

Article

Fast and Intelligent Ice Channel Recognition Based on Row Selection

Wenbo Dong ¹, Li Zhou ² , Shifeng Ding ^{1,*}, Qun Ma ¹ and Feixu Li ¹

¹ School of Naval Architecture and Ocean Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China; 209010076@stu.just.edu.cn (W.D.); maxund@163.com (Q.M.); 211210101408@stu.just.edu.cn (F.L.)

² School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; zhoul209@sjtu.edu.cn

* Correspondence: 201800000060@just.edu.cn

Abstract: The recognition of ice channels plays a crucial role in developing intelligent ship navigation systems in ice-covered waters. Navigating through ice channels with the assistance of icebreakers is a common operation for merchant ships. Maneuvering within such narrow channels presents a significant challenge for the captain's skills and ship performance. Therefore, it becomes essential to explore methods for enabling ships to navigate through these channels automatically. A key step in achieving this is the accurate recognition and extraction of boundary lines on both sides of the ice channel. An ice channel line recognition method based on the lane line detection algorithm UFAST is implemented. The method is trained and tested on the constructed ice channel dataset, with the test results showing that the average recognition accuracy reaches 84.1% and the recognition speed reaches 138.3 frames per second, meeting the real-time requirement. In order to solve the current lack of authentic ice channel images, ice channel navigation scenes are built based on UE4, and synthetic ice channel images are rendered. The method in this paper is also compared with the traditional non-intelligent Otsu threshold segmentation method and the intelligent instance segmentation method YOLACT for performance analysis. The method in this paper has 9.5% higher ice channel recognition accuracy and 103.7 frames per second higher recognition speed compared with YOLACT. Furthermore, ablation studies are conducted to analyze the relationship between the number of gridding cells in the proposed method and ice channel recognition accuracy.



Citation: Dong, W.; Zhou, L.; Ding, S.; Ma, Q.; Li, F. Fast and Intelligent Ice Channel Recognition Based on Row Selection. *J. Mar. Sci. Eng.* **2023**, *11*, 1652. <https://doi.org/10.3390/jmse11091652>

Academic Editor: Claudio Ferrari

Received: 27 July 2023

Revised: 17 August 2023

Accepted: 21 August 2023

Published: 24 August 2023



Copyright: © 2023 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/>).

Keywords: polar ships; intelligent navigation; ice channel; recognition; synthetic dataset; artificial intelligence

1. Introduction

In the past few years, the impact of global warming and the rapid melting of sea ice has led to a growing recognition of the potential value of exploiting the oil, gas, and waterway resources in polar regions. Advancements in science and technology have facilitated increased navigation through the Arctic waterway, presenting new demands and areas of research for the development, utilization, safety, and security of these routes [1]. Unmanned ships have emerged as a crucial component in global trade and marine exploration, representing the next stage in the ongoing advancement of the shipbuilding and navigation industry. However, when it comes to studying vision-based driving technology for ships in icy areas, there are numerous challenges that set them apart from inland waterway vessels and traditional sea vessels [2,3]. It is a frequent occurrence for merchant ships to navigate through ice channels or follow icebreakers. The task of maneuvering a large and inertia-laden ship in a narrow channel presents a challenge for the captain's maneuvering skills, as well as the ship's performance [4,5]. The foundation for achieving automated ship piloting in ice channels lies in identifying the boundaries of the channel from the ship's perspective and creating a digital model of the channel, along with tracking the target

ship ahead. By leveraging computer vision technology, specifically image recognition, the channel can be automatically extracted. Advancements in recognition and extraction technologies for ice channels can greatly contribute to the development and utilization of Arctic routes, enhancing navigation efficiency and safety [6].

In this paper, an ice channel line recognition method based on the lane line detection algorithm UFAST is implemented. The method is trained and tested on the constructed ice channel dataset. The ice channel dataset consists of real ice channel images and synthetic ice channel images. In order to solve the current lack of authentic ice channel images, ice channel navigation scenes are built based on UE4, and synthetic ice channel images are rendered. The method in this paper is also compared with the traditional non-intelligent Otsu threshold segmentation method and the intelligent instance segmentation method YOLACT for performance analysis. Furthermore, ablation studies are conducted to analyze the relationship between the number of gridding cells in the proposed method and ice channel recognition accuracy, and explanations are provided for the observed phenomenon.

2. Related Work

At present, the focus of onboard vision-based driving technology primarily revolves around target recognition and tracking of ships or obstacles. However, there is limited research on identifying sea ice within ice fields. Liu et al. introduced a ship detection algorithm based on YOLO V5, which demonstrated improved detection accuracy compared to the original YOLO V5 algorithm. Additionally, the proposed method resulted in a steady decrease in GIoU (Generalized Intersection over Union) values [7]. In a separate study, Wu et al. developed a deep learning-based multi-object tracking algorithm specifically for ships [8]. The results indicated a 2.23% improvement in tracking accuracy, while maintaining an average processing speed of approximately 21 frames per second, meeting the requirements for real-time tracking applications. Another research by Liu et al. focused on ship tracking and recognition [9]. They proposed a novel approach utilizing the Darknet network model and the YOLOv3 algorithm, enabling real-time ship tracking and detection as well as ship type recognition. Experimental results showcased an average recognition accuracy of 89.5% at a speed of 30 frames per second.

In the domain of image recognition involving ice-related objects, Lu et al. employed the gradient vector flow technique to process aerial images [10]. Their study focused on extracting broken ice within ice area channels and exploring the relationship between the width of icebreaker-induced fractures and the breakup of floating ice. Similarly, Cai et al. utilized an image segmentation method to segment instances of broken ice and accurately fit the parameters of individual ice blocks [11]. Their approach enabled precise analysis and characterization of the fragmented ice. Furthermore, Panchi et al. proposed a three-stage method for automated analysis of close-range optical images encompassing various ice formations [12]. Their method facilitated the recognition and classification of icebergs, deformed ice, level ice, broken ice, ice floes, floebergs, floebits, pancake ice, and brash ice, enabling comprehensive analysis of different ice types within the images.

Lane line detection has been extensively studied and successfully implemented in the field of smart cars [13]. This technology has proven to be beneficial for enhancing the capabilities of autonomous vehicles. It is worth noting that the task of detecting lane lines and identifying channels in ice areas share similarities in terms of their objectives and challenges. Traditional methods for lane line detection primarily rely on detecting the feature differences between the lane lines and the road surface. These methods utilize threshold segmentation techniques to separate the lane line features from the road [14]. However, these algorithms are sensitive to noise, lack robustness, and are susceptible to partial occlusion and lane line breakage [15]. In contrast, lane line detection methods based on deep learning can be categorized into two main approaches: Segmentation-based methods and row-based selecting methods. Segmentation methods aim to classify each pixel in the input image as either a lane line or background. Pan et al. extended the traditional convolutional neural network approach to a slice-by-slice convolutional

approach, enabling pixel-level signaling between rows and columns [16]. This method is particularly effective for detecting lane line structures with continuous and elongated shapes, demonstrating significant performance in lane line detection tasks. On the other hand, the row-based selecting method is a straightforward algorithm that performs raster segmentation for road recognition. For each segmented row, this method selects the grid that is most likely to contain a portion of the lane markings. These methods combine the advantages of high accuracy and speed. Yoo et al. employed the row-based selecting approach for lane detection, treating the task as identifying the specific location of each lane in each row [17]. They achieved an accuracy of 74% on the challenging CULane dataset. Lee et al. introduced a lightweight UNet architecture called DSUNet, which utilizes depthwise separable convolutions, for the purpose of end-to-end learning of lane detection and path prediction in autonomous driving [18]. The experimental results demonstrate that DSUNet proves to be efficient and effective in accurately detecting lanes and predicting paths in autonomous driving scenarios.

In general, there is limited research specifically targeting ice area channels, and the recognition of broken ice poses a more complex and intricate task. Therefore, the recognition of ice area channels in this study holds significant importance. Lane line detection, on the other hand, has matured in its development, and its related algorithms can serve as valuable references for the algorithms presented in this paper.

3. Dataset

Due to the method used being supervised deep learning, it is necessary to collect a large number of ice channel images captured from the perspective of the ship's first-perspective view and annotate them to form an ice channel dataset, enabling the completion of training for this method. Datasets or images are crucial for supervised learning. They provide training samples and labels, which are used to build models and make predictions. By observing a large number of samples, models can learn the underlying patterns between input features and output labels. Additionally, datasets contain valuable information that can be used to extract meaningful features, enhancing the model's understanding and accuracy. Moreover, datasets are used to evaluate the performance of models and select the best one. By splitting the dataset into training and testing sets, we can assess how well the model generalizes to unseen data and make improvements accordingly. Furthermore, data augmentation techniques can generate new samples by applying transformations such as rotation, scaling, etc., thereby increasing sample quantity and diversity and improving the model's robustness and generalization capabilities.

The ice channel dataset consists of three parts, as shown in Figure 1. The first part consists of real channel images, which are authentic photographs of ice channels. This part is collected from various sources and comprises a small quantity of 33 images, accounting for 6.5% of the dataset. The second part also consists of real ice channel images, but these images are obtained from the "Brash Ice Tests for a Panmax Bulker" experiment conducted by HSVA [19]. This part contains a smaller quantity of 23 images, representing 4.6% of the dataset. In order to supplement the quantity of the dataset and facilitate the smooth training and testing of methods, an ice channel scene was constructed based on the UE4 engine, and synthetic ice channel images were generated. This part includes 448 images, accounting for 88.9% of the dataset.

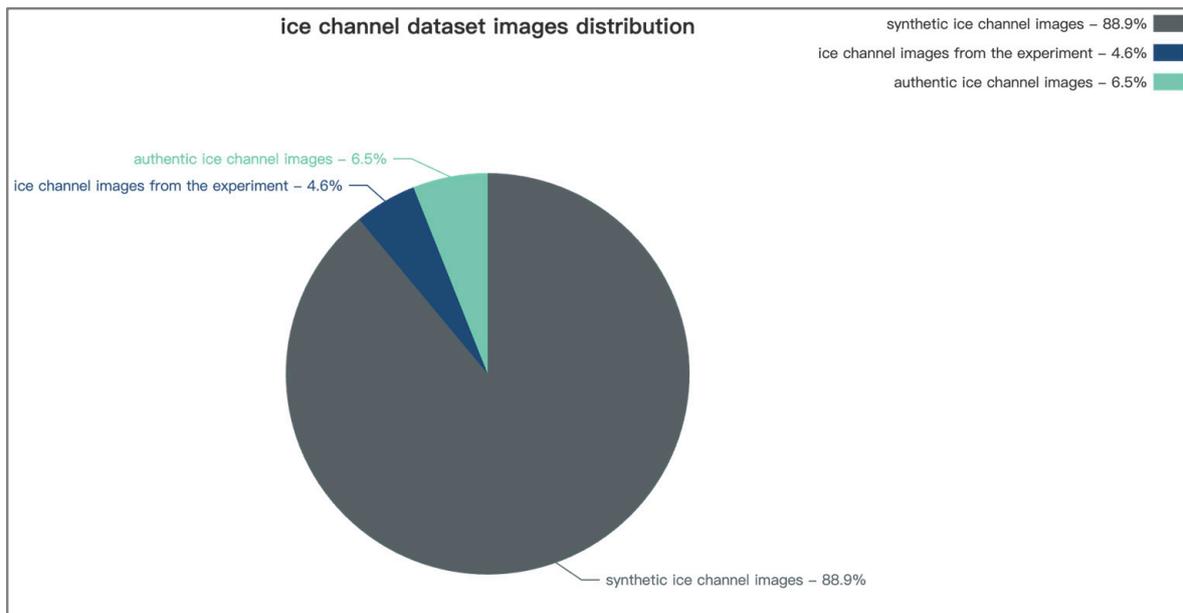


Figure 1. Ice channel dataset image distribution.

The sample images of the real ice channel dataset are shown in Figures 2 and 3. These images all adhere to the principle of being captured from the first-person perspective of a ship.

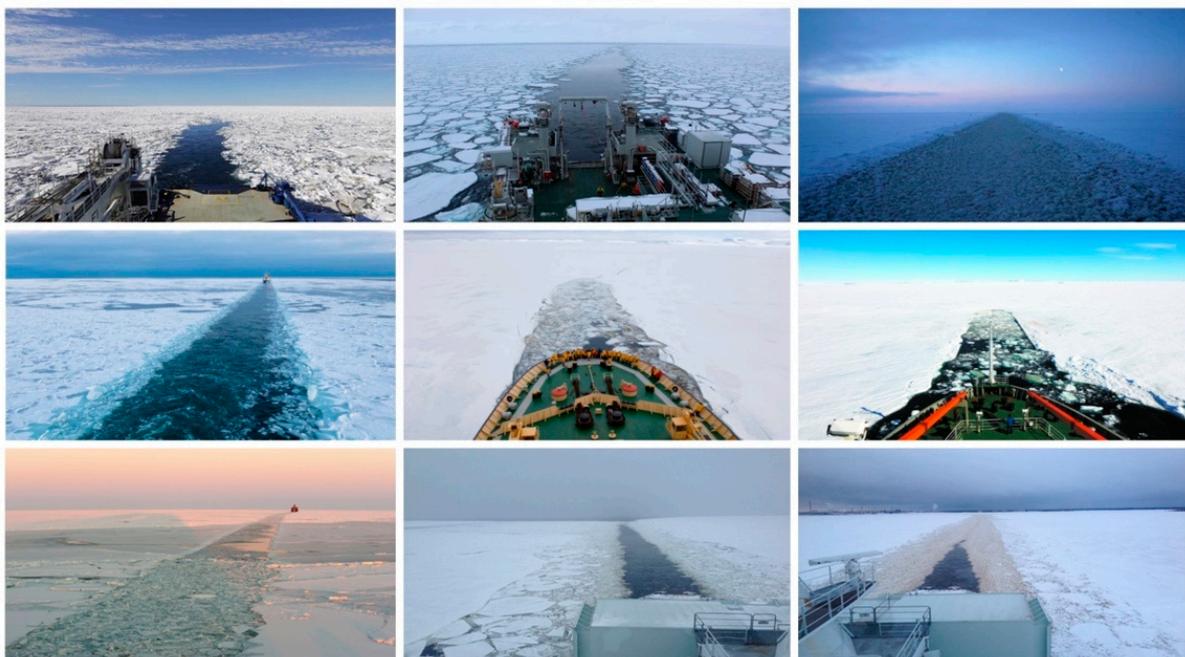


Figure 2. Sample images of the real ice channel dataset (authentic images).

The quantity of the real ice channel dataset is still not sufficient for the training and fitting of the method proposed in this article. In order to facilitate the appropriate development and testing of the ice channel recognition method, it is necessary to address the scarcity of ship images captured within ice channels. A viable solution involves constructing a synthetic ice channel dataset using platforms such as Unity3D and Unreal Engine 4. A notable example in the field of ship detection is the work of Chris M. Ward et al., which successfully generated a substantial synthetic ship image dataset using the Unity3D engine to overcome the lack of an

existing dataset [20]. The experimental results clearly indicate a significant improvement in ship classification performance when synthetic data are utilized.



Figure 3. Sample images of the real ice channel dataset (images taken from the experiment).

To model the ice channel scenario, we employ the 3D modeling software 3ds Max 2016. The process involves several steps. Initially, we construct a vast, level ice field and draw a curved channel on it, depicted in Figure 4a. Subsequently, we fragment the flat ice field to create a shattered ice field. To achieve this, we utilize the Voronoi diagram, which closely resembles an ice field with large broken ice pieces. The Voronoi diagram comprises a collection of connected polygons formed by the perpendicular bisectors of lines connecting adjacent points. We employ the RayFire [21] plug-in of 3ds Max to fragment the flat ice field, using the Voronoi fragmentation policy, as illustrated in Figure 4b. Finally, we reduce the size of the shattered ice field to 95% so that gaps between ice blocks are widened, resulting in the result shown in Figure 4c.

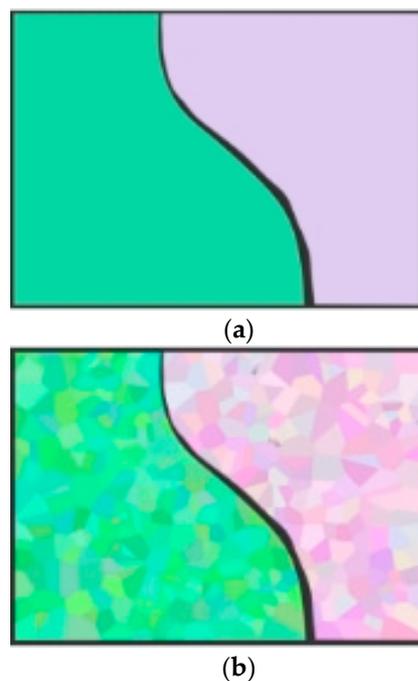


Figure 4. Cont.

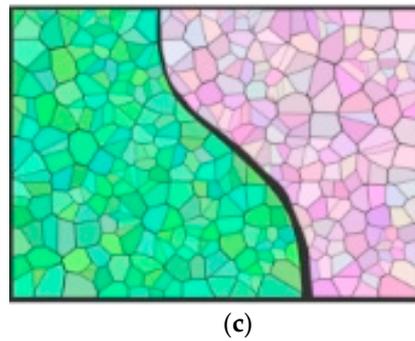


Figure 4. Ice field modeling process: (a) construct a vast and level ice field with a curved channel on it; (b) fragment the ice field; (c) the final ice field.

The next step involves importing the broken ice field model into the ice area scenario created with Unreal Engine, as shown in Figure 5. To enhance the similarity between the synthetic dataset and real ice scenarios, the ice field is initially assigned a snow material to simulate a snow-covered surface. Additionally, numerous small ice floats are inserted into the channel. To generate the synthetic dataset, a camera is placed within the scenario, which moves along the channel at a constant speed. Simultaneously, it captures images and saves them to the hard disk. The camera captures and stores 15 images per second with a resolution of 1280*720 in the PNG format, as depicted in Figure 6.

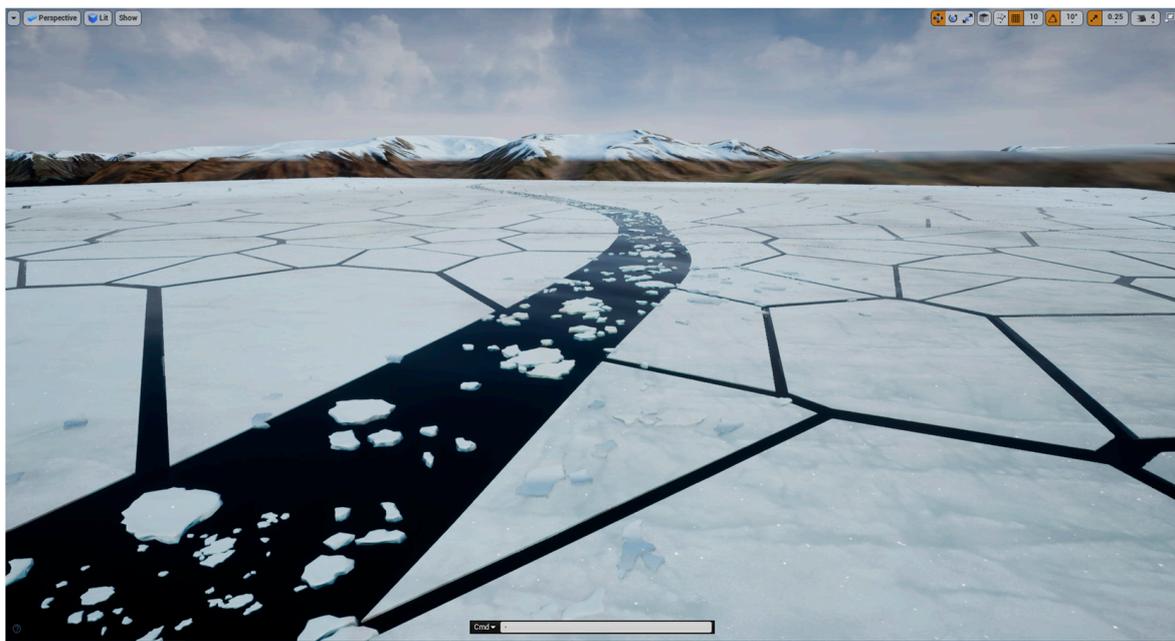


Figure 5. Arctic scenario in ue4.

To annotate the acquired images, we employ Labelme image annotation software 4.6.0. For the image segmentation method training, closed polygons are used to label the channel region. In the training of the corner point regression network, four points of a closed quadrilateral are utilized to label the four corner points of the channel. For evaluating the algorithm's accuracy, a ground-truth image with a line strip is employed to label the channel lines on both sides. In total, 448 images were sampled and labeled. Figure 7 shows a partial sample of the synthetic dataset.

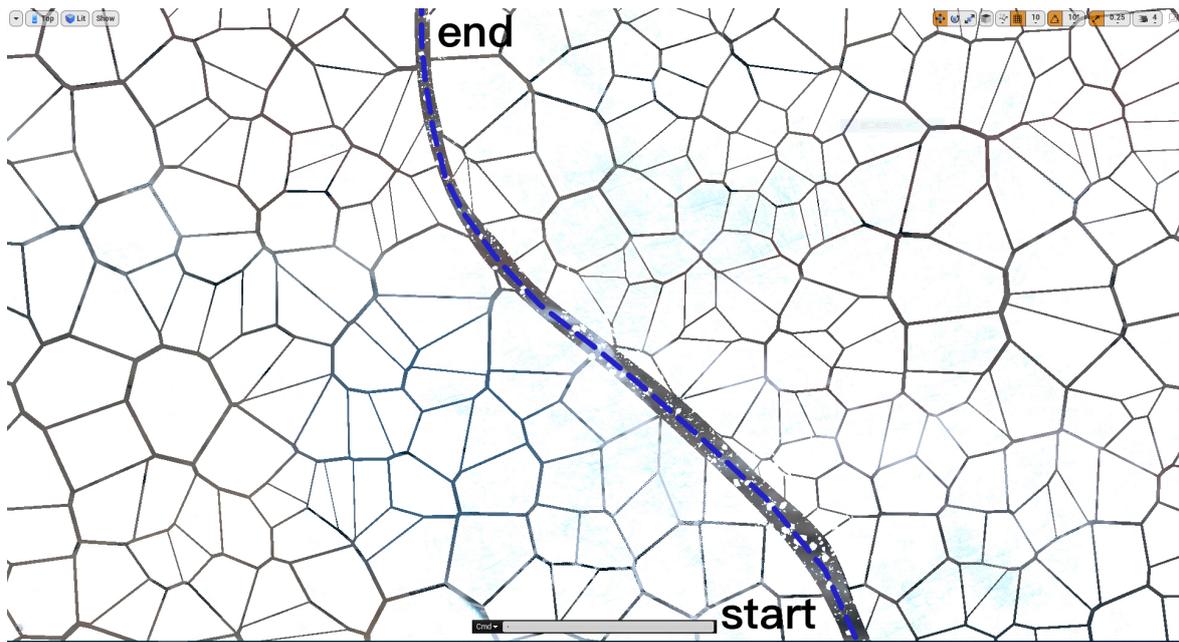


Figure 6. Camera movement sampling example.

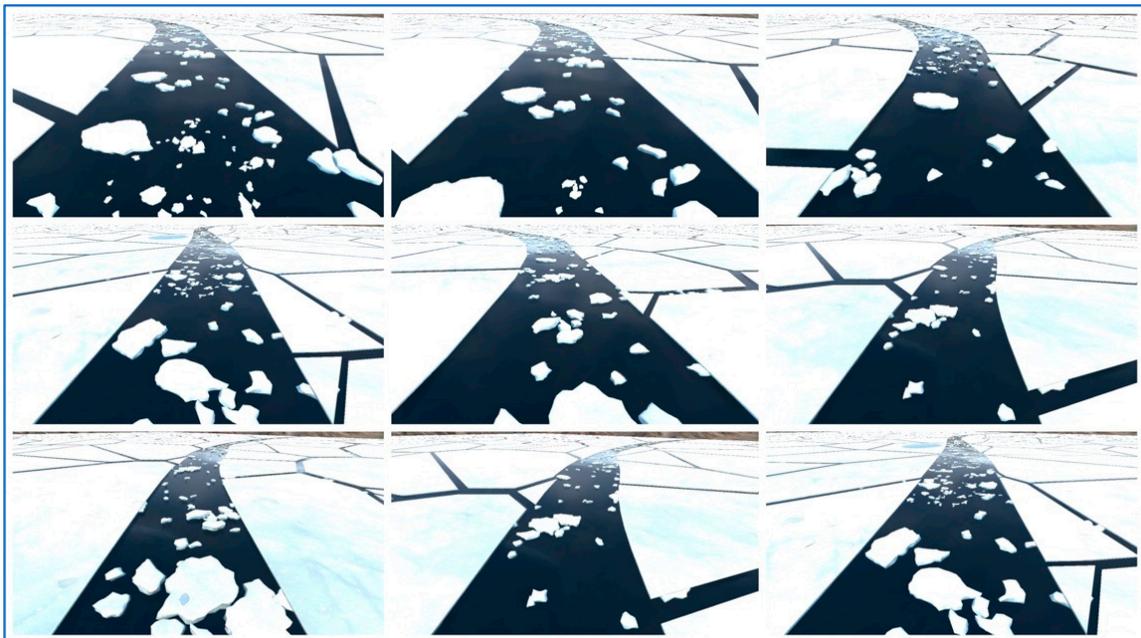


Figure 7. Sample images of the synthetic ice channel dataset.

4. Materials and Methods

The ice channel recognition task can be decomposed into the following steps. First, we captured images of the ice channel from the first-person perspective of the ship’s navigation, as shown in Figure 8a. Next, we utilized the row-based ice channel recognition method employed in this paper to identify the channel lines on both sides, as shown in Figure 8b. Finally, since the captured images of the ice channel are in a 3D perspective with a sense of depth, it is necessary to convert the channel lines from the three-dimensional perspective to a 2D top-down view for the convenience of developing algorithms for channel maintenance or obstacle avoidance within the ice channel in later stages, as shown in Figure 8c.

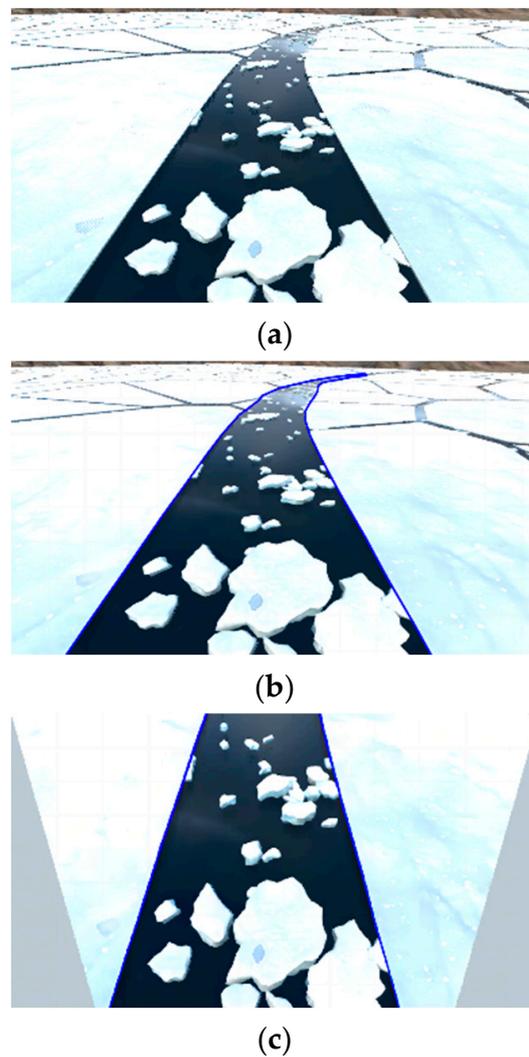


Figure 8. Definition of ice channel recognition task. (a) raw image; (b) channel lines; (c) perspective correction.

Traditional approaches to solving the recognition problem utilize image segmentation methods, which initially determine whether each pixel in the image belongs to the ice channel or not, and then employ clustering algorithms to distinguish the left and right channel lines. However, these methods tend to have high computational complexity. Moreover, image segmentation is more suitable for segmenting regions with clear boundaries, whereas channel lines lack distinct boundary areas. Hence, segmentation methods are not well-suited for addressing the challenge of channel recognition in this context.

We employed a row-based selection method to address the ice channel recognition problem. As shown in Figure 9, this method involves dividing the ice channel image into multiple rows, and each row is further divided into several cells. The recognition is accomplished by determining whether each cell contains the ice area channel line. If a cell contains the channel line, the center coordinates of that cell are marked as the coordinates of the channel line point. The ice area channel line is obtained by connecting the recognized channel line points from each row. This approach transforms the problem into a multi-cell image classification task, where each cell is classified as either containing or not containing the ice area channel line. The detailed description of this method is as follows.

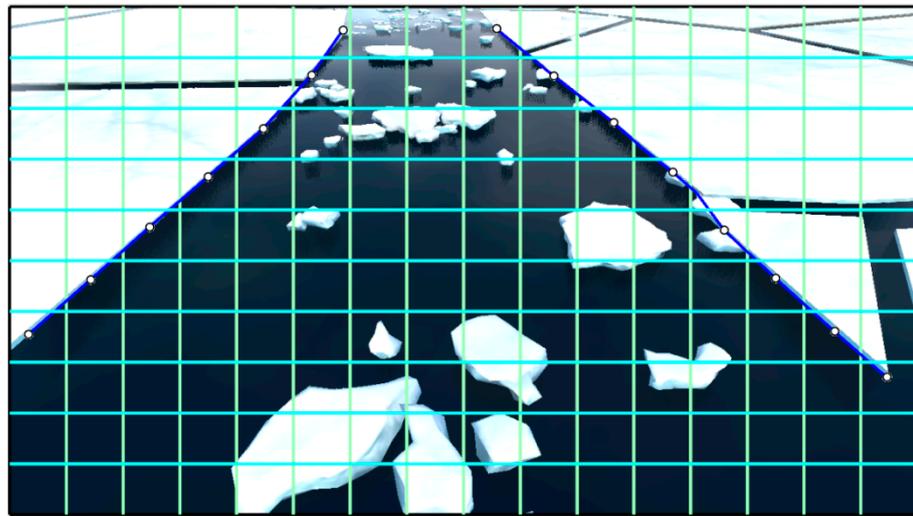


Figure 9. Row-based selection strategy.

4.1. Ice Channel Recognition

To address the ice channel recognition problem, a row-based selection method called UFAST [22] is employed. This method utilizes global image features to choose cells of the ice channel along predefined rows. The overall architecture of the method is shown in Figure 10. The lower part displays the auxiliary branch, which is active only during training. The image is first processed through Res blocks for feature extraction, then enters Group classification to classify cells containing channel lines, and finally extracts channel lines. Group classification is performed on each row anchor. Res blocks are the backbone network of this network, which uses the famous Resnet-18 to extract features from input images. During prediction, the extracted features will be fed into group classification, which is actually a fully connected layer used to obtain the probability matrix of channel lines. In group classification, the features are first linearized and then transformed into a two-dimensional probability map, where each cell contains the probability of having a channel line for that cell. By iterating through each row and finding the cell with the highest probability, the position of the lane line in that row can be determined. During training, the features are also fed into auxiliary segmentation. In auxiliary segmentation, 2D convolution is performed, followed by data normalization using BatchNorm2d. This ensures that the data do not cause instability in network performance due to large values before undergoing ReLU activation, ultimately obtaining segmentation instances for each lane line.

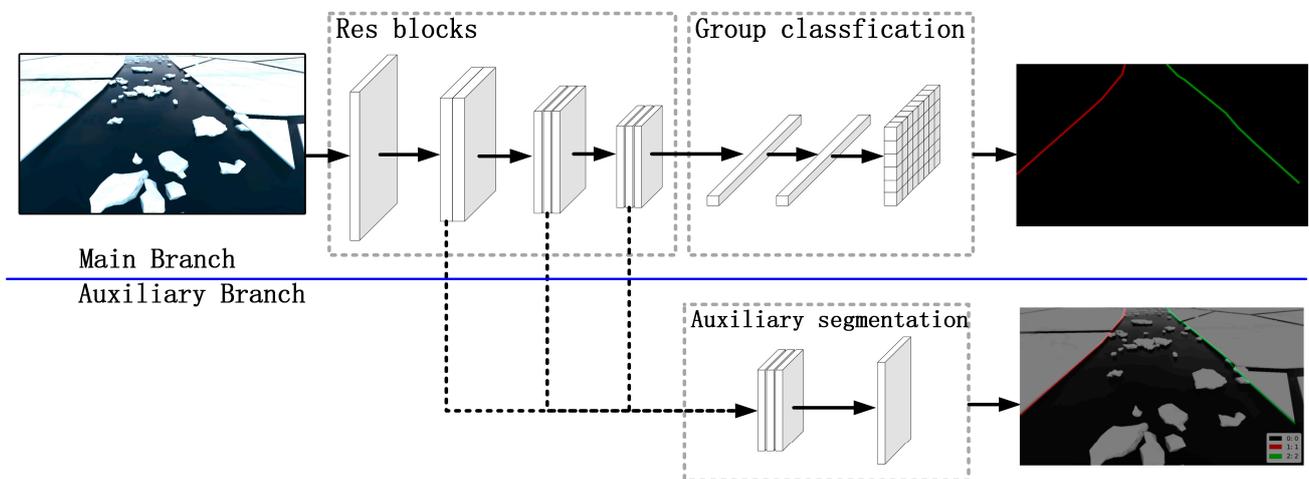


Figure 10. Overall architecture of the ice channel recognition method.

Ice channel images are divided horizontally into multiple rows, with predefined widths for each row, referred to as row anchors. The location within each row anchor is further subdivided into consistent-width cells. Identifying the ice channel involves selecting the appropriate cells that belong to the ice channel class across the predefined row anchors. Each ice channel is characterized by two boundary lines. The global image feature is denoted as X . Here, h represents the number of row anchors and w represents the number of gridding cells. The location of the i -th channel line in the j -th row anchor is determined by utilizing the classifier f^{ij} . The prediction for each ice channel's boundary lines can be expressed as follows:

$$P_{i,j} = f^{ij}(X), i \in [1,2], j \in [1,h] \tag{1}$$

In the equation above, $P_{i,j}$ is a vector with $w + 1$ dimensions and $P_{i,j}$ represents the probability of selecting gridding cells for the i -th ice channel line in the j -th row anchor. For a typical ice channel, it always consists of two lines, left and right, so i is set to the range from 1 to 2. Therefore, the problem is simplified, and the number of navigation lanes is predefined as 2. $T_{i,j}$ denotes the correct label for the right locations. Therefore, the optimization of this formulation can be expressed as follows:

$$L_{cls} = \sum_{i=1}^2 \sum_{j=1}^h L_{CE}(P_{i,j}, T_{i,j}) \tag{2}$$

In the above equation, L_{CE} represents the cross-entropy loss function and L_{cls} represents the classification loss. To indicate the presence of boundary lines, an additional dimension is introduced, resulting in $w + 1$ dimensions instead of w dimensions. It is evident that this method is simpler and faster compared to segmentation methods. For instance, if an image frame contains $H \times W$ pixels, the segmentation method would require $H \times W$ classification problems. However, the number of predefined row anchors and cells is significantly smaller than the image size, with $h \ll H$ and $w \ll W$. In contrast, this method only needs to handle $2 \times h$ classification problems, as there are only 2 boundary lines in an image. As a result, the computational cost of this method is significantly lower than that of segmentation methods.

The loss function of the method comprises both the classification loss and the similarity loss. The location of the lane is represented using a classification vector. The similarity loss function can be defined as follows:

$$L_{sim} = \sum_{i=1}^2 \sum_{j=1}^{h-1} \|P_{i,j} - P_{i,j+1}\|_1 \tag{3}$$

In the equation above, L_1 represents the norm and $P_{i,j}$ corresponds to the prediction for the j -th row anchor. The latter is associated with the shape of the boundary lines. Typically, the majority of boundary lines are straight, and even curved boundary lines tend to appear straight due to the perspective effect. To constrain the shape of the ice channel boundary lines, a second-order differential equation is employed. For any given line index i and row anchor index j , the location can be expressed as:

$$Loc_{i,j} = \operatorname{argmax}_k P_{i,j,k}, k \in [1,w] \tag{4}$$

In the equation above, k represents the location index. The range of the location index is from 1 to w , instead of $w + 1$. $P_{i,j,k}$ represents the probability of the ice channel line in the k -th cell of the j -th row of the i -th line. Performing an argmax operation on $P_{i,j,k}$ corresponds to obtaining the index of the maximum value in $P_{i,j,k}$, which indicates the index of the cell where the channel line appears in that row, i.e., the location. The expectation of predictions is utilized as an approximation of the location. To obtain the probability of different locations, the softmax function is applied:

$$Prob_{i,j} = \operatorname{softmax}(P_{i,j,1:w}) \tag{5}$$

The equation above is a vector with dimensions. The probability at each location is denoted as $P_{i,j,1:w}$. The expectation of locations can be expressed as:

$$Loc_{i,j} = \sum_{k=1}^w k \cdot Prob_{i,j,k} \tag{6}$$

The probability of the i -th ice channel line, the j -th row anchor, and the k -th location is represented as $Prob_{i,j,k}$. The second-order difference constraint function can be defined as follows:

$$L_{shp} = \sum_{i=1}^2 \sum_{j=1}^{h-2} \|(Loc_{i,j} - Loc_{i,j+1}) - (Loc_{i,j+1} - Loc_{i,j+2})\|_1 \tag{7}$$

The equation above represents the location on the i -th ice channel line and the j -th row anchor. To learn the distribution of the first-order difference of line location, the method requires additional parameters. The overall structural loss can be expressed as:

$$L_{str} = L_{sim} + \lambda L_{shp} \tag{8}$$

In the equation above, λ represents the loss coefficient. The method incorporates an auxiliary feature aggregation that operates on both global and local image features. The overall architecture of the method is depicted in Figure 10. Cross-entropy is employed as the auxiliary segmentation loss. The overall loss of the method can be formulated as:

$$L_{total} = L_{cls} + \alpha L_{str} + \beta L_{seg} \tag{9}$$

In the equation above, L_{seg} represents the segmentation loss, whereas α and β denote the corresponding loss coefficients.

4.2. Perspective Correction

The pixel coordinates on the ice channel line are (x, y) , and (u, v) represents the corresponding pixel coordinates of (x, y) in the camera perspective, as shown in Figure 11. Based on the following equation, the camera perspective coordinates (u, v) of the ice channel point (x, y) can be obtained from its 3D perspective coordinates

$$x = \frac{1}{(d+n)} [c \cdot (n-y) + u \cdot (d+y)] \tag{10}$$

$$y = \frac{vf^2}{fH - vd} \tag{11}$$

In the equation, H represents the height of the camera, f represents the focal length of the camera, c represents the horizontal coordinate of the camera in the camera perspective, $d = 2f$, $f = \sqrt{H^2 + d^2}$, and n represents the difference between the coordinate mapping point of (u, v) and d . By using Equations (1) and (2), the coordinate values of (u, v) can be calculated.

$$u = \frac{c \times (y - n) + x \times (n + d)}{d + y} \tag{12}$$

$$v = \frac{f \times H \times y}{f^2 + d \times y} \tag{13}$$

By using the coordinates of the “vanishing point” formed by the two boundary lines in the 3D perspective, the horizontal coordinate of the camera at that moment can be deduced. From the two boundary lines, we obtain four points $(u_1, y_1), (u_2, y_2), (u_3, y_1), (u_4, y_2)$. From the previous equation, we can derive the following equation:

$$x[(u_3 - u_4) + (u_1 - u_2)] = u_2(u_3 - u_4) - u_3(u_2 - u_1) \tag{14}$$

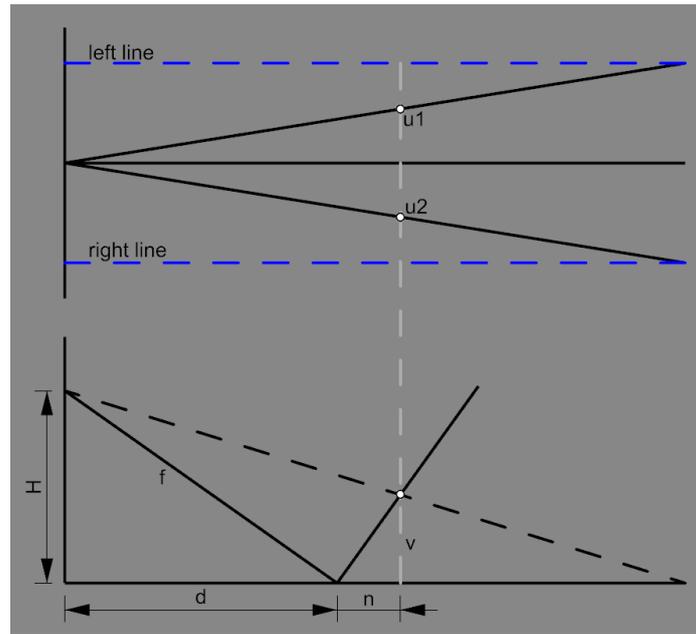


Figure 11. Perspective correction.

The result of the equation is $x = c$, which means that the horizontal coordinate of the “vanishing point” is equal to the horizontal coordinate of the camera. Based on the following system of equations:

$$\begin{cases} \frac{y_1 - v_1}{v_1} = \frac{x - u_4}{u_4 - c} \\ \frac{y_2 - v_2}{v_2} = \frac{x - u_3}{u_3 - c} \end{cases} \quad (15)$$

The calculated value of the camera’s horizontal coordinate c is:

$$c = \frac{v_1 y_2 u_3 - v_2 y_1 u_4}{y_2 y_1 - y_1 y_2} \quad (16)$$

4.3. Evaluation Criteria

The evaluation of the method is conducted in terms of recognition speed and recognition accuracy. In terms of recognition speed, the method needs to meet real-time requirements, with a minimum of 15 frames per second. Regarding recognition accuracy, the recognition accuracy of the ice channel is calculated using the following equation:

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \quad (17)$$

In the equation above, C_{clip} represents the number of predicted ice channel boundary line points that meet the accuracy requirement, and S_{clip} denotes the total number of ice channel ground-truth instances (obtained from manually labeled images and labeled information) in each clip. Regarding how to determine C_{clip} , the distance d between the predicted ice channel points and the ground-truth ice channel points is calculated for each row, as shown in Figure 12. An error tolerance value δ is set. When $d \leq \delta$, it is considered that the point meets the accuracy requirement. C_{clip} represents the number of ice area channel points that satisfy the accuracy requirement, considering all the points that meet the accuracy requirement.

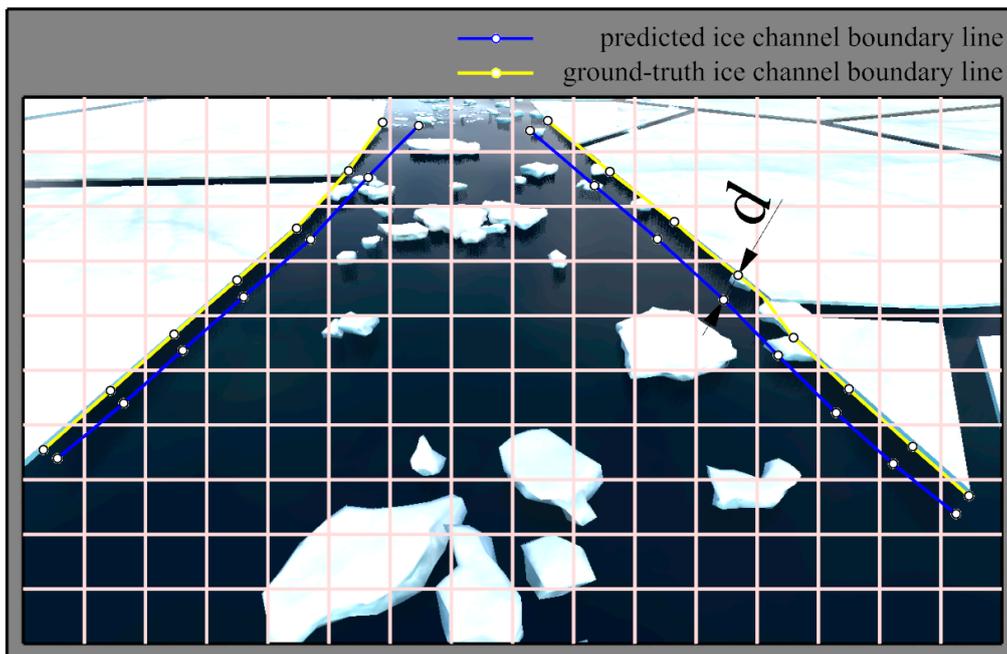


Figure 12. Determine boundary line points meet accuracy requirements.

5. Results

5.1. Training

For the dataset, we defined row anchors ranging from 160 to 710, with a step corresponding to the image height of 720 pixels. Initially, we set the number of gridding cells to 100. In the optimization process, the images are resized to a specific size. During training, we employed the Adam optimizer with an initial learning rate. The loss coefficients λ , α , and β were all set to 1 by default. The batch size was set to 4. The training epoch was set to 500 iterations. The hardware platform used in this experiment consisted of an AMD R7 3700X CPU @ 3.6 GHz processor, 16 GiB of memory, and a GeForce GTX 1080 graphics card with 8 GiB of video memory. The system software environment included Ubuntu 20.04, Python 3.7, and PyTorch 1.6. We utilized 60% of the images in the dataset as the training set. The training loss plot can be seen in Figure 13. It can be observed that the loss value rapidly decreases during the first 100 epochs of training, and then gradually approaches zero. This indicates that the model is being trained and fitted well, and the training stops at the 500th iteration.

5.2. Recognition Results

The method was tested separately on multiple parts of the dataset, and the test results are shown in Table 1. Firstly, the channel recognition speed met the real-time requirements in all parts of the dataset, with an average recognition speed of 138.3 frames per second. This reflects the advanced computational efficiency of the method proposed in this paper.

Table 1. Overall test results.

Dataset		Accuracy (%)	FP (%)	FN (%)	Speed (frames/s)
real ice channel	Authentic images	79.8	5.84	4.04	137
	experiment images	85.4	6.11	3.92	138
synthetic ice channel		87.1	6.84	3.69	140
average		84.1	6.26	3.88	138.3

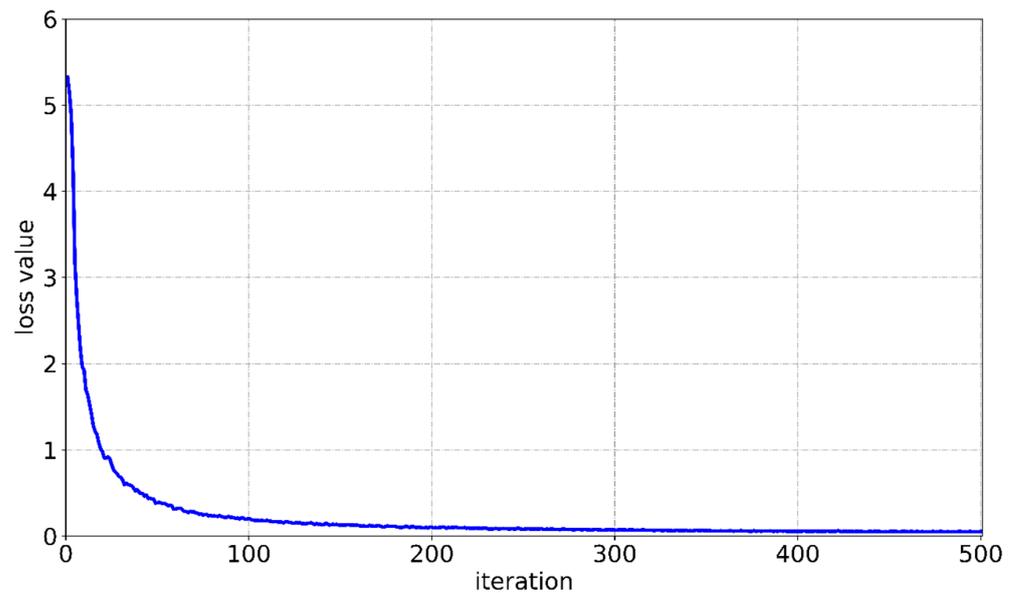


Figure 13. Ice channel recognition method training loss plot.

The average channel recognition accuracy reached 84.1%, but there were some accuracy degradation phenomena observed in the real channel dataset. This is because real channel images have higher scene complexity compared to synthetic channel images, including cases where many channel lines are occluded or not clearly visible. Figure 14 shows the recognition results on real channels, while Figure 15 displays the recognition results of images obtained from experiments on real channels. Figure 16 demonstrates the recognition results of synthetic channels. From Figure 14, it can also be seen that in the case where the channel line is obstructed, the method is able to identify the unobstructed navigation line and predict the extension of the obstructed invisible portion. It is precisely this characteristic of predicting the occluded portions of channel lines that leads to a decrease in recognition accuracy. It can be observed from Figure 14 to Figure 16 that the fitting performance is better for nearby ice channel lines, but there are more recognition deviations when the lane lines are curved in the distance. Additionally, it can be seen that channel lines in the farthest distance are not recognized. This is because, in the design of this method, only the lower 70% of the lane lines in the image are recognized. The recognition and significance of the upper 30% of the lane lines are limited due to their distance, so they have been optimized and omitted to improve the efficiency of the method.

The average false positive (FP) rate reached 6.26%, and the false negative (FN) rate reached 3.88%, where FP represents the number of false positives, which refers to predicted samples classified as channel points but are actually negative samples (incorrectly labeled as channel points), and FN represents the number of false negatives, which refers to predicted samples classified as non-channel points but are actually positive samples (missed detection of channel points).

Figure 17 illustrates the classification probability map of ice channels, which is the feature map, during the process of ice channel recognition. The cells with darker blue colors in the image represent a higher probability of containing navigation routes. In contrast, the other cells are manifested as light gray because they represent probabilities of having ice channel lines that are extremely small, in the order of 10^{-4} . By traversing through each row and obtaining the cell with the highest probability, we can determine the left and right ice channels. It can be observed that some lighter blue cells appear near the deeper blue cells. This indicates that in the vicinity of cells with higher probabilities of containing ice channel lines, there are some cells with lower probabilities of containing ice channel lines. This is because ice channel lines have a certain width, and cells near the areas with a significant contrast to seawater have higher probabilities, appearing as a deeper

blue. On the other hand, cells along other ice channel lines with lower contrast have lower probabilities, appearing as a lighter blue.

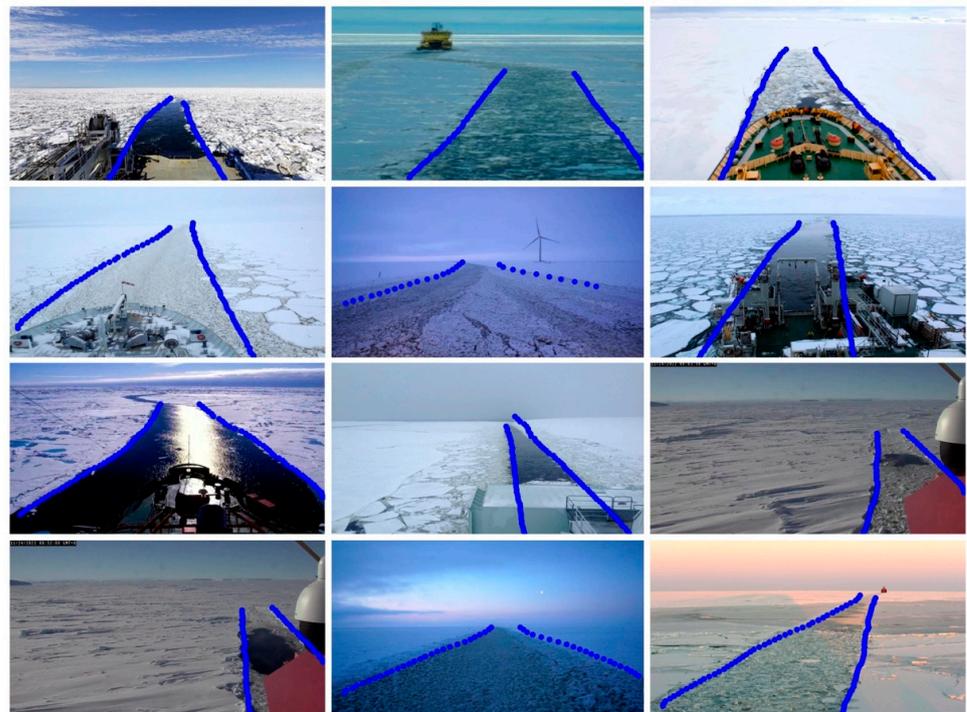


Figure 14. Ice channel recognition test effect based on real ice channel dataset.

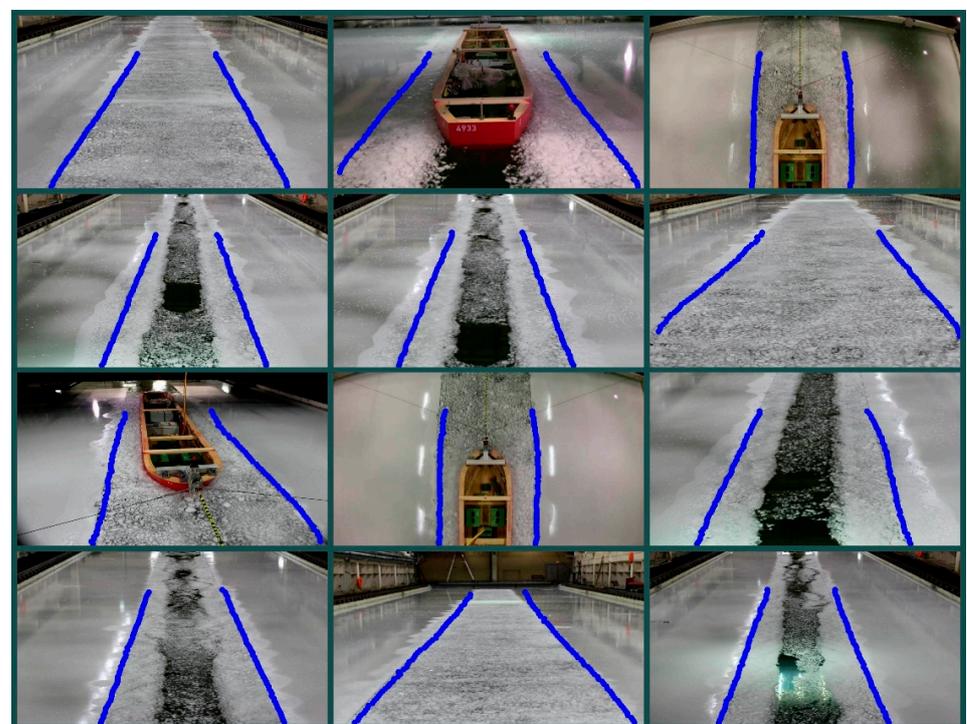


Figure 15. Ice channel recognition test effect based on real ice channel dataset (images taken from the experiment).

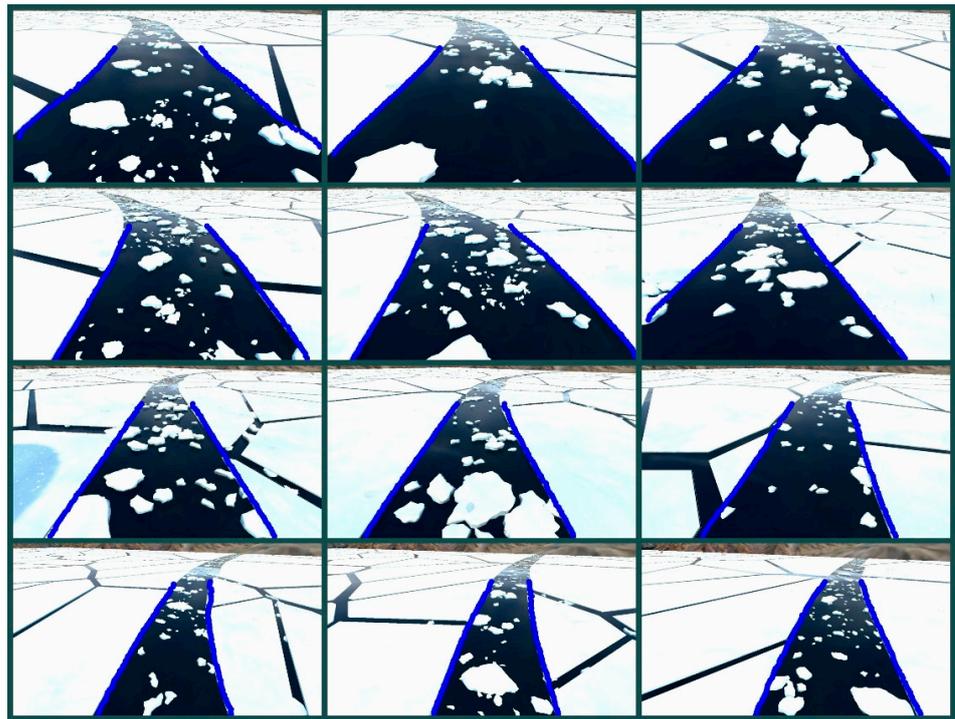


Figure 16. Ice channel recognition test effect based on synthetic ice channel dataset.

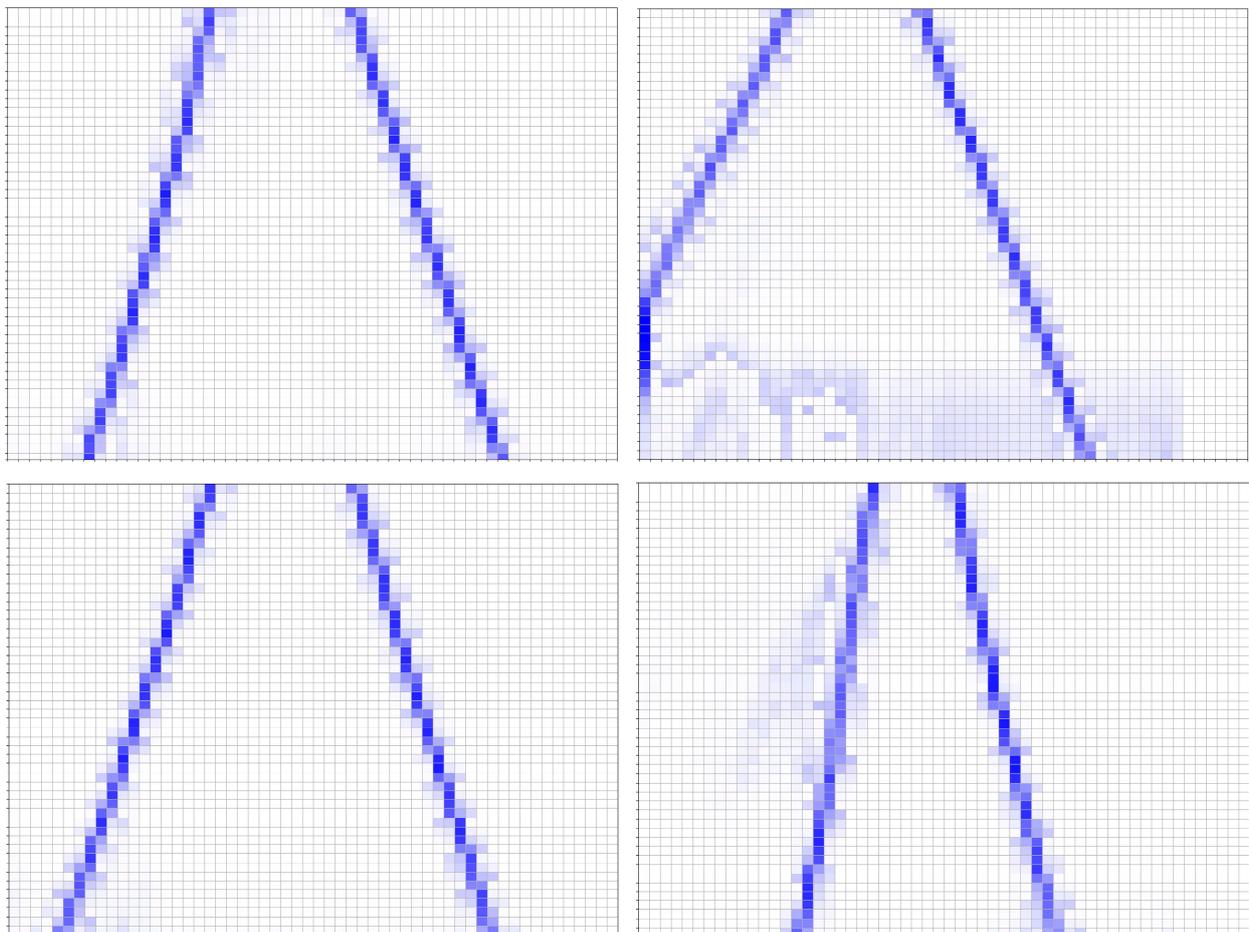


Figure 17. Feature map during ice channel recognition.

Regarding perspective correction, since the camera parameters (including focal length, camera center distance, camera height, and camera pitch angle) of the images in the real ice area dataset are unknown, no perspective correction is applied to the recognition results of the real ice area dataset. As for the synthetic ice area dataset, when constructing the dataset, the camera parameters set in UE4 were as follows: Camera height of 10 m, focal length of 28 mm, and pitch angle of 15.5 degrees. The perspective correction results are shown in Figure 18. It can be observed that the recognition results of the ice area channels, which originally had a perspective effect in the 3D view, have been successfully corrected to a top-down 2D view perspective.

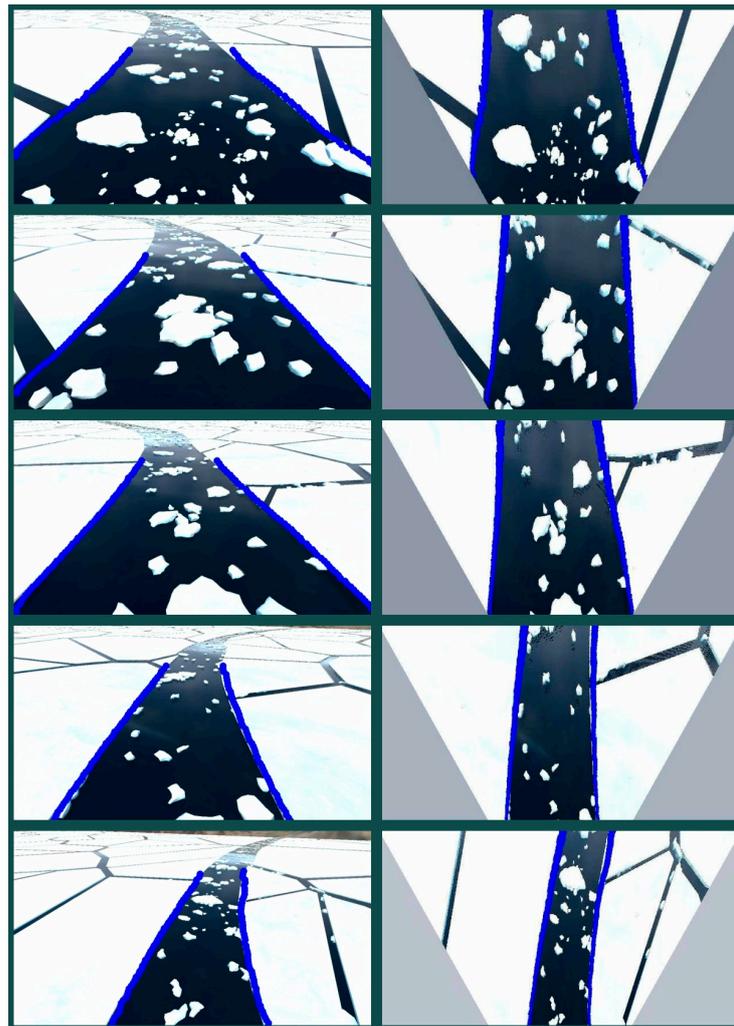


Figure 18. Ice channel after perspective correction.

5.3. Compared with Traditional Segmentation Method OTSU

The method used in this paper is based on line selection to achieve the recognition of ice channel lines, which is fundamentally different from segmentation-based methods. The segmentation-based method predicts whether each pixel in the image is a channel line or segments the channel area and then converts the area to obtain the channel lines on both sides. For the segmentation of the ice channel, it is not practical to directly segment the channel lines on both sides because there are no obvious boundaries for the navigation lines in authentic images. If there are clear boundaries, it is difficult to directly segment the navigation lines due to the small width of the boundaries and the limited pixels they contain. Therefore, when using the segmentation method, it is necessary to first segment

the channel area, obtain the closed boundary of the channel area, and then process the boundary to obtain the channel lines on both sides.

This section uses traditional non-intelligent segmentation methods to attempt to recognize ice channel lines. The method used is OTSU, which is a threshold segmentation algorithm for determining image binarization. It was proposed by Japanese scholar Otsu in 1979 and is considered one of the best algorithms for traditional image segmentation [23]. This method is computationally simple and segments the image into the foreground and background based on the grayscale characteristics of the image.

Performing OTSU segmentation on the dataset yields some sample images, as shown in Figures 19–21. OTSU performs binary segmentation on the input images, where the yellow region represents non-channel areas and the deep purple region represents channel areas. From Figure 19, it can be observed that in the presence of sun reflection, the reflected channel area is incorrectly segmented as a non-channel area. The deck of the ship is also incorrectly segmented as a channel area. From Figure 20, it can be seen that when there is a significant amount of fragmented ice in the channel, only the water area within the channel can be segmented, and the channel area cannot be fully segmented. From Figure 21, it can be seen that the OTSU method performs well on synthetic datasets with simpler scenes. However, it also segments the cracks between ice blocks into the channel area, which hinders obtaining the channel lines on both sides.

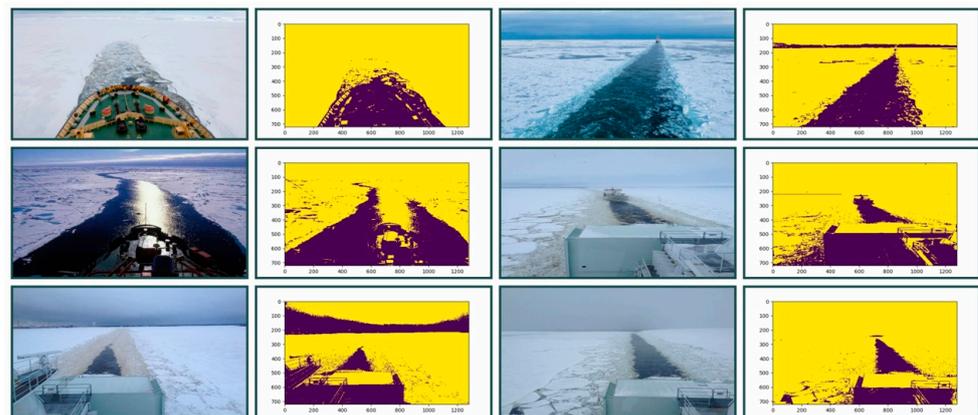


Figure 19. Sample results using Otsu segmentation on the real ice channel dataset.

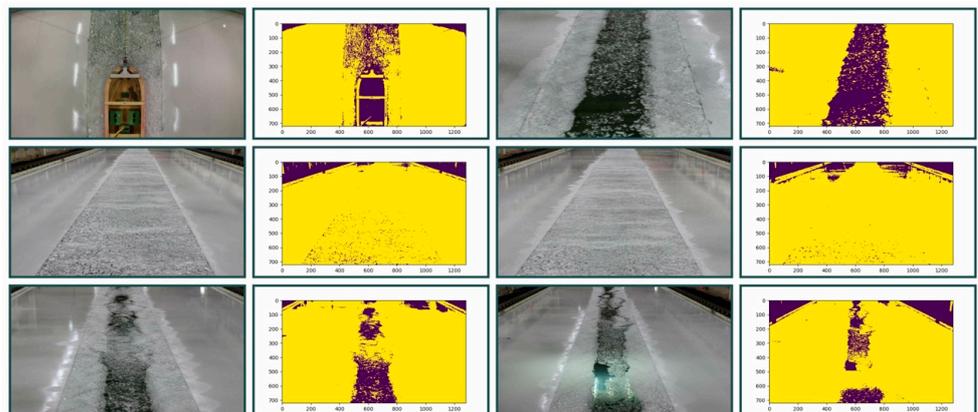


Figure 20. Sample results using Otsu segmentation on the real ice channel dataset (images taken from experiment).

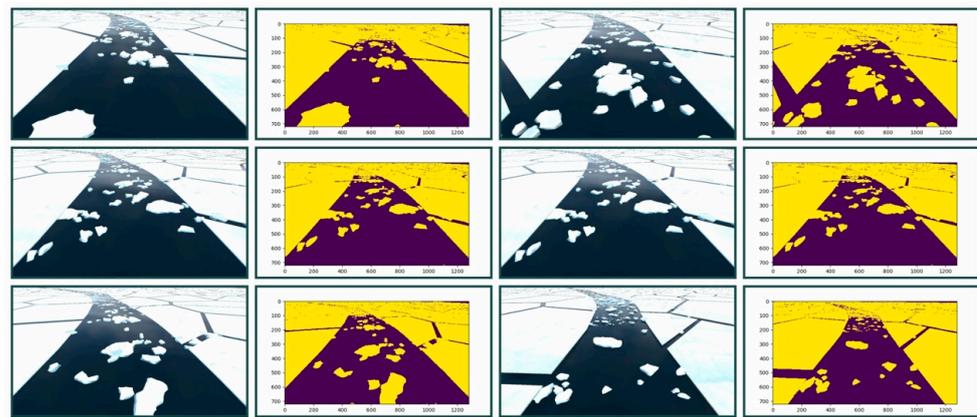


Figure 21. Sample results using Otsu segmentation on the synthetic ice channel dataset.

In conclusion, traditional segmentation methods are unable to cleanly and perfectly segment the channel area in complex authentic ice channel scenes. The segmentation results cannot be used as the output for further processing. Traditional segmentation methods are not applicable for ice channel recognition. However, the method based on row selection used in this paper can directly output the navigational lines on both sides after training.

5.4. Compared with Intelligent Segmentation Method YOLACT

Using robust image segmentation methods can delineate the boundaries of the ice channel area. The ice channel images taken at the ship’s bow present a trapezoidal shape. When the ship sails steadily, the pitch angle of the camera does not change much, so cropping the top and bottom of the channel area can obtain the left and right channel lines.

YOLACT is a real-time instance segmentation model developed by Facebook AI Research that efficiently predicts masks and object bounds in parallel by separating mask prediction into learned prototype masks and their predicted per-instance coefficients [24]. In this paper, the YOLACT instance segmentation algorithm is used to obtain the ice channel area by taking the ice channel dataset as input, then the top and bottom of the channel area are manually cropped to finally obtain the channel lines on both sides. The channel area segmentation results and channel line recognition results are shown in Figure 22. The channel areas segmented in the second column of pictures are covered with colored regions. By manually setting the upper and lower clipping lines to crop the top and bottom boundaries and after equal interval fitting processing, the channel lines shown in the second column of Figure 22 are obtained.

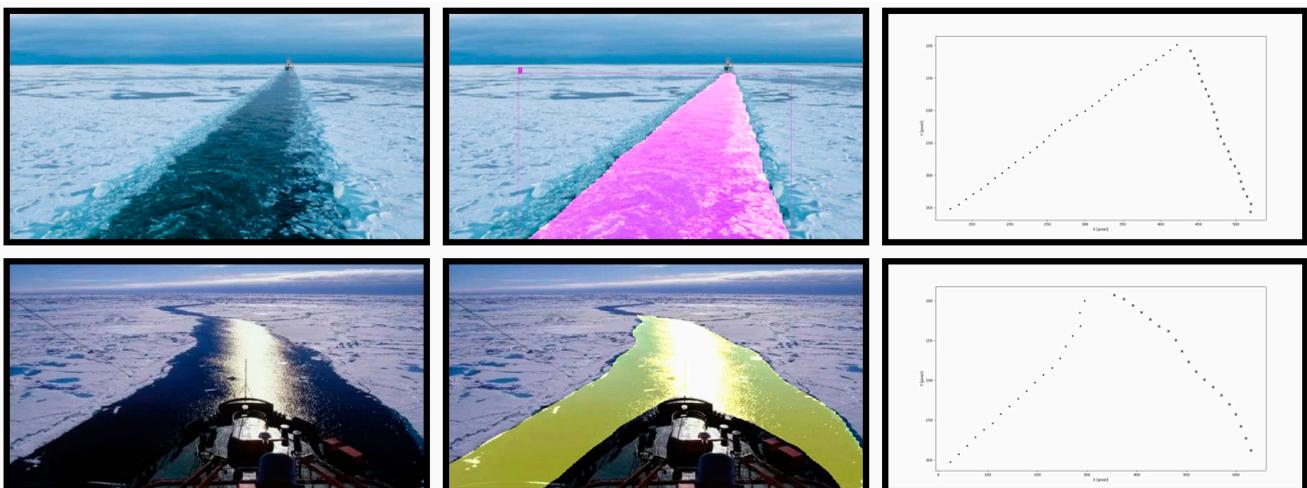


Figure 22. Cont.

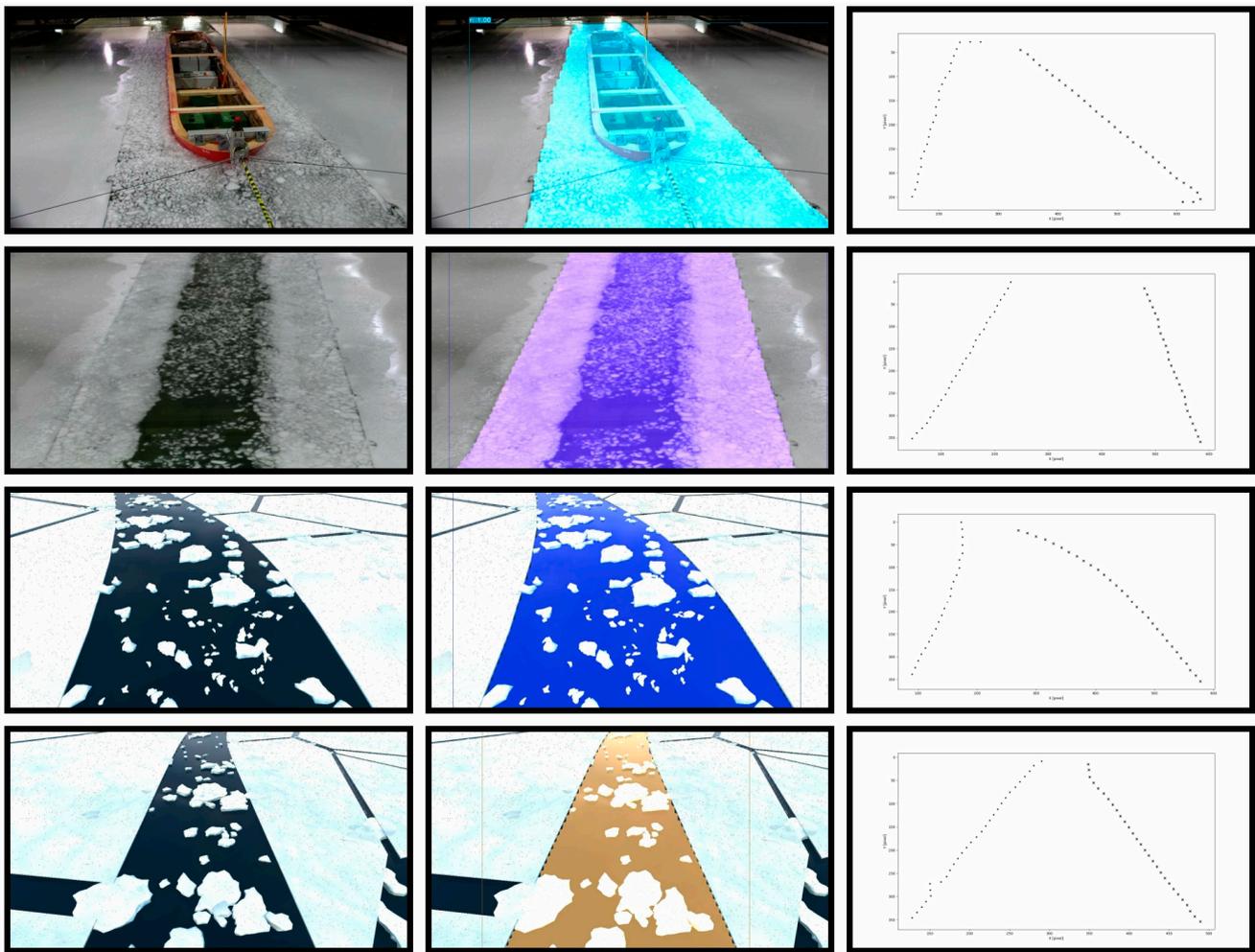


Figure 22. The channel area segmentation sample images and channel line recognition sample images.

Using this method for channel line extraction, we evaluated the two aspects of recognition accuracy and recognition speed using the same evaluation criteria as the method in this paper. Each part of the dataset is evaluated separately to obtain the performance comparison table shown in Table 2. In comparison, the method in this paper exceeds the yolact+crop method by 9.5% in recognition accuracy and is 103.7 frames per second faster in recognition speed. When using YOLACT to segment the ice channel area, it has good accuracy and robustness, but in the next crop operation, on the one hand, the upper and lower cutoff ranges need to be set manually, which makes this method difficult to apply in practice. Second, setting the upper and lower cutoff ranges introduces more errors, because only cutting off more of the upper and lower channel boundaries can leave clear left and right boundaries. In addition, due to the inherent nature of assigning labels to each pixel, the segmentation method has huge computational complexity, which is manifested as the recognition speed of the method in this paper being more than 103.7 frames per second faster than this method on the same hardware device. The method in this paper exceeds this method in both recognition accuracy and recognition speed, proving that the line selection-based method has stronger performance compared to the segmentation method. Also, all parameters are automatically tuned through training, making the method in this paper more suitable for practical applications.

Table 2. Performance comparison of ice channel recognition.

Dataset		Ours	yolact+crop	Ours	yolact+crop
		Accuracy (%)		Speed (frames/s)	
real ice channel	authentic images	79.8	72.2	137	35
	experiment images	85.4	73.7	138	34
synthetic ice channel		87.1	77.3	140	35
average		84.1	74.6	138.3	34.6
		Advanced 9.5%		Advanced 103.7 fps	

5.5. Ablation Study

As mentioned in the methodology section, gridding and selection techniques are utilized to establish the relationship between the structural information in ice channel boundary lines and the classification-based formulation. Therefore, it is reasonable to investigate the impact of different numbers of gridding cells on the method’s performance. While the number of row anchors is predefined and fixed, the number of cells in each row was varied. We divided the image using 25, 50, 100, and 200 cells in columns.

The accuracy of the algorithm was tested for each case, and the results are shown in Figure 23. It can be observed that as the number of gridding cells increases, the classification accuracy gradually decreases. This phenomenon can be attributed to the fact that more gridding cells impose greater requirements for finer-grained and more challenging classification. When the number of gridding cells increases, the area of each gridding cell becomes smaller, resulting in a decrease in the amount of image information contained within each cell. In such cases, if there is an ice channel line within a cell, the small cell area makes it challenging to include sufficient feature information for convolutional operations to extract. Therefore, determining whether a cell contains an ice channel line becomes more prone to errors. Additionally, the evaluation accuracy does not exhibit a monotonic variation. Although a smaller number of gridding cells leads to a larger localization error and higher classification accuracy, it also results in the imprecise representation of the exact location due to the larger size of the gridding cell. Based on this ablation study on the number of gridding cells, we ultimately determine that 50 is the optimal number of gridding cells in the synthetic dataset.

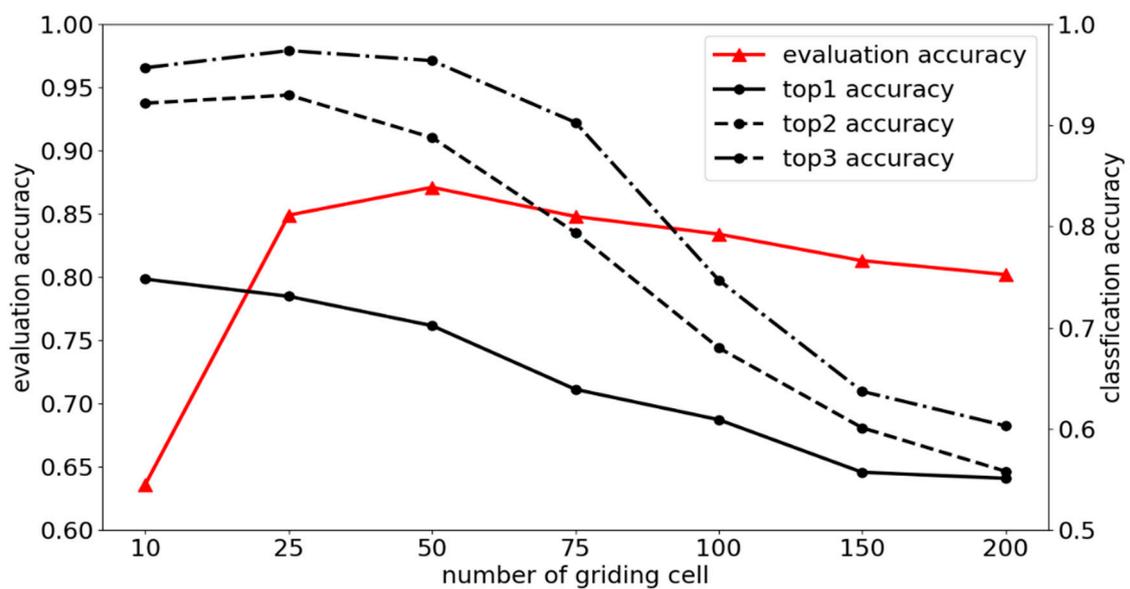


Figure 23. Performance under different numbers of gridding cells.

6. Conclusions

This paper presents an ice channel recognition method based on the UFAST lane detection algorithm. To address the problem of method training fitting, real ice channel datasets were collected and constructed. Additionally, synthetic ice channel datasets were generated using UE4. The method was trained and tested on both the real ice channel dataset and the synthetic ice channel dataset. Furthermore, an ablation study was conducted to explore the optimal determination of method parameters. Based on the comprehensive analysis, the following conclusions were drawn.

- (1) The method achieved a recognition accuracy of 84.1% on the ice channel dataset and a recognition speed of 138.8 frames per second.
- (2) The method in this paper exceeds the yolact+crop method by 9.5% in recognition accuracy and is 103.7 frames per second faster in recognition speed. The method in this paper is more suitable for practical applications.
- (3) During the ablation study, it was observed that the evaluation accuracy does not exhibit a monotonic variation. As the number of gridding cells increases, the classification accuracy gradually decreases. This is because more gridding cells require finer-grained and more challenging classification. Ultimately, based on the ablation experiments, we determine that 50 is the optimal number of gridding cells to achieve the best performance.

In future work, first, more authentic ice channel images need to be collected and used for training so that the method has greater robustness and applicability. Second, advanced networks such as EfficientNet and Transformer can be used as the backbone network of the method in this paper to extract features. Finally, when ice channel line recognition is more mature, ice channel line departure warning and the line-keeping assistance algorithm can be studied on this basis.

Author Contributions: Methodology, W.D.; Software, W.D., Q.M. and F.L.; Validation, L.Z.; Formal analysis, W.D.; Investigation, W.D., L.Z., S.D., Q.M. and F.L.; Resources, W.D.; Data curation, W.D.; Writing – original draft, W.D.; Writing – review & editing, L.Z. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Key Research and Development Program (Grant 2022YFE010700), General Projects of the National Natural Science Foundation of China (Grant 52171259), High-tech ship research project of the Ministry of Industry and Information Technology (Grant [2021]342).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Buixadé Farré, A.; Stephenson, S.R.; Chen, L.; Czub, M.; Dai, Y.; Demchev, D.; Efimov, Y.; Graczyk, P.; Grythe, H.; Keil, K. Commercial Arctic shipping through the Northeast Passage: Routes, resources, governance, technology, and infrastructure. *Polar Geogr.* **2014**, *37*, 298–324. [[CrossRef](#)]
2. Yu, L.; Wang, J.; Wang, S.; Li, H. Development strategy for polar equipment in China. *Strateg. Study Chin. Acad. Eng.* **2020**, *22*, 84–93. [[CrossRef](#)]
3. Teixeira, E.; Araujo, B.; Costa, V.; Mafra, S.; Figueiredo, F. Literature Review on Ship Localization, Classification, and Detection Methods Based on Optical Sensors and Neural Networks. *Sensors* **2022**, *22*, 6879. [[CrossRef](#)] [[PubMed](#)]
4. Mikko, S.; Fang, L.; Liangliang, L.; Pentti, K.; Anriette, B.; Jonni, L. Effect of Maneuvering on Ice-Induced Loading on Ship Hull: Dedicated Full-Scale Tests in the Baltic Sea. *J. Mar. Sci. Eng.* **2020**, *8*, 759.
5. Xie, C.; Zhou, L.; Ding, S.; Liu, R.; Zheng, S. Experimental and numerical investigation on self-propulsion performance of polar merchant ship in brash ice channel. *Ocean Eng.* **2023**, *269*, 113424. [[CrossRef](#)]
6. Escobar-Amado, C.D. *Deep Learning and Computer Vision Algorithms for Detection and Classification of Bearded Seal Vocalizations in the Arctic Ocean*; University of Delaware: Newark, DE, USA, 2022.

7. Ting, L.; Baijun, Z.; Yongsheng, Z.; Shun, Y. Ship Detection Algorithm based on Improved YOLO V5. In Proceedings of the 021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), DaLian, China, 15–17 July 2021; pp. 501–505.
8. Jin, W.; Changqing, C.; Yuedong, Z.; Xiaodong, Z.; Zhejun, F.; Qifan, W.; Ziqiang, H. Multiple Ship Tracking in Remote Sensing Images Using Deep Learning. *Remote Sens.* **2021**, *13*, 3601. [[CrossRef](#)]
9. Mingfeng, L.; Bo, L.; Shengzheng, W.; Jiansen, Z. Ship tracking and recognition based on Darknet network and YOLOv3 algorithm. *J. Comput. Appl.* **2019**, *39*, 1663–1668. [[CrossRef](#)]
10. Lu, W.; Lubbad, R.; Løset, S.; Skjetne, R. Parallel channel tests during ice management operations in the arctic ocean. In Proceedings of the Arctic Technology Conference, St. John's, N.L., Canada, 24–26 October 2016.
11. Cai, J.; Ding, S.; Zhang, Q.; Liu, R.; Zeng, D.; Zhou, L. Broken ice circumferential crack estimation via image techniques. *Ocean Eng.* **2022**, *259*, 111735. [[CrossRef](#)]
12. Panchi, N.; Kim, E.; Bhattacharyya, A. Supplementing remote sensing of ice: Deep learning-based image segmentation system for automatic detection and localization of sea-ice formations from close-range optical images. *IEEE Sens. J.* **2021**, *21*, 18004–18019. [[CrossRef](#)]
13. Du, X.; Tan, K.K. Vision-based approach towards lane line detection and vehicle localization. *Mach. Vis. Appl.* **2016**, *27*, 175–191. [[CrossRef](#)]
14. Zheng, F.; Luo, S.; Song, K.; Yan, C.-W.; Wang, M.-C. Improved lane line detection algorithm based on Hough transform. *Pattern Recognit. Image Anal.* **2018**, *28*, 254–260. [[CrossRef](#)]
15. Bar Hillel, A.; Lerner, R.; Levi, D.; Raz, G. Recent progress in road and lane detection: A survey. *Mach. Vis. Appl.* **2014**, *25*, 727–745. [[CrossRef](#)]
16. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial as deep: Spatial cnn for traffic scene understanding. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 27 April 2018.
17. Yoo, S.; Lee, H.S.; Myeong, H.; Yun, S.; Park, H.; Cho, J.; Kim, D.H. End-to-end lane marker detection via row-wise classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 6 May 2020; pp. 1006–1007.
18. Lee, D.-H.; Liu, J.-L. End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal Image Video Process.* **2023**, *17*, 199–205. [[CrossRef](#)]
19. HSVA. Brash Ice Tests for a Panamax Bulker with Ice Class 1B. Report, IO 509/12. 2013.
20. Ward, C.M.; Harguess, J.; Hilton, C. Ship classification from overhead imagery using synthetic data and domain adaptation. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; pp. 1–5.
21. Yang, Y.; Zhu, Y.; Sui, C. Study on Design and Production of Augmented Reality Work Integrated with Shadow Art Element. In Proceedings of the 2nd International Conference on Arts, Design and Contemporary Education, Moscow, Russia, 23–25 May 2016; pp. 638–641.
22. Qin, Z.; Wang, H.; Li, X. Ultra fast structure-aware deep lane detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 276–291.
23. Yousefi, J. *Image Binarization Using Otsu Thresholding Algorithm*; University of Guelph: Guelph, ON, Canada, 2011; Volume 10.
24. Bolya, D.; Zhou, C.; Xiao, F. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 24 October 2019; pp. 9157–9166.

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.