Real-Time Ship Detection and Text Recognition Using YOLO-OCR for Smart Port Applications

Abstract

This study presents a new real-time computer vision architecture designed for maritime environments that combines YOLO for object detection and PaddleOCR for text recognition. The YOLO algorithm was tuned to identify ships and text regions using a dataset of over 600 annotated photos. Two output layers with good detection accuracy (mAP 0.90, F1-score 0.89) were obtained by removing the smallest detection scale (P3) in order to speed up inference and lower computational complexity. making it appropriate for marine applications with bandwidth constraints. To enhance OCR robustness in low-quality or variable lighting conditions, detected text regions undergo a lightweight preprocessing pipeline consisting of grayscale conversion, contrast enhancement, and noise reduction. The proposed framework enables automated and continuous ship monitoring, thereby supporting compliance verification, port logistics, and security operations in real seaport environments. Furthermore, the architecture demonstrates scalability toward large-scale, real-time maritime surveillance systems.

Port Automation, Object Detection, Text Recognition, Maritime Surveillance, YOLOv5, PaddleOCR.

1 Introduction

Seaports are crucial hubs for international logistics because more than 80% of world trade is conducted by sea [1]. However, as demand has increased, complex problems have surfaced, including restricted physical growth capacity, high operating costs, port congestion, and emission regulations. These constraints have led to a need for more data-driven, intelligent solutions in seaport operations.

Advanced artificial intelligence (AI) capabilities have shown great promise to revolutionize logistics management, especially in the areas of machine learning and computer vision [2]. Contemporary ports produce enormous amounts of data by implementing technology like digital surveillance, autonomous systems, and Internet of Things sensors. This data-rich environment is perfect for implementing sophisticated visual analytics technologies that increase operational efficiency, decrease human error, and improve decision-making.

This work proposes an integrated visual recognition system that responds to industry needs by combining YOLOv5 for object identification and PaddleOCR for text recognition. Ships are automatically identified by the system, which then uses live photos to derive their names or registration numbers. Its scalable and effective digital port management solution combines reliable OCR capabilities with fast, single-stage object identification. By reducing the amount of physical interaction required for visual inspection operations, this technology lowers dangers and enables faster and more accurate completion of maritime activities. The suggested approach may also have broader applications in port security, maritime surveillance, and non-contact customs inspections.

2 Literature Review

Two crucial computer vision tasks that are incorporated into the proposed approach are text recognition and object identification. A dependable system that can identify ships and interpret the textual information associated with them, such as registration names and serial numbers, might be created with the aid of this hybrid technique. To achieve this, the project makes use of YOLOv5 for object detection and PaddleOCR for text recognition, both of which have shown state-of-the-art performance in their respective domains.

2.1 Object Detection

Deep learning-based generic object recognition has gained significant attention in computer vision and machine learning, and it has been the subject of much research in the last 10 years [3]. Significant advancements have been made throughout time in object detecting systems' efficiency and accuracy. Crucially, there are two main categories of contemporary object detection algorithms: one-stage and two-stage detectors. When it comes to the system's overall adaptability for real-time or embedded applications, as well as its detection speed and accuracy, this classification is crucial.

Two-Stage Detectors: There are several variations of two-stage object detection algorithms, with the most notable being the R-CNN family, including R-CNN [4], Fast R-CNN [5], and Faster R-CNN [6]. The original R-CNN marked a major breakthrough in modern

object detection by integrating two key principles: region proposal and convolutional neural networks (CNNs). It detects objects by first generating region proposals or anchor boxes to locate potential objects, and then classifies those regions using features extracted by a CNN. This approach achieved state-of-the-art performance at the time. However, due to its multi-step training pipeline and the separate processing of region proposals and classification, R-CNN models exhibit high computational and memory demands. To address these limitations, Fast R-CNN was introduced as an optimized framework that significantly improved both speed and efficiency. It employed a single-step training process and incorporated a multi-task loss function, allowing the model to simultaneously train the classification and bounding box regression heads. This effectively eliminated the need for separate training stages and reduced overall complexity.

Despite these improvements, the region proposal stage remained a major bottleneck, primarily due to its reliance on the selective search algorithm, which is computationally expensive and slow. To overcome this, Faster R-CNN was developed, introducing a novel component known as the Region Proposal Network (RPN). The RPN replaced the traditional selective search algorithm, enabling the model to generate region proposals directly and efficiently. This integration made Faster R-CNN both faster and more accurate, marking another significant step forward in object detection technology.

One-Stage Detectors: One-stage object detection algorithms perform detection in a single pass through the network, making them significantly faster than two-stage approaches. The most notable one-stage detectors include YOLO V1 to V5 [7], YOLO V7 [8] , Yolo V8 [9], SSD (Single Shot MultiBox Detector) [10], and RetinaNet [11].

YOLO was the first algorithm to reframe object detection as a direct regression problem—simultaneously predicting bounding boxes and class probabilities in a single forward pass. This innovation greatly accelerated detection speed and made real-time processing possible. The original YOLO model could process images at 45 frames per second (FPS), while a smaller version achieved up to 155 FPS, making it ideal for real-time applications such as video surveillance or autonomous navigation.

Despite its impressive speed, the YOLO architecture exhibits certain limitations. In particular, it tends to underperform in detecting small objects and is more prone to localization errors when compared to two-stage detectors such as Faster R-CNN. These limitations primarily stem from design decisions aimed at maximizing inference speed. However, successive versions of YOLO—including YOLOv4, YOLOv5, and YOLOv8—have demonstrated notable improvements in both accuracy and computational efficiency. It is important to note that each version of YOLO includes multiple model variants, differentiated by the number of parameters, computational requirements, and performance characteristics. The choice of model should be guided by the specific task and the level of accuracy required. For this study, YOLOv5s has been selected due to its balance between accuracy and speed. With approximately 7.2 million parameters and achieving an accuracy of around 36.7% mAP@0.5 on the COCO dataset, YOLOv5s provides real-time inference capabilities, making it particularly well-suited for the application at hand.

SSD, on the other hand, uses a dense set of default boxes and combines predictions from multiple feature maps at different resolutions. This multi-scale approach allows SSD to effectively detect objects of varying sizes, distinguishing it from YOLO in its handling of small and large objects. RetinaNet introduced a novel focal loss function, which addresses the

extreme foreground-background class imbalance in dense object detection. This innovation enables RetinaNet to achieve accuracy levels comparable to, and sometimes exceeding, those of two-stage detectors—making it one of the most accurate one-stage detectors to date.

Pros and Cons in Context of this Project: A crucial consideration in the field of object text detection and recognition is the trade-off between processing speed and detection accuracy, particularly in real-world logistical settings like seaports. Two-stage detectors, like Faster R-CNN, provide a high degree of precision, but they are less appropriate for real-time deployment scenarios because to their computational cost and slower inference time.

For real-time tasks including logistics automation, ship number recognition, and tracking, a one-stage detection strategy using YOLO (You Only Look Once) is employed in this project due to its remarkable speed and efficiency. Thanks to YOLO, which scans the entire image in a single pass, item detection and recognition can be done almost instantly. In dynamic environments where prompt choices are crucial, such as seaports, this is a huge advantage.

One-stage detectors are perfect for real-world applications due to their speed advantage, even if they typically offer slightly lower accuracy than two-stage devices. For example, the system must instantly recognize ship names and registration numbers in real-time video feeds from port surveillance cameras. YOLO's high frame rate and quick reaction are quite helpful in this situation.

Thus, the suggested system prioritizes scalability and real-time speed over slight accuracy benefits, which is a sensible and sensible choice in the context of smart port operations.

2.2 OCR Text Recognition

Text Recognition Using OCR: Optical Character Recognition (OCR) is a technology used to detect and extract text from images through a series of image processing and recognition algorithms [12]. It converts visual text into machine readable formats, enabling further processing such as data extraction, speech synthesis, or execution of machine level instructions. OCR achieves this by mimicking the optical mechanisms of the human eye. Although its performance heavily depends on the quality and clarity of the input, OCR is capable of recognizing both printed and handwritten text. This technology enables the transformation of various document types including scanned PDFs or images captured by digital devices—into editable and searchable formats [13]. Modern OCR systems follow a pipeline approach: first detecting text regions within an image, then recognizing and converting those regions into machine-readable characters. This facilitates intelligent interaction with visual data.

PaddleOCR is an open-source Optical Character Recognition (OCR) tool developed by PaddlePaddle, Baidu's deep learning platform [14]. It offers a high-performance, lightweight solution for multilingual text recognition, supporting over 80 languages. Designed to detect both horizontal and multi-directional text, PaddleOCR can independently identify text in images, making it ideal for real-world applications such as label reading, license plate recognition, and document scanning. Thanks to its modular architecture, PaddleOCR is easy to customize and can be seamlessly integrated with object detection frameworks like YOLO.

2.3 Existing Methods

Several academic projects have explored the integration of object detection and text recognition for various purposes—ranging from task automation and simplification of complex processes to enhancing safety and security measures. One notable example is the development of an Automatic Container Code Recognition (ACCR) system, designed to streamline the inspection and verification of container codes [15]. This system leverages the high accuracy of the Faster R-CNN architecture, a two-stage detector that combines region proposal mechanisms with convolutional neural networks. In this approach, each character in a valid container code—consisting of four capital letters followed by seven digits—is treated as a distinct object. The model is trained to detect 26 alphabetic characters (A–Z) and 10 numerical digits (0–9) as separate classes. Additionally, a binary search tree algorithm is employed to assemble and validate recognized characters efficiently.

The use of Faster R-CNN is crucial for achieving the precision required in this application. Through its Region Proposal Network (RPN), the model effectively learns the "objectness" of regions within an image and generates refined bounding boxes for character detection. The incorporation of anchors—rectangular windows of varying sizes and aspect ratios—enables robust detection of characters across multiple scales and positions. While the system achieved high accuracy, its two-stage detection approach is computationally more intensive compared to faster, one-stage frameworks like YOLO. Nonetheless, it demonstrates the effectiveness of combining region-based detection and character-level classification in achieving high-precision recognition of structured alphanumeric data, which is especially valuable in industrial settings like port logistics and container management.

Another similar study, introduced in [16], proposed a road safety surveillance tool designed to monitor traffic and enforce helmet laws. The system integrates the complementary capabilities of YOLO and OCR for real-time object detection and license plate recognition. It features a user-friendly PyQt graphical interface that allows bike owners to register their vehicles, which are then stored in a CSV file. A key component of the system is its use of YOLOv5 to analyze video frames in real time, detecting motorcycles, helmets, and license plates. In cases where a rider is not wearing a helmet, the model extracts the license plate text using OCR and cross-references it with the registered vehicle data to identify the offender. Once a match is confirmed, the system automatically generates an electronic violation report and issues a fine. The violators are easily contacted, and reports are sent directly to their Gmail addresses. As a result, the overall goal of enhancing road safety is significantly achieved.

3 Proposed Methodology

The proposed approach leverages the precise text recognition capabilities of PaddleOCR and the real-time object detection capabilities of the lightweight YOLOv5s. The hybrid architecture concurrently detects and extracts pertinent textual information, including ship names and visual IDs, from still photos or video feeds, forming the basis of an effective and domain-specific framework for seaport logistics.

This model's lightweight design makes it appropriate for deployment in resource-constrained

contexts. Its accuracy and processing speed balance enable real-time handling of high-throughput image streams. The system employs a two-stage pipeline, first identifying textual sections and then performing recognition, to improve accuracy and efficiency. This approach overcomes the drawbacks of traditional OCR frameworks, which are often designed for inputs with large amounts of text. In contrast, maritime images are often sparse and visually complex, with only identifying numbers or ship names included as text.

1. **Object-Level Text Localization:** A custom-trained YOLOv5s model detects object categories—*Ship*, *Text* and *Text*. Detected text regions are treated as discrete objects, allowing for precise spatial localization within complex scenes.

2. **Region-Based Text Recognition:** The localized regions are cropped, undergo a dedicated preprocessing pipeline, and are then passed to PaddleOCR for recognition.

This improves recognition accuracy by concentrating solely on pertinent image portions, while lowering computing cost in comparison to full-frame OCR. Localized OCR combined with object detection offers a scalable, effective way to improve situational awareness in automation workflows and seaport surveillance without sacrificing real-time performance.

3.1 System Design Architecture

The project started with a thorough strategy and system architecture design, where the main functionalities and overall interaction flow were carefully created. Particular focus was placed on streamlining the OCR extraction procedure to guarantee highly accurate and efficient text recognition. The design prioritized scalability, accuracy, and performance to facilitate seamless operation and flexibility. The flowchart is depicted in Figure 1 and reflects the methodical structuring of the workflow and component interactions.

3.2 Data Collection and Annotation

The dataset for this project was constructed using a combination of publicly available resources and manually curated images sourced from the internet. This approach aimed to ensure that the model could generalize effectively across diverse scenarios. One of the main challenges was the absence of any existing dataset specifically tailored to the project's requirements, while alternative commercial datasets were either inaccessible or prohibitively expensive. A total of 600 images were collected and divided into 500 for training, 60 for validation, and 60 for testing. Annotation was performed using Roboflow, with objects classified into Two categories: Ship and Text. To enhance variability and mitigate the risk of overfitting, Roboflow's augmentation tools were employed during preprocessing. Techniques such as rotation, binarization, and other transformations were applied to increase the dataset's diversity and improve the model's ability to generalize to unseen data.

3.3 Training YOLOv5s

By default, YOLOv5 is configured with `nc=80`, corresponding to 80 output classes in its prediction layer. While this configuration is appropriate for general-purpose object detection,

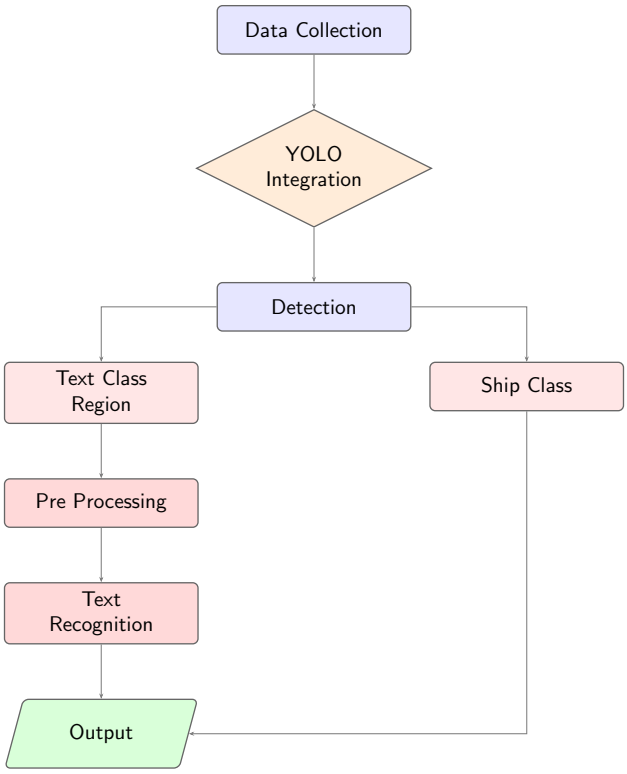

Figure 1: Methodology.

it introduces unnecessary computational overhead for the target application, where most of these classes are irrelevant. To optimize the model for the specific requirements of this work, the prediction layer was reconfigured with `nc=2`, focusing exclusively on two relevant classes: *Ship*, and *Text*. In addition, the smallest detection scale (stride 8, corresponding to a grid resolution of 80×80) was removed. This layer, typically designed for detecting small objects, is not critical for the types of targets in the maritime context. The model retains only the stride 16 (40×40) and stride 32 (20×20) layers, which are more effective for detecting medium to large objects such as vessels markings. These architectural adjustments significantly reduce computational complexity, minimize the risk of misclassification, and mitigate overfitting due to excessive model capacity. Consequently, the modified model achieves higher detection accuracy while maintaining real-time processing efficiency.

The training process of the customized YOLOv5s model follows its standard two-stage structure, comprising the *Backbone* and the *Head*. The curated dataset is first passed through the Backbone, which extracts multi-scale features from the input images, capturing both fine-grained and semantic information. These features are then forwarded to the Head, which generates detection outputs. In this implementation, the final prediction layer—originally designed for multi-class object detection—has been modified to support only the two target classes. This revised layer performs both object classification and bounding box regression, tailored to the maritime domain. As a result, the model outputs streamlined, application-specific detection results with enhanced relevance and computational efficiency.

3.4 YOLO Performance Evaluation

Its implementation and assessment results show the whole performance of the pipeline, including object identification, text recognition, and real-time operation. Even in a variety of dynamic scenarios, the YOLOv5s model's detection accuracy for the given objects Ship and Text is high. The successful recognition of the alphanumeric data in the clipped Text objects with the incorporation of OCR allows for the precise extraction of pertinent textual information.

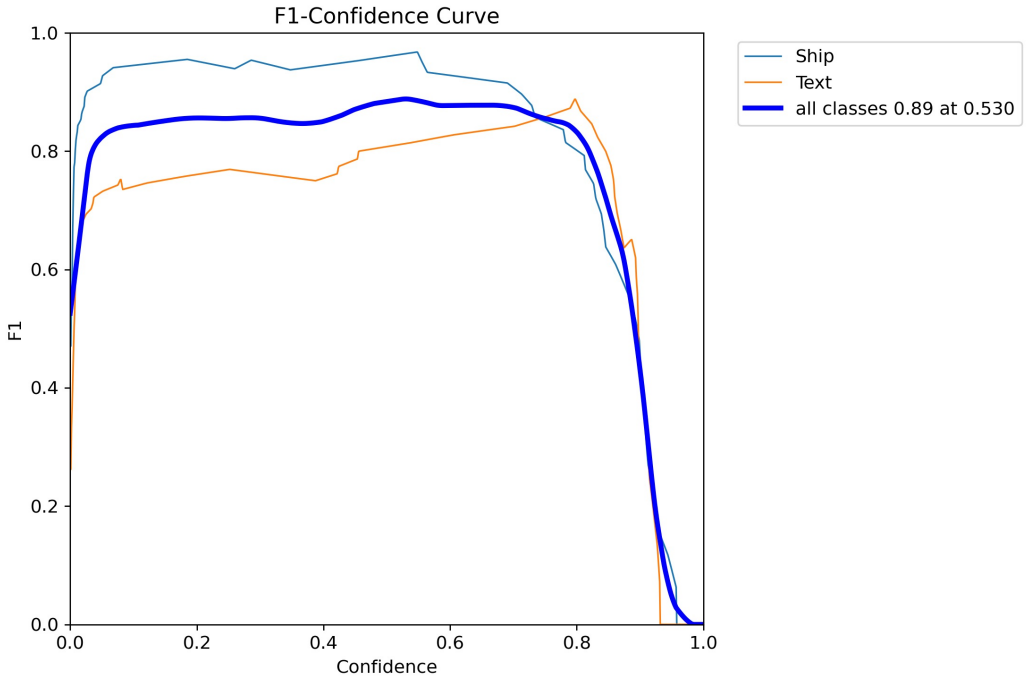

Figure 2: F1-score of the model across different confidence levels.

Fig.2 illustrates the accuracy of the object detection model across varying confidence thresholds. A high F1 score (blue line) indicates strong overall performance and balanced precision and recall.

Fig. 3 illustrates the Precision-Confidence (P) Curve, which visualizes the relationship between the model's precision and the confidence threshold used for object detection. Similar to the F1 curve, a consistently high blue line indicates strong overall detection accuracy across all object classes at specific confidence levels (peaks). However, the precision for individual object classes varies, reflecting differences in detection confidence and model certainty.

Fig. 4 presents the Recall-Confidence (R) Curve, an important metric that plots recall (true positive rate) against varying confidence thresholds. This curve provides insights into how the model's ability to identify relevant objects changes with different confidence levels. It highlights the trade-off between confidence and recall, where increasing the confidence threshold may reduce false positives but also lower recall. In the graph, confidence is represented on the X-axis and recall on the Y-axis.

Figure 5 depicts the training curves which is the model's performance as it learns from

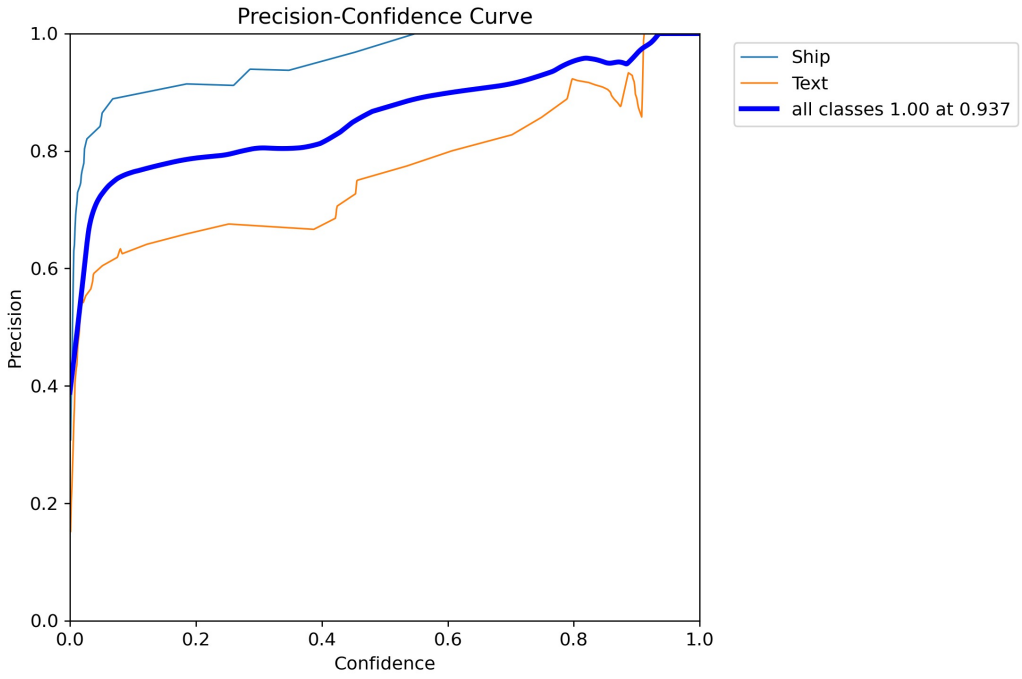

Figure 3: Precision Curve.

the training data. The horizontal axis (X-axis) represents the number of training iterations (epochs), while the vertical axis (Y-axis) indicates the loss values. A decreasing loss value reflects the model's improved ability to align its predictions with the actual ground truth data. Overall, the findings support the technological viability of the proposed OTDR system as a promising solution for digitizing key aspects of port operations, particularly in automating the detection and recognition of ships identification marks.

3.5 Text Recognition Using PaddleOCR

As illustrated in Figure 1, once the YOLOv5s model detects an object classified as *Text*, the corresponding region is cropped and passed into the recognition pathway powered by PaddleOCR. Prior to recognition, the cropped region undergoes a series of preprocessing operations integrated into the pipeline to enhance and standardize the text regions. These preprocessing steps include deskewing to correct orientation distortions, erosion to eliminate minor artifacts, noise reduction to enhance image clarity, and illumination normalization to mitigate lighting inconsistencies. The refined image is subsequently passed to the PaddleOCR engine, which performs accurate and efficient text extraction.

3.6 Deployment

The system is designed for flexible deployment, capable of processing various types of input including static images, live camera feeds, and video frames. The complete end-to-end pipeline is illustrated in Figure 1. Initially, the input is passed to the YOLOv5 model, which detects and classifies objects based on the predefined training classes. If the detected object

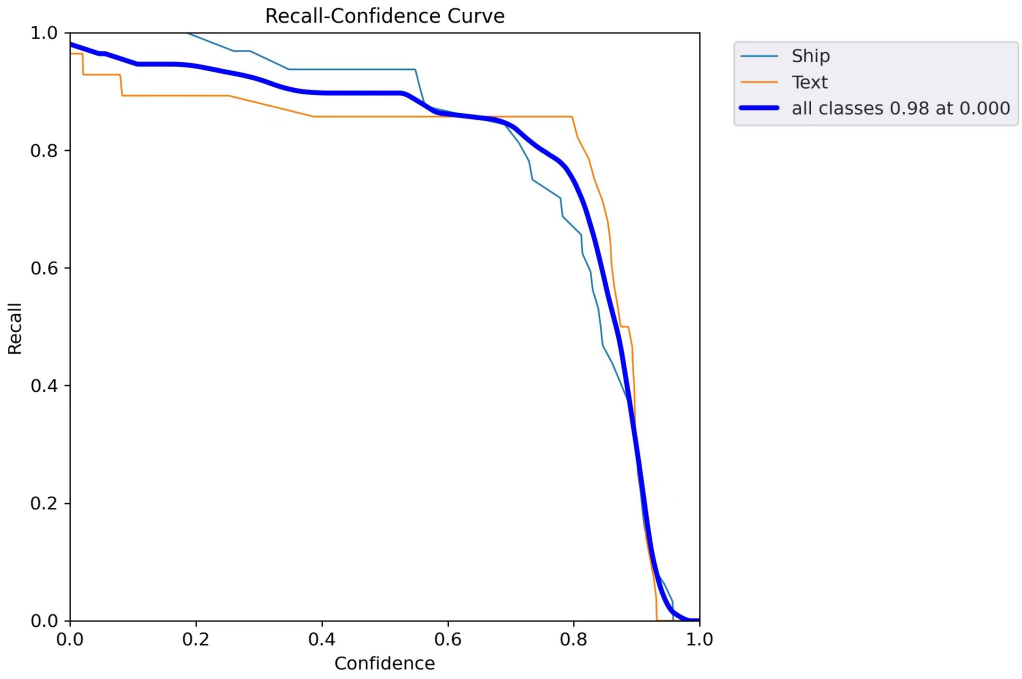

Figure 4: Recall Curve.

is identified as a ship, the result is sent directly to the output module for visualization. In cases where the detected object is classified as text, the corresponding region is routed to a preprocessing module, where enhancement techniques are applied to improve text clarity. The enhanced region is then forwarded to the PaddleOCR engine, which extracts the textual content and sends the recognized text to the output module for display. This modular architecture ensures efficient handling of both object detection and text recognition tasks.

4 CONCLUSION

This study presents a lightweight and efficient object-text detection framework specifically designed for seaport logistics and maritime surveillance applications. The proposed system integrates a modified YOLOv5s-based object detector with a PaddleOCR-powered text recognition pipeline to enable precise localization and interpretation of textual and non-textual entities under complex environmental conditions. To achieve reliable results, architectural optimizations were applied to the YOLO model, including the reduction of object classes and the elimination of the smallest detection scale. These adjustments significantly lower computational power demands while maintaining a high level of accuracy. Such refinements are especially advantageous for edge deployments and real-time applications in bandwidth-constrained or resource limited environments commonly found in seaport logistics. To further enhance recognition accuracy, especially in low-light and degraded image conditions, the pipeline incorporates domain specific preprocessing techniques prior to OCR. This results in improved text clarity and robustness in diverse imaging scenarios, thereby

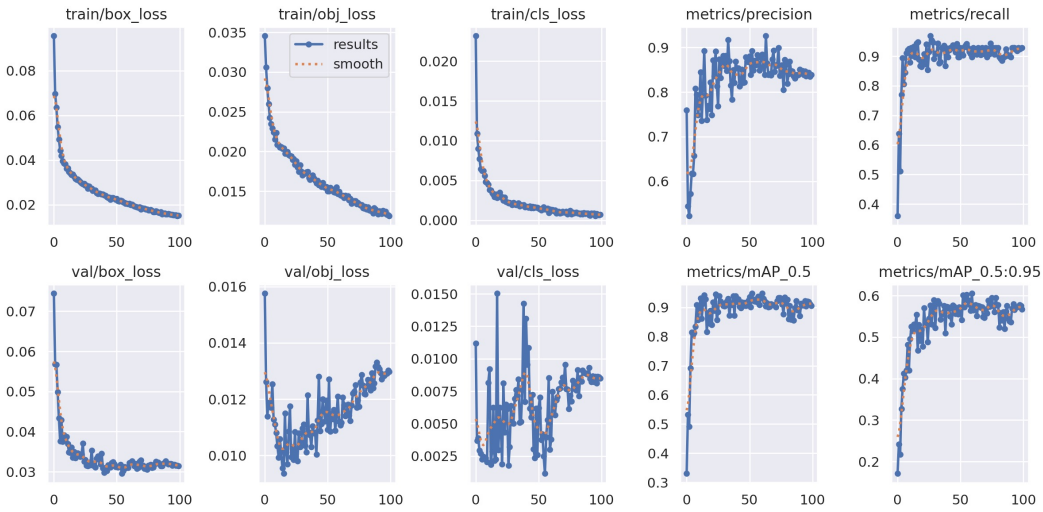

Figure 5: Final Result Graph.

increasing the reliability of automated visual inspection tasks such as ship name and identification number. The experimental results demonstrate that the system achieves a strong balance between speed and precision, making it well-suited for practical deployment in dynamic maritime contexts. Moreover, the modularity of the proposed architecture allows for easy adaptation and integration with other monitoring and data processing systems within the logistics chain. In future work will aim to explore the integration of object and text tracking mechanisms to enhance temporal consistency and reduce recognition latency across video frames. Additionally, the extension of this framework to incorporate multimodal sensor data—such as LiDAR, radar, or AIS (Automatic Identification System) to further improve situational awareness and operational safety in seaport environments. Expanding the system's adaptability to multilingual text and variable weather conditions will also be considered to broaden its global applicability.

Acknowledgments

The authors would like to express their sincere gratitude to Zhejiang University of Science and Technology for providing the research environment and resources that supported this work. Special thanks are extended to the project supervisors and colleagues for their valuable guidance and constructive feedback during the course of this study. The authors also acknowledge the support from North University of China and CHN Energy Group Huanghua Port Co. Ltd for their collaboration and contributions.

Competing Interests

The authors declare that they have no known financial or personal relationships that could inappropriately influence (bias) this work. Authors have declared that no competing interests exist.

AUTHORS CONTRIBUTIONS

Aminu Yahaya designed the study, developed the YOLO–OCR framework, conducted the experiments, and wrote the first draft of the manuscript. Rui-Cai Jia contributed to methodology development, data analysis, and manuscript revision. Xingli Gan performed data preparation, validation, and assisted with result interpretation. Chong Shen managed the literature review and contributed to the technical analysis of the study. De-lin Zhao supervised the project, provided industrial insights on seaport applications, and reviewed the manuscript.

All authors read and approved the final manuscript.

References

[1] T. H. Nguyen and S. J. Kim, "Artificial intelligence-based smart port: A review of key technologies and applications," *IEEE Access*, vol. 8, pp. 132908–132924, 2020.

[2] Z. Pan, L. Zhang, and L. Li, "Application of deep learning in intelligent logistics: A review," *IEEE Access*, vol. 9, pp. 33807–33824, 2021.

[3] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, 2015.

[7] M. A. Nazir, S. M. Sharif, and M. Y. Javed, "YOLO object detection: A review of versions from YOLOv1 to YOLOv5," *ACM Comput. Surv.*, vol. 55, no. 1, pp. 1–35, 2021.

[8] 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," in *arXiv preprint arXiv:2207.02696*, 2022. [Online]. Available: `https://arxiv.org/abs/2207.02696`

[9] M. Yaseen, "What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector," in *arXiv preprint arXiv:2408.15857*, 2024. [Online]. Available: `https://arxiv.org/abs/2408.15857`

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.

[12] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR)*, vol. 2, 2007, pp. 629–633.

[13] D. Peng, Z. Yang, J. Zhang, C. Liu, Y. Shi, K. Ding, F. Guo, and L. Jin, "UPOCR: Towards unified pixel-level OCR interface," arXiv preprint arXiv:2312.02694, 2023.

[14] PaddleOCR, "PaddleOCR: An open-source toolkit for optical character recognition," *GitHub Repository*, 2020. [Online]. Available: `https://github.com/PaddlePaddle/PaddleOCR`

[15] N. Chen, X. Ding, and H. Zhang, "Improved Faster R-CNN identification method for containers," *Int. J. Embedded Syst.*, no. May, pp. 308–317, 2020.

[16] H. Krishnan R, I. J., and V. S. Devi, "Enhancing road safety with real-time helmet detection and e-challan issuance using YOLO and OCR," *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 6, no. 5, pp. 1–8, May 2024.