frontiers | Frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 13 June 2022
doi: 10.3389/fnbot.2022.889308

# Recognition and Classification of Ship Images Based on SMS-PCNN Model

*Fengxiang Wang[1], Huang Liang[1]*, Yalun Zhang[2], Qingxia Xu[3] and Ruirui Zong[1]*

*[1] College of Electronic Engineering, Naval University of Engineering, Wuhan, China, [2] Institute of Noise & Vibration, Naval University of Engineering, Wuhan, China, [3] College of International Studies, National University of Defense Technology, Changsha, China*

In the field of ship image recognition and classification, traditional algorithms lack attention to the differences between the grain of ship images. The differences in the hull structure of different categories of ships are reflected in the coarse-grain, whereas the differences in the ship equipment and superstructures of different ships of the same category are reflected in the fine-grain. To extract the ship features of different scales, the multi-scale paralleling CNN oriented on ships images (SMS-PCNN) model is proposed in this paper. This model has three characteristics. (1) Extracting image features of different sizes by parallelizing convolutional branches with different receptive fields. (2) The number of channels of the model is adjusted two times to extract features and eliminate redundant information. (3) The residual connection network is used to extend the network depth and mitigate the gradient disappearance. In this paper, we collected open-source images on the Internet to form an experimental dataset and conduct performance tests. The results show that the SMS-PCNN model proposed in this paper achieves 84.79% accuracy on the dataset, which is better than the existing four state-of-the-art approaches. By the ablation experiments, the effectiveness of the optimization tricks used in the model is verified.

**Keywords: image classification, multi-scale, CNN, ship images, ResNet**

OPEN ACCESS

***Edited by:***
*Marco Leo,*
*National Research Council (CNR), Italy*

***Reviewed by:***
*Divya Sardana,*
*Nike, United States*
*Tyler Highlander,*
*Wittenberg University, United States*

****Correspondence:***
*Huang Liang*
*jwf20@mails.tsinghua.edu.cn*

***Received:*** *04 March 2022*
***Accepted:*** *25 April 2022*
***Published:*** *13 June 2022*

***Citation:***
*Wang F, Liang H, Zhang Y, Xu Q and Zong R (2022) Recognition and Classification of Ship Images Based on SMS-PCNN Model. Front. Neurorobot. 16:889308. doi: 10.3389/fnbot.2022.889308*

## INTRODUCTION

In the military field, ship image classification is used to conduct precise strikes against hostile targets and important to carry out counter-terrorism missions. In the civilian field, ship image classification can assist relevant departments in maritime traffic control, search and rescue, and anti-smuggling activities. Therefore, ship image classification has broad applications and technical requirements in both military and civilian fields.

Currently, there are four types of maritime target images: radar images, remote sensing images, infrared images, and visible light images. Radar (Jiang et al., 2021; Tang et al., 2021) image recognition is all-weather and daylong, which means that it is not easily affected by light and weather. Its mainstream approach is extracting and classifying the features of radar echo signals, so as to achieve autonomous deep feature extraction of the data. Remote sensing (Yang et al., 2014, 2017) image recognition is extracting geometric features such as length and contour of targets in high-resolution SAR remote sensing images, so as to enhance SAR image recognition capability. Infrared image recognition can work in a long distance, which can penetrate thick fog and work day and night.

For visible images, traditional image recognition and classification use techniques including pixel-level edge detection, genetic algorithm, and support vector machine (SVM) classification. Atsuto and Kazuhiro (2004) proposed a multi-frame image processing algorithm to extract contours as basic features of targets for vector analysis, achieving good recognition performance. Xu et al. (2017) designed a multi-level discrimination method based on the improved entropy and pixel distribution with multi-scale and multidirectional decomposition of high-frequency coefficients, which can effectively resist background interferences and improve recognition accuracy and efficiency. Yang and Kim (2012) integrated SAR and automatic identification system (AIS) datasets as one system that can display the position, size, and classification of ships on SAR images. Enriquez de Luna et al. (2005) proposed a silhouette-based decision support system for ship image classification, using an evolved version of the Curvature Scale space (CSS) to improve recognition accuracy.

In the recent years, convolutional neural networks (CNNs) have gradually been widely applied in visible light image classification and recognition. A series of classic CNN models, including AlexNet, VGG, GoogLeNet, ResNet, DenseNet, and so on, stand out in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which covers various fields such as image classification and target detection.

The AlexNet (Alex et al., 2017) model is the first multi-layer CNN with five convolutional layers, three fully connected layers, and a max-pooling layer. This state-of-the-art model is a landmark of CNN model. The proposal of the VGG (Simonyan and Zisserman, 2014) model made the 3 × 3 convolution filters mainstream and improved the accuracy based on the AlexNet, achieving the state-of-the-art results and accelerating further research on the use of deep visual representations in computer vision. Since 2014, the GoogLeNet series has begun to emerge. GoogLeNet (Christian et al., 2014) proposed the inception convolutional neural network that started application of 1 × 1 convolution, reducing the amounts of computation and improving the utilization of computational resources. Its improved model GoogLeNet-V2 (Sergey and Christian, 2015) replaces the 5 × 5 convolution with two layers of 3 × 3 convolution, and the batch normalization proposed in this paper is widely used in deep neural networks. The Inception-V3 model proposed by GoogLeNet-V3 (Christian et al., 2015) achieved the state-of-the-art in the 2015 ILSVRC classification challenge. This model summarized four guidelines for network model design and three optimization tricks to effectively reduce the number of parameters and improve computational efficiency. GoogleNet-V4 (Szegedy et al., 2017) proposed the Inception-V4 model to summarize and integrate. It designed the Inception-ResNet network, which introduces residual connections into the GoogleNet series. The ResNet (He et al., 2016) model is designed with residual connection module, which is easier to optimize and has lower computational complexity and higher accuracy. The ResNeXt (Xie et al., 2017) model improved based on the ResNet model which integrated the ideas of VGG, ResNet, and Inception series. ResNeXt proposed cardinality index to measure complexity, winning the runner-up in the 2016 ILSVRC classification challenge. The DenseNet (Huang et al., 2017) model can effectively alleviate gradient disappearance and strengthen feature propagation, and therefore, it can outperform ResNet in all aspects on Cifar-10, SVHN, and Cifar-100 standard datasets. Besides, it can reduce the number of parameters by half with the same accuracy. The Senet (Jie et al., 2017) model focuses on the channel relationship and introduces attention mechanism into the convolutional neural network, which can adaptively recalibrate channel-wise feature to improve performance.

Convolutional neural networks are also widely applied in speech recognition (Partha et al., 2020; Pradeep and Nirmaladevi, 2021; Yang et al., 2021), medical diagnosis (Seo and Kim, 2020; Toktam et al., 2020; Mustaqeem, 2021), biometrics (Alay, 2020; Sadasivan et al., 2020; Mekruksavanich and Jitpattanakul, 2021; Mohaghegh and Payne, 2021), and other fields. The exploration of convolutional neural networks in other fields provides a reference for us to use CNN for ship image recognition (Cazzato et al., 2020). Jeon and Yang (2021) proposed a classification model integrating CNN and KNN, which classifies from dual-polarization data with 10-m pixel distance to improve the efficiency of ship classification. Li et al. (2021) contribute to intelligent ship vision system by traditionally integrating image processing with machine learning and using target detection method based on CNN. Julianto et al. (2020) proposed a modified method using CNN to classify a patrol vessel dataset and achieved great results. Chen et al. (2020) combined the improved generative adversarial network (GAN) and convolutional neural network, proposing a small ship detection method, which significantly improved the accuracy and robustness of the results. Zhao et al. (2020) used the improved AlexNet convolutional neural network for deep feature extraction of ship images, which significantly improved the classification performance. Xu et al. (2020) proposed a detection method combining visual saliency and a convolutional neural network cascade, which can effectively improve ship detection accuracy and efficiency. Based on the pixel-level fusion of visible and infrared bispectral images, Gao (2020) integrated the algorithms that include image preprocessing, image smoothing, and anti-cloud interference to achieve the detection of ship targets in complex land and sea backgrounds. Ren et al. (2019) proposed a CNN framework with fewer layers and parameter. This method used the Softmax function to classify different ship types and achieved good results. Li et al. (2019) proposed a method to learn discriminative features through supervised learning and good classification performance and built two small optical ship image datasets to verify that results of this method. Hou et al. (2019) proposed a ship detection method based on the visual attention enhanced network to process optical remote sensing images, which improved the recognition accuracy and the detection performance. Endang and Agfianto (2019) combined the CNN-ZFNet architecture and the random forest method to extract the so-called best features defined in the paper and improved the ship image classification accuracy. Shao et al. (2019) proposed a saliency-aware CNN framework for ship detection, comprising comprehensive ship discriminative features. Makantasis et al. (2016) proposed that a methodology fuses a visual attention method that exploits low-level image features appropriately selected for maritime environment. Dong et al. (2018) proposed a hierarchical detection method that finishes the automatic ship detection well. Sobral et al. (2015) proposed a double-constrained robust principal component analysis (RPCA) to improve the object foreground detection in maritime scenes.

By now, various methods to solve the classification and recognition problem of ship targets by convolutional neural networks do not focus on the differences between the grain of ship images. We found that the differences between categories of ships are the hull structure, and the differences between ships of the same category are the equipment and superstructures. Therefore, for the differences in the hull structure between different categories of ships, we should use convolution kernels with larger receptive fields, which is conducive to extracting features such as hull structures that occupy large pixel areas. At the same time, for small differences between diverse ships of the same category, we should use convolution kernels with smaller receptive fields to extract fine features such as ship equipment and superstructures.

Based on the above ideas, we proposed the SMS-PCNN model to classify ship images. The model extracts the feature of images in different sizes by parallelizing convolution branches with different receptive fields. To enhance the speed of converge and avoid overfitting, three optimization tricks are introduced including enhancing specific ship dataset applications, improving training learning rate, and avoiding overfitting. Then, we collected 5,062 ship images of 12 categories on the Internet for the experiments to test the algorithm performance of the SMS-PCNN model and the effectiveness of three optimization tricks.

An outline of the paper is as follows. Section Introduction briefly introduces the research background, technical status quo, faced challenges, and the innovations of this paper. Section SMS-PCNN Model gives a detailed introduction to SMS-PCNN model, which uses three parallel CNN branches to extract the features from different receptive fields, and then classifies the feature maps based on the multilayer perceptron neural networks. Moreover, three optimization tricks of SMS-PCNN model to optimize network performance are also introduced in this chapter. Section Experiment tests the performance of the proposed SMS-PCNN model using the ship image dataset. Besides, a series of comparison experiments were also conducted to test the performance of algorithms using other state-of-the-art approaches as references. Besides, a series of ablation experiments are also conducted to verify the effectiveness of three optimization tricks used in the model. In addition, we conduct experiments on Branch1-CNN model only using 3 × 3 convolution to demonstrate the necessity of designing a multi-branch parallel structure. By comparing the accuracy of both experiments, we demonstrate that designing of a multi-branch parallel structure has better feature extraction capabilities for ship images than other. Section Conclusion summarizes the content of the paper.

# SMS-PCNN MODEL

This chapter presents the SMS-PCNN model proposed for the ship dataset and optimization tricks used by the model. SMS-PCNN uses three parallel CNN branches to extract feature maps from different receptive fields and then classifies the feature maps based on a multilayer perceptron neural network. This chapter is divided into two sections. The first section introduces the network architecture of the SMS-PCNN model in two levels. It introduces the basic single-branch architecture design for implementing the feature map extraction function in the SMS-PCNN model and then presents the SMS-PCNN model architecture design for integrating three single-branch CNNs with different receptive fields. The second section focuses on three optimization tricks for optimizing the network performance, including enhancing specific ship dataset applications, improving training learning rate, and avoiding overfitting.

## Model Architecture

This section is divided into two parts. The first part introduces the basic single-branch architecture design of SMS-PCNN. In this part, basic single-branch in parallel CNNs is used as an example, and the data processing procedure of RGB images after being inputting to each branch is introduced in a systematic way. In the second part, the overall network architecture of SMS-PCNN is presented. This part details the process of fusing feature maps from different single-branch CNNs and classifying the input images based on multilayer perceptron neural networks.

### Basic Single-Branch Architecture of SMS-PCNN Model

The SMS-PCNN model consists of three parallel CNN branches with different receptive fields. The three CNN branches have similar structures, so the first branch in the single-branch benchmark architecture network is taken as an example to illustrate the data flow after the image is inputted to the network. The first branch uses a 3 × 3 convolutional kernel with a relatively small receptive field. The schematic structure of this branch is shown as **Figure 1**.

The single-branch network design is inspired by the Resnet network (Xie et al., 2017) and consists of three parts. The first part is the CNN *Head*, which includes the convolutional layer, (group normalization (GN) layer, ReLu activation layer, and max pooling layer sequentially. The second and third parts are the two layers with similar structure, each containing two blocks. Through two layers, the feature maps will be inputted to the global average pooling layer and then be flattened into a one-dimensional feature vector.

The convolutional layer of CNN *Head* uses 64 convolution kernels of size 3 × 3 with depth 3 to filter the image of size 512 × 512 × 3. The stride is 2 pixels and the padding is 3 pixels. Then, the image goes through the GN layer and ReLu activation layer. Then, max pooling is performed on the feature maps. The size of pooling is 3 × 3, the stride is 2 pixels, and the padding is 1 pixel.

The basic architecture of the block in *Layer* is shown as follows.

A formula can express the output of the block

$$y = ReLu(F(x) + x^{'}) \tag{1}$$

where $x$ and $y$ are, respectively, the input and output of the block. $F(x)$ is the residual mapping which the network needs to learn, and $x^{'}$is the output from the side road. As shown in **Figure 2**, the block contains two roads, the main road and the side road. After entering the block, $x$ will be inputted to the main road and the side road for data processing, respectively. In the main road, feature map $x$ goes through the first convolution layer, the first batch normalization (BN) layer, the first ReLu activation layer, the second convolution layer, and the second BN layer and then becomes feature map $F(x)$. To complete the summation in Eqn (1), the dimensions of $x$ and $F(x)$ must be equal. In the s side road, if the dimensions of $F(x)$ are the same of $x$, no further data processing will be performed, which is called $x^{'} = x$. However, if the dimensions of the feature map output by the main road decrease, x needs to be processed by 1 × 1 convolution layer. The stride of 1 × 1 convolution kernel in **Figure 2** is 2 pixels, and the padding is 1 pixel. The number of channels of the 1 × 1 convolutional layer depends on the number of channels outputting $F(x)$ from the main road in the block. $F(x)$ from the main road and $x^{'}$ from the side road are activated by ReLu function after the summation of the pixels at the corresponding positions. The feature map y is the final output of this block.

There are two cascaded blocks in each layer with the same structure. The difference between the two blocks is that the resolution of the output $F(x)$ in *Block1* decreases and the number of channels changes. In *Block2,* the stride of the first convolution layer is 1 pixel. Therefore, the size of feature map and the number of channels do not change, and a 1 × 1 convolution operation is not required in *Block2*. *Block1* uses a convolution kernel with the stride of 2 pixels instead of a pooling layer to reduce the resolution of the feature map. Springenberg et al. (2014) show that max pooling layer can be replaced by a convolution layer with a stride bigger than 1 pixel without loss of accuracy. Moreover, this can reduce the error to some degree in vast majority of networks. Therefore, we use stride of 2 pixels to compress the size of the feature map to ensure that the network not only reduces the dimension and removes redundant information, but also improves the discriminative accuracy.

*Layer2* has nearly the same design as of *Layer1*, except that *Layer1* has 128 convolutional channels and *Layer2* has 64 convolutional channels. Moreover, when the image is put into the single-branch benchmark network, it passes through the CNN *Head* and 2 layers one by one. Then, the network uses global average pooling to further reduce the resolution of the image and flatten the high-dimensional feature map into a one-dimensional feature vector, pending further classification processing.

Based on the above single-branch architecture network of *Branch1*, other two branches with similar structures can be designed, which are *Branch2* and *Branch3*. The main difference is that the size of the convolutional kernel used in *Branch2* is 5 × 5, and the size of the convolutional kernel used in *Branch3* is 7 × 7. Moreover, to make the feature map of each branch consistent in size for comparison and calculation, the padding of different branches is slightly different. *Branch2* uses 5 × 5 convolutional kernels, and the padding of the first block in each layer is 2 pixels. *Branch3* uses 7 × 7 convolutional kernels, the padding of the *head* convolutional layer is 4, and the padding of the first block in each layer is 3 pixels.

## Overall Architecture of SMS-PCNN Model

Generally, ship images are natural scene images, so the input of SMS-PCNN model is RGB images. Analyzing the ship dataset, we found that the difference in appearance between different categories of ships mainly lies in the hull structure of ships, while the difference in appearance between ships of the same category lies in the superstructures and equipment of ships.

Therefore, for different categories of ships, we should pay more attention to discriminating architectural differences and use convolutional kernels with large receptive fields, which are more conducive to extracting features such as hull structures that occupy large pixel areas. For light differences between ships of the same category, we should use convolutional kernels with small receptive fields to extract features such as ship equipment and superstructures that occupy small pixel areas. For example, there are huge differences between the hull of the Ariake class destroyer and landing ship, which are two different categories of ships, so the convolution kernel with large receptive field is more advantageous. While aircraft carriers contain many categories of ships such as Queen Elizabeth class, the Nimitz class, the main difference between them is the bridge and other special equipment that occupy small pixel areas. In this case, the convolution kernel with smaller receptive field is more advantageous.

Combining the above two considerations, the SMS-PCNN model uses different sizes of receptive fields to extract features of ship images at different levels, so as to achieve a high classification accuracy. The SMS-PCNN model has three parallel branches with different sizes of convolutional kernels that is 3 × 3, 5 × 5, and 7 × 7. The receptive fields of the branches increase, so that the features of the ship images can be extracted and classified in a multi-scale manner. The network structure of SMS-PCNN model is shown in the **Figure 3**.

The number of channels in the SMS-PCNN model has been adjusted two times. In the CNN *Head*, the convolution layer contains 64 channels. In *Layer1*, this number is expanded to 128. The reason is that the feature map shrinks and part of information loses during the process of extracting features from low-dimensional to high-dimensional features. To expand the amount of data flow between layers as much as possible, we expand the number of channels. However, the number of channels is reduced to 64 in *Layer2*. That is because the redundant information increases in the process of further abstraction of high-dimensional features. So, as to further improve the accuracy and reduce the redundant feature information, we decrease the number of channels.

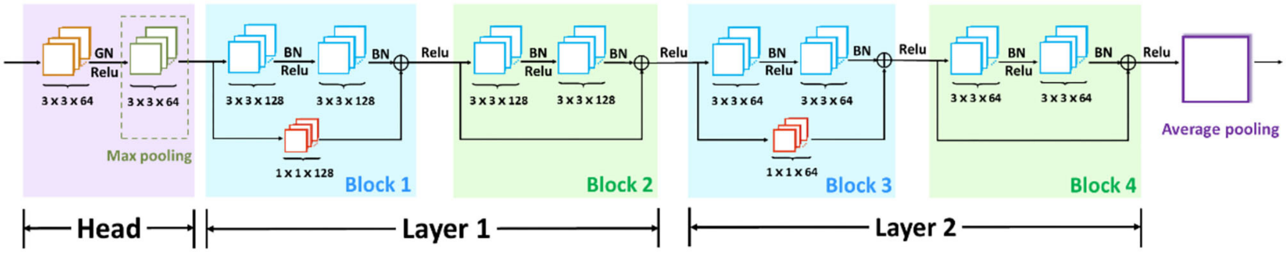


**FIGURE 1 |** Network architecture of *Branch1*.

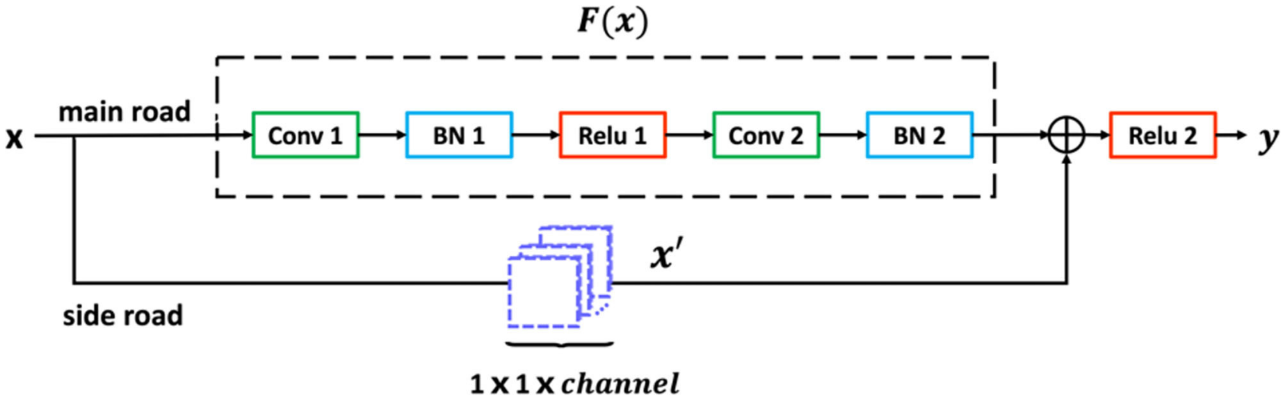


**FIGURE 2 |** Architecture of *Block*.

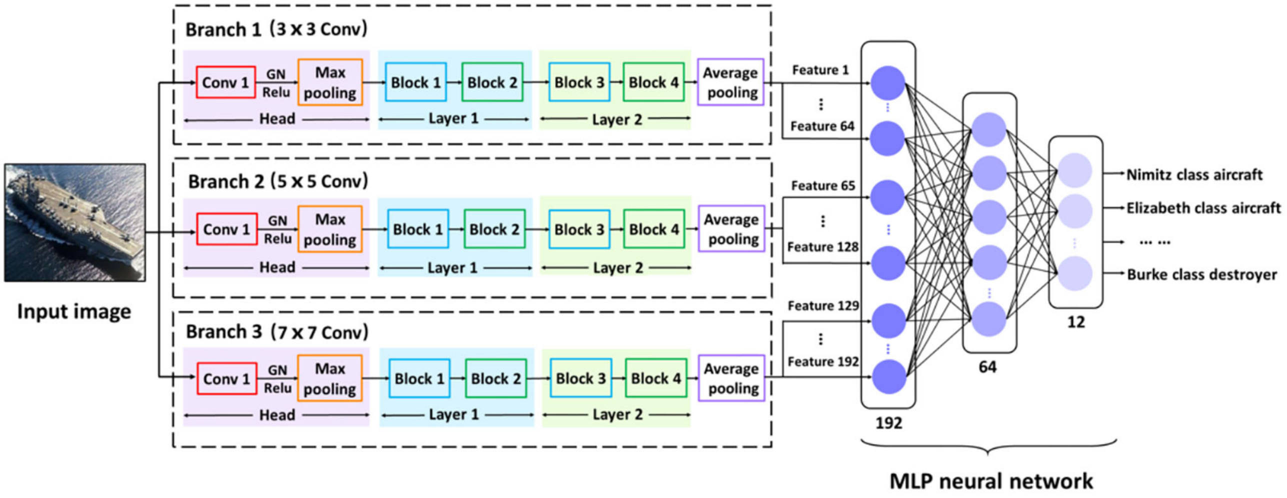


**FIGURE 3 |** SMS-PCNN model.

After generated from three parallel branches with different sizes of receptive fields, feature vectors are linearly concatenated in the SMS-PCNN model.

$$Fused\ (featurevector)_{1\times p} = [featurevector\ (branch1)_{1\times q},\ featurevector\ (branch2)_{1\times q},\ feature\ vector(branch3)_{1\times q}],\quad(2)$$

where $p$ and $q$ denote the dimensions of feature vectors. $p$ is 192 and $q$ is 64. *Branch1* denotes the branch which uses 3 × 3 convolution, *Branch2* denotes the branch which uses 5 × 5 convolution, and *Branch3* denotes the branch which uses 7 × 7 convolution.

These fused feature vectors are processed by multi-layer perceptron neural networks (MLP neural networks), mapping feature vectors with different sizes of receptive fields into different classifications of ship images by decreasing the dimension layer by layer.

Suppose after flatten layer, MLP with $L$ layer is used for feature classification, then the layer $l$ has $m^l$ neurons and the input vector is

$$a^{l-1} = [a_1^{l-1}, a_2^{l-1}, \ldots, a_{m^l}^{l-1}]^T \tag{3}$$

$a^{l-1}$is called the feature vector from layer $(l-1)$. The matrix of weight connected between neurons in layer $(l-1)$ and $l$ is:

$$W^l = \begin{bmatrix} w_{11}^l & w_{12}^l & \cdots & w_{1m^{l-1}}^l \\ w_{21}^l & w_{22}^l & \cdots & w_{2m^{l-1}}^l \\ \vdots & \vdots & \ddots & \vdots \\ w_{m^l1}^l & w_{m^l2}^l & \cdots & w_{m^lm^{l-1}}^l \end{bmatrix} \tag{4}$$

where $w_{jk}^l$ denotes the weights connected between neuron $k$ in layer $(l-1)$ and neuron $j$ in layer $l$ .

The bias vector of neurons in layer $l$ is

$$b^l = [b_1^l, b_2^l, \ldots, b_{m^l}^l]^T \tag{5}$$

then the feature vector $a^l$ from neurons in layer $l$ is

$$a^l = f^l(W^l \bullet a^{l-1} + b^l) \tag{6}$$

where the$f^l$( ) is the activation function used by neurons in layer $l$. After the signal enters the feedforward neuron network, it passes layer by layer and gets the output $a^L$ in the end. The whole process in the feedforward neural network is actually inputting the feature vectors from PCNN into the composite function$F(.; W, b)$. Finally, the output of the network $a^L$ is as follows.

$$\begin{aligned} a^L &= F\left(x; W, b\right) \\ &= f^L(W^L \bullet f^{L-1}\left(\ldots W^2 \bullet f^1\left(W^1 \bullet x + b^1\right) + b^2 \ldots\right) \\ &+ b^L) \end{aligned} \tag{7}$$

After the multilayer perceptron neural network, the SMS-PCNN model uses softmax function to normalize the vector $a^L$. Let the i-th element of the vector $a^L$ be $a_i^L$, then the result obtained after $a_i^L$ being processed by softmax function is

$$S_i = softmax(a_i^L) = \frac{\exp(a_i^L)}{\sum_j \exp(a_j^L)} \tag{8}$$

After softmax function, $a_i^L$ is mapped to the $S_i$.$S_i$ satisfies the properties of the probability distribution, $S_i \in (0,1)$ and$\sum_i S_i = 1$. Therefore, $S_i$ can represent the prediction probability. By selecting the node with the largest prediction probability, we can obtain the final classification category.

The predicted probability $S_i$ will also be inputted to loss function. In the SMS-PCNN model, we use cross-entropy loss function used. It is

$$Loss = -\sum_{i=1}^{K} p_i \log S_i \tag{9}$$

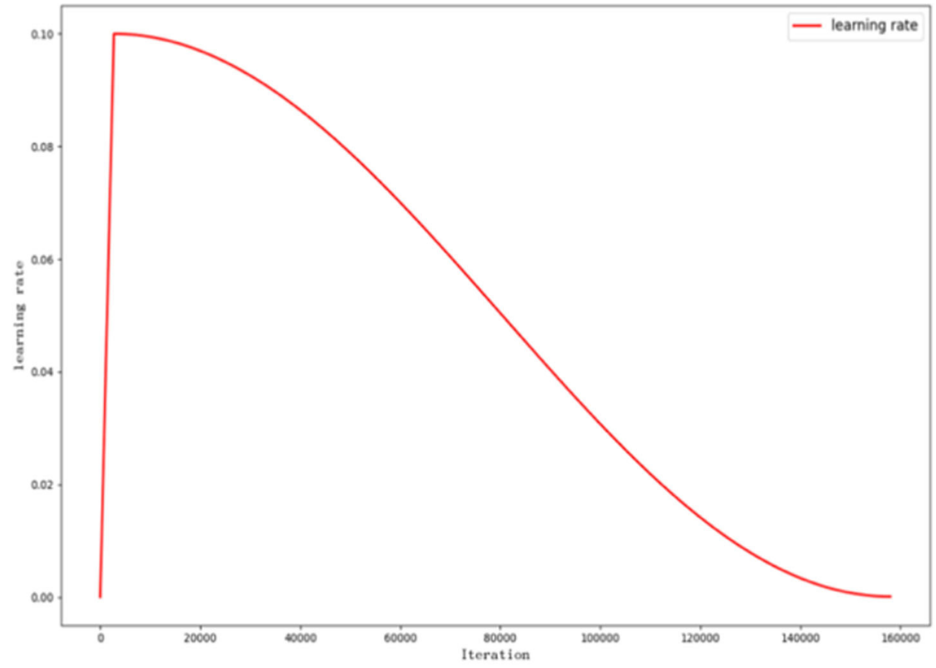


**FIGURE 4 |** Curve of learning rate adjustment.

where $K$ is the number of categories, and $p_i$ denotes the indicator variable (takes the value of 0 or 1). If the predicted category of the sample $i$ is the same as its true category, $p_i$ takes the value of 1 and 0 otherwise. Cross-entropy loss function can measure the prediction performance of the model.

To address the problem of too many parameters in the SMS-PCNN model, the SMS-PCNN model uses L2 regularization to mitigate overfitting. Regularization is a method to reduce the complexity of the model by introducing penalty terms. The L2 regularization is to add the L2 norm of the parameters after the loss function:

$$\hat{Loss} = -\sum_{i=1}^{K} p_i \log S_i + \alpha\Omega(w) \tag{10}$$

where $\hat{Loss}$ is cross-entropy loss function with L2 regularization, and $w$ is the weight coefficient matrix, and the parameter $\alpha$ controls the strength of regularization. $\Omega$ function is the L2 norm, namely, the sum of squares of the weight coefficients:

$$\Omega(w) = \|w\|_2^2 \tag{11}$$

# Tricks for Model Optimization

## Optimization for Specific Ship Dataset Applications

Generally, the distinguishing features of some ships are mainly concentrated in parts that account for a small proportion of the images, such as the bridge of aircraft carriers and the radar of frigates. Therefore, to improve the accuracy of the model, we need to retain as much information of images as possible and use images with large size to contain more information as much as possible. For images with large size, the batch size is reduced to fit the training. However, relatively low batch size will lead to bad performance of the batch normalization (BN) used in the SMS-PCNN model. In this case, Wu and Kaiming (2020) proposed group normalization (GN) to improve the dependence of BN on batch size and proved that when the batch size is <8, the error of using GN is smaller than BN.

The GN layer calculates the mean and variance based on the input data and then uses these two values to normalize the input data. Specifically, GN divides the channels of the layer into groups and uses the mean and variance of the data within the groups.

$$\hat{x_i} = \frac{1}{\sigma_i}(x_i - \mu_i) \tag{12}$$

$\hat{x_i}$ is the feature map of the layer, and $i = (i_N, i_C, i_H, i_w)$. $N$ is the batch axis, $C$ is the channel axis, $H$ and $W$ are the spatial height and width axis. $\mu$ and $\sigma$ here can be calculated by the following equations

$$\mu_i = \frac{1}{m}\sum_{k \in S_i} x_k \tag{13}$$

$$\sigma_i = \sqrt{\frac{1}{m}\sum_{k \in S_i}(x_k - \mu_i)^2 + \epsilon} \tag{14}$$

where ε is a small constant. $S_i$ is a feature collection with a size of $m$ used to calculate $\mu$ and $\sigma$. The solution of $S_i$ in GN is the following equation.

$$S_i = \{k | k_N = i_N, \left\lfloor \frac{k_C}{\frac{C}{G}} \right\rfloor = \left\lfloor \frac{i_C}{\frac{C}{G}} \right\rfloor\} \tag{15}$$

$G$ is the number of groups. In the SMS-PCNN model, we pre-set $G = 2$. $C$ is the number of channels, and $C/G$ is the number of channels in each group. "$\left\lfloor \frac{k_C}{\frac{C}{G}} \right\rfloor = \left\lfloor \frac{i_C}{\frac{C}{G}} \right\rfloor$" indicates that $i$ and $k$ are in the same channel group.

When we used GN in all the normalization layers of the SMS-PCNN models, gradient explosion appeared. Therefore, we only use GN to normalize the output of the head convolution layer of each branch whereas BN is used in other layers, which achieved good results.

## Learning Rate Optimization

The model uses gradient descent to minimize the value of loss function and thus converge to the optimal solution. During the iterations performed by the gradient descent, the learning rate controls the learning progress of the model. To optimize the model learning rate, we use cosine annealing and Warmup (He et al., 2016) technique. Warmup is used to train with a small learning rate at the beginning of to make the network familiar with the data. As the training continues, the learning rate slowly increases and reaches the initial learning rate within a pre-set iteration. Then, we use cosine annealing to adjust learning rate.

The equation of decreasing learning rate is as follows:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\frac{T_{cur}}{T}\pi)) \tag{16}$$

where $\eta_t$ is the learning rate of the $t$-th iteration. $\eta_{\min}$ is the termination learning rate. $\eta_{\max}$ is the initial learning rate. $T_{cur}$ is the current iteration. $T$ is the total iteration. The key of the formula is $\cos(\frac{T_{cur}}{T}\pi)$. As iteration increases, the value of $\cos(\frac{T_{cur}}{T}\pi)$ decreases from 1 to −1, so that the learning rate $\eta_{\max}$ decreases to $\eta_{\min}$.

The learning rate adjustment for the whole model training is shown in the **Figure 4**.

The learning rate is adjusted with iterations. In the first 2,709 iterations (3 epochs), warmup is used. As iteration gradually increases, the learning rate grows to 0.1, and then, this number decreases to $10^{-4}$ using cosine annealing.

## Optimization of Overfitting of the Ship Dataset

On the Internet, ship images are limited in amounts and in low quality, so the ship dataset we compiled from open-source data is limited in size, which leads to overfitting to some degrees when we use the SMS-PCNN model for training. For this reason, we use two methods, label smoothing and dropout, to mitigate overfitting in training.

Label smoothing is a technique proposed by Sergey and Christian (2015) in the model of inception-V2 to mitigate overfitting. The traditional one-hot coding suffers from overconfidence, which leads to overfitting. To address overconfidence, label smoothing decays the item with probability 1 in one-hot, which reduces the weight of category of the ground-truth label in the calculation of loss function. The confidence of the decayed items in one-hot is equally divided into other items, so that each category has a certain amount of confidence, which can suppress overfitting. In the SMS-PCNN model, we add the label smoothing to cross-entropy loss function, and loss function is changed as follows.

$$Loss\left(p,q\right) \ = \ -\sum_{i\,=\,1}^{K} p_i \log \ q_i \rightarrow$$

$$\hat{L}oss\left(p^{'},q\right) \ = \ -\sum_{i\,=\,1}^{K} p_i^{'} \log q_i \tag{17}$$

where $K$ denotes the number of categories, and $p_i$ denotes the indicator variable (takes the value of 0 or 1). $q_i$ is the predicted probability value.

$$p_i^{'} \ = \ (1 \ - \ \varepsilon)p_i \ + \ \varepsilon u(k) \tag{18}$$

where $\varepsilon$ is a smoothing parameter, which is a pre-set hyperparameter. $u(k)$ is a probability distribution and uniform distribution is used here, so $u(k) = \frac{1}{K}$. The result of cross-entropy loss function through label smoothing $\hat{L}oss\left(p^{'},q\right)$ is as follows.

$$\hat{L}oss\left(p^{'},q\right) \ = \ (1 \ - \ \varepsilon \ Loss\left(p,q\right) \ + \ \varepsilon \ Loss\left(u,q\right) \tag{19}$$

In addition to label smoothing, we also use dropout to prevent the model from overfitting. In dropout (Hou et al., 2019), the training process randomly drops out some neurons in hidden layers and maintains the number of neurons in input and output layers. Then, we inputted data through the modified network to propagate forward, and the loss which is calculated by the network is propagated back. Suppose the multilayer perceptron neural network has L layers, and $l \ \in \ \{1 \ldots, L\}$ denotes the neuron in layer $l$. Let $z^l$ denotes the input of the neuron in layer $l$, $y^l$ denotes the output of the neuron in layer $l$, and $W^l$ and $b^l$ denote the weight coefficients and bias of the neuron in layer $l$. Then, the feedforward neural network can be expressed by the following equation.

$$z^{(l\,+\,1)} \ = \ W^{(l\,+\,1)}y^l \ + \ b^{(l\,+\,1)} \tag{20}$$

$$y^{(l\,+\,1)} \ = f\left(z^{(l+1)}\right) \tag{21}$$

where $f$ denotes the activation function. When we use dropout during the model training, the feedforward neural network becomes of the following form.

$$r_i^l \ \sim \ Bernoulli\left(p\right) \tag{22}$$

$$\tilde{y}^{(l)} \ = \ r^{(l)} \ * \ y^{(l)} \tag{23}$$

$$z^{(l+1)} \ = \ W^{(l\,+\,1)}\tilde{y}^l \ + \ b^{(l\,+\,1)} \tag{24}$$

$$y^{(l\,+\,1)} \ = \ f\left(z^{(l\,+\,1)}\right) \tag{25}$$

where the probability $p$ *is a* pre-set hyperparameter. $r^l$ is a random variable that obeys the Bernoulli distribution and is designed to randomly generate a vector of 0 or 1 with probability $p$. Probability $p$ denotes the proportion of the elements of this vector taking the value of 0. $r^l$ is to sample each layer and multiply $y^{(l)}$ which is the output of that layer element by element. $y^{(l+1)}$ denotes the final result of Dropout.

During the prediction of the model, the weight parameter of each neural unit is multiplied by the probability $p$, $w_{test}^{(l)} \ = \ pW^{(l)}$.

In the SMS-PCNN model, we used the dropout before two fully connected layers and set the probability $p$ to be 0.3, which means that 30% of the neurons were dropped out randomly, and the experiment achieved a good accuracy improvement.

# EXPERIMENT

## Dataset

On the Internet, ship images are limited in amounts and in low quality. Therefore, there are no large-scale standard open-source ship datasets. We collected open-source images from the Internet and compiled a dataset containing 12 categories of ship targets. Part of the ship images is shown in **Figure 5**.

The number of each sample of 12 categories applicable to the image classification task is shown in **Table 1**.

In the ship dataset shown in **Table 1**, we divide the training set and valid set according to the ratio of 7:3.

## Experiment and Result

### Experiment and Result of SMS-PCNN Model

We use the SMS-PCNN model optimized by three tricks to conduct experiments. Samples of 12 categories were trained iteratively using 175 epochs with the batch size of 4. The model was trained iteratively using the stochastic gradient descent (Sebastian, 2016). Cosine annealing and warmup optimization techniques were used on the learning rate. After warmup, the learning rate is initially set to 0.1 and finally decreased to $10^{-4}$. The weight of label smoothing is set to $10^{-3}$. The SMS-PCNN model uses L2 regularization and its weight is set to $10^{-3}$. MLP uses two dropouts, both set to discarding 30% of the neurons. The batch normalization, group normalization, and MLP in the model all use the correlation functions that come with PyTorch. After being preprocessed, the natural scene images with the adjusted resolution of 512 × 512 are inputted to the SMS-PCNN model for training.

**Table 2** shows the architecture of SMS-PCNN model for a ship dataset with resolution of 512 × 512. Among them, the images are processed by *Branch1*, *Branch2,* and *Branch3* in parallel and then jointly processed by average pooling. Finally, the feature vectors from three branches are concatenated together. The concatenated feature vectors with 192 dimensions sequentially go through the input layer, the hidden layer with 64 neurons and the output layer with 12 neurons of the MLP neural network. Then, these vectors pass through the softmax function, and the results of classification are outputted.

The "64" in Conv1 denotes the number of convolutional kernels. The "128" in *Layer1* and 64 in *Layer2* both denote the number of convolutional kernels. The "×2" of each Branch

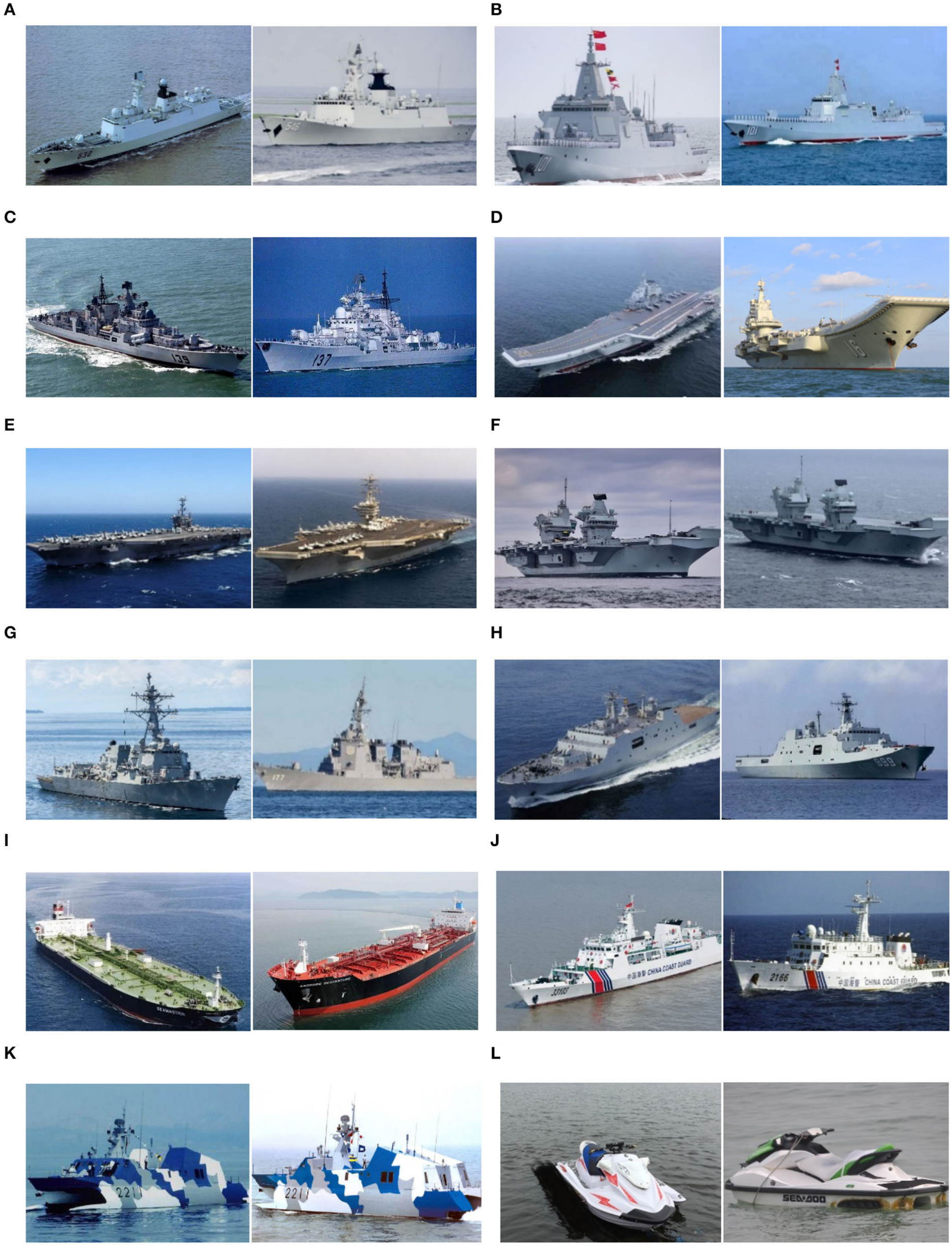


**FIGURE 5 |** Ship dataset. **(A)** Type 054 frigate, **(B)** Type 055 destroyer, **(C)** Sovremenny class destroyer, **(D)** Chinese aircraft carrier Liaoning, **(E)** Nimitz class aircraft carrier, **(F)** Queen Elizabeth-class aircraft carrier, **(G)** Arleigh Burke class destroyer, **(H)** Landing ship, **(I)** Oil tanker, **(J)** Surveillance boat, **(K)** Missile boat, and **(L)** Motorboat.

**TABLE 1 |** Distribution of ship dataset.

| Ship Category | Number of images | Ship category | Number of images |
|---|---|---|---|
| Surveillance boat | 397 | Type 055 destroyer | 409 |
| Motorboat | 474 | Arleigh Burke class destroyer | 445 |
| Landing ship | 495 | Type 054 frigate | 404 |
| Missile boat | 403 | Chinese aircraft carrier Liaoning | 378 |
| Oil tanker | 436 | Nimitz-class aircraft carriers | 502 |
| Sovremenny class destroyer | 420 | Queen Elizabeth-class aircraft carriers | 299 |

**TABLE 2 |** Architectures of SMS-PCNN model.

| Layer name | Output size | Branch1 | Branch2 | Branch3 |
|---|---|---|---|---|
| Conv1 | 256 × 256 | 3 × 3, 64, stride = 2<br>padding = 3 | 5 × 5, 64, stride = 2<br>padding = 3 | 7 × 7, 64, stride = 2<br>padding = 4 |
| Pooling | 129 × 129 | | 3 × 3 max pooling, stride = 2, padding = 1 | |
| Layer1 | 65 × 65 | $\begin{pmatrix} 3*3,128 \\ 3*3,128 \end{pmatrix} \times 2,\ \text{padding} = 1$ | $\begin{pmatrix} 5*5,128 \\ 5*5,128 \end{pmatrix} \times 2,\ \text{padding} = 2$ | $\begin{pmatrix} 5*5,128 \\ 5*5,128 \end{pmatrix} \times 2,\ \text{padding} = 2$ |
| Layer2 | 33 × 33 | $\begin{pmatrix} 3*3,128 \\ 3*3,128 \end{pmatrix} \times 2,\ \text{padding} = 1$ | $\begin{pmatrix} 5*5,64 \\ 5*5,64 \end{pmatrix} \times 2,\ \text{padding} = 2$ | $\begin{pmatrix} 5*5,64 \\ 5*5,64 \end{pmatrix} \times 2,\ \text{padding} = 3$ |
| | 1 × 1 | | Average pooling | |
| Fully connected layer1 | | | (192, 64), dropout = 0.3 | |
| Fully connected layer2 | | | (64, 12), dropout = 0.3 | |

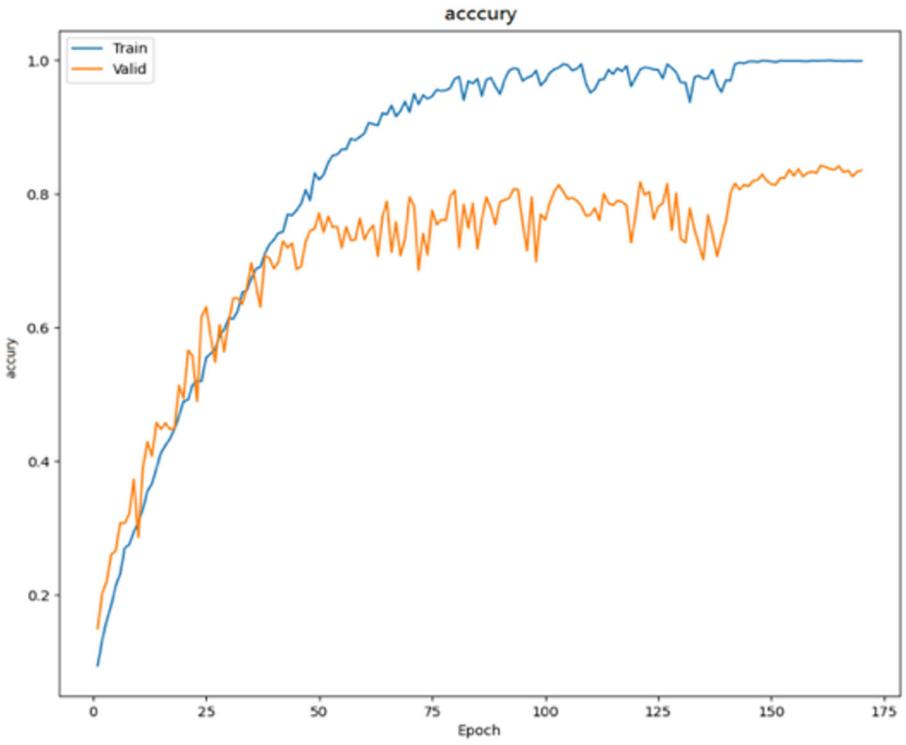


**FIGURE 6 |** Accuracy curve.

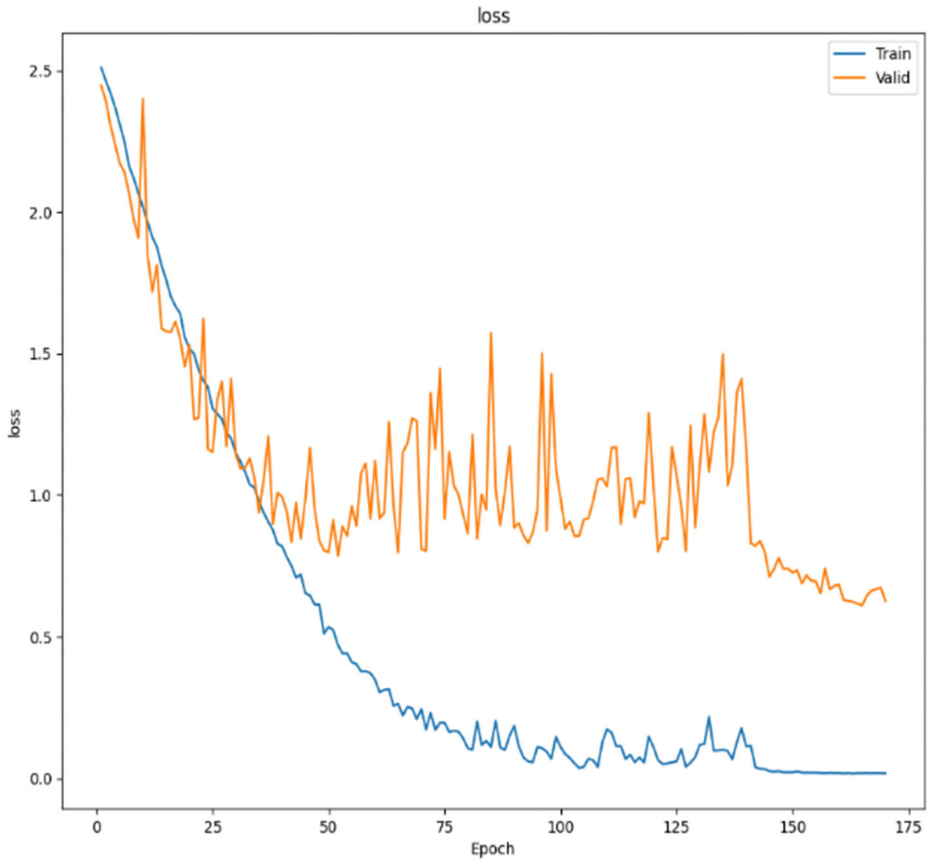


**FIGURE 7 |** Loss curve.

in *Layer1* and *Layer2* denotes that the two cascaded blocks using the same parameters. Take the parameters of fully connected layer1 (192, 64) as an example, "192" indicates the dimension of the input vector and "64" the dimension of the output vector.

Generally, the indicator to evaluate the classification effectiveness of a classifier is its accuracy. The classification accuracy is the proportion of correctly predicted results to the total sample.

As seen in **Figure 6**, the SMS-PCNN model, which combines all tricks, converges faster in training and has nearly reached the highest accuracy rate at the 100th epochs. With the increase of epochs, the learning rate gradually decreases and the training gradually tends to be stable, and the final model optimal results reach 84.79% accuracy.

The loss function in the SMS-PCNN model uses cross entropy loss function with label smoothing and is shown in the **Figure 7**. The improved loss function converges faster and more stably in the training set. At the end of the training, the model training finally converges. The decrease of the loss function on the valid set tends to be stable as the learning rate decreases and reaches the optimal result around 150th epochs. The confusion matrix of the SMS-PCNN model is shown in the following figure.

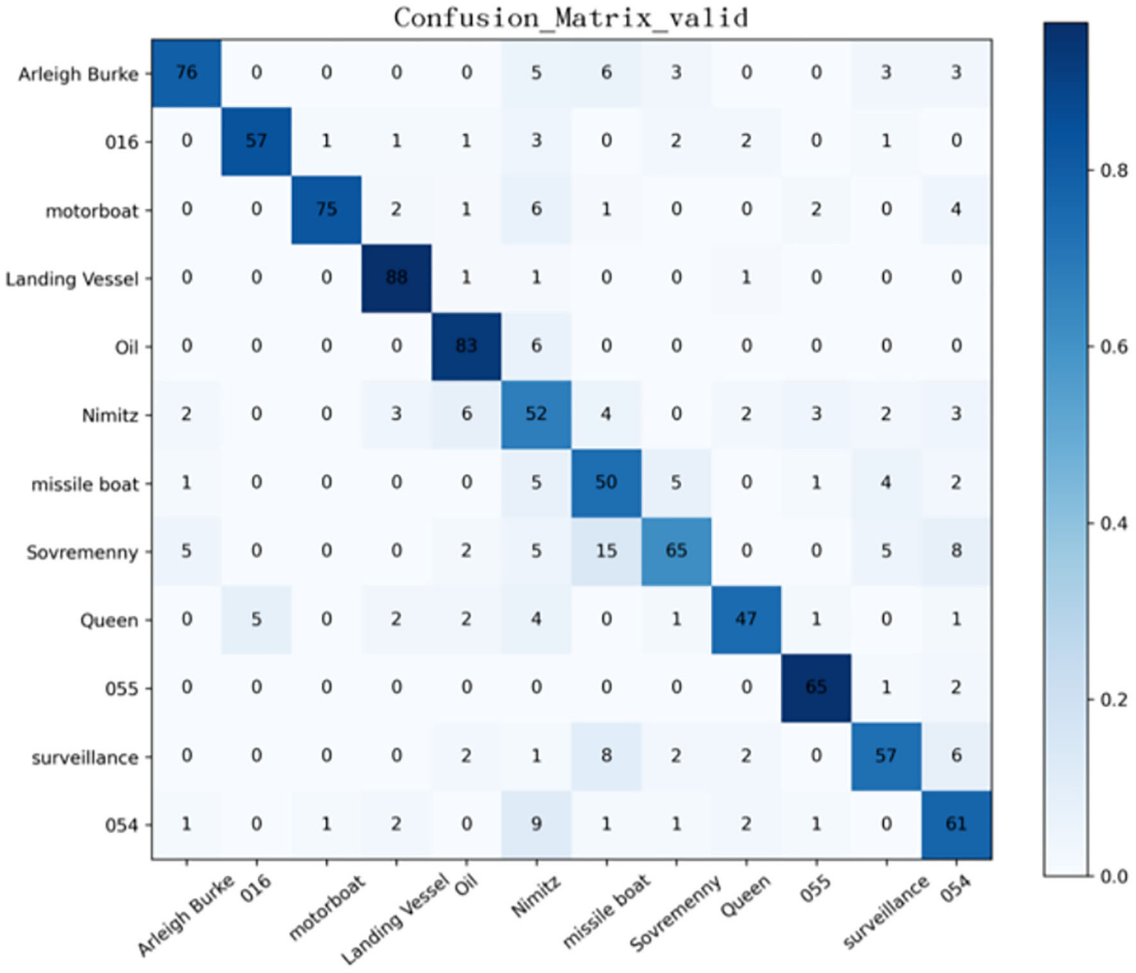
From the confusion matrix of the SMS-PCNN model on the valid set as shown in **Figure 8**, three points can be concluded. (1) The darkest areas of the confusion matrix are concentrated in the diagonal line, showing great classification performance. (2) Among all categories, landing ships, type 055 destroyers, and oil tankers have the highest classification accuracy. (3) Sovremenny class destroyers and Nimitz class aircraft carriers cannot be classified well. The reason why these two categories have a slightly worse classification performance than others is that there are diverse ships of these two categories with minor differences, resulting in the lack of obvious different features between categories.

T-sne (Laurens, 2014) clustering analysis of the SMS-PCNN model for the valid data is plotted in the **Figure 9**.

From the t-sne clustering analysis graph, we can find that the samples of each category are basically concentrated together, and the SMS-PCNN model can classify the samples well.

## Performance Comparison

This section is divided into two parts for performance comparison. The first part is the performance comparison of the SMS-PCNN model with classical model frameworks and the model with single branch. The second part is the validation of the tricks of the SMS-PCNN model.



| | Arleigh Burke | 016 | motorboat | Landing Vessel | Oil | Nimitz | missile boat | Sovremenny | Queen | 055 | surveillance | 054 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Arleigh Burke | 76 | 0 | 0 | 0 | 0 | 5 | 6 | 3 | 0 | 0 | 3 | 3 |
| 016 | 0 | 57 | 1 | 1 | 1 | 3 | 0 | 2 | 2 | 0 | 1 | 0 |
| motorboat | 0 | 0 | 75 | 2 | 1 | 6 | 1 | 0 | 0 | 2 | 0 | 4 |
| Landing Vessel | 0 | 0 | 0 | 88 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Oil | 0 | 0 | 0 | 0 | 83 | 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| Nimitz | 2 | 0 | 0 | 3 | 6 | 52 | 4 | 0 | 2 | 3 | 2 | 3 |
| missile boat | 1 | 0 | 0 | 0 | 0 | 5 | 50 | 5 | 0 | 1 | 4 | 2 |
| Sovremenny | 5 | 0 | 0 | 0 | 2 | 5 | 15 | 65 | 0 | 0 | 5 | 8 |
| Queen | 0 | 5 | 0 | 2 | 2 | 4 | 0 | 1 | 47 | 1 | 0 | 1 |
| 055 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 65 | 1 | 2 |
| surveillance | 0 | 0 | 0 | 0 | 2 | 1 | 8 | 2 | 2 | 0 | 57 | 6 |
| 054 | 1 | 0 | 1 | 2 | 0 | 9 | 1 | 1 | 2 | 1 | 0 | 61 |

**FIGURE 8 |** Confusion matrix of the valid set of SMS-PCNN model.

(1) Comparison to state-of-the-art approaches.

We compared our SMS-PCNN model with 4 state-of-the-art approaches: Resnet-50 (He et al., 2016), Alexnet (Alex et al., 2017), VGG-16 (Simonyan and Zisserman, 2014), and Resnet-18 (He et al., 2016). All the methods were compared on the valid set of our ship dataset. The accuracy comparison is presented in **Table 3**. The results in **Table 3** show that the proposed SMS-PCNN model achieved the best results among all algorithms. All the state-of-the-art comparisons on the ship dataset are fine-tuned well. These comparisons use the same hyperparameters as the SMS-PCNN model we proposed.

In addition, we conducted experiments on network architecture with single branch to analyze the classification of ships. The experiments showed that the Branch1-CNN with small receptive field of 3 × 3 convolutional kernel achieved an accuracy of 75.53%, which is lower than the accuracy of the SMS-PCNN model. This demonstrates the necessity of designing a multi-branch structure.

Comparing with the confusion matrix and t-sne clustering analysis graph in **Figures 10**, **11**, it can be clearly concluded that compared with the SMS-PCNN model, other models do not have well classification effect and have limitations in generalization performance. The confusion matrix shows that other models have low accuracy in discriminating the Queen Elizabeth-class aircraft carrier and the Chinese aircraft carrier Liaoning. The t-sne clustering analysis shows that the point representing Queen Elizabeth-class aircraft carrier and the Chinese aircraft carrier Liaoning are scattered and disorganized, showing a bad performance on clustering. The results of confusion matrix and t-sne clustering analysis are consistent and unified.

(2) Validation of tricks of SMS-PCNN model.

Basic SMS-PCNN means the SMS-PCNN model we used to perform an experiment without trick1, trick2, and trick3. These tricks are used for optimizing the network, including enhancing specific ship dataset applications, improving training learning rate and avoiding overfitting, respectively.

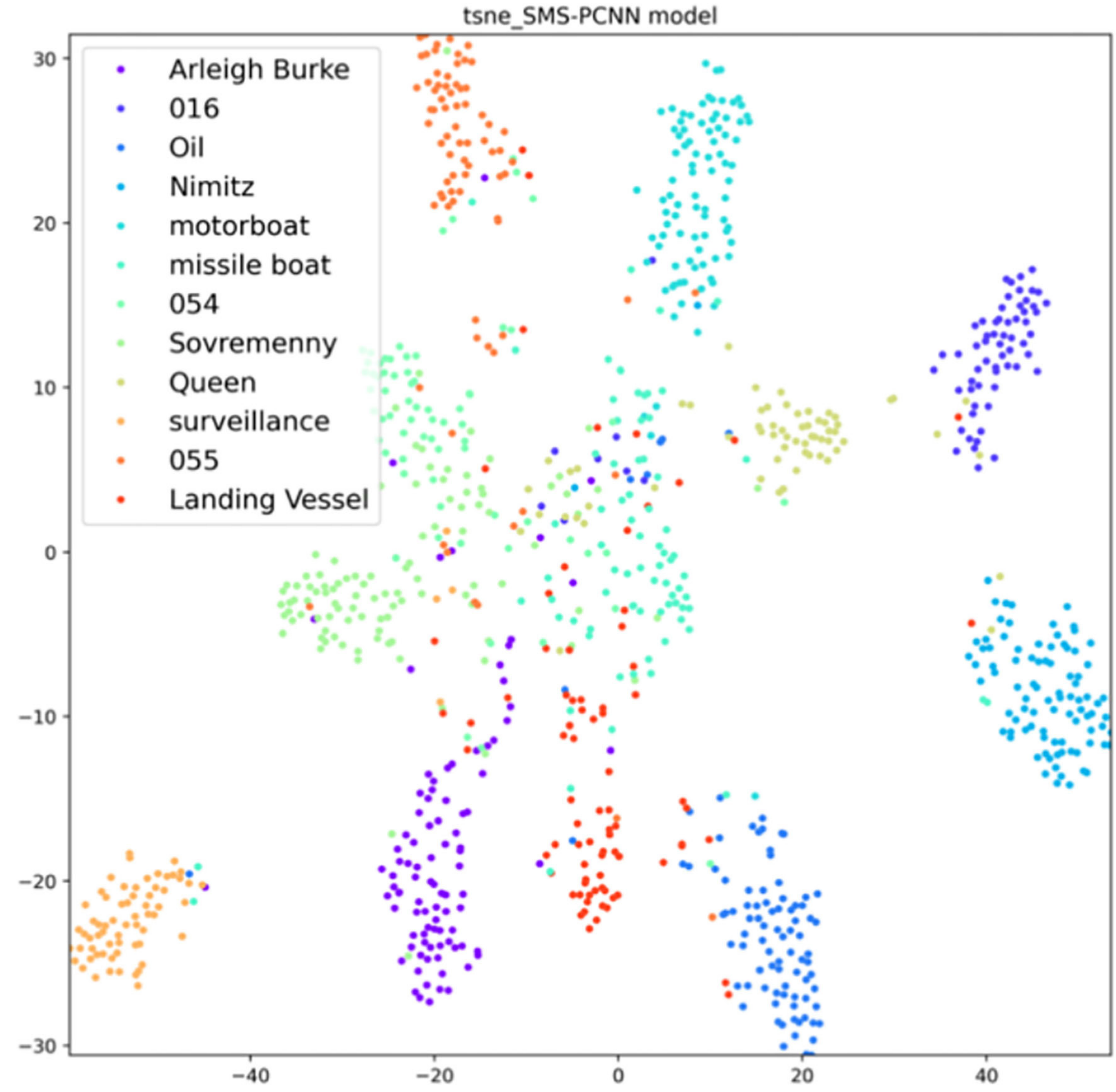


**FIGURE 9 |** SMS-PCNN model t-sne clustering analysis.

**TABLE 3 |** Accuracy comparison with state-of-the-art approaches.

| Model | Accuracy (%) |
|---|---|
| Resnet-50 | 63.82 |
| Alexnet | 64.23 |
| VGG-16 | 67.52 |
| Resnet-18 | 70.09 |
| Branch1-CNN (3 × 3 conv) | 75.53 |
| SMS-PCNN model | 84.79 |

We designed the ablation experiments. The result is shown in **Table 4**.

It demonstrates that the accuracy of Basic SMS-PCNN model without any tricks is 77.28%, which is still higher than the other state-of-the-art approaches and Branch1-CNN in valid dataset mentioned above. This demonstrates the superiority of the model design principles and the basic framework structure. Meanwhile, the ablation experiments examined that the accuracy of the Basic SMS-PCNN model with trick1 increases by 2.16–79.44%. For the Basic SMS-PCNN (add trick1 and trick2), the accuracy increases by 5.96–83.79%. When using trick1, trick2, and trick3, the model accuracy increases by 7.51–84.79%. It can be seen that the use of three tricks did improve the accuracy of SMS-PCNN model and enhance the classification effect.

## Mechanistic Analysis of the Network

To further analyze the mechanism of the SMS-PCNN model and verify the effectiveness of parallelizing multi-scale convolutional branch and adjusting the number of channels, we visualize the

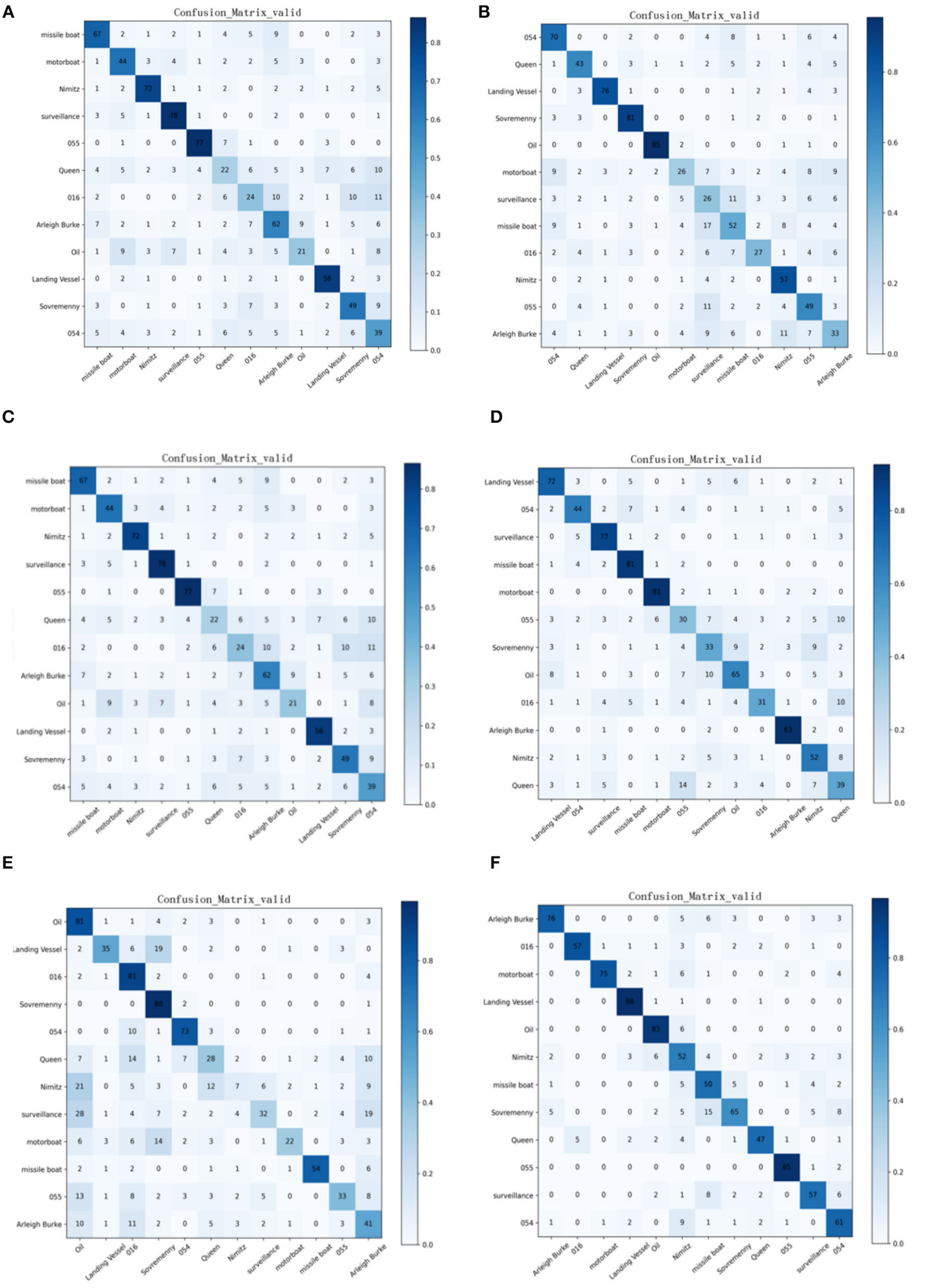


**FIGURE 10 |** Confusion matrix of state-of-the-art approaches and Branch1-CNN in valid dataset. **(A)** Resnet-50, **(B)** Alexnet, **(C)** VGG-16, **(D)** Resnet-18, **(E)** Branch1-CNN (3 × 3 conv), **(F)** SMS-PCNN model.

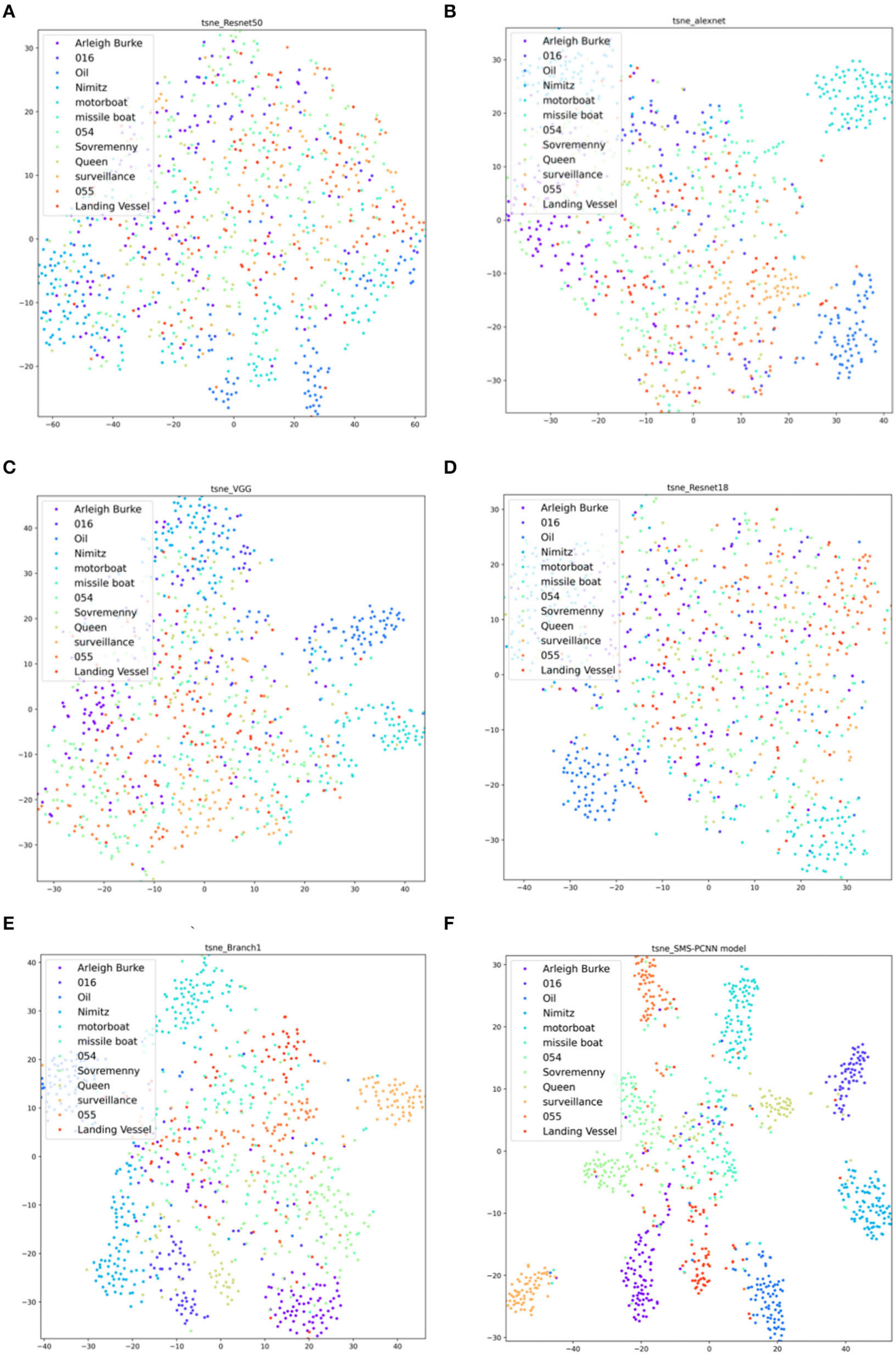


**FIGURE 11 |** T-sne clustering analysis of state-of-the-art approaches in valid dataset. **(A)** Resnet-50, **(B)** Alexnet, **(C)** VGG-16, **(D)** Resnet-18, **(E)** Branch1-CNN (3 × 3 conv), **(F)** SMS-PCNN model.

convolutional kernels and the feature maps of tanker image through each convolutional module. The head convolution layer of each Branch in the SMS-PCNN model is a convolutional kernel with depth 3 designed for three channels of RGB. When visualizing, we combine these convolutional kernels into single convolutional kernel with color. The visualization of the 64 combined convolutional kernels of the head convolutional layer of each branch of the SMS-PCNN model is shown in the following figure.

From the visualization of the convolution kernels in **Figure 12**, we can find that the head convolution layer of *Branch3* with 7 × 7 size convolution kernels mainly focuses on the edge features of images, so it can better extract features such as the hull of ships. The head convolution layers of *Branch2* and *Branch1* with 5 × 5 and 3 × 3 size convolution kernels focus on the fine-grained features of images, so it can better extract features such as ship superstructures and equipment.

We take the tanker image as an example to visualize feature maps that pass through each branch convolution module. In each branch, the tanker image passes through the CNN *Head*, *Layer1,* and *Layer2*, and the feature maps are shown in the following figure.

From **Figures 13A–C**, we can see that after the head convolution layer with different convolution kernels in three branches, the feature maps of the tanker image have different attention. The feature map from Branch3_Head with a larger receptive field is obviously coarser than that of Branch2_Head and Branch1_Head, which have smaller receptive fields. It shows that the feature map from Branch3_Head is more concerned with coarse-grained feature extraction, whereas Branch1_Head with the smallest receptive field is clearer in detail and is obviously more concerned with the fine-grained feature extraction.

**TABLE 4 |** Ablation experiments of SMS-PCNN model.

| Model | Accuracy (%) |
|---|---|
| Basic SMS-PCNN | 77.28 |
| Basic SMS-PCNN(add trick1) | 79.44 |
| Basic SMS-PCNN(add trick1 and trick2) | 83.24 |
| SMS-PCNN(add trick1-3) | 84.79 |

**Figures 13D–F** show the feature maps of the tanker image after processing by the layer1 of each branch. Branch1_layer1 obviously retains more details and pays more attention to the fine-grained features of the ship. Branch1_layer2 has fewer details than Branch1_layer1 and focuses more on the moderate scale features such as the superstructures of the tanker. Branch1_layer3 focuses on the contours of the ship and the coarse-grained features of the hull structure of ships.

**Figures 13G–I** shows the feature maps of the tanker image after the layer2 of each branch. It can be clearly seen that there are huge differences between the feature maps after all the convolution layers of each branch. The feature map of *Branch1* is clear and focuses on fine-grained features such as ship equipment. The feature map of *Branch2* focuses on the superstructures of the tanker. The feature map of *Branch3* pays attention to contours and the hull structure of the tanker.

By comparing the feature maps of different convolution of the three branches, we demonstrate the effectiveness of using multi-scale convolutional branch parallelization for ship images.

We also compared the feature maps of each convolutional layer of single branch. Taking **Figures 13A,D,G** as an example, after the processing of CNN *Head*, *Layer1,* and *Layer2*, feature maps are gradually abstracted from specific low dimension to high dimension, which is effective for image feature extraction. Besides, from **Figures 13D–F**, we observe that there is redundant clutter in part of the output from *Layer1* of each branch. Therefore, the reduced numbers of channels in *Layer2* can

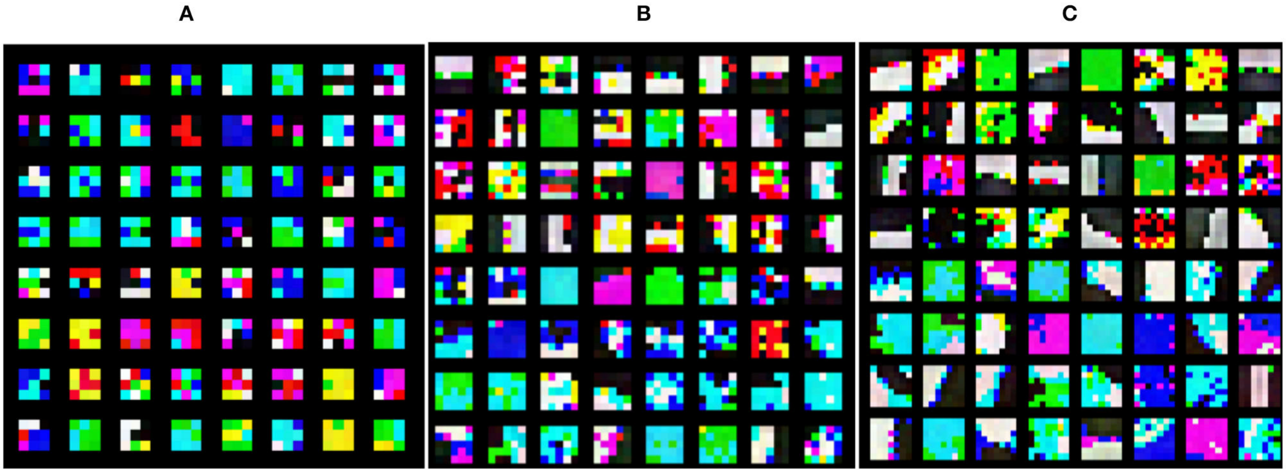


**FIGURE 12 |** Convolutional kernel visualization. **(A)** Branch1_Head(3 × 3conv), **(B)** Branch2_Head(5 × 5 conv), **(C)** Branch3_Head(7 × 7 conv).

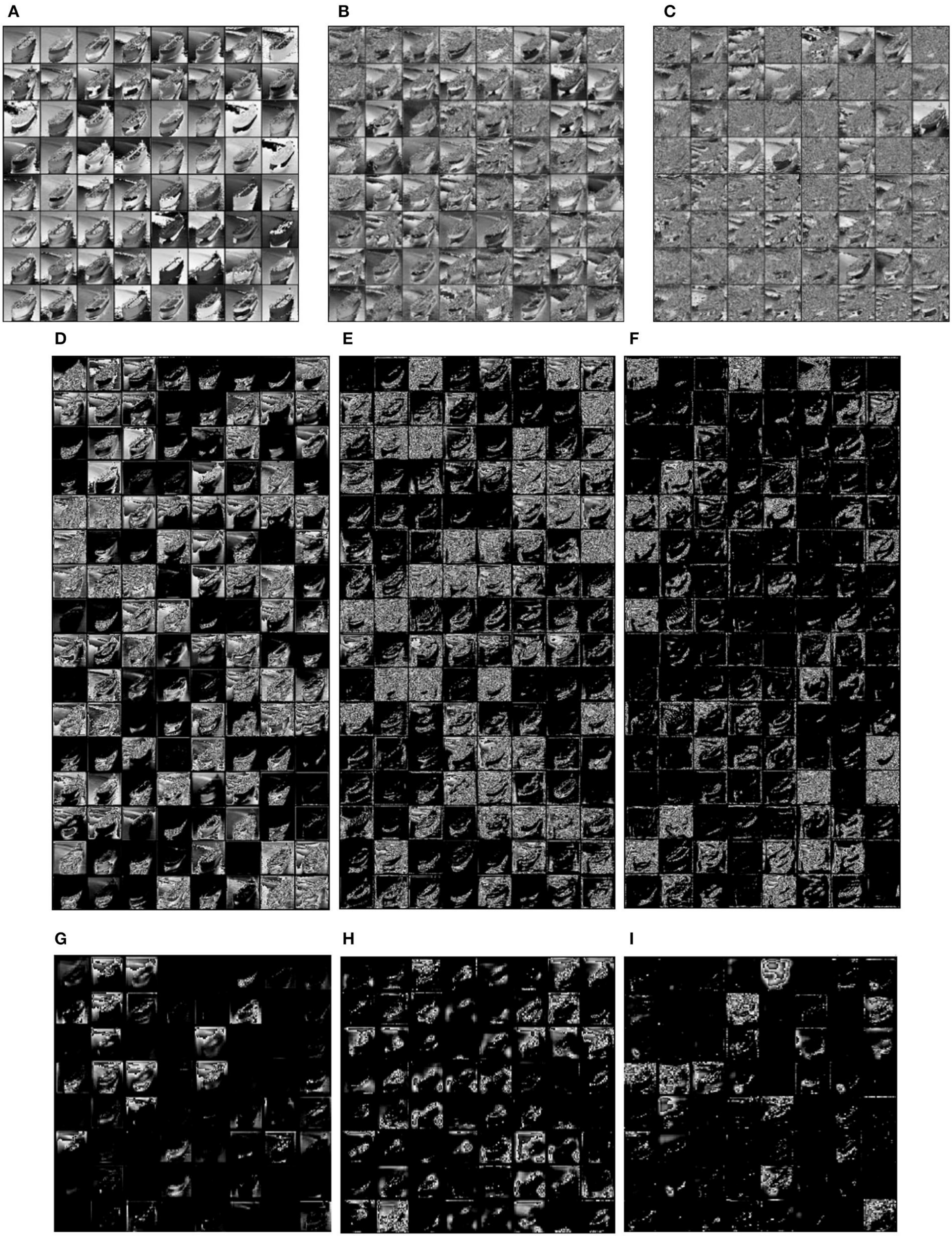


**FIGURE 13 |** Feature maps of oil tanker image after different convolution layers. **(A)** Branch1_Head(3 × 3 conv), **(B)** Branch2_Head (5 × 5 conv), **(C)** Branch3_Head(7 × 7 conv), **(D)** Branch1_layer1 (3 × 3 conv), **(E)** Branch2_layer1 (5 × 5 conv), **(F)** Branch3_layer1(7 × 7 conv), **(G)** Branch1_layer2(3 × 3 conv), **(H)** Branch2_layer2 (5 × 5 conv), **(I)** Branch3_layer2(7 × 7 conv).

filter redundant feature maps from the previous layer. This verified the effectiveness of changing numbers of channels in the network.

## CONCLUSION

We found that the differences between different categories of ships are mainly focused on the hull structures, whereas the differences between different ships of the same category are concentrated in parts such as ship equipment and superstructures. Therefore, we used convolutional kernels with larger receptive fields to extract coarse-grained features such as the hull structure of different categories of ships and used convolutional kernels with smaller receptive fields to extract fine-grained features of different ships of the same category. In this paper, we proposed the SMS-PCNN model for the recognition and classification of ship images. The SMS-PCNN model extracted features of image in different sizes by parallelizing convolutional branches with different receptive fields. In addition, the results of experiments showed that the SMS-PCNN model achieved 84.79% accuracy on a ship dataset containing samples of 12 ship categories. We also conducted a series of comparison experiments to test the performance of algorithms using other state-of-the-art approaches. The results showed that comparing with the Resnet-50, Alexnet, VGG-16, and Resnet-18 models, the SMS-PCNN model improved the accuracy by 14.7–20.97%. We also experimented with the Branch1-CNN model, and the results showed that the SMS-PCNN model with parallel structure improved the accuracy by 9.26% compared with the single-branch model, thus proving that the multi-branch parallel structure outperformed the single-branch for the ship image feature extraction.

The SMS-PCNN model uses three tricks to improve the performance, including enhancing specific ship dataset applications, improving training learning rate, and avoiding overfitting. The experiments showed that the model achieves an accuracy of 77.28% without using any tricks, which is 7.19% higher than Resnet-18. This accuracy is the highest among state-of-the-art approaches. It proved the superiority of the model design principles and the basic framework structure. To verify the effect of optimization tricks on the SMS-PCNN model, we performed a series of ablation experiments. When trick1 was added on the basic SMS-PCNN model, the accuracy increased by 2.16%. When adding trick2 to the SMS-PCNN model with trick1, the accuracy increased by 3.8%. Finally, when adding trick3 to the SMS-PCNN model with trick1 and trick2, the accuracy increased by 1.55%. The ablation experiments validated the effectiveness of the three tricks we used for the ship dataset.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

## AUTHOR CONTRIBUTIONS

FW designed research, performed research, analyzed data, and wrote the paper. HL supervised the entire research process. YZ was involved in writing part of the manuscript. QX and RZ were involved in collected the data. All authors contributed to the article and approved the submitted version.

## FUNDING

This study was funded by Naval University of Engineering, College of Electric Engineering.

## REFERENCES

Alay, N. (2020). AlBaity HH. Deep learning approach for multimodal biometric recognition system based on fusion of Iris, Face, and Finger Vein Traits. *Sensors*. 20, 5523. doi: 10.3390/s20195523

Alex, K., Ilya, S., and Geoffrey, E. H. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM*. 60, 84–90. doi: 10.1145/3065386

Atsuto, M., and Kazuhiro, F. K. (2004). Ship identification in sequential ISAR imagery. *Mach. Vis. Appl.* 15, 149–55. doi: 10.1007/s00138-004-0140-y

Cazzato, D., Cimarelli, C., Sanchez-Lopez, J. L., Voos, H., and Leo, M. A. (2020). Survey of Computer Vision Methods for 2D Object Detection from Unmanned Aerial Vehicles. *J. Imag.* 6, 78. doi: 10.3390/jimaging6080078

Chen, Z. J., Chen, D. P., Zhang, Y. S., Cheng, X. Z., Zhang, M. Y., and Wu, C. Z. Deep learning for autonomous ship-oriented small ship detection. *Saf. Sci.* (2020). doi: 10.1016./j.ssci.2020.104812

Christian, S., Liu, W., Jia, Y. Q., Pierre, S., Scott, E. R., and Dragomir, A. (2014). Going deeper with convolutions. *IEICE Trans Fundam Electron Commun Comput Sci*, abs/1409,4842.

Christian, S., Vincent, V., Sergey, I., and Zbigniewm, W. (2015). Rethinking the inception architecture for computer vision. *IEICE Trans Fundam Electron Commun Comput Sci*, abs/1512, 00567.

Dong, C., Liu, J., and Xu, F. (2018). Ship Detection in Optical remote sensor Images Based on Saliency and a Rotation-Invariant Descriptor. *Rem. Sens.* 18, 400. doi: 10.3390/rs10030400

Endang, A., and Agfianto, E. P. (2019). Ship Identification on Satellite Image Using Convolutional Neural Network and Random Forest. *IJCCS (Indones. J. Comput. Cybern. Syst).* 13, 117–26. doi: 10.22146/ijccs.37461

Enriquez de Luna, C., Otaduy, M. D., and Dorronsoro, C. (2005). A decision support system for ship identification based on the curvature scale space representation. *Proc. SPIE*. 5988, 171–82. doi: 10.1117/12.630532

Gao, X. B. (2020). Design and implementation of marine automatic target recognition system based on visible rem sensor images. *J. Coast Res.* 115, 277–9. doi: 10.2112/JCR-SI115-088.1

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. p. 770–78.

Hou, B. F., Chen, J., Yang, L., Wang, Z., and Ship, Y. (2019). Ship detection for optical remote sensor images based on visual attention enhanced network. *Sensors (Basel)*. 19, 2271. doi: 10.3390/s19102271

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. *CVPR 4700–4708.* 4, 5. doi: 10.1109/CVPR.2017.243

Jeon, H., and Yang, C. S. (2021). Enhancement of Ship Type Classification from a Combination of CNN and KNN. *Electronics*. 10, 1169. doi: 10.3390/electronics10101169

Jiang, J. H., Fu, X. J., Qin, R., Wang, X. Y., and Ma, Z. F. (2021). High-Speed Lightweight Ship Detection Algorithm Based on YOLO-V4 for Three-Channels RGB SAR Image. *Rem. Sens.* 13, 1909. doi: 10.3390/rs13101909

Jie, H., Li, S., Gang, S., and Albanie, S. S. (2017). Squeeze-and-excitation networks. *IEEE Trans Pattern Anal Mach Intell*.

Julianto, E., Khumaidi, A., and Priyonggo, P. Munadhif, I. (2020). Object recognition on patrol ship using image processing and convolutional neural network (CNN). *J. Phys. Conf. Ser*. 1450, 012081. doi: 10.1088./1742-6596/1450/1/012081

Laurens, V. M. (2014). Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* 15, 3221–45.

Li, J. H., Ruan, J. C., Xie, Y. Q. (2021). Research on the development of object detection algorithm in the field of ship target recognition[J]. *Int. Core J. Eng.* 7, 233–41. doi: 10.6919/ICJE.202101_7(1).0031

Li, Z., Zhao, B., Tang, L., Li, Z. and Feng, F. (2019). Ship classification based on convolutional neural networks. *J. Eng.* 2019, 7343–6. doi: 10.1049/joe.2019.0422

Makantasis, K., Protopapadakis, E., and Doulamis, A. (2016). Semi-supervised vision-based maritime surveillance system using fused visual attention maps. *Multimed Tools Appl.* 75, 15051–78. doi: 10.1007/s11042-015-2512-x

Mekruksavanich, S., and Jitpattanakul, A. (2021). Biometric user identification based on human activity recognition using wearable sensors: an experiment using deep learning models. *Electronics*. 10, 308. doi: 10.3390/electronics10030308

Mohaghegh, M., and Payne, A. (2021). Automated biometric identification using dorsal hand images and convolutional neural networks. *J Phys Conf Ser*. 1880, 012014. doi: 10.1088/1742-6596/1880/1/012014

Mustaqeem, K. S. (2021). MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach. *Exp. Syst. Appl.* 167. doi: 10.1016./j.eswa.2020.114177

Partha, P. B., Rappy, S., and Ki-Doo, K. (2020). An automatic nucleus segmentation and CNN model based classification method of white blood cell. *Exp. Syst. Appl.* 149, 113211.

Pradeep, S., and Nirmaladevi, P. (2021). A review on speckle noise reduction techniques in ultrasound medical images based on spatial domain, transform domain and CNN methods. *IOP Conf Ser Mat Sci Eng*. 1055, 012116. doi: 10.1088/1757-899X/1055/1/012116

Ren, Y., Yang, J., Zhang, Q., and Guo, Z. (2019). Multi-feature fusion with convolutional neural network for ship classification in optical images. *Appl. Sci.* 9, 4209. doi: 10.3390/app9204209

Sadasivan, S., Sivakumar, T. T., Joseph, A. P., Zacharias, G. C., Nair, M. S., Thampi, S. M., et al. (2020). Tongue print identification using deep CNN for forensic analysis. *J Intell Fuzzy Syst*. 38, 6415–22. doi: 10.3233/JIFS-179722

Sebastian, R. (2016). An overview of gradient descent optimization algorithms. *IEICE Transac Fundam Electron Commun Comput Sci,*, abs/1609,04747.

Seo, M., and Kim, M. (2020). Fusing visual attention CNN and bag of visual words for cross-corpus speech emotion recognition. *Sensors*. 20, 5559. doi: 10.3390/s20195559

Sergey, I., and Christian, S. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *IEICE Trans Fundam Electron Commun Comput Sci*, abs/1502,03167.

Shao, Z., Wang, L., Wang, Z., Du, W., and Wu, W. (2019). Saliency-aware convolution neural network for ship detection in surveillance video. *IEEE Transac. Circ. Syst. Video Technol.* 30, 781–94.

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:*1409,1556.

Sobral, A., Bouwmans, T., and Zahzah, E. D. Based onSaliency Maps for Foreground Detection in Automated Maritime Surveillance. *ISBC2015 Workshop Conjunct AVSS 2015, Karlsruhe, Germany*. (2015) 1–6. doi: 10.1109./AVSS.2015.7301753

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412, 6806. *and ICLR*. 2015.

Szegedy, C., Ioffe, S., and Vanhoucke, V. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Thirty-First AAAI Conference on Artificial Intelligence, San Francisco*, 4278–84.

Tang, G., Zhuge, Y. C., Claramunt, C., and Men, S. Y. (2021). N-YOLO: A SAR ship detection using noise-classifying and complete-target extraction. *Rem. Sens*. 13, 871. doi: 10.3390/rs13050871

Toktam, Z., Mohammad, M. H., and Mahmood, D. (2020). Adaptive windows multiple deep residual networks for speech recognition. *Exp Syst Appl*. 139, 112840. doi: 10.1016/j.eswa.2019.112840

Wu, Y. X., and Kaiming, H. (2020). Group Normalization. *Int. J. Comput. Vis.* 128, 742–55. doi: 10.1007/s11263-019-01198-w

Xie, S. R., and Girshick, P., Dollar, Tu Z, He, K. (2017). Aggregated residual transformations for deep neural networks. *CVPR*. 1, 3. doi: 10.1109/CVPR.2017.634

Xu, C., Yin, C., Wang, D., and Han, W. (2020). Fast ship detection combining visual saliency and a cascade CNN in SAR images. *IET Radar Sonar Navigation*. 14, 1879–87. doi: 10.1049/iet-rsn.2020.0113

Xu, F., Liu, J. H., Dong, C., and Wang, X. (2017). Ship detection in optical remote sensing images based on wavelet transform and multi-level false alarm identification. *Rem Sens*. 9, 985. doi: 10.3390/rs9100985

Yang, C., and Kim, T. Integration of SAR and AIS for ship detection and identification Ocean Sensing Monitoring. *Proc. SPIE*. (2012). 8372, 1–6 doi: 10.1117./12.920359

Yang, F., Xu, Q. Z., and Li, B. (2017). Ship detection from optical satellite images based on saliency segmentation and structure-LBP feature. *IEEE Geosci. Rem. Sens. Lett.* 14, 1–5. doi: 10.1109/LGRS.2017.2664118

Yang, G., Li, B., Ji, S., Gao, F., Xu, Q. (2014). Ship detection from optical satellite images based on sea surface analysis. *IEEE Geosci. Rem. Sens. Lett.* 11, 641–5. doi: 10.1109/LGRS.2013.2273552

Yang, Y., Hu, Y. Y., Xu, Y., and Wang, S. (2021). Two-stage selective ensemble of CNN via deep tree training for medical image classification. *IEEE Trans. Cybern*. doi: 10.1109./TCYB.2021.3061147

Zhao, S., Xu, Y., Lang, and Li, W. (2020). Optical remote sensor ship image classification based on deep feature combined distance metric learning. *J. Coast Res*. 102, 82–7. doi: 10.2112/SI102-011.1

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.