

Article

Unsupervised Oil Spill Detection in Shipborne Radar Imagery Using Autoencoder-Enhanced Q-Learning and Improved Bat Optimization

Jin Yan ^{1,2}, Binghui Chen ^{1,2} , Jin Xu ^{1,2,*} , Zekun Guo ^{1,2} , Minghao Yan ^{1,2}, Mengxin Sun ^{1,2}  and Lin Qiao ³

¹ Naval Architecture and Shipping College, Guangdong Ocean University, Zhanjiang 524091, China; yanj@gdou.edu.cn (J.Y.); chenbh@stu.gdou.edu.cn (B.C.); zkguo@stu.gdou.edu.cn (Z.G.); yanmh@stu.gdou.edu.cn (M.Y.); sunmx@stu.gdou.edu.cn (M.S.)

² Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching, Guangdong Ocean University, Zhanjiang 524088, China

³ Navigation College, Dalian Maritime University, Dalian 116026, China; qiao@dlmu.edu.cn

* Correspondence: jinxu@gdou.edu.cn; Tel.: +86-15842695866

Highlights

What are the main findings?

- The proposed unsupervised framework, combining autoencoder-based feature learning, K-Means pseudo-labeling, Q-learning, and an improved bat algorithm, effectively detected oil spill regions in shipborne radar images without using manual annotations for model training.
- The method achieved better oil-slick segmentation performance than classic intelligent optimization methods and the conventional bat algorithm, and ablation tests confirmed the value of autoencoder feature extraction, K-Means pseudo-labeling, and Q-learning-based ROI localization.

What are the implications of the main findings?

- The study provides a practical technical route for timely shipborne offshore oil spill monitoring, especially in complex radar scenes with blurred boundaries, strong co-frequency interference, and speckle noise.
- The unsupervised design reduces reliance on labeled data, which may improve deployment feasibility for emergency response and environmental protection applications in real-world marine monitoring systems.

Abstract

Marine oil spill accidents pose a serious threat to the marine ecological environment. Therefore, efficient and accurate oil spill detection is of great significance for emergency response. To address the issues of blurred oil-slick boundaries, prominent co-frequency interference and severe speckle noise in shipborne radar images, this study proposed an oil spill detection method based on radar data collected from a real oil spill event at a terminal in Dalian Bay. The proposed method integrates an autoencoder, feature dimensionality reduction, pseudo-labeling, reinforcement learning and an improved intelligent optimization algorithm. First, an autoencoder was adopted to extract compact nonlinear local features from the radar images, and principal component analysis (PCA) was employed for feature dimensionality reduction. Subsequently, K-Means clustering was used to construct pseudo-labels, and the reduced features were discretized to build the state space for reinforcement learning. Based on this, the Q-learning algorithm was introduced to automatically extract the region of interest (ROI). Finally, for the ROI, an improved bat algorithm incorporating a



Academic Editors: Keshav D. Singh, Abdul Bais and Saeid Homayouni

Received: 22 April 2026

Revised: 4 June 2026

Accepted: 5 June 2026

Published: 7 June 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

dynamic weighting factor and a multi-constraint fitness function was designed to achieve fine segmentation of the oil-slick target. The experimental results showed that the proposed method outperformed classic intelligent optimization algorithms and the conventional bat optimization algorithm in oil-slick segmentation performance. Ablation experiments further verified the effectiveness of autoencoder-based feature learning, K-Means pseudo-labeling, and Q-learning-based ROI localization. This method may provide a new technical approach for timely offshore oil spill monitoring and emergency analysis.

Keywords: oil spill detection; shipborne radar; autoencoder; Q-learning; bat algorithm

1. Introduction

With the increasing frequency of offshore oil exploitation and transportation, the risk of marine oil spill incidents has been continuously rising [1–3]. Marine oil spills can severely damage marine ecosystems, fisheries, coastal environments, regional economies, and human health [4–7]. Representative accidents, such as the Deepwater Horizon oil spill in the Gulf of Mexico, the MV Wakashio fuel spill off Mauritius, and the Penglai 19-3B platform oil spill in the Bohai Sea, have demonstrated that marine oil pollution may lead to long-term ecological contamination, costly cleanup operations, and persistent socioeconomic impacts [8–13]. Therefore, rapid and accurate oil spill detection and segmentation are of great importance for offshore emergency response, pollution control, and environmental protection.

In recent years, the development of remote sensing technology, machine learning, and deep learning has promoted significant progress in marine oil spill monitoring [14,15]. Different remote sensing data sources, including synthetic aperture radar (SAR), hyperspectral imagery, infrared imagery, optical imagery, and shipborne radar, have been used for oil spill identification under various marine conditions. However, due to differences in sensor characteristics, complex sea-surface backgrounds, speckle noise, co-frequency interference, and blurred oil-slick boundaries, reliable oil spill detection remains challenging, especially for shipborne X-band radar imagery [16,17]. In operational marine emergency response, oil spill monitoring is also increasingly related to Digital Twin of the Ocean (DTO) frameworks, where real-time observations, AI-based perception, numerical ocean models, and decision-support systems are integrated. For oil spill response, accurate oil-slick extent information can support not only situational awareness but also the initialization and updating of drift and dispersion models. Metheniti et al. [18] proposed a Coastal Crete DTO architecture for oil spill identification and forecasting, showing the potential of coupling detection modules with atmospheric, hydrodynamic, and wave models. Recent studies on AI for DTOs have also emphasized that robust perception, uncertainty quantification, multi-source fusion, and model coupling remain key challenges for operational ocean digital twins [19].

Among existing approaches, CNN-based networks and their variants have been extensively explored for oil spill segmentation. However, these approaches still suffer from insufficient detail representation and overfitting under complex sea surface backgrounds [20]. To improve oil spill segmentation performance in SAR images, Zhang et al. [21] proposed an improved DeepLabV3+ model. In this model, the lightweight MobileNetV2 was adopted to replace Xception as the backbone network. Additionally, the scSE module and BCE-Dice joint loss function were introduced to enhance the model's ability to represent local details and improve its generalization performance. To address the insufficient utilization of polarimetric information in SAR data, Ma et al. [22] developed an improved DeepLabV3+ model based on Sentinel-1 dual-polarization images. The model integrated amplitude,

phase and Cloude polarimetric decomposition parameters, and combined ResNet-101 with group normalization (GN) to improve the segmentation accuracy of oil spill boundaries. Song et al. [23] designed a scene-adaptive oil spill detection network and introduced dynamic convolution and boundary constraint modules. These enhancements improved the network's ability to extract features of complex textures and irregular boundaries from polarimetric synthetic aperture radar (PolSAR) data, thereby achieving high-precision detection across different scenarios. Meanwhile, Chen et al. [24] introduced an attention gate mechanism into U-Net. They also fused PolSAR polarimetric information with sea surface wind speed information to improve the discrimination between oil spills and background in complex scenarios.

Beyond CNN architectures, Transformer- and GAN-based models have attracted attention for improving feature generalization and addressing sample scarcity. Kang et al. [25] optimized the Swin-UPerNet model for oil spill detection in high-resolution optical satellite images. The self-attention mechanism was used to integrate global and local features, thereby improving detection performance in complex scenes. To address the scarcity of real oil spill samples, Cai et al. [26] combined SinGAN with transfer learning to construct a few-shot detection framework. Diverse samples were generated to expand the training set, and YOLOv8 was pre-trained to improve detection accuracy. For classifying oil-like targets in SAR images, Fan et al. [27] proposed a multi-task generative adversarial network (MT-GAN). This method jointly exploited classification and generation mechanisms to achieve high-precision oil spill detection and segmentation under limited sample conditions.

In the direction of hybrid algorithms and multi-source fusion, researchers have mainly improved model robustness and applicability by integrating different algorithms or multi-source information. Zhan et al. [28] proposed a hyperspectral oil spill detection method combining CNN and DBSCAN. This approach balanced accuracy and efficiency through pre-feature selection, pixel classification and clustering segmentation. Liu et al. [29] constructed a multi-source knowledge graph by fusing remote sensing images, vector data, text data and atmosphere–ocean model data. They combined rule reasoning with graph neural networks to realize oil spill identification, thereby mitigating performance degradation caused by data imbalance. Yang et al. [30] extracted multi-scale features based on wavelet transform and employed a decision fusion strategy to integrate outputs from multiple models, thereby improving recognition accuracy under limited sample conditions. For shipborne radar scenarios, Xu et al. [31] proposed a detection method combining YOLOv8 and SA-PSO for the rapid identification and refined extraction of oil spill regions. Chen et al. [32] extracted texture features based on GLCM and combined K-Means initial localization with Sauvola threshold segmentation to achieve oil spill extraction in X-band maritime radar images. To address the scarcity of labeled optical oil spill samples, ref. [33] proposed a small sample generation adversarial network (SSGAN) for oil spill segmentation in high-resolution optical satellite images. The spatial attention mechanism was introduced to integrate spatial feature information, thereby improving extraction accuracy in complex marine scenes. Recent studies have explored unlabeled or limited labeled data to reduce the dependence on manual annotation in oil spill detection. For example, Han et al. [34] theoretically showed through VC-bound analysis that mutual guide learning and Confucius tri-learning can improve generalization under limited supervision.

In summary, although existing oil spill detection and segmentation methods have shown promising performance across diverse data sources and application scenarios, several challenges remain for shipborne X-band radar imagery. First, fully supervised CNN-, Transformer-, and GAN-based approaches typically require large quantities of accurately annotated samples, yet labeled shipborne radar oil spill images are difficult to obtain during real emergency events. Second, traditional thresholding and intelligent

optimization methods are highly sensitive to speckle noise, co-frequency interference, and complex sea-surface clutter, often resulting in fragmented segmentation and false alarms. Third, oil-slick boundaries in shipborne radar images are usually weak and blurred, making it difficult for methods relying solely on gray-level or texture information to preserve complete oil-slick morphology. From the perspective of DTO-oriented oil spill emergency response, these limitations may reduce the reliability and timeliness of oil-slick information required for near-real-time decision support.

To overcome these challenges, this study proposes an unsupervised fusion framework for shipborne radar oil spill detection that integrates autoencoder-based feature learning, K-Means pseudo-labeling, Q-learning-based ROI localization, and improved bat-algorithm-based fine segmentation. These components play complementary roles within the detection pipeline. First, the autoencoder learns compact nonlinear feature representations directly from radar image patches without manual annotations, enhancing discrimination between oil slicks and complex sea-surface backgrounds. Second, K-Means clustering generates pseudo-labels and discrete feature states, enabling the construction of an unsupervised decision environment. Based on these pseudo-labels and the reduced feature space, Q-learning formulates ROI extraction as an adaptive pixel-level decision-making process, allowing suspicious oil spill regions to be automatically localized by combining pseudo-label agreement with neighborhood spatial consistency while reducing the search space for subsequent segmentation. Finally, the improved bat algorithm is applied within the extracted ROI to optimize threshold selection and refine oil-slick boundary delineation. By integrating representation learning, adaptive decision-making, and global optimization, the proposed framework provides a unified unsupervised solution for shipborne radar oil spill detection.

2. Methodologies

This study proposed an unsupervised radar-image oil spill detection model based on K-Means clustering. By integrating autoencoder-based feature learning, the Q-learning decision mechanism and an improved bat algorithm, a complete framework for automatic oil-slick extraction was established.

2.1. K-Means Clustering

K-Means clustering is a classic unsupervised learning algorithm known for its simple principle and high computational efficiency. It has been widely used in spatial partitioning, image processing, pattern recognition and geoscientific modeling. The algorithm mainly includes the following steps:

1. Randomly select k samples from the sample space as the initial cluster centers.
2. Calculate the distance from each sample to each cluster center, usually using the Euclidean distance, and assign the sample to the nearest cluster.
3. Compute the mean of the samples in each cluster and update the cluster center accordingly.
4. Repeat steps 2 and 3 until the cluster centers remain stable or the preset maximum number of iterations is reached.

The core objective of this algorithm is to achieve an optimal data partition by minimizing the sum of squared errors (SSE) between the samples and their corresponding cluster centers. The objective function is defined as:

$$SSE = \min \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (1)$$

where k denotes the number of clusters, C_i denotes the i -th cluster, μ_i denotes the center of the i -th cluster, x denotes the pixel gray value of a sample, and $\|\cdot\|$ denotes the Euclidean distance operator.

2.2. Autoencoder

Autoencoder (AE) is a typical unsupervised neural network model that learns latent low-dimensional feature representations by reconstructing the input data. It performs well in feature extraction, nonlinear dimensionality reduction and noise suppression [35]. Its basic idea is to construct an encoder–decoder architecture that uses the input data itself as the supervisory signal. By minimizing the reconstruction error between the input and output, the network automatically learns the intrinsic feature structure of the data.

The autoencoder used in this study is intentionally designed as a lightweight shallow feature encoder rather than a large deep neural network. Each input sample is a normalized 5×5 local radar image patch, corresponding to a 25-dimensional vector. The hidden layer compresses this local patch into an 8-dimensional representation, which is subsequently used for PCA and Q-learning state construction. This design was adopted for three reasons. First, the available shipborne radar oil spill data are limited, and a larger deep network may easily overfit the single-event radar scene. Second, the objective of the autoencoder in this framework is not end-to-end semantic segmentation, but compact local texture and intensity representation for subsequent unsupervised decision making. Third, the lightweight structure reduces computational cost and is more suitable for timely shipborne radar processing. Therefore, the autoencoder should be interpreted as a compact unsupervised feature-encoding module rather than as a large-scale deep feature extractor.

2.2.1. Image Patch Sampling

From the preprocessed radar image, 1000 image patches, each measuring 5×5 pixels, were randomly selected without replacement for AE training. Each patch was centered on a valid pixel, and duplicate center pixels were avoided. Patch centers were sampled across the whole image to reduce excessive local redundancy. Because the AE was trained in an unsupervised manner for local feature representation rather than class-specific classification, no class-balance constraint was imposed. The trained AE was then applied to the full image in a sliding-window manner to obtain pixel-level feature representations. Each patch was referenced to its central pixel and contained local spatial information in the neighborhood of that pixel. This sampling strategy preserves local spatial features at the pixel level.

2.2.2. AE Model Construction

The constructed AE model consists of an encoder and a decoder. The encoder maps the high-dimensional input x to a low-dimensional hidden feature h :

$$h = \sigma(W_1x + b_1) \quad (2)$$

where x is the vectorized image patch; $W_1 \in \mathbb{R}^{8 \times 25}$ is the encoder weight matrix, which compresses the 25-dimensional input into an 8-dimensional feature; $b_1 \in \mathbb{R}^8$ is the bias vector; and $\sigma(\cdot)$ is the sigmoid activation function.

The decoder reconstructs the low-dimensional hidden feature h back to the original input space $\hat{x}$:

$$\hat{x} = \sigma(W_2h + b_2) \quad (3)$$

where $W_2 \in \mathbb{R}^{25 \times 8}$ is the decoder weight matrix and $b_2 \in \mathbb{R}^{25}$ is the decoder bias.

2.2.3. AE Model Training

The training objective of the AE is to optimize the network parameters by minimizing the mean squared error (MSE) between the input and reconstructed output. The loss function is defined as:

$$\mathcal{L} = \frac{1}{M} \sum_{m=1}^M \|x_m - \hat{x}_m\|^2 \quad (4)$$

where x_m is the m -th input image patch, $\hat{x}_m$ is the corresponding reconstructed output, and M is the number of training samples. In this experiment, M was set to 1000.

2.2.4. Full-Image Feature Extraction

After AE training, for each pixel in the image, its 5×5 neighborhood was extracted to form an image patch and then fed into the trained autoencoder to obtain the corresponding 8-dimensional feature vector. A feature matrix of the same size as the input image was then generated, where each pixel corresponded to an 8-dimensional feature representation. Subsequently, PCA was applied to reduce the dimensionality of the feature matrix, compressing the 8-dimensional features into 3 dimensions and providing a feature basis for automatic oil spill target extraction.

More complex autoencoder variants, such as convolutional autoencoders, denoising autoencoders, and variational autoencoders, were not adopted in the present study because the available shipborne radar oil spill samples are limited and do not provide sufficient diversity for training larger neural architectures reliably. In addition, the proposed framework aims to maintain a lightweight unsupervised pipeline that can be applied without manually annotated training samples. Although more advanced autoencoder architectures may further improve feature representation, they would also introduce additional hyperparameters, higher computational cost, and a greater risk of overfitting under the current data condition. This issue has been added as a limitation and will be investigated in future work when larger multi-scene shipborne radar datasets become available.

2.3. Q-Learning

Q-learning is a model-free reinforcement learning algorithm based on value iteration. Its core idea is to enable the agent to search for the optimal policy by continuously updating the action-value function during interaction with the environment. Although Q-learning is usually used for dynamic optimization and control, here it is employed as a sequential decision-making mechanism rather than a supervised classifier. This is because oil spill detection in shipborne radar images has ambiguous boundaries, strong sea-clutter interference, and limited annotations. After unsupervised feature learning and K-Means pseudo-label generation, each pixel or local patch is treated as a state, and label assignment is formulated as an action-selection problem. Q-learning is then used to learn an adaptive policy for ROI localization, which reduces the search space for later threshold optimization by the improved bat algorithm.

2.3.1. Q-Table Initialization

In image segmentation tasks, Q-learning realizes pixel-level classification by constructing state-action value mapping. First, a two-dimensional Gaussian filter was used to smooth the image and reduce noise. The Gaussian kernel $G(x, y)$ can be expressed as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

where (x, y) denotes the relative coordinates within the Gaussian kernel, σ is the standard deviation of the Gaussian distribution, and $2\pi\sigma^2$ is the normalization factor, which en-

sures that the integral of the Gaussian kernel equals 1 and avoids gray-level deviation after smoothing.

State-space discretization.

Let $F \in \mathbb{R}^{H \times W \times 8}$ denote the pixel-level feature map extracted by the trained autoencoder. PCA was first applied to reduce the 8-dimensional feature vector of each valid pixel to a 3-dimensional vector, denoted as $z_i \in \mathbb{R}^3$, where $i = 1, \dots, N$, and N is the number of valid pixels. The 3-dimensional feature vectors were then standardized dimension-by-dimension to reduce the influence of scale differences.

The continuous feature space was discretized using K-Means clustering with $k = 50$. Specifically, K-Means was performed on the set $\{z_i\}_{i=1}^N$ by minimizing the within-cluster sum of squared Euclidean distances. After convergence, each pixel was assigned to the nearest cluster center. The cluster index $c_i \in \{1, 2, \dots, 50\}$ was used as the discrete Q-learning state of the corresponding pixel:

$$s_i = c_i = \arg \min_{j \in \{1, \dots, 50\}} \|z_i - \mu_j\|_2^2$$

where μ_j is the center of the j -th cluster. Therefore, the Q-learning state space was defined as $S \in \{1, 2, \dots, 50\}$. It should be noted that these 50 states do not represent oil spill classes or ground-truth labels. They only define the state index of each pixel in the Q-table. The choice of $k = 50$ was not arbitrary. As discussed in Section 5.7, different values of k from 20 to 70 were tested, and $k = 50$ achieved performances very close to the best setting while avoiding both under-partitioning and excessive state fragmentation.

The action space was defined as $\{1, 2, 3\}$, corresponding to oil spill, background and suspected oil spill, respectively. The action selected by the agent determined the final classification result of the pixel. Based on the state and action spaces, a Q-table $Q \in \mathbb{R}^{50 \times 3}$ initialized to zero was constructed. $Q(s, a)$ denotes the value estimate of the agent taking action a in state s , that is, the cumulative discounted reward associated with that action.

2.3.2. Q-Learning Training Process

The core of Q-learning training is the iterative update of the Q-table based on the Bellman equation. Each iteration follows the procedure below. First, a pixel (x, y) is randomly selected from the image to simulate the agent's random exploration in the environment. Then, a single-round training procedure is performed, in which state acquisition and action selection based on the ε -greedy strategy are carried out sequentially for each pixel:

$$a = \begin{cases} \mathcal{U}(\{1, 2, 3\}), & \eta < \varepsilon \\ \arg \max_{a'} Q(s, a'), & \text{otherwise} \end{cases} \quad (6)$$

where ε is the exploration rate, $\mathcal{U}(\{1, 2, 3\})$ denotes random selection of an action from 1 to 3, $\arg \max_{a'} Q(s, a')$ denotes selecting the action with the highest value in the current state, and $\eta \sim U(0, 1)$.

After action selection, the immediate reward was calculated by combining a pseudo-label agreement term and an 8-neighborhood spatial-consistency term. For the i -th pixel, let $a_i \in 1, 2, 3$ denote the action selected by the agent, and let $\tilde{y}_i \in 1, 2, 3$ denote the pseudo-label generated by K-Means clustering on the smoothed grayscale image. The reward function is defined as:

$$r_i(a_i) = \lambda_c r_i^{cls}(a_i) + \lambda_n r_i^{nei}(a_i) \quad (7)$$

where $r_i^{cls}(a_i)$ is the pseudo-label agreement reward:

$$r_i^{cls}(a_i) = \begin{cases} +1, a_i = \tilde{y}_i \\ -1, a_i \neq \tilde{y}_i \end{cases} \quad (8)$$

where $r_i^{nei}(a_i)$ is the normalized 8-neighborhood consistency reward:

$$r_i^{nei}(a_i) = \frac{1}{|\mathcal{N}_8(i)|} \sum_{j \in \mathcal{N}_8(i)} 1(\hat{y}_j = a_i) \quad (9)$$

where, $\mathcal{N}_8(i)$ denotes the valid 8-neighborhood of the i -th pixel, $|\mathcal{N}_8(i)|$ is the number of valid neighboring pixels, $\hat{y}_j$ denotes the current classification result of neighboring pixel j , and $1(\cdot)$ is the indicator function. The first term encourages the selected action to be consistent with the K-Means-derived pseudo-label, while the second term encourages spatially adjacent pixels to have consistent classification results.

In this study, the weights were set to $\lambda_c = 1$ and $\lambda_n = q = 0.3$. Therefore, the pseudo-label agreement term served as the dominant reward component, whereas the neighborhood-consistency term acted as a moderate spatial regularization term. This setting encourages local spatial smoothness and suppresses isolated noisy classifications, while avoiding excessive smoothing that could blur oil-slick boundaries. It should be noted that the pseudo-labels used in this reward function were generated by unsupervised K-Means clustering and were not manual ground-truth labels. The manually annotated reference mask was used only for final quantitative evaluation and was not involved in Q-learning reward calculation.

Finally, the Q-value update is performed as follows:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot [r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a)] \quad (10)$$

where α is the learning rate, which controls the update magnitude of the Q-value at each step; γ is the discount factor, which determines the weight of future rewards; s' is the next state reached after executing action a ; r is the immediate reward obtained by the agent after taking action a in state s ; $Q(s, a)$ is the Q-value of the current state-action pair; and $\max_{a'} Q(s', a')$ is the optimal Q-value corresponding to all possible actions in the next state s' .

Through iterative optimization, the Q-table eventually converges to a stable state. At this point, the optimal action for each state becomes the optimal classification decision for the corresponding pixel, thereby enabling unsupervised automatic extraction of the oil spill region in radar images.

2.4. Improved Bat Algorithm

Bat algorithm (BA) is a population-based swarm intelligence optimization algorithm that simulates the echolocation behavior of bats. Owing to its strong global search capability and fast convergence speed, it has been widely used in parameter optimization, image segmentation and feature selection. The core idea is to simulate the process by which bats locate prey through adjustments in frequency, loudness, and pulse emission rate. This mechanism maps the solution space of the optimization problem to the search space of bats and achieves adaptive search for the optimal solution through iterative population updates.

To address the sensitivity of threshold selection in radar-image oil spill segmentation, this study proposes an improved bat algorithm. By introducing a dynamic weighting factor, an improved initialization strategy and an adaptive parameter adjustment mechanism,

the proposed method enables adaptive optimization of the segmentation threshold. The workflow is shown in Figure 1.

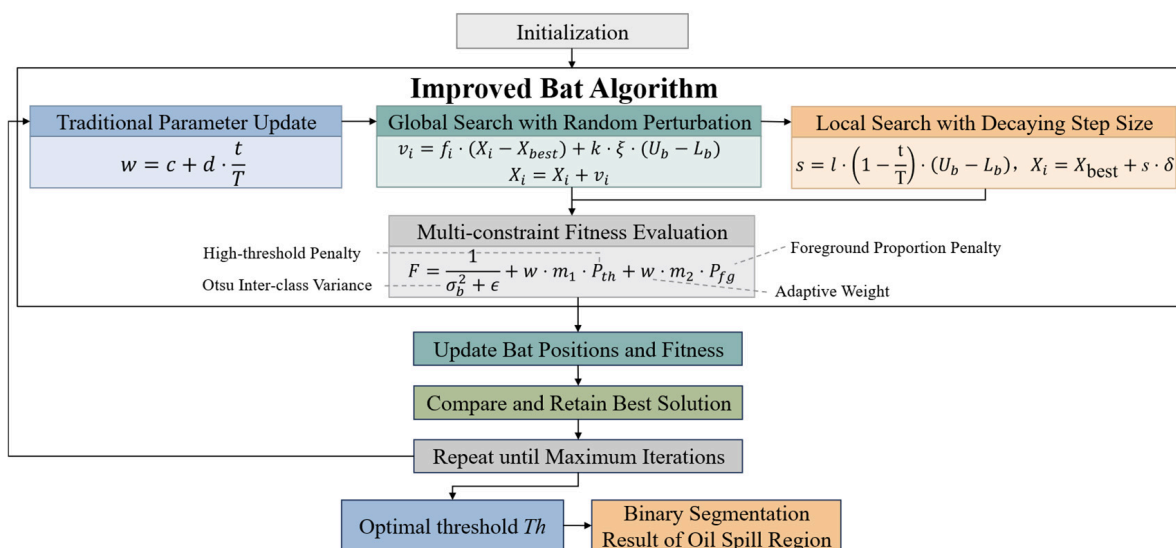


Figure 1. Flowchart of the improved bat algorithm.

2.4.1. Parameter Definitions of the Improved Bat Algorithm

1. Conventional parameter definitions

Emission frequency: The frequency controls the step-size range for bat position updating. The frequency f_i of the i -th bat is randomly generated as follows:

$$f_i = f_{min} + (f_{max} - f_{min}) \cdot r_3 \quad (11)$$

where f_{min} and f_{max} denote the lower and upper bounds of the frequency, respectively. $r_3 \sim U(0, 1)$ is a uniformly distributed random number, which ensures that the frequency fluctuates randomly within a valid range.

Loudness and pulse rate: Loudness reflects the reliability of a solution, whereas pulse rate reflects the emission probability of a bat. Their iterative update equations are given by:

$$A_i(t) = \alpha \cdot A_i(t-1) \quad (12)$$

where $A_i(t)$ denotes the loudness of the i -th bat at the t -th iteration, $A_i(t-1)$ denotes the loudness in the previous iteration, and α is the loudness decay coefficient.

$$r_i(t) = r_0(1 - e^{-\gamma t}) \quad (13)$$

where $r_i(t)$ denotes the pulse rate of the i -th bat at the t -th iteration, and γ is the pulse rate growth coefficient.

2. Improved parameter definitions

Dynamic weighting factor w : This factor is used to strengthen the weights of the high-threshold preference and foreground-ratio constraint in the fitness function. It increases linearly with the number of iterations and is defined as:

$$w = c + d \cdot \frac{t}{T} \quad (14)$$

where t is the current iteration number, T is the maximum number of iterations, and c and d are adjustment coefficients. In this experiment, c was set to 0.0045 and d was set to 0.5.

2.4.2. Population Iterative Updating with Random Perturbation

The position updating rule of the conventional BA is relatively simple and is therefore prone to local optima. In this study, improved strategies were introduced in both the global search and local search stages.

1. Global position update: A Gaussian random perturbation was introduced into the velocity update equation to escape local optima and improve search diversity:

$$v_i = f_i \cdot (X_i - X_{best}) + k \cdot \xi \cdot (U_b - L_b) \quad (15)$$

$$X_i = X_i + v_i \quad (16)$$

where X_{best} denotes the current global optimal threshold, $\xi \sim N(0, 1)$, and k is the adjustment coefficient, which was set to 0.01 in this experiment.

2. Local fine search: The conventional BA adopts a fixed step size. In this study, a step size that decays with the number of iterations was designed, so that large-step global exploration is performed in the early stage and small-step fine optimization is performed in the later stage:

$$s = l \cdot \left(1 - \frac{t}{T}\right) \cdot (U_b - L_b) \quad (17)$$

$$X_i = X_{best} + s \cdot \delta \quad (18)$$

where s is the iteration step size, $\delta \sim N(0, 1)$, and l is the adjustment coefficient, which was set to 0.2 in this experiment.

2.4.3. Fitness Function with Multi-Constraint Fusion

The conventional BA uses only a single indicator as the fitness function, which cannot simultaneously account for the high-threshold preference and foreground-ratio constraint in oil spill segmentation. Therefore, this study designed a multi-constraint fused fitness function, with the objective of minimizing the fitness value:

$$F = \frac{1}{\sigma_b^2 + \epsilon} + w \cdot m_1 \cdot P_{th} + w \cdot m_2 \cdot P_{fg} \quad (19)$$

where ϵ is a small value introduced to prevent division by zero; σ_b^2 is defined as the Otsu inter-class variance; m_1 and m_2 are adjustment coefficients, which were set to 0.3 and 0.2, respectively; $P_{th} = \frac{U_b - X_i}{U_b - \mu + \epsilon}$ is the high-threshold penalty term, where μ denotes the mean value of the valid pixels; and $P_{fg} = e^{-0.5 \cdot \frac{N_{fg}}{N}}$ is the foreground-ratio penalty term, where N_{fg} is the number of foreground pixels and N is the total number of valid pixels.

2.5. Evaluation Metrics

To quantitatively evaluate the segmentation performance, the final segmentation results were compared with the manual reference masks generated through visual interpretation described in Section 2.4. These reference masks were manually delineated from the preprocessed radar images and were independent of both the K-Means-derived pseudo-labels used for Q-learning reward construction and the K-Means-discretized state labels used for state-space construction. Therefore, the reported precision, accuracy, Dice coefficient, specificity, and IoU measure the agreement between the algorithmic segmentation results and the independent manual reference masks, rather than agreement with K-Means pseudo-labels.

In the binary segmentation task of oil spill detection, the prediction results can be described in terms of four types of pixel classification outcomes: TP (true positive) refers to oil spill pixels correctly predicted as oil spill, FP (false positive) refers to background pixels incorrectly predicted as oil spill, TN (true negative) refers to background pixels correctly predicted as background, and FN (false negative) refers to oil spill pixels incorrectly predicted as background. Based on these definitions, the five evaluation indicators are calculated as follows:

$$Precision = \frac{TP}{TP + FP} \quad (20)$$

where precision measures the proportion of correctly predicted oil spill pixels among all pixels predicted as oil spill, thereby reflecting the reliability of positive predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (21)$$

where accuracy represents the proportion of all correctly classified pixels among all pixels in the image, providing a global measure of classification performance.

$$Dice = \frac{2TP}{2TP + FP + FN} \quad (22)$$

where the Dice coefficient is a similarity metric commonly used in segmentation tasks. It evaluates the overlap degree between the predicted oil spill region and the reference region, with a higher value indicating better segmentation consistency.

$$Specificity = \frac{TN}{TN + FP} \quad (23)$$

where specificity reflects the ability of the method to correctly identify background pixels and is therefore an important indicator of the method's resistance to false alarms.

$$IoU = \frac{TP}{TP + FP + FN} \quad (24)$$

where Intersection over Union (IoU), also known as the Jaccard index, is another overlap-based metric for segmentation evaluation. It calculates the ratio of the intersection area to the union area between the predicted segmentation result and the manual reference mask.

2.6. Parameter Settings and Hyperparameter Summary

These hyperparameters shown in Table 1 were manually specified based on preliminary trials, algorithm stability, and commonly used settings in Q-learning and bat-algorithm-based image segmentation. A full grid-search or sensitivity analysis was not performed in the present study because the available shipborne radar oil spill dataset is limited. Therefore, the robustness of the proposed framework to hyperparameter variation remains a limitation and will be investigated in future work using larger multi-scene datasets.

Table 1. Manually specified hyperparameters used in the proposed framework.

Module	Parameter	Value	Role	Setting Basis
Gaussian smoothing	σ	1 pixel	Controls smoothing strength before ROI extraction	Empirical compromise between speckle suppression and boundary preservation
Q-learning reward	q	0.3	Weight of neighborhood-consistency reward	Empirical setting
Q-learning	Training episodes	10	Number of Q-table training rounds	Empirical setting
State discretization	k	50	Number of K-Means states	Selected according to the ablation study in Section 5.10
Bat algorithm	N	30	Population size	Empirical setting
Bat algorithm	T	30	Maximum iterations	Empirical setting
Bat algorithm	α	0.7	Loudness decay coefficient	Empirical setting
Bat algorithm	γ	0.05	Pulse-rate growth coefficient	Empirical setting
Bat algorithm	f_{min}, f_{max}	0, 0.1	Frequency bounds	Empirical setting
Bat algorithm	r_0	0.5	Initial pulse rate	Empirical setting
Improved BA	k	0.01	Gaussian perturbation coefficient	Empirical setting
Improved BA	l	0.2	Local search step-size coefficient	Empirical setting
Dynamic weight	c	0.0045	Weight-growth coefficient	Empirical setting
Dynamic weight	d	0.5	Initial weight offset	Empirical setting
Fitness function	m_1	0.3	High-threshold penalty weight	Empirical setting
Fitness function	m_2	0.2	Foreground-ratio penalty weight	Empirical setting

3. Materials and Methods

3.1. Research Data

The experimental dataset was obtained from a real oil spill event that occurred near a coastal terminal in Dalian Bay, which was triggered by a pipeline failure during tanker unloading operations. During the oil cleanup patrol, the experimental vessel Yu Kun, operated by Dalian Maritime University, collected the corresponding oil spill monitoring data using a shipborne radar system, as shown in Figure 2. All preprocessed images were sequentially acquired by a professional oil spill monitoring system equipped with an X-band horizontally polarized maritime radar. The detection range of the system was 0.75 nautical miles (approximately 1.39 km), and the collected dataset consisted of raw radar images with a resolution of 1024×1024 pixels. The radar range scale was 0.75 nautical miles, corresponding to approximately 1.39 km from the radar center to the outer boundary of the observation area. Therefore, the original 1024×1024 radar PPI image covered a circular detection area with a diameter of approximately 2.78 km. The nominal Cartesian pixel spacing of the original radar display image was approximately

2.71 m × 2.71 m per pixel. This value represents the spatial sampling interval of the radar display image rather than the intrinsic range or azimuth resolution of the radar sensor.

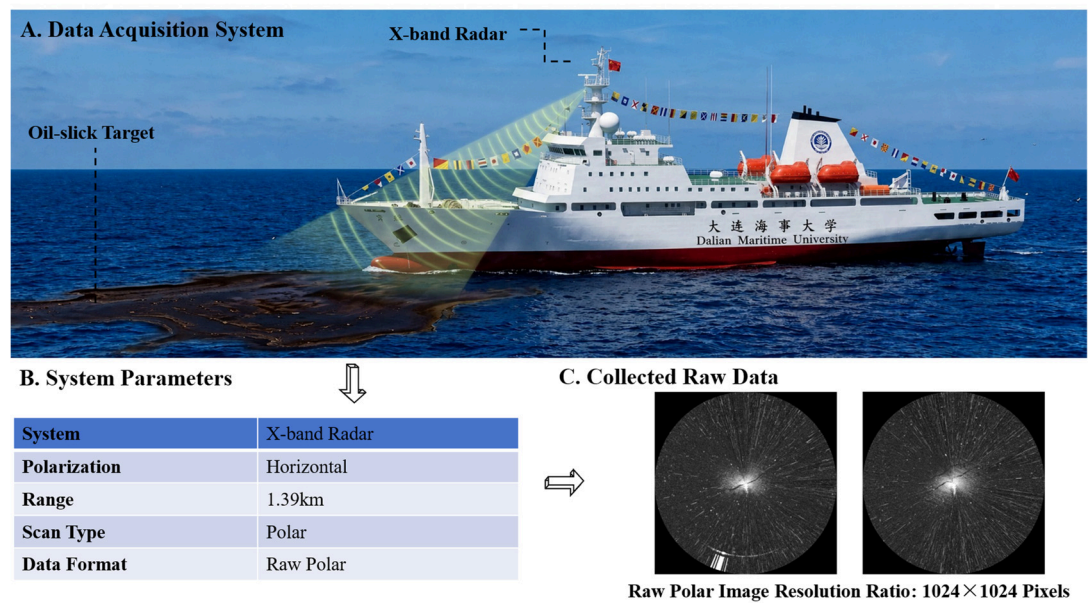


Figure 2. Data acquisition architecture.

3.2. Data Preprocessing

To ensure reliable oil-slick segmentation, the raw radar data were preprocessed as shown in Figure 3. The original radar PPI image was resampled into a 512 × 2048 range–azimuth representation, where 512 radial bins correspond to a detection range of 0.75 nautical miles and 2048 azimuth bins cover 360°. Therefore, the nominal radial spacing is approximately 2.71 m, and the angular spacing is approximately 0.176° per bin. Since the along-azimuth spacing varies with range, spatial measurements should be interpreted based on the radial and angular sampling intervals rather than a uniform square-pixel grid.

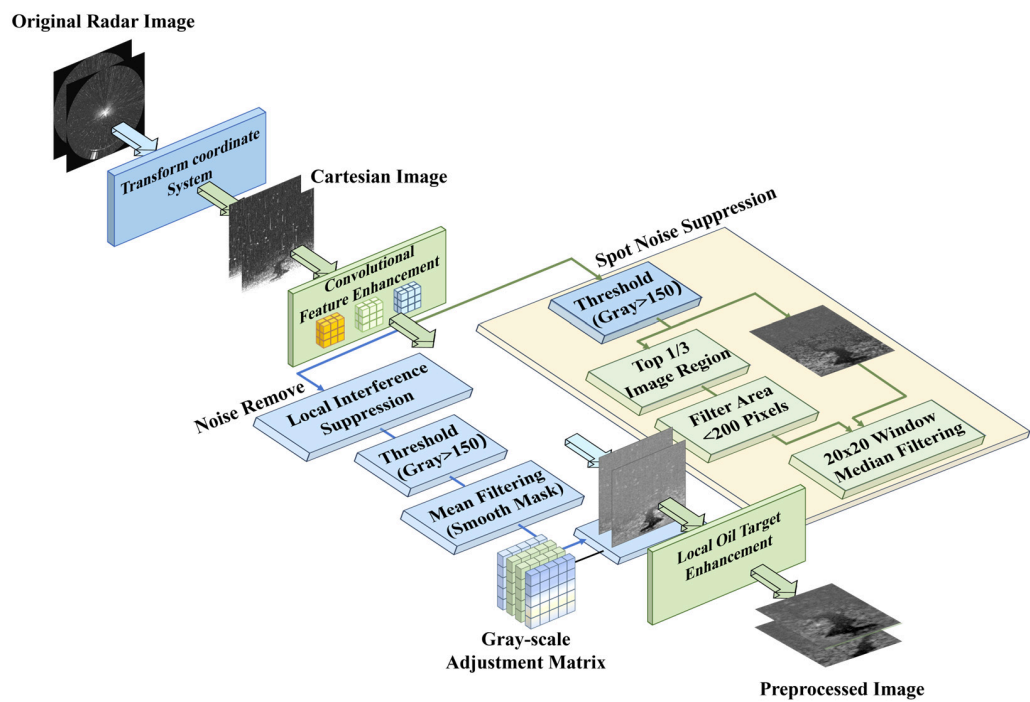


Figure 3. Flowchart of data processing.

The raw polar-coordinate radar data were then transformed into Cartesian coordinates for image processing. A convolution kernel was applied to enhance edge features and suppress noise. Co-frequency interference was reduced by extracting high-gray-value regions with a threshold of 150 followed by mean filtering. Speckle noise was further suppressed by detecting isolated high-brightness connected components using a grayscale threshold of 120 and an area threshold of 200 pixels, followed by median filtering with a 20×20 window. These thresholds were empirically selected according to the intensity distribution and spatial characteristics of the shipborne radar images. And a systematic sensitivity analysis of preprocessing parameters remains for future work. Finally, grayscale adjustment and local contrast enhancement were applied to improve the distinguishability of the oil-slick region from the background. This preprocessing pipeline improved image separability and provided suitable input for subsequent segmentation.

3.3. Experimental Procedure

The experimental workflow of this study is shown in Figure 4. First, compact nonlinear representations were learned from preprocessed image patches using a lightweight auto-encoder, followed by PCA for feature dimensionality reduction. Subsequently, K-Means clustering was used to generate pseudo-labels and construct the state space, and Q-learning was applied to perform pixel-level initial classification and ROI extraction. Finally, an improved bat algorithm was introduced for fine segmentation of the oil slick within the ROI, thereby enabling the accurate detection of oil spill targets.

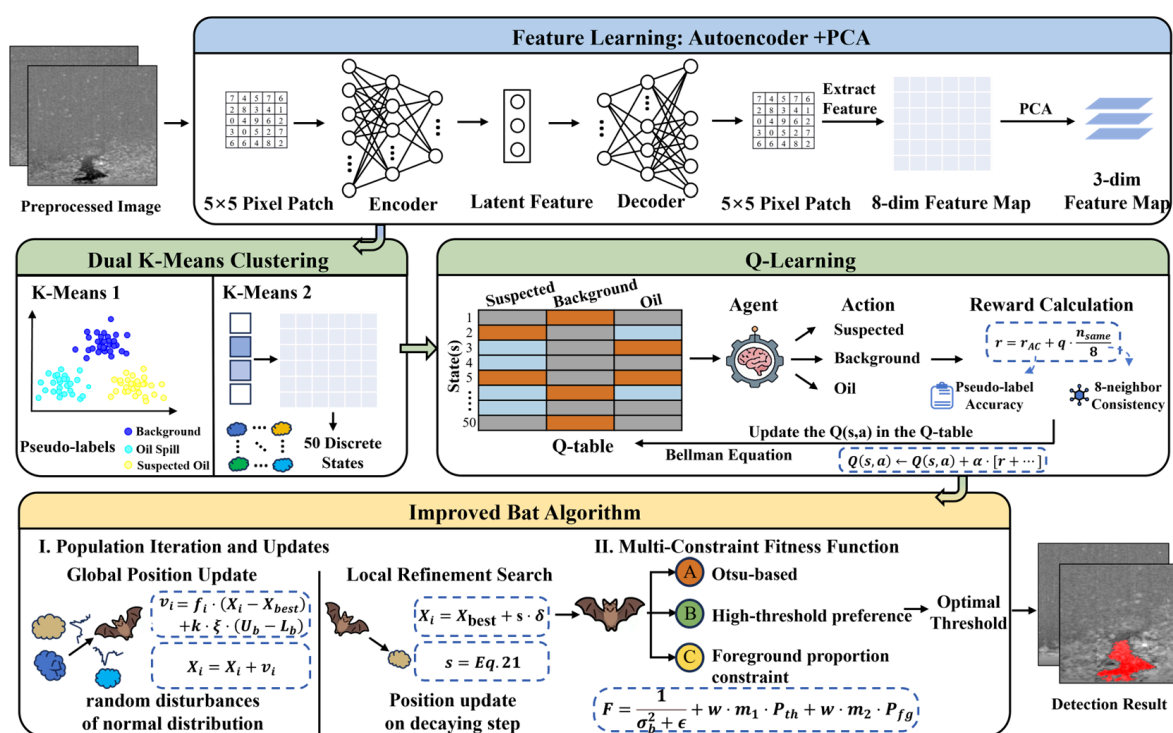


Figure 4. Workflow of the proposed oil spill detection framework from preprocessed shipborne radar input to final oil-slick segmentation.

3.4. Manual Reference Mask Annotation

To avoid circular validation between pseudo-label generation and quantitative evaluation, an independent manual reference mask was created for the radar image. The reference mask was generated through visual interpretation of the preprocessed shipborne radar image based on the observable characteristics of the oil-slick region. During the interpretation process, the K-Means pseudo-labels used for Q-learning reward construction, the

50-state K-Means discretization results, and the final segmentation outputs of the proposed method were not consulted. The oil spill region was manually delineated and converted into a binary mask, where oil spill pixels were assigned a value of 1 and background pixels were assigned a value of 0.

This manually annotated mask was used only for the final quantitative evaluation, including precision, accuracy, Dice coefficient, specificity, and IoU. It was not used in autoencoder training, K-Means pseudo-label generation, Q-learning reward calculation, state-space construction, bat-algorithm optimization, parameter tuning, or post-processing. Therefore, the manual reference mask is independent of the K-Means pseudo-labels used during training. Although the manual reference mask provides an independent basis for evaluation, it was generated from visual interpretation rather than from a multi-expert consensus. This limitation has been acknowledged in the Discussion. The initial mask was generated through visual interpretation and then checked by another researcher familiar with shipborne radar image interpretation. Disagreements were reviewed and corrected to obtain the final reference mask.

4. Results

4.1. ROI Extraction

First, the preprocessed radar images were smoothed by two-dimensional Gaussian filtering to reduce the influence of noise on subsequent feature extraction, clustering analysis and classification. The Gaussian smoothing parameter σ was empirically set to 1 pixel to suppress local speckle noise while preserving weak oil-slick boundaries. Larger σ values may blur narrow slick structures, whereas smaller values may retain excessive radar noise. Two separate K-Means operations were used in the ROI extraction stage. Specifically, the first step clustered the grayscale values of the smoothed image into three classes with SSE as the optimization objective, corresponding to oil spill, background and suspected oil spill regions. Pixel-level pseudo-labels were thereby generated and used as the reference basis for the classification-accuracy reward in the Q-learning reward function, as shown in Figure 5a. The second step applied PCA to reduce the 8-dimensional pixel-level features (extracted by the autoencoder) to three dimensions. The resulting feature matrix was then clustered with $k = 50$, thereby discretizing the continuous feature space into a finite state set and constructing the state space for Q-learning. To capture deep pixel features, a fully connected autoencoder was constructed to learn a nonlinear representation of the data in an unsupervised manner. The action space was defined as $\{1, 2, 3\}$ and matched to the three pseudo-labels generated by K-Means to realize pixel-level classification decisions, as shown in Figure 5b. The Q-table parameters were optimized through 10 training iterations, and the ϵ -greedy strategy was used during training to balance exploration and exploitation. The exploration rate was designed to encourage exploration at the beginning and exploitation later. Meanwhile, the action selection results were compared with the K-Means pseudo-labels, and classification accuracy was used as the reward signal. The reward function was further optimized by incorporating an 8-neighborhood consistency coefficient. Finally, the initial segmentation result was generated by the converged Q-table, as shown in Figure 5c. Subsequently, the initial segmentation result was binarized, as shown in Figure 5d, and further processed by speckle-noise removal and hole filling to obtain a complete ROI mask, as shown in Figure 5e. Finally, the mask was multiplied element-wise with the preprocessed radar image to obtain the oil spill ROI, as shown in Figure 5f, providing a basis for subsequent oil-slick boundary analysis and feature extraction.

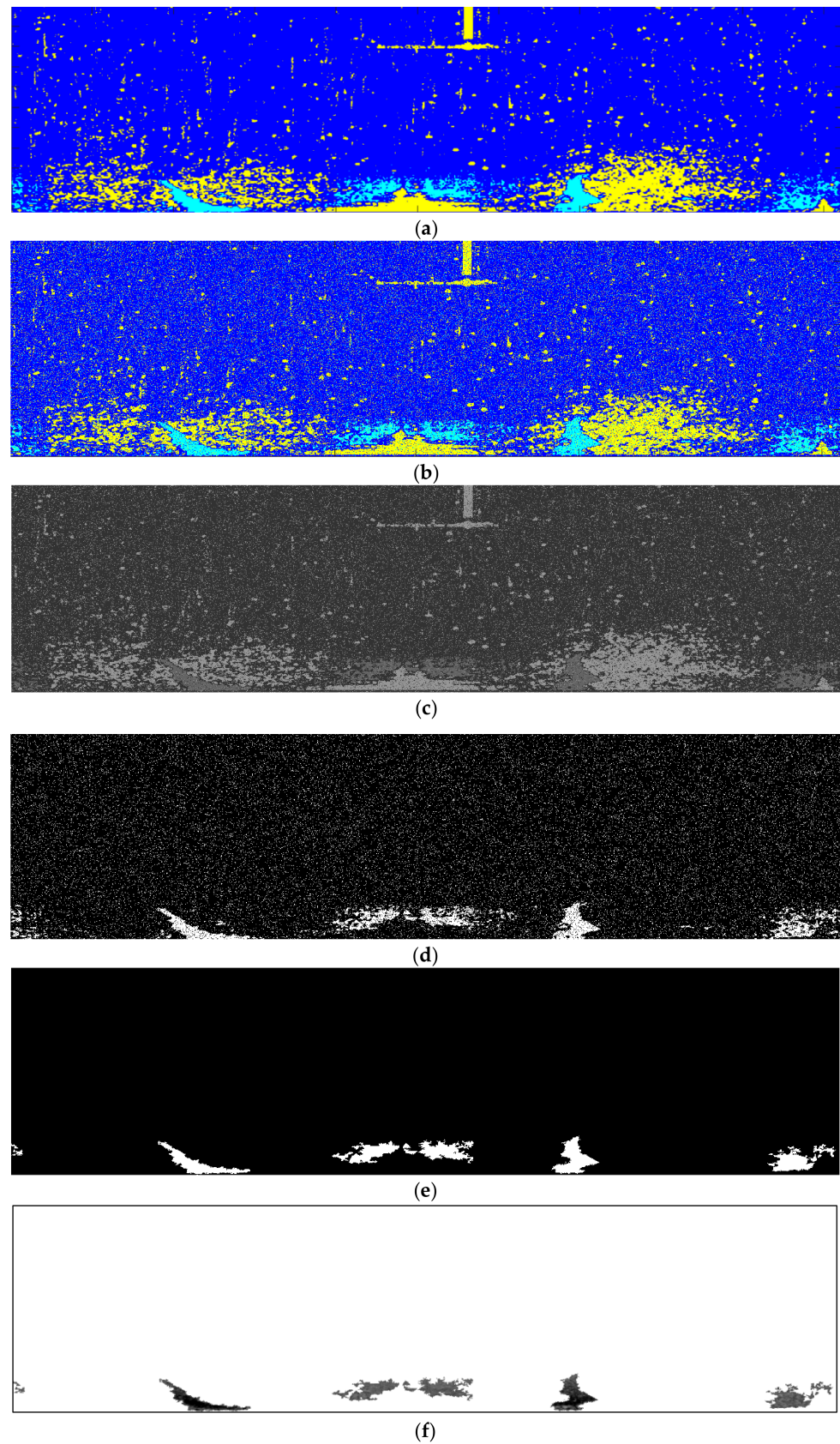


Figure 5. ROI extraction, (a) pseudo-label, (b) classification result, (c) preliminary classification of Q-Learning, (d) binary mask, (e) ROI mask, (f) ROI.

4.2. Oil Spill Target Segmentation

This study used the improved bat algorithm to extract oil spill targets from non-background pixels within the ROI, as shown in Figure 6. In this experiment, the bat population size N was set to 30, the maximum number of iterations T was set to 30, the initial loudness decay coefficient α was set to 0.7, the frequency bounds were set to $f_{min} = 0$ and $f_{max} = 0.1$, the initial pulse rate was set to $r_0 = 0.5$, and the pulse-rate growth coefficient γ was set to 0.05. For the improved bat algorithm, the population size, maximum iterations, and fitness-function coefficients were chosen to balance accuracy and efficiency. A moderate population size was enough to maintain diversity, while a larger one increased computation without clear gains. The maximum iterations were set according to the observed convergence trend, and the fitness coefficients were tuned to improve segmentation accuracy, reduce false alarms, and avoid overly complex threshold solutions. Through a multi-generation evolutionary mechanism, the algorithm first randomly selected samples to generate initial candidate thresholds. It then performed global optimization over the threshold solution space by combining a globally updated position with random perturbation and a local fine-search strategy with an adaptive decay step size. Meanwhile, candidate thresholds were evaluated using the multi-constraint fused fitness function. The optimal threshold for oil-slick segmentation within the ROI was then determined automatically, generating the initial segmentation result shown in Figure 6a. To improve the segmentation quality, isolated noise was removed from the initial segmentation result. Additionally, isolated noisy regions inside and outside the oil spill target were filtered using a connected-component area threshold, as shown in Figure 6b. To visually present the spatial distribution of the segmentation result, the final segmentation mask was fused with the denoised radar image, as shown in Figure 6c.

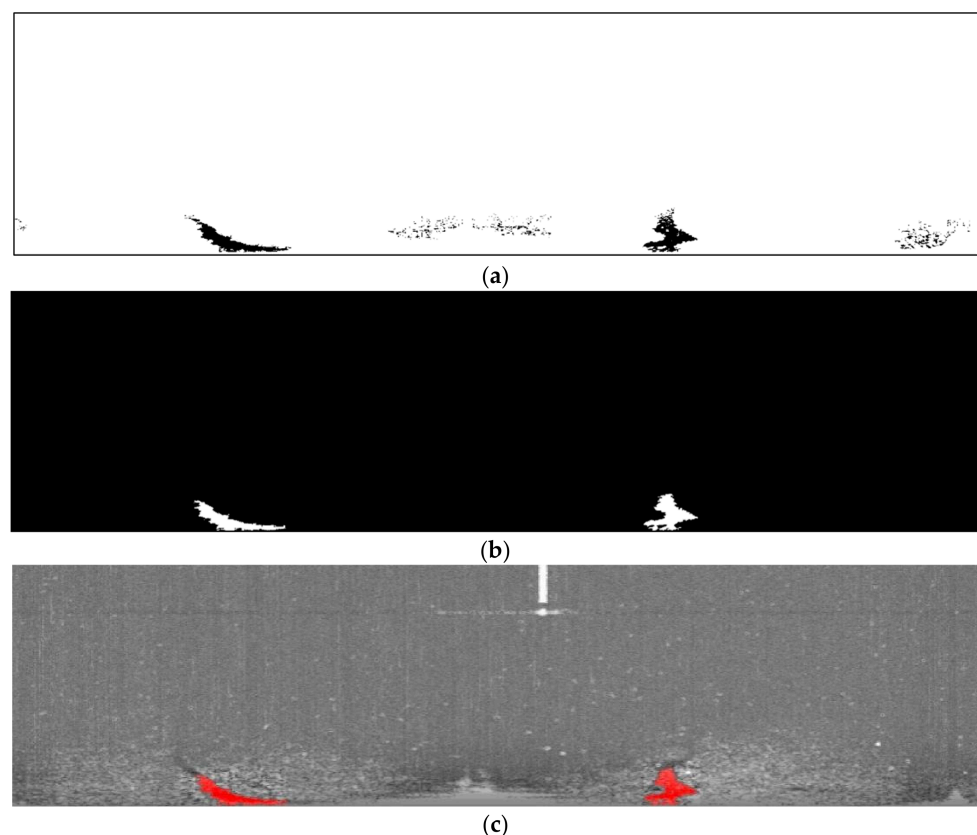


Figure 6. Oil spill target segmentation result. (a) Preliminary segmentation, (b) noise removal, (c) fused result.

To facilitate the presentation of shipborne radar oil spill detection results, the visualization output was further transformed into a polar coordinate system, as shown in Figure 7. The first sample is shown in Figure 7a, and the new test sample is shown in Figure 7b. This result can intuitively reflect the distribution characteristics of the oil spill target in the radar detection space and provide a basis for the subsequent quantitative analysis of oil spill features and the verification of algorithm performance.

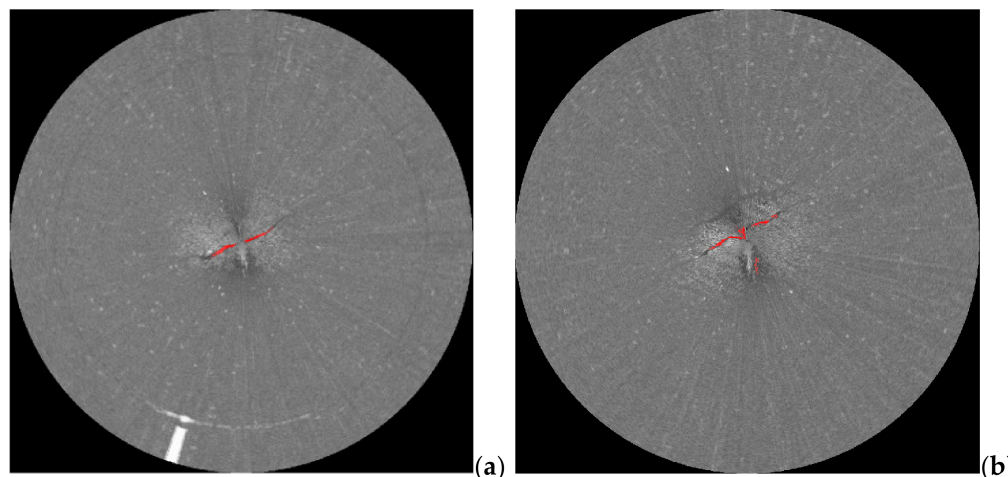


Figure 7. Result in the polar coordinate system. (a) Sample 1. (b) Sample 2.

5. Discussion

5.1. The Comparative Analysis with Classic Optimization Segmentation Algorithms

To comprehensively evaluate the performance of the proposed oil spill segmentation method, three classical intelligent optimization-based segmentation algorithms were selected for comparison, namely the genetic algorithm-based maximum inter-class variance method (GA_Otsu, called Method 1 here), the ant colony optimization-based Kapur entropy method (ACO_Kapur, called Method 2 here), and the particle swarm optimization-based Tsallis entropy method (PSO_Tsallis, called Method 3 here). The segmentation results of these methods on the oil spill detection ROI image are shown in Figure 8. For Method 1, the optimal threshold was 60. Although the oil spill contour could be captured, the method was sensitive to weak edges, resulting in fragmentation and loss of small oil patches and thus insufficient integrity. For Method 2, the optimal threshold was 76. Its segmentation performance was better than that of Method 1. However, it still showed deficiencies in preserving edge details and maintaining consistency in weak-signal regions, and under-segmentation was observed in parts of the oil spill region to varying degrees. For Method 3, the optimal threshold was 86. Compared with the first two methods, the integrity of the segmentation result was improved. Nevertheless, noticeable over-segmentation still appeared in the bow and stern regions, along with many small false-positive regions, as indicated by the red boxes, indicating that its ability to distinguish oil spills from background noise remained limited. In contrast, the proposed method better preserved the true morphology of the oil spill, produced clearer and more continuous boundaries, and exhibited higher consistency with the actual covered area. Although a small number of fragmented false-positive regions were still present in the initial segmentation result, their number and density were both lower than those of Method 3. This indicates that the proposed method achieved better overall performance in reducing false positives and false negatives.

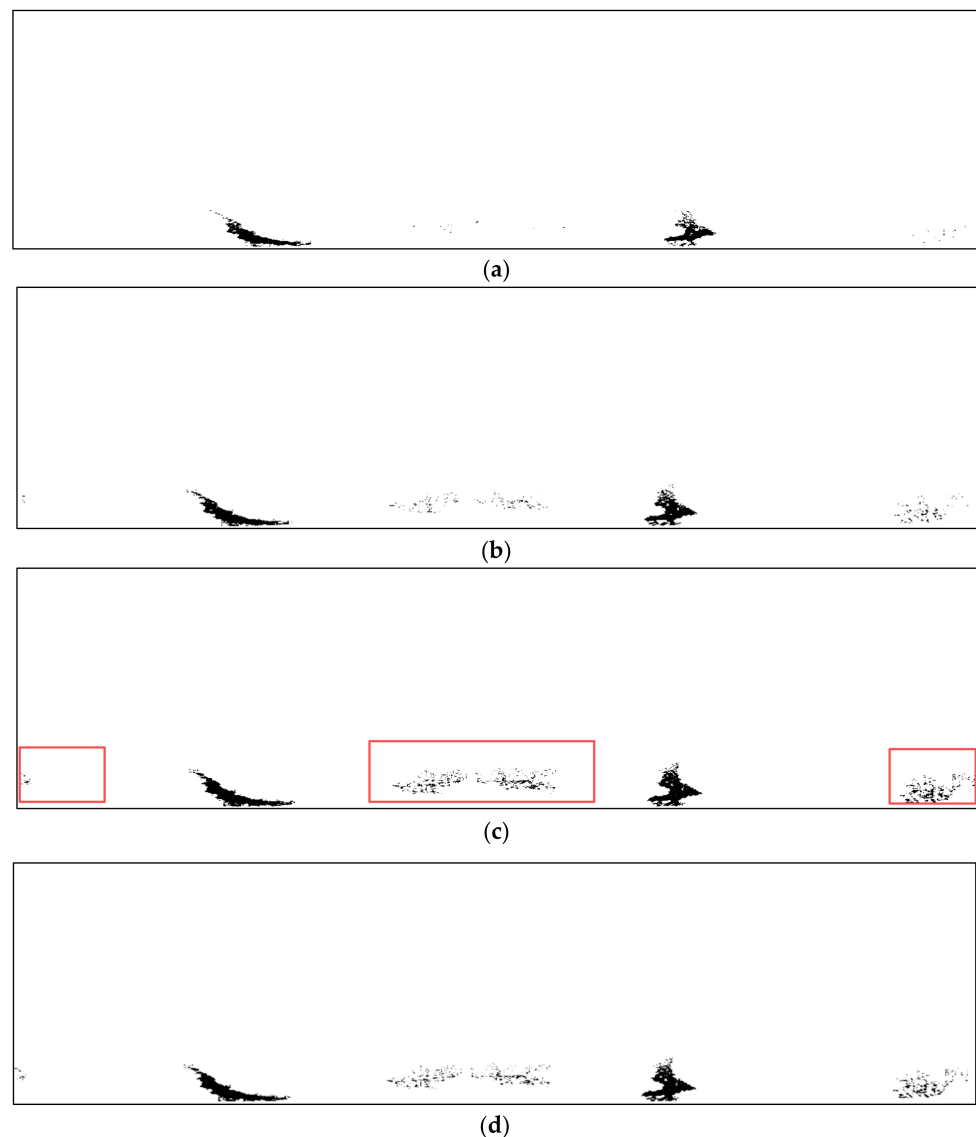


Figure 8. The segmentations of classic intelligent optimization algorithms. (a) GA_Otsu, (b) PSO_Tsallis, (c) ACO_Kapur, (d) the segmentation of our method.

5.2. Comparison with Domain-Relevant Baselines

To address the concern regarding the absence of modern and domain-relevant baselines, this section adds comparative experiments with methods that have been applied to shipborne X-band radar oil spill imagery. The comparison focuses on both visual extraction quality and the ability of each method to preserve oil-slick continuity under background clutter.

Figure 9 presents a visual comparison between the GLCM + K-Means + Sauvola method and the proposed method. As shown in Figure 9a, the GLCM + K-Means + Sauvola method can identify the main oil-slick regions, indicating that texture-based features are useful for locating dark oil-slick targets in shipborne radar imagery. However, several limitations remain. First, the extracted boundaries are relatively rough. Second, some local regions are not spatially continuous. Third, a few isolated false-positive white spots still appear in the background. Fourth, the extracted oil-slick area is noticeably smaller than that obtained by the proposed method, suggesting under-segmentation of weak oil-slick regions.

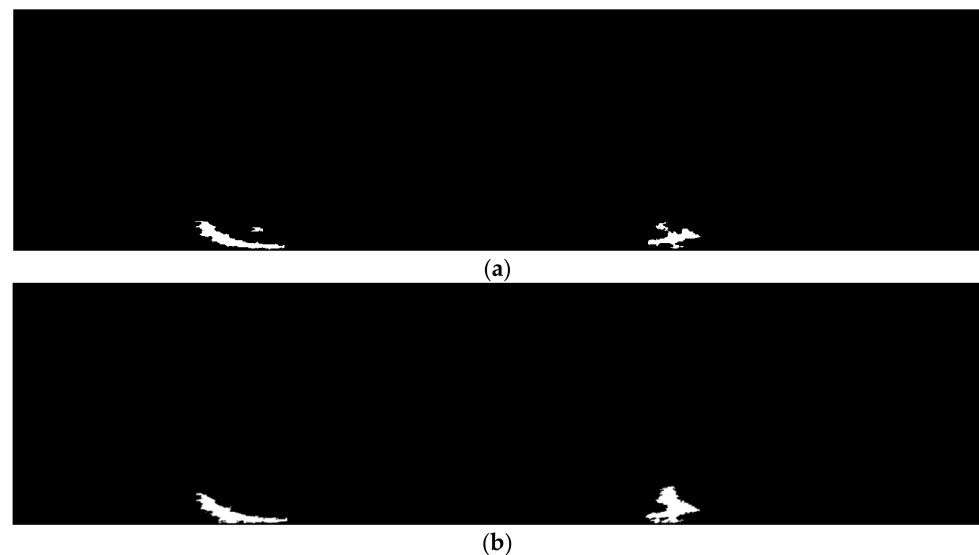


Figure 9. Visual comparison with domain-relevant baselines. (a) GLCM + K-Means + Sauvola, (b) proposed method.

By contrast, the proposed method in Figure 9b produces a more compact and cleaner extraction result. The two main oil-slick targets are both preserved, isolated background noise is reduced, and the boundaries are smoother and more spatially coherent. These results indicate that the proposed framework has stronger capability in suppressing background clutter and preserving the integrity of oil-slick regions.

In addition to the GLCM + K-Means + Sauvola method, the YOLOv8 + SA-PSO method of Xu et al. [32] was included as a learning-based domain-relevant baseline. This method is closely related to the present study because it was developed for shipborne X-band radar oil spill imagery and combines deep-learning-based target localization with swarm-intelligence-based refinement. For consistency, the same preprocessed radar image was used for visual comparison.

As shown in Figure 10a, YOLOv8 + SA-PSO can identify several candidate oil-slick regions in the lower part of the radar image, indicating its ability to locate suspicious targets under strong background clutter. However, several limitations can be observed. First, the result is mainly represented by rectangular candidate regions, which inevitably include a considerable number of background pixels. Second, some detection boxes overlap or appear spatially fragmented, particularly in the right oil-slick area. Third, weak and elongated oil-slick boundaries are not accurately delineated. In addition, local high-intensity sea-surface clutter may introduce redundant candidate boxes.

By contrast, the proposed method generates a pixel-level binary extraction result, as shown in Figure 10b. The extracted oil-slick regions are more consistent with the elongated dark structures in the radar image, and the left, middle, and right oil-slick targets are preserved without being represented by coarse rectangular boxes. Compared with YOLOv8 + SA-PSO, the proposed method reduces background over-coverage and provides a clearer description of oil-slick boundaries. These results suggest that YOLOv8 + SA-PSO is effective for rapid candidate-region localization, whereas the proposed method is more suitable for fine pixel-level oil-slick extraction in this case.

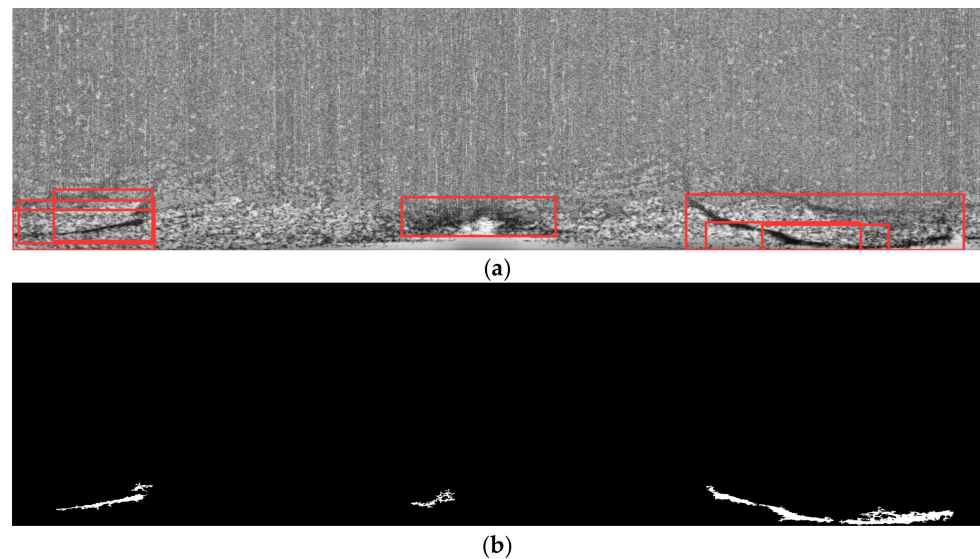


Figure 10. Visual comparison with domain-relevant baselines. (a) YOLOv8 + SA-PSO, where the red bounding boxes indicate candidate oil-slick regions. (b) proposed method.

The fully supervised CNN-, Transformer-, and GAN-based models reviewed in Section 1 were not directly trained and tested in the main comparison because the shipborne radar data used in this study contain limited annotated scenes. Training such models on the same image used for evaluation would introduce data leakage and would not provide a fair assessment of generalization ability. Therefore, the comparison focuses on methods that can be applied under limited-label or no-label conditions and are closely related to shipborne X-band radar imagery, including classical optimization methods, the added domain-relevant baselines, and ablation experiments. Accordingly, the revised manuscript avoids claiming general superiority over all modern deep learning methods.

5.3. Comparative Analysis with the Conventional Bat Algorithm

In the segmentation results of the conventional bat algorithm, oil spill targets were missing in the regions marked by the green boxes as shown in Figure 11. In addition, obvious holes appeared inside the target, thereby compromising the integrity of the oil spill. Although this method could effectively suppress fragmented false-positive regions in the bow and stern areas, its ability to extract the main body of the oil spill remained insufficient. In contrast, the proposed method produced a more continuous oil spill contour with a significantly reduced hole area inside the target. This allows for a more accurate reflection of the actual coverage of the oil spill, thus improving the consistency between the segmentation result and the true distribution.

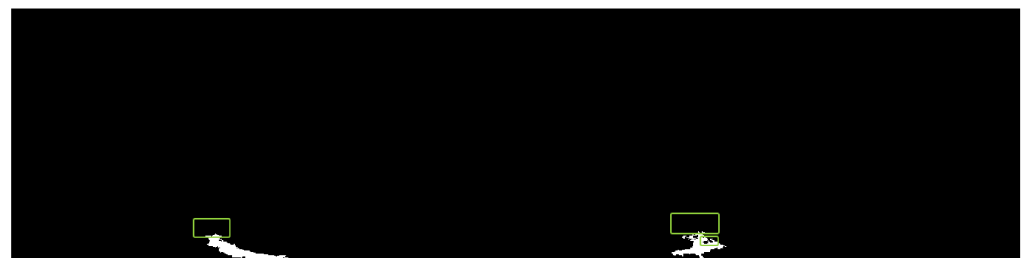


Figure 11. The segmentation result of the traditional bat algorithm.

To further evaluate the computational efficiency and convergence behavior of the proposed improved bat algorithm, an additional comparison with the conventional bat algorithm was conducted under the same parameter setting, namely population size $N = 30$

and maximum iteration number $T = 30$, as shown in Table 2 (efficiency comparison of the conventional and improved bat algorithms). Each algorithm was independently repeated 30 times on a 512×2048 image. The timing was measured only for the bat-algorithm threshold optimization core, excluding image display, file writing, post-processing, and coordinate transformation.

Table 2. Efficiency comparison of the conventional and improved bat algorithms.

Metric	Conventional BA	Improved BA	Improvement
Runtime per 512×2048 image	0.1366 ± 0.0038 s	0.1232 ± 0.0059 s	9.77% reduction, $1.11 \times$ speedup
Iterations to 95% convergence	8.30 ± 8.04	1.00 ± 0.00	87.95% fewer iterations, $8.30 \times$ faster

The average runtime of the improved BA was 0.1232 ± 0.0059 s per image, whereas the conventional BA required 0.1366 ± 0.0038 s per image. This corresponds to a 9.77% reduction in runtime and a $1.11 \times$ speedup. Therefore, the proposed improvement reduced the required convergence iterations by 87.95%, corresponding to an $8.30 \times$ faster convergence rate. It should be noted that the raw objective values of the two algorithms are not directly comparable because the improved BA adopts a multi-constraint fitness function, whereas the conventional BA uses only the Otsu between-class variance criterion. Therefore, normalized convergence progress was used to compare the convergence behavior. The results indicate that the proposed improvement mainly enhances convergence speed and convergence consistency. The overall CPU-based workflow still requires further acceleration for strict frame-by-frame real-time deployment.

5.4. Autoencoder Ablation Experiment

In the segmentation result obtained after removing the autoencoder, obvious misclassifications appeared in the regions marked by the blue boxes shown in Figure 12. Background regions that did not belong to the oil spill target were incorrectly labeled as oil spill, severely damaging the integrity and accuracy of the target region. Meanwhile, a large area of false positives appeared in the bow and stern regions, where substantial background noise unrelated to the oil spill was incorrectly identified as an oil spill target. This phenomenon seriously interfered with the judgment of the true oil spill distribution. In contrast, the proposed algorithm demonstrated clear advantages in segmentation accuracy and robustness. Its segmentation result fit the actual contour of the oil spill target more precisely, with clear and continuous boundaries, effectively avoiding the mis-segmentation problem. At the same time, the area of false-positive regions in the bow and stern regions was significantly smaller than that of the compared method. This greatly reduced the interference of background noise and made oil spill target identification more accurate and reliable, with higher consistency with the true oil spill distribution.

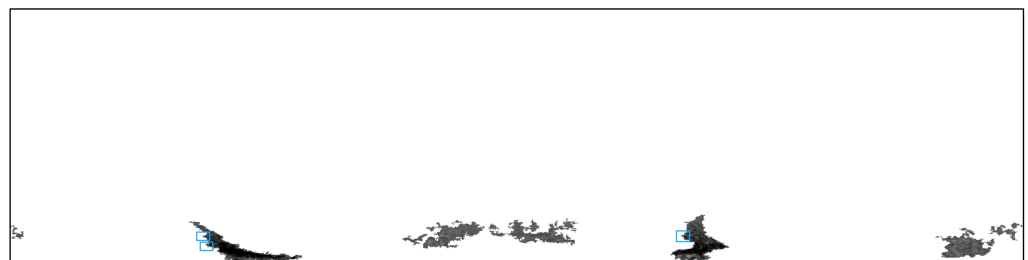


Figure 12. The segmentation result with the autoencoder removed.

5.5. Comparative Analysis of Replacing K-Means with GMM

To clarify the GMM replacement experiment, the exact GMM configuration was added. In the pseudo-label generation stage, the Gaussian-smoothed grayscale intensity values were modeled using a three-component GMM. In the state-discretization stage, the PCA-reduced three-dimensional autoencoder features were modeled using a 50-component GMM, so that the number of discrete Q-learning states remained consistent with the proposed K-Means-based setting. Both GMMs were estimated using the EM algorithm with diagonal covariance matrices, three replicates, and regularization values of 1×10^{-6} and 1×10^{-5} , respectively.

In the segmentation result obtained by replacing K-Means with GMM, the GMM-based method could roughly outline the overall contour of the oil spill target. However, a large amount of co-frequency interference noise appeared in the key region marked by the yellow box. Part of the noise was incorrectly identified as the oil spill region, resulting in obvious false-positive interference and affecting the judgment of the true oil spill coverage. In contrast, the proposed method performed better in the results shown in Figure 13. It could not only identify the main oil spill region more accurately but also effectively suppress the interference noise within the yellow box, reducing the misclassification of bright regions as an oil spill.

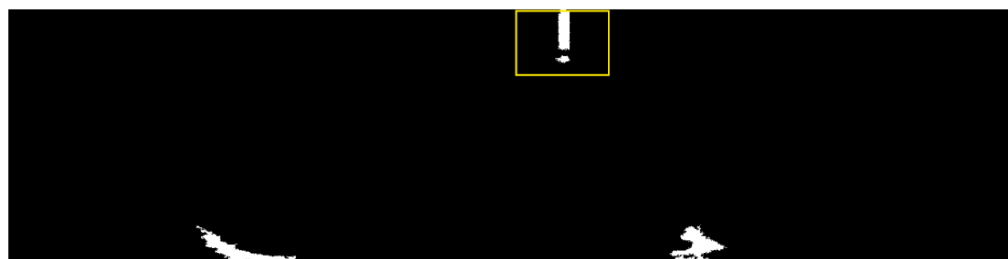


Figure 13. The segmentation results after replacing K-Means with GMM.

The decrease in Dice and IoU after replacing K-Means with GMM may be attributed to the sensitivity of GMM to speckle noise, co-frequency interference, and non-Gaussian radar clutter. In particular, local noise and interference may be fitted as Gaussian components and then incorrectly assigned to oil-slick-related regions. Therefore, the degradation is interpreted as noise-sensitive component fitting and possible over-partitioning of the feature space rather than supervised overfitting.

5.6. Q-Learning Ablation Experiment

To further verify the contribution of the Q-learning module in ROI localization, an ablation experiment was conducted by removing Q-learning from the proposed framework. In this variant, the autoencoder feature extraction, PCA dimensionality reduction, and K-Means state discretization were retained, while the Q-table, ϵ -greedy action selection, reward function, and iterative value update process were completely removed. Instead, the final class of each discrete state was determined directly by majority voting based on the K-Means pseudo-labels. Therefore, this experiment evaluates whether the reinforcement-learning-based decision mechanism can improve oil-slick region extraction beyond a simple state-level clustering assignment.

As shown in Figure 14, after removing Q-learning, the extracted oil spill regions became noticeably incomplete and fragmented. Several weak oil-slick areas were missed, especially in the central and right parts of the radar image. In addition, the segmented regions were thinner and less continuous than those obtained by the proposed method. This indicates that direct state-label mapping is insufficient to maintain spatial consistency under

speckle noise and co-frequency interference. Without the reward-guided update process, the model mainly depends on local feature clustering and lacks an adaptive mechanism to correct uncertain pixel-level decisions.

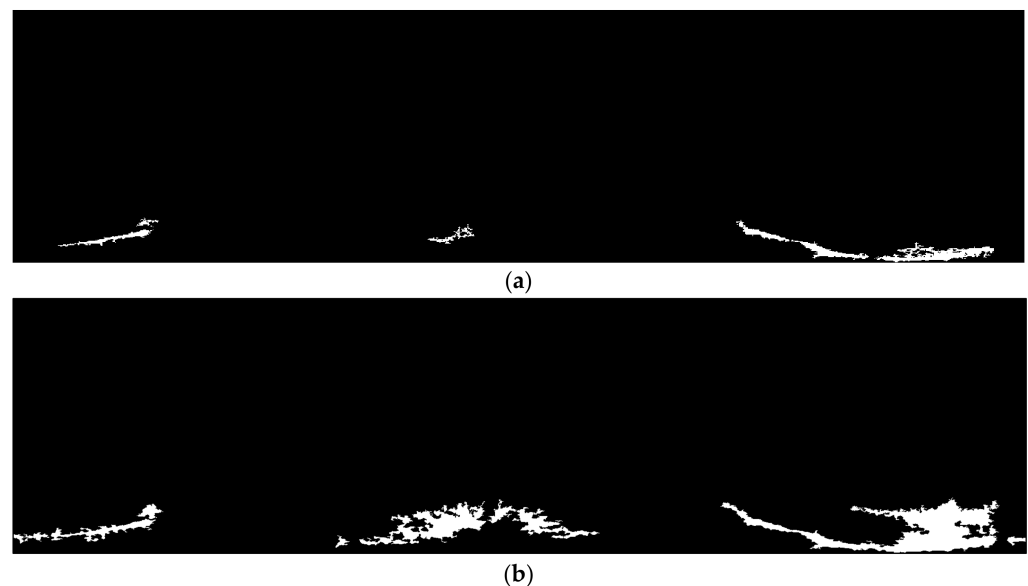


Figure 14. Q-learning ablation experiment. (a) Result of the proposed method with Q-learning, (b) result after removing Q-learning.

In contrast, the proposed method with Q-learning produced more complete and continuous ROI extraction results. The neighborhood-consistency term in the reward function helped suppress isolated noisy responses and encouraged adjacent pixels with similar spatial structures to obtain consistent labels. Meanwhile, the pseudo-label agreement term guided the Q-table toward stable state-action mapping. As a result, Q-learning improved the robustness of ROI localization and provided a more reliable search region for the subsequent improved bat-algorithm segmentation.

These results demonstrate that Q-learning is not merely an additional classification step, but an important adaptive decision module in the proposed unsupervised framework. By combining pseudo-label guidance and local spatial consistency, Q-learning effectively reduces missed detection and fragmentation in weak oil-slick regions, thereby improving the completeness and stability of the final oil spill extraction.

5.7. Quantitative Comparative Analysis

The metric results in Figure 15 and Table 3 show that the proposed model performed well in the oil spill segmentation task. Specifically, the accuracy reached 99.78%, indicating that the model achieved a high overall discriminative ability in pixel-level classification. The precision was 89.65%, reflecting that a high proportion of the regions predicted as oil spills were true oil spills, which demonstrates good prediction reliability. The Dice coefficient reached 84.75%, indicating a high degree of overlap between the oil spill region segmented by the proposed model and the true oil spill region. The specificity was 99.93%, showing that the algorithm could effectively avoid misclassifying background regions as oil spill targets, which is well suited to the high-gray-value characteristics of oil spill regions in radar images. The IoU reached 73.54%, directly reflecting the spatial agreement of the segmentation result, and this value fully demonstrates that the segmentation boundary of the algorithm matches the true oil spill boundary well.

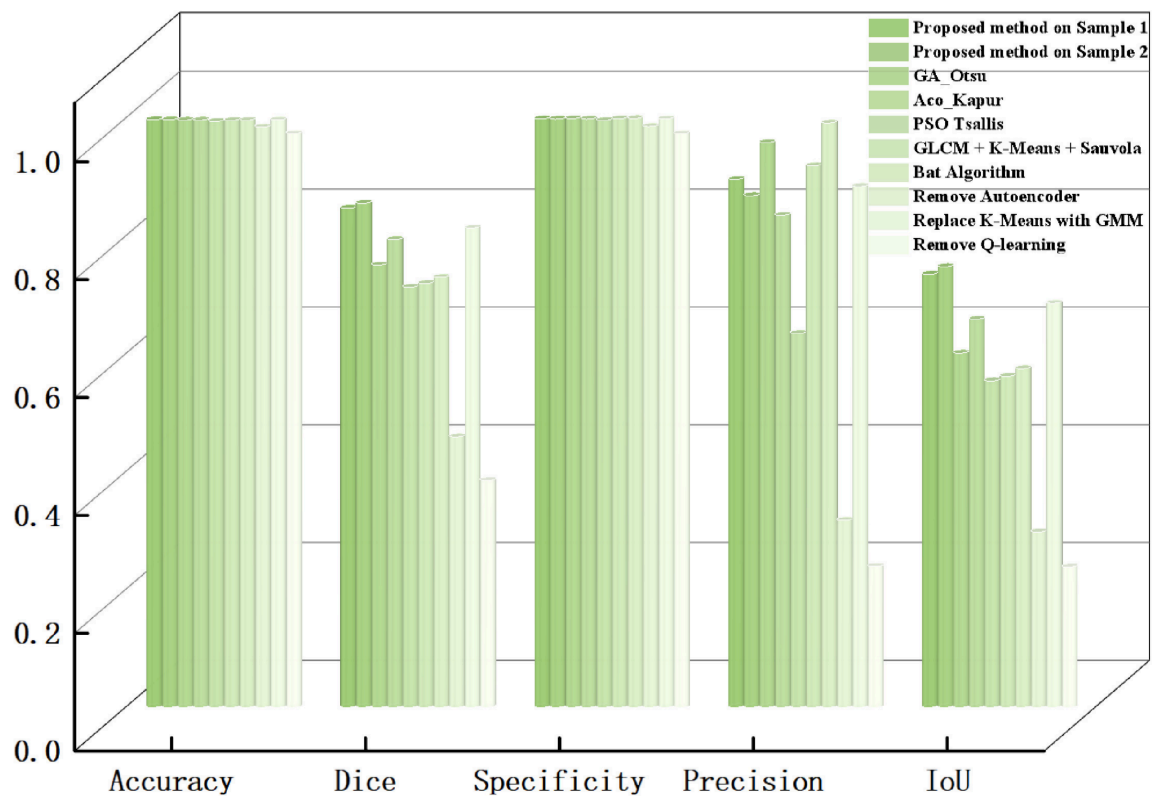


Figure 15. Quantitative comparison of segmentation results using manually interpreted reference masks.

Table 3. Quantitative comparison of different oil spill segmentation methods.

Method	Accuracy	Dice	Specificity	Precision	IoU
Proposed Method on Sample 1	99.78%	84.75%	99.93%	89.65%	73.54%
Proposed Method on Sample 2	99.77%	85.61%	99.89%	86.88%	74.84%
GA_Otsu	99.69%	75.13%	99.98%	95.90%	60.17%
Aco_Kapur	99.70%	79.44%	99.89%	83.58%	65.90%
PSO Tsallis	99.51%	71.36%	99.64%	63.49%	55.47%
GLCM + K-Means + Sauvola	99.65%	71.99%	99.96%	91.96%	56.24%
Bat Algorithm	99.68%	73.03%	100.00%	99.24%	57.52%
Remove Autoencoder	98.52%	45.96%	98.64%	31.78%	29.84%
Replace K-Means with GMM	99.74%	81.40%	99.92%	88.46%	68.63%
Remove Q-learning	97.42%	38.56%	97.41%	23.98%	23.88%

Although the manual reference masks were generated independently of the K-Means pseudo-labels, the manual delineation of oil-slick boundaries in shipborne radar imagery remains subject to uncertainty because oil-slick edges are often blurred and affected by speckle noise, co-frequency interference, and sea-surface clutter. Therefore, the quantitative metrics should be interpreted as measures of agreement with manually delineated reference masks rather than absolute physical measurements of the oil spill boundary. Future work will involve multiple independent annotators and inter-annotator consistency analysis to further improve the reliability of reference-mask generation.

5.8. Computational Efficiency Analysis

To evaluate the computational efficiency of the proposed framework, the runtime of each major processing stage was recorded on a CPU-based desktop platform. The tested

shipborne radar image had a size of 512×2048 pixels. The complete workflow consisted of three main stages: image preprocessing, ROI extraction, and oil-slick segmentation within the extracted ROI.

The preprocessing stage required 0.300 s. The ROI extraction stage based on autoencoder-enhanced Q-learning required 54.062 s. Within this stage, image reading and normalization took only 0.007 s, while three-class K-Means pseudo-label generation required 5.336 s. The autoencoder feature extraction module required 1.176 s, including 0.015 s for autoencoder training and 1.157 s for full-image feature extraction. PCA dimensionality reduction required 0.298 s. The most time-consuming operation was the K-Means discretization of the PCA-reduced feature space into 50 states, which required 30.724 s. The Q-learning training process required 14.831 s for ten training episodes, and the final ROI generation and saving step required 1.673 s.

After ROI extraction, the oil-slick segmentation stage required 3.2204 s. In this stage, the core loop of the improved bat algorithm required 1.0867 s, while binary post-processing and result saving required 0.6479 s. Image fusion and color composition required 0.4003 s, and polar-coordinate transformation and interpolation required 0.2043 s.

As shown in Table 4, the runtime distribution is highly unbalanced across different processing stages. The ROI extraction stage accounts for 93.89% of the total computational cost, whereas image preprocessing and oil-slick segmentation contribute only 0.52% and 5.59%, respectively. These results indicate that the final segmentation module is relatively lightweight compared with the ROI extraction stage. Overall, the complete processing time for one 512×2048 radar image was approximately 57.582 s, including 0.300 s for preprocessing, 54.062 s for ROI extraction, and 3.2204 s for oil-slick segmentation. The runtime distribution shows that the computational cost is mainly concentrated in the ROI extraction stage, especially the 50-state K-Means discretization and Q-learning training. In contrast, autoencoder feature extraction, PCA dimensionality reduction, and the improved bat-algorithm-based segmentation introduce relatively limited computational overhead. Therefore, the current CPU implementation can provide timely oil-slick extraction results, while further acceleration of the state discretization and reinforcement learning stages would be necessary for strict frame-by-frame real-time deployment.

Table 4. Runtime distribution and proportion of each processing stage.

Stage Name	Runtime (s)	Percentage of Total Time	Remarks
Image Preprocessing	0.300	0.52%	Core bottleneck stage
ROI Extraction	54.062	93.89%	
Oil-Slick Segmentation	3.220	5.59%	
Total Full Process	57.582	100.00%	512×2048 -pixel radar image

5.9. Limitation on Look-Alike Discrimination and Multi-Sensor Verification

Although the proposed framework achieved accurate oil-slick extraction from shipborne X-band radar imagery, this study was based mainly on a single radar sensing modality. Therefore, the current method does not explicitly distinguish oil slicks from radar-dark look-alike phenomena, such as biogenic films, internal waves, low-wind areas, and ship wakes. This is an important limitation for operational deployment.

To reduce the uncertainty of radar-only interpretation, auxiliary visual and infrared observations were used for qualitative verification when available. As shown in Figure 16a, daytime visible-light observation confirmed the presence of a surface film spatially consistent with the radar-detected slick. At night, infrared imagery in Figure 16b also showed an abnormal surface pattern corresponding to the suspected oil-slick region. These observa-

tions support the interpretation that the detected radar-dark region was associated with an actual oil spill rather than a purely radar-induced artifact.

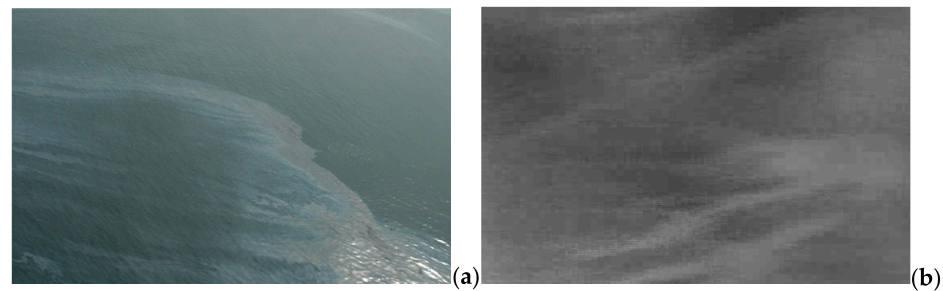


Figure 16. Qualitative auxiliary verification of the radar-detected oil-slick region. (a) Daytime visible-light image, (b) nighttime infrared image.

However, these auxiliary observations were used only for post-event qualitative confirmation and were not integrated into the detection algorithm. The present framework therefore cannot yet be regarded as a complete look-alike discrimination system. Future work will construct multi-scenario datasets including confirmed oil slicks and typical look-alikes, and will incorporate multi-source information such as visible-light images, infrared images, wind fields, ship trajectory data, and temporal radar sequences to improve look-alike suppression.

5.10. Ablation Study on the Number of K-Means States

To justify the selection of the number of K-Means states in the Q-learning state space, an ablation study was conducted by varying k from 20 to 70. The quantitative results are shown in Table 5. When k was set to 20 or 30, the Dice coefficient and IoU were relatively low, indicating that an insufficient number of states may not fully represent the PCA-reduced feature space. When k increased to 40–60, the segmentation performance became stable. Specifically, $k = 40$ achieved the highest Dice coefficient of 85.64% and IoU of 74.88%, while $k = 50$ achieved very similar values of 85.61% and 74.84%, respectively. The differences between $k = 40$ and $k = 50$ were only 0.03% in Dice and 0.04% in IoU.

Table 5. Ablation results for different numbers of K-Means-clusters.

k	Accuracy	Dice	Specificity	Precision	IoU
20	99.75%	82.39%	99.98%	97.44%	70.05%
30	99.71%	82.82%	99.85%	82.12%	70.67%
40	99.78%	85.64%	99.94%	92.24%	74.88%
50	99.77%	85.61%	99.89%	86.88%	74.84%
60	99.76%	85.25%	99.89%	86.52%	74.29%
70	99.75%	82.79%	99.98%	97.38%	70.63%

When k was further increased to 70, the Dice coefficient and IoU decreased to 82.79% and 70.63%, respectively, suggesting that an overly fine state partition does not bring additional benefits and may reduce the robustness of Q-learning state representation. Therefore, the experimental results indicate that the model is relatively insensitive to the exact value of k within the range of 40–60. Considering both segmentation accuracy and state-space granularity, $k = 50$ was selected as a balanced setting rather than an arbitrary choice.

The results also indicate that the 50-state discretization preserved the discriminative information of the PCA-reduced autoencoder features. If the discretization were too coarse, different radar patterns such as oil slicks, background clutter, and suspected oil spill regions

would be merged into the same state, resulting in lower segmentation accuracy. This was observed for $k = 20$ and $k = 30$, where the Dice and IoU were lower. Conversely, overly fine partitioning increased state fragmentation and reduced robustness, as shown by the performance decrease at $k = 70$. The stable performance within $k = 40$ – 60 , together with the high Dice and IoU values obtained at $k = 50$, demonstrates that the 50-state representation retained sufficient discriminative information for Q-learning-based ROI extraction.

6. Conclusions

This study focused on the requirements of automated shipborne radar oil spill detection and segmentation. It developed an unsupervised detection framework that integrates lightweight autoencoder-based feature representation, reinforcement-learning-based ROI localization, and intelligent optimization-based threshold segmentation. Experimental results on real Dalian Bay oil spill radar data show that the combination of an autoencoder and PCA can effectively extract local nonlinear features from radar images. This provides an effective feature representation for subsequent reinforcement-learning-based classification. Guided by K-Means pseudo-labels, Q-learning can achieve stable ROI extraction for oil spill detection while reducing the search space and background interference in subsequent segmentation. Through dynamic weighting, random perturbation, and a multi-constraint fitness function, the improved bat algorithm enhances the global optimization capability of threshold search. It also alleviates, to a certain extent, the issues of local optima, threshold sensitivity, and incorrect segmentation of ship structures and noise that are common in conventional optimization algorithms. Both quantitative and qualitative comparisons were conducted. The results indicate that the proposed model outperforms traditional thresholding methods, classical intelligent optimization algorithms, and simplified models without feature learning in terms of accuracy, precision, Dice coefficient, and IoU. Without manual annotation, the proposed method can obtain oil spill segmentation results with more continuous boundaries, more complete internal regions, and fewer false-positive areas. These characteristics better adapt to the imaging characteristics of shipborne radar and the requirements for timely processing and robustness in maritime emergency scenarios. This provides a reference for unmanned nearshore oil spill monitoring, intelligent recognition and quantitative assessment. From a DTO-oriented perspective, although this study does not develop a complete ocean digital twin, the proposed method can serve as an automatic perception module for shipborne radar oil spill monitoring. Its extracted ROI, mask, boundary, centroid, and area can provide minute-level timely inputs for drift/dispersion models, multi-source fusion, and decision support. Future work will couple these outputs with hydrodynamic, wave, wind, and oil-transport models.

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

Funding: This research was supported by the Guangdong Basic and Applied Basic Research Foundation, grant numbers 2024A1515110096 and 2025A1515010886; the National Natural Science Foundation of China, grant number 52271359; the Postgraduate Education Innovation Project of Guangdong Ocean University, grant numbers 202421, 202539, and 202551; the Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching, grant number 080508132401; and the Zhanjiang Non-Funded Science and Technology Research Project, grant number 2024B01049.

Data Availability Statement: The data collection department did not agree to share the analysis data.

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

References

- Chen, J.; Chen, H.; Shi, J.; Yan, T.; Gu, M.; Huang, X. Factor Diagnosis and Governance Strategies of Ship Oil Spill Accidents Based on Formal Concept Analysis. *Mar. Pollut. Bull.* **2023**, *196*, 115606. [\[CrossRef\]](#)
- Geng, B.; Zhou, W.; Lu, L.; Cao, Q.; Hong, M.; Wen, D. Risk Assessment of Offshore Oil Spill in the Northern South China Sea: A Stochastic Scenario Modeling Approach. *Front. Mar. Sci.* **2022**, *9*, 986379. [\[CrossRef\]](#)
- Chen, J.; Zhang, W.; Wan, Z.; Li, S.; Huang, T.; Fei, Y. Oil Spills from Global Tankers: Status Review and Future Governance. *J. Clean. Prod.* **2019**, *227*, 20–32. [\[CrossRef\]](#)
- Liu, J.; Tao, L.; Zhu, D.; Zhang, S.; Wang, J. Mitigating Marine Oil Spills: A Review of Spill Behavior, Response Technologies, and Emerging Strategies. *Mar. Pollut. Bull.* **2026**, *224*, 119178. [\[CrossRef\]](#)
- Keramea, P.; Zodiatis, G.; Sylaios, G. Analysis of Major Global Oil Spill Incidents: Part 1—Environmental and Ecological Impacts. *J. Mar. Sci. Eng.* **2026**, *14*, 153. [\[CrossRef\]](#)
- Eklund, R.L.; Knapp, L.C.; Sandifer, P.A.; Colwell, R.C. Oil Spills and Human Health: Contributions of the Gulf of Mexico Research Initiative. *GeoHealth* **2019**, *3*, 391–406. [\[CrossRef\]](#)
- Hong, S.; Yoon, S.J.; Kim, T.; Ryu, J.; Kang, S.-G.; Khim, J.S. Response to Oiled Wildlife in the Management and Evaluation of Marine Oil Spills in South Korea: A Review. *Reg. Stud. Mar. Sci.* **2020**, *40*, 101542. [\[CrossRef\]](#)
- Beyer, J.; Trannum, H.C.; Bakke, T.; Hodson, P.V.; Collier, T.K. Environmental Effects of the Deepwater Horizon Oil Spill: A Review. *Mar. Pollut. Bull.* **2016**, *110*, 28–51. [\[CrossRef\]](#) [\[PubMed\]](#)
- Keramea, P.; Spanoudaki, K.; Zodiatis, G.; Gikas, G.; Sylaios, G. Oil Spill Modeling: A Critical Review on Current Trends, Perspectives, and Challenges. *J. Mar. Sci. Eng.* **2021**, *9*, 181. [\[CrossRef\]](#)
- Scarlett, A.G.; Nelson, R.K.; Gagnon, M.M.; Reddy, C.M.; Grice, K. Very Low Sulfur Fuel Oil Spilled from the MV Wakashio in 2020 Remains in Sediments in a Mauritius Mangrove Ecosystem Nearly Three Years after the Grounding. *Mar. Pollut. Bull.* **2024**, *209*, 117283. [\[CrossRef\]](#) [\[PubMed\]](#)
- Scarlett, A.G.; Nelson, R.K.; Gagnon, M.M.; Holman, A.; Reddy, C.M.; Sutton, P.A.; Grice, K. MV Wakashio Grounding Incident in Mauritius 2020: The World's First Major Spillage of Very Low Sulfur Fuel Oil. *Mar. Pollut. Bull.* **2021**, *171*, 112917. [\[CrossRef\]](#)
- Guo, J.; Liu, X.; Xie, Q. Characteristics of the Bohai Sea Oil Spill and Its Impact on the Bohai Sea Ecosystem. *Chin. Sci. Bull.* **2013**, *58*, 2276–2281. [\[CrossRef\]](#)
- Guo, W.; Zhang, S.; Wu, G. Quantitative Oil Spill Risk from Offshore Fields in the Bohai Sea, China. *Sci. Total Environ.* **2019**, *688*, 494–504. [\[CrossRef\]](#)
- Temitope Yekeen, S.; Balogun, A.-L. Advances in Remote Sensing Technology, Machine Learning and Deep Learning for Marine Oil Spill Detection, Prediction and Vulnerability Assessment. *Remote Sens.* **2020**, *12*, 3416. [\[CrossRef\]](#)
- Fan, J.; Sui, Z.; Wang, X. Multiphysical Interpretable Deep Learning Network for Oil Spill Identification Based on SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 4202015. [\[CrossRef\]](#)
- Liu, P.; Liu, B.; Li, Y.; Chen, P.; Xu, J. Oil Spill Detection on X-Band Marine Radar Images Based on Sea Clutter Fitting Model. *Heliyon* **2023**, *9*, e20893. [\[CrossRef\]](#)
- Meng, T.; Nunziata, F.; Yang, X.; Buono, A.; Migliaccio, M. Model-Based Comparisons of near-Coincident TerraSAR-X and COSMO-SkyMed VV-Polarized SAR Measurements over Sea Surface with and without Oil Slicks. *Geo Spat. Inf. Sci.* **2024**, *27*, 822–835. [\[CrossRef\]](#)
- Metheniti, V.; Parasyris, A.; Fazzini, N.; Outmani, S.; Correia, M.; Goddard, J.; Alexandrakis, G.; Kozyrakis, G.V.; Vettorello, L.; Keeble, S.; et al. Coastal Crete: A Digital Twin of the Ocean for Oil Spill Identification and Forecasting. In *Proceedings of the OCEANS 2025 Brest, Brest, France, 16–19 June 2025*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–8.
- Metheniti, V.; Parasyris, A.; Pereira, R.S.; Kazanjian, G. Recent Advancements and Challenges in Artificial Intelligence for Digital Twins of the Ocean. *Climate* **2026**, *14*, 3. [\[CrossRef\]](#)
- Das, K.; Janardhan, P.; Narayana, H. Application of CNN Based Image Classification Technique for Oil Spill Detection. *Indian J. Geo Mar. Sci.* **2023**, *52*, 5–14. [\[CrossRef\]](#)
- Zhang, J.; Yang, P.; Ren, X. Detection of Oil Spill in SAR Image Using an Improved DeepLabV3+. *Sensors* **2024**, *24*, 5460. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ma, X.; Xu, J.; Wu, P.; Kong, P. Oil Spill Detection Based on Deep Convolutional Neural Networks Using Polarimetric Scattering Information from Sentinel-1 SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4204713. [\[CrossRef\]](#)
- Song, D.; Huang, Q.; Gao, H.; Wang, B.; Zhang, J.; Chen, W. Adaptive Oil Spill Detection Network for Scene-Based PolSAR Data Using Dynamic Convolution and Boundary Constraints. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *130*, 103914. [\[CrossRef\]](#)
- Chen, Y.; Wang, Z. Marine Oil Spill Detection from SAR Images Based on Attention U-Net Model Using Polarimetric and Wind Speed Information. *Int. J. Environ. Res. Public Health* **2022**, *19*, 12315. [\[CrossRef\]](#)
- Kang, J.; Yang, C.; Yi, J.; Lee, Y. Detection of Marine Oil Spill from PlanetScope Images Using CNN and Transformer Models. *J. Mar. Sci. Eng.* **2024**, *12*, 95. [\[CrossRef\]](#)

26. Cai, Y.; Chen, L.; Zhuang, X.; Zhang, B. Automated Marine Oil Spill Detection Algorithm Based on Single-Image Generative Adversarial Network and YOLO-v8 under Small Samples. *Mar. Pollut. Bull.* **2024**, *203*, 116475. [[CrossRef](#)]
27. Fan, J.; Liu, C. Multitask GANs for Oil Spill Classification and Semantic Segmentation Based on SAR Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 2532–2546. [[CrossRef](#)]
28. Zhan, C.; Bai, K.; Tu, B.; Zhang, W. Offshore Oil Spill Detection Based on CNN, DBSCAN, and Hyperspectral Imaging. *Sensors* **2024**, *24*, 411. [[CrossRef](#)] [[PubMed](#)]
29. Liu, X.; Zhang, Y.; Zou, H.; Wang, F.; Cheng, X.; Wu, W.; Liu, X.; Li, Y. Multi-Source Knowledge Graph Reasoning for Ocean Oil Spill Detection from Satellite SAR Images. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *116*, 103153. [[CrossRef](#)]
30. Yang, J.; Ma, Y.; Hu, Y.; Jiang, Z.; Zhang, J.; Wan, J.; Li, Z. Decision Fusion of Deep Learning and Shallow Learning for Marine Oil Spill Detection. *Remote Sens.* **2022**, *14*, 666. [[CrossRef](#)]
31. 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. [[CrossRef](#)]
32. Chen, R.; Li, B.; Jia, B.; Xu, J.; Ma, L.; Yang, H.; Wang, H. Oil Spill Identification in X-Band Marine Radar Image Using K-Means and Texture Feature. *PeerJ Comput. Sci.* **2022**, *8*, e1133. [[CrossRef](#)] [[PubMed](#)]
33. Liu, C.; Ma, Y.; Fan, J. Small sample generation adversarial network semantic segmentation based on optical remote sensing oil spill images. *Mar. Environ. Sci.* **2025**, *44*, 107–115. [[CrossRef](#)]
34. Han, Z.; Bai, L.; Pan, B.; Ren, P. From Mutual Guide to Confucius Tri-Learning: A Theoretical Justification. *Pattern Recognit.* **2026**, *178*, 113447. [[CrossRef](#)]
35. Carrasco-García, M.G.; Rodríguez-García, M.I.; Ruiz-Aguilar, J.J.; Deka, L.; Elizondo, D.; Turias Domínguez, I.J. Oil Spill Classification Using an Autoencoder and Hyperspectral Technology. *J. Mar. Sci. Eng.* **2024**, *12*, 495. [[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.