

Article

Ship Contour: A Novel Ship Instance Segmentation Method Using Deep Snake and Attention Mechanism

Chen Chen ^{1,2}, Songtao Hu ^{1,2}, Feng Ma ^{3,*} , Jie Sun ⁴, Tao Lu ^{1,2}  and Bing Wu ³

¹ School of Computer Science & Engineering, Wuhan Institute of Technology, Wuhan 430205, China; chenchen0120@wit.edu.cn (C.C.); husongtao0609@gmail.com (S.H.); lut@wit.edu.cn (T.L.)

² Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan 430205, China

³ Intelligent Transportation System Center, Wuhan University of Technology, Wuhan 430205, China; bing.wu@whut.edu.cn

⁴ Nanjing Smart Water Transportation Technology Co., Ltd., Nanjing 210028, China; sunjie@smartwaterway.com

* Correspondence: martin7wind@whut.edu.cn

Abstract: Ship instance segmentation technologies enable the identification of ship targets and their contours, serving as an auxiliary tool for monitoring, tracking, and providing critical support for maritime and port safety management. However, due to the different shapes and sizes of ships, as well as the complexity and fluctuation of lighting and weather, existing ship instance segmentation approaches frequently struggle to accomplish correct contour segmentation. To address this issue, this paper introduces Ship Contour, a real-time segmentation method for ship instances based on contours that detects ship targets using an improved CenterNet algorithm. This method utilizes DLA-60 (deep layer aggregation) as the core network to ensure detection accuracy and speed, and it integrates an efficient channel attention (ECA) mechanism to boost feature extraction capability. Furthermore, a Mish activation function replaces ReLU to better adapt deep network learning. These improvements to CenterNet enhance model robustness and effectively reduce missed and false detection. In response to the issue of low accuracy in extracting ship target edge contours using the original deep snake end-to-end method, a scale- and translation-invariant normalization scheme is employed to enhance contour quality. To validate the effectiveness of the proposed method, this research builds a dedicated dataset with up to 2300 images. Experiments demonstrate that this method achieves competitive performance, with an accuracy rate of $AP_{0.5:0.95}$ reaching 63.6% and a recall rate of $AR_{0.5:0.95}$ reaching 67.4%.

Keywords: ship instance segmentation; deep snake; attention mechanism



Received: 9 January 2025

Revised: 27 January 2025

Accepted: 6 March 2025

Published: 8 March 2025

Citation: Chen, C.; Hu, S.; Ma, F.; Sun, J.; Lu, T.; Wu, B. Ship Contour: A Novel Ship Instance Segmentation Method Using Deep Snake and Attention Mechanism. *J. Mar. Sci. Eng.* **2025**, *13*, 519. <https://doi.org/10.3390/jmse13030519>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the rapid development of artificial intelligence technology, researchers have begun to experiment with computer vision to achieve ship instance segmentation, which has had a significant impact on the fields of water traffic management [1,2] and ship assisted driving [3]. Compared to object detection, ship instance segmentation can provide more accurate ship position and shape information. Traditional ship segmentation approaches [4], on the other hand, frequently relied on manually defined features and criteria, which may not perform well in complicated environments or when ships vary dramatically. Furthermore, these conventional approaches frequently required manual parameter and threshold adjustments, making it difficult to respond to various scenarios and ship targets.

As deep learning gained popularity, deep neural network-based instance segmentation techniques appeared and excelled in a variety of segmentation tasks. However, when the ships become massive or they are crowded together, the existing deep learning-based instance segmentation methods generally perform unsatisfactorily. For example, the classic instance segmentation algorithm “You Only Look At Coefficients” (Yolact) [5] often misinterprets multiple overlapping ship targets as a single target during segmentation, which can lead to a notable decline in performance. Furthermore, intricate aquatic settings and interference signals (such illumination and fogging) consistently impacted the model’s stability and significantly decreased their accuracy. The distribution of ship datasets is highly irregular, with scarce samples, making it challenging for some general instance segmentation algorithms to accurately capture the features of ship targets. For example, the Segmentation of Objects by Localization (SOLO) [6] instance segmentation algorithm incorrectly identify other obstacles on the water surface as ship targets. They are all in desperate need of optimization and enhancement.

Therefore, this research proposes a real-time ship instance segmentation method called Ship Contour, which utilizes the CenterNet [7] to act as a target detector to detect ship targets, with DLA-60 [8] as a feature extraction network. In particular, the proposed method integrates the efficient channel attention (ECA) [9] mechanism into the block module, which greatly improves the model’s features extraction ability and detection accuracy, while the increased computational cost can be negligible. The Mish activation function [10] is also utilized in place of the ReLU activation function in order to adjust to the deeper network structure of DLA-60 and compensate for the ReLU activation function’s rapid decrease in accuracy in deeper networks. During the segmentation stage, the deep snake [11] approach is used, which progressively optimizes the target’s initial contour into the target’s border for target segmentation. In addition, a scale- and translation-invariant normalizing approach is presented to address the challenge of contour distortion in the convex region and enhance the contour’s quality. To assess the effectiveness of the Ship Contour algorithm, a dataset of over 2300 images linked to ships has been created. These images were picked carefully, which covers a wide range of scenarios, including ports, coastal regions, inland rivers, daytime and night-time, foggy days, etc. This dataset is designed to enhance the model’s capacity for generalization, enabling it to precisely segment ship targets across a range of complex environments.

The remaining part of this paper is organized as follows. Section 2 reviews the relevant literature. Section 3 describes the Ship Contour approach. Section 4 presents the ship segmentation dataset and verifies the effectiveness of the algorithm. Finally, Section 5 provides the conclusion regarding the application of the approach.

2. Literature Review

2.1. Instance Segmentation Algorithms

Instance segmentation is a current topic in computer vision, which seeks to identify the location and mask of each unique object instance. Instance segmentation is more sophisticated than object detection [12] and categorization [13], and it has numerous applications in autonomous driving, medical image analysis, robotics, etc. The number of instance segmentation techniques has increased along with the advancement of deep learning. Deep learning-based instance segmentation techniques provide improved efficiency and accuracy by autonomously extracting image features and reducing image noise without the need for human interaction, in comparison to classic vision approaches.

Instance segmentation methods can be classified into two types: region instance segmentation methods and contour instance segmentation methods. The region-based instance segmentation approach can be further classified into single-stage and two-stage

instance segmentation methods. Among them, instance segmentation is carried out in two steps using two-stage segmentation methods. Generally, the first stage is designed for target detection, which generates candidate boxes or regions to identify image areas that may contain the target. Using the candidate boxes or regions as input, pixel-level segmentation and classification are subsequently carried out in the following stage. High segmentation accuracy is typically attained by these two-stage segmentation methods. He et al. [14] presented a two-stage segmentation approach called Mask R-CNN, which integrated two major modules of object detection and segmentation. The fundamental network structure was based on the excellent ResNet-FPN structure [15,16], and multi-level feature maps were employed to enhance the detection performance of small and multi-scale targets. Furthermore, it substituted region of interest (ROI) align for ROI pooling, eliminating with the rounding process and maintaining the accuracy of floating-point quantities. Additionally, it made use of the fully convolutional network (FCN) structure [17] to build an end-to-end network using convolution and deconvolution operations, resulting in satisfactory segmentation outcomes and pixel-level classification. In order to further improve instance segmentation performance, Liu et al. [18] presented the path aggregation network (PANet), which aggregates low- and high-level features on the basis of Mask R-CNN. According to Huang et al. [19], the current mask scoring system makes use of classification markers, which lacks specific evaluation procedures. Thus, an MS R-CNN model was suggested, and the mask evaluation criteria were adjusted based on Mask R-CNN. The model instance segmentation performance was enhanced by including a Mask IOU branch to forecast Mask. However, this strategy also showed some significant shortcomings. In fact, objects in images often contain complex edge shapes, which are very difficult to fit in a segmentation module, especially when significant errors occur in the detector, since it mostly relies on a detection box. Furthermore, these operations generally need a large amount of computation.

In order to improve the efficiency of instance segmentation, researchers have focused on single-stage instance segmentation approaches. These approaches use intense pixel categorization and position regression on the original image to achieve instance segmentation and localization. Currently, single-stage instance segmentation approaches can be divided into anchor-based and anchor-free segmentation methods depending on the use of anchor boxes. Yoloact [5] uses anchor boxes as its foundation and adds a mask branch to the pre-existing single-stage object identification model. It uses two branch tasks running in parallel to achieve instance segmentation. Prototype masks are one method that makes use of FCN to create a set of common prototype masks. The other is mask coefficients, which anticipate a sequence of mask coefficients for each occurrence by adding more heads to the object detection branch. Yoloact's real-time object detection and instance segmentation capabilities come with a speedy, high-performing feature set. It could process targets of different scales and shapes by using anchor boxes for object detection and position regression. However, Yoloact performs unsatisfactorily when facing overlapping targets. Wang et al. [20] introduced a single-stage anchor free model, namely Solov2, that dynamically segmented each instance in the image based on position and size, without bounding boxes. This model lowered the amount of inference calculation, increasing detecting speed and accuracy. Yang et al. [21] integrated a convolutional block attention module (CBAM) into the detection head of the Solov2 for refined features taking into account the attention weight coefficients in both the channel and spatial dimensions, resulting in a significant reduction in error detection rates. Despite much completed work as discussed previously, single-stage segmentation approaches generally perform worse than two-stage segmentation methods in terms of accuracy.

In fact, contour is also an important representation of instance segmentation, which is mostly based on the active contour snake approach [22]. The traditional snake segmentation algorithm optimized an energy function defined by pixel brightness or gradient in order to reduce the energy function while matching the target edge. The snake approach required fewer parameters than pixel-based representations. However, this method tends to identify local optimal solutions since the manually set energy function is non-convex and prone to being stuck in local optima. Presently, researchers have merged contour-based segmentation techniques with deep learning, which was inspired by Mask-RCNN and Yolo-Series. In contrast to the conventional method of finding object outlines by maximizing energy functions, Marcos et al. [23] and Yang et al. [24] proposed novel instance segmentation algorithms that directly regressed object boundary points from RGB images. These approaches have demonstrated better results in instance segmentation. Peng et al. [11] proposed deep snake for representing objects through edge modeling. A learning-based strategy was used to execute the classic snake algorithms concept, which involved first giving an initial shape and then extracting picture attributes at each vertex. Circular convolution was then used to learn features from the contour. In order to conform the contour to the object boundary, offsets were finally regressed at each vertex. Compared to two-stage segmentation algorithms, this approach achieved state-of-the-art performance and could be used for real-time segmentation problems.

Recently, prominent companies all over the world thrown themselves into the development and application of pre-trained large models. In the domain of image segmentation, large models of segmentation such as the Segment Anything Model (SAM) [25] and Segment Everything in Context (SegGPT) [26] also garnered widespread attention. These large-scale models employ the interaction segmentation method. Using the SAM as an example, the SAM employs the prompt paradigm defined in natural language processing to computer vision, generating effective segmentation masks for any given prompt such as foreground or background points, target boxes, and text. The model has been trained on generic datasets containing 11 million images and over a billion masks, resulting in a robust zero-shot capability. While these models perform excellently in many scenarios, the fact is that they often perform worse than the specially designed model in specialized domains. This is because the data and task characteristics in specialized fields differ significantly from general scenarios. Furthermore, such large models typically rely on substantial computational resources, which pose significant challenges for fine-tuning the models and executing real-time tasks.

2.2. Ship Instance Segmentation Algorithms

Ship instance segmentation is the process of precisely separating ship objects from the backdrop in an image in order to give precise ship location in an FPV or BEV image. A few algorithms have recently been proposed by researchers to increase the precision and effectiveness of ship segmentation jobs. Sun et al. [27] suggested Global Mask R-CNN as a solution to the edge redundancy segmentation and insufficient smoothness of masks in ship instance segmentation. During the pooling phase, global pixels were taken into account using exact ROI pooling. For instance, in mask prediction, the global mask head used two pathways and extracted more global semantic information from them. This method increased the accuracy of ship instance segmentation while preserving global information. However, it did not address the issue of unduly complicated calculations, which make real-time segmentation practically impossible. An orientated silhouette matching network (OSM-Net) was presented by Huang et al. [28] to segment revolving ships in remote sensing images. In order to better match the ship silhouette, they created a novel orientated polar template mask using orientated mask IoU, in contrast to earlier instance segmentation

techniques. To increase detection precision, a multiscale feature propagation and fusion module was employed. Ma et al. [29] proposed a customized neural network-based ship segmentation algorithm named MrsNet. It used a lightweight and effective Faster YOLO network to extract features at various levels from bird's-eye-view (BEV) radar images, capturing deep semantic features of ship pixels as well as fine-grained edge information.

In fact, many data sources for ship segmentation are infrared or synthetic aperture radar (SAR) images. The visible images, which contain richer color and texture information, are more suited for ship segmentation. Zhang et al. [30] suggested SwinSeg, which is a hybrid network that combines a lightweight multi-layer perceptron with a SwinTransformer. They gathered and labelled a dataset of maritime vessels, with which they verified the SwinSeg's efficacy. Nevertheless, small target ships and crowded water traffic situations were not taken into account in their work.

Previous methods for segmenting ship instances were mostly created for clear days and infrequently took the impact of bad weathers into account. Sun et al. [31] developed a ship instance segmentation framework (IRDCLNet) based on dynamic contour learning and interference reduction in foggy settings in order to overcome this issue. The interference reduction module was created to address the issue of missing ship detection and lessen interference brought on by fog. To assess efficacy, 5739 real and simulated ship photos in foggy settings were assembled into a dataset, namely Foggy ShipInsseg. The Foggy ShipInsseg dataset experiments demonstrated that this method produced state-of-the-art results.

It is clear from the description above that most circumstances could be satisfied by the instance segmentation techniques now in use. Nonetheless, there is a dearth of research on instance segmentation for ships. In addition, ships have a variety of shapes, such as ellipses, rectangles, and irregular shapes. Accurately segmenting complex-shaped ships may have many issues when using the commonly used instance segmentation algorithms. These algorithms lack large-scale annotated datasets and specialized evaluation metrics for ships, which prevents them from completely taking into account the unique requirements and scenarios of ships. Instead, they frequently employ non-dedicated datasets for training and evaluating. Consequently, the Ship Contour approach is proposed in this work, which considers the special requirements of instance segmentation in the fields of ship navigation. It can resolve the challenges of precisely capturing ship targets amid the complicated landscapes and various weather discussed previously.

3. A Proposed Approach

Figure 1 represents Ship Contour, a ship segmentation process inspired by deep snake. It is made up of three modules: candidate boxes with CenterNet, initial contour suggestion, and contour deformation. Specifically, CenterNet predicts a bounding box, which is then used to create a diamond contour by taking the midpoint of each edge. Following that, the diamond contour is uniformly sampled to 40 points, and 40 offsets are generated via circular convolution. Sequentially, the extreme points of the instance target (top, leftmost, bottom, rightmost pixels) can be obtained by deforming the four vertices of the diamond. At the extreme points, a line segment is stretched along the direction of the bounding box to better represent the position information of the target. The line segment is usually $1/4$ the length of the bounding box, with the two ends connected to form an octagonal contour as the initial contour. Similarly, 128 spots are selected uniformly from the octagonal contour. The Ship Contour takes the original contour as input and returns 128 offsets pointing from each point to the desired boundary point, resulting in instance segmentation.

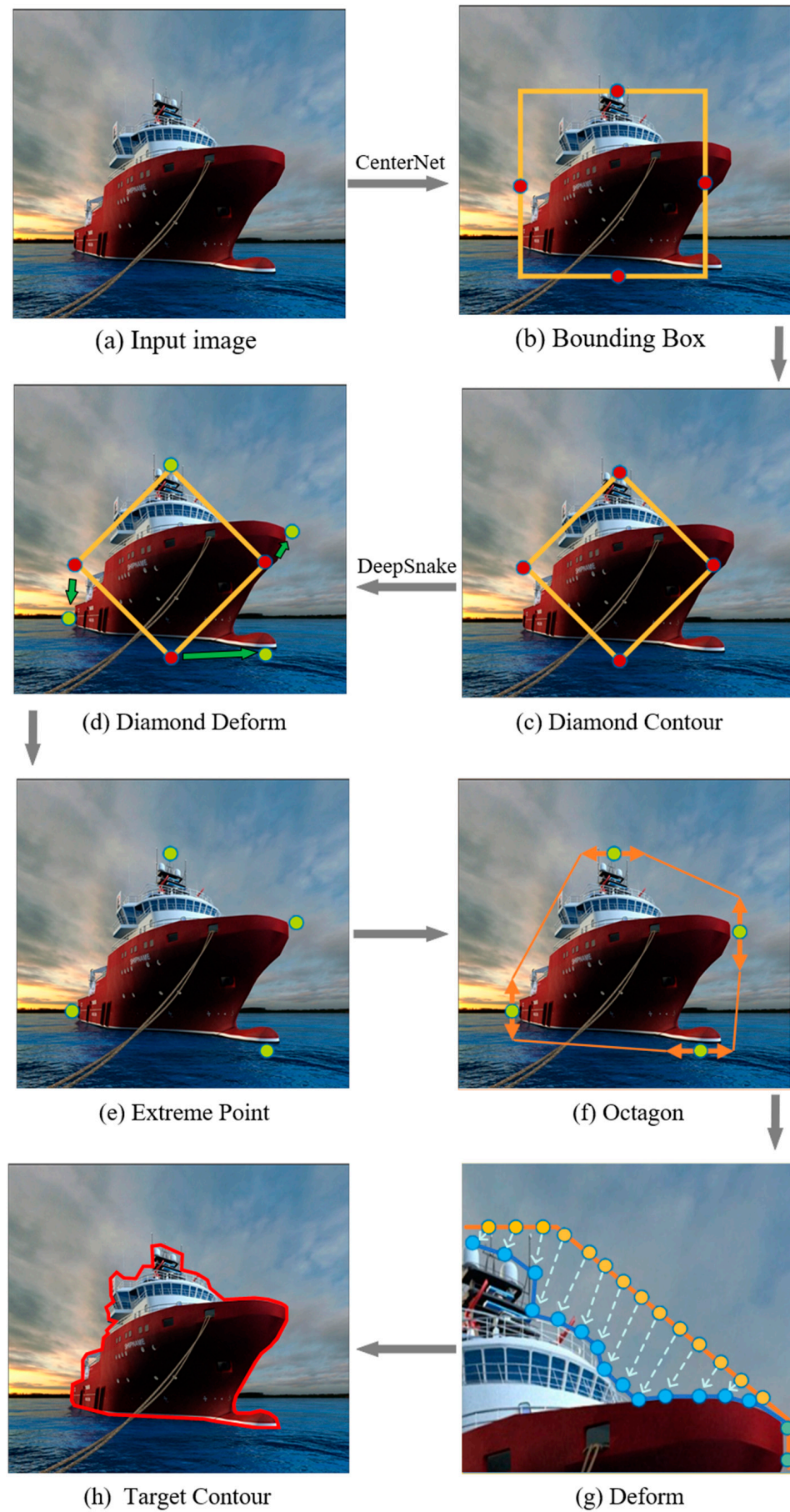


Figure 1. The flowchart of Ship Contour.

3.1. An Improved CenterNet Detector

CenterNet is an anchor-free box model that uses center points to simulate anchor boxes. To be more specific, it employs a threshold to filter the top 100 peak points of the heatmap generated by the feature extraction network before determining the final target center point. Regression analysis was then performed to establish the target's category and position based on the picture properties of the center point.

The classic deep snake achieved a reasonable compromise between speed and accuracy by employing the DLA-34 structure. However, in navigation, the ship targets are generally small. As a result, the accuracy of the DLA-34 network might not be satisfactory. By employing a more intricate DLA-60 structure as the feature extraction network, our research enhanced CenterNet to guarantee that the segmentation module can obtain more accurate bounding boxes in navigation. The left half of Figure 2 describes the entire pipeline of CenterNet, with the boundary regression representing the width height regression and the center point of the target represented by the center heatmap.

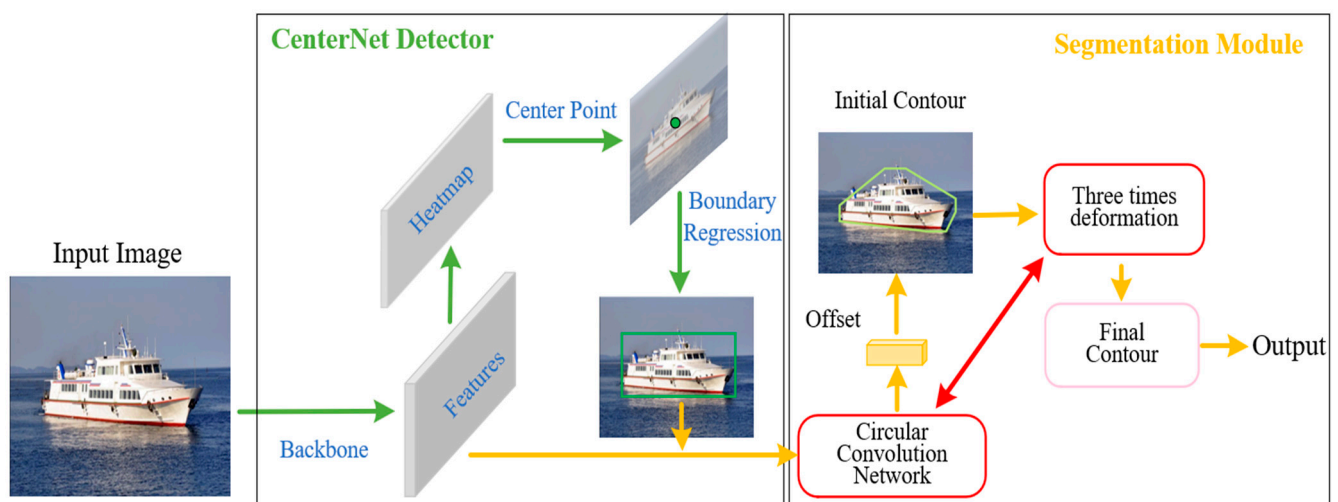


Figure 2. The overall architecture diagram of Ship Contour.

DLA is a unique framework that merges features iteratively and hierarchically, resulting in improved accuracy with less computation. The network structure of DLA is shown in Figure 3, with the aggregation node in green and the convolutional block in black. The yellow arrow denotes iterative deep aggregation (IDA), while the red box represents hierarchical deep aggregation. The function of HAD is to combine the features of shallower and deeper layers, resulting in a better combination than simple residual networks (ResNets) and dense networks (DenseNets) [32]. The encoding step of a network includes IDA and HAD. The process of decoding involves combining the outputs of nearby HDA structures. During the merging process, deconvolution (Deconv) is employed to transform low-resolution features. Deformable convolution (Deformconv) [33] is applied to the outputs of neighboring HDAs, which are then repeatedly combined to form a tree-like structure. The network architecture of the aggregation nodes and convolutional block is shown in Figure 4. Figure 3 makes use of these two structures, where the core function of the aggregation node is to integrate and compress the input features. Its main objective is to extract the key information from the inputs and, through learning, select and project the critical features so that the output dimension matches the single input dimension. In this way, the aggregation node effectively merges features from different levels, eliminating redundant data and conserving the most valuable attributes for the subsequent layers. This feature of the aggregation node optimizes the flow of information, strengthens the

expressive capability of the model, and prevents information overload, thus improving both computational efficiency and accuracy.

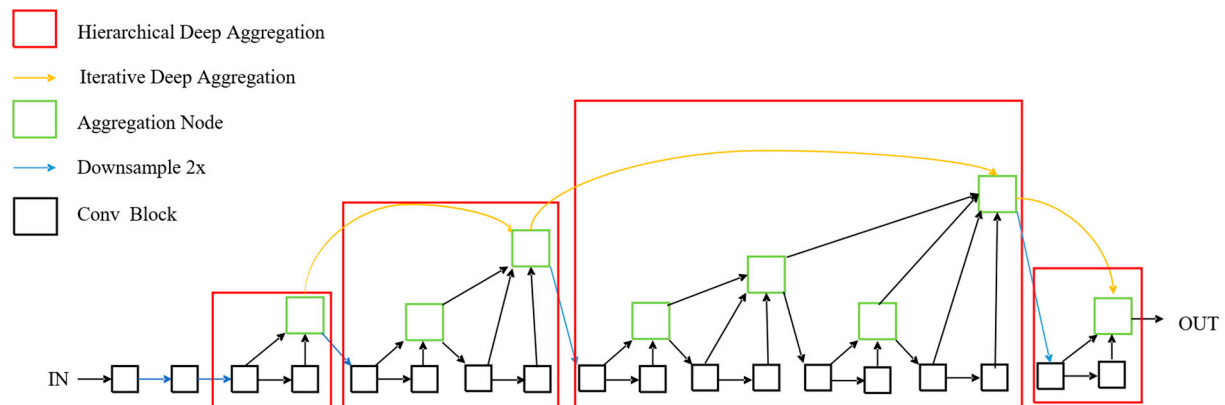


Figure 3. The network architecture of DLA.

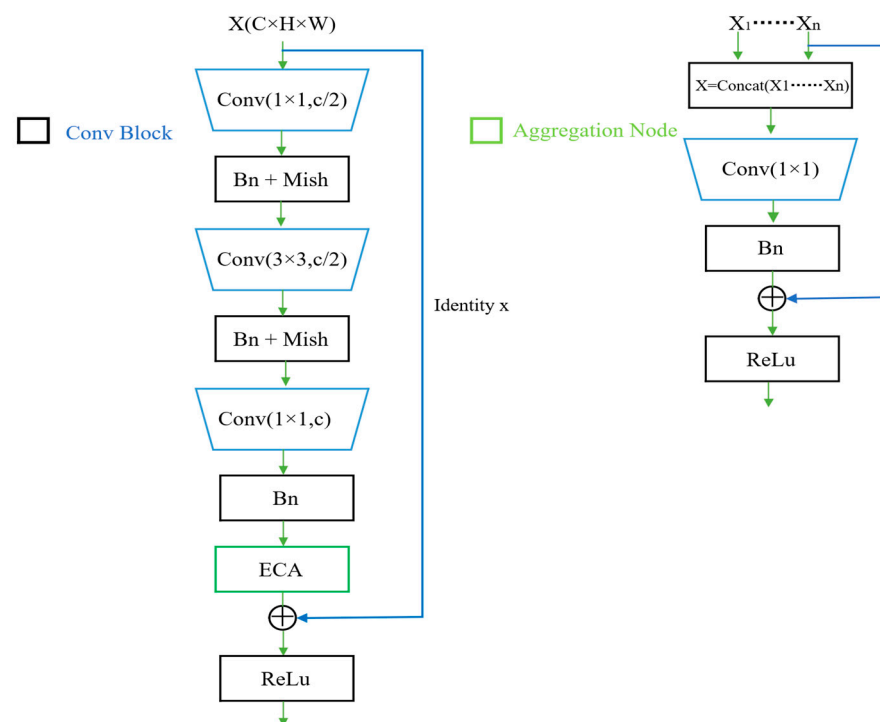


Figure 4. The network structure of the convolutional block and aggregation node.

In navigation, objects on water are frequently obscured by waves and sunlight glare, making the channel correlation between features especially crucial. ECA [34] is well known as a lightweight and effective channel attention module, which avoids dimensionality reduction. Meanwhile, appropriate cross-channel interaction enhances the representation ability of features. Therefore, this research introduces the ECA module into each basic convolutional block of DLA-60, which enhances detection performance by better capturing the salient characteristics of objects on water.

Furthermore, the Mish activation function replaces the original ReLU activation function. Mish is a smooth activation function that can help train deep neural networks and ease the gradient vanishing issues. As the depth of the network increases, the ReLU activation function reduces accuracy rapidly, whereas Mish's accuracy declines more slowly.

To verify the performance of the proposed detector, the DLA-60 network and the improved DLA-60 network are utilized to extract characteristics from ship targets individually,

as illustrated in Figure 5. The two sets of feature images on the right side were extracted from two different networks when they were processing the same image on the left. It is evident that the improved network performed better, which gives more details.

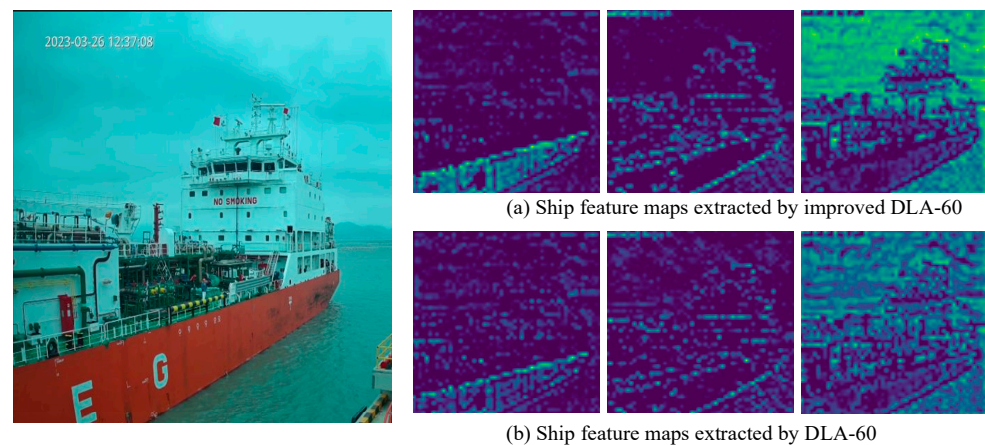


Figure 5. Comparison among two different networks.

3.2. The Ship Contour for Ship Segmentation

The contour segmentation is built on the precise target candidate boxes generated by the CenterNet detector. Object contour is a cycle graph consisting of a sequence of vertices connected in a closed cycle along the object silhouette. Given an initial contour, classic snake algorithms treat the vertices' coordinates as a set of variables and optimize an energy functional based on these variables. The energy function has two components: internal and external. The internal formula governs the contour convergence, while the external formula constrains the contour's shape to fit the real shape. Because the energy functional is often nonconvex and built using low-level image characteristics, the deformation process favors local optimal solutions. As a result, an active contour model is used to optimize contour deformation, as discussed at the beginning of Section 3 and shown in Figure 1. This research designs a feature extraction network using circular convolution. When combined with the snake algorithm, it uses a deep learning approach to calculate the offsets of the contour points, allowing for continual optimization of the initial contour in the direction of the desired border. We will next give a thorough rundown of our process.

Circular convolution is a particular form of one-dimensional convolution that is widely used on discrete signals. Circular convolution for feature learning is illustrated in Figure 6. On a contour, blue nodes indicate the input features, green nodes represent the kernel function, and orange nodes represent the output features. Circular convolution produces output features that are the same length as their input features. The features of contour vertices can be thought of as a 1-D discrete signal $f: \mathbb{Z} \rightarrow \mathbb{R}^D$. The contour features are defined as shown in Equation (1) and are handled as a periodic signal:

$$(f_N)_i \triangleq \sum_{j=-\infty}^{\infty} f_{i-jN} \quad (1)$$

They are also proposed to encode the periodic features using the circular convolution, which is described by Equation (2):

$$(f_N * k)_i = \sum_{j=-r}^r (f_N)_{i+j} k_j \quad (2)$$

where N is the number of contour vertices $\{x_i | i = 1, \dots, N\}$, k is a learnable kernel function, and the operator $*$ is the standard convolution.

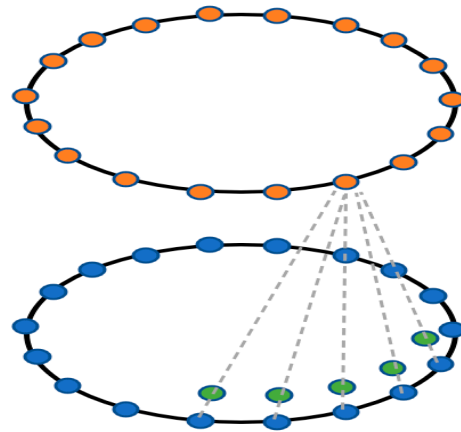


Figure 6. Circular convolution.

Given a contour with N vertices $\{x_i | i = 1, \dots, N\}$, we first generate feature vectors for each vertex. The input feature f_i for a vertex x_i is a concatenation of learning-based features and the vertex coordinate: $[F(x_i); x'_i]$, where F is the feature maps and x'_i is a translation-invariant version of vertex x_i . The feature maps F are created by applying a CNN backbone on the input image, which the deep snake shares with the detector in our instance segmentation algorithm. The image feature $F(x_i)$ is calculated by performing the bilinear interpolation of features at the vertex coordinate x_i . The appended vertex coordinate indicates the spatial relationship between contour vertices.

The contour vertices features were then fed into the segmentation network. As shown in Figure 7, the ship contour segmentation network is made up of three main components: a backbone, a feature fusion block, and a prediction head. The backbone is made up of eight “CirConv-Bn-ReLU” layers that use residual skip connections for all layers. “CirConv” stands for circular convolution. The fusion block seeks to combine information from all contour points at different scales. It concatenates features from all layers in the backbone and forwards them through a 1×1 convolution layer followed by max pooling. The fused feature is then concatenated with the feature of each vertex. The prediction head uses three 1×1 convolution layers to analyze vertex features and generate vertex-wise offsets.

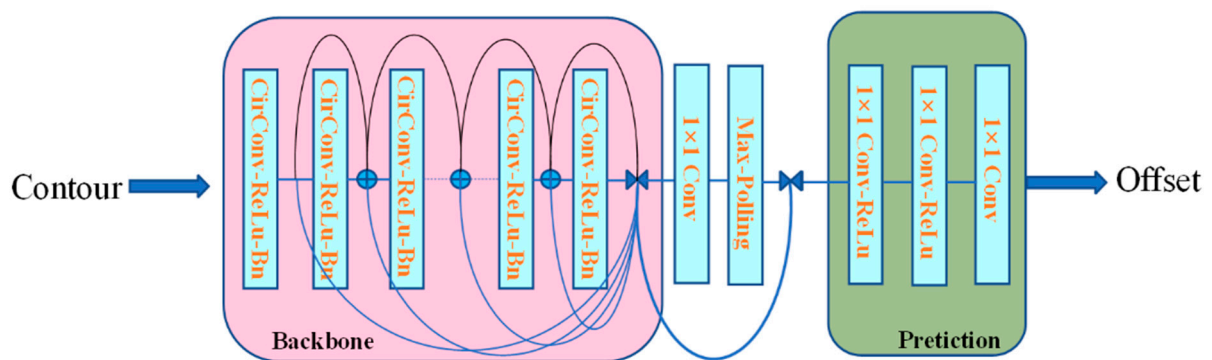


Figure 7. Ship Contour segmentation network architecture.

The deformation iteration of the original deep snake normalizes the contour by subtracting each vertex coordinate by the minimum value over all vertices on the contour to enable translation-invariant contour deformation, which is described by Equation (3).

$$x'_n = x_n - x_{\min}, \quad y'_n = y_n - y_{\min} \quad (3)$$

where (x_n, y_n) is the value of the vertex on the contour of the input backbone network, $(x_{\min}, y_{\min})$ represents minimum value over all vertices on the corresponding contour, and (x'_n, y'_n) is the value after this normalization.

However, the distance between the vertex and the target is not constant. For instance, when ships are of a large size, their vertices are frequently located far from the target place. It takes several iterations to reach the goal point due to the difficulty of learning translation-invariant normalization, which adds to the complexity of the model. Inspired by Liu et al. [35], this research makes use of a scale- and translation-invariant normalization scheme to ease this problem, which is expressed in Equation (4).

$$x'_n = \frac{x_n - x_{\min}}{x_{\max} - x_{\min}}, y'_n = \frac{y_n - y_{\min}}{y_{\max} - y_{\min}} \quad (4)$$

where $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ denote the minimum and maximum values over all vertices for a specific contour, respectively. The range of processed (x'_n, y'_n) values is constrained to $[0, 1]$; therefore, the resulting offset prediction should similarly be limited. As shown in Equation (5), a Tanh activation function is applied to reduce the offset inside $[0, 1]$, and the output is then multiplied by the width and height of the appropriate target. Specifically, the adjustment for width is illustrated in Equation (6), and the adjustment for height is shown in Equation (7):

$$\{(\Delta x_n, \Delta y_n)\}_{n=1}^N = \text{Tanh}(\text{Snake}(\{F(x_n, y_n)\}_{n=1}^N)) \quad (5)$$

$$x_n^{\text{new}} = x_n^{\text{old}} + (x_{\max} - x_{\min}) * \Delta x_n \quad (6)$$

$$y_n^{\text{new}} = y_n^{\text{old}} + (y_{\max} - y_{\min}) * \Delta y_n \quad (7)$$

where $(\Delta x_n, \Delta y_n)$ is the offset of contour vertices calculated through a segmentation network, $F(x_n, y_n)$ is the feature map of contour vertices, and N is the number of contour vertices. $(x_n^{\text{old}}, y_n^{\text{old}})$ represents the contour vertex input into the segmentation network, whereas $(x_n^{\text{new}}, y_n^{\text{new}})$ represents the output by overlaying the input value with the segmentation network's offset.

The improved contour deformation method addresses the challenging regression of contour vertices caused by their separation from the target and is suitable for training ship datasets. As discussed above, contour deformation will be performed three times to assure contour quality, while extreme point prediction loss and contour vertex deformation loss will be utilized to constrain the original contour so that it gradually approaches the goal contour. The extreme point loss function is represented as Equation (8).

$$L_{\text{ex}} = \frac{1}{4} \sum_{i=1}^4 \text{Smooth}_{l1}(y_i^{\text{ex}} - y_i^{\text{ex}}) \quad (8)$$

where Smooth_{l1} is a smooth loss function proposed in Fast R-CNN [36] for regression tasks. Figure 8 illustrates the process where predicted contour points (represented in yellow) move closer to the ground truth contour points (represented in green) under the influence of the Smooth L1 loss function. This loss function can quickly converge when the predicted contour point error is considerable in the beginning of training process and has strong robustness when the gradient changes slowly in the middle and later stages of training. y_i^{ex} is the predicted extreme point coordinate, and y_i^{ex} is the coordinates of the labeled extreme

points. The average loss value is utilized because there are 4 extreme points. The loss of contour vertices is defined as Equation (9).

$$L_{iter} = \frac{1}{N} \sum_{i=1}^N Smooth_{l1} (y'_i - y_i^{gt}) \quad (9)$$

where y'_i is the predicted contour vertex coordinate, y_i^{gt} is the ground-truth boundary coordinate. N is the number of sampling points, which is fixed at 128 in order to accommodate the majority of target shapes.

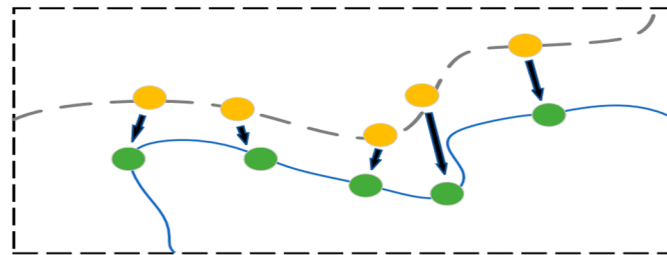


Figure 8. Contour points regression using Smooth L1 loss.

4. A Case Study

4.1. A Dataset for Ship Instance Segmentation

The performance of ship segmentation is heavily dependent on the quality and size of the dataset. Currently, there are not many ship images in publicly accessible datasets like ImageNet [37] and COCO [38], nor are there sufficient kinds of ships included in these images. The images typically show ships unhindered, which contrasts sharply with actual locations like harbors and ocean surfaces. These samples limit the model's capacity to generalize to other types of ships, reducing its performance in real-world circumstances when using them for model training.

This research establishes a real-world ship dataset called “2023 Ship-seg”. A total of $2300 \times 512 \times 512$ images are selected and labeled carefully, including ferries, cargo ships, and other vessels of different sizes. These images are taken from surveillance cameras around inland and coastal rivers, including Zhoushan in Zhejiang, Nanjing Port on the Yangtze River waterway, and Banqiao Ferry Wharf in Jiangsu. This dataset incorporates a range of weather conditions, including sunny, overcast, and foggy days, as well as different times of day, such as dawn, dusk, and night, to ensure data diversity. The images have a variety of backgrounds, including navigational structures, buoys, reefs, and so on. Figure 9 shows some images from the dataset, and Table 1 presents the details of the dataset. Finally, frames are extracted from the captured sample videos, images with a high number of diverse ship instances are selected, and the Labelme v5.2.1 tool is used to meticulously annotate these images. In particular, many images with obscured ships are selected to guarantee stability and generalization.

However, it is worth noting that the dataset still has some limitations. Currently, our collaborations are limited to a few waterways and ports, which means the dataset cannot cover all scenarios. Additionally, since most of the samples come from surveillance systems, the perspective in the dataset is primarily that of an overview, with only a small portion being real-time shots taken from cruise ships. In the future, we plan to continuously expand the dataset, especially by introducing more data from the perspective of the cockpit, to enhance the diversity and representativeness of the dataset.

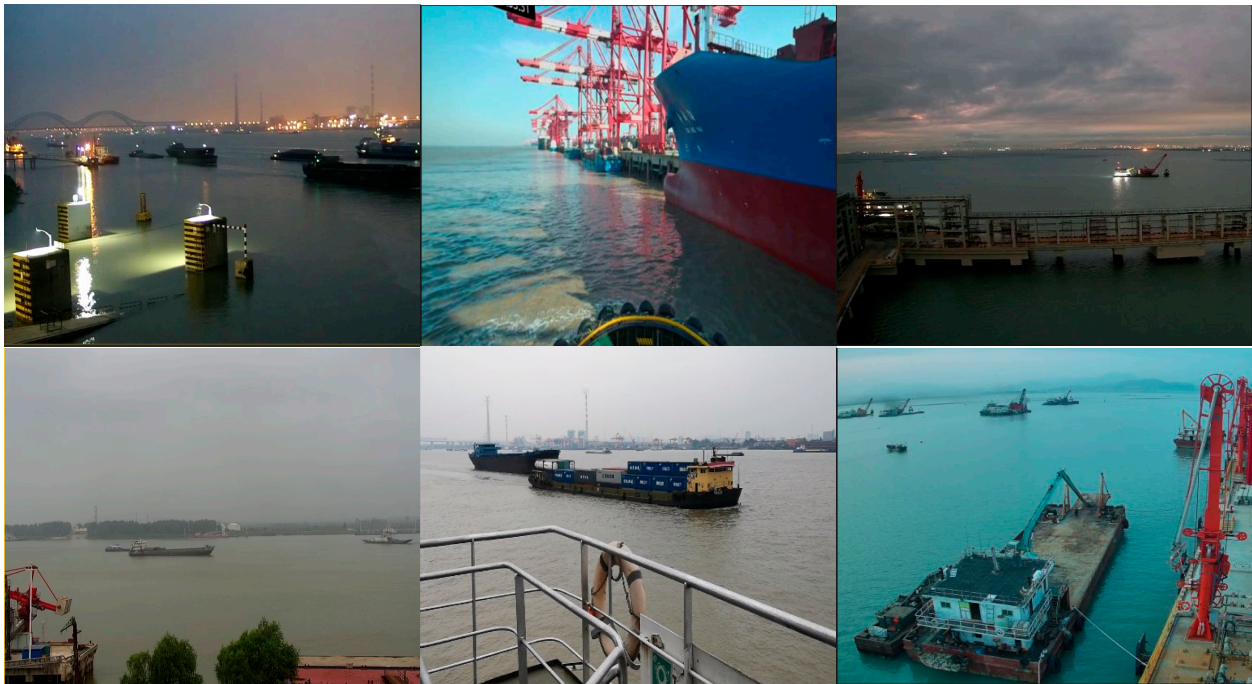


Figure 9. Examples of the 2023 Ship-seg dataset.

Table 1. The detailed information of the 2023 Ship-seg dataset.

Dataset	2023 Ship-seg
Images	2300
Train; Instances	1610; 4392
Test; Instances	690; 1867
Number of the boat instances	6259
Size	512
Type	Visual images
Task	Instance segmentation

4.2. An Experimental Platform

To verify the effectiveness of the proposed method, the PyCharm platform was used to build an experimental platform with Pytorch framework. Versions 11.4 and 11.6 of CUDA and cudnn are equivalent. NVIDIA GeForce RTX3090 was the computational power acceleration unit, and Python 3.7 was the coding language. With 32GB of RAM, an Intel Core i7-12700K PC was used for training and testing. The 2023-ships dataset was split into training and testing sets in a 7:3 ratio. The size of training image was set to 512 × 512, the batch size was 16, and the whole training lasted 300 rounds. An Adam optimizer was used for training, with a starting learning rate of 0.0001 and a momentum of 0.9.

4.3. Comparison with State-of-the-Art Methods

The ship instance segmentation performance of Ship Contour and the most advanced instance segmentation algorithms—Yolov8, Segformer, Solov2, Yolact++, U-Net, and deep snake—are compared in this research. Our primary emphasis is on the measures of average precision (AP) and average recall (AR). In addition, as Table 2 illustrates, the model’s complexity was calculated using giga floating-point operations per second (Gflops) and parameter quantity (Param).

Table 2. Quantitative comparison results.

	Backbone	AP _{0.5} (%)	AP _{0.5:0.95} (%)	AR _{0.5} (%)	AR _{0.5:0.95} (%)	Gflops	Param
Yolov8	Darknet-53	0.934	0.617	0.956	0.653	42.7 M	11.7
Solov2	Resnet-101	0.940	0.575	0.953	0.653	62.1 M	65.0
U-net	Vgg-16	0.960	-	0.96	-	226.1 M	24.9
Segformer	MiT-B2	0.913		0.931		113.4 M	27.3
Yolact++	Resnet-101	0.912	0.58	-	-	42.1 M	21.3
Deep Snake	DLA-34	0.931	0.618	0.955	0.664	25.9 M	16.3
Ship Contour	DLA-60	0.944	0.636	0.965	0.674	42.3 M	23.5

As shown in Table 2, Ship Contour performed better than other methods. Specifically, Ship Contour was 3.2% higher than Yolov8 as measured by AR_{0.5} indicator. The AP_{0.5:0.95} evaluation metric of Ship Contour model was significantly higher than that of Solov2, Yolact++ and Segformer algorithms. Although the AP_{0.5} metric of the Ship Contour model was 1.6% lower than that of U-Net, the efficiency of U-Net was extremely inefficient in real-time segmentation tasks. Because of its tremendous computational complexity, U-Net demands a huge amount of processing resources. Additionally, the segmentation results from U-Net often exhibit rough edges, whereas Ship Contour provides a more natural effect. This meticulous edge processing makes Ship Contour more appealing in practical applications. With the exception of segmentation efficiency, the Ship Contour approach performed better than the deep snake algorithm in every other metric. Specifically, due to the use of a deeper feature extraction network, DLA-60, the inference efficiency of Ship Contour decreases by approximately 32% in practical applications. However, this trade-off is worthwhile given the significant improvement in detection accuracy, particularly for applications that prioritize high-quality segmentation results. Yolact++ performs well in general environments, but its segmentation effectiveness significantly decreases in cases where two or more ships occlude each other. Our dataset contains many such occlusion scenarios, which leads to a notable decline in Yolact++'s segmentation accuracy. In contrast, Ship Contour shows more stable performance in the face of occlusions, demonstrating less susceptibility to their impact. It is worth mentioning that Mask R-CNN also faces similar issues. Despite demonstrating excellent performance in many tasks, particularly in object detection, Mask R-CNN tends to produce masks with noticeable gaps or discontinuities in instance segmentation tasks when there are occlusions and overlaps. When multiple objects overlap, Mask R-CNN may fail to correctly assign pixels to different instances, resulting in inaccurate segmentation outcomes.

To further validate the performance of Ship Contour, we compared the proposed method with the deep snake, Yolov8, and Segformer methods on the “2023 Ship-seg” dataset. Figure 10 compares the effects of ship instance segmentation between Ship Contour and deep snake in complex ports with low light and fog. Figure 10a shows how Ship Contour can distinguish and segment ship targets that are visible to the human eye. Deep snake only recognized ship targets with apparent characteristics, and the segmentation effect on remote targets was poor. Deep snake detected tugboats as two targets, as seen in Figure 10b. In dark and foggy conditions, Ship Contour distinguished more targets and provided greater accuracy in ship segmentation than deep snake, as shown in Figure 10c,d. While Ship Contour operated at a lower reasoning frame rate of 30 fps compared to deep snake's 47 fps, it achieved a significant enhancement in segmentation accuracy, particularly in challenging scenarios. This improvement demonstrated that the trade-off was justified.

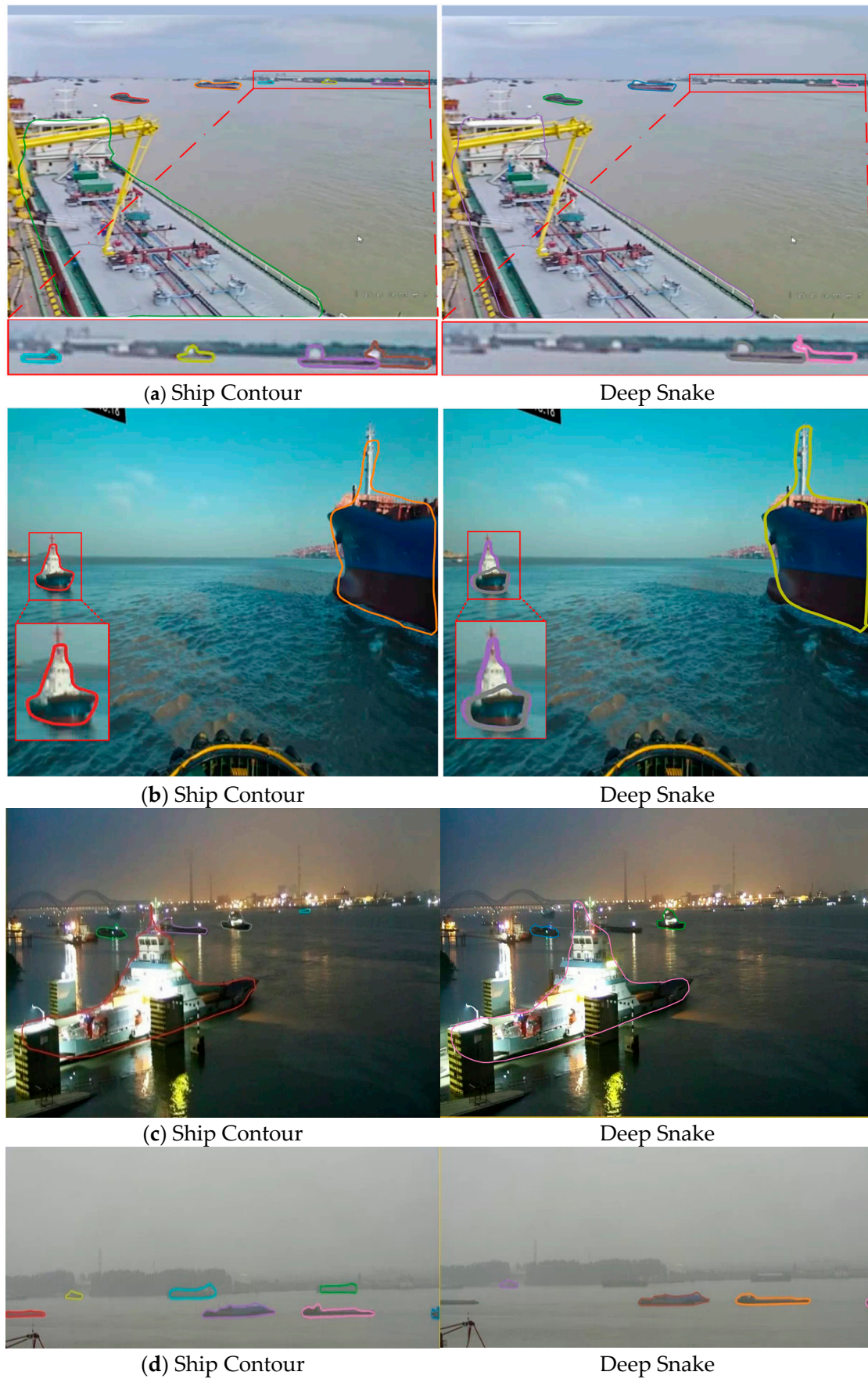


Figure 10. Performance comparison between Ship Contour and deep snake.

As elaborated previously, Ship Contour is a contour-based segmentation strategy, whereas YOLOv8 is a mask-based segmentation approach. In order to make the comparison between the two methods more evident, the DrawContours module in OpenCV is used to treat the outer border of the YOLOv8-trained mask as the target contour, as seen in Figure 11. YOLOv8 achieves an inference speed of 75 fps, significantly outperforming the proposed method in efficiency. However, it is important to note that contour-based segmentation methods typically consume more resources during the inference process compared to mask-based approaches. Consequently, when performing video inference with YOLOv8 and subsequently drawing ship target contours using the DrawContours method, its inference speed decreased to 31 fps, aligning closely with the performance of Ship Contour.

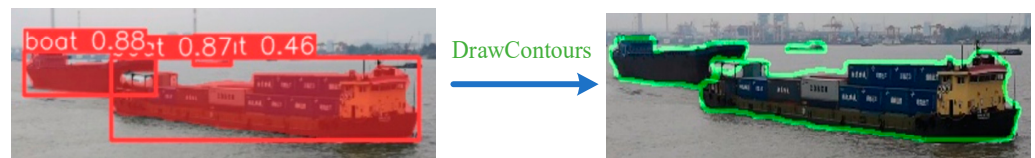


Figure 11. A schematic diagram of YOLOv8 contour segmentation.

As illustrated in Figure 12, the two methods produced similar results for ship target segmentation when ships were close. However, in terms of the segmentation impact of long-distance targets, the contour provided by Ship Contour was smoother and more in line with the actual ship contour detected by human eyes. In the medium distance object segmentation application scenario, such as ship berthing, YOLOv8 exhibits object segmentation mistakes, while ship contour achieved satisfactory results.

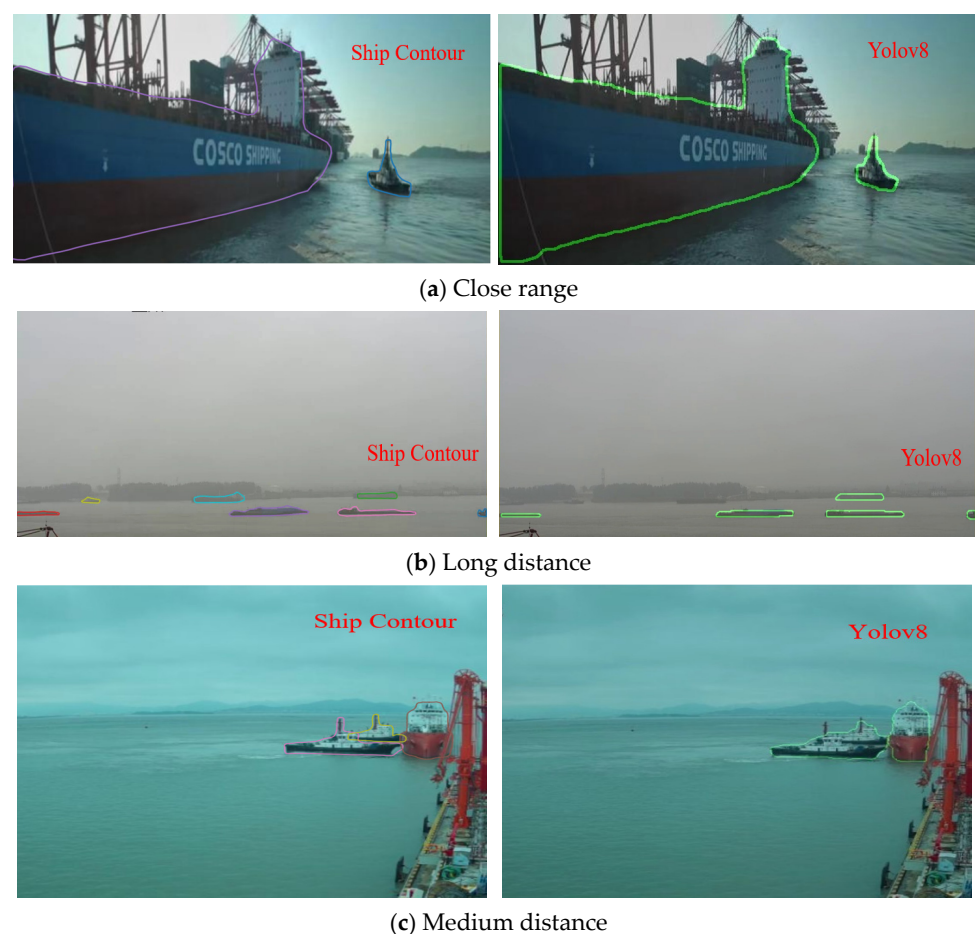


Figure 12. Comparison between Ship Contour and YOLOv8.

SegFormer is a transformer-based segmentation model, widely recognized as a state-of-the-art (SOTA) algorithm. Unlike traditional convolutional methods, SegFormer utilizes transformer layers to capture both global and local information, allowing it to better adapt to complex scenarios. The model consists of an encoder and a decoder. The encoder employs a hierarchical transformer module to extract high-resolution shallow features and low-resolution fine features, while the decoder integrates these features using a multilayer perceptron (MLP) to generate the final segmentation results.

To compare the performance between transformer-based and traditional convolutional segmentation methods, SegFormer was evaluated on the 2023 Ship-seg dataset. SegFormer achieved an inference speed of 28 fps, which is slightly lower than the frame rate of Ship Contour. The detailed performance is shown in Figure 13.

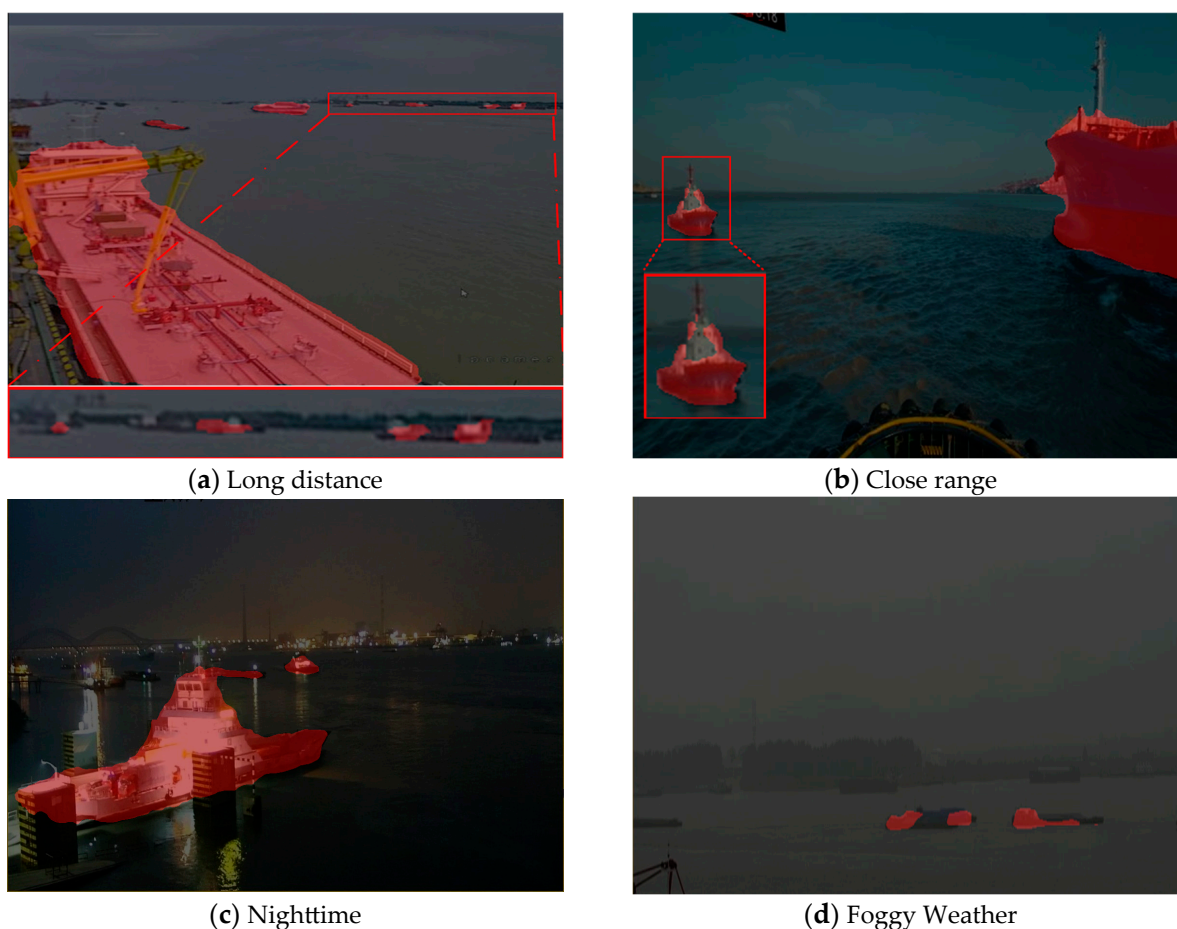


Figure 13. Performance of Segformer on the 2023 Ship-seg dataset.

The subfigures (a)–(d) in Figure 13 correspond to the four scenarios presented in Figure 10. As shown, SegFormer performs excellently in segmenting medium and large-sized objects, while the segmentation of small objects is less effective. Additionally, there are significant issues with segmentation under nighttime and foggy conditions.

4.4. Ablation Study

In order to evaluate the performance of each component of Ship Contour, an ablation study was conducted. Specifically, the various components of this research were gradually removed, and their performance were tested on the “2023 Ship-seg” dataset. The ablation study findings are reported in Table 3. The model M0 denotes the baseline model, which is the traditional deep snake. Model M1 is an augmented baseline model that employs an improved CenterNet detector, denoted as C-Net+. The model M1 AR_{0.5:0.95} shows a

1% increase in recall rate and a 1.8% improvement in the accuracy index $AP_{0.5:0.95}$, which enhances baseline model segmentation accuracy and lowers the rate of missed identification of hard-to-detect targets in complex situations. Comparing this straightforward change to the original one, the accuracy increased, and the missed detection rate decreased. On the basis of the model M0, a contour deformation optimization (CDO) model is introduced, denoted as M2, which has a superior segmentation ability for non-concave shaped ship targets. The accuracy of the CDO model $AP_{0.5:0.95}$ rises by 0.6%, $AP_{0.5}$ increases by 1%, and the other two indicators yields improvements. Model M3 integrates both the CDO model and the C-Net+ model, which outperforms the baseline model (M0) by 1.8% in accuracy $AP_{0.5:0.95}$ and 1.0% in recall rate $AR_{0.5:0.95}$. Additionally, M3 performs admirably, with an $AP_{0.5}$ of 94.4% and an $AR_{0.5}$ of 96.5%. The results show that the task of ship instance segmentation has been successfully completed by the suggested improvements.

Table 3. Ablation results.

Method	C-Net+	CDO	$AP_{0.5}$ (%)	$AP_{0.5:0.95}$ (%)	$AR_{0.5}$ (%)	$AR_{0.5:0.95}$ (%)
M0	×	×	0.931	0.618	0.955	0.664
M1	✓	×	0.942	0.634	0.965	0.671
M2	×	✓	0.941	0.624	0.964	0.665
M3	✓	✓	0.944	0.636	0.965	0.674

4.5. Experimental Results and Analysis Based on Public Datasets

To further validate the effectiveness of the proposed approach, segmentation experiments were conducted on the publicly available MariBoats dataset [39] and the COCO-Boat dataset.

The MariBoats dataset comprises 6200 images, with all the ships labeled under a single category, namely ‘ship’, resulting in 15,777 ship segmentation annotations. The experimental results are shown in Table 4, where the experimental parameter settings for Solov2, deep snake, and Ship Contour remained consistent with those described above. The findings demonstrated that Ship Contour struck a balance between performance and efficiency, showing significant improvements compared to deep snake. Specifically, its $AP_{0.5:0.95}$ and $AR_{0.5:0.95}$ scores were both more than 2% higher. Compared to Solov2, Ship Contour exhibited advantages across all evaluated metrics. Figure 14 illustrates the segmentation results of the proposed method on the MariBoats dataset.

Table 4. Experimental results using MariBoats dataset.

Method	Backbone	$AP_{0.5}$ (%)	$AP_{0.5:0.95}$ (%)	$AR_{0.5}$ (%)	$AR_{0.5:0.95}$ (%)	FPS
Solov2	Resnet-101	0.869	0.573	0.911	0.642	26
Deep Snake	DLA-34	0.865	0.590	0.913	0.644	47
Ship Contour	DLA-60	0.877	0.613	0.917	0.665	30

In addition, experiments were conducted on the COCO-Boat dataset, a subset extracted from the COCO dataset containing samples with boat instances. This dataset includes over 3146 images, covering a diverse range of boat instances under varying conditions.

The Ship Contour model achieved an $AP_{0.5}$ of 40.8% and an $AR_{0.5}$ of 56.1% on the COCO-Boat validation set. Figure 15 illustrates the segmentation results of the proposed method on the COCO dataset.



Figure 14. The segmentation results of the proposed method in the MariBoats datasets.

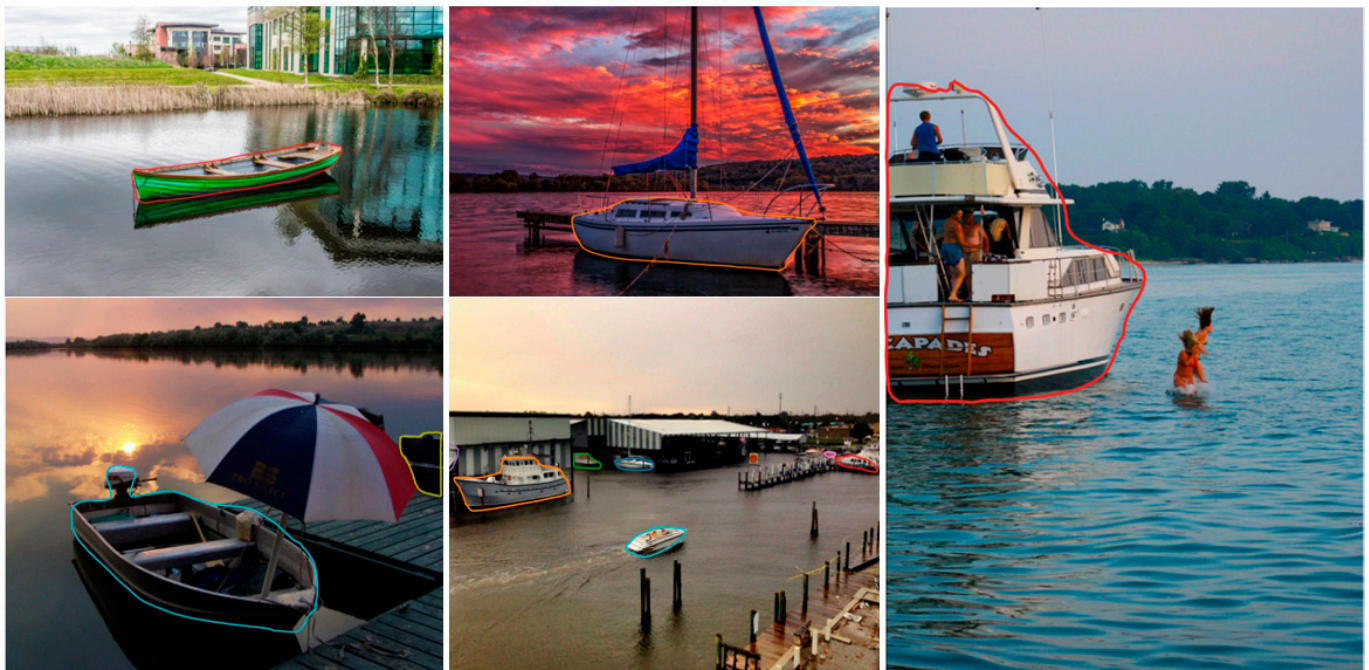


Figure 15. The segmentation results of the proposed method using the COCO-Boat dataset.

5. Conclusions

This paper proposed a real-time algorithm called Ship Contour in order to address the problem of ship instance segmentation in complex environments. To verify the effectiveness of this method, various typical water sceneries were gathered, and a ship segmentation dataset of over 2300 images was constructed, named 2023 Ship-seg. Ship Contour outperforms on this dataset, with an accuracy rate of $AP_{0.5:0.95}$ up to 63.6% and a recall rate of $AR_{0.5:0.95}$ reaching 67.4%, and all other metrics are also improved. Ship Contour outperforms deep snake and Yolov8 in the detection and segmentation of ships under a variety of interference circumstances, including bad weather, lighting situations such as fog and

night, ship berthing and departure, extreme scenarios, and difficult interference settings. Furthermore, Ship Contour also optimizes the contour deformation problem of the deep snake algorithm, resolving the issue of challenging contour regression in the convex area of the bow when dealing with large target ships with complex geometries. Especially, it can achieve a segmentation efficiency of 30 fps per seconds on the RTX3090 graphics card. Above all, the approach proposed in this paper improves segmentation performance in complex water navigation scenarios with real-time capabilities.

In the future, research will continue to increase the size of datasets, gather more images of watercraft, and thoroughly assess the efficacy of algorithms to keep up with the constantly evolving needs of water navigation. Ship contour is a practical application that offers robust assistance with activities pertaining to port safety, marine safety, and ship inspection, and it can be applied in multiple critical fields.

Specifically, the technology can be used for vessel traffic management, helping authorities develop effective navigation strategies through real-time monitoring and recognition of vessels in waters, thereby reducing collision risks. Additionally, the method can be used for automated channel monitoring by precisely segmenting obstacles in channels, ensuring safe and clear waterways. In terms of port management, Ship Contour can integrate with AIS systems to monitor the positions and statuses of vessels in real time, assisting in berthing and departure operations, thus enhancing the efficiency and safety of port operations.

Author Contributions: Conceptualization, C.C. and F.M.; Data curation, S.H.; Formal analysis, T.L. and B.W.; Investigation, C.C.; Methodology, S.H. and F.M.; Project administration, C.C. and T.L.; Resources, J.S.; Software, S.H. and J.S.; Supervision, C.C. and F.M.; Validation, F.M.; Writing—original draft, S.H.; Writing—review and editing, C.C. and F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is financially supported by the National Natural Science Foundation of China under Grants No. 52201415 and 52171352, the National Key R&D Program of China (Grant No. 2023YFB4302300), and the Graduate Innovative Fund of Wuhan Institute of Technology under Grant No. CX2023285, Fund of State Key Laboratory of Maritime Technology and Safety (No. 16-10-1).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: Author Jie Sun was employed by the company Nanjing Smart Water Transportation Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Petković, M.; Vujović, I. Distance Estimation Approach for Maritime Traffic Surveillance Using Instance Segmentation. *J. Mar. Sci. Eng.* **2024**, *12*, 78. [\[CrossRef\]](#)
2. Murthy, A.S.D.; Rao, S.K.; Naik, K.S.; Das, R.P.; Jahan, K.; Raju, K.L. Tracking of a Manoeuvring Target Ship Using Radar Measurements. *Indian J. Sci. Technol.* **2015**, *8*, 1. [\[CrossRef\]](#)
3. Kim, H.; Kim, D.; Park, B.; Lee, S.-M. Artificial Intelligence Vision-Based Monitoring System for Ship Berthing. *IEEE Access* **2020**, *8*, 227014–227023. [\[CrossRef\]](#)
4. Eum, H.; Bae, J.; Yoon, C.; Kim, E. Ship Detection Using Edge-Based Segmentation and Histogram of Oriented Gradient with Ship Size Ratio. *Int. J. Fuzzy Log. Intell. Syst.* **2015**, *15*, 251–259. [\[CrossRef\]](#)
5. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9156–9165.

6. Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; Li, L. SOLO: Segmenting Objects by Locations. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XVIII. Springer: Berlin/Heidelberg, Germany, 2020; pp. 649–665.
7. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850v2.
8. Yu, F.; Wang, D.; Shelhamer, E.; Darrell, T. Deep Layer Aggregation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2403–2412.
9. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
10. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. *arXiv* **2019**, arXiv:1908.08681.
11. Peng, S.; Jiang, W.; Pi, H.; Li, X.; Bao, H.; Zhou, X. Deep Snake for Real-Time Instance Segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8530–8539.
12. Zheng, X.; Wang, H.; Shang, Y.; Chen, G.; Zou, S.; Yuan, Q. Starting from the Structure: A Review of Small Object Detection Based on Deep Learning. *Image Vis. Comput.* **2024**, *146*, 105054.
13. Duan, H.; Ma, F.; Miao, L.; Zhang, C. A Semi-Supervised Deep Learning Approach for Vessel Trajectory Classification Based on AIS Data. *Ocean Coast. Manag.* **2022**, *218*, 106015. [[CrossRef](#)]
14. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
16. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
17. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [[CrossRef](#)] [[PubMed](#)]
18. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
19. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask Scoring R-CNN. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6402–6411.
20. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. SOLOv2: Dynamic and Fast Instance Segmentation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17721–17732.
21. Yang, Q.; Peng, J.; Chen, D.; Zhang, H. Road Scene Instance Segmentation Based on Improved SOLOv2. *Electronics* **2023**, *12*, 4169. [[CrossRef](#)]
22. Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active Contour Models. *Int. J. Comput. Vis.* **1988**, *1*, 321–331. [[CrossRef](#)]
23. Marcos, D.; Tuia, D.; Kellenberger, B.; Zhang, L.; Bai, M.; Liao, R.; Urtasun, R. Learning Deep Structured Active Contours End-to-End. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8877–8885.
24. Yang, Z.; Xu, Y.; Xue, H.; Zhang, Z.; Urtasun, R.; Wang, L.; Lin, S.; Hu, H. Dense RepPoints: Representing Visual Objects with Dense Point Sets. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer: Cham, Switzerland, 2020; Volume 12366, pp. 227–244.
25. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 3992–4003.
26. Wang, X.; Zhang, X.; Cao, Y.; Wang, W.; Shen, C.; Huang, T. SegGPT: Segmenting Everything In Context. *arXiv* **2023**, arXiv:2304.03284.
27. Sun, Y.; Su, L.; Luo, Y.; Meng, H.; Li, W.; Zhang, Z.; Wang, P.; Zhang, W. Global Mask R-CNN for Marine Ship Instance Segmentation. *Neurocomputing* **2022**, *480*, 257–270. [[CrossRef](#)]
28. Huang, Z.; Li, R. Orientated Silhouette Matching for Single-Shot Ship Instance Segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 463–477. [[CrossRef](#)]
29. Ma, F.; Kang, Z.; Chen, C.; Sun, J.; Deng, J. MrisNet: Robust Ship Instance Segmentation in Challenging Marine Radar Environments. *J. Mar. Sci. Eng.* **2023**, *12*, 72. [[CrossRef](#)]
30. Zhang, Y.; Li, C.; Shang, S.; Chen, X. SwinSeg: Swin Transformer and MLP Hybrid Network for Ship Segmentation in Maritime Surveillance System. *Ocean. Eng.* **2023**, *281*, 114885. [[CrossRef](#)]

31. Sun, Y.; Su, L.; Luo, Y.; Meng, H.; Zhang, Z.; Zhang, W.; Yuan, S. IRDCLNet: Instance Segmentation of Ship Images Based on Interference Reduction and Dynamic Contour Learning in Foggy Scenes. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6029–6043. [[CrossRef](#)]
32. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
33. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable Convnets V2: More Deformable, Better Results. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9300–9308.
34. Yue, Z.; Yanyan, F.; Shangyou, Z.; Bing, P. Facial Expression Recognition Based on Convolutional Neural Network. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, Beijing, China, 18–20 October 2019; pp. 410–413.
35. Liu, Z.; Liew, J.H.; Chen, X.; Feng, J. DANCE: A Deep Attentive Contour Model for Efficient Instance Segmentation. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 345–354.
36. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
37. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
38. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the Computer Vision—ECCV 2014: 13th European conference, Zurich, Switzerland, 6–12 September 2014; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer: Cham, Switzerland, 2014; Volume 8693, pp. 740–755.
39. Sun, Z.; Meng, C.; Huang, T.; Zhang, Z.; Chang, S. Marine Ship Instance Segmentation by Deep Neural Networks Using a Global and Local Attention (GALA) Mechanism. *PLoS ONE* **2023**, *18*, e0279248. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.