Article

Classification of Small Targets on Sea Surface Based on Improved Residual Fusion Network and Complex Time–Frequency Spectra

Shuwen Xu [1],*, Xiaoqing Niu [1], Hongtao Ru [1] and Xiaolong Chen [2]

1   National Key Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China;
    niuxq5261@163.com (X.N.); hongtaoru@stu.xidian.edu.cn (H.R.)
2   Naval Aviation University, Yantai 264001, China; cxlcxl1209@163.com
*   Correspondence: swxu@mail.xidian.edu.cn

Abstract: To address the problem that conventional neural networks trained on radar echo data cannot handle the phase of the echoes, resulting in insufficient information utilization and limited performance in detection and classification, we extend neural networks from the real-valued neural networks to the complex-valued neural networks, presenting a novel algorithm for classifying small sea surface targets. The proposed algorithm leverages an improved residual fusion network and complex time–frequency spectra. Specifically, we augment the Deep Residual Network-50 (ResNet50) with a spatial pyramid pooling (SPP) module to fuse feature maps from different receptive fields. Additionally, we enhance the feature extraction and fusion capabilities by replacing the conventional residual block layer with a multi-branch residual fusion (MBRF) module. Furthermore, we construct a complex time–frequency spectrum dataset based on radar echo data from four different types of sea surface targets. We employ a complex-valued improved residual fusion network for learning and training, ultimately yielding the result of small target classification. By incorporating both the real and imaginary parts of the echoes, the proposed complex-valued improved residual fusion network has the potential to extract more comprehensive features and enhance classification performance. Experimental results demonstrate that the proposed method achieves superior classification performance across various evaluation metrics.

Keywords: target classification; residual network; complex neural networks; time–frequency spectra

1. Introduction

For a considerable period, the detection and classification of small targets on the sea surface have posed significant challenges to maritime detection, primarily attributed to the intricate sea surface environment, the interference of sea clutter, and the faint returns from small targets on the sea. These small targets typically exhibit minimal radar cross section (RCS) and slow movement. In addition, the physical mechanism of sea clutter generation is complex and depends on many factors, resulting in sea clutter exhibiting inhomogeneous non-stationary and non-Gaussian statistical characteristics, which makes it difficult to detect small targets on the sea surface [1]. Conventional algorithms for detection and classification of sea surface targets primarily rely on statistical models, with their efficacy contingent upon the alignment between the established clutter distribution model and realistic clutter nature. While statistical model-based techniques work effectively in certain contexts, the inherent complexity and variability of sea clutter frequently cause a disparity between the established statistical model and the realistic clutter nature, leading to a sharp deterioration in detection performance.

In recent years, deep learning has been developing rapidly [2–4]. With its inherent superiority, deep learning is capable of end-to-end learning, directly from raw radar echo data to the generation of target detection results, and has also been widely applied in the field of radar target detection. The application of artificial intelligence to maritime target detection

techniques was first proposed at the end of the last century. Haykin et al. first proposed a new method for detecting signals in noise in 1995, using neural networks to realize the separation of signals and noise in sea clutter [5]. Subsequently, new artificial intelligence techniques have been continuously introduced, and great progress has been made in the fields of artificial intelligence-based sea target detection [6–9], target classification [10,11], and clutter suppression [12–15].

Target classification is also known as pattern recognition, and common classification algorithms in machine learning are the K-nearest neighbor (KNN) [16], support vector machine (SVM) [17], random forest (RF) [18], and convolutional neural network (CNN). Among these, CNNs stand out for automated feature extraction capabilities. Commonly utilized CNN architectures comprise the LeNet [19], AlexNet [20], Visual Geometry Group (VGG) [21], and Deep Residual Network (ResNet) [22]. In recent years, researchers have integrated CNNs with tasks of maritime target detection and classification, presenting a series of algorithms. In 2019, Mou et al. constructed a dataset of radar plane position indicator (PPI) images and and trained an improved CNN, which successfully verified the feasibility of CNN in maritime target detection [23]. In 2022, Shi et al. demonstrated the successful application of CNNs for the classification of sea clutter and small targets [10]. In 2023, Qu et al. creatively used the time–frequency spectra of radar echoes as the feature inputs, constructed a dataset of radar time–frequency images, and achieved good classification and detection performance with a CNN [24]. In 2023, Xu et al. employed the concept of migration learning and combined a pre-trained CNN and block-whitened time–frequency spectra, achieving an effective classification of different sea targets in the background of strong clutter [25]. It is worth mentioning that, in 2018, Trabelsi et al. proposed the deep complex network (DCPN), which extended the learning range of neural networks from real numbers to complex numbers, and experimentally validated that complex-valued convolutional neural networks (CV-CNN) can achieve better classification performance [26]. This successful attempt paved the way for new applications of CV-CNNs. Scholars have since combined these networks with tasks in their respective fields, demonstrating experimentally that CV-CNNs achieve strong performance in areas such as detection, classification, and clutter suppression. In 2019, Zhang et al. introduced CV-CNNs for the classification of synthetic aperture radar (SAR) images, investigating the impact of various complex-valued activation functions on classifier performance [27]. They also creatively proposed the complex-valued adaptive moment estimation (CV-Adam) optimization algorithm tailored for CV-CNNs. In 2020, Yu et al. introduced a new CV-CNN, the complex-valued full convolutional neural network (CV-FCNN), specifically for SAR image classification. CV-FCNN replaces the pooling and fully connected layers in CV-CNN with convolutional layers, thereby avoiding complex pooling operations and reducing the risk of overfitting, which resulted in high classification accuracy [28]. In 2021, Zhang et al. further advanced SAR image classification by proposing an amplitude–phase-type activation function better suited for CV-CNNs, experimentally demonstrating its superiority over real-valued convolutional neural networks (RV-CNNs) [29]. In 2022, Wang et al. used the complex-valued radar echo signals as inputs, and utilized complex-valued U-Net (CV-UNet) to differentiate between targets and clutter to achieve the suppression of sea clutter, which greatly improved the target detection probability [30]. In 2022, Zhang et al. extended CV-CNNs to the realm of graph neural networks (GNNs), proposing a novel complex-valued graph neural network (CV-GNN) for ISAR (inverse synthetic aperture radar) image classification [31]. Recently, in 2024, Zhou et al. integrated the strengths of complex-valued neural networks with attention mechanisms to perform automatic target recognition for SAR images featuring multi-scale attributes [32]. Theoretically, complex-valued convolutional neural networks offers considerable promise in detection and classification of maritime targets, a field that has hitherto received little attention and that therefore requires further research.

Typically, radar echoes are in the form of complex numbers, containing magnitude and phase information. However, in the current practices of target detection and classification, radar echo data are often processed to be in the form of real numbers for the purpose of

neural network training, which may involve transforming the data into the magnitude spectra, power spectra, or images. Unfortunately, such processes come with the expense of overlooking crucial phase information. Therefore, this paper addresses the issue of inadequate utilization of radar echo data and the difficulty of classifying small maritime targets in complex, non-uniform sea clutter environments. We extend the classification neural network from the real-valued neural network to the complex-valued neural network, introduce and improve a complex-valued residual network, and construct a complex-valued time–frequency spectrum small target classification dataset using four different radar-measured echo data. Based on this, we propose a small maritime target classification algorithm that leverages an improved residual fusion network and complex time–frequency spectra. In this paper, our main innovative work is as follows:

1. The measured radar echo data from four different small targets were collected, and a corresponding complex time–frequency spectrum dataset was constructed for the first time using the short-time Fourier transform (STFT). This dataset will be used in subsequent small maritime target classification experiments. The complex time–frequency spectrum dataset is stored in the form of complex numbers, which preserves the phase information of the radar echoes and is helpful for target classification.
2. Our complex-valued improved residual fusion network is constructed upon the ResNet50. It employs the complex-valued residual unit as the fundamental module, integrating the spatial pyramid pooling (SPP) module for feature fusion across various receptive fields. Furthermore, the conventional residual block layer is replaced with the multi-branch residual fusion (MBRF) module to maximize feature information utilization. Simultaneously, the fully-connected linear classification layer of the network is substituted with two 1×1 convolutional layers. This modification not only reduces the network's parameter count to a certain extent but also can improves classification accuracy.

We conducted simulation experiments with the complex-valued improved residual fusion network using the above complex time–frequency spectrum dataset, and the experimental results show that our proposed improved residual fusion network achieves a significant improvement in all classification performance evaluation metrics.

The structure of this paper is organized as follows. Section 2 introduces the foundational principles of complex-valued classification neural networks and provides detailed descriptions of the component modules of the improved residual fusion network, along with the overall network architecture. In Section 3, we elaborate on the dataset, loss function, network parameter settings, and model evaluation metrics used in this study. Subsequently, we conduct a comparison experiment of real-valued neural network, a comparison experiment of complex-valued neural network, and an ablation experiment to assess the effectiveness and robustness of the proposed classification algorithm. Finally, Section 4 concludes the paper with a summary of our results and a discussion of future work.

2. Classification Network Design

2.1. Overview of Complex-Valued Neural Networks

2.1.1. Complex-Valued Convolution

Usually, the CNN extracts the features we need from input data through multiple convolutional layers for subsequent network detection or classification. In the convolution operation, the convolutional kernel operates on the original input feature map in a sliding window manner to generate the output feature map, as shown in Figure 1.
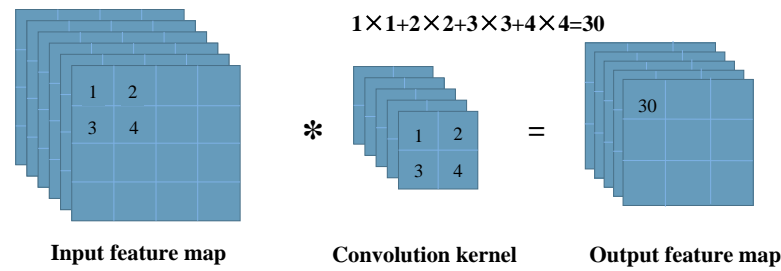
1×1+2×2+3×3+4×4=30

Input feature map   **Convolution kernel**   **Output feature map**

Figure 1. Real-valued convolution operation.

Complex-valued convolution is fundamentally similar to real-valued convolution. However, to handle input feature maps in the form of complex numbers, the convolution kernel in complex-valued convolution also takes on a complex form, comprising real and imaginary parts. The corresponding complex-valued convolution kernel can be represented as $W = a + ib$, where a and b are the real and imaginary parts of the complex-valued convolution kernel, respectively. When performing complex-valued convolution operations, the complex-valued convolution formula can be derived from the principle of complex multiplication:

$$\begin{aligned} Y &= X * W \\ &= (x + iy) * (a + ib) \\ &= (x * a - y * b) + i(x * b + y * a), \end{aligned} \tag{1}$$

where $X = x + iy$ denotes the complex-valued input feature map, $W = a + ib$ denotes the complex convolution kernel, and Y denotes the complex-valued output feature map. From Equation (1), it can be seen that the complex-valued convolution is actually a combination of four independent real-valued convolution operations, and the specific process is clearly represented in Figure 2.
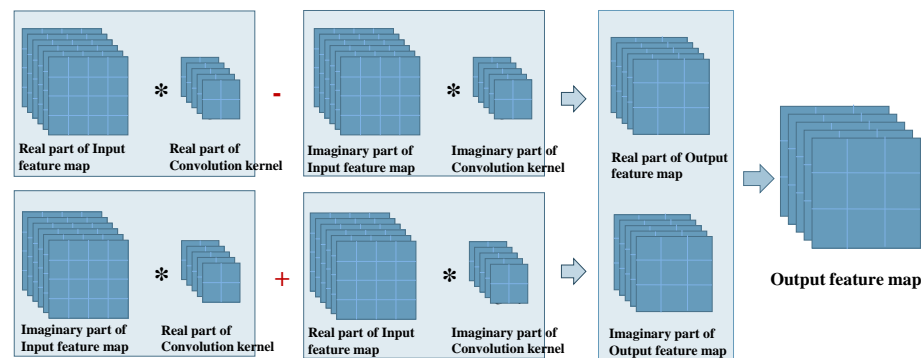

Figure 2. Complex-valued convolution operations.

2.1.2. Complex-Valued Batch Normalization

Batch normalization (BN) is usually used in conjunction with a convolutional layer, which processes the input data so that the output data are all standard normal distributions. This operation has been shown to be effective in achieving better convergence and faster training of the network, in addition to preventing the occurrence of overfitting. For the BN layer of the complex-valued neural network, the batch input dataset is assumed to be $\mathbf{X} : A = \{X_1, X_2, \cdots, X_n\}$, where $X = x + iy$ and n is the size of Batchsize. The BN process also consists of four major steps.

1. Find the mean of the input data, where the expression for the mean solution is

$$E(\mathbf{X}) = \left[\begin{array}{c} E(R(\mathbf{X})) \\ E(I(\mathbf{X})) \end{array} \right] = \left[\begin{array}{c} \frac{1}{n} \sum\limits_{i=1}^{n} R(\mathbf{X}_i) \\ \frac{1}{n} \sum\limits_{i=1}^{n} I(\mathbf{X}_i) \end{array} \right] \tag{2}$$

where $R(\cdot)$ and $I(\cdot)$ denote the operations of extracting the real and imaginary parts of a complex number, respectively, and $E(\cdot)$ denotes the operation of finding the mean.

2. Find the covariance matrix of the input data, which is expressed as

$$\mathbf{V} = \left[\begin{array}{cc} Cov\{R(\mathbf{X}), R(\mathbf{X})\} & Cov\{R(\mathbf{X}), I(\mathbf{X})\} \\ Cov\{I(\mathbf{X}), R(\mathbf{X})\} & Cov\{I(\mathbf{X}), I(\mathbf{X})\} \end{array} \right] \tag{3}$$

where *Cov* denotes the covariance operation.

3. Normalize the input data using the obtained mean and covariance matrices.

$$\widetilde{\mathbf{X}} = \mathbf{V}^{-\frac{1}{2}}(\mathbf{X} - E(\mathbf{X})) \tag{4}$$

4. To enhance the network's expressive capability, perform scaling and shifting transformations on the normalized data to obtain the final output $\widetilde{\mathbf{Y}}$:

$$\widetilde{\mathbf{Y}} = \gamma \widetilde{\mathbf{X}} + \beta \tag{5}$$

where the translation parameter β is initialized with a value of $0 + i0$ and its real and imaginary parts are two trainable parameters; the scale scaling parameter $\gamma = \left[\begin{array}{cc} \gamma_{rr} & \gamma_{ri} \\ \gamma_{ir} & \gamma_{ii} \end{array} \right]$ is initialized with a value of $\gamma = \left[\begin{array}{cc} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{array} \right]$ and it contains three trainable parameters γ_{rr}, γ_{ii}, and γ_{ri}.

2.1.3. Complex-Valued Activation Function

The convolutional operation in neural networks is a linear operation. Therefore, to enhance the network's ability to perform nonlinear fitting during the training and learning process, the involvement of activation functions is typically required to work in tandem. Common neural network activation functions are Sigmoid, Tanh, ReLU, LeakyReLU, and Softmax.

The complex-valued activation function is defined by treating the real and imaginary parts separately and performing the activation operation independently. This can be expressed by

$$\mathbf{y} = H(R(\mathbf{x}) + iI(\mathbf{x})) = H(R(\mathbf{x})) + iH(I(\mathbf{x})) \tag{6}$$

where $H(\cdot)$ represents the activation function used. The activation function used in this paper is the complex ReLU activation function, i.e., $H(\cdot) = \text{ReLU}(\cdot)$.

2.1.4. Complex-Valued Pooling

The pooling layer is a standard component of neural networks, typically employed for feature dimensionality reduction. It decreases the output size of the neural network feature map, enhancing computational efficiency and mitigating overfitting. Unlike convolutional layers, the pooling process lacks learnable parameters and solely executes straightforward operations on the input data. Common pooling methods include maximum pooling and average pooling. The pooling process is illustrated in Figure 3.
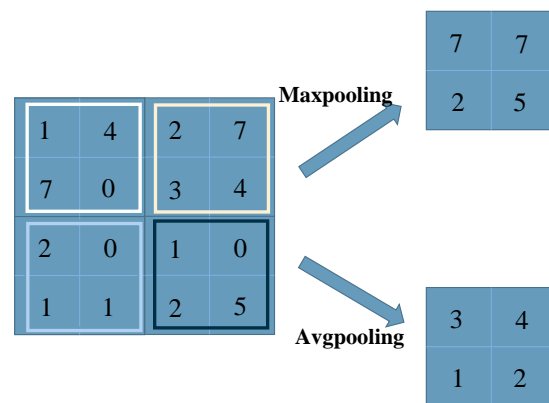
Figure 3. Pooling operation (kernel size = 2).

Similarly, complex-valued pooling is also established on the foundation of real-valued pooling. It divides the complex number into real and imaginary parts, respectively, conducting pooling operations on each part independently before merging them.

2.2. Improved Residual Fusion Network

This subsection provides a detailed description of the complex-valued improved residual fusion network utilized in this paper, which is built upon ResNet50 [22]. ResNet offers significant advantages over mainstream CNN. By employing residual units as the base module, ResNet can construct networks that are deeper than traditional CNNs without encountering the issues of vanishing or exploding gradients. Additionally, ResNet features fewer parameters compared to conventional CNNs. Despite the increased depth of the ResNet, the actual number of parameters does not grow as rapidly due to the presence of residual connections. This characteristic facilitates faster and more efficient training. ResNet primarily aims to address the challenges of gradient vanishing and explosion associated with increasing network depth. Common variants of ResNet include ResNet50 and ResNet101, where the numbers 50 and 101 denote the respective number of layers in the residual network.

The residual network is composed of multiple residual blocks, with the internal structure of each block illustrated in Figure 4. Each residual block includes a shortcut connection that adds the input directly to the block's output. This shortcut ensures that gradients are correctly propagated during backpropagation, preventing issues such as gradient disappearance and explosion that arise from having an excessively deep network. This added branch is referred to as a shortcut. The residual block can be implemented in two forms, as shown in Figure 5: the identity block and the convolution block. In the identity block, the input and output of the residual block are directly summed and merged through the shortcut branch, used when the input and output dimensions are the same. The convolution block, on the other hand, adjusts the dimensions of the input and output through a convolutional layer in the shortcut branch before summing them, used when the input and output dimensions differ. Each residual layer in ResNet includes a convolution block and several identity blocks. Since different residual layers have varying feature dimensions, the convolution block serves to adjust the feature dimensions at the beginning of each residual block to match the input requirements of the specific residual layer.

The ResNet50 network involved in this paper is stacked by four residual block layers, and the number of residual blocks stacked within each residual block layer is 3, 4, 6, and 3 in order. The specific network structure is shown in Figure 5.
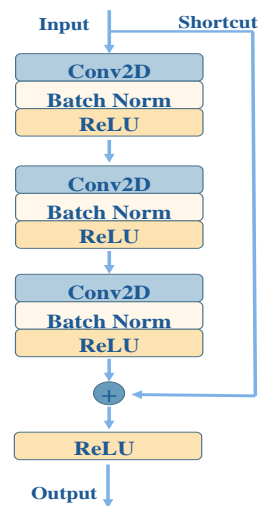
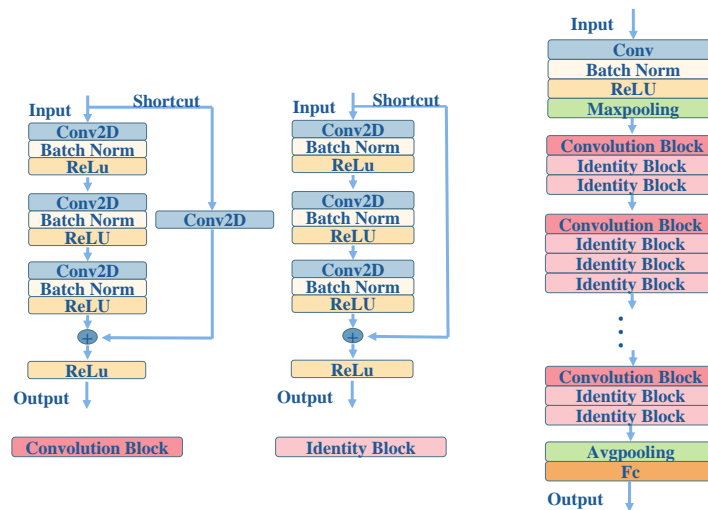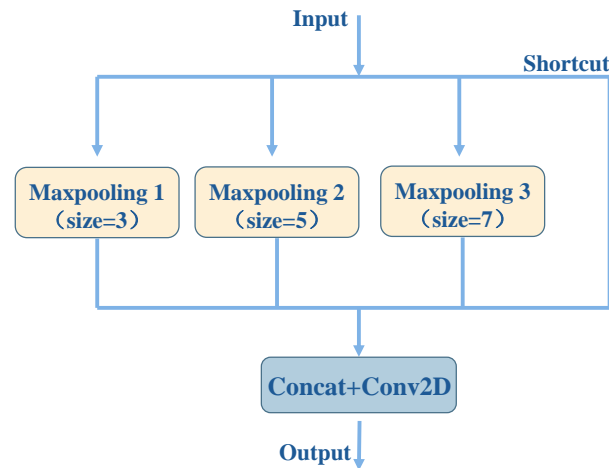**Figure 4.** The structure of the residual block.

Figure 5. The structure of ResNet50.

2.2.1. SPP Module

As previously mentioned, the pooling operation serves the purpose of reducing feature dimensionality and resizing the feature map, thus resulting in less redundant information and better computational efficiency. However, the simple operations of taking the mean or maximum value during pooling often result in the loss of useful information, leading to underutilization of the extracted features. Moreover, radar echoes contain a significant amount of valuable information, especially regarding small targets. Due to their limited receptive field range, small targets are prone to being overlooked in the pooling process, potentially leading to their omission. Alternatively, reducing the size of the pooling kernel to address this issue may result in the loss of feature information redundancy. To resolve these conflicts, we employ the spatial pyramid pooling (SPP) module instead of the conventional pooling layer. This approach aims to enhance feature utilization by fusing feature maps with different receptive fields, thereby optimizing network performance.

As is well known, different sizes of pooling kernels correspond to different receptive fields of the original feature map. A smaller pooling kernel corresponds to a smaller receptive field, enabling the feature maps after dimensionality reduction to retain more detailed information from the original maps. Conversely, a larger pooling kernel corresponds to a larger receptive field, allowing the feature maps after dimensionality reduction to capture

more high-dimensional feature information. The spatial pyramid pooling (SPP) module capitalizes on this principle by generating feature maps across different receptive fields using pooling kernels of various sizes, which are then fused together. The specific structure of the SPP module is depicted in Figure 6. In our experiments, we set the pooling kernel sizes to 1, 3, 5, and 7, respectively. This configuration enables us to capture a diverse range of receptive field sizes, allowing for more effective utilization of feature information.
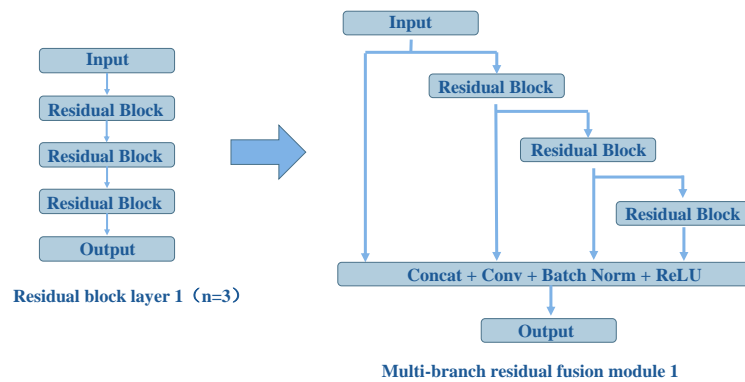

Figure 6. The structure of SPP.

2.2.2. MBRF Module

As depicted in Figure 5, the ResNet50 comprises four residual block layers, connected in series to form the primary structure of the network. Each residual block layer consists of several residual blocks arranged sequentially. Residual networks are renowned for their ease of optimization and their ability to enhance accuracy through appropriate increases in network depth. However, excessively deep networks may result in the loss of important information. The radar echo data, after undergoing deep convolution, yield higher-dimensional feature information. However, for the classification of small targets on the sea surface, the underlying details of the feature information are crucial and cannot be overlooked. Achieving a better balance between network depth and the transfer of information across shallow layers is essential. Therefore, we propose to laterally increase the network width without altering the network depth, thereby broadening the branch of feature information transmission. By establishing the multi-branch residual fusion module, we aim to enhance the utilization rate of feature information, thereby optimizing network performance.

The MBRF module presents a denser residual structure compared to the common residual module, transforming the internal residual network into a series–parallel structure. The residual blocks between layers remain connected in a sequential manner. However, within each residual block layer, the residual blocks are no longer simply connected in series. Instead, the output of each residual block is used as a skip connection to the end of that residual block layer. These feature maps are then fused together to form the output of the residual block layer, which is passed as input to the next residual block layer. The specific network structure of the MBRF module is illustrated in Figure 7, where only the first residual block layer is taken as an example. In addition to considering the output of each residual block separately as a parallel branch, we incorporate an additional original input branch as a large residual connection linked to the final output of the network. This consolidation can enhance feature fusion effects. Therefore, the final output of each residual block layer integrates multi-scale feature information, significantly reducing the loss of shallow details and thereby enhancing the utilization of small target features. This improvement plays a critical role in small target classification tasks.

Figure 7. The structure of MBRF (residual block refers to the convolution block and identity block mentioned in Figure 5).

2.2.3. Overall Network Architecture

In general, the sea surface environment is complex, and the radar echo data carry a lot of sea clutter information. When the target is small or far away, the echo signal of the target is weak and can easily be lost in the clutter signal, which brings great difficulties to both detection and classification. As a matter of fact, the problem of classifying small maritime targets is more difficult compared to other types of targets because small maritime targets can be caused by Doppler aliasing with sea clutter. Therefore, our method in this paper chooses to utilize the time–frequency ridge differences to classify the targets and further improves the classification performance through the proposed network.

Deep learning relies on learning the features extracted by neural networks to perform detection and classification tasks, so the performance of the network is closely related to the completeness of the feature information. Increasing the number of layers in the network can extract higher dimensional feature information and improve network performance, but at the cost of losing detailed features, resulting in missed detection or misclassification of small targets. This issue is particularly pronounced in sea surface target detection and classification, where, when the signal-to-clutter ratio is low, the distinguishing features between the target and clutter become less apparent, resulting in poor performance in detection and classification. To address this issue, we propose a complex-valued improved ResNet, which has shown significant performance improvements in the classification of small sea surface targets.

First, most neural networks are real-valued, requiring radar echo data to be converted into amplitude or power values for training and learning. This conversion results in the loss of phase information carried in the echo data, leading to underutilization of the data. To overcome this, we extend real-valued networks to complex-valued networks, allowing the original echo data to be directly input into the network for training and learning after preprocessing. The complex-valued networks can simultaneously learn amplitude and phase information, making them highly suitable for sea surface target detection and classification.

Second, our proposed improved network model builds on ResNet50 and introduces the MRBF and SPP modules. Compared to the original ResNet, MRBF adds numerous skip connections in the feature extraction module. This transforms the originally serial connections into a combination of serial and parallel connections, allowing the network to increase its width without reducing its depth. This structural improvement can retain more detailed features and reduce the possibility of losing detailed feature information, which is often the key to correctly classifying small surface targets, making the improved ResNet more conducive to detecting and classifying small surface targets.

In summary, this paper combines the SPP module and the MBRF module with the ResNet50 network to construct our improved residual fusion network. The network is

shown in Figure 8, and the parameter settings, inputs, and outputs of each layer of the network are detailed in Table 1.
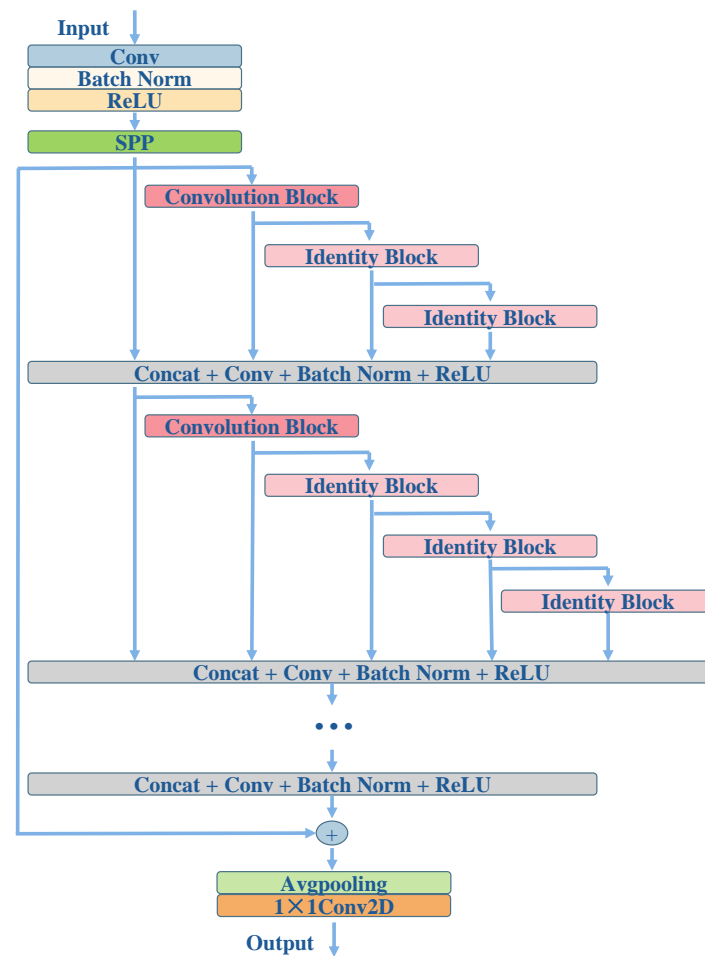

**Figure 8.** Overall architecture of the improved residual fusion network.

Table 1. Composition of each layer of the improved residual fusion network and feature map output size.

Layer Name	Component	Output Size (Batch Size, Number of Channels, Feature Map Size)
Input		$32 \times 1 \times 96 \times 64$
Convolution layer 1	Complex convolution (kernel size = 7)	$32 \times 64 \times 48 \times 32$
SPP module	Complex maxpooling (kernel size = 3, 5, 7)	$32 \times 64 \times 24 \times 16$
MBRF module 1	Complex residual block × 3	$32 \times 256 \times 24 \times 16$
MBRF module 2	Complex residual block × 4	$32 \times 512 \times 12 \times 8$
MBRF module 3	Complex residual block × 6	$32 \times 1024 \times 6 \times 4$
MBRF module 4	Complex residual block × 3	$32 \times 2048 \times 3 \times 2$
Pooling	Complex avgpooling (output size = 1)	$32 \times 2048 \times 1 \times 1$
Convolution layer 2	Complex convolution (kernel size = 1)	$32 \times 4 \times 1 \times 1$
Output		$32 \times 4 \times 1 \times 1$

3. Experimental Design and Analysis of Results

In our experiments, we first conduct target detection on the radar echo data to identify the distance unit where the target is located. Subsequently, we construct our complex time–frequency spectrum dataset based on the acquired target echo data using time–frequency transformation. Finally, we assign labels to the dataset and input it into the improved

complex residual fusion network for training, thereby obtaining the final classification results. The overall flow of the experiment is illustrated in Figure 9. In this subsection, we will provide a detailed description of the dataset used for the experiment, the loss function employed for neural network training, the parameter design of the network model, and an analysis of the experimental results.
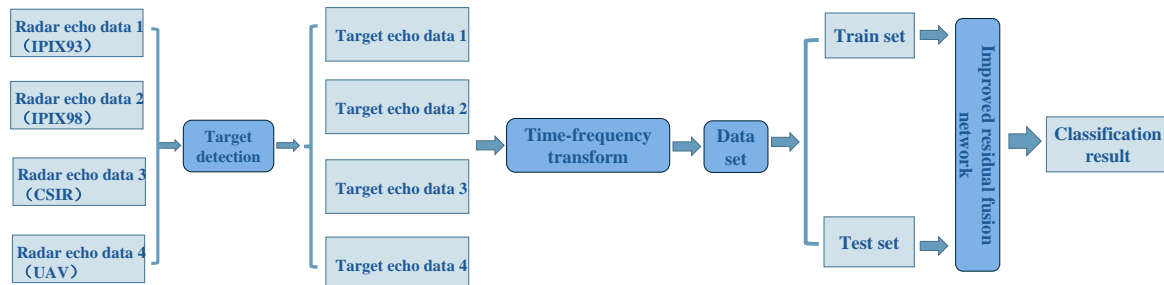
Figure 9. Overall flow chart of the experiment.
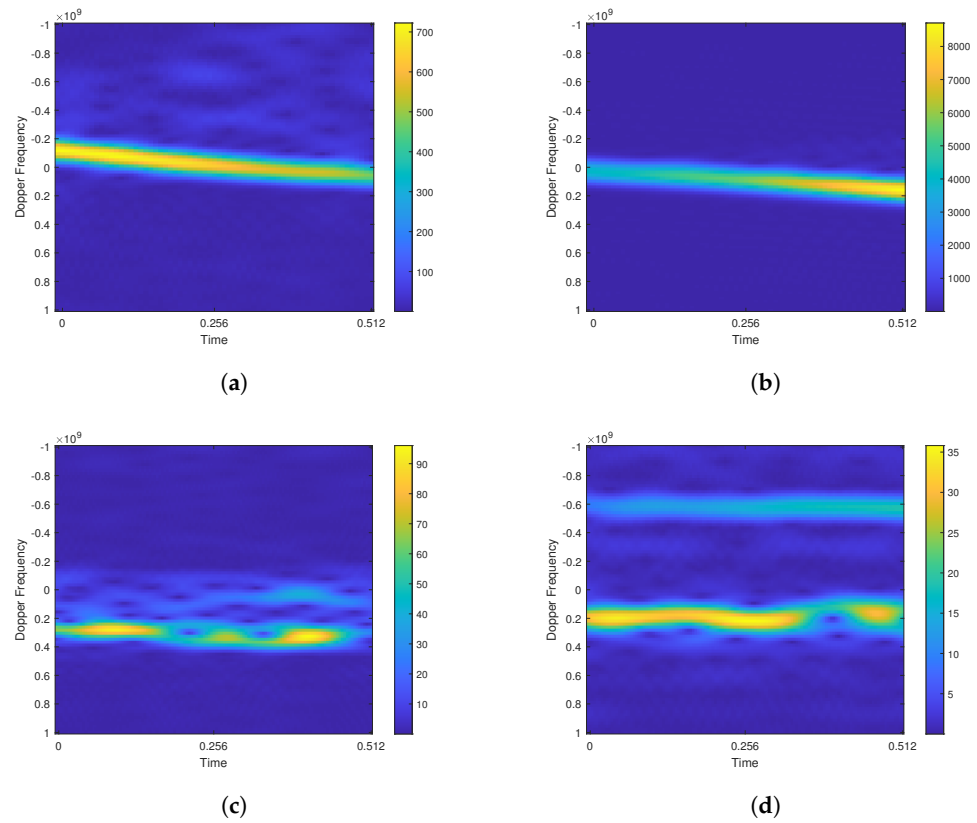
3.1. Dataset

In this paper, we opt for the time–frequency spectrum as the input data for target classification, as it amalgamates the strengths of both the time and frequency domains. Moreover, to preserve the original phase information of the data, we forgo constructing a time–frequency images dataset and directly create the dataset in the form of complex numbers using the original time–frequency spectrum data. The time–frequency spectra of the radar echoes, post time–frequency transformation, are directly input into the complex-valued neural network in the form of a complex matrix for feature extraction and classification purposes.

3.1.1. Description of Data Sources

In constructing the target classification dataset, we select four types of representative radar-measured echo data of small targets on the sea surface for our experiment, namely floating orbs on the sea surface, floating fishing boats, speedboats, and unmanned aerial vehicles (UAVs). The first two types of target data are sourced from radar echo data collected by a team from McMaster University, Canada, using an IPIX (Intelligent Pixel processing X-band) radar in 1993 and 1998, respectively. The third type of target data was collected by the Fynmeet radar, situated on the west coast of South Africa, in 2006. Lastly, the fourth type of sea surface low-altitude UAV target data was collected by a radar located on Lingshan Island, operating at X-band, in dwell mode, and in VV polarization, at an altitude of approximately 430 m.

3.1.2. Data Set Construction and Data Preprocessing

The original radar echo signal comprises a two-dimensional distance pulse matrix, where small targets typically occupy only one distance unit. Thus, we initially conduct target detection on the radar echo to identify the distance unit where the target is located. Subsequently, the echo data from this distance unit is extracted individually as the target echo signal. To construct the time–frequency spectrum dataset, we perform time–frequency transformation on the target echo signal using the short-time Fourier transform (STFT). The STFT yields complex-valued data, containing both the time and frequency features of the target. The time–frequency transformation converts the one-dimensional target echo sequence into a two-dimensional time–frequency spectrum, facilitating preliminary feature extraction. Schematics of the time–frequency spectra for the four types of small targets in the dataset used in this experiment are illustrated in Figure 10. It is intuitively clear from the figure that the fourth type of targets has the most distinctive features, whereas the first and second categories of targets show similar features on the time–frequency spectra, which are more difficult to distinguish.

Figure 10. Time–frequency spectrums of four types of small surface targets: (**a**) Class 1 (IPIX 93); (**b**) Class 2 (IPIX 98); (**c**) Class 3 (CSIR); (**d**) Class 4 (UAV)

The complex time–frequency spectrum dataset constructed encompasses four types of target data, with the number of samples for each type and their respective dataset divisions listed in Table 2. To ensure smooth convergence of the network during training and to expedite the process, the complex time–frequency spectrum data undergo normalization before being input into the network for training.

Table 2. Data sample size and dataset division.

Target Class	Total Number of Samples	Train Set	Test Set
Target 1	8074	6257	1817
Target 2	4585	3685	900
Target 3	7785	5812	1973
Target 4	7857	5892	1965

3.2. Neural Network Parameterization

3.2.1. Loss Function

The cross-entropy loss is a commonly used loss function for target classification, expressed by

$$CELoss = -\sum_{i=1}^{n} \log(p_i) \tag{7}$$

where $\gamma \geq 0$, n is the number of samples, and p_i is the probability that sample i belongs to the correct category i. The focal loss used in this paper is improved on the basis of the cross-entropy loss, and its formula is

$$FocalLoss = -\sum_{i=1}^{n} (1 - p_i)^{\gamma} \log(p_i) \tag{8}$$

The focal loss function is based on the cross-entropy loss function with the addition of the adjustment factor $(1 - p_i)^\gamma$. The higher the confidence score of the sample through the network output, the lower the value of the corresponding adjustment factor. The purpose of this operation is to reduce the weight of easy-to-categorize samples in the training process, so that the network is more inclined to the training of difficult-to-categorize samples in the training process, thus improving the network classification performance. In the paper published in 2018, Kaiming He's team first proposed the focal loss function, and they experimentally verified that their network works best when the value of γ is taken as 2 [33]. In the experiments conducted in this paper, various values were also set for training. The final experimental results indicate that our network achieves the best classification performance when the value of γ is set to 1. These experimental results are presented in Table 3. Because the research in this paper focuses on complex-valued residual networks, the experiments in the following table are all done on the complex-valued ResNet50.

Table 3. Comparison of the classification performance under different values of γ.

γ	Classification Performance	
	Accuracy	**Kappa Coefficient**
0.5	0.927	0.905
1	0.937	0.913
1.5	0.921	0.902
2	0.913	0.881
2.5	0.901	0.864
3	0.898	0.857

3.2.2. Training Parameter Design

In this paper, the network is built under the pytorch framework for experimental validation, and the hardware configuration of the experiment is as follows: the CPU is 12th Gen Intel(R) Core(TM) i5-12400F, the GPU is NVIDIA GeForce RTX 3060, and the running memory is 32GB. The rest of the relevant parameter settings for the training and simulation experiments are shown in Table 4.

Table 4. Parameter settings for simulation experiments.

Parameters	Values
Epoch	50
Batch size	32
Learning rate	0.0001
Optimizer	Adam

3.3. Experimental Results and Analysis

In this subsection, we validate the performance of the proposed improved residual fusion network using the measured radar echoes. Initially, we introduce some evaluation metrics used to measure the target classification results. Subsequently, we set up comparison experiments between the real-valued neural network and the complex-valued neural network, as well as the ablation experiment of the improved residual fusion network. Finally, we conduct a comprehensive evaluation and analysis based on the results obtained from the experiments.

3.3.1. Evaluation Indicators

The common evaluation metrics for classification problems include accuracy, precision, recall, and $F1$ score. In this paper, we also introduce the Kappa coefficient as a evaluation metric to address potential network model bias resulting from imbalanced proportions of positive and negative samples. Before introducing the aforementioned evaluation indices, we first explain the concept of a confusion matrix. The confusion matrix is a metric used to

assess the performance of a model, primarily employed in judging the performance of a classifier. For a multi-classification problem, the confusion matrix is defined as shown in Table 5 (taking the four-classification task involved in this experiment as an example and Class 1, Class 2, Class 3, and Class 4 represent the four types of sea targets mentioned in Section 3.1: floating orbs, floating fishing boats, speedboats, and UAVs, respectively), where TP_i indicates that the sample of class i is correctly classified as class i, and F_iP_j indicates that the sample of class i is incorrectly classified as class j.

Table 5. Multi-classification confusion matrix.

		Prediction			
		Class 1	**Class 2**	**Class 3**	**Class 4**
	Class 1	TP_1	F_1P_2	F_1P_3	F_1P_4
True	Class 2	F_2P_1	TP_2	F_2P_3	F_2P_4
	Class 3	F_3P_1	F_3P_2	TP_3	F_3P_4
	Class 4	F_4P_1	F_4P_2	F_4P_3	TP_4

1. Accuracy, Precision and Recall;

Accuracy is defined as the proportion of results predicted by the model as positive samples to the total number of samples, serving as an evaluation of the overall performance of the classifier. In a multi-classification problem, when calculating the accuracy of a category, the samples of this category are considered as positive samples, whereas the rest of the samples are treated as negative samples. The rest of the indicators in the calculation of positive and negative sample definition are similar to this. The accuracy can be expressed as

$$Accuracy = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{j=1,j\neq i}^{n} F_jP_i} \tag{9}$$

Precision is defined as the proportion of all samples that are actually categorized correctly out of those predicted by the model to be positive. Distinguished from the accuracy, the precision is an evaluation index for the classification results of a certain category. The precision can be written as

$$Precision = \frac{TP_i}{TP_i + \sum_{j=1,j\neq i}^{n} F_jP_i} \tag{10}$$

Recall is defined as the proportion of all positive samples for which the model predicts correctly, and is also specific to a certain category. The recall can be expressed as

$$Recall = \frac{TP_i}{TP_i + \sum_{j=1,j\neq i}^{n} F_iP_j} \tag{11}$$

The overall precision and recall in a multicategorization task refer to the average precision and average recall across all categories.

2. $F1$ score;

The F1 score is a comprehensive evaluation metric that jointly considers precision and recall, calculated based on precision and recall. The F1 score is represented as

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

3. Kappa Coefficient.

In classification tasks where there is an imbalance in the proportion of samples in the dataset, such as a ratio of positive to negative samples of 1:9, the model may achieve a high accuracy by correctly categorizing negative samples while incorrectly categorizing positive samples. However, in reality, the positive samples are not correctly recalled at all. To address this issue, we need to introduce an evaluation metric that penalizes the model for its "bias" to supplement the lack of accuracy and enhance the overall performance of the model. We call this the Kappa coefficient, which is expressed as

$$Kappa = \frac{p_0 - p_e}{1 - p_e} \tag{13}$$

Here, $p_0 = Accuracy$, $p_e = \dfrac{\sum\limits_{i=1}^{n} \left(TP_i + \sum\limits_{j=1, j\neq i}^{n} F_j P_i \right) \times \left(TP_i + \sum\limits_{j=1, j\neq i}^{n} F_i P_j \right)}{\left(\sum\limits_{i=1}^{n} \left(TP_i + \sum\limits_{j=1, j\neq i}^{n} F_j P_i \right) \right)^2}$. According to Equation (13), the more unbalanced the confusion matrix is, the lower the kappa coefficient will be, which represents stronger "bias" of the model and poorer accuracy of inaccurate the classification results.

### 3.3.2. Comparison Experiments, Ablation Experiments, and Analysis of Experimental Results

1. Comparison Experiments

To validate the effectiveness of our structural improvements and the advantages of extending real-valued neural networks to complex-valued ones, we conducted a series of comparative experiments on the classification performance of both real-valued and complex-valued neural networks. The CNNs used in these experiments include AlexNet [20], VGG16 [21], MobileNet [34], CNN-LSTM [35], FCNN [28], ResNet50 [22], and our proposed Im-ResNet50. The corresponding complex-valued neural networks were built on the same underlying network architectures. The training data were sourced from the complex time–frequency spectrogram dataset we constructed. For the complex-valued neural networks, the complex time–frequency spectrum matrix is normalized and input into the networks for training. For the real-valued neural networks, the complex time–frequency spectrum matrix is first modulated, then normalized, and subsequently fed into the networks for training. Furthermore, the experimental settings of hyperparameter, training loss function, and experimental environment are kept consistent across all networks.

First, we conducted the comparison experiment under the real-valued neural network, the results of which are shown in Figure 11 and Table 6.

Table 6. Comparison of the performance of different real-valued classification networks.

	Accuracy	Precision	Recall	*F*1 Score	Kappa Coefficient
AlexNet	0.868	0.825	0.829	0.824	0.821
VGG16	0.878	0.844	0.845	0.842	0.835
MobileNet	0.879	0.864	0.853	0.852	0.837
CNN-LSTM	0.876	0.818	0.832	0.822	0.830
FCNN	0.876	0.822	0.865	0.830	0.829
ResNet50	0.888	0.863	0.857	0.856	0.848
Improved ResNet50	0.893	0.874	0.866	0.868	0.855
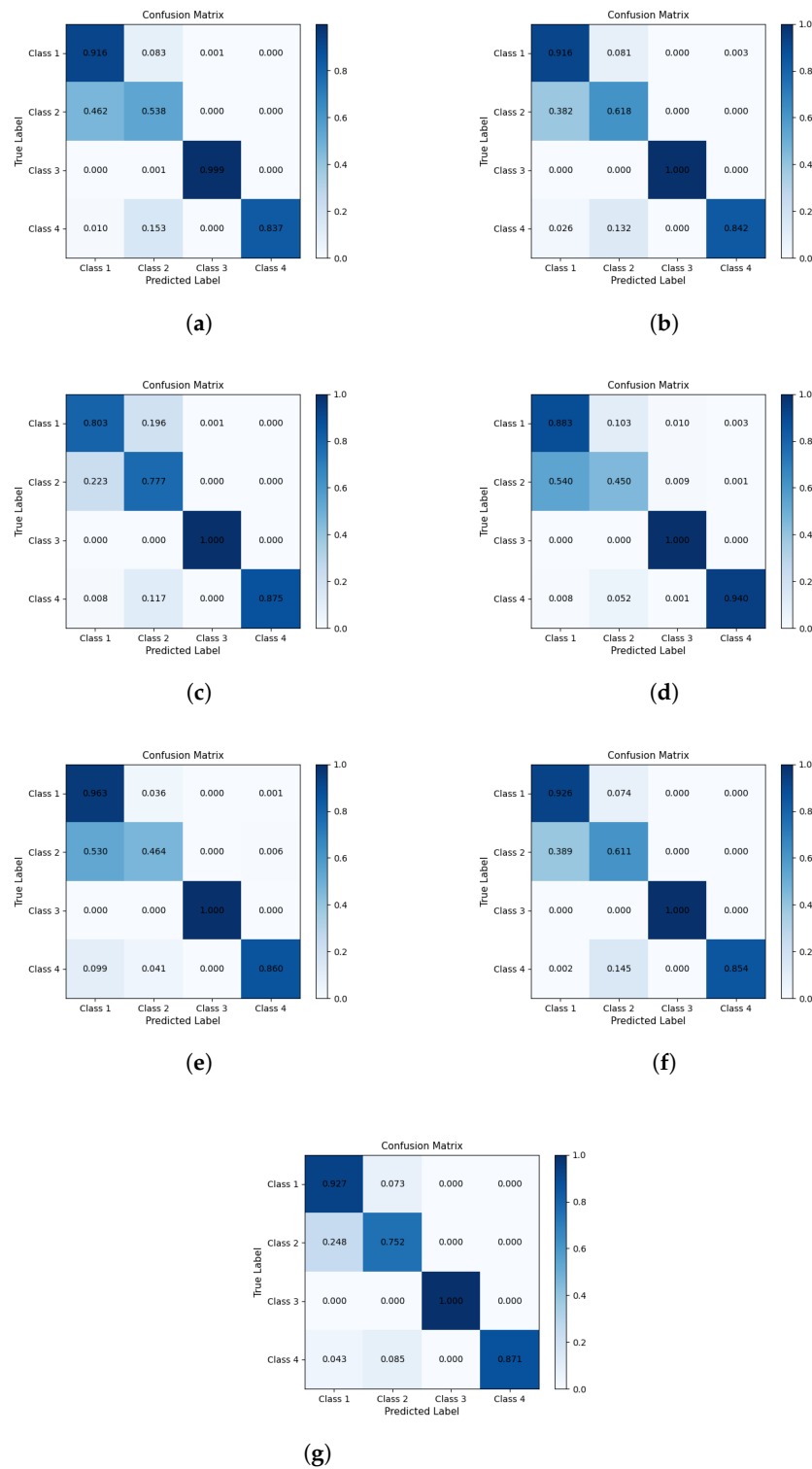
Figure 11. Confusion matrix for different real-valued classification networks: (**a**) AlexNet; (**b**) VGG16; (**c**) MobileNet; (**d**) CNN-LSTM; (**e**) FCNN; (**f**) ResNet; (**g**) Improved ResNet.

As shown in the confusion matrices in Figure 11, overall, the probability of correctly classifying Class 1 and Class 3 targets is higher, whereas Class 2 and Class 4 targets are relatively more difficult to classify accurately. The experimental results in Table 6 indicate that our Im-ResNet outperforms the original ResNet50 across all evaluation metrics. This improvement is attributed to the introduction of the MRBF and SPP modules, which enhance feature reuse within the network and increase the classification accuracy for all target

types, particularly those that are more difficult to classify. Additionally, compared to other real-valued classification neural networks, our Im-ResNet demonstrates superior classification performance, further validating the effectiveness of the structural enhancements we proposed for Im-ResNet.

Next is the comparison experiment using the complex-valued neural networks, the results of which are shown in Figure 12 and Table 7.

Table 7. Comparison of the performance of different complex-valued classification networks.

	Accuracy	Precision	Recall	*F*1 Score	Kappa Coefficient
CV-AlexNet	0.872	0.815	0.828	0.818	0.825
CV-VGG16	0.899	0.870	0.870	0.868	0.864
CV-MobileNet	0.892	0.841	0.858	0.846	0.852
CV-CNN-LSTM	0.885	0.829	0.846	0.838	0.842
CV-FCNN	0.893	0.853	0.859	0.854	0.854
CV-ResNet50	0.912	0.858	0.890	0.863	0.878
CV-Im-ResNet50	0.940	0.913	0.918	0.915	0.917

Comparing the confusion matrices in Figures 11 and 12, it is evident that complex-valued neural networks perform better overall than real-valued neural networks in classifying the more challenging second and fourth types of targets. The experimental results in Tables 6 and 7 further demonstrate that, under the same network's structure, complex-valued neural networks achieve superior classification performance compared to real-valued neural networks, with improvements across all evaluation metrics. This is due to the ability of complex-valued neural networks to fully leverage both amplitude and phase information, thereby enhancing the utilization of feature information. Additionally, from the classification results of all complex-valued neural networks presented in Table 7, our proposed CV-Im-ResNet outperforms the other complex-valued networks across all evaluation metrics, further confirming the effectiveness and superiority of the proposed CV-Im-ResNet.

Finally, combining the experimental results in Tables 6 and 7, all the improvements we made to the original ResNet50 contribute to the improvement of the classification performance, and the proposed CV-Im-ResNet has the optimal classification performance, in which the classification accuracy is improved from 88.8% to 94%, and the Kappa coefficient also increases by about 7 percentage points. This result indicates that our proposed CV-Im-ResNet has better robustness and superiority.

2. Ablation Experiments with the Improved Residual Fusion Network

To validate the effectiveness and rationality of each module in our proposed improved residual fusion network, we conducted ablation experiments to verify their contributions. These experiments involve retaining only the SPP module, the MBRF module, or the 1×1 convolutional classification module, individually, while comparing their classification performance against the original ResNet50. Since this study is based on the complex time-frequency spectrum dataset, all experiments are conducted on the complex neural network. Table 8 presents the results of the ablation experiments for various evaluation metrics.

Table 8. Results of ablation experiments.

	Accuracy	Precision	Recall	*F*1 Score	Kappa Coefficient
CV-ResNet50	0.912	0.858	0.890	0.863	0.878
CV-ResNet50 + SPP	0.934	0.899	0.915	0.905	0.909
CV-ResNet50 + MBRF	0.936	0.898	0.913	0.908	0.911
CV-ResNet50 + 1×1 Conv Classification	0.920	0.882	0.905	0.895	0.892
CV-Im-ResNet	0.940	0.913	0.918	0.915	0.917
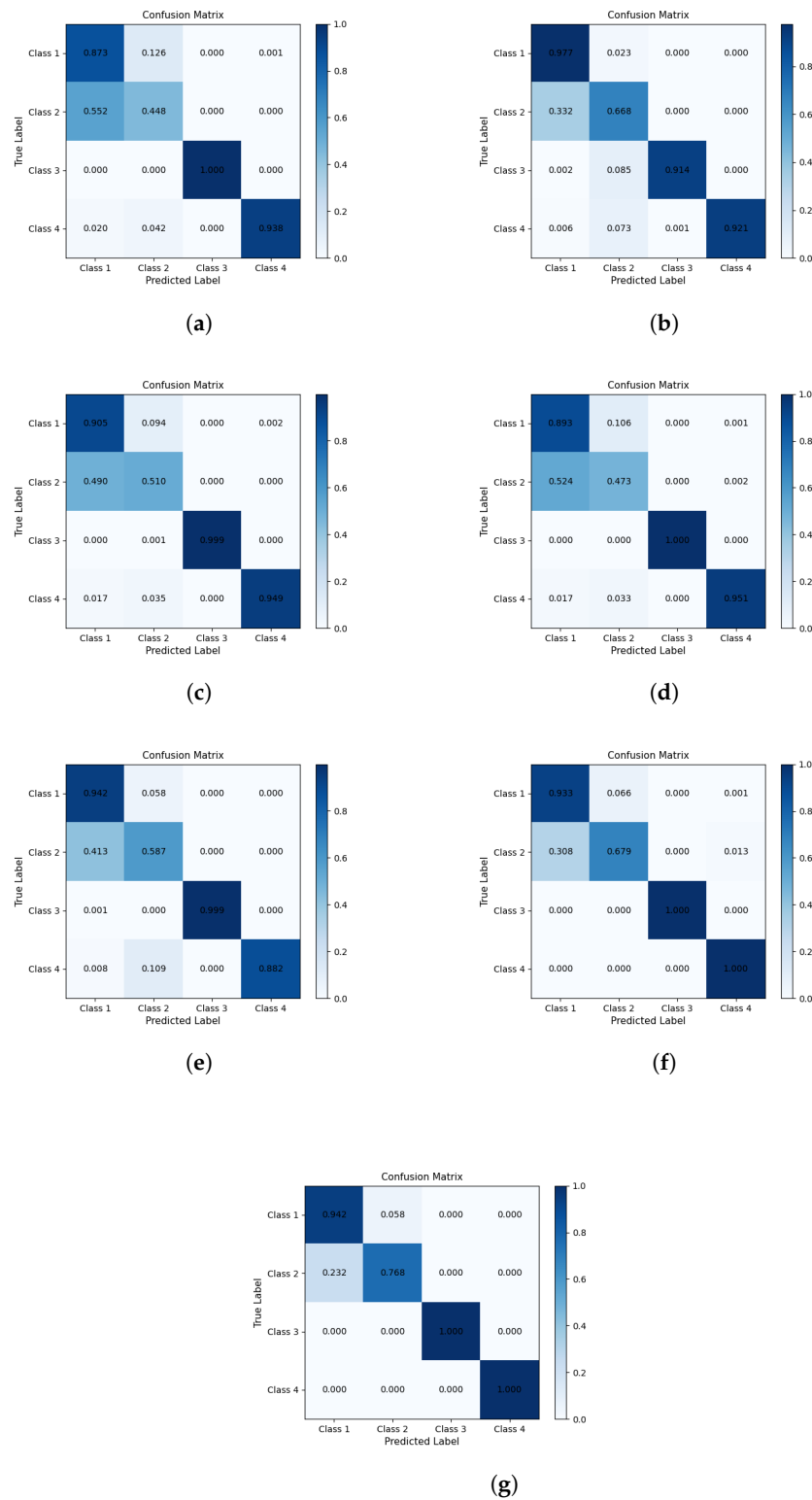
Figure 12. Confusion matrix for different complex-valued classification networks: (**a**) CV-AlexNet; (**b**) CV-VGG16; (**c**) CV-MobileNet; (**d**) CV-CNN-LSTM; (**e**) CV-FCNN; (**f**) CV-ResNet; (**g**) CV-Im-ResNet.

From Table 8, it can be observed that the introduction of each individual module contributes to the improvement of the network's classification performance. The network with the added SPP module shows improvements in all classification performance metrics compared to the original network. This is because the SPP module can simultaneously consider feature

information from different receptive fields, alleviating the problem of small target information loss caused by deep convolutional networks. The introduction of the MBRF module also significantly improves the network's classification performance, with particularly notable enhancements in accuracy, F1 score, and Kappa coefficient, making it the most outstanding among all individual modules. This is due to the fact that the MBRF module widens the network's width laterally without changing its depth, and the jump connections across different feature dimensions ensure the completeness of feature information during forward propagation, thus improving feature utilization. Additionally, the 1×1 convolutional classification module also contributes to the improvement in classification performance. Although its impact on classification performance is not as pronounced as the first two modules, its main role is to reduce the network's parameter compared to the original fully connected linear classification layer, thereby improving the training speed. Finally, the CV-Im-ResNet, which integrates all modules, performs the best in all experiments, achieving the highest values for all evaluation metrics. This indicates that the introduction of each module contributes to the network's classification performance, and that the collective effects of all these modules result in a higher level of overall classification performance of the network.

4. Conclusions

To address the challenges posed by the insufficient utilization of radar echo data in complex and non-uniform sea clutter backgrounds, as well as the difficulty in classifying small targets on the sea surface, we propose a classification algorithm for small sea surface targets based on an improved residual fusion network and complex-valued time–frequency spectra. This algorithm introduces SPP and MBRF modules on top of the ResNet50, enabling more effective feature fusion and utilization. Additionally, by combining the complex-valued time–frequency spectrum dataset with complex-valued neural networks, we fully exploit both the magnitude and phase information in the data, thereby enhancing the performance of the small target classifier. Our experimental results also demonstrate that the proposed CV-Im-ResNet50 improves classification accuracy from 88.8% to 94% compared to the original ResNet50, with the Kappa coefficient increasing by 7 percentage points. This not only validates the effectiveness of the network's structural enhancements but also highlights the strong suitability and excellent generalization capability of complex-valued neural networks for sea surface target detection tasks. However, the proposed CV-Im-ResNet inevitably faces challenges, such as increased network width and longer training time, due to the introduction of the complex-valued convolution. Future work will focus on optimizing the network structure to obtain a more lightweight classification network without sacrificing classification performance.

Author Contributions: Conceptualization, S.X. and X.N.; methodology, X.N. and H.R.; software, X.N.; validation and formal analysis, X.N. and H.R.; writing—original draft preparation, X.N.; writing—review and editing, S.X.; supervision, S.X. and X.C.; project administration, S.X.; funding acquisition, S.X. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The partial original data presented in the study are openly available at http://soma.McMaster.ca/ipix (accessed on 3 April 2023) and https://researchspace.csir.co.za (accessed on 13 December 2018).

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

References

1. Xu, S.; Bai, X.; Guo, Z.; Shui, P. Status and prospects of feature-based detection methods for floating targets on the sea surface. *J. Radars* **2020**, *9*, 684–714.
2. Dong, Y.N.; Liang, G.S. Research and discussion on image recognition and classification algorithm based on deep learning. In Proceedings of the 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 8–10 November 2019; IEEE: Piscataway, NJ, USA, 2019 ; pp. 274–278.

3. Wang, P. Research and design of smart home speech recognition system based on deep learning. In Proceedings of the 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, 10–12 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 218–221.

4. Kounte, M.R.; Tripathy, P.K.; Pramod, P.; Bajpai, H. Analysis of Intelligent Machines using Deep learning and Natural Language Processing. In Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), Tirunelveli, India, 15–17 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 956–960.

5. Haykin, S.; Li, X.B. Detection of signals in chaos. *Proc. IEEE* **1995**, *83*, 95–122. [CrossRef]

6. Guo, Z.X.; Shui, P.L. Anomaly based sea-surface small target detection using K-nearest neighbor classification. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4947–4964. [CrossRef]

7. Wang, J.; Li, S. Maritime radar target detection in sea clutter based on CNN with dual-perspective attention. *IEEE Geosci. Remote Sens. Lett.* **2022**, *20*, 1–5. [CrossRef]

8. Li, Y.; Xie, P.; Tang, Z.; Jiang, T.; Qi, P. SVM-based sea-surface small target detection: A false-alarm-rate-controllable approach. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1225–1229. [CrossRef]

9. Metcalf, J.; Blunt, S.D.; Himed, B. A machine learning approach to cognitive radar detection. In Proceedings of the 2015 IEEE Radar Conference (RadarCon), Arlington, VA, USA, 10–15 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1405–1411.

10. Shi, Y.; Guo, Y.; Yao, T.; Liu, Z. Sea-surface small floating target recurrence plots FAC classification based on CNN. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5115713. [CrossRef]

11. Zhou, Q.; Xu, H.; Wang, Z.; Zhang, Z.; Zhang, X. Radar Sea Clutter Feature Classification Based on Machine Learning. In Proceedings of the 2022 3rd China International SAR Symposium (CISS), Shanghai, China, 2–4 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–7.

12. Wu, Z.; Pei, J.; Huo, W.; Huang, Y.; Zhang, Y.; Yang, H. A machine learning approach to clutter suppression for marine surveillance radar. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 3137–3140.

13. Callaghan, D.; Burger, J.; Mishra, A.K. A machine learning approach to radar sea clutter suppression. In Proceedings of the 2017 IEEE Radar Conference (RadarConf), Seattle, WA, USA, 8–12 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1222–1227.

14. Tang, X.; Li, D.; Cheng, W.; Su, J.; Wan, J. A novel sea clutter suppression method based on deep learning with exploiting time-frequency features. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021, IEEE: Piscataway, NJ, USA, 2021; Volume 5, pp. 2548–2552.

15. Mou, X.; Chen, X.; Guan, J.; Dong, Y.; Liu, N. Sea clutter suppression for radar PPI images based on SCS-GAN. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 1886–1890. [CrossRef]

16. Abeywickrama, T.; Cheema, M.A.; Taniar, D. K-nearest neighbors on road networks: A journey in experimentation and in-memory implementation. *arXiv* **2016**, arXiv:1601.01549. [CrossRef]

17. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]

18. Rigatti, S.J. Random forest. *J. Insur. Med.* **2017**, *47*, 31–39. [CrossRef] [PubMed]

19. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]

20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

21. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

23. Mou, X.; Chen, X.; Guan, J.; Chen, B.; Dong, Y. Marine target detection based on improved faster R-CNN for navigation radar PPI images. In Proceedings of the 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), Chengdu, China, 23–26 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.

24. Qu, Q.; Liu, W.; Wang, J.; Li, B.; Liu, N.; Wang, Y.L. Enhanced CNN-based small target detection in sea clutter with controllable false alarm. *IEEE Sensors J.* **2023**, *23*, 10193–10205. [CrossRef]

25. Xu, S.; Ru, H.; Li, D.; Shui, P.; Xue, J. Marine radar small target classification based on block-whitened time–frequency spectrogram and pre-trained CNN. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5101311. [CrossRef]

26. Trabelsi, C.; Bilaniuk, O.; Zhang, Y.; Serdyuk, D.; Subramanian, S.; Santos, J.F.; Mehri, S.; Rostamzadeh, N.; Bengio, Y.; Pal, C.J. Deep complex networks. *arXiv* **2017**, arXiv:1705.09792.

27. Zhang, Y.; Hua, Q.; Xu, D.; Li, H.; Mu, H. A Complex-Valued Convolutional Neural Network with Different Activation Functions in Polarimetric SAR Image Classification. In Proceedings of the 2019 International Radar Conference (RADAR), Toulon, France, 23–27 September 2019, IEEE: Piscataway, NJ, USA, 2019; pp. 1–4.

28. Yu, L.; Hu, Y.; Xie, X.; Lin, Y.; Hong, W. Complex-Valued Full Convolutional Neural Network for SAR Target Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1752–1756. [CrossRef]

29. Zhang, Y.; Hua, Q.; Jiang, Y.; Li, H.; Xu, D. Cv-MotionNet: Complex-valued convolutional neural network for SAR moving ship targets classification. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; IEEE: Piscataway, NJ, USA, 2021, pp. 4280–4283.

30. Wang, Y.; Zhao, W.; Wang, X.; Chen, J.; Li, H.; Cui, G. Nonhomogeneous sea clutter suppression using complex-valued U-Net model. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4027705. [CrossRef]

31. Zhang, Y.; Yuan, H.; Li, H.; Wei, C.; Yao, C. Complex-valued graph neural network on space target classification for defocused ISAR images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4512905. [CrossRef]

32. Zhou, X.; Luo, C.; Ren, P.; Zhang, B. Multiscale Complex-Valued Feature Attention Convolutional Neural Network for SAR Automatic Target Recognition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 2052–2066. [CrossRef]

33. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

34. Masykur, F.; Adi, K.; Nurhayati, O.D. Classification of paddy leaf disease using MobileNet model. In Proceedings of the 2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED), Sukabumi, Indonesia, 28–29 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–4.

35. Patra, G.R.; Naik, M.K.; Mohanty, M.N. ECG Signal Classification Using a CNN-LSTM Hybrid Network. In Proceedings of the 2023 2nd International Conference on Ambient Intelligence in Health Care (ICAIHC), Bhubaneswar, India, 17–18 November 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.