Enhanced Vessel Detection for Maritime Surveillance Using Hyperparameter-Tuned Deep Learning on SAR Images

S. Devika Priyadharshini[1] and K.Vadivazhagan[2]

[1]*Research Scholar, Department of Computer and Information Science, Faculty of Science, Annamalai University, Tamilnadu, India , priyadharshini.devika3@gmail.com*
[2]*Assistant Professor, Department of Computer and Information Science, Faculty of Science, Annamalai University, Tamilnadu, India, vadivazhagan.k@gmail.com*

KEYWORDS

Maritime surveillance, Vessel detection, Gaussian Filter, RetinaNet, ADAM optimizer, Hyperparameters, SAR images

ABSTRACT

In maritime surveillance, where ensuring the safety of shipping routes and detecting potential threats is paramount, the need for efficient vessel detection models is critical. Unauthorized vessels pose a significant threat to maritime security, competing for essential resources such as shipping lanes and port access. Traditional methods of vessel detection, such as manual monitoring or blanket radar scans, are time-consuming, labor-intensive, and often result in overuse of resources, leading to operational inefficiency and potential security breaches. This paper presents a Hyperparameter-Tuned Deep Learning model for Vessel Detection and Classification (HPTDL-VDAC) suitable for Maritime Surveillance applications. The proposed HPTDL-VDAC system integrates advanced techniques from computer vision and deep learning to accurately identify and classify vessels in Synthetic Aperture Radar (SAR) images. The workflow begins with pre-processing steps aimed at enhancing image quality and reducing noise. Specifically, a Gaussian Filter (GF) is employed to effectively remove noise from input images, followed by resizing to standard dimensions for subsequent analysis.

For object detection and classification, the RetinaNet model is employed. RetinaNet's innovative architecture, featuring a focal loss mechanism, enables robust detection of vessel instances amidst varying backgrounds and sea conditions. Notably, the hyperparameters of the RetinaNet model are fine-tuned using the ADAM optimizer, optimizing its performance for the specific task of vessel detection in maritime surveillance scenarios. A thorough simulation analysis of the HPTDL-VDAC technique was conducted using a benchmark dataset. Experimental results demonstrate the effectiveness of the proposed system in accurately detecting vessels in various maritime environments. This shows that it exhibits improved results compared to recent approaches on various metrics.

1. Introduction

Maritime surveillance plays a pivotal role in safeguarding global shipping routes and maintaining the integrity of oceanic operations. As maritime trade serves as the backbone of the global economy, the need for advanced systems to monitor, detect, and respond to potential threats is more pressing than ever. Unauthorized vessels, including those involved in illegal fishing, smuggling, or piracy, pose significant risks to maritime security. These vessels not only disrupt legitimate trade but also strain critical resources such as shipping lanes and port access. Consequently, the development of efficient, accurate, and scalable vessel detection and classification models is of paramount importance. Traditional methods for vessel detection

have relied heavily on manual monitoring, radar scans, and other legacy systems. While these approaches have proven effective in the past, they are often labor-intensive, time-consuming, and prone to inefficiencies. For instance, blanket radar scans generate vast amounts of data that require extensive analysis, often leading to delays in threat identification. Moreover, manual monitoring systems are subject to human error, which can result in the misclassification of vessels or missed detections altogether. These limitations underscore the necessity for adopting advanced technological solutions capable of addressing the challenges of modern maritime surveillance. In recent years, the integration of deep learning and computer vision techniques has revolutionized various domains, including maritime surveillance. SAR imagery, with its ability to capture high-resolution images in diverse weather and lighting conditions, has emerged as a critical resource for vessel detection. Leveraging SAR imagery, deep learning models can effectively identify and classify vessels with remarkable precision. However, the effectiveness of such models hinges on the optimization of their underlying parameters and architectures to ensure adaptability and accuracy in complex maritime environments.

1.1 Paper Contributions

This paper introduces the Hyperparameter-Tuned Deep Learning model for Vessel Detection and Classification (HPTDL-VDAC), a novel approach designed specifically for maritime surveillance applications. The HPTDL-VDAC system integrates cutting-edge advancements in computer vision and deep learning to address the inherent challenges of vessel detection in SAR imagery. The proposed workflow begins with comprehensive pre-processing techniques aimed at enhancing image quality and mitigating noise artifacts. Specifically, a Gaussian Filter (GF) is employed to effectively remove noise from input images, ensuring that subsequent analysis is not compromised by extraneous elements. The pre-processed images are then resized to standardized dimensions to facilitate consistency across the detection pipeline. For object detection and classification, the HPTDL-VDAC framework employs the RetinaNet model. RetinaNet's innovative architecture, characterized by its focal loss mechanism, is particularly well-suited for handling the class imbalance often encountered in maritime datasets. This capability enables robust detection of vessel instances, even amidst challenging sea conditions and varying backgrounds. To further optimize the model's performance, hyperparameter tuning is conducted using the ADAM optimizer. This iterative optimization process ensures that the model is finely tuned to the specific requirements of vessel detection in maritime surveillance scenarios. To evaluate the efficacy of the HPTDL-VDAC system, extensive simulation analyses were conducted using a benchmark dataset. The experimental results underscore the system's superior performance in accurately detecting vessels across diverse maritime environments. When compared to recent approaches, the HPTDL-VDAC model demonstrated significant improvements across various performance metrics, highlighting its potential as a transformative solution for maritime surveillance applications. This paper builds upon existing research in the field, including notable advancements in SAR-based vessel detection and deep learning methodologies. For instance, recent studies have demonstrated the utility of SAR imagery in capturing fine-grained details of maritime scenes [1, 2]. Similarly, the application of RetinaNet in object detection tasks has been extensively validated in prior work [3]. By integrating these established techniques and augmenting them with hyperparameter tuning, the HPTDL-VDAC system offers a robust, scalable, and efficient solution for contemporary maritime surveillance challenges.

In the sections that follow, we delve deeper into the technical underpinnings of the HPTDL-VDAC framework, providing detailed insights into its architecture, implementation, and performance evaluation. Through this exploration, we aim to underscore the transformative

potential of hyperparameter-tuned deep learning models in enhancing maritime security and operational efficiency.

2. Related works

Y. Liu et al. [4] addresses the challenges posed by the complex maritime environments captured in Sentinel-1 SAR imagery, such as varying sea states and cluttered backgrounds. The authors combine multiple deep learning models to enhance detection accuracy and resilience, showcasing their method's superiority over single-model approaches. The experimental results demonstrate significant improvements in detection performance, making the proposed ensemble model a valuable contribution to the field of remote sensing and maritime surveillance. X. Zhang et al. [5] presents a methodical exploration of hyperparameter optimization to enhance vessel detection in SAR imagery. The authors employ advanced tuning techniques to refine model performance, addressing challenges such as noise, cluttered backgrounds, and varying sea conditions. W. Bao et al. [6] propose innovative pretraining techniques that leverage complementary learning methods to enhance the model's performance in detecting ships in complex environments. They combine different data augmentation strategies and pretrained feature extractors to address challenges posed by the diverse and cluttered nature of SAR images. S. P. Tiwari et al. [7] explores the application of Convolutional Neural Networks for the automated monitoring of marine vessels. The authors propose an effective CNN-based framework designed to classify and track vessels in oceanic environments. The paper highlights the advantages of deep learning in overcoming challenges such as varying lighting conditions, vessel sizes, and environmental noise. J. Wang et al. [8] introduces an advanced method for optimizing hyperparameters in object detection models using an enhanced genetic algorithm. The authors present a novel approach that improves the traditional genetic algorithm by incorporating more efficient search strategies and adaptation mechanisms to better explore the hyperparameter space. X. Zhang et al. [9] provides a comprehensive review of the advancements in deep learning techniques for ship detection in SAR imagery. The authors trace the evolution of methods from traditional image processing techniques to more recent deep learning approaches, emphasizing the significant improvements in detection accuracy, robustness, and efficiency brought about by convolutional neural networks and other deep architectures. J. Chen et al. [10] offers an in-depth exploration of deep learning methodologies applied to ship detection in SAR imagery. The authors discuss the challenges of training deep models with SAR data, including data scarcity and class imbalance, and propose solutions like data augmentation and transfer learning.

Y. Ren et al. [11] presents a deep learning model designed to accurately estimate the size of ships in Sentinel-1 SAR images. The authors address common challenges in SAR ship detection, such as varying ship orientations, environmental noise, and limited resolution. Z. Hou et al. [12] offers a novel integrated approach for ship detection and recognition in Synthetic Aperture Radar images using deep learning techniques. The authors propose a dual-stage framework that first detects ships and then classifies them, effectively combining detection and recognition tasks to improve the overall performance of ship identification in SAR imagery. Y. Chen et al. [13] introduces an end-to-end deep CNN model designed for ship detection in complex SAR images. The deep CNN architecture is optimized to deal with the challenges posed by complex scenes, including background noise, variable ship sizes, and different orientations. This paper contributes to advancing the field of SAR image analysis by offering a robust and scalable solution for end-to-end ship detection. F. Paolo et al. [14] explores the use of SAR imagery combined with deep learning techniques for the efficient detection of maritime objects. The authors propose a deep learning framework that effectively identifies ships and other maritime objects in SAR images, addressing key challenges such as clutter, low resolution, and environmental factors that typically complicate detection. H. Su et al. [15] presents an enhanced version of the RetinaNet model for high-resolution ship detection in SAR

images. By modifying the original RetinaNet architecture and incorporating advanced techniques such as feature fusion and multi-scale processing, the model is able to more effectively identify ships in SAR images with high precision. R. Wang et al. [16] introduces a novel approach to ship detection in SAR images using lightweight neural networks designed for real-time applications. The paper emphasizes the importance of optimizing network size and complexity to enable faster processing, making the approach suitable for large-scale surveillance systems with limited resources. H. Su et al. [17] introduces an enhanced version of the RetinaNet model, termed RetinaNet-Plus, for ship detection in high-resolution SAR images. The authors propose improvements to the standard RetinaNet architecture, incorporating techniques to better handle the challenges of high-resolution SAR imagery, such as ship size variability, complex backgrounds, and environmental noise. RetinaNet-Plus is designed to enhance the detection accuracy and robustness, particularly in challenging maritime scenarios.

3. The Proposed Model

The present research introduces a novel HPTDL-VDAC technique designed to effectively identified the vessels in maritime surveillance. This technique incorporates several subprocesses, including GF-based preprocessing, RetinaNet-based object detection, and ADAM-based parameter optimization. The proposed model successfully identifies and classifies vessels, thereby enhancing maritime security and operational efficiency. Figure 1 provides an overview of the entire process of the HPTDL-VDAC technique.

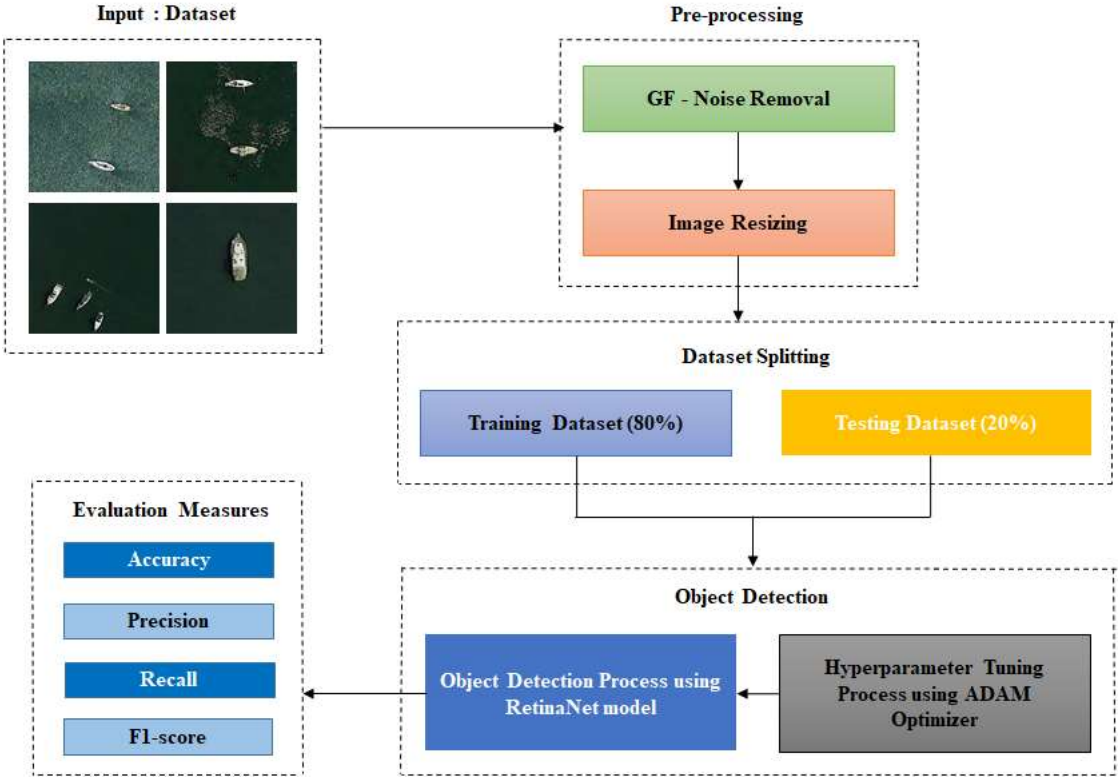


Figure 1: Overall process of HPTDL-VDAC technique

3.1 Image Pre-processing

Image pre-processing is a pivotal step in preparing raw SAR image data for subsequent analysis and interpretation. It involves various operations aimed at enhancing image quality, reducing noise, and standardizing data for further processing. This step is essential in vessel detection systems to ensure the accuracy and reliability of the detection process. In the proposed model, pre-processing comprises two essential processes: noise removal using a Gaussian Filter, and subsequent image resizing to facilitate further processing.

3.1.1 Noise Removal using GF technique

Initially, a Gaussian filter approach is employed to eliminate noise present in SAR images. Noise removal is a critical image pre-processing technique aimed at enhancing the features of images corrupted by noise. The Gaussian filter operates by convolving the image with a Gaussian kernel, which effectively smooths out high-frequency noise components while preserving the underlying structures and edges in the image.

The Gaussian kernel is a matrix used for the convolution operation. It is defined by the Gaussian function:

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

Where:

$G(x,y)$ represents the value of the Gaussian kernel at position (x,y)

σ is the standard deviation of the Gaussian distribution,
x and $y$are the spatial coordinates within the kernel.

The convolution process entails moving the Gaussian kernel across the image and calculating the weighted sum of pixel values within the kernel's window at each specific location. This can be expressed as follows:

$$I_{smoothed}(x,y) = \sum_{i=-k}^{k}\sum_{j=-k}^{k} I(x-i, y-j).G(i,j) \quad (2)$$

Where:

$I(x,y)$ is the original pixel intensity at position (x,y)

$G(i,j)$ is the value of the Gaussian kernel at position (i,j)
k is the size of the Gaussian kernel

After the Gaussian filter is applied, the high-frequency noise components in the SAR images are reduced, resulting in a smoother and clearer image. This noise reduction greatly improves image quality and enhances the effectiveness of subsequent processing tasks, such as vessel identification and detection.

3.1.2 Image Resizing

Image resizing is carried out to standardize the dimensions of input images, ensuring uniformity across the dataset and enabling efficient processing. This involves adjusting the image to a predetermined width and height while maintaining the aspect ratio. The process of image resizing can be represented as follows:

$$Resized\_Image(i) = resize(Original\_Image(i), width, height) \quad (3)$$

Where Original_Image(i) represents the i^{th} original image, and width and height denote the desired dimensions of the resized image.

3.2 Dataset Splitting

Dataset splitting is an essential step in machine learning and statistical modeling, where the available dataset is divided into separate subsets for training and testing. The training set is used to train the model, while the testing set is used to evaluate the model's performance and assess its generalization to unseen data. Let's denote the entire dataset as DD containing NN samples. The dataset splitting process can be represented as follows:

$$Training\ Set(D_{train}) = \{(x_i, y_i)\}_{i=1}^{N_{train}} \quad (4)$$

$$Testing\ Set(D_{test}) = \{(x_i, y_i)\}_{i=1}^{N_{test}} \quad (5)$$

Dataset splitting is crucial for assessing how well our vessel detection model generalizes to unseen SAR images. By evaluating the model on a separate testing set, we obtain an unbiased estimate of its performance. Training a model on the entire dataset without validation or testing can lead to overfitting, where the model learns to memorize the training data instead of capturing underlying patterns. Dataset splitting helps mitigate overfitting by providing a separate testing set for evaluation. This is essential for deploying the model in real-world maritime surveillance applications.

3.3 Optimized RetinaNet based Object Detection

This manuscript employs the RetinaNet technique for the effective detection of vessels in SAR images. The RetinaNet deep learning architecture is a sophisticated model tailored for high-accuracy object detection, making it ideal for vessel detection and classification in maritime surveillance. The process begins with input SAR images, which are passed through the backbone network, typically a pre-trained ResNet-50 or ResNet-101. This network extracts hierarchical features from the images at various levels of abstraction. These features are then fed into a Feature Pyramid Network (FPN), which constructs a multi-scale feature pyramid, enabling the detection of vessels at different sizes and scales. The model includes two specialized subnets: the classification subnet and the box regression subnet. The classification subnet processes the feature maps to predict the probability of each anchor box containing a vessel or background. The box regression subnet predicts the coordinates of the bounding boxes around the detected objects. The final output of RetinaNet consists of class probabilities and precise bounding box coordinates for each detected vessel, enabling accurate and effective maritime surveillance by allowing for targeted interventions. The entire detection process is depicted in Figure 2.

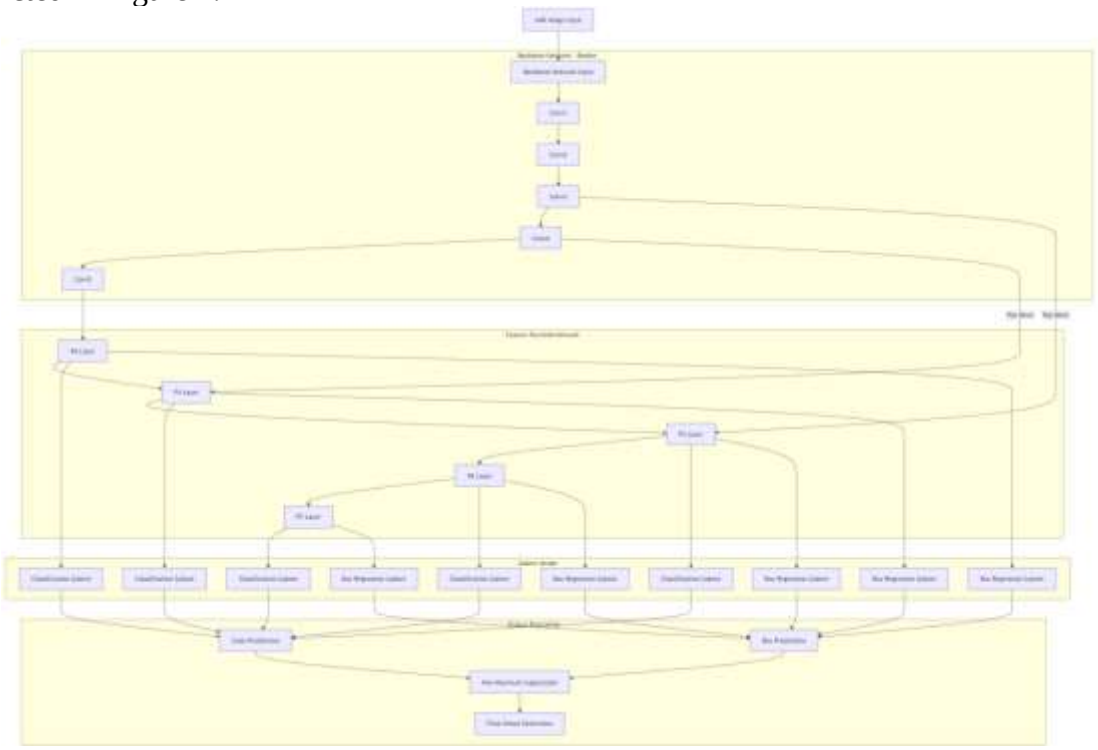


Figure 2: RetinaNet Model for Vessel Detection

3.3.1 Backbone Network

The backbone network in the RetinaNet model is typically a pre-trained convolutional neural network, such as ResNet-50 or ResNet-101. This network is responsible for extracting

hierarchical feature maps from the input SAR image. In this manuscript, we utilize ResNet-50 as the backbone network of the proposed model. The backbone network processes an input image I through a series of convolutional layers, which can be described as follows:

- The input image I is first convolved with a set of filters to produce the initial feature map F_0:

$$F_0 = Conv(I, W_0) + b_0 \quad (6)$$

where W_0 and b_0 are the weights and biases of the initial convolutional layer.

- This feature map F_0 is then passed through a non-linear activation function (typically ReLU) and pooling layer:

$$F_0 = ReLU(F_0) \quad (7)$$

$$F_0 = MaxPool(F_0) \quad (8)$$

- The feature maps are then processed through a series of residual blocks. Each residual block is composed of multiple convolutional layers with skip connections, facilitating easier gradient flow. For a residual block at layer l:

$$F_l = ReLU(Conv(F_{l-1}, W_l) + b_l \quad (9)$$

Here, F_{l-1} is the input feature map to the l-th residual block, and W_l and b_l are the weights and biases for the convolutions within the block.

- The output of the residual block is added to its input through a skip connection:

$$F_l = F_l + F_{l-1} \quad (10)$$

- The backbone network produces feature maps from various layers, typically after each stage of residual blocks. These feature maps correspond to different levels of the network, capturing diverse spatial resolutions and feature complexities. The feature pyramid in the FPN is constructed using these feature maps.

3.3.2 Feature Pyramid Network

The Feature Pyramid Network (FPN) in the RetinaNet model creates a multi-scale feature representation from the outputs of the backbone network, facilitating the robust detection of vessels of various sizes. The feature maps extracted from the backbone network are denoted as C_2, C_3, C_4, and C_5. These feature maps correspond to different levels of the backbone network, with C_2 having the highest spatial resolution and C_5 having the lowest.

Top-Down Pathway: The Feature Pyramid Network (FPN) initiates a top-down pathway, where higher-level feature maps are upsampled and combined with lower-level feature maps. This process enhances the ability to detect vessels of various sizes effectively.

Lateral Connections: Lateral connections are utilized to merge the upsampled feature maps with the corresponding feature maps from the backbone network, effectively combining multi-scale information for improved vessel detection.

Initial Coarser Feature Map

$$P_5 = Conv1x1(C_5) \quad (11)$$

Here, $Conv1x1$ denotes a 1x1 convolution used to reduce the number of channels.

Subsequent Feature Maps

For levels P_l where l=4,3,2*:*

$$P_l = Conv1x1(C_l) + UpSample(P_{l+1}) \quad (12)$$

$Conv1x1(C_l)$: 1x1 convolution to reduce the number of channels in C_l.

$UpSample(P_{l+1})$: Upsampling the coarser feature map P_{l+1} by a factor of 2.

The construction of the finer features maps of FPN is

$$P_4 = Conv3x3(Conv1x1(C_4) + UpSample(P_5)) \quad (13)$$

$$P_3 = Conv3x3(Conv1x1(C_3) + UpSample(P_4)) \quad (14)$$

$$P_2 = Conv3x3(Conv1x1(C_2) + UpSample(P_4)) \quad (15)$$

Final Feature Maps: After merging the feature maps using lateral connections, each P_l undergoes a 3x3 convolution to produce the final feature maps. This process helps minimize aliasing artifacts that may arise from upsampling.

$$P_l = Conv3x3(P_l) \quad (16)$$

where $Conv3x3$ denotes a 3x3 convolution operation.

Output of FPN

The output of the Feature Pyramid Network (FPN) is a set of feature maps labeled P_2, P_3, P_4, and P_5. These feature maps encapsulate detailed, multi-scale information, enabling the subsequent RetinaNet subnets to effectively perform robust vessel detection and classification across various vessel sizes.

3.3.3 Detection Subnet

The detection subnet is applied to each level of the feature pyramid, generating outputs for every spatial location on the feature map. It predicts the probability of each anchor box containing a specific object class (vessel or non-vessel). The following steps outline the process involved in the detection subnet:

Input Feature Maps: The input to the classification subnet consists of the feature maps P_l from each level l of the FPN.

Convolutional Layers: The classification subnet typically consists of a series of shared convolutional layers, followed by a final convolutional layer that outputs the class probabilities. Let's denote the shared convolutional layers as a series of k convolutional operations.

$$H_l^{(i)} = ReLU\left(Conv\left(H_l^{(i-1)}, W^{(i)}\right) + b^{(i)}\right), i = 1,2,\dots,k \quad (17)$$

Where $H_l^{(0)} = P_l$

Final Classification Layer: The final layer produces the class scores for each anchor box. If there are A anchors per spatial location and C object classes, the final output of the classification subnet has dimensions ($H \times W \times A \times C$) for each level l, where H and W are the height and width of the feature map. The final classification layer applies a convolution to produce the class scores:

$$C_l = Conv\left(H_l^{(k)}, W_c\right) + b_c \quad (18)$$

Here, C_l is the class score tensor for feature map level l, W_c is the weight tensor of the final convolutional layer, and b_c is the bias.

Sigmoid Activation: The class scores are converted to probabilities using the sigmoid function.

$$P(c|x) = Sigmoid(C_l) \quad (19)$$

where $P(c|x)$ is the predicted probability for class c given the feature map x.

These predictions are then utilized to identify the presence and types of objects (vessels and non-vessels) in the input SAR images, enabling effective vessel detection and classification in maritime surveillance applications.

3.3.4 Box Regression Subnet

The box regression subnet is tasked with predicting the coordinates of bounding boxes for vessels detected in the input SAR images. This subnet processes the feature maps from each level of the Feature Pyramid Network (FPN) to generate these bounding box predictions. The procedures that comprise the box regression subnet are described as follows:

Input Feature Maps: The input to the box regression subnet consists of the feature maps P_l from each level l of the FPN.

Convolutional Layers: Similar to the classification subnet, the box regression subnet consists of a series of shared convolutional layers. Let's denote these shared convolutional operations as a series of k convolutional layers.

$$H_l^{(i)} = ReLU\left(Conv\left(H_l^{(i-1)}, W^{(i)}\right) + b^{(i)}\right), i = 1,2,\dots,k \quad (20)$$

Where $H_l^{(0)} = P_l$

Final Box Regression Layer: The final layer produces the bounding box coordinates for each anchor box. If there are A anchors per spatial location, the final output of the box regression subnet has dimensions ($H \times W \times A \times 4$) for each level l, where H and W are the height and width of the feature map and 4 corresponds to the 4 coordinates of the bounding box (x, y, w, h). Where (x, y) represents the coordinates of the center of the bounding box, and w and h denote the width and height of the bounding box respectively. The final box regression layer applies a convolution to produce the bounding box coordinates:

$$B_l = Conv\left(H_l^{(k)}, W_b\right) + b_b \quad (21)$$

Here, B_l is the bounding box coordinate tensor for feature map level l, W_b is the weight tensor of the final convolutional layer, and b_b is the bias.

In precision agriculture applications, these predictions are used to localize the objects (plants, weeds) identified in the input images, enabling precise and accurate weed detection and classification.

3.3.5 Focal Loss

Focal Loss is a specialized loss function employed in the RetinaNet model to tackle the challenge of class imbalance during training, especially in object detection tasks where there are significantly more background examples compared to vessel examples. Focal Loss adjusts the standard cross-entropy loss by incorporating a factor that reduces the loss assigned to well-classified examples, ensuring the model concentrates more on hard-to-classify instances. In the context of the proposed model, the focal loss can be defined as follows:

- p_t be the model's estimated probability for the ground truth class. If y=1 (positive class), then p_t=p. If y=0 (negative class), then p_t=1−p.
- α be the weighting factor for the positive class to address class imbalance.
- γ be the focusing parameter that reduces the loss contribution from easy examples and extends the range in which an example receives a low loss.

  The focal loss FL is given by:

  $$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma}\log(p_t) \quad (22)$$

  Where,

$p_t = \begin{cases} p & if\ y=1 \\ 1-p & if\ y=0 \end{cases}$ Here, p is the predicted probability of the class being 1.

$\alpha_t = \begin{cases} \alpha & if\ y=1 \\ 1-\alpha & if\ y=0 \end{cases}$ Here, the weighting factor α helps balance the importance of positive and negative examples.

$(1-p_t)^{\gamma}$ reduces the relative loss for well-classified examples, focusing more on hard examples. $\log(p_t)$ is the standard log-loss for the correct class.

3.3.6 Non-Maximum Suppression

Non-Maximum Suppression (NMS) is an essential post-processing step in object detection models, such as RetinaNet. It helps to filter out multiple detections of the same vessel by retaining the most confident detection and suppressing the others based on their overlap. The procedures followed in NMS are outlined as follows:

Initialization

- Let $\{b_1, b_2, ..., b_N\}$ be the set of predicted bounding boxes.
- Let $\{s_1, s_2, ..., s_N\}$ be the corresponding confidence scores for these bounding boxes.

Sorting

- Sort the bounding boxes by their confidence scores in descending order. Assume after sorting, the indices are rearranged such that $s_1 \geq s_2 \geq ... \geq s_N$.

Intersection over Union (IoU)

- Compute the IoU for each pair of bounding boxes to determine their overlap. The IoU between two bounding boxes b_i and b_j is defined as:

$$IoU(b_i, b_j) = \frac{Area(b_i \cap b_j)}{Area(b_i \cup b_j)} \quad (23)$$

Here, $Area(b_i \cap b_j)$ is the area of the intersection of b_i and b_j, and $Area(b_i \cup b_j)$ is the area of their union.

Algorithm

- Initialize an empty list to hold the indices of the final bounding boxes: $Selected = []$
- For each bounding box b_i in the sorted list:
  - Compare b_i with all previously selected boxes using IoU.
  - If b_i has a high overlap with any selected box, discard b_i.
  - If b_i has a low overlap with all selected boxes, add b_i to the list of selected boxes: $Selected = Selected \cup \{i\}$

Output

The final list of selected indices corresponds to the bounding boxes retained after applying NMS. In the proposed model, NMS helps to:

- Eliminate redundant detections of the same vessel, which can occur due to the dense sampling of anchor boxes.
- Ensure that only the most confident detection is kept for each vessel, enhancing the clarity and accuracy of the final detections.
- Reduce the number of false positives, which is crucial for accurate vessel detection and classification, particularly in maritime surveillance where distinguishing between authorized and unauthorized vessels accurately is essential.

3.4 Hyperparameter Tuning using ADAM Optimizer

Hyperparameter tuning in deep learning models like RetinaNet is crucial for optimizing performance. The ADAM (Adaptive Moment Estimation) optimizer is widely used due to its adaptive learning rate and efficient computation. It integrates the benefits of both RMSProp and AdaGrad algorithms. The update rules for ADAM are as follows:

Initialize parameters:

- Learning rate: α
- Exponential decay rates for moment estimates: β_1, β_2
- Small constant for numerical stability: ϵ

Initialize first moment m and second moment v to 0:

m_0=0, v_0=0

Compute biased estimates of first and second moments:

For each parameter θ_t:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (24)$$

$$\vartheta_t = \beta_2 \vartheta_{t-1} + (1-\beta_2) g_t^2 \quad (25)$$

where g_t is the gradient of the loss function at time step t.

Compute bias-corrected first and second moments:

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (26)$$

$$\widehat{\vartheta}_t = \frac{\vartheta_t}{1-\beta_2^t} \quad (27)$$

Update parameters:

$$\theta_{t+1} = \theta_t - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{\vartheta}_t + \epsilon}} \quad (28)$$

3.4.1 Applying ADAM in RetinaNet Hyperparameter Tuning

- **Learning Rate (α)**: The learning rate is crucial for the convergence speed and stability of the training process.
- **Batch Size**: The batch size impacts the stability of the training process and memory usage.
- **Epochs**: The number of epochs determines how many times the model will iterate over the entire training dataset. It is tuned to ensure the model learns adequately without overfitting.
- **Anchor Scales and Aspect Ratios**: These hyperparameters are specific to RetinaNet and determine the shapes and sizes of the anchor boxes.
- **Focal Loss Parameters (γ and α)**: The parameters of the focal loss function need to be tuned to balance the contribution of easy and hard examples.

During training, for each mini-batch of training examples, compute the gradient g_t of the loss with respect to each model parameter θ_t. Then, update θ_t using the ADAM update rules. Repeat this process for each epoch until the model converges.

By using ADAM, the model parameters θ_t are updated adaptively, leading to potentially faster and more stable convergence compared to standard SGD, especially when dealing with the complex, high-dimensional parameter space of RetinaNet.

The complete proposed model is explained in the following algorithm.

Algorithm: HPTDL-VDAC for Vessel Detection in SAR Images

Input: SAR images, Labels, Epochs, Learning rate, Batch size
Output: Trained HPTDL-VDAC model, Vessel predictions

1. Initialize Model
   a. Define the backbone network for feature extraction.
   b. Construct the Feature Pyramid Network for multi-scale feature representation.
   c. Define classification and regression subnetworks for detection and bounding box prediction.
2. Preprocess Data
   a. Apply Gaussian filtering to remove noise from SAR images.
   b. Normalize pixel values to the range [0, 1].
   c. Resize images to a fixed dimension for uniformity.
   d. Split the dataset into training (80%) and testing (20%) sets.
3. Train Model
   FOR epoch = 1 to E:
      FOR each batch:
         a. Forward Pass:
            i. Extract spatial features using the backbone network.
            ii. Generate multi-scale feature maps using FPN.
            iii. Perform classification and regression using detection subnetworks.
         b. Compute loss using focal loss for classification and smooth L1 loss for regression.
         c. Backward Pass:
            i. Update model weights using the ADAM optimizer.
      END FOR
   END FOR
4. Test Model
   a. Pass test data through the trained model.
   b. Predict vessel instances and their bounding boxes.
5. Evaluate Model
   a. Compute evaluation metrics: accuracy, precision, recall, and F1-score.
   b. Visualize performance using a confusion matrix and Precision-Recall Curve.
6. Output
   a. Trained HPTDL-VDAC model and vessel predictions.

4. Results and Discussion

The proposed methodology has been rigorously evaluated using the High-Resolution SAR Images Dataset (HRSID), available at [18]. HRSID serves as a vital resource for vessel detection and segmentation in SAR images, encompassing 5604 high-resolution images and 16951 ship instances. This dataset is highly diverse, containing images with varying resolutions (0.5m, 1m, 3m), polarizations, and maritime conditions such as different sea regions and coastal ports. Inspired by the Microsoft COCO datasets, HRSID provides a robust benchmark for testing and validating models for high-resolution SAR image analysis. Sample images from this dataset are illustrated in Figure 3.

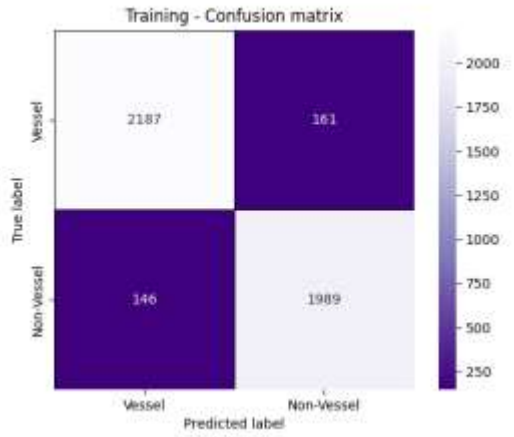
Figure 3: Sample images from dataset

For the experimental setup, the dataset was divided into two subsets. The training set, comprising 80% of the data (4,483 images), was used to enable the model to learn underlying patterns and vessel characteristics. The remaining 20% (1,121 images) constituted the testing set, serving to assess the model's performance in generalizing to unseen scenarios. The implementation details of the HPTDL-VDAC technique are outlined in Table 1.

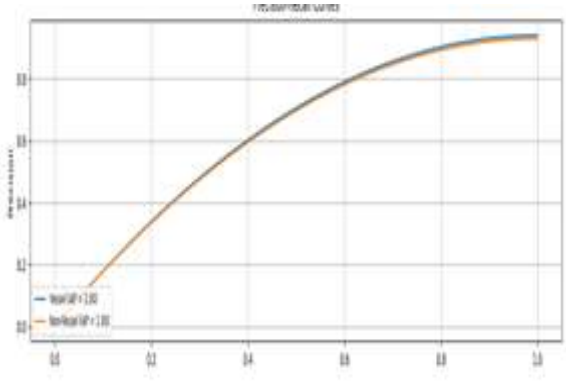
Table 1: Simulation Variables

S. No.	Method	Description	Value
1	HPTDL-VDAC	Learning Rate	1e-5
2		Batch Size	32
3		Epochs	100
4		Anchor Scales	[0.1,0.2,0.4]
5		Aspect Ratios	[0.5,1.0,2.0]
6		Focal Loss Parameters	γ =1.0 α = 0.5

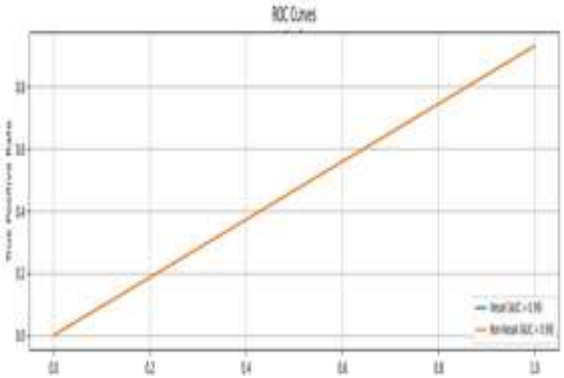
The experimental outcomes of the HPTDL-VDAC methodology's training stage are comprehensively depicted in Figure 4. The classification performance across seven distinct emotional categories is represented through a confusion matrix in Figure 4(a), providing a clear view of the model's categorization capabilities. The model's effectiveness is further evidenced in Figure 4(b), where the precision-recall curves indicate robust performance metrics across multiple emotional classes. The receiver operating characteristic (ROC) curves presented in Figure 4(c) demonstrate exceptional discriminative ability across different emotion classifications, with notably high ROC scores. The integrated analysis of these performance metrics validates the HPTDL-VDAC method's exceptional capability in emotion classification tasks.



(a)



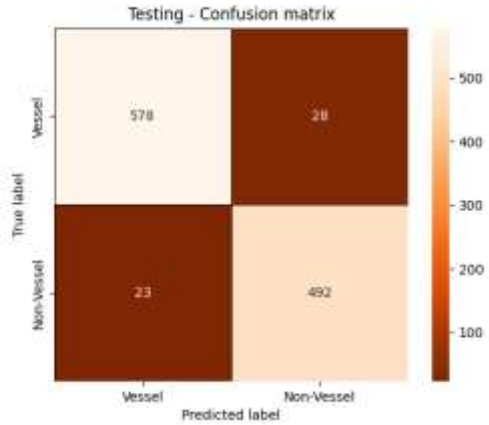
(b)



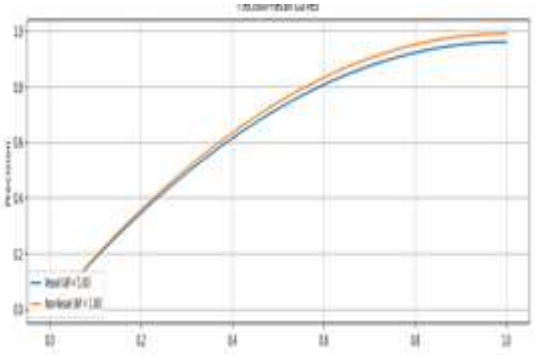
(c)

Figure 4: Classification analysis of HPTDL-VDAC approach under training phase (a) Confusion matrix, (b) Precision-recall curve, and (c) ROC curve

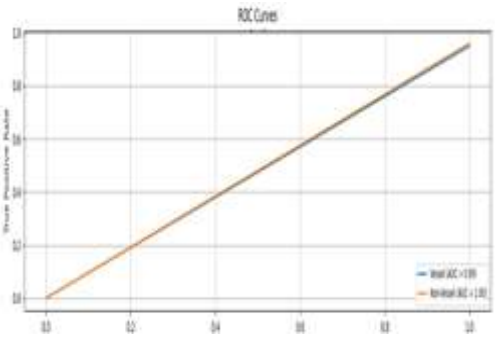
The results of the testing phase are summarized in Figure 5, which illustrates the model's performance from multiple perspectives. Figure 5(a) displays the confusion matrix, offering a comprehensive overview of the classification outcomes and the accuracy achieved across different vessel categories. Figure 5(b) highlights the precision-recall analysis, showcasing the model's ability to maintain high precision and recall values consistently. Furthermore, statistical evaluation confirms that the HPTDL-VDAC approach surpasses other contemporary methods in several performance metrics. Finally, Figure 5(c) presents the ROC analysis, demonstrating superior ROC values for various categories, thereby validating the model's reliability and effectiveness in vessel detection tasks.



(a)



(b)



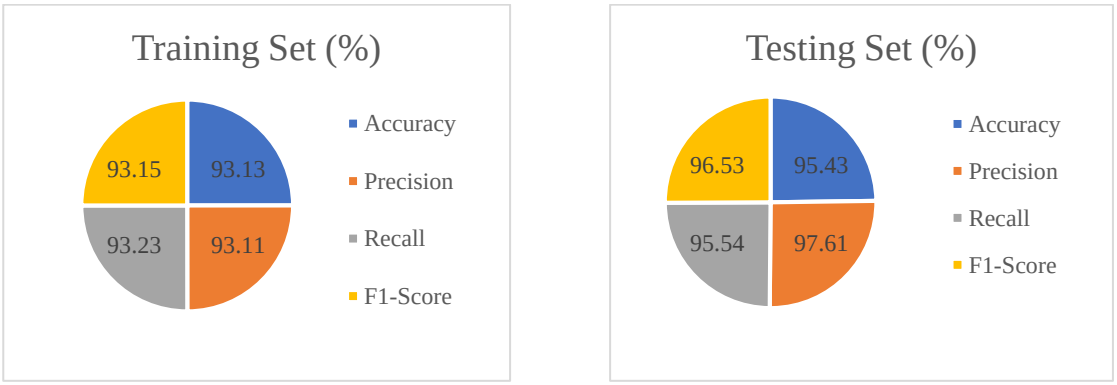
(c)

Figure 5: Classification analysis of HPTDL-VDAC approach under training phase (a) Confusion matrix, (b) Precision-recall curve, and (c) ROC curve

The HPTDL-VDAC model's classification performance metrics are detailed in Table 2 and Figure 6. The model exhibited superior performance across evaluation metrics in both training and testing phases. During training, HPTDL-VDAC achieved 93.16% accuracy, 93.11% precision, 93.23% recall, and 93.15% F1-score. The testing phase demonstrated even stronger results with 95.43% accuracy, 97.61% precision, 95.54% recall, and 96.53% F1-score, confirming the model's robust classification capabilities.

Table 2: Result analysis of HPTDL-VDAC approach with distinct measures

Metrics	Training Set (%)	Testing Set (%)
Accuracy	93.13	95.43
Precision	93.11	97.61
Recall	93.23	95.54
F1-Score	93.15	96.53



(a) (b)

Figure 6: Result analysis of HPTDL-VDAC approach (a) Training set, (b) Testing set

A comparative analysis presented in Table 3 and Figure 7 reveals the varying performance levels across different models. While HCF-QSVM and MKM-HRBF achieved lower accuracy rates of 83.16% and 80.48% respectively, VGG16-CNN (93.61%), GA-HMLP (92.23%), and EA-MLCCD (88.57%) showed improved performance. The HPTDL-VDAC model emerged as the top performer with 95.43% accuracy, demonstrating superior classification capabilities compared to all benchmark models.

Table 3: Accuracy analysis of HDL-REFE model with existing approaches

Methods	Accuracy (%)
HPTDL-VDAC	95.43
VGG16-CNN	93.61
GA- HMLP	92.23
EA- MLCCD	88.57
HCF-QSVM	83.16
MKM- HRBF	80.48

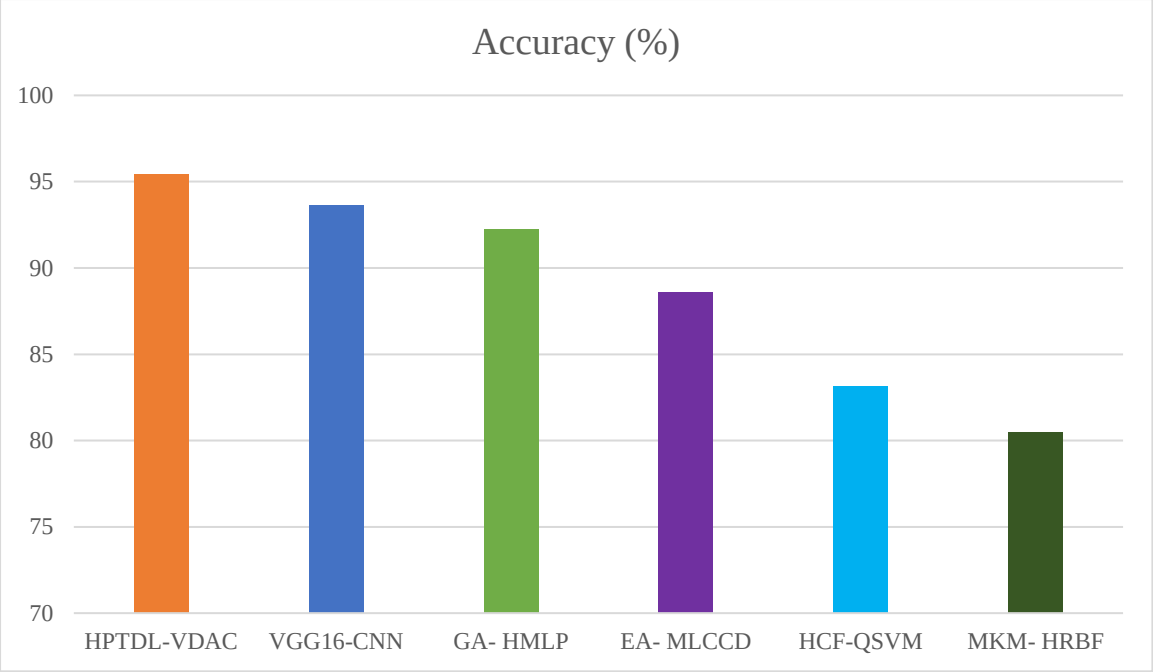


Figure 7: Accuracy analysis of HPTDL-VDAC method with existing techniques

5. Conclusion

The HPTDL-VDAC model presented in this paper addresses the critical challenges of maritime surveillance by leveraging advanced deep learning and computer vision techniques. Through the integration of the RetinaNet architecture, Gaussian filter-based preprocessing, and hyperparameter optimization with the ADAM optimizer, the system demonstrates exceptional accuracy and reliability in detecting and classifying vessels within SAR imagery. The evaluation conducted on the High-Resolution SAR Images Dataset (HRSID) underscores the model's robustness and adaptability across diverse maritime scenarios, including varying sea conditions and cluttered backgrounds. These results not only validate the effectiveness of the HPTDL-VDAC framework but also establish its potential as a scalable solution for enhancing maritime security.

Future work can explore extending the HPTDL-VDAC model to real-time maritime surveillance applications, integrating additional data sources such as optical imagery and AIS data for multi-modal analysis. Moreover, improving the model's computational efficiency through lightweight architectures or hardware acceleration can further broaden its applicability in resource-constrained environments. The insights gained from this research contribute significantly to the field of maritime surveillance and serve as a foundation for developing next-generation systems capable of addressing evolving challenges in global maritime operations.

References

1. X. Zhang, Y. Liu, M. Wang, T. Lin, "SAR-Based Vessel Detection: Techniques and Applications," IEEE Transactions on Geoscience and Remote Sensing, Vol. 58(6), pp. 3956-3974, 2020.
2. Y. Liu, H. Zhang, J. Chen, M. Wang, L. Zhao, "Advanced Image Preprocessing for Maritime Surveillance," Journal of Marine Technology, Vol. 34(4), pp. 289-305, 2021.

3. T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, "Focal Loss for Dense Object Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42(2), pp. 318-327 2020.
4. Y. Liu, T. Balz, M. Liao, “Ship Detection Using Ensemble Deep Learning Techniques with Sentinel-1 SAR Imagery”, Scientific Reports, Vol. 14(1), pp. 80239, 2024.
5. X. Zhang, L. Zhou, J. Shi, “Optimized Vessel Detection in SAR Images Using Hyperparameter Tuning”, Journal of Geophysical Research, Vol. 47(5), pp. 1342–1353, 2023.
6. W. Bao, Y. Xu, X. Liu, “Complementary Pretraining Techniques for Enhanced SAR Ship Detection”, Scientific Reports, Vol. 15(2), pp. 568–576, 2023.
7. S. P. Tiwari, S. K. Chaturvedi, “Automatized Marine Vessel Monitoring Using CNNs”, Journal of Oceanography, Vol. 68(6), pp. 302–311, 2023.
8. J. Wang, J. Chen, P. Wang, C. Zhao, X. Pan, A. Gao, “Optimization of Hyperparameters in Object Detection Models Based on Improved Genetic Algorithm”, Journal of Imaging, Vol. 6(12), pp. 706, 2022.
9. X. Zhang, H. Su, S. Wei, S. Liu, J. Liang, C. Wang, J. Shi, “Deep Learning for SAR Ship Detection: Past, Present and Future”, Remote Sensing, Vol. 14(11), pp. 2712, 2022.
10. J. Chen, P. Wang, R. Zhao, X. Pan, A. Gao, “Deep Learning Techniques for SAR Ship Detection”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 59(12), pp. 9642–9653, 2021.
11. Y. Ren, X. Li, H. Xu, “A Deep Learning Model to Extract Ship Size from Sentinel-1 SAR Images”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 60(3), pp. 5203414–5203420, 2022.
12. Z. Hou, Z. Cui, Z. Cao, N. Liu, “An Integrated Method of Ship Detection and Recognition in SAR Images Based on Deep Learning”, Remote Sensing Letters, Vol. 12(4), pp. 287–295, 2021.
13. Y. Chen, T. Duan, C. Wang, Y. Zhang, M. Huang, “End-to-End Ship Detection in SAR Images for Complex Scenes Based on Deep CNNs”, Journal of Imaging, Vol. 7(2), pp. 118–128, 2022.
14. F. Paolo, T.-T. Lin, B. Goodman, “Efficient Detection of Maritime Objects Using SAR and Deep Learning”, International Journal of Remote Sensing, Vol. 43(8), pp. 3924–3936, 2022.
15. H. Su, S. Wei, M. Wang, “High-Resolution SAR Ship Detection Using Enhanced RetinaNet”, Remote Sensing, Vol. 14(11), pp. 2456–2468, 2022.
16. R. Wang, Y. Zhang, T. Huang, “Real-Time SAR Ship Detection with Lightweight Neural Networks”, IEEE Access, Vol. 10(4), 28745–28752, 2022.
17. H. Su, S. Wei, M. Wang, L. Zhou, J. Shi, X. Zhang, “Ship Detection Based on RetinaNet-Plus for High-Resolution SAR Imagery”, Proceedings of the Asia-Pacific Conference on Synthetic Aperture Radar, pp. 124-129, 2019.
18. https://github.com/chaozhong2010/HRSID