JSTT JOURNAL OF SCIENCE AND TRANSPORT TECHNOLOGY

Journal of Science and Transport Technology
Journal homepage: https://jstt.vn/index.php/en

UNIVERSITY OF TRANSPORT TECHNOLOGY

Enhancing Inland Waterway Safety and Management through Machine Learning-Based Ship Detection

Dung Van Tran[1], Thu-Hien Thi Hoang[2,*], Hai-Bang Ly[2]
[1]Port Authority of Inland Waterway Area 1, VIWA, Vietnam
[2]University of Transport Technology, Hanoi 100000, Vietnam

Article info
Type of article:
Original research paper

DOI:
https://doi.org/10.58845/jstt.utt.2024.en.4.3.39-52

[*]**Corresponding author:**
Email address:
hienhtt@utt.edu.vn

Received: 29/07/2024
Revised: 17/09/2024
Accepted: 19/09/2024

Abstract: Efficient ship detection is essential for inland waterway management. Recent advances in artificial intelligence have prompted research in this field. This study introduces a real-time ship detection model utilizing computer vision and the YOLO object detection framework. The model is designed to identify and locate common inland waterway vessels, such as container ships, passenger vessels, barges, ferries, canoes, fishing boats, and sailboats. Data augmentation techniques were employed to enhance the model's ability to handle variations in ship appearance, weather, and image quality. The system achieved a mean Average Precision (mAP) of 98.4%, with precision and recall rates of 96.6% and 95.0%, respectively. These results demonstrate the model's effectiveness in practical applications. Its ability to generalize across diverse vessel types and environmental conditions suggests its potential integration into video surveillance for improved maritime safety, traffic control, and search and rescue operations.

Keywords: Computer Vision; Ship Detection; YOLOv8 algorithm; Artificial intelligence; Roboflow platform;

1. Introduction

Ship detection in waterways is crucial for diverse maritime management applications. Accurate identification of vessels is the initial step in tracking their positions, movement patterns, and other pertinent data. This task is essential for the surveillance of both inland and international waterways [1, 2]. In the civilian sector, ship detection aids traffic regulation, mitigates the risk of collisions and accidents, and ensures vessel safety. It also facilitates infrastructure planning, improves cargo transport efficiency, and contributes to environmental protection. Additionally, it provides essential data for urban planning along waterways and for responding to emergencies. Precise ship detection is therefore a key factor in enhancing the overall management and fostering the sustainable development of waterways, particularly inland waterways.

Multiple technologies and methods currently exist for ship detection in inland waterways. Among these methods, radar is widely used [3–6]. Radar systems detect and track vessels within a designated area, operate under all weather conditions, and provide precise information about vessel location and movement. However, this method presents some challenges, notably high installation and maintenance costs and the necessity for human interpretation and data analysis. Surveillance camera systems installed at strategic locations along waterways capture visual data of traffic conditions. These systems may also incorporate pressure and sound sensors for vessel detection and tracking [7]. This approach offers the advantage of providing direct visual information about the waterway traffic; however, its effectiveness can be hindered by weather and lighting conditions and requires substantial data storage and analysis. Automatic Identification Systems (AIS) enable vessels to transmit and receive information regarding their position, speed, course, and other relevant data [8–10]. This system allows authorities and traffic management to monitor vessel activities in real time. It also readily integrates with other technologies, such as radar and GPS. However, AIS requires vessels to be equipped with compatible devices and may experience limitations in areas with weak or absent signal coverage. Other ship detection approaches employed globally include Global Positioning Systems (GPS) [11, 12], remote sensing and satellite imagery [13, 14], and sonar hydroacoustic sensors [15]. Each of these methods presents unique advantages and disadvantages, with the selection of an appropriate method depending on specific requirements, environmental factors, and budgetary constraints.

Existing ship detection methods for waterway management are often limited by cost and accuracy, and their performance is often affected by weather and environmental factors. Modern river and inland waterway management faces additional challenges such as increased vessel traffic, illicit activities, and personnel shortages [16]. The continuous rise in vessel traffic in rivers and inland waterways not only places a burden on the transportation system but also elevates the risk of collisions and accidents. Illegal activities, such as smuggling and unauthorized resource extraction, pose threats to both the environment and security. Additionally, relying on manual surveillance is expensive and risks human error. To address these issues, the development of automated, efficient, and affordable ship detection methods is crucial.

In recent years, spurred by the rapid advancement of the fourth industrial revolution, Artificial Intelligence (AI) has found growing applications across various societal sectors [17]. AI, a field in computer science, focuses on creating computer systems capable of performing tasks that typically require human intelligence. Machine learning (ML), a subset of AI, involves the development of techniques that enable systems to learn from data and solve specific problems. By constructing models for image-based object recognition, AI and ML have been explored for application in fields such as transportation [18], healthcare [19], agriculture [20], and retail [21]. These advances have led to AI and ML becoming integral components of science and technology, offering solutions to various problems through intelligent automation. Automating ship detection using AI and ML offers several benefits [22]. This enables continuous, 24/7 surveillance of all vessels in a defined area, thereby enhancing the overall monitoring efficiency. Automation also reduces the risk of violations and accidents by quickly identifying rule infractions and providing warnings about potential collisions. In addition, incorporating AI and ML into maritime surveillance systems improves their adaptability and dependability.

With the progress of AI, numerous studies have investigated ML models for ship detection. The key criteria for these models include the capacity to identify ships from different perspectives, detect various ship types, and achieve high accuracy. Recent research has focused on enhancing ship detection under low visibility conditions and across diverse image scenarios, as demonstrated by Liu et al. [23]. In this study, they applied AI and ML models, including Random Forest, Decision Tree, Naive Bayes, and Convolutional Neural Network (CNN), to 4000 satellite images of ships, resulting in a robust ship detection model [24]. Among these models, Random Forest demonstrated the highest accuracy, achieving 97.2% with Red Green Blue (RGB) images and 98.9% with Hue, Saturation, and Value (HSV) images. Additional research has explored ML models for ship detection based on radar and remote sensing data [25–27]. However, to date, few studies have utilized ML to develop a ship detection model based on surveillance camera imagery. This highlights the need for a robust AI/ML model capable of accurately recognizing various ship types from multiple angles. This research introduces a real-time ship detection model that utilizes YOLO V8 and trained on a diverse dataset of 17,707 images, with a particular focus on leveraging surveillance camera imagery, an approach not extensively explored in previous studies.

2. Database description and analysis

The dataset used in this study comprises 17,707 images sourced from two primary locations: (1) 756 images of various ship types, including container ships, passenger vessels, barges, ferries, canoes, fishing boats, and sailboats, captured by the authors using a smartphone and collected from the internet; and (2) 16,951 images obtained from open database repositories.

To ensure dataset diversity and cover a wide spectrum of real-world scenarios, the selected images include various ship types, hull sections, scales, viewpoints, lighting conditions, positions within the frame, and occlusion levels. The images also depict ships in complex environments. All images in the dataset were manually labeled with precise ship annotations and bounding boxes using the Roboflow platform, a tool designed for computer vision data management and preparation. The dataset was divided into three subsets for model development and evaluation: training (80%), validation (10%), and testing (10%). The training set is used to train the ML model, allowing it to learn features and make predictions. The validation set helps adjust model hyperparameters and monitor training progress. The test set provides an independent model performance assessment. A sample of the collected data is shown in Figure 1.

Fig. 1. Illustration of images collected in the dataset (includes open-source images from various online repositories)

3. Machine learning Methods

3.1. YOLO

3.1.1 Introduction of YOLO

YOLO (You Only Look Once), a computer vision algorithm introduced in 2015 by Joseph Redmon, is designed to detect objects in images [28]. Unlike traditional methods, which often require multiple processing steps, YOLO's unique architecture enables it to predict both bounding boxes and object classes in a single pass of an image. This streamlined approach results in exceptional computational efficiency, and thus, YOLO is particularly well-suited for real-time applications in which rapid object detection is essential [29]. For example, in autonomous vehicles navigating complex urban environments, the onboard computer vision system must rapidly and accurately identify pedestrians, other vehicles, and traffic signs. YOLO's ability to process an entire image and generate all necessary predictions simultaneously makes it a strong candidate for such tasks. This real-time capability is vital for ensuring the safety and responsiveness of self-driving cars. In addition to its speed advantage, YOLO has received recognition for its accuracy. Since its initial release, multiple versions of YOLO have been developed, each iteratively improving both speed and accuracy. This ongoing development has made YOLO a popular choice for various object detection applications, including security, surveillance, robotics, and industrial automation [29].

3.1.2. YOLO working mechanism

The YOLO model, which was initially trained on the ImageNet dataset, was adapted for object detection [28,30]. The final layer predicts both the likelihood of an object belonging to a specific class and the coordinates defining its location in the image. YOLO realizes this by partitioning the input image into an S x S grid. Each cell in the grid is tasked with detecting objects whose centers fall within its boundaries. Each cell generates multiple bounding box predictions, each with an associated confidence score indicating the model's certainty that the box contains an object and the accuracy of its prediction. To refine the output, YOLO selects the most accurate bounding box for each individual cell. This is achieved by calculating the Intersection over Union (IOU), which is a metric measuring the overlap between the predicted and actual bounding boxes, and selecting the box with the highest IOU. Non-maximum suppression (NMS) further improves YOLO's accuracy by eliminating redundant or inaccurate bounding boxes after the initial predictions. This ensures that each object is represented by a single, well-defined bounding box.

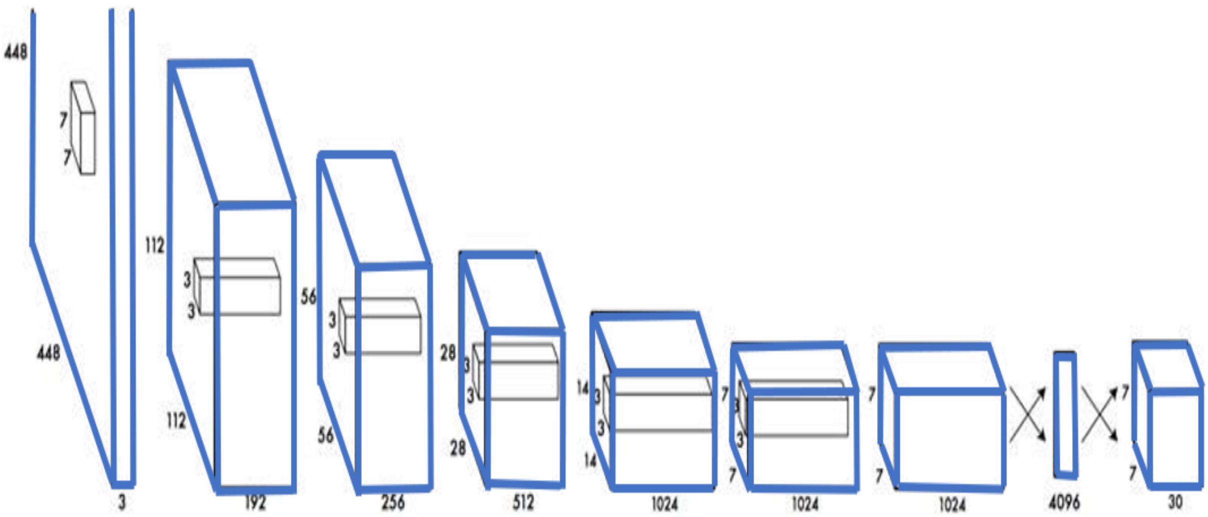


Fig. 2. Illustration of YOLO's structure (adapted from [30])

For instance, in an image of multiple ships, YOLO first divides the image into a grid. Each cell then analyzes its assigned area and predicts multiple bounding boxes for potential ships. YOLO then calculates the IOU for each box, selecting the one with the highest overlap with the actual ship. Finally, NMS removes any redundant or overlapping boxes, leaving only accurate bounding boxes for each ship in the image. This multi-step process allows YOLO to efficiently and accurately detect objects in real-time, making it useful in various applications, such as autonomous vehicles, security systems, and industrial automation.

3.2. Performance indices of model

Performance metrics are the primary tools to assess the accuracy and effectiveness of object detection models. The key metrics were mean average precision (mAP), precision, and recall [29]. To understand these metrics, it is helpful to first define four common variables using a binary confusion matrix, as shown in Fig. 3. The axes of this matrix represent two properties of the label: 'True' and 'False'. When both the actual and predicted labels are 'True', the case is labeled as true positive (TP). When both labels are 'False', it's labeled as true negative (TN). False negative (FN) denotes the situation where the actual label is 'True' but the predicted label is 'False'. Conversely, false positive (FP) indicates that the actual label is 'False' while the predicted label is 'True' [31].

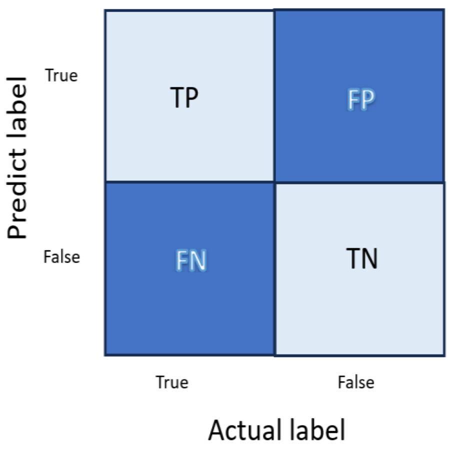


Fig. 3. Binary Confusion Matrix

Precision, ranging from 0 to 1, represents the proportion of correctly predicted "True" labels among all predicted "True" labels. In the ship detection context, high precision indicates high confidence in the identification of a specific ship type:

$$P = \frac{TP}{TP+FP} \in [0, 1]$$

Recall, which ranges from 0 to 1, represents the proportion of correctly predicted "True" labels among the total number of actual "True" labels. High recall for ship detection indicates the algorithm's strong ability to detect all instances of a particular ship type in the dataset:

$$R = \frac{TP}{TP+FN} \in [0, 1]$$

mAP is a metric used to evaluate the performance of computer vision models. It is calculated as the average of the Average Precision (AP) metric across all classes in the model. The mAP can be used to compare different models on the same task or different versions of the same model. Higher mAP values ranging from 0 to 1 indicate better performance. For a given category, Average Precision (AP) refers to the area under the curve plotted using recall and precision:

$$AP_i = \int_0^1 P_i\,(R_i)dR_i$$

The mAP of multiple categories is defined as follows:

$$mAP = \frac{\sum_1^n AP_i}{n} \in [0, 1]$$

In ML, optimizing the loss function is critical for effective model training. For object detection tasks using the YOLO algorithm, the loss function is composed of three components: box loss, class loss, and object loss.

Box loss measures the algorithm's capacity to accurately locate an object's center and predict its bounding box. It quantifies the discrepancy between the predicted and actual bounding boxes for objects in the training data. A smaller box loss value indicates a close match between the predicted and actual bounding boxes. Here, object loss is the probability that an object exists within a defined region of interest (ROI). A high object loss value indicates that the object exists in the target image region. Optimizing object loss focuses on improving the model's ability to correctly identify the presence of objects. Class loss evaluates the algorithm's ability to assign the correct class label to each detected object. The error between the predicted class probabilities of each object and the ground truth labels was measured. Lower class loss values correspond to more accurate class predictions. The variation in box, class, and object losses is typically tracked over epochs, which represents the number of iterations through the entire training dataset. Because the dataset was divided into batches, each epoch involved training on all batches. The chosen number of epochs, often determined by experience and intuition, can be quite large, with values exceeding 3000 not uncommon.

3.3. Methodology workflow

The ship detection workflow using the YOLO algorithm is illustrated in Figure 4. This process involves the following steps:

a) Data preparation: Images of the ships were collected and labeled.

b) Data splitting: The dataset was divided into training, validation, and testing sets.

c) YOLO model training: The YOLO version is selected, training parameters (epochs, learning rate) are configured, and the model is trained on the training set.

d) Model validation: The model was evaluated on the validation set using performance metrics.

e) Model testing: The model was tested on a testing set. If the performance requirements are satisfied, the model proceeds to deployment. Otherwise, it moves to optimization and fine-tuning steps.

f) Model optimization and fine-tuning: Based on the testing results, the hyperparameters were adjusted and optimized. This may involve returning to either the data preparation or model training steps.

g) Model deployment: The trained model meets the performance criteria and is exported and deployed on the target device or application for ship detection.

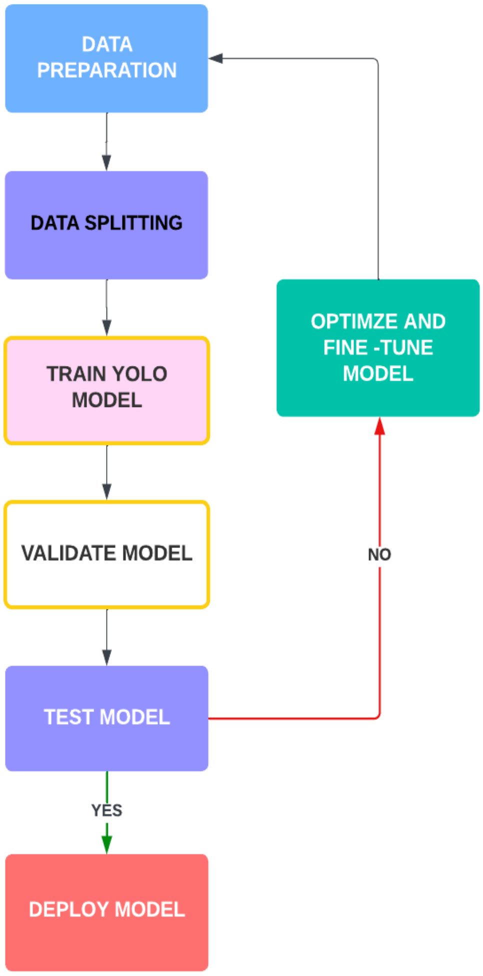


Fig. 4. Methodology workflow

4. Results and discussion

4.1. Model construction and development

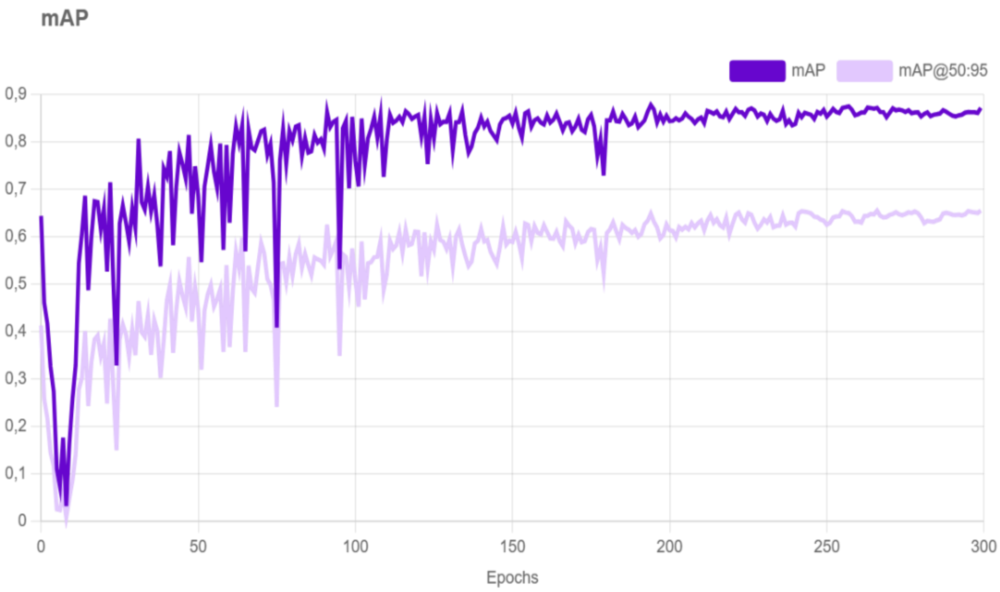
A ship detection model was developed using the YOLOv8 algorithm, leveraging its pretrained architecture for object detection tasks. The 17,707 image dataset was preprocessed to standardize image dimensions and enhance training efficiency. The preprocessing steps included resizing, normalization, and data augmentation techniques, which improved the model's ability to handle variations in ship appearance, weather conditions, and image quality. The pre-trained YOLOv8 model was then fine-tuned on the prepared dataset using backpropagation and gradient descent optimization to minimize the combined box, object, and class loss functions. Hyperparameter tuning, which involves adjusting the learning rate and batch size, was performed to optimize model performance. Note that the pretrained YOLOv8 model was fine-tuned using the automated capabilities provided by the Roboflow platform to optimize its performance on the curated dataset. The model's performance was rigorously assessed on the validation set using metrics such as mAP, precision, and recall, to guide further iterative refinement. The final model was evaluated on an independent test set to assess its generalizability. The results of this evaluation are presented in the following section.

The initial model was trained using 1139 images. Although this limited dataset provided a foundation for model development, the resulting performance, as illustrated in Figure 5, revealed an mAP of 87.2%, a precision of 89.7%, and a recall of 83.9%. These metrics, while indicating promising initial results, also highlight the potential for improvement with a larger and more diverse training set. The relatively high precision suggests that the model was generally confident in its positive ship identifications, with few false positives. However, the lower recall indicates that a significant proportion of actual ships in the images were not detected. This could be attributed to the limited size and variability of the initial dataset, which may not have adequately captured the full range of ship appearances and environmental conditions encountered in real-world scenarios. The performance obtained on this smaller dataset served as a baseline for further model development. Subsequent training iterations incorporated a larger dataset to address the limitations identified in the initial phase and ultimately enhance the model's overall accuracy and robustness.

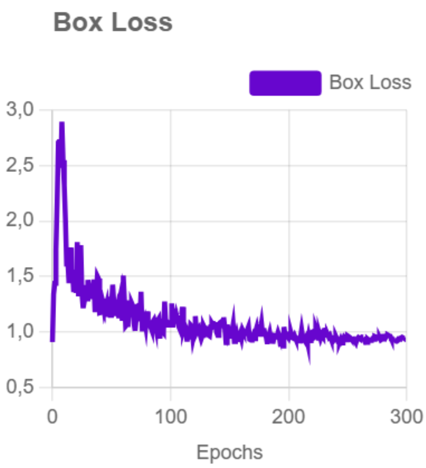
The model was further refined by incorporating additional images and applying image augmentation techniques, such as rotation, expanding the total dataset to 17,707 images. Despite reducing the loss functions observed during training on this augmented dataset (Figure 6), the overall performance metrics did not substantially improve. In fact, the recall rate decreased by 1.8% compared to the previous model iterations. The final performance metrics for this version were: mAP 88.8%, precision 91.0%, and recall 82.1%. The reduction in recall despite the increased dataset size and augmentation technique indicates that the specific augmentation methods employed may have introduced artifacts or noise that negatively affected the model's detection capabilities. Further investigation and adjustment of the augmentation strategies are required to ensure their positive impact on model performance.

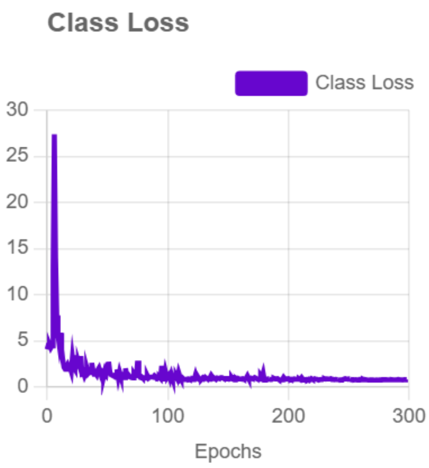
The model was further refined through a detailed examination of the dataset, during which low-quality images and those with incorrect or missing labels were removed (Figure 7). The curation process reduced the dataset size from 17,707 images to 16,530 images. Subsequent retraining on the refined dataset resulted in a significant improvement in model accuracy and a decrease in all loss functions compared to the previous iteration (Figure 8). This underscores the importance of data quality in ML because even a small proportion of problematic images can adversely affect model performance. Removing these potentially misleading or uninformative examples enabled the model to better learn the distinguishing features of ships and their surrounding environments. Consequently, more accurate predictions and improved generalizability to new images were obtained. The enhanced performance obtained following dataset curation demonstrates the critical role of data preprocessing in the development of robust and reliable ML models.

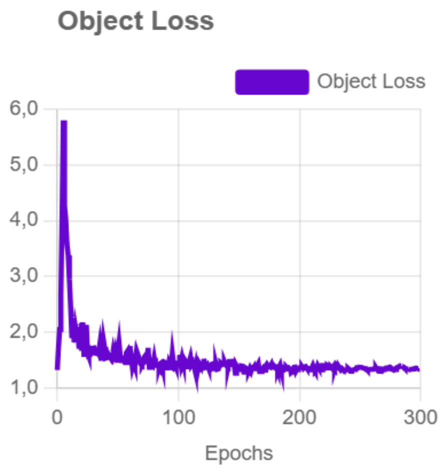
Table 1 presents a comprehensive summary of the performance metrics for each model version. This table serves as a valuable reference for comparing model iterations and assessing their predictive capabilities.



(a)

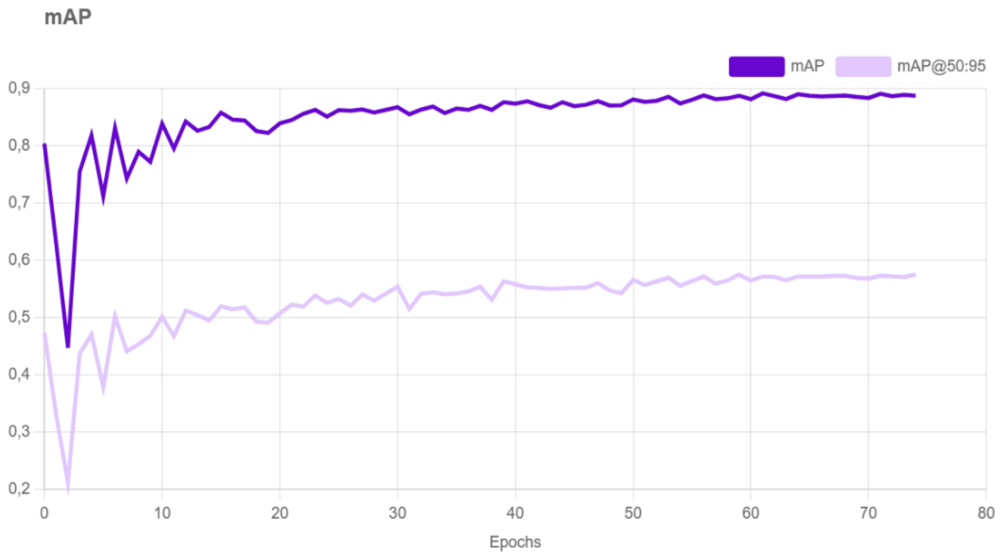




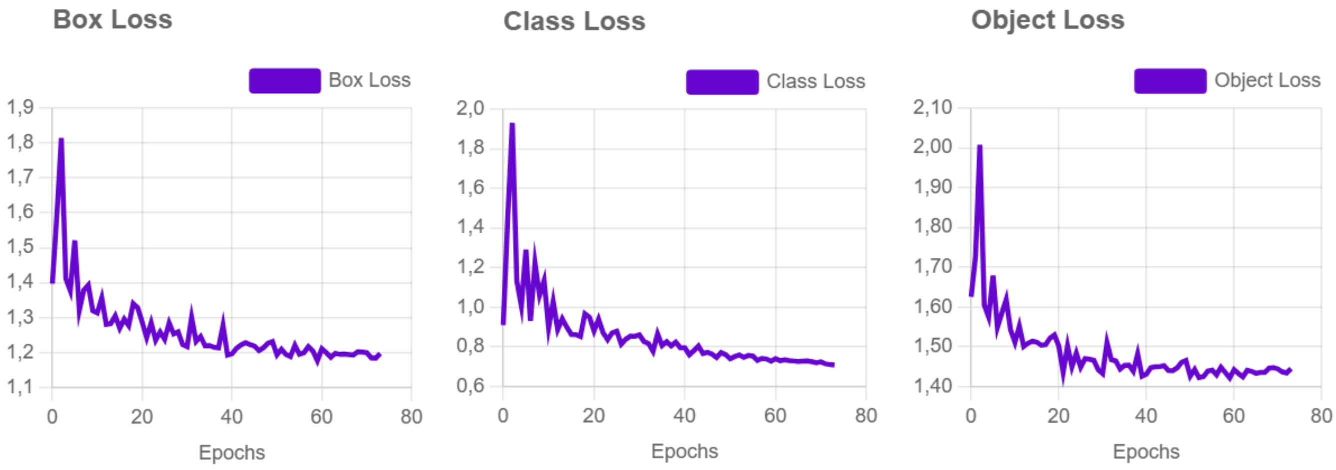


(b) (c) (d)

Fig. 5. Training process of Version 1 model: (a) mAP; (b) Box loss; (c) Class Loss; (d) Object Loss

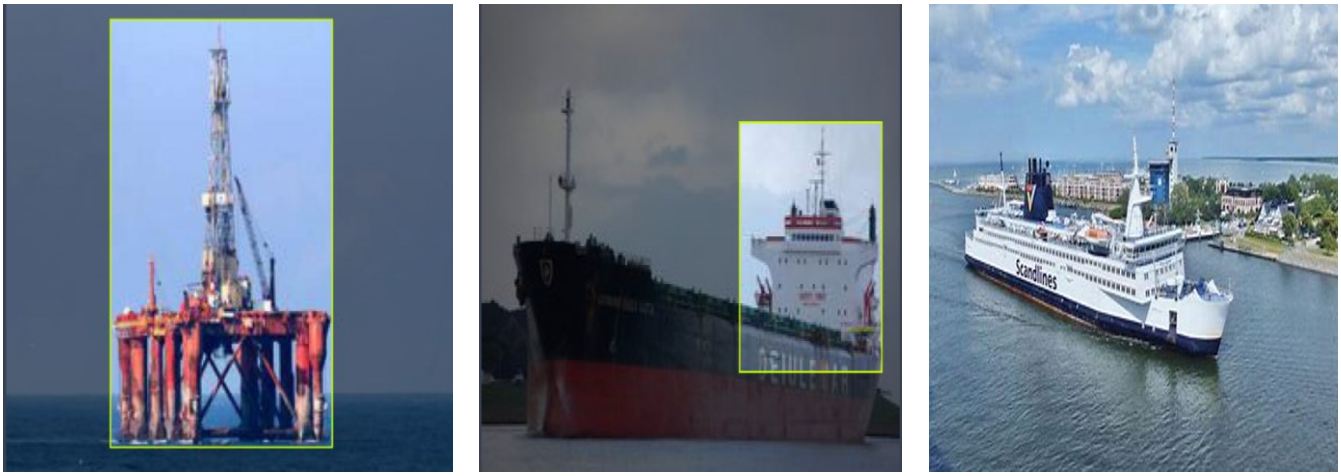


(a)



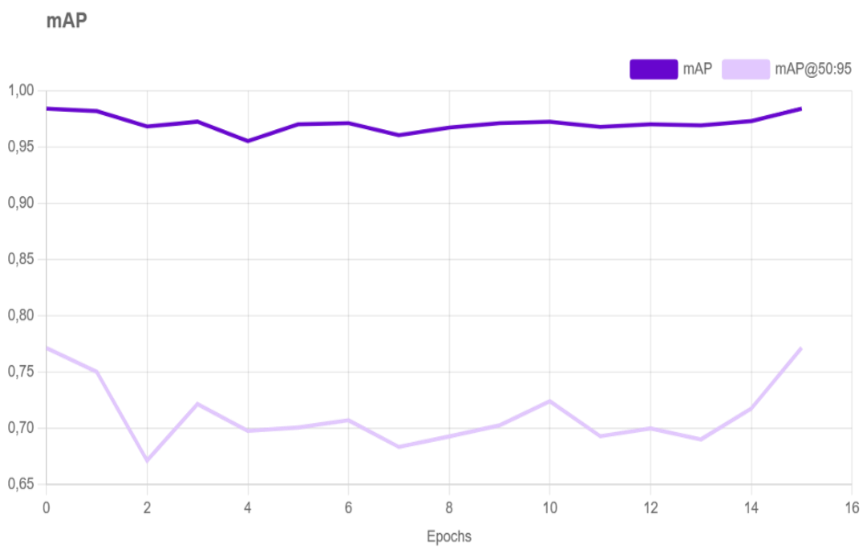
(b) (c) (d)

Fig. 6. Training process of Version 2 model: (a) mAP; (b) Box loss; (c) Class Loss; (d) Object Loss

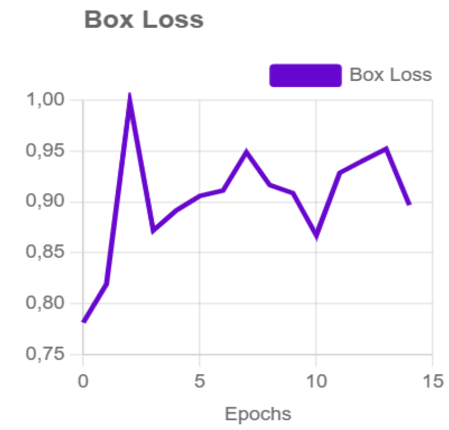


(a) (b) (c)

Fig. 7. Illustration of excluded images from the database: (a, b) Image with labeling problems; (c) Image without labeling



(a)



(b) (c) (d)

Fig. 8. Training process of Version 3 model: (a) mAP; (b) Box loss; (c) Class Loss; (d) Object Loss

Table 1. Accuracy of different developed YOLO models

Version	No. of images	mAP (%)	Precision (%)	Recall (%)
Version 1	1139	87.2	89.7	83.9
Version 2	17707	88.8	91.0	82.1
Version 3	16530	98.4	96.6	95

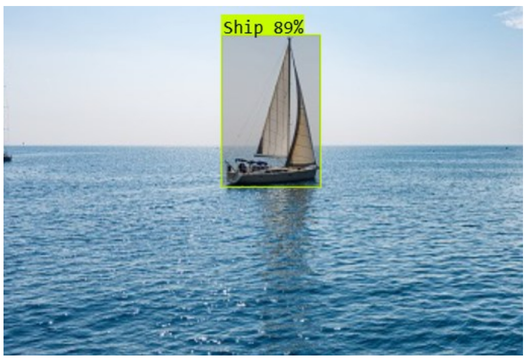
4.2. Model performance

In this section, the final refined ship detection model's performance is evaluated using an independent test set. The evaluation focused on the model's accuracy in identifying and locating diverse ship types under various image conditions.

A selection of images from the test set along with their corresponding detection results are presented to visually demonstrate the model's capabilities. Each image shows the detected ships enclosed within bounding boxes, accompanied by confidence scores indicating the model's certainty in its predictions. These examples illustrate the model's proficiency in detecting a wide range of ship types, including container ships, passenger vessels, barges, ferries, canoes, fishing boats, and sailboats, under diverse viewpoints, lighting conditions, and occlusion levels.

The evaluation results indicate that the model performed well on the test set. The high mAP, precision, and recall scores across the various ship classes demonstrate the model's accuracy and robustness. Several factors contributed to this performance, including the utilization of the efficient and accurate YOLOv8 algorithm, the comprehensive and diverse dataset that reflects real-world imaging conditions, and the thorough data curation process that removed inaccurate or noisy samples, thereby allowing effective model learning.

(a)



(b)

(c)

(d)

Fig. 9. Illustration of model performance showing the confidence on: (a) fishing, (b) sailing, (c) container ship, (d) passenger ships

The model's ability to generalize across diverse ship types, viewpoints, lighting conditions, and occlusion levels positions it as a potential tool for various maritime applications, such as surveillance, traffic management, and search and rescue operations. Reference [32], utilizing Faster R-CNN on a dataset of 7000 ship images, reported an mAP between 89.38% and 93.92%. In contrast, the proposed YOLOv8 model, which was trained on a more extensive and diverse dataset, achieved a considerably higher mAP of 98.4%. This result indicates the potential of our approach, which employs the YOLOv8 architecture and a larger dataset, to achieve superior ship detection accuracy in various scenarios.

4.3. Model deployment

To illustrate the practical application and real-world potential of the developed ship detection model, a mobile-friendly deployment was implemented. The model is optimized for efficient execution on mobile devices, retaining core detection capabilities and reducing computational requirements. A QR code linked to a web-based interface hosting the model is provided to facilitate access and demonstration. Users can scan the code on their smartphone or tablet to directly interact with the proposed model. The interface allows uploading or capturing images of waterways, which are then processed in real time. The detected ships are highlighted in the image and displayed with their respective confidence scores (Fig. 10).

Try on mobile

Fig. 10. Model deployment using QR code

This deployment strategy aims to enhance accessibility by using a QR code and a web-based interface, thereby making the model readily available to users who do not require specialized software or technical knowledge. The interface allows for real-time interaction with the model, allowing users to directly experience its capabilities through instant processing and analysis of uploaded or captured images. The mobile-friendly nature of this deployment emphasizes the model's potential for real-world maritime surveillance, traffic monitoring, and recreational boating. This interactive demonstration underscores the practicality of the proposed model and its potential integration into existing or future maritime monitoring systems.

5. Conclusions and Perspectives

In this study, an automated ML model for detecting various ship types was developed. Rigorous evaluation of the model's predictive capabilities were rigorously evaluated using mAP, precision, and recall. The results showed an mAP of 98.4%, precision of 96.6%, and recall of 95.0%, demonstrating the model's high reliability and accuracy in ship identification. These findings highlight the potential of the model for real-world maritime applications. For instance, integrating the model into existing vessel traffic service (VTS) systems could provide operators with real-time information on vessel locations and classifications, thereby aiding traffic management, collision avoidance, and search and rescue operations, ultimately enhancing maritime safety and efficiency. Additionally, the model's accurate identification of specific ship types can be used to monitor compliance with fishing regulations. Automatic vessel classification would enable authorities to more effectively identify and track illegal, unreported, and unregulated (IUU) fishing activities, thereby contributing to sustainable fisheries management. The model's robust performance across diverse conditions, including variations in lighting and weather, suggests its potential deployment in challenging environments where manual surveillance is difficult or impractical, thus extending the reach of maritime monitoring and enforcement.

In summary, the proposed automated ship detection model represents a significant step forward in maritime surveillance and management. The accuracy, reliability, and adaptability of the system to various scenarios opens up a wide range of potential applications for improving safety, efficiency, and sustainability in the maritime sector. Although this study established a reliable method for general ship detection, future investigations could explore the finer-grained classification of individual ship types, potentially increasing their utility in specific maritime applications.

References

[1]. N. Wawrzyniak, T. Hyla, A. Popik. (2019). Vessel Detection and Tracking Method Based on Video Surveillance. *Sensors*, 19(23), 5230.

[2]. N. Wawrzyniak, T. Hyla, I. Bodus-Olkowska. (2022). Vessel identification based on automatic hull inscriptions recognition. *PLoS ONE*, 17(7), e0270575.

[3]. S. Zainuddin, I. Pasya, N.E. Abd. Rashid, N. Zaiaami, A. Maryam, R.S.A. Raja Abdullah, M.S.A. Megat Ali. (2019). Maritime radar: a review on techniques for small vessels detection. *Journal of Electrical and Electronic Systems Research (JEESR)*, 14, 30-45.

[4]. L. Daniel, S. Hristov, X. Lyu, A.G. Stove, M. Cherniakov, M. Gashinova. (2017). Design and Validation of a Passive Radar Concept for Ship Detection Using Communication Satellite Signals. *IEEE Transactions on Aerospace and Electronic Systems*, 53(6), 3115-3134.

[5]. A. Dzvonkovskaya, K.-W. Gurgel, H. Rohling, T. Schlick. (2008). Low power High Frequency Surface Wave Radar application for ship detection and tracking. *2008 International Conference on Radar*, pp. 627-632.

[6]. J. Barnum. (1986). Ship detection with high-resolution HF skywave radar. *IEEE Journal of Oceanic Engineering*, 11(2), 196-209.

[7]. S. Fefilatyev, D. Goldgof, M. Shreve, C. Lembke. (2012). Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Engineering*, 54, 1-12.

[8]. Y. Lu, H. Ma, E. Smart, B. Vuksanovic, J. Chiverton, S.R. Prabhu, M. Glaister, E. Dunston, C. Hancock. (2021). Fusion of Camera-based Vessel Detection and AIS for Maritime Surveillance. *2021 26th International Conference on Automation and Computing (ICAC)*, pp. 1-6.

[9]. T.N. Hannevik, Ø. Olsen, A.N. Skauen, R. Olsen. (2010). Ship detection using high resolution satellite imagery and space-based AIS. *2010 International WaterSide Security Conference*. pp. 1-6.

[10]. F.M. Vieira, F. Vincent, J.-Y. Tourneret, D. Bonacci, M. Spigai, M. Ansart, J. Richard. (2016). Ship detection using SAR and AIS raw data for maritime surveillance. *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 2081-2085.

[11]. S. Ryan, M. Petovello, G. Lachapelle.

(1998). Augmentation of GPS for ship navigation in constricted water ways. *Proceedings of the 1998 National Technical Meeting of The Institute of Navigation*, pp 459-467.

[12]. C. Zhang, C. Guo, D. Zhang. (2018). Ship navigation via GPS/IMU/LOG integration using adaptive fission particle filter. *Ocean Engineering*, 156, 435-445.

[13]. X. Wang, J. Liu, O.I. Khalaf, Z. Liu. (2021). Remote sensing monitoring method based on BDS-based maritime joint positioning model. *Computer Modeling in Engineering & Sciences*, 127(2), 801-818.

[14]. L. Yao, Y. Liu, Y. He. (2018). A Novel Ship-Tracking Method for GF-4 Satellite Sequential Images. *Sensors*, 18(7), 2007.

[15]. N. Wawrzyniak, G. Zaniewicz. (2016). Detecting small moving underwater objects using scanning sonar in waterside surveillance and complex security solutions. *2016 17th International Radar Symposium (IRS)*, pp 1-5.

[16]. B. Solomon, E. Otoo, A. Boateng, D. Ato Koomson. (2021). Inland Waterway Transportation (IWT) in Ghana: A case study of Volta Lake Transport. *International Journal of Transportation Science and Technology*, 10(1), 20-33.

[17]. P.S. Sajja. (2021). Introduction to Artificial Intelligence. *Sajja, P.S. (ed.) Illustrated Computational Intelligence: Examples and Applications*, pp. 1-25. Springer, Singapore.

[18]. O. Hotkar, P. Radhakrishnan, A. Singh, N. Jhamnani, R.V. Bidwe. (2023). U-Net and YOLO: AIML Models for Lane and Object Detection in Real-Time. *Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing*, pp. 467-473. ACM, Noida India.

[19]. S. Narayanan, B.G.Sudha, A.Kishan, N.Darapaneni, A.R. Paduri. (2024). Segmentation and Classification of Medical Images for Pneumonia Diagnosis. *Interdisciplinary Research in Technology and Management*, pp. 185-197. CRC Press.

[20]. S. Pooja, K.P. Kumar, A.S. Sushmitha Urs, V.B. Raj, B.R. Madhu, S.V. Kumar. (2024). Cognitive Object Detection: A Deep Learning Approach with Auditory Feedback. *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, pp 162-167. IEEE.

[21]. P. Kaushik, S.P. Singh Rathore, P. Kaur, H. Kumar, N. Tyagi. (2023). Leveraging Multiscale Adaptive Object Detection and Contrastive Feature Learning for Customer Behavior Analysis in Retail Settings. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11, 326-343.

[22]. Z. Shao, W. Wu, Z. Wang, W. Du, C. Li. (2018). SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. *IEEE Transactions on Multimedia*, 20(10), 2593-2604.

[23]. R.W. Liu, W. Yuan, X. Chen, Y. Lu. (2021). An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system. *Ocean Engineering*, 235, 109435.

[24]. A.O. Hashi, I.H. Hussein, O.E.R. Rodriguez, A.A. Abdirahman, M.A. Elmi. (2022). Ship Detection Approach Using Machine Learning Algorithms. *Saeed, F., Mohammed, F., and Ghaleb, F. (eds.) Advances on Intelligent Informatics and Computing*. pp. 16-25. Springer International Publishing, Cham.

[25]. T. Zhang, X. Zhang, J. Shi, S. Wei. (2020). HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167, 123-153

[26]. Z. Long, W. Suyuan, C. Zhongma, F. Jiaqi, Y. Xiaoting, D. Wei. (2020). Lira-YOLO: A lightweight model for ship detection in radar images. *Journal of Systems Engineering and Electronics*, 31(5), 950-956.

[27]. J. Li, C. Qu, J. Shao. (2017). Ship detection in SAR images based on an improved faster R-CNN. *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, pp 1-6. IEEE.

[28]. M. Hussain. (2024). Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo. *IEEE Access*, 12, 42816-42833.

[29]. L. Li, L. Jiang, J. Zhang, S. Wang, F. Chen. (2022). A complete YOLO-based ship detection method for thermal infrared remote sensing images under complex backgrounds. *Remote Sensing*, 14(7), 1534.

[30]. J. Redmon, S. Divvala, R. Girshick, A. Farhadi. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp 779-788.

[31]. N. Seliya, T.M. Khoshgoftaar, J.V. Hulse. (2009). A study on the relationships of classifier performance metrics. *2009 21st IEEE international conference on tools with artificial intelligence*, pp 59-66. IEEE.

[32]. K. Kim, S. Hong, B. Choi, E. Kim. (2018). Probabilistic ship detection and classification using deep learning. *Applied Sciences*, 8(6), 936.