Research and Application of Remote Sensing Image Recognition Model for Maritime Ships Based on Improved DCNN

Xiao Guo

***Abstract*—Maritime ship remote sensing image recognition holds immense importance for maritime safety management and military operations. This study enhances recognition efficiency by optimizing residual network 34 through spatial transformation networks, varied convolutional kernel sizes, and branch networks enriching image category discrimination. An advanced deep convolutional neural network model is devised, showcasing an enhancement in accuracy by 4.2% and 14.9% compared to residual network 152 and 34, respectively. The model attains a loss of 0.03 with recall rate and F1 value of 89.7% and 0.93, indicating high recognition accuracy. With an average training time of 84ms, it demonstrates superior computational efficiency. These findings emphasize the model's effectiveness in improving maritime ship remote sensing image recognition, thereby enhancing marine safety monitoring, facilitating marine resource management and development, and presenting promising applications in maritime operations.**

***Index Term*—Identification; Introducing branch networks; Maritime ships; Remote sensing images; Residual network; Spatial transformation network**

I. Introduction

As the continuous advancement of remote sensing technology, remote sensing image (RSI) recognition of maritime ships has become an important research direction in fields such as ocean monitoring and maritime traffic control. The traditional RSI recognition methods for maritime ships mainly rely on technologies such as radar and sonar. However, these technologies are often affected by factors such as weather and environment in practical applications. At the same time, traditional technologies require manual feature extraction and classifier design, which demands a lot of time and results in poor recognition performance [1-2]. In recent years, deep learning, as a new type of artificial intelligence technology, can directly learn feature representations from raw images and automatically extract local and global features from images. It has been broadly applied in the RSI recognition [3-4].

Li et al. designed a high-resolution detection method for rotating bounding boxes with convolutional neural networks (CNNs) to raise the reliability and accuracy of ship RSIs. The method utilized a dual branch regression network to independently predict variables and combined multi-level features through an adaptive pool of spatial changes. The results showed that the ship features obtained by this method were more compact [5]. Dong et al. designed a robust rotation insensitive target detection model with CNNs to reduce the uncertainty of target direction angle in ship RSIs. The model used a bounding box filtering method to enable the detector to adapt to different ground real environments. The outcomes showed that the detection effect of the model was good when the training data was limited [6]. Zhang et al. designed an RSI anchor free rotation detection technique with Gaussian mask model to reduce the issues of high computational cost and low accuracy in existing detection methods. The technique extracted multi-scale feature maps (FMs) through a U-shaped network with selective cascade modules, and modeled with the geometric features of ships. The results showed that this technique outperformed other techniques in detection [7]. Cui et al. designed an anchor free rotation ship detection model based on selective kernel network to address the issue of poor recognition performance of ship RSIs in complex scenes. They designed two customized modules, orthogonal pooling and soft rotation non-maximum suppression, to detect ships as key points. The findings indicated that the model had high computational efficiency and accuracy [8]. You et al. designed a detection algorithm based on improved regression models and feature enhancement to address the issues of semantic information confusion and boundary discontinuity in ship RSI recognition. The algorithm balanced the position and semantic information of multi-scale FMs through feature refinement networks, and used supervised learning-based attention networks to suppress background noise. The results showed that the algorithm had high robustness and accuracy [9]. Wang et al. artificially solved the problem of poor performance of traditional CNNs in ship RSI processing, and designed an image recognition algorithm based on lightweight CNNs and multi-feature cascading. It preprocessed complex remote sensing scenes through lightweight CNNs and used multi-source feature cascading decision-making to eliminate error alarms. The results showed that the algorithm was highly efficient [10]. Guo et al. designed a ship RSI recognition technology based on rotating scales and CNNs to address the issue of poor robustness in ship RSI detection. The technology extracted features through intersection and union balanced sampling and used a balanced feature pyramid. The outcomes denoted that the technology had high robustness and accuracy [11].

In summary, many scholars have utilized deep learning techniques to the recognition of ship RSIs. However, most scholars only used basic deep learning frameworks or

Manuscript received June 11, 2024; revised October 15, 2024

Xiao Guo is a Professor in Hebi Institute of Engineering and Technology, Henan Polytechnic University, Hebi, 458031, China. (Corresponding Author, e-mail: xiaoguo9246@outlook.com).

combined them with other technologies, which still has the problem of high model complexity. Therefore, based on the improved framework of Residual Network-34 (ResNet-34) in Deep Convolutional Neural Network (DCNN), this study further optimizes it using Spatial Transformer Network (STN) and convolutional kernels of different sizes, and combines branch networks to design a ship RSI recognition model based on the improved DCNN to improve recognition efficiency and accuracy. The innovation of the research lies in the optimization of ResNet-34, which enhances the model's ability to learn ship features. At the same time, combined with branch networks, the accuracy of ship image recognition is improved, providing new ideas for the processing of ship RSIs.

The article mainly contains three parts. The first part is the design of ship RSI recognition technology based on improved ResNet-34 network. Firstly, a ship RSI feature extraction model is constructed, and then a ship RSI classification method based on branch network is designed. The second part is the experiment findings of the model. The first section is the performance analysis with the raised ResNet-34 network. The second section is the actual application effect analysis. The third part is a summary of the entire article and points out the deficiencies of the research.

II. Methods and Materials

This part mainly elaborates on the construction of a maritime ship RSI recognition model with improved DCNN. Firstly, a feature extraction method for ship RSIs is designed and improved on the basis of the DCNN architecture ResNet-34. Then, a branch network is introduced to further optimize the feature extraction model.

A. Ship remote sensing image feature extraction model based on improved ResNet-34

The research first focuses on feature extraction of RSIs. Because of the small target size of ship images, the feature information extracted through traditional DCNN models is relatively single and easily affected by background information, resulting in low recognition accuracy [12]. ResNet-34 is a DCNN architecture that overcomes problems such as vanishing gradients, exploding gradients, overfitting, and degradation by introducing residual blocks and batch normalization techniques [13]. Therefore, the study uses ResNet-34 for feature extraction of ship RSIs. Firstly, the study uses an STN to eliminate interference from background information. Additionally, it extracts multi-scale information by utilizing convolution kernels of different sizes to design an improved ResNet-34 network model. STN represents a representative model of a spatial attention mechanism. It can transform various deformation data in space and automatically capture important regional characteristics. STN can ensure that images achieve comparable results to the original image despite undergoing operations such as cropping, translation, or rotation [14]. STN contains three principal components: a positioning network, a grid generator, and a sampler. The positioning network is responsible for regressing transformation parameters. By inputting feature images, it can output spatial transformation parameters through a series of hidden network layers, as illustrated in equation (1).

$$\theta = f_{loc}(U) \tag{1}$$

In equation (1), θ represents the spatial transformation parameter, $f_{loc}(\cdot)$ represents the fully connected or convolutional function, and U represents the input FM. There are various forms of spatial transformation parameters, and different affine transformations can obtain outputs of different dimensions, whose size depends on the type of transformation. A grid generator is a sampling network constructed based on the parameters predicted by the transformation model. These parameters are the output obtained by oversampling a set of points in the input image [15]. A mapping relationship can be obtained through a grid generator, which can represent the correspondence between the feature image and the output feature image through matrix operations. The expression is shown in equation (2).

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = T_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{2}$$

In equation (2), $\left(x_i^s, y_i^s\right)$ denotes the target pixel position on the output FM, T denotes affine transformation, $\left(x_i^t, y_i^t\right)$ represents the sampling position on the corresponding input FM, and A_θ represents the transformation matrix. By using θ_{11}, θ_{12}, θ_{21}, and θ_{22}, the angle of image rotation can be determined, and by using θ_{13} and θ_{23}, the translation amount of image width and height can be determined. The sampler uses the sampling grid and the input FM as inputs to generate the output and calculate the transformed FM. The calculation method is indicated in equation (3).

$$V_i^c = \sum_n^H \sum_m^W U_{n,m}^c \max(0, 1 - \left|x_i^s - m\right|) \max(0, 1 - \left|y_i^s - n\right|) \tag{3}$$

In equation (3), V_i^c indicates the pixel value of the coordinate (x_i^t, y_i^t) in the c channel of the output FM. $U_{n,m}^c$, H, and W mean the pixel values of the inner coordinate, the height, and the width in the input FM, respectively. STN does not rely on a specific network architecture and can be inserted into any CNN architecture, providing spatial transformation capabilities for different tasks and applications. Meanwhile, due to the different morphological structures of different categories of ships, the key to ship RSI classification lies in the extraction of small features. If only a single scale convolution kernel is used, the extracted feature information will be relatively single, which will affect the accuracy of ship classification. Research extracts multi-scale information by introducing parallel convolutional layers and STN concatenation. The specific structure is indicated in Figure 1.

In Figure 1, the parallel convolution module has two channels. One channel contains a 1×1 convolution kernel and a 3×3 convolution kernel, while the other channel contains a 1×1 convolution kernel and a 5×5 convolution kernel, with a batch normalization layer added after each convolution kernel. Parallel convolutional layers perform convolution operations on FMs using convolution kernels of

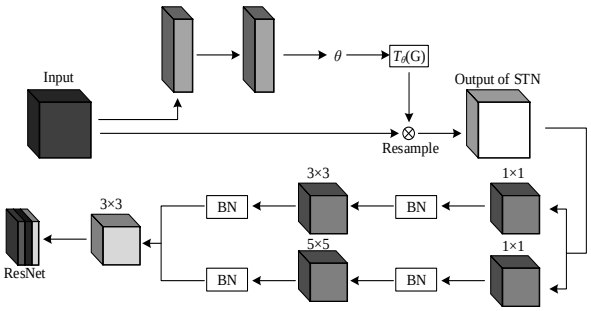


Figure 1 Parallel convolutional layers and STN concatenation structure

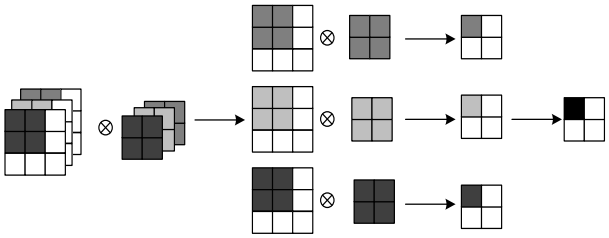


Figure 2 The calculation of row convolutional layers

different sizes, resulting in new FMs of varying sizes. These are then upsampled to the output FM size, enriching the image's features without altering the original FM size. It encodes and decodes the relevant feature information in the image from a global perspective, thereby enhancing image recognition performance. Equation (4) shows the various branch expressions of parallel convolutional layers.

$$\begin{cases} Y_1 = conv_{1\times1}(X) \\ Y_2 = conv_{3\times3}(X) \\ Y_3 = conv_{5\times5}(X) \end{cases} \tag{4}$$

In equation (4), Y_1, Y_2, and Y_3 represent the outputs of 1×1, 3×3, and 5×5 convolution kernels, respectively. $conv(\cdot)$ represents convolution operation. X represents input FM. It merges the outputs of two channels along the depth direction and outputs them through a 3×3 convolutional kernel. The calculation method for the output and its output size is shown in equation (5).

$$\begin{cases} Y = [Y_1, Y_2, Y_3] \\ O_s = \dfrac{I_s - F_s + 2\times P_d}{S_t} + 1 \end{cases} \tag{5}$$

In equation (5), Y represents the comprehensive output, O_s represents the output size, I_s represents the input size, F_s represents the convolution kernel's size, P_d represents padding, and S_t represents step size. The calculation process of parallel convolutional layers is denoted in Figure 2.

B. Classification method for ship remote sensing images based on branch networks

Although the optimized ResNet-34 network has been used to construct feature extraction methods for ship RSI recognition, ship images have features with large inter-class and small intra-class differences. If only the feature extraction method is used to distinguish ship RSIs, the effect is not very good [16-17]. Therefore, based on the improved ResNet-34 network, a base branch network and a meta embedding branch network are constructed to increase inter-class differences and reduce intra-class differences, and fused with the designed feature extraction network. In the base branch network, the first step is to construct a loss function (LF) and study the use of a central LF. The central LF can be utilized to enhance the compactness of samples belonging to the same class and to increase the dispersion of samples belonging to different classes. Furthermore, the deep features derived from this method exhibit high discriminability. Finally, the center LF's expression is represented in equation (6).

$$L_C = \frac{1}{2}\sum_{i=1}^{m}\left\| x_i - c_{y_i} \right\|_2^2 \tag{6}$$

In equation (6), L_C represents the center LF, x_i represents the feature vector of the i th sample, c_{yi} denotes the center vector of y_i categories, and m represents the batch size. Next, it updates the gradient and class center using the calculation method shown in equation (7).

$$\begin{cases} \dfrac{\partial L_C}{\partial x_i} = x_i - c_{y_i} \\ \Delta c_j = \dfrac{\sum_{i=1}^{m}\delta(y_i = j)\times(c_j - x_i)}{1 + \sum_{i=1}^{m}\delta(y_i = j)} \end{cases} \tag{7}$$

In equation (7), $\dfrac{\partial L_C}{\partial x_i}$ represents gradient, Δc_j represents class center, and $\delta(y_i = j)$ represents condition, which means when y_i is the j class. In class center updates, if the condition is met, the value of $\delta(\cdot)$ is 1, and if the condition is not met, the value of $\delta(\cdot)$ is 0. If only the central LF is used to supervise network learning, the calculated loss value may be small, leading to overfitting [18]. The study combines the central LF with the normalized exponential LF to enhance feature recognition. The total LF of the base branch network is indicated in equation (8).

$$\begin{cases} L_S = -\sum_{i=1}^{m}\log\dfrac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \\ L_b = L_S + \lambda_C L_C \end{cases} \tag{8}$$

In equation (8), L_b represents the total LF of the base branch network, L_S represents the normalized exponential LF, W_j represents the j column of the weights in the fully connected layer at the end of the network, b represents bias, d represents the feature dimension, and λ_C represents the limiting variable, whose values range from 0 to 1. Its functions are to balance the center LF and the normalized exponential LF. The larger the λ_C, the more compact the samples of the same type are, while the smaller the λ_C, the opposite is true. Due to the fact that using only base branch networks can only make samples of the same category more compact, the classification accuracy for samples of different categories is insufficient. The study further distinguishes images using meta embedding branch

networks, which integrate the features of the image itself with corresponding memory features, quantify the recognition degree of known categories using feature norms, and increase the difference between categories without being affected by class center interference, thereby enhancing the discrimination between categories [19]. Firstly, it obtains the discriminative center of the meta embedding branch network. The discriminative center can effectively capture the core features of categories, enhance inter class discrimination, and the acquisition process includes two steps: neighborhood sampling and propagation. The acquisition process is shown in Figure 3.

In Figure 3, during the neighborhood sampling stage, the network randomly selects a batch of samples during training and divides them into small batches based on their labels. These small batch features are used to update the class center of each group to ensure consistency within the class. In the propagation stage, the network optimizes the feature representation and category center through alternating updates, ensuring that the feature vectors of each sample are as close as possible to the center of the corresponding group of the sample, while staying away from the centers of other groups. In the meta embedding branch network, due to the lack of supervision ability for updating small data samples, a memory module is added to the network to connect features. The connected features are called meta embedding features, and then images are classified based on this feature. The classification process includes two steps. Firstly, it performs memory synthesis using the calculation method shown in equation (9).

$$f_m = o_T C = \sum_{i=1}^{N} o_i c_i \tag{9}$$

In equation (9), f_m represents memory features, o represents weight coefficients, and C represents class centers. Among them, the function of weight coefficients is to evaluate the weight between memory features and category centers, and then scale the memory features through a scaling coefficient to dynamically change them. The calculation method is shown in equation (10).

$$f'_m = \frac{1}{\eta} \times (f_m + f_b) \tag{10}$$

In equation (10), f'_m represents meta embedding features, f_b represents features obtained by the ResNet-34 network, and η represents scaling coefficients. The scaling factor is the distance between the features obtained by the ResNet-34 and their nearest category center, calculated as shown in equation (11).

$$\eta = R_e(f_b, M) = \min_t \left\| f_b - c_i \right\|_2 \tag{11}$$

In equation (11), R_e represents the reachability function, and M represents the class center of a certain feature. In the context of scaling factors, a relatively small value means that the input sample is known, whereas a relatively large value means that the input sample is unknown. Then, cosine similarity is used to calculate the classification results, as shown in equation (12).

$$P_M = \kappa \times \cos(f'_m, w_k) = \kappa \times \overline{f'_m} \times \overline{w_k} \tag{12}$$

In equation (12), κ represents the control parameter, $\overline{f'_m}$ represents the regularization of meta embedding features, and $\overline{w_k}$ represents the regularization of weights. The next step is to calculate the LF, and study use the sum of the maximum marginal LF and the cross entropy LF as the LF of the meta embedded branch network. The calculation method is shown in equation (13).

$$L_m = \varepsilon_m L_M(f_b, M) + L_{ce} \tag{13}$$

In equation (13), L_m represents the LF of the meta embedded branch network, L_M represents the maximum marginal LF, L_{ce} represents the cross entropy LF, and ε_m refers to the weight. The implementation of the meta

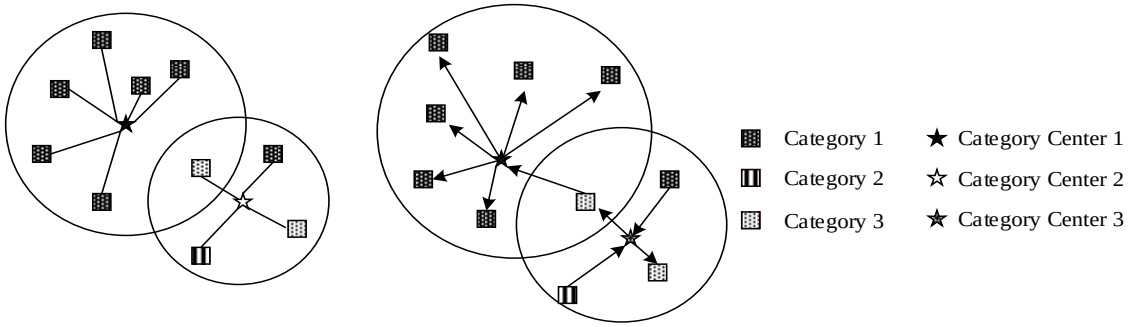


Figure 3 The process of obtaining the discrimination center

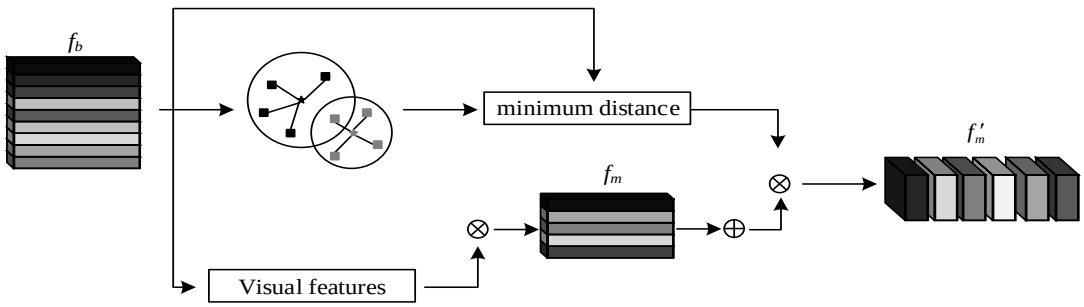


Figure 4 The Implementation process of meta embedding branch network

embedded branch network is shown in Figure 4.

Usually, image recognition involves fusing all features and directly classifying them. However, in the recognition of ship RSIs, unknown categories appear. Therefore, conventional methods have poor performance in classifying ship RSIs [20]. The study adopts an end-to-end fusion approach to combine the base branch network with the meta embedding branch network to determine the category of ship RSIs. The classification result after fusion is shown in equation (14).

$$P = \beta P_M + (1-\beta) P_B \tag{14}$$

In equation (14), P refers to the fused classification result, P_B represents the classification result of the base branch network, and β represents the calibration degree. Finally, a constraint is set for the classification results to reduce the influence of unknown categories, and the constraint expression is indicated in equation (15).

$$f = \begin{cases} \arg\max_{i\in\{1,\ldots,N\}} P_D(C_i \mid x), P \geq \xi \\ \text{unknown}, P < \xi \end{cases} \tag{15}$$

In equation (15), ξ represents the threshold used to distinguish between unknown and known categories, and unknown represents unknown. The implementation process of the designed ship image recognition model is shown in Figure 5.

III. Results

This part mainly elaborates on the experimental results based on the improved DCNN maritime ship RSI recognition model. Firstly, the effectiveness analysis of the recognition model was analyzed, and then the practical application of the recognition model was discussed, to fully assess the model.

A. Performance analysis of ship RSI recognition model

To prove the effectiveness of the designed ship RSI recognition algorithm, experiments were conducted using

TABLE I
PARAMETERS OF 7 DIFFERENT CATEGORIES OF SHIPS

Category	Ship width	Ship length	Draft
1	24.2	182.9	8.2
2	16.9	172.9	6.6
3	20.5	153.9	6.4
4	16.9	135.0	4.6
5	17.7	121.0	4.4
6	16.5	102.3	4.3
7	12.2	89.4	4.1

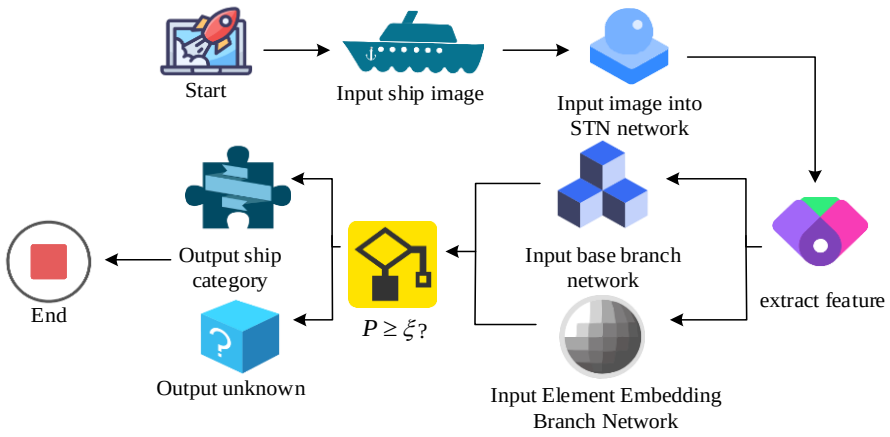


Figure 5 Implementation process of ship image recognition model

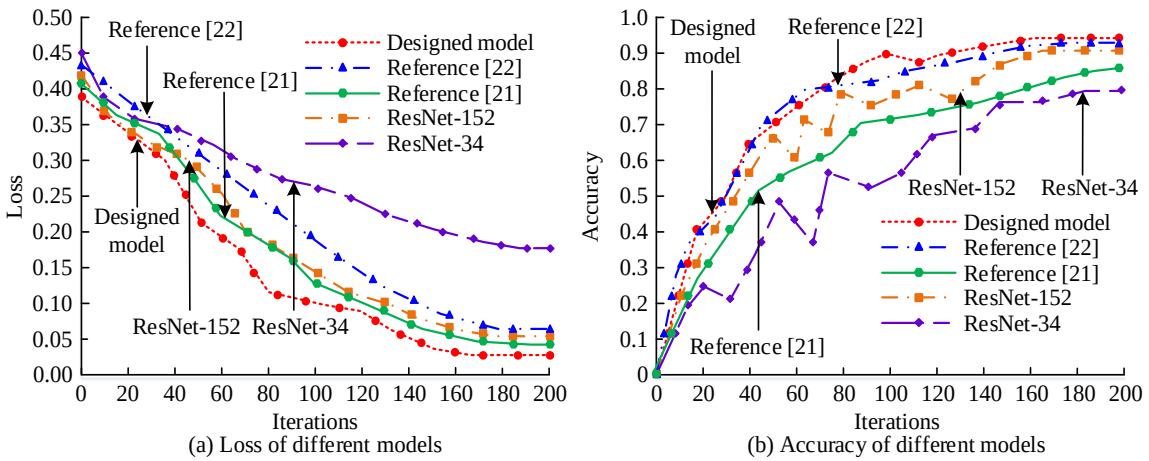


Figure 6 Loss and accuracy of different models

Python on a computer with an Intel i9-9900K central processor, an RTX 2080Ti graphics processor, and a 32GB RAM. Seven different categories of ships were selected from the HRSC2016 dataset, and the ship parameters are indicated in Table I.

Firstly, the amount of iterations was set to 200 and the batch size was set to 64. The loss and accuracy of the design model were calculated during the training, and compared with the loss and accuracy of the models in reference [21], reference [22], ResNet-152, and ResNet-34. The comparison results are shown in Figure 6.

Figures 6 (a) and (b) show the losses of different models, respectively. From Figure 6 (a), the loss curves of all three models showed a decreasing trend and gradually flatten out. At the max amount of iterations, the loss of ResNet-34 model was 0.18, the loss of ResNet-152 model was 0.06, the loss of the model in reference [21] was 0.04, the loss of the model in reference [22] was 0.07, and the loss of the designed algorithm was 0.03. Compared with the other four models, the loss value of the designed algorithm was significantly reduced, indicating that the convergence effect of the designed model was good and that it could better fit the training data. In Figure 6 (b), the accuracy curves of the design model, the model in reference [22], the model in reference [20], ResNet-152, and ResNet-34 models all showed a gradually increasing trend and gradually reached a stable state. When the max amount of iterations was reached, the accuracy of the five models was 95.3%, 92.8%, 86.7%, 91.1%, and 80.4%, respectively. Compared with the model in reference [22], the model in reference [20], ResNet-152 and ResNet-34, the designed model's accuracy increased by 2.5%, 8.6%, 4.2% and 14.9%, respectively. The above results demonstrated the effectiveness of the design model and proved its good effectiveness in remote sensing ship image recognition. This may be because the design model builds a deeper network structure, but compared to traditional networks, it has lower complexity and requires fewer parameters. The next step was to study the recognition accuracy of the designed algorithm under different signal-to-noise ratios (SNR), and compare it with ResNet-152, ResNet-34, and DCNN. The outcomes are indicated in Table II.

From Table II, when the SNR was 5dB, the recognition accuracy of the designed model was 94.23%. Compared with ResNet-152, ResNet-34, and DCNN, the accuracy of the designed algorithm was promoted by 1.26%, 4.55%, and 6.10%, respectively. When the SNR was 10dB, the recognition accuracy of the designed model was 96.62%, which was 1.84%, 5.36%, and 5.88% higher than the other three algorithms, respectively. When the SNR was 15dB, the recognition accuracy of the four models was 98.73%, 96.45%, 92.44%, and 93.16%, respectively. The design model's accuracy was greatly higher than other algorithms, further proving its superior performance. This may be due to the introduction of residual mapping in the design model, which uses skip connections for residual learning to solve problems such as gradient vanishing, thereby improving recognition accuracy. Finally, the recall and F1 values of the design model, the model in reference [22], the model in reference [21], ResNet-152, and ResNet-34 were calculated separately, as shown in Figure 7.

TABLE II
RECOGNITION ACCURACY OF DIFFERENT MODELS UNDER DIFFERENT SNRS

Model	SNR (db)		
	5	10	15
Designed model	94.23	96.62	98.73
ResNet-152	92.97	94.78	96.45
ResNet-34	89.68	91.26	92.44
DCNN	88.13	90.74	93.16

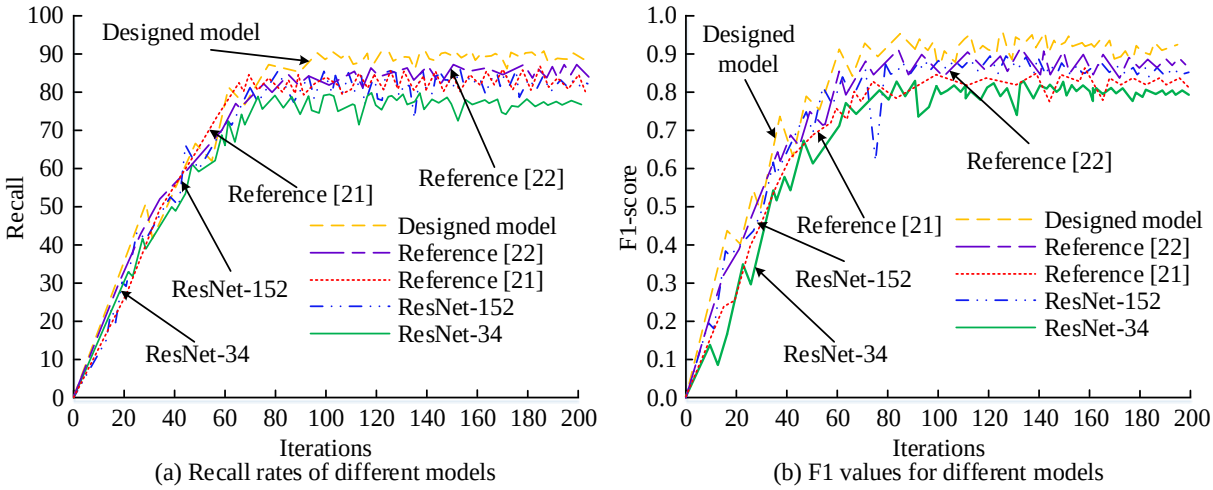
Figures 7 (a) and (b) show the recall rates and F1 values of different models. The two of the three models showed an increasing trend with the increase of iteration times and gradually flatten out. In Figure 7 (a), the recall rate of the designed algorithm was 89.7%, the recall rate of the model in reference [22] was 84.5%, the recall rate of the model in reference [21] was 80.1%, the recall rate of ResNet-152 was 83.3%, and the recall rate of ResNet-34 was 77.9%. The recall rate of the design model was higher than the other four models. In Figure 7 (b), the F1 values of the three models were 0.93, 0.88, 0.83, 0.86, and 0.80, respectively. The F1 values of the designed model were 0.5, 1.0, 0.7 and 0.13 higher than those of the other two models, respectively. This means that in the recognition of ship images, the design algorithm is significantly less affected by background information interference, indicating that the design algorithm can better recognize ship RSIs and has high reliability.

B. Analysis of the practical application effect of ship RSI recognition model

To prove the effect of the designed ship RSI recognition model in practical applications, the study first used different models to detect 7 different types of ships on the HRSC2016 dataset, and the recognition outcomes are indicated in Figure 8.

From Figure 8, in the first type of ship, the design model's recognition rate was 91.3%, which was 4.2% and 11.7% higher than the recognition rates of the other two models. In the second type of ships, the recognition rates of the three models were 90.6%, 85.2%, and 83.9%, respectively. In the third class of ships, the design model's recognition rate was 92.4%, the recognition rate of ResNet-152 was 82.7%, and the recognition rate of ResNet-34 was 81.2%. In the fourth, fifth, sixth, and seventh types, the recognition rates of the designed model were 93.0%, 89.9%, 94.2%, and 91.1%, respectively. In different types of ships, the design model's recognition rate was significantly higher than that of other models, indicating that the design model could accurately recognize RSIs of different types of ships, proving its strong generalization ability. This may be because the design algorithm extracts multi-scale information through different convolution kernels, resulting in better recognition performance on different types of images. Next step was to calculate the average accuracy and average training time of different models for RSI recognition of different types of ships, as shown in Table III.

From Table III, in the RSIs of seven different types of ships, the designed model's recognition accuracy was



(a) Recall rates of different models

(b) F1 values for different models

Figure 7 Recall rates and F1 values of different models

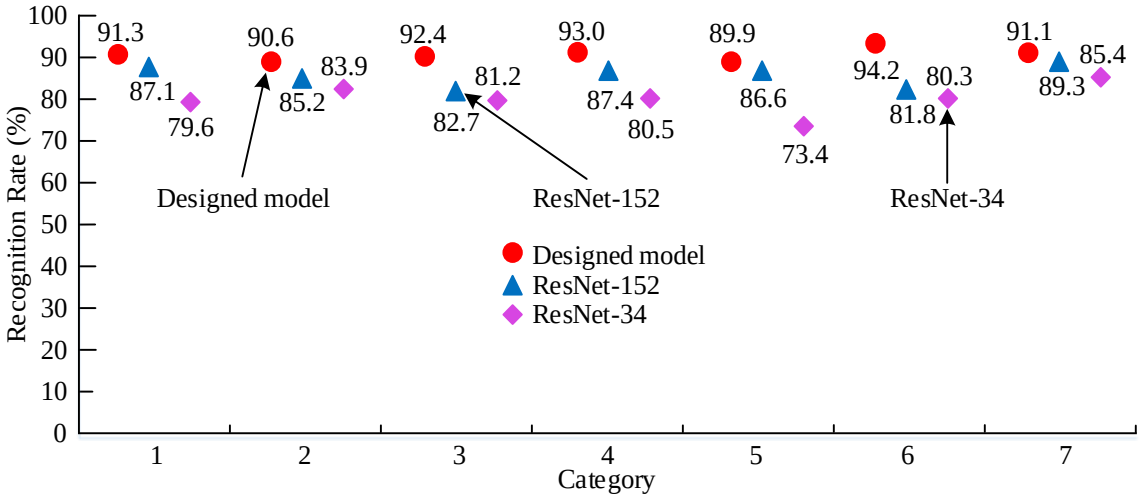


Figure 8 The recognition results of different models on different ships

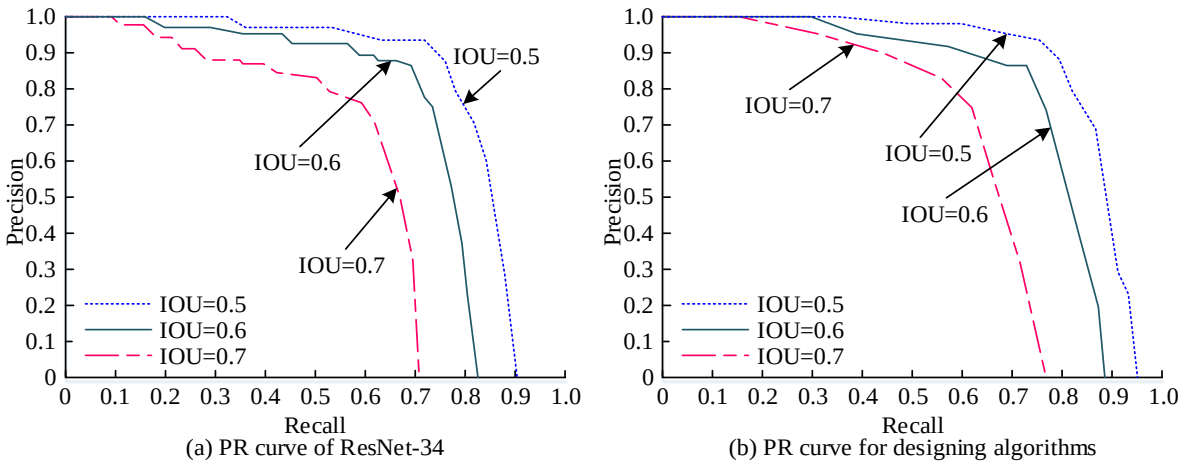
TABLE III
AVERAGE ACCURACY AND TIME OF DIFFERENT MODELS FOR RECOGNIZING SHIP IMAGES OF DIFFERENT TYPES

Model	DCNN	ResNet-34	ResNet-152	Designed model
1	69.36%	77.84%	85.31%	90.44%
2	78.75%	81.76%	85.59%	87.18%
3	68.07%	68.59%	77.81%	89.67%
4	62.08%	74.82%	75.97%	89.26%
5	63.56%	65.79%	71.53%	90.97%
6	75.92%	76.70%	84.19%	92.28%
7	78.96%	74.87%	81.38%	91.61%
Average precision (%)	70.96	74.34	80.25	90.20%
Time (ms)	105	101	86	84

90.44%, 87.18%, 89.67%, 89.26%, 90.97%, 92.28%, and 91.61%, respectively, with an average accuracy of 90.20%, greatly higher than the other three models. The design model's average training time, ResNet-152, ResNet-34, and DCNN were 84ms, 86ms, 101ms, and 1.5ms, respectively. The average training time of the design model was lower than that of other models. The above results demonstrated that the design model had high discrimination for RSIs of different ships, and also demonstrated its high computational efficiency. This is because the research combines the base branch network with the meta embedding branch network through end-to-end fusion, reducing image classification time and improving computational efficiency.

Finally, it calculated the PR curves under different Intersection Over Union (IOU) ratios and compared them with the out comes of the ResNet-34 model, as denoted in Figure 9.

Figures 9 (a) and (b) express the PR curves of the ResNet-34 model and the designed model under different IOU values, respectively. From Figure 9 (a), when the IOU values were 0.5, 0.6, and 0.7, the areas under the PR curve of the ResNet-34 model were 0.722, 0.847, and 0.906, respectively. From Figure 9 (b), the areas under the PR curve of the design model were 0.785, 0.893, and 0.958, respectively. The area under the PR curve of the designed model was higher than that of the ResNet-34 model under



(a) PR curve of ResNet-34 (b) PR curve for designing algorithms

Figure 9 PR curves of different models under different IOU values

different IOU values, indicating good overall performance of the designed model and proving its practicality.

IV. Conclusion

As the quick advancement of deep learning technology, its application in RSI recognition of maritime ships is becoming increasingly widespread. However, traditional RSI recognition methods suffer from low accuracy and efficiency in processing maritime ship images. This study proposed an improved DCNN algorithm, which first optimized the ResNet-34 model by using STN and convolutional kernels of different sizes to extract multi-scale features. Then, a branch network was introduced to enhance the discrimination between feature categories to improve recognition performance. The outcomes denoted that the designed model's recognition accuracy was as high as 95.3%, which was improved by 4.2% and 14.9% compared to the ResNet-152 and ResNet-34 models, respectively, proving its good recognition performance. When the SNR was 15dB, the designed model's recognition accuracy was 98.73%, significantly higher than other models, proving its high accuracy. In practical application analysis, the designed model's average recognition accuracy for 7 different types of ships reached 90.20%, far exceeding other models, with an average training time of 84ms, proving its high computational efficiency and further demonstrating its good performance. The above results demonstrated the effectiveness and practicality of the design model. However, the study did not explore the training time and resource consumption of the model on large-scale datasets. In the future, the model will be further optimized to reduce computational complexity and further improve its running speed.

References

[1] Wen G, Cao P, Wang H, et al. "MS-SSD: Multi-scale single shot detector for ship detection in remote sensing images," *Applied Intelligence*, vol. 53, no.2, pp1586-1604, 2023.

[2] Zhang Y M, Li R Q. "A lightweight multi-target detection method for infrared remote sensing image ships," *Journal of Network Intelligence*, vol. 8, no.2, pp535-545, 2023.

[3] Zou H, He S, Wang Y, Li R, Cheng F, Cao X. "Ship detection based on medium-low resolution remote sensing data and super-resolved feature representation," *Remote Sensing Letters*, vol. 13, no.4, pp323-333, 2022.

[4] Preethi P, Mamatha H R. "Region-based convolutional neural network for segmenting text in epigraphical images," *Artificial Intelligence and Applications*, vol. 1, no.2, pp119-127, 2023.

[5] Li L, Zhou Z, Wang B, Miao L, Zong H. "A novel CNN-based method for accurate ship detection in HR optical remote sensing images via rotated bounding box," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no.1, pp686-699, 2020.

[6] Dong Z, Lin B. "Learning a robust CNN-based rotation insensitive model for ship detection in VHR remote sensing images," *International Journal of Remote Sensing*, vol. 41, no.9, pp3614-3626, 2020.

[7] Zhang X, Wang G, Zhu P, Zhang T, Li C, Jiao L. "GRS-Det: An anchor-free rotation ship detector based on Gaussian-mask in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no.4, pp3518-3531, 2020.

[8] Cui Z, Leng J, Liu Y, Zhang T, Quan P, Zhao W. "SKNet: Detecting rotated ships as keypoints in optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no.10, pp8826-8840, 2021.

[9] You Y, Ran B, Meng G, Li Z, Liu F, Li Z. "OPD-Net: Prow detection based on feature enhancement and improved regression model in optical remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no.7, pp6121-6137, 2020.

[10] Wang N, Li B, Wei X, Wang Y, Yan H. "Ship detection in spaceborne infrared image based on lightweight CNN and multisource feature cascade decision," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no.5, pp4324-4339, 2020.

[11] Guo H, Yang X, Wang N, Song B, Gao X. "A rotational libra R-CNN method for ship detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no.8, pp5772-5781, 2020.

[12] Cui Z, Wang X, Liu N, Cao Z, Yang J. "Ship detection in large-scale SAR images via spatial shuffle-group enhance attention," *IEEE Transactions on Geoscience and Remote Sensing*, 59, no.1, pp379-391, 2020.

[13] Fu J, Sun X, Wang Z, Fu K. "An anchor-free method based on feature balancing and refinement network for multiscale ship detection in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no.2, pp1331-1344, 2020.

[14] He J, Wang Y, Liu H. "Ship classification in medium-resolution SAR images via densely connected triplet CNNs integrating Fisher discrimination regularized metric learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no.4, pp3022-3039, 2020.

[15] Wang X, Liu J, Liu X, Liu Z, Khalaf O I, Ji J, Ouyang Q. "Ship feature recognition methods for deep learning in complex marine environments," *Complex & Intelligent Systems*, vol. 8, no.5, pp3881-3897, 2022.

[16] Mokayed H, Quan T Z, Alkhaled L, Sivakumar V. "Real-time human detection and counting system using deep learning computer vision techniques," *Artificial Intelligence and Applications*, vol. 1, no.4, pp221-229, 2023.

[17] Sun X, Lin J C W. "A target recognition algorithm of multi-source remote sensing image based on visual Internet of Things," *Mobile Networks and Applications*, vol. 27, no.2, pp784-793, 2022.

[18] Kong W, Liu S, Xu M, Yasir M, Wang D, Liu W. "Lightweight algorithm for multi-scale ship detection based on high-resolution SAR images," *International Journal of Remote Sensing*, vol. 44, no.4, pp1390-1415, 2023.

[19] Almi'ani N, Anbar M, Karuppayah S, et al. "Feature Selection and 1DCNN-based DDOS Detection in Software-Defined Networking," *Engineering Letters*, vol. 32, no.7, 2024.
[20] Chen K, Wang L, Wu H, et al. "Background-Aware Correlation Filter for Object Tracking with Deep CNN Features," *Engineering Letters*, vol. 32, no.7, 2024.
[21] Yasir M, Jianhua W, Mingming X, Hui S, Zhe Z, Shanwei L, Hossain M S. "Ship detection based on deep learning using SAR imagery: a systematic literature review," *Soft Computing*, vol. 27, no.1, pp63-84, 2023.
[22] Xu Z, Sun J, Huo Y. "Ship images detection and classification based on convolutional neural network with multiple feature regions," *IET Signal Processing*, vol. 16, no.6, pp707-721, 2022.