IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 June 2024, accepted 29 July 2024, date of publication 5 August 2024, date of current version 13 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438792

APPLIED RESEARCH

Deep Learning-Powered Ship IMO Number Identification on UAV Imagery

ZHI-BO CAO
Unmanned System Research Center, China People's Police University, Guangzhou 510006, China
e-mail: caozhibo@126.com

This work was supported by the Research and Innovation Project of China People's Police University's Ph.D. Research and Innovation Program under Grant BSKY202304.

ABSTRACT The paper presents a novel framework for ship International Maritime Organization (IMO) number Identification using unmanned aerial vehicles (UAVs). It comprises three integrated modules: ship IMO region detection, text detection within detected IMO regions, and IMO number extraction from detected text. Furthermore, considering real-world implementation, the paper details the framework's practical deployment on UAV and proposes an algorithm for efficiently extracting IMO numbers from the detected text. To ensure robust performance evaluation, a comprehensive evaluation metric is established for screening various ship IMO region detection, text detection, and recognition algorithms. Through extensive experimentation, YOLOx_S, DB_R18, and SVTR were identified as optimal for ship IMO region detection, text detection, and text recognition respectively. Finally, we acknowledge the presence of potential false detections in the results and emphasize that while the comprehensive evaluation metric offers valuable insights, it should not be the sole criterion for algorithm selection.

INDEX TERMS Ship detection, IMO number identification, UAV-based recognition, deep learning framework.

I. INTRODUCTION

As integral components of international trade and maritime transport, ships play a pivotal role globally. However, accurately detecting and identifying the identity information of ships is crucial for maritime safety, ocean management, and port operations. The technology for UAV-based detection and recognition of ship IMO numbers is of paramount importance, necessitating research in this field. Moreover, the application of UAV-based ship IMO number detection and recognition holds significant importance in the maritime domain.

a) Enhancing Maritime Security: The ship's IMO number serves as a unique identifier, containing specific information about the ship, such as the ship owner, type, and dimensions. Utilizing UAV-based detection and recognition technology for ship IMO numbers allows for the remote acquisition of this information. By cross-referencing the obtained IMO numbers with relevant databases, the legitimacy and safety of the ship can be ensured. This technology aids in reducing the presence of illegal ships, thereby elevating the level of maritime security and preventing potential threats and unlawful activities.

b) Promoting Ocean Management and Regulation: In terms of ocean management and regulation, accurately detecting and identifying ship IMO numbers is vital for supervising maritime traffic, managing port operations, and enforcing marine environmental protection policies. UAVs, equipped with aerial surveillance and intelligent image processing technologies, can monitor and identify ships across extensive sea areas, providing marine managers with real-time ship information and data support. This capability enhances the management and utilization efficiency of marine resources.

c) Increasing Port Operation Efficiency: UAV-based detection and recognition of ship IMO numbers play a significant role in port operations. Traditionally, ports rely on manual inspections and record-keeping, which are inefficient and prone to errors. By employing UAV technology, automatic detection and identification of ships within the port area can be achieved, improving operational efficiency and accuracy.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This reduces the time ships spend entering and leaving ports and optimizes the utilization of port resources.

d) Driving Technological Innovation: UAV-based detection and recognition of ship IMO numbers is a relatively emerging field of research that encompasses multiple disciplines, including UAV technology, ship recognition algorithms, and image processing. Research in this area fosters technological innovation and method improvements, such as advancements in UAV image acquisition technology, optimization of target detection and recognition algorithms, and the construction of ship information databases. These innovations and improvements will support the development of the maritime field and the intelligent advancement of ship transportation.

Although UAV-based detection and recognition of ship IMO numbers offer numerous potential advantages, several challenges and limitations must be addressed:

a) Image Quality and Angle: Ship IMO numbers are typically located on the side or stern of the ship. When UAVs capture images from various angles and altitudes, factors such as lighting conditions, ship orientation, and distance can affect image quality, making the IMO number difficult to recognize clearly.

b) Complex Background and Occlusion: During aerial photography, UAVs may encounter complex background environments, such as waves, buildings, and other ships. These background elements can interfere with the recognition of the ship's IMO number, rendering it blurred or unclear. Additionally, the ship may be partially obscured by other structures or equipment, further complicating the identification process.

c) Algorithm Accuracy and Real-Time Processing: Ship IMO number detection and recognition algorithms need to be highly accurate and capable of real-time processing to promptly identify the IMO number and provide accurate results. The performance of these algorithms depends on the quality of image processing techniques, the training and updating of machine learning models, and other factors, necessitating continuous improvement and optimization.

To address the aforementioned challenges and issues, we have developed a UAV-based framework for the detection and recognition of ship IMO numbers, along with an implementation process for this framework on UAVs. Additionally, we have proposed an algorithm for extracting IMO numbers based on the text in the ship IMO region. We collected a dataset containing images of ships and their IMO regions from MarineTraffic. Furthermore, we introduced a comprehensive evaluation metric for selecting the most effective algorithms for ship IMO number detection and recognition. Through detailed discussions of the experimental results, we fine-tuned the selection process and identified the algorithms best suited for UAV-based ship IMO number detection and recognition.

II. RELATED WORK

The related work on the UAV-based framework for detecting and recognizing ship IMO numbers primarily involves three technical domains: the application of artificial intelligence in UAVs, research on deep learning-based object detection algorithms, and research on deep learning-based text detection and recognition algorithms. Below, we introduce relevant research work in these three areas:

A. APPLICATION OF ARTIFICIAL INTELLIGENCE IN UAVS

As artificial intelligence technology rapidly evolves, its applications in the UAV field are advancing swiftly, showcasing immense potential. Currently, the application of AI technology in UAVs is primarily concentrated in the following two areas:

1) AUTONOMOUS NAVIGATION AND FLIGHT CONTROL

Artificial intelligence can be utilized to develop autonomous navigation and flight control systems for UAVs. By employing machine learning and deep learning algorithms, UAVs can learn tasks such as environmental perception, path planning, and obstacle avoidance, enabling them to achieve autonomous flight and navigation capabilities.

Wang et al. [1] proposed a deep reinforcement learning method that allows UAVs to perform navigation tasks in complex environments. This method enhances UAV's ability to navigate autonomously by learning from interactions with the environment. Tullu et al. [2] introduced a real-time 3D path planner based on deep learning, which enables UAVs to navigate through obstacle-free paths to reach their destinations. Lee et al. [3] addressed the limitations of existing active obstacle avoidance systems for small aircraft. They proposed an autonomous navigation method for small UAVs in plantations, demonstrating how AI can be tailored to specific operational environments. He et al. [4] developed a path planner based on an interpretable deep neural network for quadcopters to navigate autonomously in unknown environments. Zhang et al. [5] introduced a deep reinforcement learning (DRL) method that enables UAVs to navigate in environments with random and dynamic obstacles. This method allows UAVs to adapt to changing conditions and avoid collisions.

Dooraki and Lee [6] described the use of a deep reinforcement learning self-training controller for autonomous navigation in both static and dynamic environments. The algorithm learns to generate continuous actions to control the UAV effectively. Obaid et al. [7] proposed a novel training data selection method that accelerates training convergence, significantly enhancing the UAV's obstacle avoidance capabilities. Badrloo et al. [8] conducted a comprehensive survey on various image-based obstacle detection techniques. The survey reviewed over 110 papers published in 23 high-impact computer science journals over the past 20 years. It categorized techniques into monocular (single-camera) and binocular obstacle detection methods. Monocular methods are more suitable for small robots such as Micro Aerial Vehicles (MAVs) and compact UAVs, which often have limited processing power. The study also highlighted that until recently, both monocular and binocular methods relied on traditional image processing techniques, which are inadequate for real-time applications. Consequently, deep learning networks have become the focus for developing fast and reliable obstacle detection solutions. The research identified that detecting narrow, small, and moving obstacles, as well as achieving rapid obstacle detection, remains a significant challenge that future studies need to address.

2) OBJECT DETECTION AND TRACKING

Artificial intelligence can significantly enhance the capabilities of UAVs in object detection and tracking. Utilizing computer vision and deep learning technologies, UAVs can identify and track ground targets, vehicles, pedestrians, and more, enabling applications in surveillance, search and rescue, and beyond.

Nousi et al. [9] demonstrated that a sophisticated neural network-based detection and tracking system could achieve real-time operation on embedded hardware. Hong et al. [10] addressed the challenge of target detection in wildlife monitoring. By developing deep-learning models and utilizing aerial images from UAVs, they successfully detected and identified bird species across various environments. This method provides high accuracy and speed, offering an effective solution for bird monitoring and disease spread prediction.

Wang et al. [11] showcased the use of SiamMask, a simple yet effective approach for visual object tracking and semi-supervised video object segmentation. This method demonstrated real-time performance and achieved state-of-the-art results on general datasets. Zhang et al. [12] proposed a coarse-to-fine object detection method for UAV imagery, utilizing a lightweight convolutional neural network (CNN) and deep motion saliency to enhance detection accuracy and efficiency. Rabah et al. [13] introduced a strategy to integrate commonly used object detection and tracking algorithms with UAV control software, designed to run on heterogeneous resource-limited computing units on UAVs.

Isaac-Medina et al. [14] explored the benchmarking of UAV target detection and tracking by evaluating four detection architectures and three tracking frameworks across three datasets, providing insights into performance under various conditions. Saetchnikov et al. [15] investigated the challenges of CNNs in detecting objects from UAV aerial imagery, addressing issues such as limited camera perspective and spatial resolution, as well as insufficient training data for certain object classes. The study examined the effectiveness of different deep neural networks on target detection with limited pre-trained datasets. Wei et al. [16] utilized UAVs (DJI Phantom4RTK) and the YOLOv4 (You Only Look Once) object detection deep neural network to collect images of mature rice crops and detect rice ears, generating density prescription maps for the crops.

The integration of UAVs and AI offers numerous other advantages, such as more efficient data collection and sensing capabilities, precise mapping and 3D modeling, and quick response and flexibility. This synergy creates opportunities for innovation and improvement across various industries and applications.

B. DEEP LEARNING-BASED OBJECT DETECTION ALGORITHMS

Deep learning has achieved remarkable breakthroughs in the field of object detection, becoming one of the most influential technologies in computer vision. The objective of object detection is to accurately identify and locate specific objects within images or videos. Based on whether there is a pre-selection box extraction network involved in the detection process, deep learning-based object detection algorithms can be divided into two-stage and one-stage algorithms.

1) TWO-STAGE DETECTION ALGORITHMS

Two-stage deep learning algorithms consist of two learning phases. The first phase uses selective search or region proposal networks (RPNs) to extract candidate regions (pre-selection boxes). The second phase classifies these regions to obtain the final object detection results.

R-CNN: Initially, CNNs were primarily used for binary image classification tasks [17]. However, they could not handle multi-object detection tasks. To address this, Girshick et al. [18] proposed the Region-based Convolutional Neural Network (R-CNN), which enabled the detection and classification of multiple objects in an image by using candidate regions.

Fast R-CNN: Girshick et al. [20] improved upon R-CNN by introducing Fast R-CNN, utilizing the VGG-16 network [19] to accelerate the training process.

Faster R-CNN: To simplify the complex training process of R-CNN and Fast R-CNN, Ren et al. [21] proposed Faster R-CNN, which shared the feature extraction network with the RPN, thereby avoiding the stacking of multiple independent components and achieving faster training speed and higher recognition accuracy.

Feature Pyramid Network (FPN): Lin et al. [22] developed a top-down architecture known as the Feature Pyramid Network (FPN), which created high-level semantic feature maps at all scales through lateral connections. This architecture showed significant improvements in various applications as a general feature extractor.

Cascade R-CNN: To address noisy detection results at low Intersection over Union (IoU) thresholds and the decline in detection performance as IoU thresholds increased, Cai et al. [23] proposed Cascade R-CNN. This multi-stage object detection architecture used progressively larger IoU thresholds at each stage to enhance the filtering of false positives.

Non-Local Networks: Wang et al. [24] introduced a building block called ‘‘non-local operations’’ as a general method to capture long-range dependencies. This approach achieved better performance in object detection on general datasets.

Libra R-CNN: Pang et al. [25] addressed the limitations of detection performance due to imbalance during the training process. They proposed Libra R-CNN, which included three components: IoU-balanced sampling, balanced feature pyramids, and balanced L1 loss. These components respectively addressed imbalances at the sample, feature, and objective levels. Experimental results showed that Libra R-CNN significantly improved object detection performance.

These advancements in two-stage detection algorithms have paved the way for more accurate and efficient object detection, contributing to the broader field of computer vision and its applications.

2) ONE-STAGE DETECTION ALGORITHMS

Unlike two-stage detection algorithms, one-stage algorithms directly predict the positions and categories of objects from the image, offering a more streamlined design and higher real-time performance.

YOLO: Introduced by Redmon et al. [26], YOLO treats the object detection problem as a regression problem. A single neural network predicts bounding boxes and class probabilities directly from the entire image, allowing for end-to-end optimization of detection performance. The unified network structure simplifies the detection pipeline and improves real-time performance.

SSD (Single Shot MultiBox Detector): Proposed by Liu et al. [27], SSD utilizes a single deep neural network to detect objects in images. It discretizes the output space of bounding boxes into a set of default boxes with different aspect ratios and scales for each feature map location. SSD eliminates the need for proposal generation and subsequent pixel or feature resampling stages, making it easier to train and integrate into systems requiring detection components.

YOLO9000 (YOLOv2): Advanced by Redmon et al. [28], YOLO9000 is capable of real-time detection of over 9000 different object categories. It offers high detection accuracy and speed, with wide applications in image recognition, video surveillance, and autonomous driving.

RetinaNet: Developed by Lin et al. [29], RetinaNet addresses the foreground-background class imbalance problem in dense detectors by introducing Focal Loss. This allows RetinaNet to achieve accuracy surpassing all existing state-of-the-art two-stage detectors while maintaining comparable speed to one-stage detectors.

CornerNet: Introduced by Law et al. [30], CornerNet represents object bounding box detection as a pair of keypoints (the top-left and bottom-right corners) using a single convolutional neural network. The concept of corner pooling is introduced to help the network better locate corner points.

YOLOv3: Redmon et al. [31] modified the previous YOLOv2 by introducing an updated training network, resulting in faster recognition speeds and higher accuracy.

CenterNet: Proposed by Zhou et al. [32], CenterNet models objects as the center points of their bounding boxes. It finds these center points through keypoint estimation and regresses other object attributes like size, 3D position, orientation, and pose.

YOLOv4: Proposed by Bochkovskiy et al. [33], an optimized framework that improved recognition accuracy without compromising detection speed compared to YOLOv3.

Subsequent research has yielded numerous enhancements to YOLO, such as YOLOX [34], YOLOv6 [35], and YOLOv7 [36], focusing primarily on improving prediction accuracy. The evolution of these one-stage algorithms continues to push the boundaries of real-time object detection, making them invaluable in various applications requiring high-speed and accurate object recognition.

C. DEEP LEARNING-BASED TEXT DETECTION AND RECOGNITION ALGORITHMS

Text detection and recognition hold significant practical importance. Extracting text information from images can enable automated processing and analysis, which is invaluable for handling large-scale image data, such as in digitizing archives, library documents, invoices, and forms. Text in images can serve as keywords or metadata for indexing and retrieval, benefiting applications like image libraries, social media platforms, and search engines. Converting text in images to readable characters aids visually impaired individuals in accessing image information. Additionally, text detection and recognition are crucial in various fields that require reading text from images. Deep learning has become a focal point in this research area. Below, we highlight the major research work in the past five years.

1) DEEP LEARNING-BASED TEXT DETECTION ALGORITHMS

TextSnake [37]: Long et al. proposed a text detection framework called TextSnake, which addresses the limitations of existing text detection methods in handling irregular texts. TextSnake flexibly represents scene text, effectively managing free-form text instances, including curved text. It achieved impressive performance in text detection tasks, surpassing existing methods on relevant benchmark datasets.

PSENet [38]: Wang et al. introduced the Progressive Scale Expansion Network (PSENet), a novel approach that can accurately detect text instances of any shape. PSENet solves the problems of low localization accuracy and erroneous detection due to closely spaced text instances by progressively expanding scales. Experiments demonstrated PSENet's effectiveness and superiority on multiple benchmark datasets.

PAN [39]: Wang et al. proposed the Pixel Aggregation Network (PAN), an efficient and accurate arbitrary-shaped text detector. By employing a low computational cost segmentation head and learnable post-processing, PAN achieves precise text detection while maintaining speed. Experiments on several standard benchmark datasets validated PAN's superiority and performance.

DBNet [40]: Liao et al. introduced the Differentiable Binarization (DB) module, which performs binarization in the segmentation network, simplifying the post-processing and enhancing text detection performance. Combined with a lightweight backbone network, DBNet achieved state-of-the-art results on multiple benchmark datasets.

DRRG [41]: Zhang et al. proposed the Deep Relational Reasoning Graph Network (DRRG) for arbitrary-shaped text detection. Utilizing an innovative local graph model and graph convolutional network, DRRG allows end-to-end training and demonstrates state-of-the-art performance on public datasets.

FCENet [42]: Zhu et al. proposed the Fourier Contour Embedding (FCE) method in FCENet, which effectively represents and detects text of any shape. Experiments showed that FCENet is accurate and robust, capable of handling highly curved scene text contours. FCENet outperformed state-of-the-art methods in arbitrary-shaped text detection.

DBNet++ [43]: Liao et al. enhanced their previous DBNet framework by introducing an efficient Adaptive Scale Fusion (ASF) module in DBNet++. The ASF module adaptively merges features of different scales to improve scale robustness. Combining the DB and ASF modules with a segmentation network, DBNet++ improved the post-processing and scale robustness of segmentation-based scene text detection methods. It achieved state-of-the-art results in detection accuracy and speed across multiple benchmark datasets.

2) DEEP LEARNING-BASED TEXT RECOGNITION ALGORITHMS

Text detection and recognition are of immense practical importance due to their applications in automated processing, analysis, image indexing, and accessibility for visually impaired individuals. Extracting textual information from images helps in various domains, such as digitizing archives, social media, and search engines. The recent advancements in deep learning have significantly contributed to the field of text recognition. Below, we discuss the major research works in deep learning-based text recognition over the past few years.

ASTER [44]: Shi et al. addressed the challenges of recognizing perspective and curved text in natural scenes by proposing ASTER. This method includes a rectification network and a recognition network that adaptively corrects text in the input images and directly predicts character sequences from the rectified images. The experiments demonstrated ASTER's advanced rectification and recognition performance, showing its capability to enhance detectors in end-to-end recognition systems.

NRTR [45]: To address the slow training speed and high complexity of existing recognition methods, Sheng et al. introduced NRTR, a Non-Recurrent Sequence-to-Sequence Text Recognizer. NRTR eliminates recurrence and convolution, using self-attention mechanisms to extract image features and recognize text. NRTR achieved state-of-the-art or highly competitive performance on both regular and irregular benchmarks and significantly reduced the required training time (at least 8 times faster than the best models in the literature).

SAR [46]: Li et al. proposed a simple yet powerful baseline method for irregular scene text recognition, utilizing off-the-shelf neural network components and only word-level annotations. This method achieved state-of-the-art performance on irregular text recognition benchmarks and comparable results on regular text datasets.

SATRN [47]: Lee et al. introduced SATRN, an architecture for recognizing arbitrarily shaped scene text. SATRN uses self-attention mechanisms to describe the 2D spatial dependencies of characters in text images and achieves text recognition with arbitrary arrangements and large character spacings through global context propagation. SATRN outperformed all existing models by an average of 4.5 percentage points on irregular text benchmarks and achieved state-of-the-art performance on two regular text benchmarks.

RobustScanner [48]: Yue et al. proposed RobustScanner for scene text recognition by focusing on the decoding process. It uses a position-enhanced branch and a dynamic fusion mechanism to decode individual characters with a dynamic ratio between context and positional cues, ensuring robustness in both context-rich and context-scarce application scenarios. Experiments showed it achieved state-of-the-art results on common regular and irregular text recognition benchmarks and maintained performance in no-context benchmarks, demonstrating robustness across different application scenarios.

ABINet [49]: Fang et al. introduced ABINet, an autonomous, bidirectional, and iterative method for scene text recognition. ABINet effectively models language rules through gradient blocking, bidirectional feature representation, and iterative correction with a language model, showing superior performance on low-quality images and achieving state-of-the-art results on several mainstream benchmarks.

MASTER [50]: Lu et al. proposed MASTER, a self-attention-based scene text recognizer. MASTER excels in handling spatial distortion issues by learning input-output attention and self-attention within encoders and decoders, leading to more robust intermediate representations. MASTER's high training parallelism and efficient memory caching mechanism offer high training efficiency and inference speed. Extensive experiments across various benchmarks demonstrated MASTER's excellent performance on both regular and irregular scene texts.

SVTR [51]: Du et al. proposed SVTR, a scene text recognition method adopting an image tokenization-based framework that eliminates sequence modeling. SVTR performs hierarchical staged operations through component-level mixing, merging, or combination to perceive character patterns both globally and locally, achieving multi-granularity character component perception. Character recognition is achieved through simple linear prediction. Experiments on English and Chinese scene text recognition tasks validated SVTR's effectiveness.

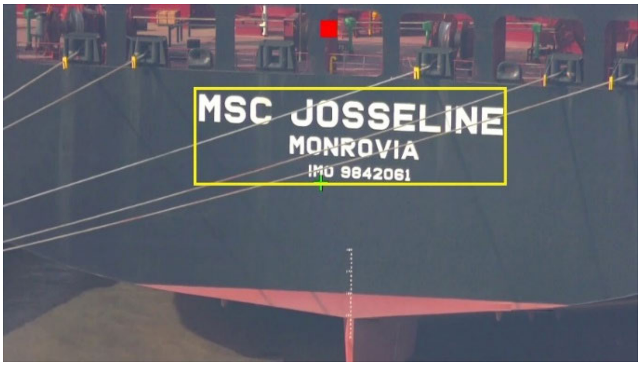

FIGURE 1. IMO region of the ship(Yellow Box).

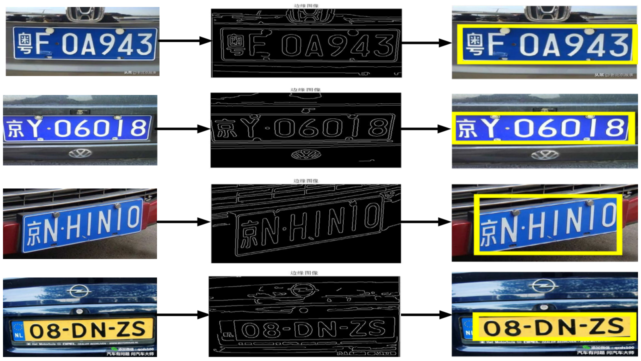

FIGURE 2. License plate region detection using classical image processing methods.

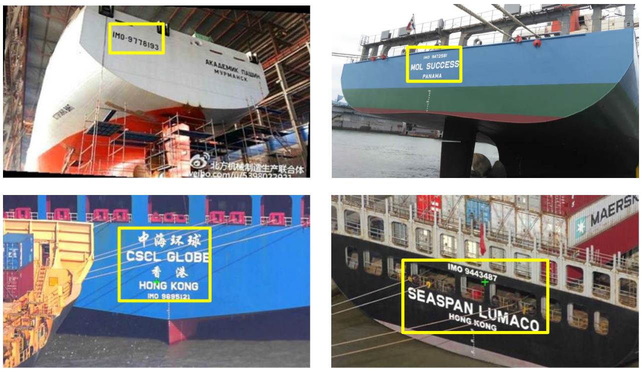

FIGURE 3. Positions of IMO numbers within the ship IMO region.

These advancements showcase the significant progress made in the field of text recognition through deep learning, highlighting the capability to handle both regular and irregular text patterns in various real-world applications.

III. SHIP IMO DETECTION AND RECOGNITION FRAMEWORK

A. PROBLEM ANALYSIS

For subsequent discussion, it is essential to clearly define the scope of the ship IMO region. As per our research requirements, the ship IMO region primarily includes the ship IMO number and other identifying information (Fig. 1). The detection and recognition of a ship's IMO number are similar to but not entirely the same as license plate detection and recognition.

a) Firstly, the ship IMO region differs from the license plate region. The license plate region has fixed contour features, which can be detected using image processing techniques such as Gaussian filtering and edge detection [52] (Fig. 2). In contrast, the background of the ship IMO region lacks distinct contour features and is generally a solid color, making it difficult to detect using the aforementioned methods.

b) Unlike the relatively fixed position of a license plate number, the position of the IMO number within the ship IMO region is not fixed (Fig. 3). Therefore, even if the textual information within the ship IMO region is identified, further efforts are required to distinguish the IMO number from the other text.

Therefore, our solution strategy is to address the detection of the ship IMO region using deep learning-based object detection algorithms and to solve the recognition of the ship IMO number using deep learning-based text detection and recognition algorithms. Additionally, the application scenarios for detecting and recognizing ship IMO numbers are primarily focused on seaports and anchorages. Especially in anchorages, detecting and recognizing ship IMO numbers necessitates the use of UAVs or other unmanned equipment. UAVs must first detect the ship to then identify the ship's IMO region, subsequently recognize the ship's IMO number, and ultimately achieve the surveillance and monitoring of the specified ship. Based on the above analysis, the main problems that our framework needs to address can be summarized as follows:

a) Object detection for ship and ship IMO region.

b) Text detection within the ship IMO region.

c) Text recognition within the ship IMO region.

B. THE FRAMEWORK

In response to the aforementioned issues, we have devised a deep learning-based framework for the detection and recognition of ship IMO numbers (Fig. 4). Initially, the ship detection network examines the input ship image, outlining the ship within the picture and cropping the selected area to obtain an image containing only the ship. Subsequently, the ship IMO region detection network scrutinizes the cropped image, identifying the ship IMO area within it. The image is then cropped according to the selected region, with the cropping area being slightly larger than the ship IMO region to facilitate subsequent processing. Next, the text detection and recognition network is employed to recognize the text within the ship IMO region of the image. Finally, key information is extracted from the recognized text content to obtain the ship's IMO number.

How can UAVs utilize the aforementioned framework for the recognition of ship IMO numbers? To address this, we have devised the execution process of the aforementioned framework on UAVs (Fig. 5).

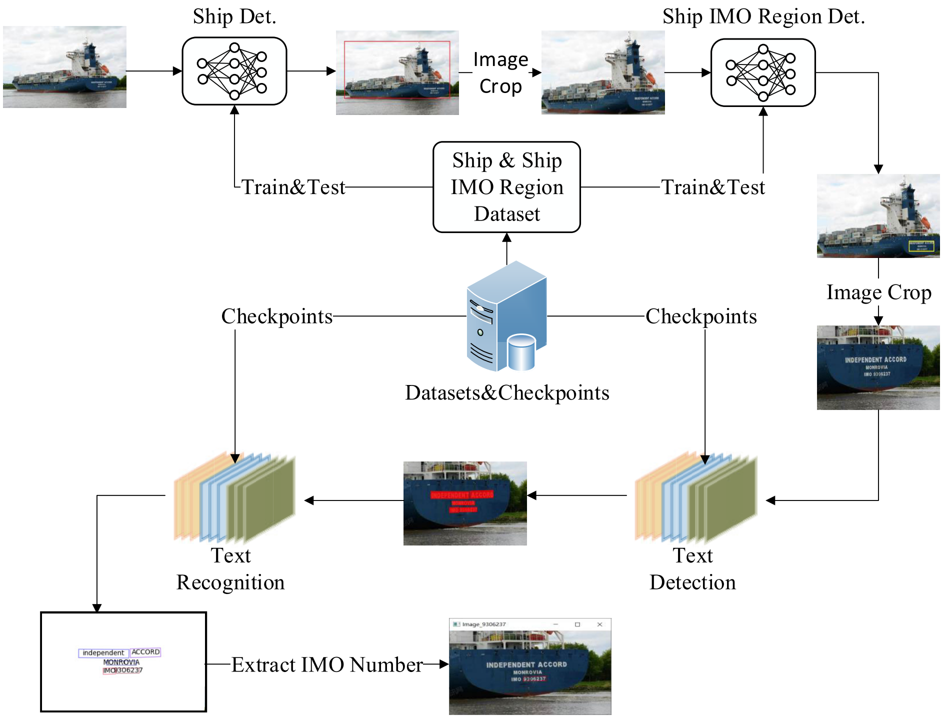

FIGURE 4. Framework for detecting and recognizing ship IMO numbers.

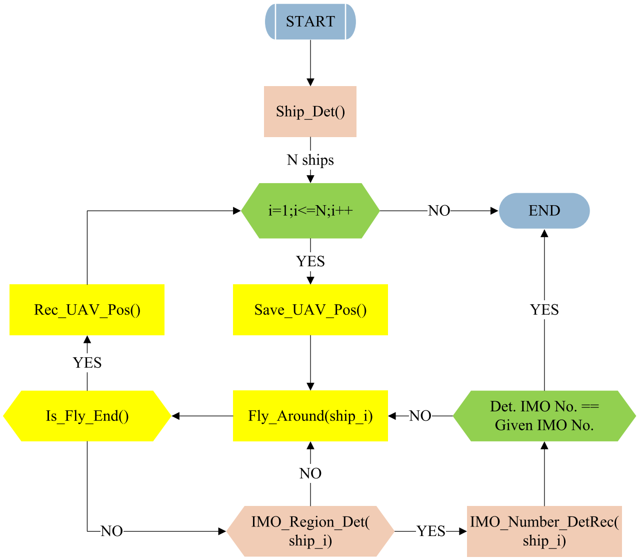

FIGURE 5. The execution process of the ship IMO number detection and recognition framework on UAV.

The execution process of the ship IMO number detection and recognition framework on UAVs is illustrated in Fig. 5. The main functionalities of the modules are as follows:

a)Ship_Det(): Used by UAV to detect the current location of ships and return the number of ships, denoted as N.

b)Save_UAV_Pos(): Saves the current position information of the UAV, including coordinates and altitude.

c)Fly_Around(ship_i): Circumnavigates around the i-th ship (ship_i).

d)Is_Fly_End(): Determines whether the current circumnavigation has ended.

e)Rec_UAV_Pos(): Returns to the position before the UAV's circumnavigation.

f)IMO_Region_Det(ship_i): Determines whether the image captured by the UAV contains information about the ship IMO region.

g)IMO_Number_DetRec(ship_i): Detects and recognizes the IMO number within the ship IMO region.

Initially, the control center of the harbor sends the position information of the controlled ships and their IMO numbers to the UAV control system. The UAV control system selects an appropriate UAV based on its availability to fly to the position of the controlled ship. Then, utilizing the object detection algorithm, the UAV detects ships within the specified area, using the Ship_Det() module. Assuming N ships are detected, the UAV saves the current position information using Save_UAV_Pos(). Subsequently, the UAV circumnavigates around the first ship using Fly_Around(ship_i) and checks if the circumnavigation has ended using Is_Fly_End(). If not, the UAV determines whether the captured photo contains information about the ship IMO region using IMO_Region_Det(ship_i). If it does not, the UAV continues circumnavigating; otherwise, IMO_Number_DetRec(ship_i) is employed to extract the IMO number within the ship IMO region. The extracted IMO number is then compared with the IMO number provided by the control center. If they match, the program ends; if not, the UAV continues circumnavigating. This process repeats for each ship until the IMO number is found or all N ships have been circumnavigated.

C. IMPLEMENTATION OF THE FRAMEWORK

The successful implementation of the UAV-based ship IMO number detection and recognition process hinges on the performance of the selected embedded graphics card. Currently, the mainstream embedded graphics cards include Nvidia Jetson TX1, Jetson TX2, Jetson Xavier NX, and Jetson AGX Xavier. To ensure the chosen model offers better generalizability, we selected an NVIDIA Quadro M1000M from our laboratory, which boasts performance approximately midway among these graphics cards, to test the detection speed of all models.

As shown in Table 1 (data sourced from Nvidia, the M1000M has 512 CUDA cores, surpassing the 256 cores of both TX1 and TX2. Its FP32 computing power (used for storing deep learning model parameters by default) is 1017 GFLOPs, exceeding the 512 and 750 GFLOPs of TX1 and TX2, respectively. Although the M1000M's FP32 capability also surpasses NX's 845 GFLOPs, this does not necessarily imply faster model inference speed than NX. This is because the NX GPU employs the more recent Volta architecture, whose tensor cores significantly accelerate matrix operations, enhancing deep learning model inference speed beyond that of the M1000M. The AGX surpasses the M1000M in almost all performance parameters listed. Therefore, overall, the M1000M approximately represents the median performance level among these four mainstream embedded graphics cards.

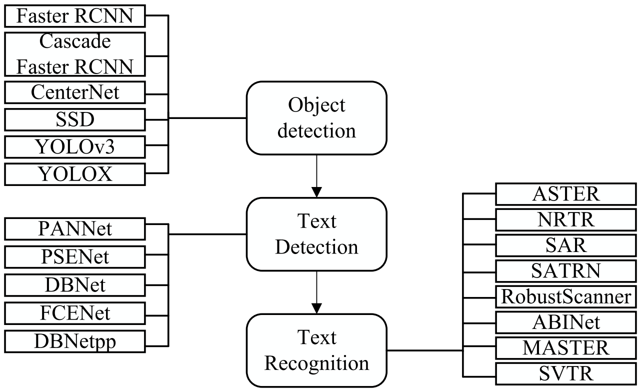

FIGURE 6. Algorithms to be considered in ship IMO number detection and recognition.

In the context of the framework for detecting and recognizing ship IMO numbers, we primarily examine the mainstream algorithms for object detection, text detection, and text recognition developed in recent years (Fig. 6). These algorithms' computational requirements on the GPU during the ship IMO number recognition process are of particular interest.

Object Detection Algorithms: These are mainly used for detecting ships and ship IMO regions. The algorithms under consideration include Faster RCNN, Cascade Faster RCNN, CenterNet, SSD, YOLOv3, and YOLOX. Text Detection Algorithms: These are employed to detect text within the ship IMO region. The algorithms include PANet, PSENet, DBNet, FCENet, and DBNetpp.

Text Recognition Algorithms: These algorithms recognize the text detected within the ship IMO region. The algorithms include ASTER, NRTR, SAR, SATRN, RobustScanner, ABINet, MASTER, and SVTR.

Algorithm 1 Extract IMO Number

```
Require: ship IMO Region image, imoRegionImg;
Ensure: ship IMO number, imo_number;
text_RegionDet ← text_detection(imoRegionImg)
text_RegionRec ← text_recognition(text_RegionDet)
imo_number ← NONE
if 'imo' in text_RegionRec.lower() then
    for item in text_RegionRec do
        if item is a 7 digits then
            imo_number ← item
            break
        end if
    end for
end if
return imo_number if imo_number is not NONE else -1
```

Since the text recognized by the text recognition algorithm includes all text within the ship IMO region and not just the ship IMO number, we need to extract the ship IMO number from this text. Given that a ship IMO region contains at most one ship IMO number, which is a 7-digit number, we only need to perform a simple check on the recognition result. If the recognized text contains the character 'IMO' followed by a 7-digit number, we can identify the ship's IMO number. The pseudocode for ship IMO number recognition is presented in Algorithm 1.

IV. METRICS

A. EVALUATION METRICS FOR OBJECT DETECTION ALGORITHMS

The evaluation of object detection algorithms, specifically for detecting ships and ship IMO regions, focuses on assessing whether the algorithms can run smoothly and their overall detection performance. The goal is to determine if the algorithms can maintain precision while recalling as many true samples as possible. The performance metrics include AP@[0.5:0.95], AR@[0.5:0.95], F-Measure@[0.5:0.95], and detection speed FPS.

AP@[0.5:0.95]: This metric calculates the mean precision over a range of Intersection over Union (IoU) thresholds, from 0.5 to 0.95 with a step of 0.05. It is abbreviated as AP.

TABLE 1. Five graphics cards performance comparison.

GPU	Quadro M1000M	Jetson TX1	Jetson TX2	Jetson Xavier NX	Jetson AGX Xavier
Architecture	Maxwell	Maxwell 2.0	Pascal	Volta	Volta
Power Consumption (W)	40	10	7.5	15	30
Clock Frequency (MHz)	993	998	1300	1100	1377
Memory (GB)	4	4	8	8	32
Memory Type	GDDR5	LPDDR4	LPDDR4	LPDDR4x	LPDDR4x
Memory Bus Width	128-bit	64-bit	128-bit	128-bit	256-bit
Memory Bandwidth (GB/s)	80.19	25.60	58.30	51.20	137.00
CUDA Cores	512	256	256	384	512
Tensor Cores	N/A	N/A	N/A	48	64
FP32 (GFLOPS)	1017	512	750	845	1410

AR@[0.5:0.95]: This metric calculates the mean recall over the same IoU threshold range, from 0.5 to 0.95 with a step of 0.05. It is abbreviated as AR.

F-Measure@[0.5:0.95]: This is a comprehensive performance metric for object detection, calculated as the harmonic mean of AP and AR over the range of IoU thresholds. It is abbreviated as F-Measure and calculated by formula 2×(AP×AR)/(AP+AR).

FPS: Typically expressed in frames per second, this value indicates the number of images the system can process per second.

B. EVALUATION METRICS FOR TEXT DETECTION AND RECOGNITION

For text detection within the ship IMO region, the primary evaluation metrics include Hmean-IoU and FPS. Additionally, we use the product of Hmean-IoU and FPS to assess the overall performance of text detection and select the optimal model algorithm based on this metric. Text detection fundamentally falls under scene segmentation but differs as it pertains to a binary classification problem.

$$Hmean - IoU = n\frac{\prod_{i=1}^{n} IoU_i}{\sum_{i=1}^{n} IoU_i}. \tag{1}$$

$$[4pt]IoU_i = \frac{1}{m_i}\sum_{j=1}^{m_i}\frac{TP_j}{TP_j + FP_j + FN_j}. \tag{2}$$

The calculation of Hmean-IoU is shown in Equation (1), where the computation of IoU_i is derived from Equation (2). In Equation (1), n represents the number of categories to be detected. In Equation (2), m_i denotes the number of detected objects for category i. Here, TP_j refers to the intersection area between the predicted region and the ground truth for object j, FP_j represents the remaining area of the ground truth after subtracting TP_j, and FN_j signifies the remaining area of the predicted region after subtracting TP_j.

For text recognition within the ship IMO region, the primary evaluation metric is word accuracy (*word_acc*), which is defined as the ratio of correctly detected characters to the total number of characters.

C. COMPREHENSIVE EVALUATION METRIC

$$Eff\_Acc = \begin{cases} \alpha X + (1-\alpha)\beta(\frac{Y}{T_{med}})^2 & 0 \le Y \le T_{med} \\ \alpha X + (1-\alpha)f(Y) & T_{med} \le Y \le T_{max} \\ \alpha X + 1 - \alpha & T_{max} \le Y \end{cases} \tag{3}$$

$$f(Y) = \beta + (1-\beta)\sqrt{\frac{Y - T_{med}}{T_{max} - T_{med}}}. \tag{4}$$

In practical applications of object detection, text detection, and text recognition, it is crucial to consider both detection accuracy and speed. To address this, we propose a comprehensive evaluation metric (Equations 3 and 4) that balances accuracy and speed, thereby selecting the most efficient detection and recognition algorithms.

Eff_Acc represents this comprehensive evaluation metric. X denotes detection accuracy, which is: F-measure for object detection, Hmean-IoU for text detection, and *word_acc* for text recognition. The value of X ranges from [0,1].

Y represents detection speed, measured in FPS. The influence on Eff_Acc is modeled as follows: When Y starts from zero and approaches T_{med}, the influence on Eff_Acc increases gradually through the term $(\frac{Y}{T_{med}})^2$; when Y moves from T_{med} to T_{max}, the influence on Eff_Acc decreases gradually through the term $\sqrt{\frac{(Y-T_{med})}{(T_{max}-T_{med})}}$; if Y exceeds T_{max}, its influence on Eff_Acc becomes zero.

The parameter α ranges from [0,1]. A higher value of α indicates a greater impact of detection accuracy on Eff_Acc, while a lower value indicates a greater impact of detection speed on Eff_Acc. The parameter β also ranges from [0,1]. Within the interval [0, T_{med}], a higher β means that detection speed has a larger impact on Eff_Acc. Within the interval [T_{med}, T_{max}], a lower β means that detection speed has a greater impact on Eff_Acc.

Meanwhile, the optimal detection speed for UAV is determined by a complex interplay of factors, including real-time performance, computational constraints, and target characteristics. To ensure timely transmission of detection results, a sufficiently high detection speed is necessary. Given that our target objects are large and slow-moving ships, the demand for real-time performance can be somewhat relaxed. Considering these factors, we have established two detection speed thresholds: $T_{med} = 12$fps as the minimum to maintain basic real-time capabilities and $T_{max} = 18$fps as the maximum to balance real-time performance with computational efficiency.

As the accurate identification of ship IMO numbers is our primary focus, we have assigned a higher weight to detection accuracy. We have set the weights for detection accuracy and speed as $\alpha = 0.7$ and $\beta = 0.7$, respectively. A value of $\beta = 0.7$ indicates that a detection speed of T_{med} is sufficient to meet the basic requirements, while $\alpha = 0.7$ emphasizes the importance of detection accuracy.

V. EXPERIMENTS

A. DATASETS

1) DATASET FOR OBJECT DETECTION

As there is currently no publicly available dataset for ship IMO region detection, we collected 500 raw images containing ships and IMO regions from the website https://routes.shipxy.com/. We then expanded the original dataset through data augmentation techniques, ultimately obtaining a self-built dataset containing 2000 images. We manually annotated the ships and IMO regions in the obtained image dataset, and divided the dataset into training, validation, and test sets at a ratio of 8:1:1. This allowed us to construct a self-built dataset suitable for our ship IMO detection task, laying a foundation for subsequent algorithm training and evaluation.

2) DATASET FOR TEXT DETECTION AND RECOGNITION

IIIT5K Dataset: A standard dataset for regular text recognition. The IIIT5K dataset is derived from 5000 scene photos with varying types and complexities of backgrounds in India. It includes 3000 images as the training set and 2000 as the test set. Each image is annotated with the text regions and their corresponding text content. IIIT5K has become a standard benchmark for scene text recognition tasks, with numerous methods being evaluated on this dataset, promoting the development of this field.

SVT Dataset: A standard dataset for regular text recognition. The SVT dataset is one of the standard benchmark datasets for evaluating scene text recognition methods. The SVT dataset is derived from Google Street View images of building facades. It includes 630 test images, without providing any training images. The images exhibit significant scaling, rotation, and skew variations. The text is particularly blurred and of low resolution. SVT brings scene text recognition tasks closer to real-world applications, becoming an important standard for evaluating method performance in complex real-world scenarios.

ICDAR2013 Dataset, abbreviated as IC13: A standard dataset for regular text recognition. This dataset is mainly used for text detection and recognition and provides samples of various document types and complexities. IC13 includes multiple sub-datasets, including text image samples from natural scenes; it also includes text image samples extracted from digital documents; and it includes handwritten text image samples for handwritten text recognition tasks. These images cover different handwriting fonts, styles, and difficulty levels.

ICDAR2015 Dataset, abbreviated as IC15: Designed for performance evaluation of irregular text detection and recognition. IC15 is derived from real-world scene images, including 1500 training images and 500 test images. The image content includes text in various backgrounds such as buildings, billboards, menus, etc. The text characteristics include arbitrary orientation, size, font, and handwritten or printed styles. This dataset poses great challenges for scene text detection and recognition tasks, containing blurred text of different orientations and types, and high background complexity.

SVTP Dataset: SVTP is a dataset specifically designed for irregular scene text recognition tasks. Like SVT, the SVTP dataset is generated by Google's Street View Time Machine, containing text from 11 countries, with a total of 639 annotated images. In SVTP, text often undergoes various perspective distortions due to the shooting angle. This poses a significant challenge for text detection and recognition tasks, as many existing text recognition algorithms assume the text is planar and do not consider the impact of perspective distortion.

CUTE80 Dataset, abbreviated as CT8: CT8 is a Chinese dataset for irregular text detection and recognition. It contains images from 8 scene categories, including street scenes, merchandise, municipal engineering, etc. The dataset contains a total of more than 13,000 images and provides text box annotations and text content annotations. The CT8 dataset aims to promote the development of text detection and recognition technologies, especially for the detection and recognition of Chinese text in living scenarios. With its broad coverage and fine-grained annotations, CT8 is an important benchmark dataset for Chinese scene text detection and recognition. Researchers can use this dataset to train and evaluate the performance of text detection and recognition models.

B. EXPERIMENTAL RESULTS

For the sake of convenience in subsequent discussions, we need to simplify some technical terms. The simplified terms are listed in Table 2:

TABLE 2. Technical terminology(Term.) and their abbreviation(Abbr).

Term.	Abbr.	Term.	Abbr.
resnet18	**R18**	resnet50-oclip [54]	**R50-oc**
resnet50	**R50**	resnet50-dcnv2 [55]	**R50-dc**
Mobilenetv2 [53]	**MBv2**	F-measure	**F1**
Faster RCNN	**FR**	darknet53 [26]	**DN53**
Cascade Faster RCNN	**CFR**	AP	**P**
CenterNet	**CN**	AR	**R**
PANNet	**PAN**	PSENet	**PSE**
DBNet	**DB**	FCENet	**FCE**
DBNetpp	**DBpp**	RobustScanner	**RS**
tiny	**T**	small	**S**
medium	**M**		

TABLE 3. Comparison of object detection results on the test dataset for ships and ship IMO regions.

MODELs	Backbone	P	R	F1	GFLOPs	Params	FPS	Eff_Acc
FR	R18	0.697	0.736	0.716	145	28.29M	5.12	0.539
	R50	0.695	0.729	0.712	195	41.35M	4.18	0.524
CFR	R18	0.734	0.766	0.750	170	56.33M	2.83	0.536
	R50	0.736	0.768	0.752	218	69.4M	2.62	0.536
CN	R18	0.55	0.633	0.589	38.8	19.09M	5.53	0.457
	R50	0.633	0.693	0.662	50.3	32.11M	4.28	0.490
SSD	vgg16	0.716	0.756	0.735	87.7	24.53M	2.63	0.525
	MBv2	0.294	0.326	0.309	**0.69**	**3.04M**	14.03	0.479
YOLOv3	DN53	0.704	0.753	0.728	46.28	61.95M	5.64	0.556
	MBv2	0.668	0.724	0.695	1.77	3.67M	**25.41**	0.786
YOLOx	T	0.691	0.725	0.708	3.2	5.03M	23.67	0.795
	S	0.737	0.767	0.752	13.4	8.97M	17.23	**0.820**
	M	**0.786**	**0.808**	**0.797**	36.75	25.28M	7.66	0.643

TABLE 4. Comparison of object detection results on the test dataset for ships and ship IMO regions.

MODELs	Backbone	Hmean-IoU	GFLOPs	Params	FPS	Eff_Acc
PAN	R18	0.785	10.96	**12.26M**	16.400	0.837
PSE	R50-oc	0.848	35.75	29.24M	6.860	0.662
DB	R18	0.817	**6.25**	12.34M	**21.740**	0.872
	R50	0.85	11.56	25.41M	14.450	0.863
	R50-dc	0.854	8.88	26.28M	13.980	0.860
	R50-oc	0.864	14.19	25.43M	12.060	0.824
FCE	R50	0.853	10.19	26.26M	17.500	**0.893**
	R50-oc	0.86	12.81	26.28M	13.740	0.860
DBpp	R50	0.862	15.35	26.03M	11.320	0.790
	R50-dc	0.868	12.67	26.91M	11.810	0.811
	R50-oc	**0.888**	17.97	26.05M	10.420	0.780

1) EXPERIMENTAL RESULTS OF SHIP AND SHIP IMO REGION DETECTION

The experimental results for ship and ship IMO region detection are presented in Table 3. The experiments were conducted using the OpenMMLab's MMDetection module, and the models were trained on an Nvidia RTX 3060. The input image size was standardized to 512 × 512. For the five object detection models listed in the table, different backbone networks were employed, including R18, R50, VGG16, DB53, and MBv2, as well as the T, S, and M versions of YOLOx.

The experimental results demonstrate that YOLOx with the M backbone (denoted as YOLOx_M) achieved the best performance in P, R, and F1, with values of 0.786, 0.808, and 0.797, respectively. The SSD with the MBv2 backbone (SSD_MBv2) had the lowest single-precision floating-point operations (0.69 GFLOPs). YOLOv3 with the MBv2 backbone (YOLOv3_MBv2) achieved the highest FPS at 25.41 but had the poorest performance in P, R, and F1. All models had parameter sizes under 100MB, ensuring minimal performance impact on the 4GB VRAM of the M1000M. The results indicate that YOLOx_S had the best Eff_Acc score of 0.82, making it the most suitable model for deployment on UAVs for ship and ship IMO region detection.

2) EXPERIMENTAL RESULTS OF TEXT DETECTION IN SHIP IMO REGIONS

Table 4 presents the experimental results of text detection, comparing five models primarily utilizing backbones R18, R50, R50-oc, and R50-dc. The Hmean-IoU results in the table are derived from the Openmmlab mmocr module, with model training conducted on an NVIDIA A100-SXM4-80GB GPU, using the IC15 dataset, and the text detection images uniformly sized at 320 × 320.

The experimental results indicate that DBpp_R50-oc achieved the best Hmean-IoU result of 0.888. DB_R18 had the lowest single precision floating point operations at 12.26 GFLOPs and the fastest text detection speed at 21.74 FPS. All models had parameter sizes below 30MB, which had minimal impact on the M1000M's performance. The final results show that FCE_R50 had the highest Eff_Acc at 0.893. Although DBpp_R50-oc had the best Hmean-IoU result, its FPS on the M1000M was only 10.42, significantly slower compared to FCE_R50's 17.5 FPS. In terms of Eff_Acc, DB_R18 was slightly lower than FCE_R50.

3) EXPERIMENTAL RESULTS OF TEXT RECOGNITION IN SHIP IMO REGIONS

Given that the computational load for text recognition is significantly influenced by the size of the text area to be recognized, we need to estimate the average text recognition area for ship IMO regions. Initially, all images requiring text recognition are resized to fit within 320 × 320 pixels, maintaining the original aspect ratio. We randomly selected eight images containing ship IMO numbers for text region detection, utilizing FCE_R50, and calculated the detected text area's pixel count for each image (Fig. 7). The average text area to be recognized was approximately 5416 pixels. Considering potential deviations in real-world scenarios,

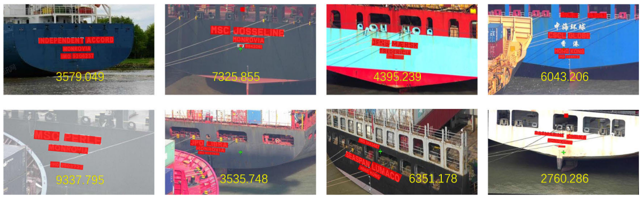

FIGURE 7. Pixel value calculation for IMO text area to be recognized in ships.

TABLE 5. The experimental results for word accuracy (*word_acc*) on text recognition within the six different datasets. The *acc_mean* represents the average *word_acc* measured across the six different datasets.

MODELs	regular text			irregular text			acc_mean
	IIIT5K	SVT	IC13	IC15	SVTP	CT8	
ASTER	0.936	0.895	0.928	0.767	0.806	0.851	0.864
NRTR	0.915	0.883	0.937	0.723	0.778	0.75	0.8310
SAR	0.953	0.884	0.937	0.76	0.833	**0.903**	0.8783
SATRN	**0.96**	0.92	**0.961**	0.803	0.884	0.899	0.9045
RS	0.951	0.893	0.932	0.756	0.808	0.872	0.8687
ABINet	0.96	0.938	0.955	**0.812**	**0.887**	0.879	**0.9052**
MASTER	0.949	0.897	0.952	0.763	0.847	0.885	0.8822
SVTR	0.857	**0.918**	0.944	0.745	0.839	**0.903**	0.8677

TABLE 6. Evaluation results of text recognition models for 80 × 80 ship IMO region.

MODELs	acc_mean	GFlops	Params	FPS	Eff_Acc
ASTER	0.864	**0.91**	20.94M	**22.13**	0.905
NRTR	0.8310	40.83	66.64M	2.82	0.593
SAR	0.8783	31.01	57.41M	2.32	0.623
SATRN	0.9045	63.04	65.59M	3.19	0.648
RS	0.8687	12.78	47.96M	8.36	0.710
ABINet	**0.9052**	5.75	**14.69M**	19.37	**0.934**
MASTER	0.8822	12.41	46.33M	3.51	0.635
SVTR	0.8677	3.59	24.52M	34.48	0.907

we quantified the text area to be recognized as 6400 pixels, equivalent to an 80 × 80 text region.

Table 5 presents the evaluation results based on the datasets IIIT5K, SVT, IC13, IC15, SVTP, and CT8. The experimental results are derived from Openmmlab's mmocr module, with models trained on an NVIDIA A100-SXM4-80GB GPU. The *acc_mean* represents the average *word_acc* measured across the six datasets. The results indicate that SAR achieved the highest *word_acc* on the CT8 dataset, with a score of 0.903; SATRN achieved the best results on the IIIT5K and IC13 datasets, with scores of 0.96 and 0.961, respectively; ABINet performed the best on the IIIT5K, IC15, and SVTP datasets, with scores of 0.96, 0.812, and 0.887, respectively; SVTR achieved the highest scores on the SVT and CT8 datasets, with scores of 0.918 and 0.903, respectively. Ultimately, ABINet showed the highest *acc_mean* across all six datasets, with a value of 0.9052.

Table 6 details the evaluation results of text recognition models for ship IMO region, with the text recognition region being 80 × 80 pixels. The results show that ASTER has the lowest single-precision floating-point operations, at 0.91 GFLOPs; SVTR has the fastest text recognition speed, at 34.48 FPS; ABINet has the smallest parameter size, at 14.69 MB, and also exhibits the highest Eff_Acc, at 0.934, while SVTR's Eff_Acc is slightly lower than ABINet's.

VI. DISCUSSION

A. DISCUSSION OF EXPERIMENTAL RESULTS FOR SHIP IMO REGION DETECTION

For the object detection of ship IMO regions, the optimally selected detection algorithm is YOLOx_S. Based on the current mainstream edge computing GPU chips, we selected a mid-range performance GPU chip, the M1000M, for our laboratory experiments. On this foundation, we evaluated object detection algorithms for ship IMO regions and identified the optimal performing algorithm, YOLOx_S (see Fig. 8).

From Fig. 8(1), it is evident that in the experimental results for ship IMO region object detection, mainstream two-stage object detection algorithms such as FR and CFR exhibit excellent performance, achieving optimal results in terms of P, R, and F1. In contrast, some one-stage algorithms fall short of the performance levels of FR and CFR. However, algorithms like SSD_vgg16, YOLOv3_DN53, YOLOv3_MBv2, YOLOx_T, YOLOx_S, and YOLOx_M can meet or exceed these performance levels.

As shown in Fig. 8(2) and (3), two-stage algorithms significantly surpass most one-stage algorithms in terms of GFLOPs and parameter counts. This indicates that two-stage algorithms are not suitable for deployment on UAVs, as UAVs have limited graphic computation and storage space compared to large-scale GPU computing platforms. Fig. 8(2), (3), and (4) also illustrate that smaller GFLOPs and parameter counts do not necessarily equate to higher FPS. For example, despite its smaller GFLOPs and parameter counts, the SSD_MBv2 model lags behind models like YOLOv3_MBv2, YOLOx_T, and YOLOx_S in terms of FPS.

From Fig. 8(5) and (6), we can observe that in terms of the comprehensive performance evaluation metric, models YOLOv3_MBv2, YOLOx_T, and YOLOx_S significantly outperform other models. However, YOLOx_M, despite having a slightly lower Eff_Acc, achieves a higher F1 score than the other three models. This discrepancy is primarily due to YOLOx_M's low FPS. If more powerful embedded graphic computing chips become available in real-world applications, increasing the FPS of YOLOx_M could potentially alter the final selection outcome.

B. DISCUSSION OF EXPERIMENTAL RESULTS FOR TEXT DETECTION IN IMO SHIP REGIONS

As illustrated in Fig. 9(1), the text detection algorithms show minimal differences in their performance based on the Hmean-IoU metric. Among them, DBpp_R50-oc yields the best results, whereas PAN_R18 performs the worst. From subfigures (2), (3), and (4) of Fig. 9, it is evident that detection algorithms employing the R18 backbone exhibit lower GFLOPs and Params, along with higher FPS; conversely, algorithms using the R50 backbone display the opposite trend. Detection algorithms with the oc [54] module have higher GFLOPs compared to those with the dc [55] module, but the Params are higher with the dc module than with the oc module. This indicates that GFLOPs and Params are not always directly proportional. From subfigures (5) and (6) of Fig. 9, it can be observed that DB_R18 achieves the highest FPS, although its Hmean-IoU evaluation result is lower than that of FCE_R50, leading to a slightly lower Eff_Acc compared to FCE_R50.

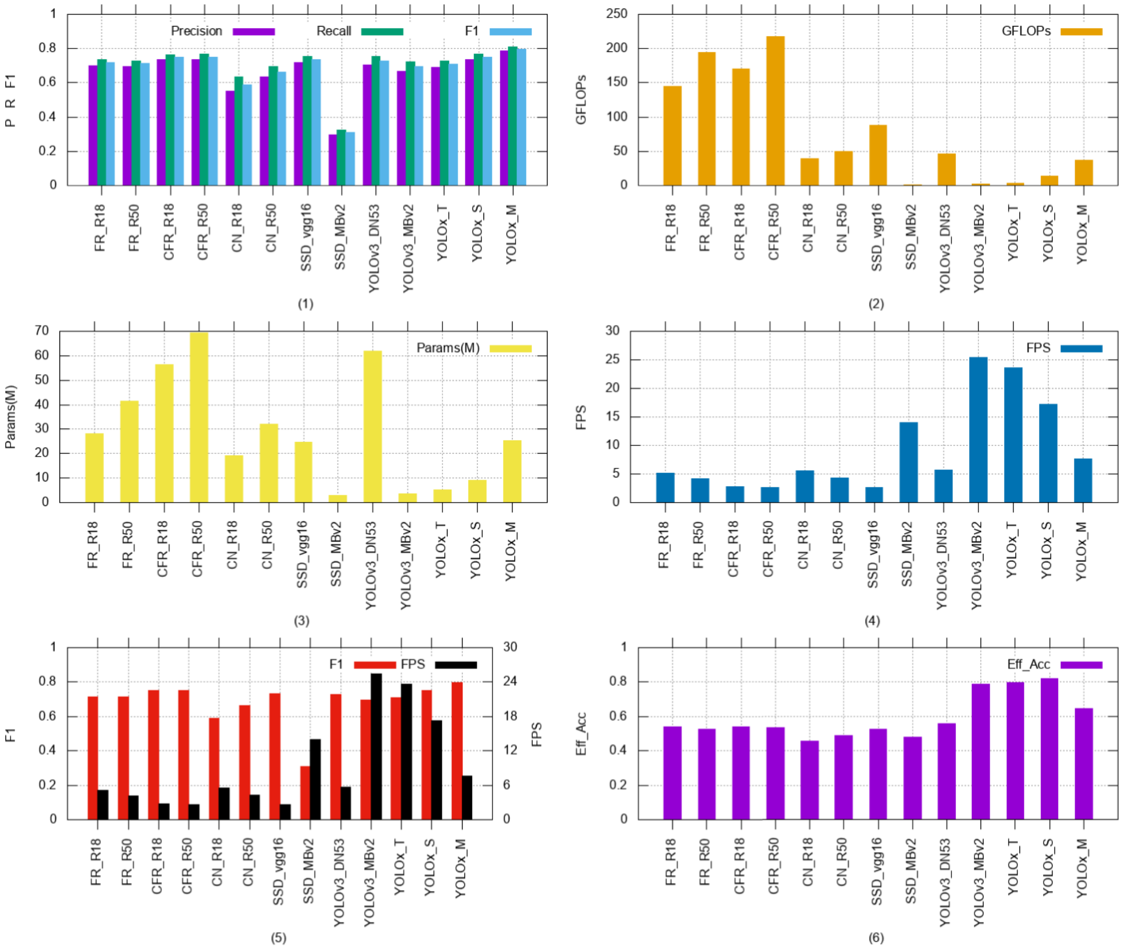

FIGURE 8. Comparison of experimental results for ship IMO region object detection algorithms.

C. DISCUSSION OF EXPERIMENTAL RESULTS FOR TEXT RECOGNITION IN SHIP IMO REGIONS

From Fig. 10, it is evident that the accuracy of the eight text recognition algorithms is generally higher for regular text than for irregular text. However, there are exceptions. For instance, the SAR algorithm exhibits lower accuracy for the regular text dataset SVT compared to the irregular text dataset CT8. Similarly, the SVTR algorithm shows lower accuracy for the regular text dataset IIIT5K compared to the irregular text dataset CT8.

Performance across the IC15 dataset is generally poorer than on other datasets, indicating significant room for improvement in text recognition algorithms on this particular dataset. The recognition accuracy on the CT8 dataset varies considerably, with the lowest accuracy at 0.75 and the highest exceeding 0.9. Overall, the SATRN and ABINet algorithms demonstrate strong performance across the six text datasets, with ABINet showing a slight edge over SATRN in terms of recognition accuracy.

Fig. 11(1) displays the average evaluation results of text recognition algorithms across the six text datasets. From Fig. 11(2) and 11(3), it is evident that the measured GFLOPs and Params(M) exhibit an inverse correlation with FPS, though not strictly. Generally, as the GFLOPs and Params(M) decrease, the FPS increases. For instance, the GFLOPs and Params(M) results for NRTR, SAR, SATRN, RS, and MASTER are significantly higher than those for ASTER, ABINet, and SVTR, yet their FPS is considerably lower than that of ASTER, ABINet, and SVTR.

This inverse relationship is not absolute, as seen when the GFLOPs and Params(M) results are similar, such as between NRTR, SAR, and SATRN, where a positive correlation is observed, and similarly between ASTER and SVTR. Fig. 11(4) highlights that ABINet achieves the highest recognition accuracy in the evaluation of text recognition within IMO ship regions.

D. ANALYSIS OF FAILURES IN MODEL DETECTION AND RECOGNITION

1) ANALYSIS OF FAILURES IN IMO REGION DETECTION

From Fig. 12, it is evident that the ship IMO region detection model occasionally misidentifies two adjacent ships as a single ship (IOU=0.3). Additionally, it may erroneously

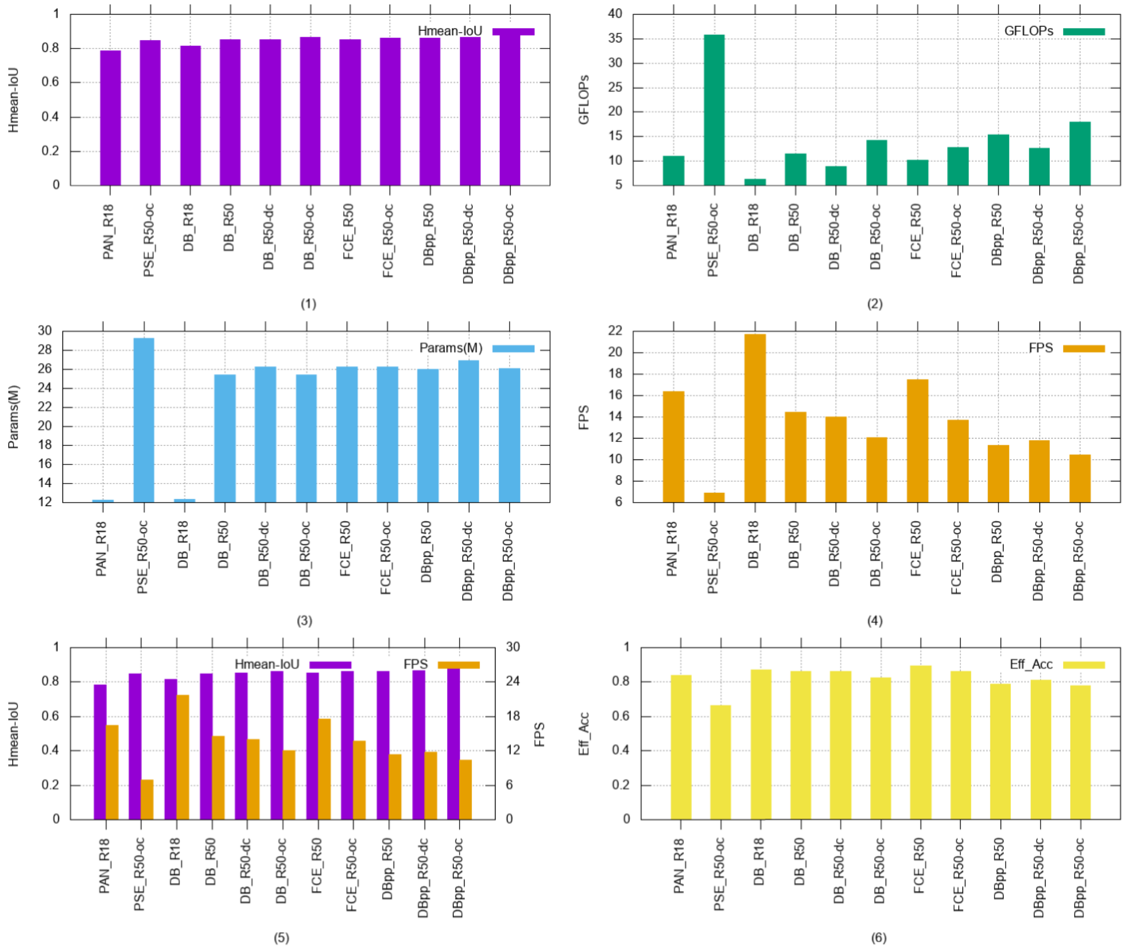

FIGURE 9. Comparative results of text detection experiments in ship IMO regions.

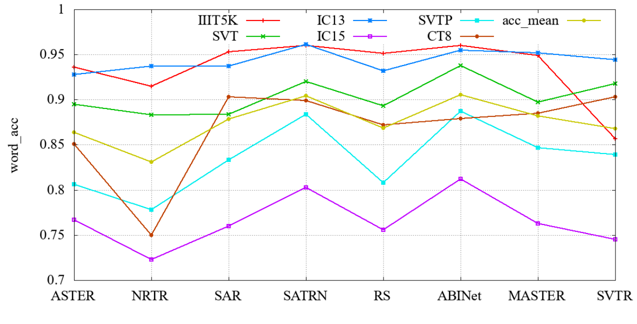

FIGURE 10. Comparative results of text detection experiments in ship IMO regions.

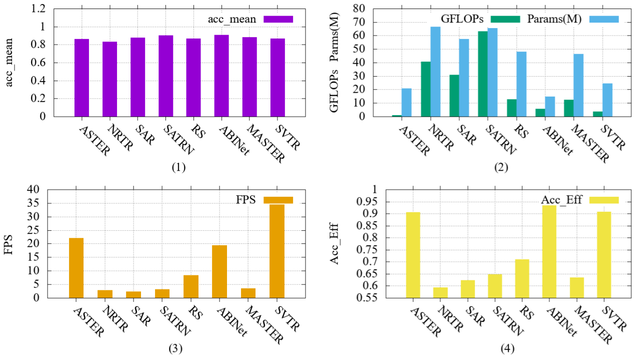

FIGURE 11. Comparative results of text recognition in Ship IMO regions.

recognize containers on the ship or parts of the ship's body as the IMO region or the ship itself. Therefore, the ship IMO region detection model exhibits instances of false positives. The probability of false positives is predominantly below an IOU of 0.5, with fewer instances above this threshold. Simultaneously, the probability of correctly detected targets remains above 0.6. Thus, setting the IOU threshold to 0.5 can help reduce false positives in practical applications of the model.

2) ANALYSIS OF FAILURES IN IMO REGION TEXT DETECTION

In the performance evaluation of text detection for ship IMO regions, the model FCE_R50 has slightly better evaluation results compared to DB_R18. However, these evaluations consider only text coverage accuracy and detection speed, without addressing the issue of text coverage coherence. The FCE_R50 model demonstrates poor coherence in text

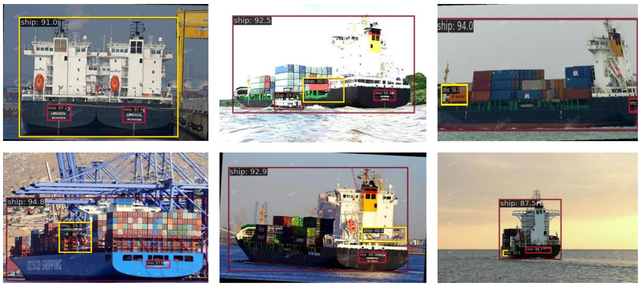

FIGURE 12. Typical error results in ship IMO region detection based on YOLOx_S.

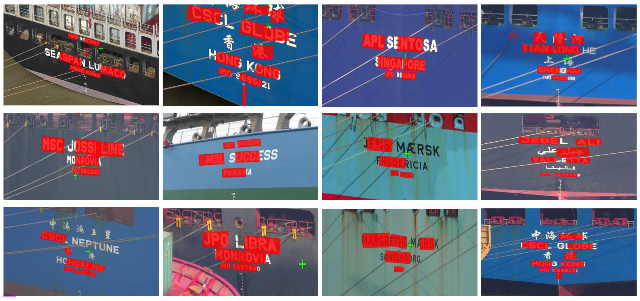

FIGURE 13. Text detection of ship IMO region by model FCE_R50.

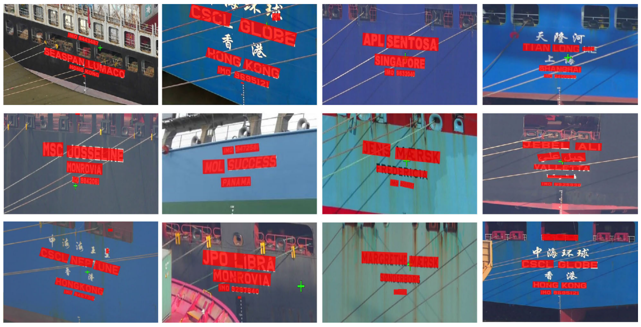

FIGURE 14. Text detection of ship IMO region by model DB_R18.

detection for ship IMO regions (as shown in Fig. 13), frequently failing to fully cover IMO numbers and sometimes detecting a single IMO number as two separate text segments. There are also instances of false positives. On the other hand, the DB_R18 model provides more accurate text detection for IMO numbers (as shown in Figure 14), with no issues of incomplete IMO number coverage and no cases of detecting a single IMO number as multiple text segments. Additionally, it has fewer instances of false positives regarding IMO numbers, thus having a minimal impact on the final recognition. Based on a comprehensive analysis of the two text detection models, it is evident that DB_R18 is more suitable for use in UAV-based ship IMO region text detection.

3) ANALYSIS OF FAILURES IN IMO REGION TEXT RECOGNITION

In the performance evaluation of text recognition models for ship IMO regions, the performance of models ABINet and SVTR is quite similar. The evaluation of text recognition models is primarily based on the accuracy and speed of recognizing all texts. However, for ship IMO region text recognition, the focus is on the accuracy of recognizing the text "IMO" and IMO numbers. Therefore, we need to specifically examine the accuracy of ABINet and SVTR in recognizing the text "IMO" and IMO numbers in the test samples.

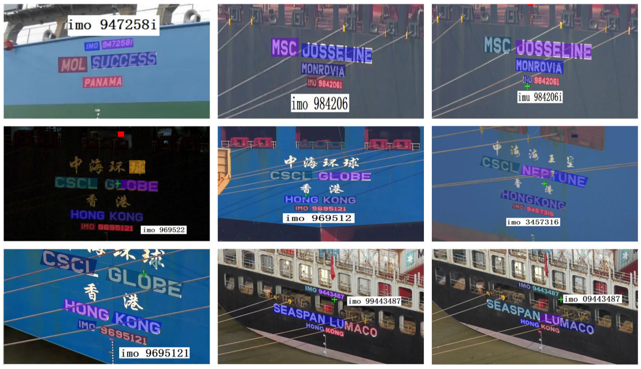

FIGURE 15. Text recognition of ship IMO region by ABINet.

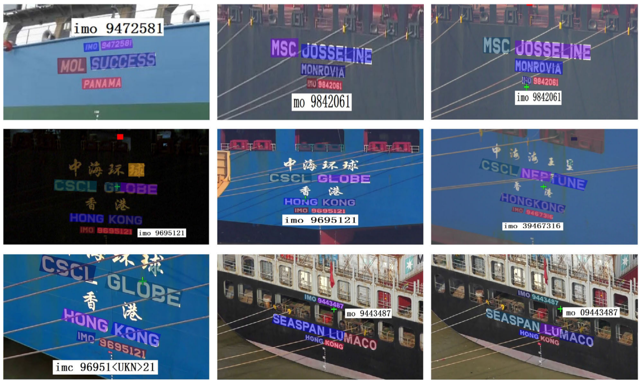

FIGURE 16. Text recognition of ship IMO region by SVTR.

Fig. 15 and Fig. 16 illustrate the errors made by ABINet or SVTR during the recognition of texts in ship IMO regions. ABINet shows accurate recognition of the text "IMO", but makes errors in recognizing IMO numbers, such as misidentifying the last digit "1" as the letter "i" twice and failing to recognize it once. There are also instances of misidentifying the first digit and once misrecognizing "IMO" as "IMU". On the other hand, SVTR's errors mainly occur in recognizing the text "IMO", with more accurate recognition of IMO numbers. For example, it frequently recognizes "IMO" as "MO" and once as "IMC", while the errors in recognizing IMO numbers include occasionally adding an extra digit to the first position.

Comparing the recognition results, it is evident that most of SVTR's errors can be mitigated through programming. For instance, if the recognized number has eight digits, only the last seven digits need to be considered; if the recognized text is "MO" and there is a seven or eight-digit number, "MO" can be inferred to mean "IMO". In contrast, the recognition errors made by ABINet for ship IMO numbers cannot be resolved through programming alone and would require additional training samples and model retraining, which is costly. Overall, SVTR proves to be more practical and suitable as the text recognition model for ship IMO regions in UAV applications.

VII. CONCLUSION

The implementation of ship IMO number detection and recognition on UAVs holds significant importance, enhancing maritime patrol efficiency, improving maritime safety, fostering the development of maritime trade, and maintaining order at sea. This, in turn, provides critical support for maritime safety and economic development. In light of this, the present study proposes a framework for detecting and recognizing ship IMO numbers, encompassing three main functionalities: detecting and recognizing ships and IMO regions, detecting text within ship IMO regions, and recognizing text within these regions.

Additionally, this study describes the specific implementation steps of this framework on UAVs from an engineering perspective and proposes an algorithm for extracting IMO numbers from the recognized text in IMO regions. The study introduces a comprehensive evaluation metric for selecting algorithms suitable for UAVs, applied to the utilized object detection, text detection, and text recognition algorithms. The results of the final experiments, filtered through this comprehensive evaluation metric, indicate that the most suitable object detection algorithm for ship and IMO region detection on UAVs is YOLOx_S. For text detection in ship IMO regions, the optimal algorithm is FCE_R50, although the performance of DB_R18 is very close. For text recognition in ship IMO regions, ABINet is the optimal algorithm, with SVTR showing very close performance.

During the discussion phase, we identified several points of improvement. The YOLOx_S algorithm for object detection exhibits some false positives in ship and IMO region detection, which can be mitigated by setting a higher IOU threshold. The FCE_R50 text detection algorithm encounters issues with text continuity, often failing to fully detect IMO numbers, whereas DB_R18 performs better in terms of text detection continuity. Both ABINet and SVTR text recognition algorithms show errors in recognizing the text "IMO" and IMO numbers. However, ABINet's errors cannot be resolved through programming alone, requiring additional training samples, whereas most of SVTR's errors can be corrected programmatically at a lower cost.

Through the discussion of experimental results, we have found that the comprehensive evaluation metric's selection results are relatively reliable but should not be the sole criterion for selection. For instance, the text detection algorithm for ship IMO regions should also consider text continuity, and the text recognition algorithm should also factor in the accuracy of recognizing ship IMO numbers.

REFERENCES

[1] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2124–2136, Mar. 2019.

[2] A. Tullu, B. Endale, A. Wondosen, and H.-Y. Hwang, "Machine learning approach to real-time 3D path planning for autonomous navigation of unmanned aerial vehicle," *Appl. Sci.*, vol. 11, no. 10, p. 4706, May 2021.

[3] H. Y. Lee, H. W. Ho, and Y. Zhou, "Deep learning-based monocular obstacle avoidance for unmanned aerial vehicle navigation in tree plantations: Faster region-based convolutional neural network approach," *J. Intell. Robotic Syst.*, vol. 101, no. 1, pp. 1–18, Jan. 2021.

[4] L. He, N. Aouf, and B. Song, "Explainable deep reinforcement learning for UAV autonomous path planning," *Aerosp. Sci. Technol.*, vol. 118, Nov. 2021, Art. no. 107052.

[5] S. Zhang, Y. Li, and Q. Dong, "Autonomous navigation of UAV in multi-obstacle environments based on a deep reinforcement learning approach," *Appl. Soft Comput.*, vol. 115, Jan. 2022, Art. no. 108194.

[6] A. R. Dooraki and D.-J. Lee, "A multi-objective reinforcement learning based controller for autonomous navigation in challenging environments," *Machines*, vol. 10, no. 7, p. 500, Jun. 2022.

[7] A. A. Obaid and H. Koyuncu, "Obstacle avoidance in unmanned aerial vehicles using image segmentation and deep learning," in *Proc. Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Ankara, Turkey, Oct. 2022, pp. 912–915.

[8] S. Badrloo, M. Varshosaz, S. Pirasteh, and J. Li, "Image-based obstacle detection methods for the safe navigation of unmanned vehicles: A review," *Remote Sens.*, vol. 14, no. 15, p. 3824, Aug. 2022.

[9] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, "Embedded UAV real-time visual object detection and tracking," in *Proc. IEEE Int. Conf. Real-time Comput. Robot. (RCAR)*, Irkutsk, Russia, Aug. 2019, pp. 708–713.

[10] S.-J. Hong, Y. Han, S.-Y. Kim, A.-Y. Lee, and G. Kim, "Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery," *Sensors*, vol. 19, no. 7, p. 1651, Apr. 2019.

[11] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[12] J. Zhang, X. Liang, M. Wang, L. Yang, and L. Zhuo, "Coarse-to-fine object detection in unmanned aerial vehicle imagery using lightweight convolutional neural network and deep motion saliency," *Neurocomputing*, vol. 398, pp. 555–565, Jul. 2020.

[13] M. Rabah, A. Rohan, M.-H. Haghbayan, J. Plosila, and S.-H. Kim, "Heterogeneous parallelization for object detection and tracking in UAVs," *IEEE Access*, vol. 8, pp. 42784–42793, 2020.

[14] B. K. S. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, T. P. Breckon, and H. P. H. Shum, "Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Montreal, BC, Canada, Oct. 2021, pp. 1223–1232.

[15] I. V. Saetchnikov, E. A. Tcherniavskaia, and V. V. Skakun, "Object detection for unmanned aerial vehicle camera via convolutional neural networks," *IEEE J. Miniaturization Air Space Syst.*, vol. 2, no. 2, pp. 98–103, Jun. 2021.

[16] L. Wei, Y. Luo, L. Xu, Q. Zhang, Q. Cai, and M. Shen, "Deep convolutional neural network for Rice density prescription map at ripening stage using unmanned aerial vehicle-based remotely sensed images," *Remote Sens.*, vol. 14, no. 1, p. 46, Dec. 2021.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[18] 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.*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

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

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.

[23] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6154–6162.

[24] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7794–7803.

[25] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 821–830.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[27] 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)*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.

[28] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525.

[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[30] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, Mar. 2020.

[31] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[32] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[33] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[34] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[35] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[36] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, Jun. 2023, pp. 7464–7475.

[37] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 20–36.

[38] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 9328–9337.

[39] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 8439–8448.

[40] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11474–11481.

[41] S.-X. Zhang, X. Zhu, J.-B. Hou, C. Liu, C. Yang, H. Wang, and X.-C. Yin, "Deep relational reasoning graph network for arbitrary shape text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 9696–9705.

[42] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 3122–3130.

[43] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 919–931, Jan. 2023.

[44] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.

[45] F. Sheng, Z. Chen, and B. Xu, "NRTR: A no-recurrence sequence-to-sequence model for scene text recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sydney, NSW, Australia, Sep. 2019, pp. 781–786.

[46] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8610–8617.

[47] J. Lee, S. Park, J. Baek, S. J. Oh, S. Kim, and H. Lee, "On recognizing texts of arbitrary shapes with 2D self-attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, Jun. 2020, pp. 2326–2335.

[48] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, "RobustScanner: Dynamically enhancing positional clues for robust text recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 135–151.

[49] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 7094–7103.

[50] N. Lu, W. Yu, X. Qi, Y. Chen, P. Gong, R. Xiao, and X. Bai, "MASTER: Multi-aspect non-local network for scene text recognition," *Pattern Recognit.*, vol. 117, Sep. 2021, Art. no. 107980.

[51] Y. Du, Z. Chen, C. Jia, X. Yin, T. Zheng, C. Li, Y. Du, and Y.-G. Jiang, "SVTR: Scene text recognition with a single visual model," 2022, *arXiv:2205.00159*.

[52] R. Sun, T. Lei, Q. Chen, Z. Wang, X. Du, W. Zhao, and A. K. Nandi, "Survey of image edge detection," *Frontiers Signal Process.*, vol. 2, Mar. 2022, Art. no. 826967.

[53] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520.

[54] W. Yu, Y. Liu, W. Hua, D. Jiang, B. Ren, and X. Bai, "Turning a CLIP model into a scene text detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, Jun. 2023, pp. 6978–6988.

[55] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets V2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 9300–9308.

ZHI-BO CAO received the Ph.D. degree in computer application technology from the South China University of Technology, Guangzhou, China, in 2014.

During the Ph.D. degree, his research interests include developing log models for high-performance computing centers, optimizing scheduling algorithms for high-performance computing environments through logs and log models, and thereby reducing the energy consumption of high-performance computing centers. He has published six SCI and EI retrieved articles and obtained two invention patents. His current research interest includes artificial intelligence algorithms, focusing on how to apply artificial intelligence algorithms to real-world work environments to solve real-world problems. This includes how to solve the problem of using drones to detect and identify IMO numbers of ships for maritime management departments and how to use those algorithms to solve the problem of collision and strike on unmanned aerial vehicles.