

Article

SBNN: A Searched Binary Neural Network for SAR Ship Classification

Hairui Zhu, Shanhong Guo *, Weixing Sheng and Lei Xiao

School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; hairui.zhu@njust.edu.cn (H.Z.); shengwx@njust.edu.cn (W.S.); leixiao@njust.edu.cn (L.X.)

* Correspondence: guosh@njust.edu.cn

Abstract: The synthetic aperture radar (SAR) for ocean surveillance missions requires low latency and light weight inference. This paper proposes a novel small-size Searched Binary Network (SBNN), with network architecture search (NAS) for ship classification with SAR. In SBNN, convolution operations are modified by binarization technologies. Both input feature maps and weights are quantized into 1-bit in most of the convolution computation, which significantly decreases the overall computational complexity. In addition, we propose a patch shift processing, which can adjust feature maps with learnable parameters at spatial level. This process enhances the performance by reducing the information irrelevant to the targets. Experimental results on the OpenSARShip dataset show the proposed SBNN outperforms both binary neural networks from computer vision and CNN-based SAR ship classification methods. In particular, SBNN shows a great advantage in computational complexity.

Keywords: radar information intelligent computing; binary neural networks; convolutional neural networks; ship classification; deep learning



Citation: Zhu, H.; Guo, S.; Sheng, W.; Xiao, L. SBNN: A Searched Binary Neural Network for SAR Ship Classification. *Appl. Sci.* **2022**, *12*, 6866. <https://doi.org/10.3390/app12146866>

Academic Editors: Feng Gao, Jin Zheng and Qizhi Xu

Received: 30 May 2022

Accepted: 3 July 2022

Published: 7 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The synthetic aperture radar (SAR) has been widely applied in ocean surveillance due to its all-weather working with the capability of wide range detection [1]. Nowadays, busy international routes have increased the pressure of marine traffic control. For environmental protection, it is necessary to strengthen the monitoring of fishing vessels. Thus, timely ship classification is important for efficient ocean trade route maintenances and sustainable ocean resource developments [2,3]. Meantime, ship classification is an important part of maritime domain awareness (MDA), which holds great value in terms of the national security.

Traditional SAR ship classification methods mainly use the hand-crafted feature extraction and classification, including support vector machine (SVM) [4], decision tree [5], random forest [6], Bayesian classifier [7], Adaboost [8], etc. However, when the inputted SAR images are obtained from complex ship shapes, high sea states, and different types of radars, these manually designed methods provide unsatisfactory results with decreased accuracy and weak robustness.

With the remarkable achievements of deep learning in pattern recognition and image processing [9–12], the deep convolutional neural networks (CNN) are applied to SAR ship classification.

A study about SAR image classification based on transfer learning reveals that SAR ship data require re-training more layers [13]. The classification network, which is adapted from VGGNet [14], contains 13 convolution layers, 5 pooling layers and 1 linear layer. Experimental simulations show the finetuned VGGNet has a good classification accuracy, but with a high complexity and a weak robustness.

A Dual-Polarization SAR classification network based on deep learning and hybrid channel feature loss is proposed in [15]. Features in two different polarized channels are

respectively extracted by the same backbone which is also adapted from VGGNet. Then, predictions of the categories can be obtained with an improved classifier and fused features. The Dual-Polarization classification network is trained with a novel hybrid channel feature loss. However, the bloated architecture and the high requirement on input resolution make this method have an extremely high computational complexity.

A deep learning network, HOG-ShipCLSNet, is proposed with histogram of oriented gradient and principal component analysis for SAR ship classification [16]. In HOG-ShipCLSNet, principal component analysis extracts features in histograms of oriented gradient processed from SAR images while a CNN extracting features in original SAR images. This method achieves good results on both the Sentinel-1 three categories classification task and the Gaofen-3 six categories classification task. Due to the complex and massive computation, it is difficult to implement HOG-ShipCLSNet on mobile platforms.

A Squeeze-and-Excitation Laplacian pyramid network, SE-LPN-DPFF, is proposed for Dual-Polarization SAR ship classification. SE-LPN-DPFF uses the VV channel, the VH channel and the polarization coherence as input. The network architecture includes Dual-Polarization feature fusion, Squeeze-and-Excitation and Laplacian pyramid. Although SE-LPN-DPFF reaches a high accuracy, its large predication latency and dependency on Dual-Polarization data limits the practical application potential.

Following the concept of the data-driven model, a novel small-size Searched Binary Neural Network (SBNN) for SAR ship classification is proposed. Our network has advantages of high classification accuracy, low computational complexity and low hardware implementation requirements, which means SBNN is friendly to mobile or small SAR platforms. We summarize our main contributions as follows:

1. A Searched Binary Neural Network with a decreased computational complexity for SAR ship classification is proposed. Most of the existing SAR ship classification networks are hand-crafted or adapted from computer vision general network. Due to the large complexity of these networks, it is difficulty to implement them on mobile or embedded devices. In this paper, only normalized SAR images are used as input. Then, features are extracted by SBNN, where most of convolution operators are binarized into 1-bit. Compared with existing floating-point SAR ship classification methods, SBNN keeps the sensing capability and classification accuracy at the same level as the state-of-the-art models and achieves a $15\times$ reduction in the number of multiply-accumulate calculations (MAdds).

2. Searching a target network for SAR ship classification with network architecture search (NAS) is researched in this paper. The searched CNN gives a higher accuracy than adapted or hand-crafted SAR classification CNNs. Nowadays, NAS can break through the limitation of subjective understanding and has been considered as an automatic way to design neural networks. To obtain an accurate CNN for SAR ship classification, we construct a super-net and iteratively solve the corresponding bi-level optimization problem under the guide of NAS. However, we observe that pooling operations and Skip Connection acquire large weights through experiments. The searched cells which are nearly filled by weight-free operations lead a decreased feature extraction capability of the corresponding target network. To improve the performance, weight-free candidate operations are ignored when searching the target CNN for SAR ship classification.

3. A novel method to construct an efficient binary network for SAR ship classification is proposed. The searched CNN with the excellent accuracy is too complex to be implemented on mobile or embedded devices. With the development of binary networks, the gap between binary networks and floating-point networks is narrowing. With the help of binary technology including distribution reshaping and binary blocks, we construct binary cells replacing the floating-point cells. After binarization and scaling, the 1-bit network, SBNN, strikes a good trade-off between accuracy and efficiency. Compared with its floating-point prototype, SBNN only requires 0.37 M MAdds with a 2% lower accuracy than the searched CNN. Meanwhile, thanks to the integration of NAS, our proposed SBNN outperforms binary neural networks from computer vision by 2% for the accuracy while nearly achieving a $3\times$ reduced number of MAdds.

4. A patch shift processing is proposed to enhance the learning of SBNN. SAR images usually have a poorer quality than natural optical images, which lays stress on the necessity of learning the feature distribution for a binary networks. The existing binary network technology reshapes the feature distribution channel-wisely without touching the spatial level. Recently, research about spatial attention has shown that processing the spatial information has a positive effect on performance. Hence, we divide feature maps into several patches. Each patch of each channel is adjusted by a learnable parameter. This processing enhances the performance of our binary network by reducing the local noise and the information irrelevant to the targets. Experimental results show our proposed processing further increases the accuracy of SBNN.

The rest of this paper is organized as follows: Section 2 reviews the related work. In Section 3, SBNN is presented with details. The results of experiments and ablation studies are shown in Section 4. Section 5 concludes this paper.

2. Related Work

2.1. Network Architecture Search

NAS is an emergent researching field, after neural networks having achieved many great results. The purpose of NAS is to find an automatic way to design network architectures with pre-defined goals and the extent of the application scope of neural networks. According to the different methods to explore the searching space, NAS has three main categories: reinforcement learning-based searching, evolutionary-based searching and gradient-based searching.

Reinforcement learning-based searching is the most important part in the history of NAS. A meta controller is trained for generating network architectures. Several ideals from reinforcement learning-based searching have a great impact on NAS, even including other approaches. To reduce the searching cost, the concept of searching node cell architecture is proposed [17], where the target network consists of stacked searched cells. Searching cells has a much lower cost than searching a whole network. Then, the weights-sharing technology is proposed to significantly reduce the search time [18]. Trained weights in a searched network are kept after evaluating to avoid retraining the next searched networks from scratch.

Evolutionary algorithms are adopted to force network architectures evolve forward a pre-defined target with genetic operations. Xie et al. encodes a network architecture into a query with genes [19]. An improved evolutionary searching algorithms is proposed to effectively resist the noise in training [20].

In order to further improve the efficiency of searching, gradient-based searching is proposed. The first algorithm in this approach is differentiable architecture searching (DARTS), where NAS has been defined as bi-level optimization problem and the target architecture can be found by gradient descent [21]. Meanwhile, all candidate operations are trained together in a super-net. However, DARTS is still only affordable on workshops and servers. To bring NAS to personal computers, DARTS and partial channel connection are combined together as PC-DARTS [22]. As a result, the hardware requirement has been largely reduced with a higher searching efficiency.

2.2. Binary Network

Binary network is the most extreme form of network quantization, where weights and feature maps are quantized into 1 bit. Hence, the computational complexity of a binary network is at a very low level.

The pioneer of 1-bit neural networks is BNN, which is trained with end-to-end back-propagation [23]. Although BNN has an acceptable performance, floating-point networks outperform BNN much for accuracy. To improve the accuracy, a scale factoring is assigned to each channel in XNOR-Net [24]. The prototype floating network has been updated and shortcuts are used to largely enhance the representational capability in Bi-Real-Net [25]. Then, several improvements are proposed in ReActNet including new block architectures,

distributional loss, and channel-wise reshaping [26]. Last, AdamBNN is proposed based on ReActNet for improving the training of binary networks [27]. Nowadays, the gap between 1-bit networks and their floating-point counterparts is narrowed.

2.3. Spatial Information Processing

Spatial information processing are usually applied to enhance learning capability in computer vision. Networks with spatial information processing have established new records in many challenging datasets, especially large datasets. Recently studies show spatial information processing can be operated at pixel level and patch level.

The most common way to process spatial information at the pixel level is by applying self-attention on each pixel of feature maps. A non-local sensing module is proposed, which can be integrated in most of the common vision network architectures [28]. Weights for each pixel are calculated and added to the original feature maps in this module. To avoid a long sequence for Transformer encoders, a local multi-head self-attention module is proposed, which uses slicing windows to limit the input length [29]. However, those methods give guaranteed performance based on a high computational complexity.

Recently, patch level processing has been developing and becomes popular. This approach divided feature maps into several patches. Vision Transformer encodes 16×16 pixels into a patch and divides an image into total 196 patches [30]. The patches are added by a sin signal and then processed by Transformer encoders. Compact Convolutional Transformer employs learnable parameters instead of fixed signals, which proves that spatial addition operators with learnable parameters have a positive effect [31].

3. SBNN

3.1. Overall Network Architecture

Figure 1 and Table 1 show the proposed SBNN. The inputs of SBNN are normalized SAR images. For a lower computational complexity, binary convolution is widely used in the novel network. In detail, SBNN contains one floating-point Stem convolution layer, five binary convolutional cells and one floating-point classifier. The same as common networks for classification, the forward propagation has 3 stages to ensure a strong feature extraction capability, where the size of the feature maps is reduced twice and the number of channels are doubled twice. The convolutional cells have two types, i.e., Normal Cell and Reduction Cell, and they will be introduced in Section 3.2. In addition, we apply a very small initial channel number in SBNN in order to further guarantee low computational complexity and high memory efficiency.

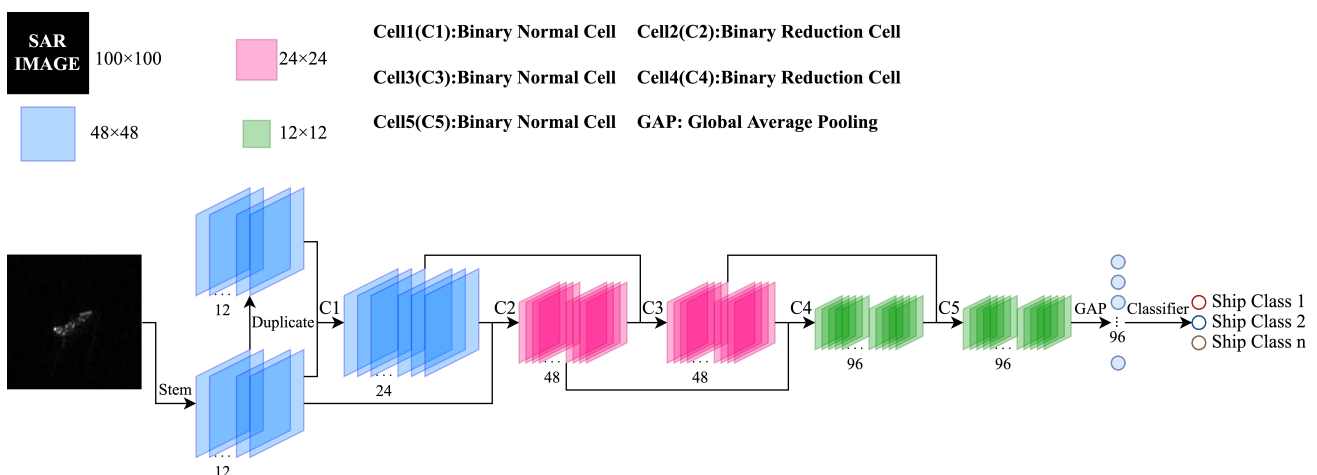


Figure 1. The overall network architecture of SBNN.

Table 1. The details of SBNN network architecture.

Name	Type	Input(s)	Input Shape(s)	Output Shape
Stem	5 × 5 Convolution	SAR Image	1 × 100 × 100	12 × 48 × 48
Cell1	Binary Normal Cell	#1 Stem #2 Stem	#1 12 × 48 × 48 #2 12 × 48 × 48	24 × 48 × 48
Cell2	Binary Reduction Cell	#1 Stem #2 Cell1	#1 12 × 48 × 48 #2 24 × 48 × 48	48 × 24 × 24
Cell3	Binary Normal Cell	#1 Cell1 #2 Cell2	#1 24 × 48 × 48 #2 48 × 24 × 24	48 × 24 × 24
Cell4	Binary Reduction Cell	#1 Cell2 #2 Cell3	#1 48 × 24 × 24 #2 48 × 24 × 24	96 × 12 × 12
Cell5	Binary Normal Cell	#1 Cell3 #2 Cell4	#1 48 × 24 × 24 #2 96 × 12 × 12	96 × 12 × 12
GAP	Global Average Pooling	Cell5	96 × 12 × 12	96
Classifier	Linear	GAP	96	3 or 6

3.2. Searched Normal Cell and Reduction Cell

With the development of deep learning, the gap between binary networks and their real-valued counterparts is narrowing [25,26]. We construct the binary cells for SAR ship classification under the following steps. First, efficient floating-point node convolutional cells, Normal Cell and Reduction Cell, are searched by NAS. Then, the searched cells are modified and binarized. The last step involves inserting our proposed patch shift processing. We believe this method can keep the advantages of NAS with high accuracy and small size with a much lower computational complexity benefiting from binarization.

The architectures of Normal Cell and Reduction Cell in SBNN are shown in Figure 2. Each cell has eight convolutional operations. The connections and types of the operations are the main difference between Normal Cell and Reduction Cell. As shown in Figure 2, Normal Cell contains three Binary Conv 5 × 5 Operations, three Binary Conv 3 × 3 Operations, two Binary Dilated Conv 5 × 5 Operations and Reduction Cell contains three Binary Reduction Conv 5 × 5 Operations, two Binary Conv 3 × 3 Operations, one Binary Reduction Dilated Conv 5 × 5 Operations, one Binary Dilated Conv 5 × 5 Operations and one Binary Dilated Conv 3 × 3 Operation.

In each cell, linear layers reshape the inputs at first. Then, the patch shift processing enhances the presentation of feature maps. Next, the binary convolutional operations extract and output deep features. The types and connections of those binary convolutional operations are set following the searched floating-point cells.

3.2.1. Efficient Searching

Thanks to the high flexibility, NAS can break through the limitations of subjective understanding and imagination [17,18,32]. Nowadays, NAS is considered as an automatic way to reduce the cost of time and labor in designing network architectures. In this paper, the prototype floating-point network is searched by PC-DARTS, which has a low searching cost and a friendly hardware requirement [22]. Take the information propagated from node i to node j as an example, the mixture operation $f_{i,j}(\cdot)$ in searching stage can be formulated as:

$$f_{i,j}(x_i; S_{i,j}) = \sum_{o \in O} \frac{\exp\{\alpha_{i,j}^o\}}{\sum_{o' \in O} \exp\{\alpha_{i,j}^{o'}\}} o(S_{i,j} * x_i) + (1 - S_{i,j}) \quad (1)$$

where: O denotes a pre-defined space of operations, and $o(\cdot)$ is a fixed operation in O . x_i is the output of node i . $\alpha_{i,j}^o$ is a learnable parameter for weighting the candidate operation $o(\cdot)$, and $S_{i,j}$ denotes the randomly sampled channels.

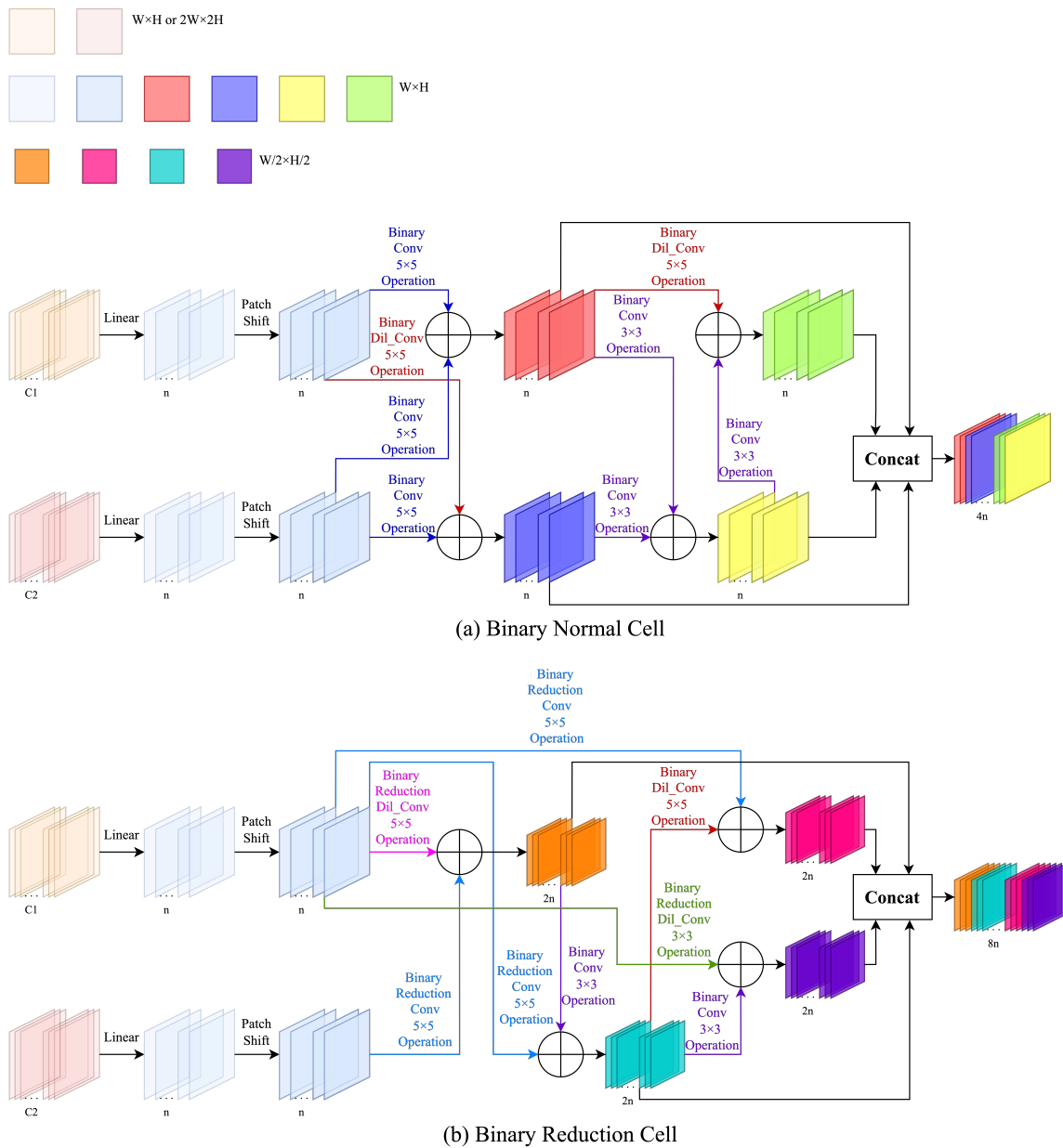


Figure 2. The architectures of the cells in SBNN.

Sampling channels can directly reduce the searching cost and improve the memory efficiency. On the downside, the searching may become unstable. Edge normalization in PC-DARTS ensures the stability of searching. The computation of x_j , which denotes the output of node j , is:

$$x_j = \sum_{i < j} \frac{\exp\{\beta_{i,j}\}}{\sum_{i' < j} \exp\{\beta_{i',j}\}} f_{i,j}(x_i; S_{i,j}) \tag{2}$$

where $\beta_{i,j}$ is an edge normalization coefficient.

When searching on a natural optical image dataset, such as ImageNet, the candidate operations are usually 3×3 Conv, 5×5 Conv, 3×3 Dil_Conv, 5×5 Dil_Conv, Average Pooling, Max Pooling and Skip Connect [21]. Each 3×3 or 5×5 Conv operation contains two Depthwise Separable Convolution blocks. A Dil_Conv operation only has one Depthwise Separable Convolution blocks where the first convolution is dilated [33]. According to the recently researches, Max Pooling has an unsatisfied performance in binary networks [24].

PC-DARTS gives weight-free operations, i.e., Average Pooling, Max Pooling and Skip Connect, very large values resulting an unstable searching in the SAR ship dataset which has a small data size and a high task difficulty. Weight-free operations can give more consistent outputs, which means they are more likely to be placed in the target network. For reaching a better classification accuracy, weight-free operations from the original PC-DARTS are ignored. We only consider weight-equipped operations including 3×3 Conv, 5×5 Conv, 3×3 Dil_Conv, 5×5 Dil_Conv as candidate operations in searching.

3.2.2. Modification and Binarization

Floating point networks give an excellent performance, but they always have a strict requirement on hardware. Although the computing cost of binary networks is small, it is difficult to search a binary network directly because of the weak learning capability of binary convolution. Recent research on modifying and binarizing floating-point networks in computer vision reveal that the gap of accuracy between a floating-point network and a binary networks sharing a similar architecture is narrowed. Hence, we modify and binarize the searched float point network by replacing the floating-point operations with the binary operations, which have a lower computational complexity. In the proposed SBNN, binary operations consist of one ReActNet Block [26] or two staked ReActNet Blocks. The hyperparameters (e.g., Kernel Size, Dilation, etc.) of the first convolution in each ReActNet block are set as the same as the corresponding Depthwise Separable Convolution in the floating-point operations. The details of modification and binarization are shown in Table 2, and the architectures of ReActNet Blocks are shown in Figure 3.

Table 2. The details of the modification and binarization of SBNN.

Original Operation	Binary Operation	Original Block(s)	Binary Block(s)	First Convolution Kernel	First Convolution Stride	First Convolution Dilation
3×3 Conv	Binary 3×3 Conv	#1 Depthwise Separable Convolution	#1 ReActNet Normal Block	#1 3×3	1	1
		#2 Depthwise Separable Convolution	#2 ReActNet Normal Block	#2 3×3	1	1
5×5 Conv	Binary 5×5 Conv	#1 Depthwise Separable Convolution	#1 ReActNet Normal Block	#1 5×5	1	1
		#2 Depthwise Separable Convolution	#2 ReActNet Normal Block	#2 5×5	1	1
3×3 Dil_conv	Binary 3×3 Dil_conv	Depthwise Separable Convolution	ReActNet Normal Block	3×3	1	2
5×5 Dil_conv	Binary 5×5 Dil_conv	Depthwise Separable Convolution	ReActNet Normal Block	5×5	1	2
Reduction 3×3 Conv	Binary Reduction 3×3 Conv	#1 Depthwise Separable Convolution	#1 ReActNet Reduction Block	#1 3×3	2	1
		#2 Depthwise Separable Convolution	#2 ReActNet Normal Block	#2 3×3	1	1
Reduction 5×5 Conv	Binary Reduction 5×5 Conv	#1 Depthwise Separable Convolution	#1 ReActNet Reduction Block	#1 5×5	2	1
		#2 Depthwise Separable Convolution	#2 ReActNet Normal Block	#2 5×5	1	1
Reduction 3×3 Dil_conv	Binary Reduction 3×3 Dil_conv	Depthwise Separable Convolution	ReActNet Reduction Block	3×3	2	2
Reduction 5×5 Dil_conv	Binary Reduction 5×5 Dil_conv	Depthwise Separable Convolution	ReActNet Reduction Block	5×5	2	2

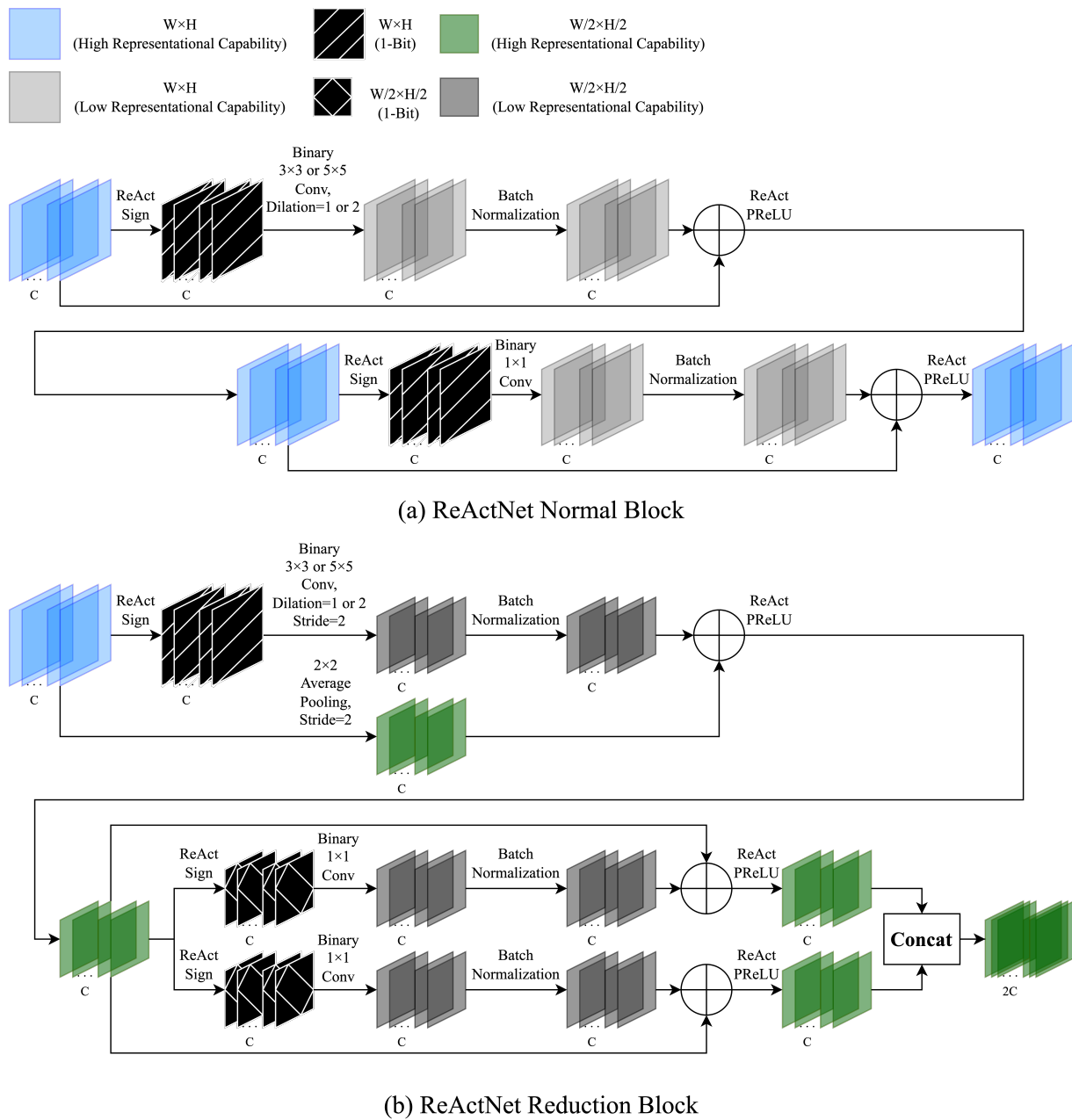


Figure 3. The architectures of the ReActNet Blocks in SBNN.

In ReActNet Blocks, floating-point feature maps are transformed into binary feature maps by ReAct Sign function. The traditional Sign function may harm the training due to an unsuitable truncation threshold. When the threshold is low, lots of background noise makes the binarized feature maps become unclear. Meanwhile, a high threshold produces too little useful information in the binarized feature maps. ReAct Sign reshapes the distribution with channel-wise parameters, which is a learnable method to find the suitable thresholds for binarization:

$$a_c^{\text{binary}} = h(a_c^{\text{float}}) = \begin{cases} 1 & \text{if } a_c^{\text{float}} > p_c \\ -1 & \text{if } a_c^{\text{float}} \leq p_c \end{cases} \quad (3)$$

where the superscripts binary and float refer to binary and floating-point values, respectively, p_c is a learnable parameter, c indicates the index of a channel.

The core of binary convolutional networks is the binary cross-correlation between 1-bit inputs and 1-bit weights. Compared with the floating-point cross-correlation, the binary form has a $64\times$ computation efficiency and can be described as:

$$x_{\text{binary}} * w_{\text{binary}} = \text{popcount}(\text{XNOR}(x_{\text{binary}}, w_{\text{binary}})) \quad (4)$$

where x and w denote inputs and weights respectively.

SBNN has 4 types of binary convolution with different hyper-parameters, which are demonstrated in Figure 4.

The traditional binary convolutional networks have a poor performance with the low representational capability. The shortcuts in ReActNet Blocks largely increase the number of different choices in a pixel after binary convolution and gives a representation capability close to floating-point convolution.

Feature maps from the shortcut and the Batch Normalization [34] are added through an addition operator. Then, ReAct PReLU shifts the distribution and activates the added feature maps. The formula of ReAct PReLU is shown below:

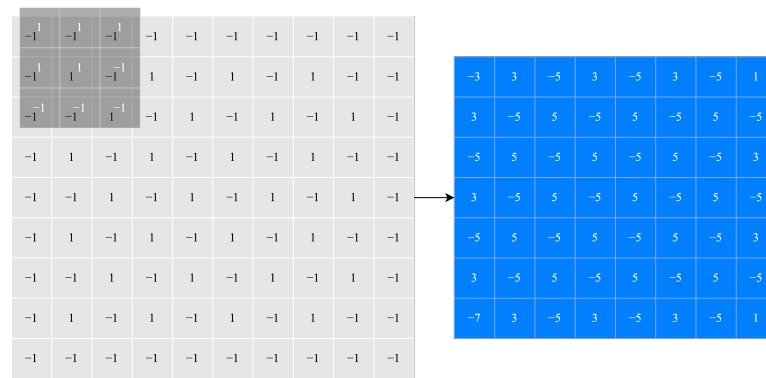
$$f(a_c) = \begin{cases} a_c - \gamma_c + \xi_c & \text{if } a_c > \gamma_c \\ \epsilon_c(a_c - \gamma_c) + \xi_c & \text{if } a_c \leq \gamma_c \end{cases} \quad (5)$$

where ϵ , ξ , γ are trainable parameters, c indicates the index of a channel.

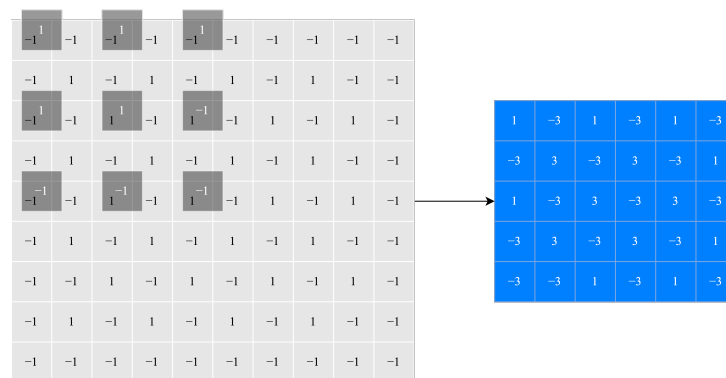
3.2.3. Patch Shift Processing

SAR ship images, which contain coherent speckle noise, always have a poorer quality than natural optical images. Hence, it is more difficult to train a binary classification network about the distribution of the feature maps. ReAct Sign and React PReLU put effort on reshaping the distribution channel-wisely. However, those functions do not cover any adjustment at spatial level. Inspired by recent researches on spatial attention of neural networks [30,31,35], we propose a spatial representation reinforcement method named as patch shift processing, which is illustrated in Figure 5. Firstly, the proposed patch shift processing divides the input feature maps into $p \times p$ patches. Then, a learnable parameter is given to each patch in each channel. Finally, the features are reinforced by adder operators. This processing is trained with other weights in the network through backpropagation. Important areas can be highlighted and information irrelevant to the target is reduced in feature maps.

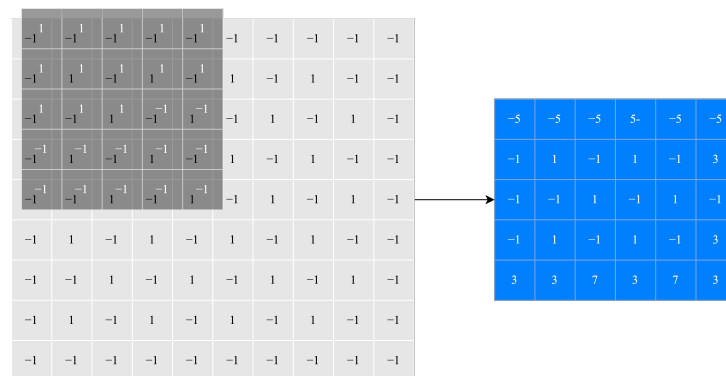
In Figure 5, the sample which is directly binarized without the processing has too few useful information to be learnt. On the contrary, the processed image gives a high-quality 1-bit feature map with a clear shape of the ship after binarization. Considering the training difficulty and the computational complexity, we set p as 12 and place the proposed patch shift processing after linear layers in each cell.



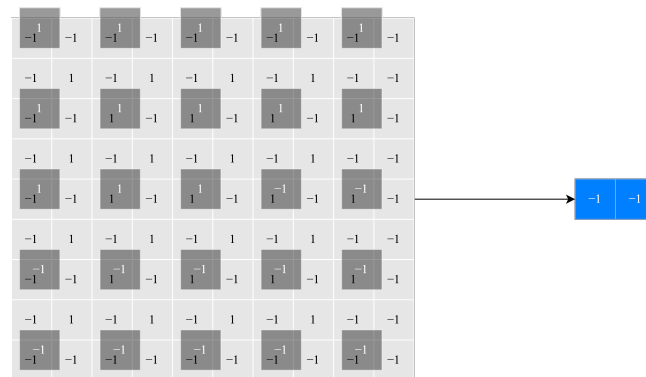
3x3 Binary Convolution



3x3 Binary Dilated Convolution



5x5 Binary Convolution



5x5 Binary Dilated Convolution

Figure 4. The demonstration of 4 different types of binary convolution in SBNN.

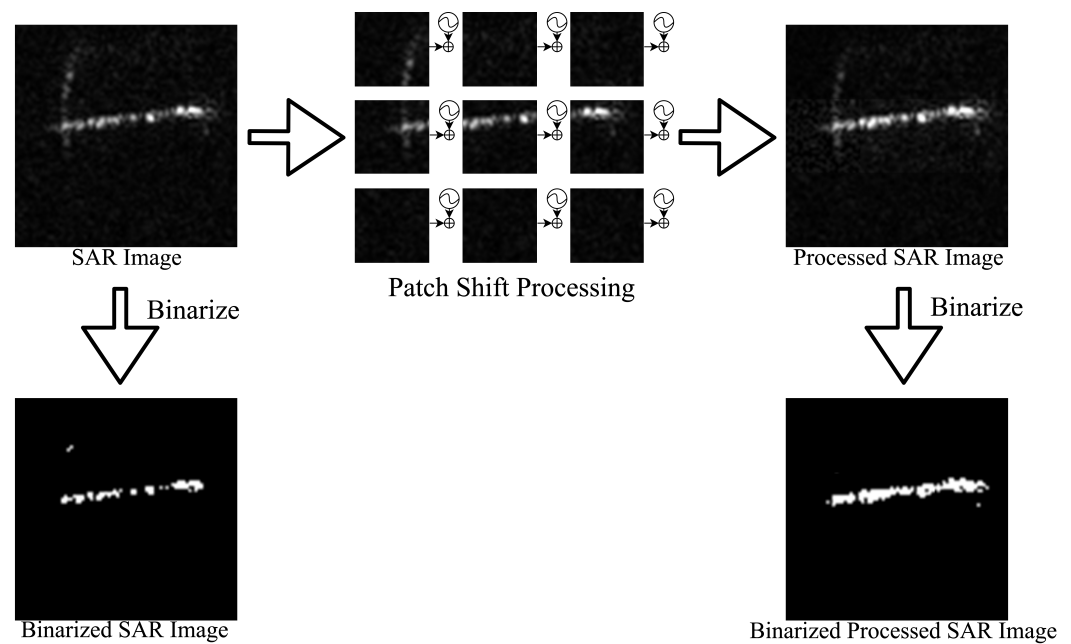


Figure 5. The illustration of patch shift processing.

4. Experiments

Our experiments are run on OpenSARShip which is an open access SAR ship image dataset [36]. A personal computer with Intel core i5-11600 CPU, 16G memory and only single RTX 3060 is used in the experiments. The operating system is Ubuntu20.04 LTS. SBNN is driven by python3. Not only the searching and the training of the floating-point network, but also the training of SBNN are conducted with Pytorch neural network framework [37]. The used version of Pytorch is 1.91+cu111, where +cu111 means CUDA toolkit 11.1 is required for GPU acceleration.

4.1. Dataset

OpenSARship is a widely used dataset in SAR ship classification research and was released by Huang et al. in 2018. OpenSARship contains 11,346 SAR ship images covering many categories of ships from Sentinel 1 satellite. Furthermore, the single-look complex data which contain both VV and VH polarization images with a high range resolution for targets constitute the final datasets in our experiments. The number of different categories of ships in OpenSARship are distributed in a wide range. We set two tasks to verify the effectivity of the proposed SBNN following the researches about SAR ship classification [16,38]. Those two tasks are the three categories SAR ship classification and the six categories Dual-Polarization SAR ship classification respectively.

The three categories SAR ship classification task includes bulk carrier, container ship and tanker which own approximately 80% of international routes [39]. To mitigate the imbalance number of different categories, we use the same way of HOG-ShipCLSNet [16] to create the dataset. We take 70% of the least number of all three categories as the number of training targets for each category. The rest targets are set in the test set. In addition, the VV and VH SAR images of a same target are treated as two independent samples. Table 3 shows the sample numbers for each category. The six categories Dual-Polarization SAR ship classification task includes bulk carrier, container ship, tanker, cargo, general cargo and fishing. We take 80% of the least number of all six categories as the number of training targets for each category. Similarly, the rest targets are set in the test set. In the six categories task, each sample consists of both the VV and VH SAR images of a same target. Table 4 shows the sample numbers for each category in the six categories Dual-Polarization SAR ship classification. Some samples used in experiments are shown in Figure 6.

Table 3. The sample numbers of each category in the three categories SAR ship classification.

Ship Category	All VH Samples	All VV Samples	Training Samples	Test Samples
Bulk Carrier	333	333	338	328
Container Ship	573	573	338	808
Tanker	242	242	338	146

Table 4. The sample numbers of each category in the six categories SAR Dual-Polarization ship classification.

Ship Category	All Samples	Training Samples	Test Samples
Bulk Carrier	333	100	233
Cargo	671	100	571
Container Ship	573	100	473
Fishing	125	100	25
General Cargo	142	100	42
Tanker	242	100	142

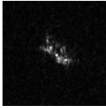
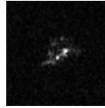
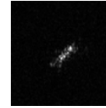
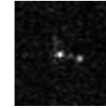
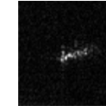
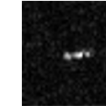
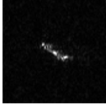
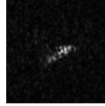
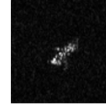
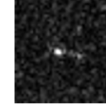
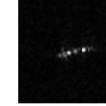
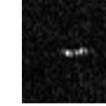
						
	VH	VH	VH	VH	VH	VH
						
	VV Bulk Carrier	VV Cargo	VV Container Ship	VV Fishing	VV General Cargo	VV Tanker
Three Categories SAR Ship Classification	✓	✓				✓
Six Categories Dual-Polarization SAR Ship Classification	✓	✓	✓	✓	✓	✓

Figure 6. SAR ship samples in OpenSARShip.

4.2. Experimental Details

4.2.1. Training

To improve the learning with a small data set size, we apply a data augmentation policy including random cropping and random horizontal flipping to expand the diversity of data. The inputs of SBNN share a same shape of $1 \times 100 \times 100$, where 1 indicates VV or VH polarization.

OpenSARShip dataset provides uint8 files, which have been processed and generated from SAR products with related automatic identification system information. Compared with prior studies using PCA or a sub-network in data pre-processing, SBNN requires a much easier data pre-processing where only center cropping and resizing are employed. Considering the targets are surrounded by a large area of clear background, each sample is processed as follows.

In the three categories classification task, targets share a similar size. To reduce the distortion effect of resizing, targets larger than 128×128 are center-cropped to 112×112 , and targets smaller than 112×112 are resized to 112×112 .

In the six-categories dual-polarization classification task, targets are lying on a large range of size, which means the distortion effect of resizing is obvious and unavoidable. All targets are resized to 128×128 , and then center-cropped to 112×112 .

Notice that the data pre-processing provides a size of 112×112 which is slightly larger than the input of SBNN. This size is instrumental in applying the data augmentation policy, where random cropping produces data with a size of 100×100 .

With reference to AdamBNN [27] about training binary networks, we use a distillation learning [40] method to train the weights in SBNN. Binary networks have a low learning capability, which means direct training with the cross-entropy loss and hard labels could

result in poor performance. The soft label generated by floating-point networks can help the learning of binary networks. When training SBNN, the floating-point network, which acts as a teacher, is the searched CNN by PC-DARTS, the prototype of SBNN. The initial channel number of the teacher is 8 and the cell number is 8. SBNN is the student and trained with the KL divergence loss which can be computed as below:

$$\mathcal{L} = -\frac{1}{b} \sum_{\text{class}} \sum_{i=1}^b P_{\text{class}}^{\text{float}}(x_i) \log\left(\frac{P_{\text{class}}^{\text{binary}}(x_i)}{P_{\text{class}}^{\text{float}}(x_i)}\right) \quad (6)$$

where: b is the batch size, class indicates the category of a sample, P is the SoftMax output of the floating-point network or the binary network.

In this paper, Adam optimizer [41] updates the weights in SBNN. To be specific, the weight decay for the convolution and linear layers is set as 0. The weight decay for other learnable parameters, such as the weights in Batch Normalization, is set as 1×10^{-5} . Training has two stages. Each stage has 256 epochs, a batch size of 32 and a base learning rate of 0.001. We apply a linear scheduler to change the learning rate during training, the specific learning lr_{ep} for each epoch can be calculated as:

$$lr_{ep} = 0.001 \times \left(1 - \frac{ep}{256}\right) \quad (7)$$

where: ep denotes the number of trained epochs.

In the first stage, all binary convolutional operators are replaced by the floating-point convolutional operators. The network learns the distribution of feature maps. In the second stage, we load the weights from the first stage and replace the floating-point convolutional operators with binary convolutional operators. Then, the network is trained with the same policy of the first stage.

4.2.2. Inferring

Center cropping takes over from random cropping and generates appropriate inputs of SBNN during inferring. In the three categories SAR ship classification task, the corresponding category of the neuron which produces the highest probability is the predication of a sample. In the six-categories Dual-Polarization SAR ship classification, we use a simple decision fusion when an input contains two images. In detail, the VV and VH SAR images are successively inputted into SBNN. If different polarization data of the same target give different results, the result with a higher probability will be the final prediction. The additional computation of this decision fusion can be ignored, and the network can be trained as same as the single polarization task.

4.3. Results

4.3.1. Comparison with CNN-Based SAR Ship Classification Methods

Table 5 lists the results of SBNN and the four other floating-point CNN-based SAR ship classification methods in the three categories task. Those methods for comparison are: finetuned VGG [13], plain CNN [42], group squeeze excitation sparsely connected CNN (GSESCNNs) [43] and the combination of HOG, PCA and deep learning (HOG-ShipCLSNet) [16]. Notice that SBNN and the above methods except HOG-ShipCLSNet do not require additional computation out of the networks.

Table 6 lists the results of SBNN with fusion and three other floating-point CNN-based Dual-Polarization SAR ship classification methods in the six-categories task. The counterparts are: VGG with hybrid channel feature loss [15], mini hourglass region extraction and dual-channel efficient fusion network [44] and the squeeze-and-excitation Laplacian pyramid network (SE-LPN-DPPF) [38].

As shown in Tables 5 and 6, our proposed SBNN outperforms all other methods and reaches the best accuracy: 80.03% and 56.73% for the two tasks, respectively. On the hardware implementation side, SBNN also has great advantages. The number of weights

in SBNN is only 0.37 M. Moreover, the computational complexity of SBNN is very small with a MAdds number of 4.67 M which is almost $19\times$ lower than the HOG-ShipCLSNet.

The confusion matrixes of SBNN in the three-categories task and the six-categories task are shown in Tables 7 and 8, respectively. In the three-categories task, SBNN has a good accuracy for each category. In the six-categories task, most samples of bulk carrier, cargo, container ship and fishing can be classified correctly.

Table 5. The results in the three-categories SAR ship classification task.

Method	Accuracy	MAdds	Weights
Finetuned VGG [13]	69.27%	13.84 B	15.52 M
Plain CNN [42]	67.41%	2.17 B	47.44 M
GSESCNNs [43]	74.98%	—	—
HOG-ShipCLSNet [16]	78.15%	89.46 M (Not Including HOG and PCA)	65.11 M
SBNN (ours)	80.03%	4.67 M	0.37 M

Table 6. The results in the six-categories Dual-Polarization SAR ship classification task.

Method	Accuracy	MAdds	Weights
VGG With Hybrid Channel Feature Loss [15]	55.26%	27.68 B	15.01 M
Mini Hourglass Region Extraction and Dual-Channel Efficient Fusion Network [44]	54.93%	≥ 369.93 B (Dynamic)	7.45 M
SE-LPN-DPFF [38]	56.66%	—	—
SBNN with fusion (ours)	56.73%	9.34 M	0.37 M

Table 7. The confusion matrix of SBNN in the three-categories SAR ship classification task.

	Bulk Carrier	Container Ship	Tanker
Bulk Carrier	67.07%	29.88%	3.05%
Container Ship	12.25%	84.16%	3.59%
Tanker	3.42%	8.90%	87.67%

Table 8. The confusion matrix of SBNN in the six-categories Dual-Polarization SAR ship.

	Bulk Carrier	Cargo	Container Ship	Fishing	General Cargo	Tanker
Bulk Carrier	60.52%	20.60%	18.45%	0.00%	0.00%	0.43%
Cargo	17.16%	70.23%	3.68%	3.15%	0.35%	5.43%
Container Ship	30.87%	10.78%	57.29%	0.00%	0.42%	0.63%
Fishing	0.00%	32.00%	0.00%	64.00%	0.00%	4.00%
General Cargo	23.81%	52.38%	9.52%	0.00%	2.38%	11.90%
Tanker	9.15%	71.83%	4.23%	5.63%	0.00%	9.15%

4.3.2. Comparison with Modern Computer Vision Binary Networks

Table 9 lists the results of SBNN and other three modern computer vision binary networks in the three-categories task. The compared binary networks are: Bi-RealNet [25], ReActNet [26] and AdamBNN [27]. Not only is the number of weights obtained, but also the binary MAdds and the floating-point MAdds are counted separately. According to the research [25,26], the number of total MAdds can be calculated by adding the number of floating-point MAdds to $1/64$ of the number of binary MAdds. From Table 9, SBNN has the best accuracy and the lowest number of MAdds close to 30% of other binary networks. In the meantime, the size of SBNN is much smaller than other networks listed together, which benefits the implement on the tiny-size mobile devices.

Table 9. The comparison with modern computer vision binary networks in the three-categories SAR ship classification task.

Method	Accuracy	Binary MAdds	Floating Point MAdds	MAdds	Weights
Bi-RealNet18 [25]	76.76%	307.89 M	10.77 M	15.58 M	11.18 M
ReActNet [26]	70.83%	884.74 M	0.67 M	14.49 M	28.33 M
AdamBNN [27]	78.16%	884.74 M	0.67 M	14.49 M	28.33 M
SBNN (ours)	80.03%	95.51 M	3.18 M	4.67 M	0.37 M

4.4. Ablation Study

4.4.1. Candidate Operations

We conduct an ablation experiment to verify the positive effect of deleting weight-free operations from the candidate operation list. We compare the original candidate operations and only the weight-equipped operations, i.e., 3×3 Conv, 5×5 Conv, 3×3 Dil_Conv, 5×5 Dil_Conv. To obtain the fair results, the rest configurations of searching, modifying, scaling and training are kept as same. Table 10 shows the results in the three categories task. The accuracy of the searched CNN and the corresponding binary CNN are improved by 2.4% and 1.5%, respectively, which means our proposed search policy, deleting weight-free operation, can give a better target network in SAR ship classification.

Table 10. Comparison of different candidate operations.

Operations	Network	Accuracy
7 (Original)	Searched Floating Point CNN by PC-DARTS (IC = 8, L = 8)	79.72%
	Binary CNN	78.63%
4 (Only Weight-equipped Operations)	Searched Floating Point CNN by PC-DARTS (IC = 8, L = 8)	82.14%
	Binary CNN (SBNN)	80.03%

4.4.2. Patch Shift Processing

To verify the enhancement from the proposed patch shift processing, we construct a new network through removing the patch shift processing of SBNN. The new network is trained as same as SBNN in the three categories task. The results of using the patch shift processing and not using are listed in Table 11. We find that the patch shift processing nearly gives an improvement of 1%. As a result, SBNN has an accuracy over 80%, which outperforms several full floating-point networks designed for SAR ship classification.

Table 11. Ablation study on the patch shift processing.

Patch Shift Processing	Accuracy
✓	80.03%
×	79.33%

5. Conclusions

In this paper, a Searched Binary Network, SBNN, for SAR ship classification is proposed. Experimental results show our network achieves good results in different tasks with a strong robustness. The proposed SBNN outperforms other networks for the model size and the computational complexity. In particular, the number of MAdds in SBNN is very small, which means SBNN has a strong potential for being implemented on the tiny-size mobile devices.

Author Contributions: Conceptualization, H.Z., W.S. and S.G.; methodology, H.Z.; software, H.Z.; validation, H.Z., L.X., S.G. and W.S.; formal analysis, H.Z. and L.X.; investigation, H.Z. and L.X.; data curation, H.Z. and S.G.; writing—original draft preparation, H.Z.; writing—review and editing, H.Z., S.G. and L.X.; visualization, H.Z.; supervision, S.G.; project administration, W.S.; funding, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62001227.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within this article.

Acknowledgments: The authors thank the reviewers for their help on the article during review progress.

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

References

1. Bruschi, S.; Lehner, S.; Fritz, T.; Soccorsi, M.; Soloviev, A.; van Schie, B. Ship surveillance with TerraSAR-X. *IEEE Trans. Geosci. Remote Sens.* **2010**, *49*, 1092–1103. [\[CrossRef\]](#)
2. Petit, M.; Stretta, J.M.; Farrugio, H.; Wadsworth, A. Synthetic aperture radar imaging of sea surface life and fishing activities. *IEEE Trans. Geosci. Remote Sens.* **1992**, *30*, 1085–1089. [\[CrossRef\]](#)
3. Park, J.; Lee, J.; Seto, K.; Hochberg, T.; Wong, B.A.; Miller, N.A.; Takasaki, K.; Kubota, H.; Oozeki, Y.; Doshi, S.; et al. Illuminating dark fishing fleets in North Korea. *Sci. Adv.* **2020**, *6*, eabb1197. [\[CrossRef\]](#)
4. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [\[CrossRef\]](#)
5. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [\[CrossRef\]](#)
6. Ho, T.K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844.
7. Friedman, N.; Geiger, D.; Goldszmidt, M. Bayesian network classifiers. *Mach. Learn.* **1997**, *29*, 131–163. [\[CrossRef\]](#)
8. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [\[CrossRef\]](#)
9. Abdal, R.; Qin, Y.; Wonka, P. Image2stylegan: How to embed images into the stylegan latent space? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4432–4441.
10. Yasarla, R.; Sindagi, V.A.; Patel, V.M. Syn2real transfer learning for image deraining using gaussian processes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 2726–2736.
11. Zhong, Y.; Deng, W.; Wang, M.; Hu, J.; Peng, J.; Tao, X.; Huang, Y. Unequal-training for deep face recognition with long-tailed noisy data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7812–7821.
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [\[CrossRef\]](#)
13. Wang, Y.; Wang, C.; Zhang, H. Ship classification in high-resolution SAR images using deep learning of small datasets. *Sensors* **2018**, *18*, 2929. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. Zeng, L.; Zhu, Q.; Lu, D.; Zhang, T.; Wang, H.; Yin, J.; Yang, J. Dual-polarized SAR ship grain classification based on CNN with hybrid channel feature loss. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 4011905. [\[CrossRef\]](#)
16. Zhang, T.; Zhang, X.; Ke, X.; Liu, C.; Xu, X.; Zhan, X.; Wang, C.; Ahmad, I.; Zhou, Y.; Pan, D.; et al. HOG-ShipCLSNet: A novel deep learning network with hog feature fusion for SAR ship classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–22. [\[CrossRef\]](#)
17. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8697–8710.
18. Pham, H.; Guan, M.; Zoph, B.; Le, Q.; Dean, J. Efficient neural architecture search via parameters sharing. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4095–4104.
19. Xie, L.; Yuille, A. Genetic cnn. In Proceedings of the IEEE international Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1379–1388.
20. Real, E.; Aggarwal, A.; Huang, Y.; Le, Q.V. Regularized evolution for image classifier architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4780–4789.
21. Liu, H.; Simonyan, K.; Yang, Y. Darts: Differentiable architecture search. *arXiv* **2018**, arXiv:1806.09055.
22. Xu, Y.; Xie, L.; Zhang, X.; Chen, X.; Qi, G.J.; Tian, Q.; Xiong, H. Pc-darts: Partial channel connections for memory-efficient architecture search. *arXiv* **2019**, arXiv:1907.05737.
23. Courbariaux, M.; Hubara, I.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv* **2016**, arXiv:1602.02830.
24. Rastegari, M.; Ordonez, V.; Redmon, J.; Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 525–542.

25. Liu, Z.; Wu, B.; Luo, W.; Yang, X.; Liu, W.; Cheng, K.T. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 722–737.
26. Liu, Z.; Shen, Z.; Savvides, M.; Cheng, K.T. Reactnet: Towards precise binary neural network with generalized activation functions. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 143–159.
27. Liu, Z.; Shen, Z.; Li, S.; Helweggen, K.; Huang, D.; Cheng, K.T. How do adam and training strategies help bnns optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 6936–6946.
28. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
29. Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; Tran, D. Image transformer. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4055–4064.
30. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
31. Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H. Escaping the big data paradigm with compact transformers. *arXiv* **2021**, arXiv:2104.05704.
32. Yang, Z.; Wang, Y.; Chen, X.; Shi, B.; Xu, C.; Xu, C.; Tian, Q.; Xu, C. Cars: Continuous evolution for efficient neural architecture search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1829–1838.
33. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
34. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 448–456.
35. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 10012–10022.
36. Huang, L.; Liu, B.; Li, B.; Guo, W.; Yu, W.; Zhang, Z.; Yu, W. OpenSARShip: A dataset dedicated to Sentinel-1 ship interpretation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *11*, 195–208. [[CrossRef](#)]
37. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
38. Zhang, T.; Zhang, X. Squeeze-and-excitation Laplacian pyramid network with dual-polarization feature fusion for ship classification in sar images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 4019905. [[CrossRef](#)]
39. Zhang, H.; Tian, X.; Wang, C.; Wu, F.; Zhang, B. Merchant vessel classification based on scattering component analysis for COSMO-SkyMed SAR images. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1275–1279. [[CrossRef](#)]
40. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
42. Hou, X.; Ao, W.; Song, Q.; Lai, J.; Wang, H.; Xu, F. FUSAR-Ship: Building a high-resolution SAR-AIS matchup dataset of Gaofen-3 for ship detection and recognition. *Sci. China Inf. Sci.* **2020**, *63*, 140303. [[CrossRef](#)]
43. Huang, G.; Liu, X.; Hui, J.; Wang, Z.; Zhang, Z. A novel group squeeze excitation sparsely connected convolutional networks for SAR target classification. *Int. J. Remote Sens.* **2019**, *40*, 4346–4360. [[CrossRef](#)]
44. Xiong, G.; Xi, Y.; Chen, D.; Yu, W. Dual-polarization SAR ship target recognition based on mini hourglass region extraction and dual-channel efficient fusion network. *IEEE Access* **2021**, *9*, 29078–29089. [[CrossRef](#)]