

Article

Marine Radar Oil Spill Detection Method Based on YOLOv8 and SA_PSO

Jin Xu ^{1,2,3,4,5,†} , Yuanyuan Huang ^{1,2}, Haihui Dong ^{1,2,*}, Lilin Chu ^{1,2,3,4,†}, Yuqiang Yang ^{1,3,4} , Zheng Li ^{1,2,3,4}, Sihan Qian ^{1,2} , Min Cheng ^{1,2}, Bo Li ^{1,2,3,4,5}, Peng Liu ⁶ and Jianning Wu ²

¹ Shenzhen Institute of Guangdong Ocean University, Shenzhen 518116, China; jinxu@gdou.edu.cn (J.X.); huangyuanyuan@stu.gdou.edu.cn (Y.H.); 18861913315@stu.gdou.edu.cn (S.Q.); 13625632485@stu.gdou.edu.cn (M.C.); boli@gdou.edu.cn (B.L.)

² Naval Architecture and Shipping College, Guangdong Ocean University, Zhanjiang 524091, China; sk.rt2@stu.gdou.edu.cn

³ Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching, Guangdong Ocean University, Zhanjiang 524088, China

⁴ Technical Research Center for Ship Intelligence and Safety Engineering of Guangdong Province, Zhanjiang 524088, China

⁵ Key Laboratory of Philosophy and Social Science in Hainan Province of Hainan Free Trade Port International Shipping Development and Property Digitization, Hainan Vocational University of Science and Technology, Haikou 570100, China

⁶ Navigation College, Dalian Maritime University, Dalian 116026, China; liupeng@dlmu.edu.cn

* Correspondence: donghh@gdou.edu.cn; Tel.: +86-138-2827-8906

† These authors contributed equally to this work.

Abstract: In the midst of a rapidly evolving economic landscape, the global demand for oil is steadily escalating. This increased demand has fueled marine extraction and maritime transportation of oil, resulting in a consequential and uneven surge in maritime oil spills. Characterized by their abrupt onset, rapid pollution dissemination, prolonged harm, and challenges in short-term containment, oil spill accidents pose significant economic and environmental threats. Consequently, it is imperative to adopt effective and reliable methods for timely detection of oil spills to minimize the damage inflicted by such incidents. Leveraging the YOLO deep learning network, this paper introduces a methodology for the automated detection of oil spill targets. The experimental data pre-processing incorporated denoise, grayscale modification, and contrast boost. Subsequently, realistic radar oil spill images were employed as extensive training samples in the YOLOv8 network model. The trained detection model demonstrated rapid and precise identification of valid oil spill regions. Ultimately, the oil films within the identified spill regions were extracted utilizing the simulated annealing particle swarm optimization (SA-PSO) algorithm. The proposed method for offshore oil spill survey presented here can offer immediate and valid data support for regular patrols and emergency reaction efforts.

Keywords: oil spill; marine radar; YOLO v8



Citation: Xu, J.; Huang, Y.; Dong, H.; Chu, L.; Yang, Y.; Li, Z.; Qian, S.; Cheng, M.; Li, B.; Liu, P.; et al. Marine Radar Oil Spill Detection Method Based on YOLOv8 and SA_PSO. *J. Mar. Sci. Eng.* **2024**, *12*, 1005. <https://doi.org/10.3390/jmse12061005>

Academic Editor: Merv Fingas

Received: 16 May 2024

Revised: 10 June 2024

Accepted: 13 June 2024

Published: 16 June 2024



Copyright: © 2024 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

Oil is one of the most vital energy sources for humans, and as economic development advances, so does the worldwide demand for oil [1]. This has already led to larger-scale offshore oil extraction and transportation activities. Oil spills can result from ship collisions during transport or from explosions on extraction platform [2]. Following an oil spillage incident on the ocean apparent, bumper oil sheen will quickly drift and spread due to the influence of waves and sea breezes, wreaking havoc on the marine ecology and the surrounding natural environment. In addition to gravely harming the marine environment and causing the deaths of numerous marine animals, plants, birds, fish, and mammals, oil spills worldwide also pose a grave risk to human health [3]. The cleanup process for oil spill pollution is time-consuming and labor-intensive, requiring a

substantial investment of financial, material, and human resources. Additionally, ocean regions contaminated by oil become unsuitable for tourism and fishing [4]. They also affect maritime traffic, complicating the transportation of shipping operations. A notable oil spill occurred at the Bohai Penglai 19-3 oilfield on 4 June 2011, which was being jointly developed by ConocoPhillips and CNOOC. This spill impacted 6200 square kilometers of ocean region, with 870 square kilometers experiencing severe pollution [5]. On 27 April 2021, approximately 9419 tons of cargo oil leaked into the Yellow Sea due to a collision between the Panamanian-flagged cargo ship *Sea Justice* and the Liberian-flagged oil tanker *Symphony*. This incident contaminated 786.5 km of coastline and 4360 square kilometers of ocean regions [6].

The nation will suffer the least amount of damage if oil spill monitoring data can be accessed quickly for emergency action plans [7–14]. Marine radar can be mounted on cleanup vessels for oil spill monitoring, which is ideal for real-time assisted pollution control. This oil spill monitoring technology comprises three main aspects: image preprocessing, extraction of meaningful oil spill surveillance regions, and oil film separation [15,16]. Image preprocessing involves denoise, gray scale modification, and contrast boost. Methods for extracting effective oil spill monitoring regions are mainly divided into two kinds: gray-scale distribution matrix and texture feature classification. Oil film segmentation currently involves adaptive thresholding and machine learning classification methods. Oil spill radar image treatment is a tremendous question, apart from the above, and many scholars have conducted research on various aspects [17–20].

The application of deep learning in marine radar oil spill monitoring technology is infrequent. We employed the YOLOv8 model here for marine radar oil spill region detection. The experimental results illustrated that the YOLOv8 model can accurately and swiftly extract the effective oil spill monitoring regions. In this paper, the oil film segmentation is conducted by using the simulated annealing particle swarm optimization (SA_PSO) method. While the particle swarm optimization (PSO) method offers numerous advantages, including simple implementation with few parameters, straightforward convergence with strong robustness, and wide applicability, it does have a drawback: a tendency to settle into local optimal solutions. To address this limitation, the simulated annealing (SA) algorithm's competence to flee locally optimal solutions is incorporated into the PSO algorithm, facilitating faster escape from local optima while searching for the best solution. The experimental results corroborated the practicality of the SA_PSO methodology for oil film segmentation.

2. Materials and Methods

2.1. Materials

The analysis data were acquired by a marine radar transceiver, which was integrated with a computer system featuring a monitor that displayed the processed imagery, as shown in Figure 1. The short wavelength of the X-band radar produces finer echo signals, resulting in high resolution and the ability to detect targets at longer distances. Additionally, X-band radar has strong penetration capabilities, enabling it to penetrate rain and snow. The X-band marine radar was chosen as the platform for experimental data collection here.

With the rotation of the X-band horizontal polarization radar antenna, the system was capable of capturing and storing digital representations of the clutter signals. It could capture between 28 and 45 images in one minute. The radar measurement distance is adjustable through variations in pulse width. To enhance the image resolution of oil film targets, the experimental data detection range was specifically set to 0.75 nautical miles, resulting in images with a resolution of 1024×1024 pixels. The type of oil spills in the experimental data were crude oil spilled from oil tankers and storage tanks at oil terminals.

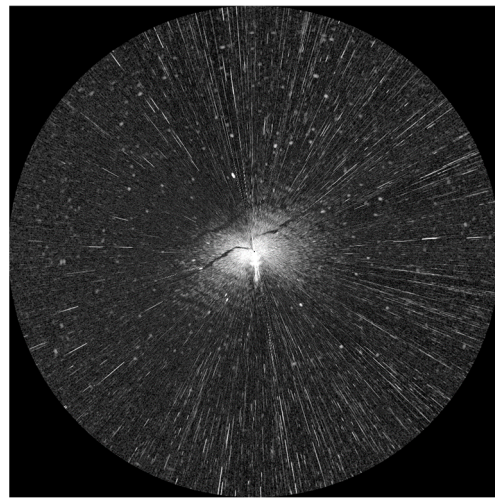


Figure 1. The experimental marine radar remote-sensing image in the polar coordinate system.

2.2. Experimental Process

The experimental process is shown in Figure 2. First, the original marine radar images were preprocessed. Afterwards, the preprocessed images were loaded into the YOLOv8 model for training to generate an oil film prediction model. Furthermore, the new preprocessed images were input into the prediction model to obtain the oil film detection result images. Then, the SA_PSO algorithm was used to preliminarily segment oil film targets. Finally, speckle noises were removed to obtain the final segmentation results.

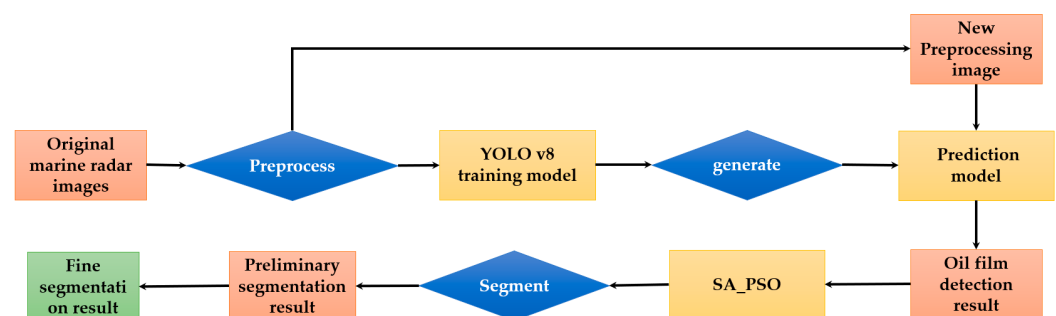


Figure 2. Experimental process.

2.3. Data Preprocessing

The data preprocessing process is shown in Figure 3. The detailed process includes:

- The original image under the polar coordinate system was converted into Cartesian coordinate system, as shown in Figure 4a.
- The data in Cartesian coordinate system were convolved with the row vectors of $[-1, -1, 4, -1, -1]$. The co-frequency interference noise pixels in the Cartesian coordinate system exhibit bright features compared to adjacent horizontal pixels. Thus, the above row vector that highlights the central pixel was used to enhance co-frequency interference noises.
- The co-frequency interferences were extracted according to the gray threshold. After this, the mean filter was accessed to smooth the co-frequency interference noises [15].
- The smoothed co-frequency interference image was binarized again using the gray threshold [16].
- The speckle noises were extracted by the pixel-quantity threshold.
- The median filter of the 20×20 window was used to remove speckle noises, as shown in Figure 4b.
- The noise reduction image was processed for gray correction, as shown in Figure 4c.

h. An overall grey contrast enhancement was applied to the image, as shown in Figure 4d.

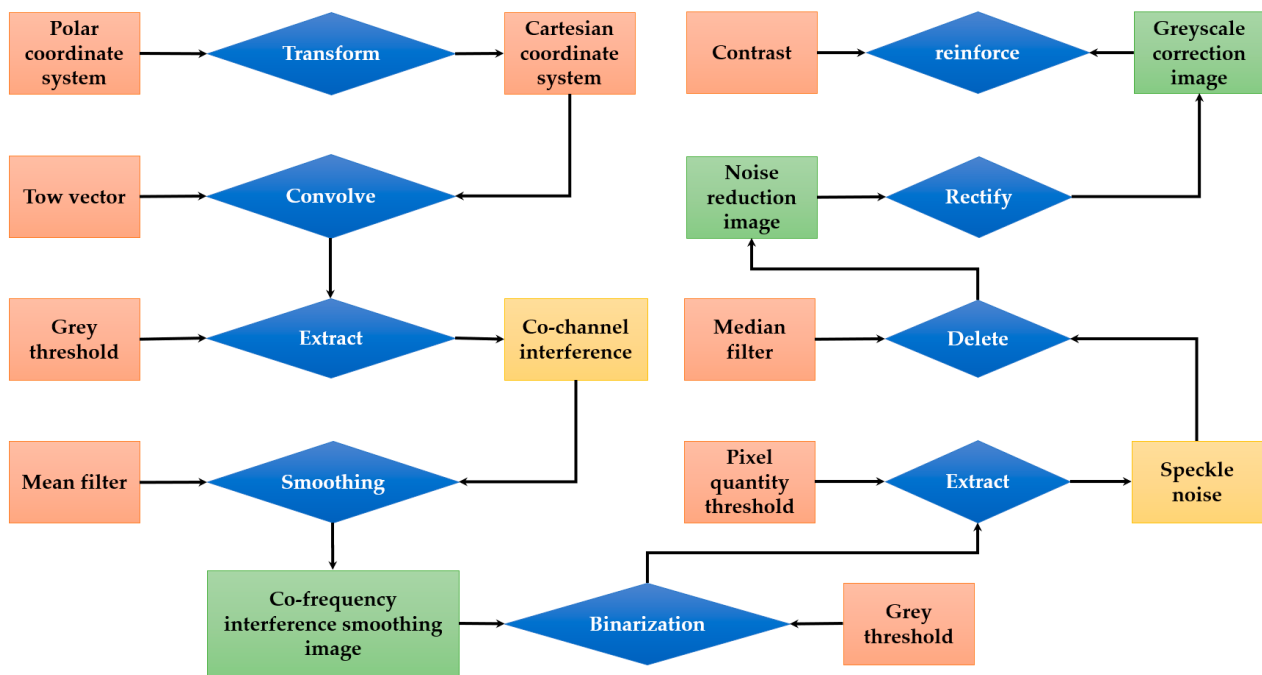


Figure 3. Data preprocessing scheme.

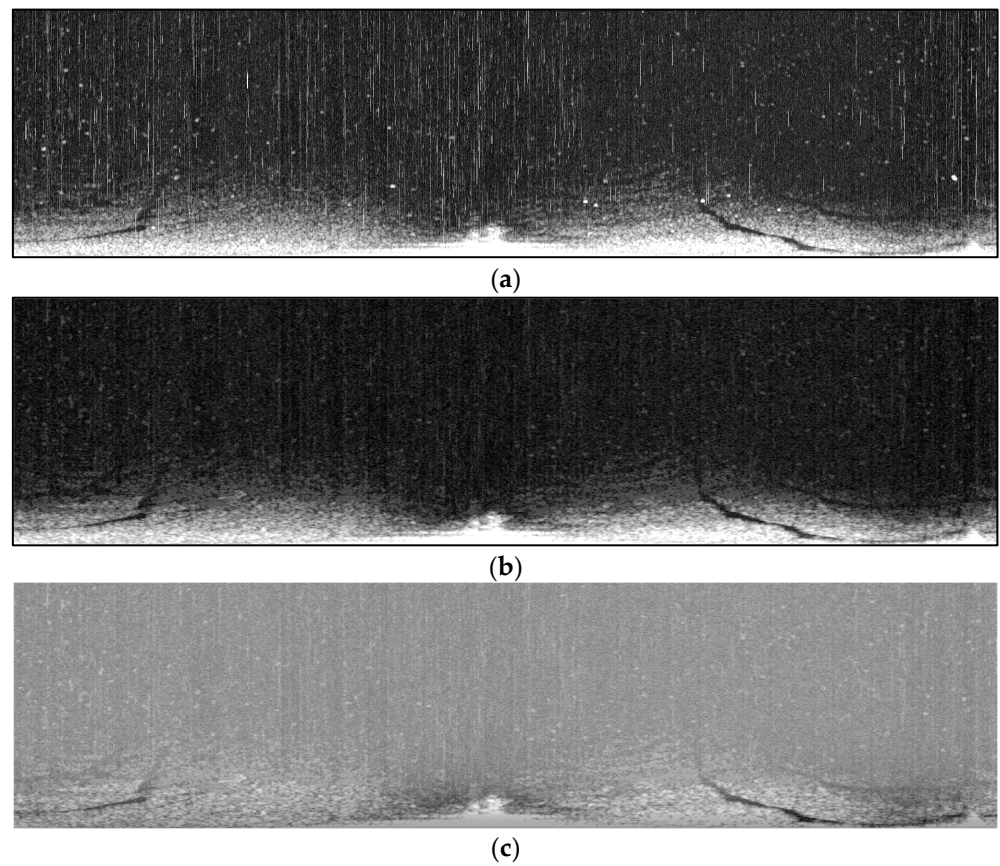


Figure 4. Cont.

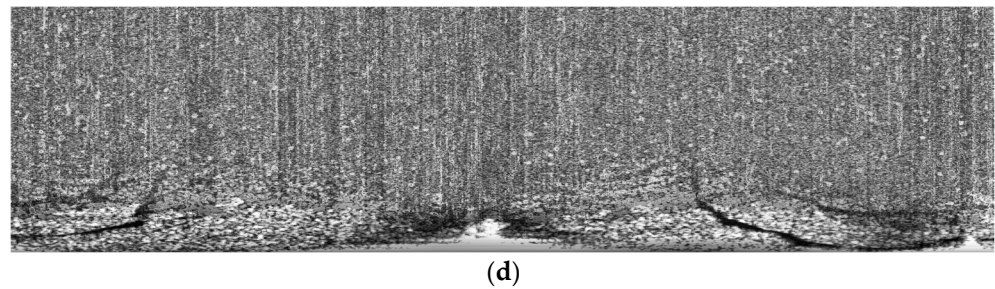


Figure 4. The preprocess: (a) Cartesian coordinate system conversion; (b) noise reduction; (c) gray correction; and (d) local contrast enhancement.

2.4. YOLO

Spaceborne and airborne oil spillage detection utilizing deep learning techniques has emerged as a dominant methodology [21]. Nevertheless, the implementation of deep learning methods for marine radar oil spillage detection remains limited. Deep learning technology has the capability to automatically extract oil film characteristics from remote sensing data, eliminating the need for intricate algorithmic designs. The development of a target detection network mainly focuses on two directions: two-stage algorithms and one-stage algorithms [22]. The main difference between them is that two-stage algorithms need to use a feature extractor to generate a series of pre-selected boxes that may contain the objects to be detected, then apply certain filtering rules to filter the pre-selected frames [23]. By contrast, the one-stage algorithm can extract features directly in the network to predict object classification and location. Because of the constraints related to computer memory usage and communication expenses, the two-step approach is less favored compared to the one-stage approach. The one-step approach is suitable for real-time dynamic monitoring due to its precision and low resource consumption, which can meet the significant demand for oil spill cleanup and management. Hence, the one-step YOLOv8 model was employed here for marine radar oil spillage detection.

The YOLOv8 structure consists of the following four primary components: Input, Backbone Network, Neck, and Output (Figure 5). The input side is responsible for receiving raw images and pre-processing them. In YOLOv8, the input side usually uses Mosaic data enhancement technique to stitch multiple images for increasing the diversity of the training data. In addition, preprocessing operations, such as adaptive image scaling and gray-scale filling, are performed to resize the image into the input size and format required by the network. The backbone network is the core part of YOLOv8 and is responsible for extracting features from the input images. In the backbone network, a series of convolutional layers, pooling layers, and other operations are usually employed to gradually extract feature information from the image. Common backbone network structures in YOLOv8 include Convolutional Layer (Conv), Connect to Fusion (C2F), and Spatial Pyramid Pooling (SPPF), which can effectively capture semantic and spatial information in the image. The Neck end is responsible for further processing and fusion of the features extracted from the backbone network. In YOLOv8, the Neck side is usually designed based on the Path Aggregation Network (PAN) structure, which fuses feature maps at different scales through operations such as up-sampling, down-sampling, and feature splicing, so as to improve the network's ability to detect and recognize targets. The output side is the last part of YOLOv8 and is responsible for generating detection results and outputting them. In the output side, a decoupled head structure is usually used to decouple the classification and regression processes to improve the efficiency and accuracy of the model. In addition, the output side includes steps, such as positive and negative sample matching and loss calculation, in order to evaluate and optimize the detection results. The YOLOv8 network uses the Task Aligned Assigner method to weight the classification scores and the regression scores. The loss calculation contains classification calculation and regression loss calculation. Binary

Cross-Entropy (BCE) is used to calculate classification loss and Complete Intersection over Union (CIoU) loss functions.

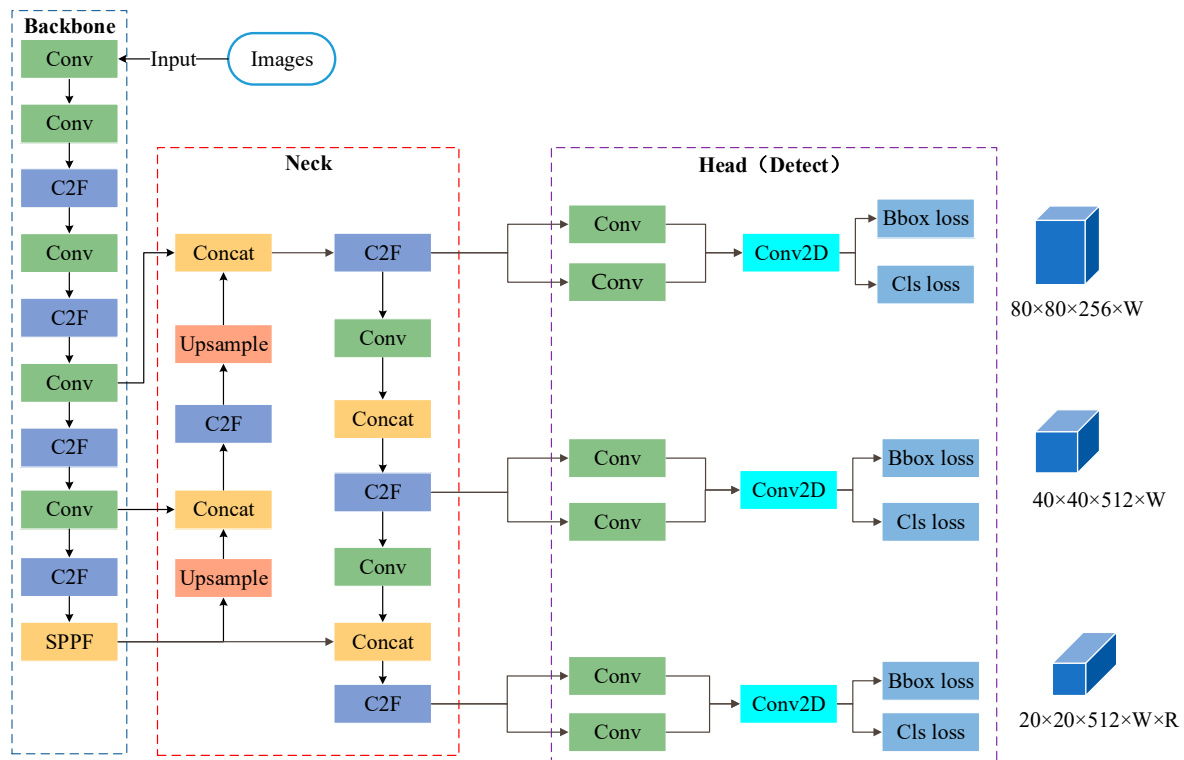


Figure 5. The YOLOv8 model architecture.

2.5. Sample Labeling Tool

LabellImg is an open-source image annotation tool designed specifically for creating bounding boxes and assigning labels to objects in images. Developed primarily in Python and utilizing the Qt framework for its Graphical User Interface (GUI), LabellImg provides a friendly operating platform for efficiently annotating large datasets for object detection and localization tasks. Bounding boxes around objects can be quickly and precisely drawn with LabellImg. The annotations in various formats, including the PASCAL VOC XML format, YOLO, and CreateML, can also be exported correspondingly. Labeling was used here to mark oil film targets in marine radar images, as shown in Figure 6.



Figure 6. Sample labeling method.

2.6. Model Training Evaluation Indicators

The model training evaluation indicators include the Precision-Confidence Curve and the Recall-Confidence Curve.

a. The Precision-Confidence Curve

YOLO assigns a confidence score to each predicted bounding box, representing the model certainty. The Precision-Confidence Curve in YOLO illustrates the connection between the precision of predicted bounding boxes and their confidence scores. Precision in object detection refers to the percentage of true positive predictions among all positive predictions, indicating how accurately the model predicts the existence of an object inside a bounding box. As the confidence threshold increases, the model becomes more selective, predicting fewer bounding boxes. This may result in lower recall but higher precision. Conversely, decreasing the threshold leads to the model predicting more bounding boxes, potentially increasing precision due to more false positives but also increasing recall;

b. The Recall-Confidence Curve

The Recall-Confidence Curve captures the relationship between the model confidence in its predictions and recall, or ability to detect true objects. The recall gauges the percentage of real objects that the model correctly identifies. The Recall-Confidence Curve displays the recall value against the confidence threshold. As the confidence threshold rises, the model becomes more selective and only predicts bounding boxes with higher confidence scores. However, this may lead to certain real objects being missed due to lower confidence scores, resulting in a decrease in recall.

2.7. Particle Swarm Optimization

In PSO models, a group of particles, firstly randomly initialized without volume or mass, are considered as feasible solutions to an optimization problem [24]. Subsequently, the algorithm iteratively searches for the optimal solution. During each iteration, particles update themselves by tracking two extreme values: the best solution found by the particle itself, known as individual optimal solution $P_{best,i}$ and the second the best solution found by the entire population, known as the global best G_{best} . The particle performs the selection of velocity v and position x by the following:

$$v_i(k+1) = \omega v_i(k) + c_1 r_1 (P_{best,i}(k) - x_i(k)) + c_2 r_2 (G_{best} - x_i(k)) \quad (1)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (2)$$

where ω is the inertia weight coefficient. c_1 and c_2 are used to control the influence of individual best position and global best position on particle movement. By adjusting the values of c_1 and c_2 , a balance between local and global search can be achieved, thus enhancing the algorithm performance and convergence speed. $v_i(k)$ and $x_i(k)$ denote the velocity and position of the particle at the k^{th} iteration. r_1 and r_2 are random numbers uniformly distributed in $[0, 1]$. If any dimension of a particle position exceeds the specified upper bound (Upbound) or lower bound (Low bound), it is directly assigned the value of Upbound or Low bound, respectively. Due to the normalization of the grayscale values of the image, the Upbound was set to '1' and the Low bound was set to '0' here. The ω determines the tendency of particles to maintain their current velocity. A higher ω value makes particles more inclined to retain their previous velocities, which helps the algorithm move quickly across the search space to explore new regions. However, if ω is set too high, it can lead to instability in the search process, and the algorithm may even miss the global optimal solution. When $\omega = 1.1$, the PSO algorithm exhibits faster exploration ability through experimental comparisons. It enables the PSO algorithm to find candidate solutions closer to the global optimal solution more quickly, and a stable oil film threshold can be obtained.

2.8. Simulated Annealing

SA is a probability based global optimization algorithm that simulates the process of material annealing and cooling at high temperatures [25]. It is widely used in the field of signal processing [26]. The principle of the classic simulated annealing algorithm is as indicated below:

- a. The beginning temperature T_0 is set for the initial state of the object.
- b. Given E_g represents the internal energy of the current best point in the population, T_i represents the current temperature, and the energy of a new state i after iteration is E_i . If E_g is greater than E_i , the new state is accepted as the current state. Then $E_g = E_i$. Otherwise, the state is accepted with a certain probability determined by the acceptance probability formula p_i . The expression for p_i is as follows:

$$p_i = \begin{cases} 1 & E_i < E_g \\ \exp\left(-\frac{E_i - E_g}{T_g}\right) & E_i \geq E_g \end{cases} \quad (3)$$

- c. If the new state is accepted, the new temperature decreases gradually as:

$$T_{t+1} = \alpha T_t \quad (4)$$

where α is the cooling attenuation factor, $0 < \alpha < 1$, T_{t+1} is the new temperature, T_t is the temperature of last state.

- d. Back to step 2 and repeat the iteration process until the termination condition is met, such as several consecutive new solutions not being accepted, reaching the preset number of iterations, or temperature threshold, etc.
- e. When the algorithm stops, the current solution E_g is output as the optimal solution.

2.9. Simulated Annealing Particle Swarm Optimization

As an efficient intelligent optimization algorithm, the PSO algorithm requires minimal parameter tuning, which is easily implementable. It avoids complex operations and can swiftly solve intricate optimization problems. However, when dealing with optimization problems featuring multiple local optima, the algorithm is prone to getting stuck in local optima, resulting in slow convergence. To address this issue, when particles in the population become trapped in local optima, efforts should be made to facilitate their escape from these local optima, thereby enhancing the diversity of the entire population. Considering that the SA algorithm can probabilistically accept suboptimal solutions during the optimization process, it can effectively prevent the algorithm from getting trapped in local optima during iterative searches. Therefore, we integrated the core principles of the SA into the PSO algorithm for finding the segmentation threshold between real and suspected oil spills, as shown in Figure 7.

- a. Set an initial temperature T_0 and cooling attenuation factor α based on the initial state of particles in the population. The T_0 is a crucial parameter that determines the hotness of the algorithm during the search process. Typically, the initial temperature needs to be set high enough to ensure that the algorithm can escape from local optima during the initial search phase and prevent premature convergence. The value of '100', as a relatively large number, often satisfies this requirement. The value of α determines the rate at which the temperature decreases in each iteration step. A smaller value of α , such as '0.95', implies a relatively slower decrease in temperature, which allows the algorithm to explore the search space for a longer time, thereby increasing the possibility of finding the global optimal solution. So, the T_0 was set to '100', the α was set to '0.95' here.

- b. Particle i computes its velocity v_i and position x_i through Formula (1).

- c. Calculate the fitness $f(x_i)$ of the current position x_i based on the evaluation objective function.

- d. If $f(x_i) < f(P_{best.i})$, then set $P_{best.i} = x_i$. If $f(x_i) > f(P_{best.i})$, utilizing the above-mentioned simulated annealing acceptance probability p_i to apply the Metropolis criterion [27]. If the

probability p_i is greater than a random number within the range $[0, 1]$, then the state $P_{best.i}$ is still accepted. If this condition is not met, then the velocity v_i and position x_i will be recomputed.

e. If $P_{best.i}$ is accepted as the new value, then $f(P_{best.i})$ and $f(G_{best})$ are compared according to the evaluation objective function. If $f(P_{best.i}) > f(G_{best})$, then $G_{best} = P_{best.i}$.

f. If the number of iterations reaches the maximum iteration J (The J was set to '100' here), G_{best} is determined as the threshold for the particle swarm optimization algorithm. If the condition is not met, proceed back to step (2) for further iterations.

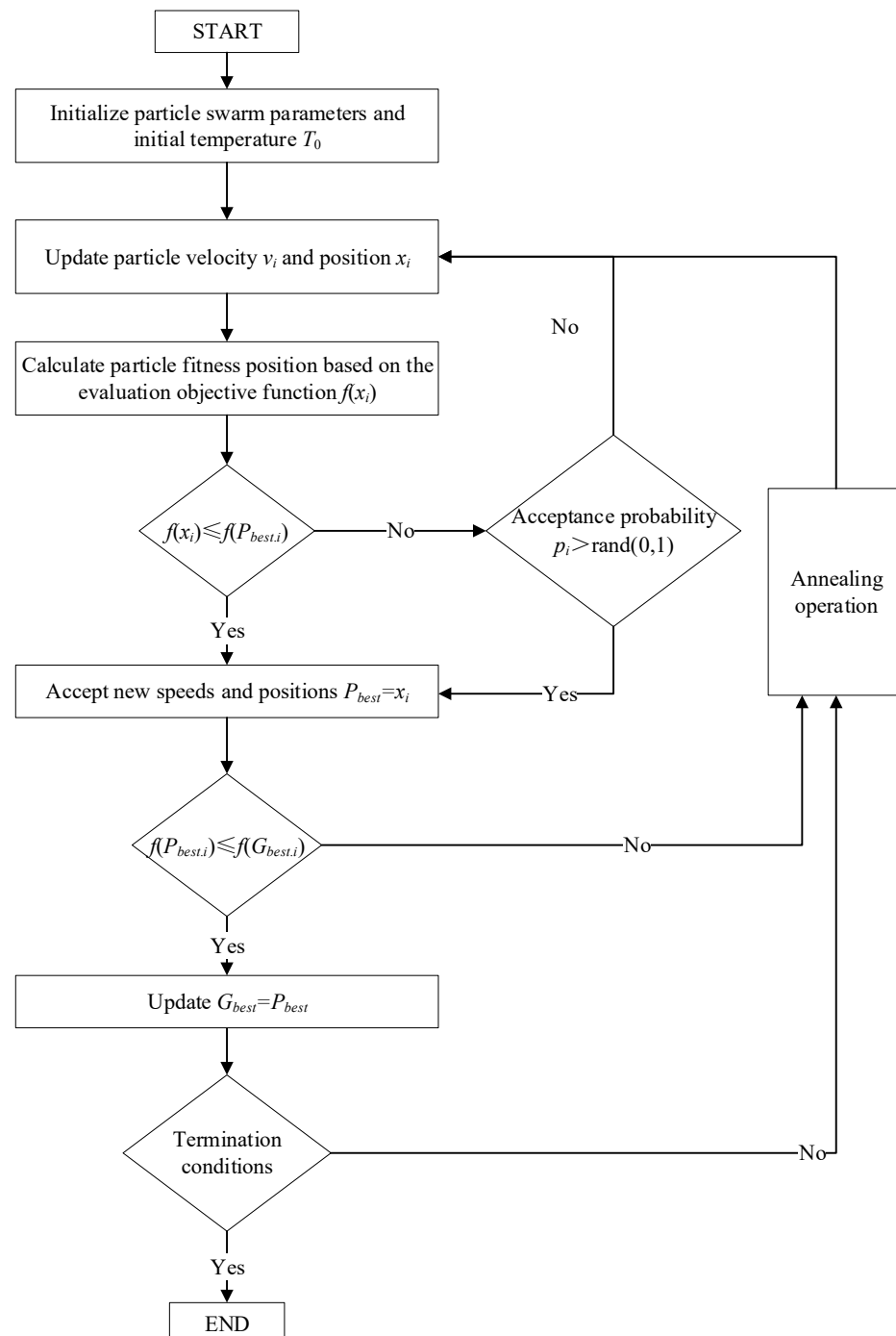


Figure 7. SA_PSO algorithmic process.

Ultimately, based on SA_PSO method, dual thresholds were calculated iteratively for separating the real and the suspected oil films here, respectively.

3. Results

3.1. Model Training Curve Analysis

The Precision-Confidence Curve and the Recall-Confidence Curve are shown in Figure 8. As the confidence level increases, so does the accuracy of the model in detecting oil sheen. At a confidence threshold of '0.846', the precision rate stood at '100%', as reflected in Figure 8a. Due to insufficient sample size, the model may not be able to fully learn the distribution of data, resulting in unstable performance on validation or testing sets. This is the reason why the confidence curve oscillates between '0.6' and '0.8'. The collection of oil spill data from maritime radar is an ongoing work in the future. The model's ability to identify oil slicks improves as the single recall value increases. The model had a single recall of '0.92', indicating that the model exhibits high accuracy in oil identification and localization, as shown in Figure 8b.

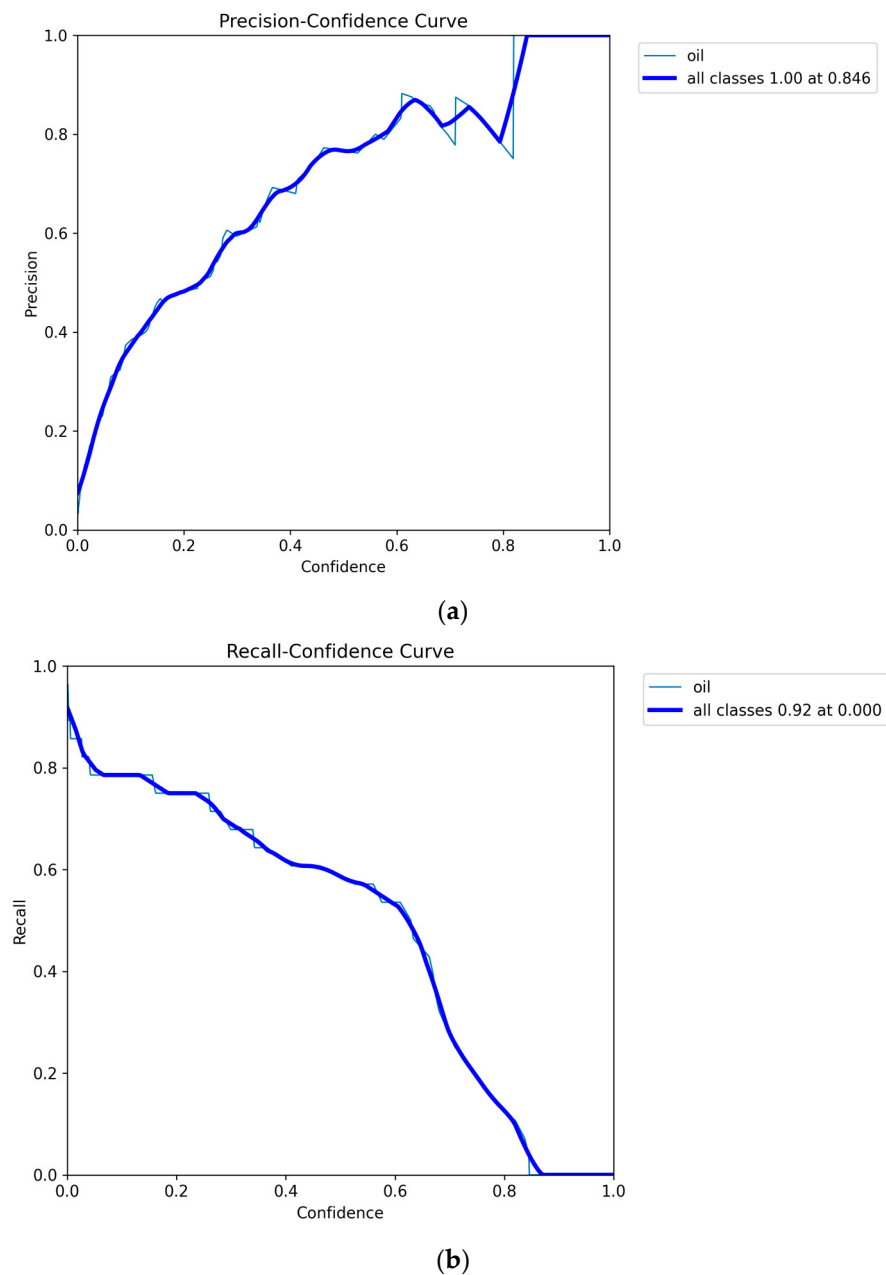


Figure 8. YOLOv8 model training curve: (a) the Precision-Confidence Curve; and (b) the Recall-Confidence Curve.

3.2. Oil Film Prediction

The YOLOv8n model, which was trained here, was employed to predict the oil slick targets within the newly preprocessed marine radar image and the results obtained were shown in Figure 9. The location and size of the effective oil spill monitoring area marked in red in Figure 9a were subsequently extracted in Figure 9b.

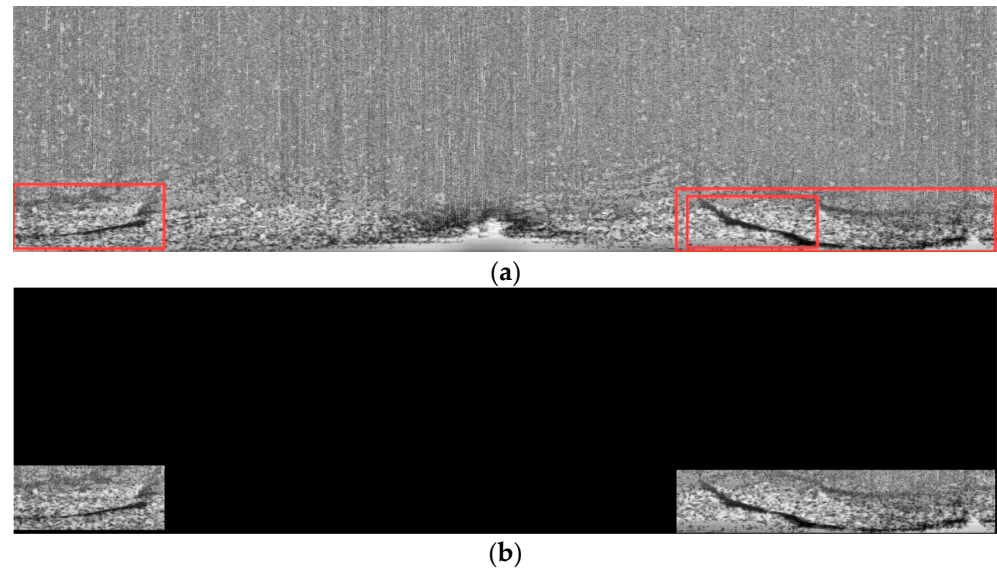


Figure 9. YOLOv8 model prediction results: (a) preliminary detection result; and (b) the oil film regions were preserved.

3.3. Oil Film Segmentation

The classification result was obtained by using SA_PSO, as shown in Figure 10a. Then, the true oil spills were preserved by removing the sparkles, as shown in Figure 10b. Finally, the oil film identification image was converted from the Cartesian coordinate to the Polar coordinate, as shown in Figure 10c.

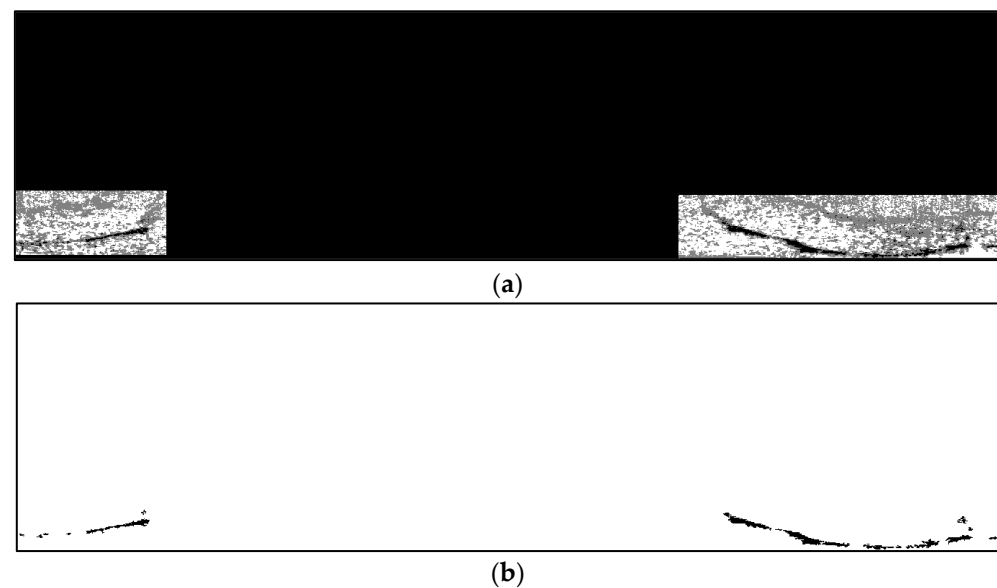


Figure 10. Cont.

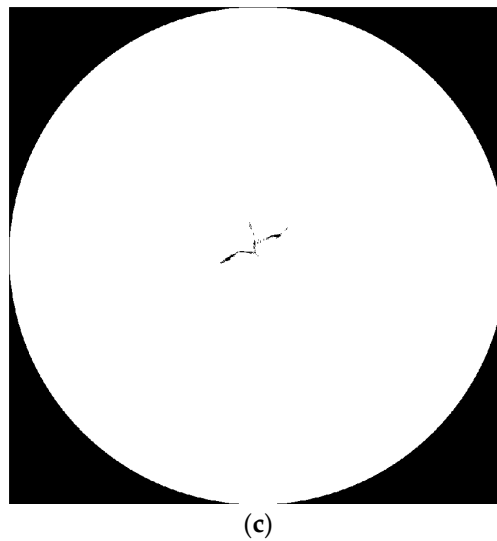


Figure 10. The oil spill segmentation results: (a) the segmentation result of SA_PSO; and (b) the real oil pill result; (c) The polar coordinates result.

4. Discussion

4.1. Comparison of Prediction Results of Different Training Models

There are five official versions of YOLOv8: 8n, 8s, 8m, 8l, and 8x. Since the 8l and 8x models were too large, only the 8n, 8s, 8m models were trained here for comparison. The three training models were used to predict oil spill targets on preprocessed marine radar images, as shown in Figure 11. The prediction times are displayed in Table 1. The average detection speed of the 8n model for each image was 24.9 ms, with fine prediction results. The effective oil spill monitoring regions were detected, while the ship wake region was not misidentified in Figure 9b. The 8s model exhibited an even detection tempo of 49.8 ms. The detection performance of the oil spill effective monitoring area was excellent, but the ship wake region was misidentified as the oil spill region in Figure 11a. The 8m model exhibited an even detection tempo of 49.8 ms. Although the ship wake region was not misidentified, some effective oil spill detection regions were missed (blue box), as shown in Figure 11b. Therefore, the YOLOv8n model was chosen for subsequent oil film segmentation work in this paper.

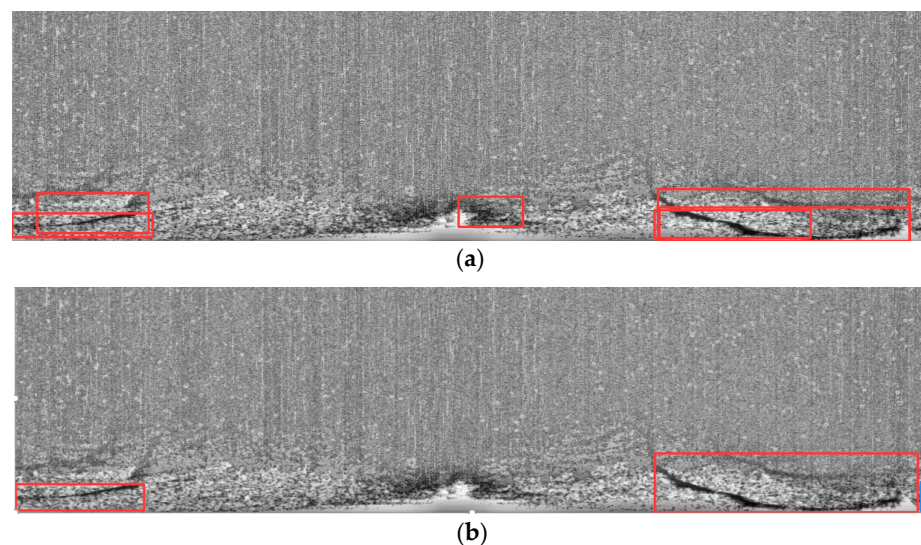


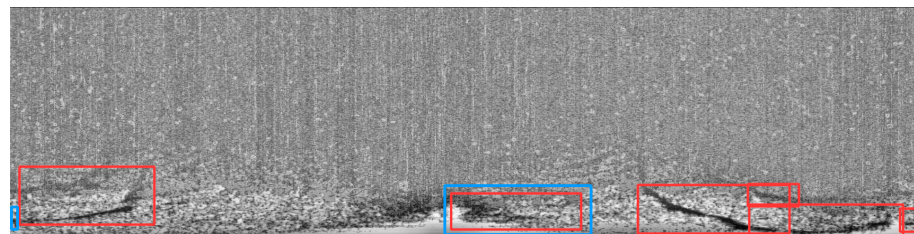
Figure 11. The prediction results of different training models in red color: (a) YOLOv8s model; and (b) YOLOv8l model.

Table 1. The prediction times of different training models.

Training Model	Forecast Time
YOLOv8n	24.9 ms
YOLOv8s	49.8 ms
YOLOv8m	109.6 ms

4.2. Comparison with the Prediction Result of YOLOv5n Model

The YOLOv5n model, after being trained, was utilized for predicting the oil spill targets in Figure 4d with a computational time of 1.3 ms, as shown in Figure 12. The YOLOv5n model detects the ship wake as oil spill targets which marked in blue box in the middle of Figure 12. The YOLOv8n model used here did not misidentify the ship wake region as oil spill targets in Figure 9a. The trained YOLOv5n model did not detect the small oil spill target in the left corner of Figure 12, whereas the method in this paper was more comprehensive.

**Figure 12.** The YOLOv5n model prediction result in red color.

4.3. Comparison with U-Net Semantic Segmentation Network

The U-Net of PyTorch is a deep learning architecture for image semantic segmentation tasks. It combines encoder and decoder for semantic segmentation of high-resolution input images down to the pixel level. The encoder extracts the features through convolution and pooling operations while reducing the spatial resolution. By employing up-sampling and inverse convolution techniques, the decoder diminishes the eigenmaps output from the encoder to match the dimensions of the original input image. This section combines the eigenmaps from the respective encoder layers to produce the segmentation outcomes. The uniqueness of U-Net is that it introduces jump connections, which connect the feature maps of each layer in the encoder to the feature maps of the corresponding layer in the decoder, helping the network to better recover detailed information.

In reference [28], the number of U-Net training epochs was set to '200', the batch size was set to '1', the learning rate was set to ' 10^{-5} ', the validation dataset accounted for 10%, the number of classes was set to '3', the prediction computing time was 2.56 s. The same parameter settings were used here for comparative discussion. The experimental results are shown in Figure 13.

The target detection network is faster than the semantic segmentation network, but it must be combined with other algorithms or models to segment the oil films. The semantic segmentation network directly assigns image pixels to different categories for achieving accurate segmentation of the oil spills at once. However, the semantic segmentation network has very high requirements for the sample labelling, and the suspected oil slicks are the most difficult to mark in the marine radar training images. Thus, many suspected oil slick segmentations were generated in green during our experiment. In terms of the number of real oil film pixels detected in red, due to Figure 13b identifying false positive targets in the ship wake region and sparkles as real oil films, many more real oil film image pixels were identified, as shown Table 2.

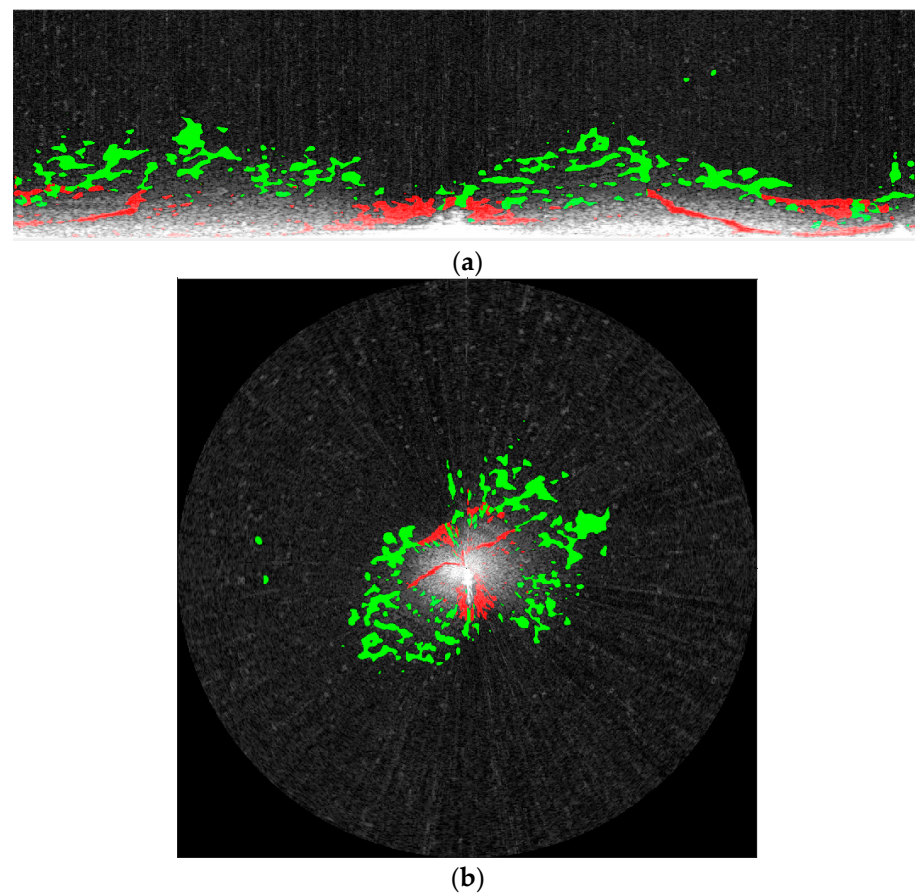


Figure 13. The U-Net segmentation results: (a) Cartesian coordinate system; and (b) Polar coordinate system.

Table 2. Comparison with the detection pixel numbers of real oil films.

Method	Detection Pixel Number of Real Oil Films
Proposed method	304
U-Net	5384

4.4. Comparison of Performance with Other Machine Learning Threshold Segmentation Methods

Three machine learning methods were used to compare with the SA-PSO method. The first one is the traditional PSO method [29]. The second one is the FCM-based marine radar oil spill segmentation method [16]. The last one is the traditional K-Means method [30]. The segmentations of 4 methods of Figure 9b shown similar results. Because there were little wave echoes in the far range in the experimental data, the upper 2/3 region of the contrast-enhanced image (Figure 4d) was uniformly assigned the same gray value ‘128’, as shown in Figure 14.



Figure 14. Contrast enhanced image with no-wave-region information removed.

Figure 14 was segmented by the above four methods and the results obtained were shown in Figure 15. All the methods identified some ship wake regions as oil slick targets. Among them, there were fewer errors of SA-PSO method. The initial segmentation of SA-PSO method also contained fewer noise results. The corresponding results of PSO and FCM methods were similar with more noises. However, the K-Means method became excessively noisy. The K-Means method segmented suspected oil slicks with weak image features into real ones (in red boxes), while also adding many blocky false targets. The PSO and FCM methods also determined some of the suspected oil films as real ones (in red boxes). The SA-PSO preliminary segmentation were relatively accurate and can show better results by simply excluding speckle noises.

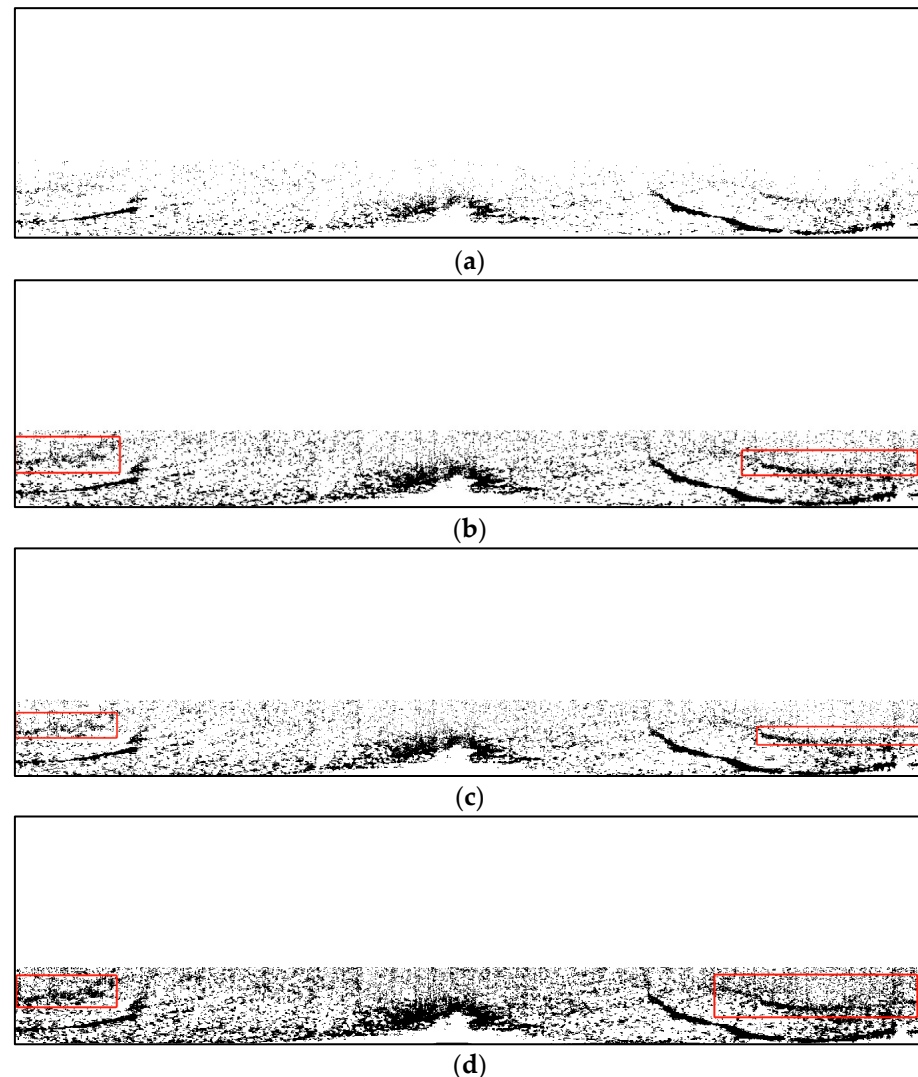


Figure 15. Comparison of four machine learning threshold segmentation methods: (a) SA-PSO; (b) PSO; (c) FCM; and (d) K-means.

4.5. Validation of Experimental Results

During the daytime, the accuracy of marine radar oil spill detection can be judged by visual observation or unmanned aerial vehicle (UAV) visible light data, as shown in Figure 16a. During the nighttime, infrared or laser fluorescence data can be used to verify the accuracy of marine radar oil spill monitoring method. Our marine radar experimental data were collected at night, and the corresponding thermal infrared images (Figure 16b) were obtained. In the thermal infrared images, the gray value of the oil film is slightly lower than that of water [31]. The thermal infrared images corresponding to the oil spill

locations in the experimental data also show the characteristics of the oil films, which can assist in verifying the feasibility of our method.

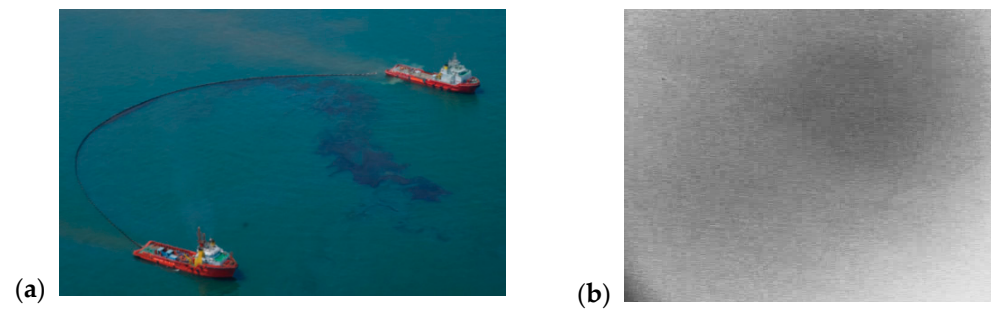


Figure 16. Auxiliary verification methods for oil spill detection: (a) visible light data can verify the performance of marine radar oil spill monitoring; and (b) Infrared data can be used to verify the oil spill detection results of maritime radar at night.

4.6. The Impacts of Weather Conditions on Marine Radar Oil Spill Detection

The marine radar oil spill monitoring technology has the capability to detect oil spills in a large range and has a relatively low construction cost. Despite its promising application prospects, this technology is still constrained by weather conditions. When the weather causes excessive ocean waves, ships equipped with marine radar are unable to go out to sea to perform pollution clean-up tasks. When the sea surface is too calm, the radar cannot retrieve certain wave echoes, making it difficult to identify the oil films. Moreover, the ability of the oil films to absorb waves is weak in both cases. In addition, for deep learning networks, training and detection data under the same sea conditions (including sea breeze) can achieve better results. To establish a comprehensive oil spill detection model, continuous collection of marine radar oil spill data under different sea conditions is needed.

5. Conclusions

In this study, the preprocessed images of marine radar oil spills underwent training using the YOLOv8 model. In contrast to prior methodologies, the proposed approach exhibited a notable improvement in detection efficiency. Solely tasked with pinpointing the location and dimensions of oil spills, the YOLOv8 deep learning network was exclusively employed. The SA_PSO algorithm was applied for the final segmentation. As more marine radar oil spill images are gathered across diverse oceanic conditions and deep learning networks are refined, enhanced techniques for oil spill detection will be realized. The future research focus will focus on how to distinguish offshore oil films from false positive targets such as shadows, wind, and plumes, etc.

Author Contributions: Conceptualization, J.X. and Y.H.; methodology, J.X., Y.H. and H.D.; software, L.C. and Y.Y.; validation, L.C. and Z.L.; formal analysis, S.Q., M.C. and B.L.; investigation, J.X., Y.H. and L.C.; resources, J.X., B.L. and P.L.; data curation, J.X. and H.D.; writing—original draft preparation, Y.H. and J.X.; writing—review, B.L. and H.D.; visualization, L.C.; supervision, Y.H. and J.W.; project administration, H.D. and J.X.; funding acquisition, H.D. and J.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Guangdong Province, grant numbers 2022A1515011603, 2023A1515011212, the Special Projects in Key Fields of Ordinary Universities in Guangdong Province, grant numbers 2021ZDZX1015, 2022ZDZX3005, the Natural Science Foundation of Shenzhen City, grant numbers JCYJ20220530162200001, JCYJ20210324122813036, Program for scientific research start-up funds of Guangdong Ocean University, grant numbers 060302132009, 060302132106, Postgraduate Education Innovation Project of Guangdong Ocean University grant numbers 202421, 2023XSLT_032, National Natural Science Foundation of China, grant number 52271359.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Armaroli, N.; Balzani, V. The Future of Energy Supply: Challenges and Opportunities. *Angew. Chem. Int. Ed. Engl.* **2007**, *46*, 52–66. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Prabowo, A.R.; Bae, D.M. Environmental Risk of Maritime Territory Subjected to Accidental Phenomena: Correlation of Oil Spill and Ship Grounding in the Exxon Valdez's Case. *Results Eng.* **2019**, *4*, 100035. [\[CrossRef\]](#)
3. Salomidi, M.; Katsanevakis, S.; Borja, A.; Braeckman, U.; Damalas, D.; Galparsoro, I.; Mifsud, R.; Mirto, S.; Pascual, M.; Pipitone, C.; et al. Assessment of Goods and Services, Vulnerability, and Conservation Status of European Seabed Biotopes: A Stepping Stone towards Ecosystem-Based Marine Spatial Management. *Mediterr. Mar. Sci.* **2012**, *13*, 49. [\[CrossRef\]](#)
4. Sherman, E.F. Specific Oil Spill Incidents. In *Managing the Risk of Offshore Oil and Gas Accidents*; Edward Elgar Publishing: Cheltenham, UK, 2019; pp. 408–446. ISBN 9781786436740.
5. Yin, J.; Feng, J.; Wang, Y. Social Media and Multinational Corporations' Corporate Social Responsibility in China: The Case of ConocoPhillips Oil Spill Incident. *IEEE Trans. Prof. Commun.* **2015**, *58*, 135–153. [\[CrossRef\]](#)
6. Cao, R.; Rong, Z.; Chen, H.; Liu, Y.; Mu, L.; Lv, X. Modeling the Long-Term Transport and Fate of Oil Spilled from the 2021 A Symphony Tanker Collision in the Yellow Sea, China: Reliability of the Stochastic Simulation. *Ocean Model.* **2023**, *186*, 102285. [\[CrossRef\]](#)
7. Krig, S. Image Pre-Processing. In *Computer Vision Metrics*; Springer International Publishing: Cham, Switzerland, 2016; pp. 35–74. ISBN 9783319337616.
8. Ullah, Z.; Farooq, M.U.; Lee, S.-H.; An, D. A Hybrid Image Enhancement Based Brain MRI Images Classification Technique. *Med. Hypotheses* **2020**, *143*, 109922. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Sonali; Sahu, S.; Singh, A.K.; Ghreera, S.P.; Elhoseny, M. An Approach for De-Noising and Contrast Enhancement of Retinal Fundus Image Using CLAHE. *Opt. Laser Technol.* **2019**, *110*, 87–98. [\[CrossRef\]](#)
10. Al-Ruzouq, R.; Gibril, M.B.A.; Shanableh, A.; Kais, A.; Hamed, O.; Al-Mansoori, S.; Khalil, M.A. Sensors, Features, and Machine Learning for Oil Spill Detection and Monitoring: A Review. *Remote Sens.* **2020**, *12*, 3338. [\[CrossRef\]](#)
11. Conceição, M.R.A.; de Mendonça, L.F.F.; Lentini, C.A.D.; da Cunha Lima, A.T.; Lopes, J.M.; de Vasconcelos, R.N.; Gouveia, M.B.; Porsani, M.J. SAR Oil Spill Detection System through Random Forest Classifiers. *Remote Sens.* **2021**, *13*, 2044. [\[CrossRef\]](#)
12. Li, K.; Yu, H.; Xu, Y.; Luo, X. Detection of Oil Spills Based on Gray Level Co-Occurrence Matrix and Support Vector Machine. *Front. Environ. Sci.* **2022**, *10*, 1049880. [\[CrossRef\]](#)
13. Misra, A.; Balaji, R. Simple Approaches to Oil Spill Detection Using Sentinel Application Platform (SNAP)-Ocean Application Tools and Texture Analysis: A Comparative Study. *J. Ind. Soc. Remote Sens.* **2017**, *45*, 1065–1075. [\[CrossRef\]](#)
14. Chehresa, S.; Amirkhani, A.; Rezairad, G.-A.; Mosavi, M.R. Optimum Features Selection for Oil Spill Detection in SAR Image. *J. Ind. Soc. Remote Sens.* **2016**, *44*, 775–787. [\[CrossRef\]](#)
15. Liu, P.; Li, Y.; Liu, B.; Chen, P.; Xu, J. Semi-Automatic Oil Spill Detection on X-Band Marine Radar Images Using Texture Analysis, Machine Learning, and Adaptive Thresholding. *Remote Sens.* **2019**, *11*, 756. [\[CrossRef\]](#)
16. Li, B.; Xu, J.; Pan, X.; Ma, L.; Zhao, Z.; Chen, R.; Liu, Q.; Wang, H. Marine Oil Spill Detection with X-Band Shipborne Radar Using GLCM, SVM and FCM. *Remote Sens.* **2022**, *14*, 3715. [\[CrossRef\]](#)
17. Ajadi, O.A.; Meyer, F.J.; Tello, M.; Ruella, G. Oil Spill Detection in Synthetic Aperture Radar Images Using Lipschitz-Regularity and Multiscale Techniques. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 2389–2405. [\[CrossRef\]](#)
18. Chen, G.; Li, Y.; Sun, G.; Zhang, Y. Application of Deep Networks to Oil Spill Detection Using Polarimetric Synthetic Aperture Radar Images. *Appl. Sci.* **2017**, *7*, 968. [\[CrossRef\]](#)
19. Guo, Y.; Zhang, H.Z. Oil Spill Detection Using Synthetic Aperture Radar Images and Feature Selection in Shape Space. *Int. J. Appl. Earth Obs. Geoinf.* **2014**, *30*, 146–157. [\[CrossRef\]](#)
20. Gil, P.; Alacid, B. Oil Spill Detection in Terma-Side-Looking Airborne Radar Images Using Image Features and Region Segmentation. *Sensors* **2018**, *18*, 151. [\[CrossRef\]](#)
21. Huang, X.; Zhang, B.; Perrie, W.; Lu, Y.; Wang, C. A Novel Deep Learning Method for Marine Oil Spill Detection from Satellite Synthetic Aperture Radar Imagery. *Mar. Pollut. Bull.* **2022**, *179*, 113666. [\[CrossRef\]](#)
22. Wen, H.; Dai, F.; Yuan, Y. A Study of YOLO Algorithm for Target Detection. *Proc. Int. Conf. Artif. Life Robot.* **2021**, *26*, 622–625. [\[CrossRef\]](#)
23. Du, L.; Zhang, R.; Wang, X. Overview of Two-Stage Object Detection Algorithms. *J. Phys. Conf. Ser.* **2020**, *1544*, 012033. [\[CrossRef\]](#)
24. Wang, C.-F.; Liu, K. A Novel Particle Swarm Optimization Algorithm for Global Optimization. *Comput. Intell. Neurosci.* **2016**, *2016*, 9482073. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Delahaye, D.; Chaimatanan, S.; Mongeau, M. Simulated Annealing: From Basics to Applications. In *Handbook of Metaheuristics*; Springer International Publishing: Cham, Switzerland, 2019; pp. 1–35. ISBN 9783319910857.

26. Chen, S.; Luk, B.L. Adaptive Simulated Annealing for Optimization in Signal Processing Applications. *Signal Process.* **1999**, *79*, 117–128. [[CrossRef](#)]
27. Liu, S.; Huang, F.; Yan, B.; Zhang, T.; Liu, R.; Liu, W. Optimal Design of Multimissile Formation Based on an Adaptive SA-PSO Algorithm. *Aerospace* **2021**, *9*, 21. [[CrossRef](#)]
28. Li, B.; Xu, J.; Chu, L.; Yang, Y.; Huang, X.; Liu, P. Oil Film Semantic Segmentation Method in X-Band Marine Radar Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1503205. [[CrossRef](#)]
29. van Zyl, J.P.; Engelbrecht, A.P. Set-Based Particle Swarm Optimisation: A Review. *Mathematics* **2023**, *11*, 2980. [[CrossRef](#)]
30. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-Means Clustering Algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat.* **1979**, *28*, 100. [[CrossRef](#)]
31. Fingas, M.; Brown, C. Review of Oil Spill Remote Sensing. *Mar. Pollut. Bull.* **2014**, *83*, 9–23. [[CrossRef](#)]

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.