV2 is a game-changer, significantly reducing the computational cost of the model. The design uses linear activation after filtering the input and output tensors instead of non-linear activation functions like ReLU to reduce computation further. Additionally, MobileNetV3 includes a Squeeze-Exitation (SE)[11] module, as shown in Figure 4.b. Unlike other models that integrate SE modules as separate blocks in the architecture, such as ResNet [15], MobileNetV3 integrates them in parallel with IRB connections. This integration slightly increases model size but the reassurance of reduced computational cost is a testament to the efficiency of the model.

The MobileNetV3 SE module implements a new activation function called h-swish[13]. The h-swish function is a variant of the Swish activation function, defined as follows:

$$swish(x) = x.\sigma(\beta x) = \frac{x}{1+e^{-\beta x}} \qquad (1)$$

Where σ(βx) is the sigmoid function, and β is the trainable parameter. If β = 1, it is known as a sigmoid weighted linear unit function. However, computing this function is computationally expensive. The h-swish function modifies the Swish activation to improve computational efficiency and is defined as follows:

$$h - swish(x) = x\frac{ReLU6(x+3)}{6} \quad (2)$$

ReLU activation function, limiting its output to a maximum value of 6, and the bottleneck block producing an impressive improvement in feature maps with residual connections and the Squeeze-Exitation module (SE), we were inspired to adopt this as the primary backbone model for the WaSR-T architecture.

C. Experimental Scenario

In Figure 7, we show the architecture of the proposed Lightweight WaSR-T network model with all the stages, including its various blocks and input feature maps. The selected MobileNetV3 layer applies downsampling in the encoder section to reduce the image size. Upsampling and transposition convolutions are applied in the decoder section to generate segmentation masks for each input image.

We chose this stage because it is an activation layer (ReLU) with the highest convolutional filter bands in its feature map size category (256 × 192, 96 × 128, 48 × 64, et cetera). For example, Stage 2 resizes the image to 96×128 with 24 bands. Stage 3 resizes the image to 48×64 with 40 bands. Stage 4 resizes the image to 16×12 with 160 bands. Finally, Stage 5 resizes the image to 16 × 12 with 960. After that, the Temporal Context Module (TCM) extracts temporal information from the context and target frame embeddings. It then combines this information with the target frame embedding via concatenation. TCM reduces the dimensions of the feature map per frame by a specific process, which is designed to preserve the structure and quantity of input channels to the decoder. This procedure is processed using shared 1×1 convolutional layers to project feature maps per frame into N/2 dimensions per frame.

After that, the decoder upsets and merges with the previous output layer and each MobileNetV3 Stage. The output form from Stage 2 goes to the Feature Fusion Module (FFM) in the decoder section to combine low-level and high-level features. The output form follows Stage 3, as it directly

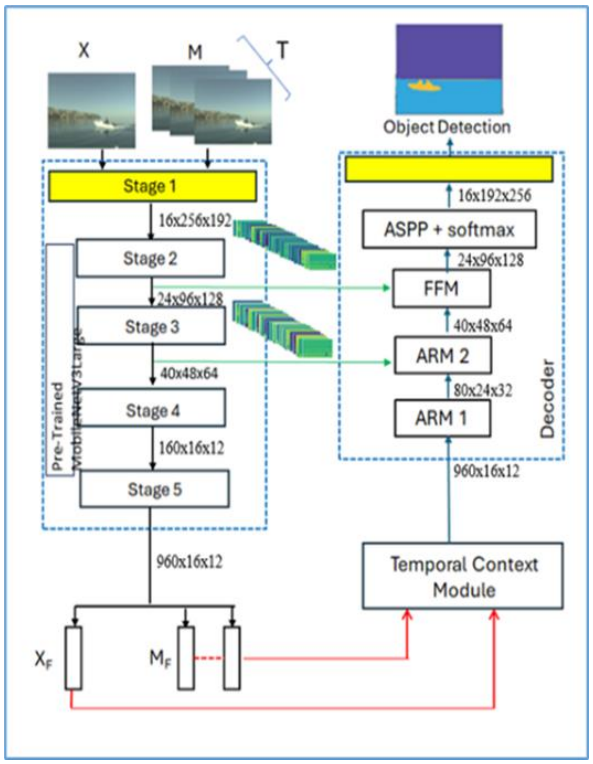

Figure 7 Lightweight WaSR-T

features from the bottom of the encoder to Attention Refinement Modules 2 (ARM2), the decoder. After Stage 5, the process follows the original. Input and output images are the same size. The output of the proposed model is a 512 × 384 image mask segmentation with three bands, showcasing our model's impressive ability to detect three separate classes. The table below provides a detailed list of layers and the number of parameters in each decoding layer of our proposed model.

Fine-tuning process and feature extraction phase are described in this section. The main goal is to extract relevant image embeddings by relying on models trained on different marine image datasets. Meanwhile, the image embeddings extracted in this phase are fed into the feature selection phase, which will be discussed in the next section. Compared with previous studies, the feature selection phase uses adam optimization technique to improve recognition accuracy, select only important features, and reduce the feature representation space of the entire proposed framework.

Table 1 Lightweight WaSR-T layer

Part	Operation (Block)	Input Shape	Output Shape
Encoder (Base Model: MobileNetV3 Large)	Input Layer (Rescalling, Conv2D)	512 x 384 x 3	256 x 192 x 16
	Downsampling#1 (bneck, 3x3)	256 x 192 x 16	128 x 96x 24
	Downsampling#2 (bneck, 3x3)	128 x 96 x 24	64 x 48 x 40
	Downsampling#3 (bneck 3 x 3)	64 x 48 x 40	32 x 24 x 80
	Downsampling#4 (bneck 5 x 5)	32 x 24 x 80	16 x 12 x 160
	Downsampling#5 (bneck 5 x 5)	16 x 12 x 160	16 x 12 x 960
Decoder	TemporalContextModule TCM	16 x 12 x 960	16 x 12 x 960
	AttentionRefinementModule ARM1	16 x 12 x 960	32 x 24 x 80
	AttentionRefinementModule ARM2	32 x 24 x 80	32 x 24 x 80
	FeatureFusionModule FFM (upsampl	64 x 48 x 40	64 x 48 x 40
	ASP	128 x 96x 24	128 x 96x 24

We trained the proposed Lightweight WaSR-T network using 290 image datasets, which include previous sequential frames with T=5 and corresponding image annotations and divide the training dataset into mini-batches of 6 images each to enhance the efficiency of the training process. The input image size for this training is 512 x 384 x 3. The Adam (Adaptive Moment Estimation) optimizer, known for its adaptability, uses square gradients to adjust the learning rate. It tracks the moving average of gradients (an approach called momentum) and can also assess moments adaptively. This training run NVIDIA DGX1 and implemented using python 2.0.0 and torch-vision 0.15 library with following parameters.

Table 2 Training parameters

Parameter	Value
Learning rate	$10^{-0.6}$
Learning rate decay	0.9
Weights decay	$10^{-0.6}$
Epoch	500
Batch size	6
Momentum value	0.9
Patience	50

With the parameters above, the best weight is produced when the step reaches 10184 with an epoch value of 290, a train/loss value of at least 0.0009, and a value of 0.995 for Val/Accuracy.

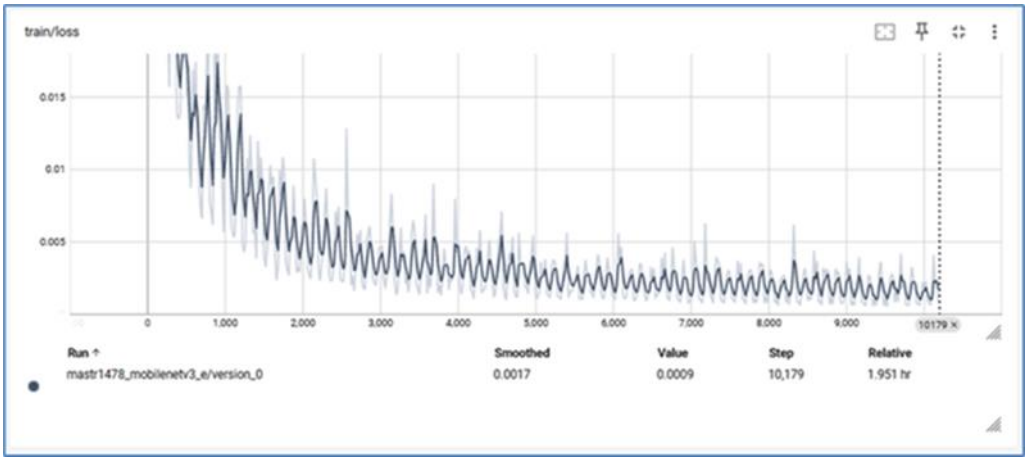

Figure 8 Train/loss Value Lightweight WaSR-T training

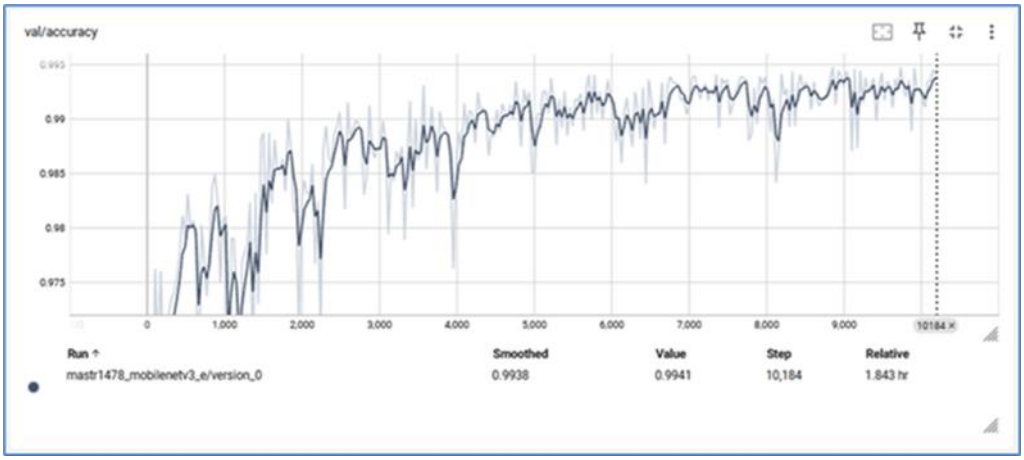

Figure 9 Val/Accuracy Value Lightweight WaSR-T training

3 Result and Discussion

In general, to evaluate the performance of a water surface object detection network, MODS Benchmark[16] can be used

With the new weight, we compared the original WaSR-T with the light and evaluated it using subjective assessments[17] and [18] with the following criteria:

- If the network output produces a boat class label that perfectly overlaps with the boat's pixel label location and area parallel to the boat's GT, then the subjective assessment is a true positive (TP)
- If the network output produces a boat class label with insufficient overlap with the location and the pixel label area of the boat is slightly spread out relative to the boat's GT, then the subjective assessment is a false positive (FP)
- If the network output assigns a boat class label outside the correct location, and the pixel label area of the boat is scattered relative to the GT boat, the subjective assessment classifies it as a false negative (FN)

We evaluate the overall metrics to measure the model network performance between the Lightweight WaSR-T and the native WaSR-T using several vital parameters.:

- precision Pr= TP/(TP+FP)
- recall Re=TP/(TP+FN)

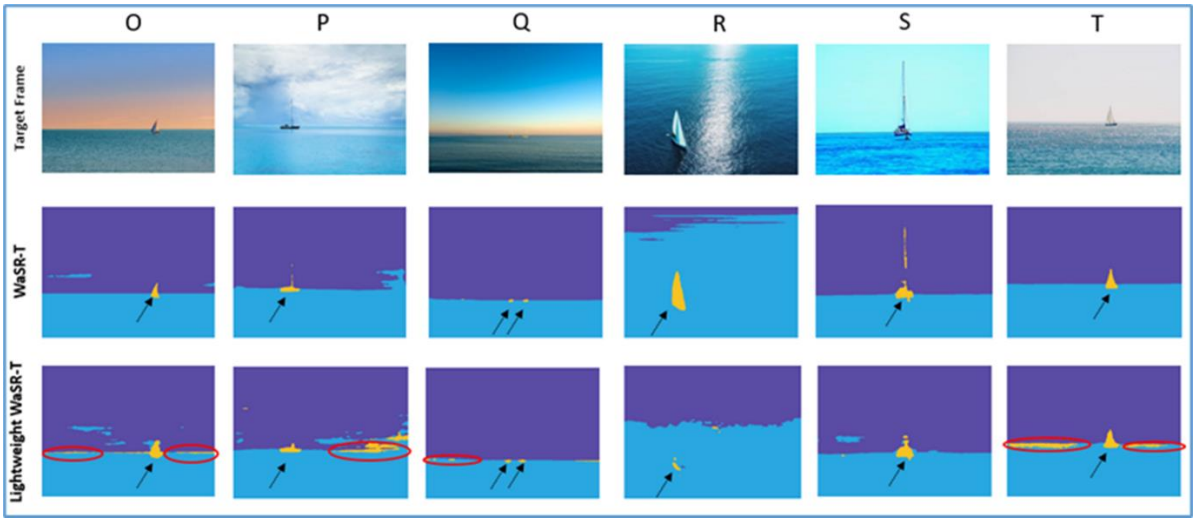

Figure 10 Sample Comparison image segmentation of Lightweight WaSR-T and original

According to the qualitative results in Table 3, the Lightweight WaSR-T, when assessing a true positive (TP), assigns a boat class label identical to the original and thoroughly detects the boat's pixel label area. The boat pixel label area is fully detected. If the network output produces a subjective assessment as FP, the Lightweight WaSR-T produces the boat pixel label area slightly spread out or smaller than the actual boat label.

Table 3 Computational load parameters of both network models run on ASUS ExpertBook

Model	Test Image	%CPU	%Memory	Total Processing Time	Rate [s/it]
WaSR-T	140	190	13.2	1:13:56	20.07
Lightweight WaSR-T	140	160	4.3	0:02:45	1.33

Table 4 Qualitative results of WaSR-T and Lightweight WaSR-T

Model	True Positive (TP)	False Positive (FP)	False Negative (FN)	Recall (Re)	Precision (Pr)
WaSR-T	94	40	15	70.15%	86.24%
Lightweight WaSR-T	77	49	14	61.11%	86.14%

In subjective assessment as FP, the number of false detections produced by the original is smaller than the Lightweight WaSR-T Network. The most common source of false detections is due to water reflections on the water surface or the interface between the water surface and the sky. The network still detects the boat's pixel label area in this case. However, the true value of the system lies in its ability to use the detected ship's class label to warn approaching ships, thereby enhancing safety operation the tsunami buoy. Therefore, the quantitative results of Lightweight WaSR-T with subjective assessment as TP and FN are useful for intelligent computer vision for tsunami buoy.

4 Conclusion

This discussion has covered developing and implementing the proposed Lightweight WaSR-T for detecting ships approaching tsunami buoys as an integral part of an intelligent computer vision system in the open ocean domain. Based on quantitative results and computational load evaluation, researchers designed Lightweight WaSR-T as the main component of an intelligent computer vision system on a tsunami buoy, promising further

implementation on single-board computing devices with small architectural components, such as Jetson Nano or similar.

References

[1] L. Zhao, F. Yu, J. Hou, P. Wang, and T. Fan, "The role of tsunami buoy played in tsunami warning and its application in South China Sea," *Theor. Appl. Mech. Lett.*, vol. 3, no. 3, p. 032002, 2013, doi: 10.1063/2.1303202.

[2] Y. Peng, Y. Qin, X. Tang, Z. Zhang, and L. Deng, "Survey on Image and Point-Cloud Fusion-Based Object Detection in Autonomous Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 22772–22789, Dec. 2022, doi: 10.1109/TITS.2022.3206235.

[3] L. Peng, H. Wang, and J. Li, "Uncertainty Evaluation of Object Detection Algorithms for Autonomous Vehicles," *Automot. Innov.*, vol. 4, no. 3, pp. 241–252, Aug. 2021, doi: 10.1007/s42154-021-00154-0.

[4] N. GENGEÇ, O. EKER, H. ÇEVİKALP, A. YAZICI, and H. S. YAVUZ, "Visual object detection for autonomous transport vehicles in smart factories," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 29, no. 4, pp. 2101–2115, Jul. 2021, doi: 10.3906/elk-2008-62.

[5] S. A. Khalil, S. Abdul-Rahman, S. Mutalib, and N. M. A. A. Dazlee, "Object Detection for Autonomous Vehicles with Sensor-based Technology Using YOLO," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 129–134, Mar. 2022, doi: 10.18201/ijisae.2022.276.

[6] B. Bovcon and M. Kristan, "WaSR—A Water Segmentation and Refinement Maritime Obstacle Detection Network," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12661–12674, Dec. 2022, doi: 10.1109/TCYB.2021.3085856.

[7] L. Žust and M. Kristan, "Temporal Context for Robust Maritime Obstacle Detection," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.05352

[8] M. Teršek, L. Žust, and M. Kristan, "eWaSR -- an embedded-compute-ready maritime obstacle detection network," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.11249

[9] B. Bovcon, J. Muhovic, J. Pers, and M. Kristan, "The MaSTr1325 dataset for training deep USV obstacle detection models," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 3431–3438. doi: 10.1109/IROS40897.2019.8967909.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, Dec. 2016, doi: 10.1109/CVPR.2016.90.

[11] A. Howard *et al.*, "Searching for MobileNetV3," May 2019, [Online]. Available: http://arxiv.org/abs/1905.02244

[12] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385

[16] B. Bovcon, J. Muhovič, D. Vranac, D. Mozetič, J. Perš, and M. Kristan, "MODS -- A USV-oriented object detection and obstacle segmentation benchmark," May 2021, doi: 10.1109/TITS.2021.3124192.

[17] Z. Chen and H. Zhu, "Visual Quality Evaluation for Semantic Segmentation: Subjective Assessment Database and Objective Assessment Measure," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5785–5796, Dec. 2019, doi: 10.1109/TIP.2019.2922072.

[18] G. Csurka, D. Larlus, and F. Perronnin, "What is a good evaluation measure for semantic segmentation?," in *Procedings of the British Machine Vision Conference 2013*, 2013, pp. 32.1-32.11. doi: 10.5244/C.27.32.