


## Research Article

# Highly Robust Synthetic Aperture Radar Target Recognition Method Based on Simulation Data Training

Liping Hu,<sup>1</sup> Canming Yao,<sup>2</sup> Jian Huang,<sup>3</sup> Jinfan Liu,<sup>1</sup> and Guanyong Wang<sup>1,4</sup> 

<sup>1</sup>Science and Technology on Electromagnetic Scattering Laboratory, Beijing Institute of Environmental Features, Beijing 100854, China

<sup>2</sup>School of Electronics and Communication Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen 518107, China

<sup>3</sup>Beijing Institute of Tracking and Telecommunications Technology, Beijing 100094, China

<sup>4</sup>Beijing Institute of Radio Measurement, Beijing 100854, China

Correspondence should be addressed to Guanyong Wang; [guanbingwang@126.com](mailto:guanbingwang@126.com)

Received 3 July 2022; Revised 23 August 2022; Accepted 6 September 2022; Published 21 September 2022

Academic Editor: Mohammad Alibakhshikenari

Copyright © 2022 Liping Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sufficient synthetic aperture radar (SAR) data is the key element in achieving excellent target recognition performance for most deep learning algorithms. It is unrealistic to obtain sufficient SAR data from the actual measurements, so SAR simulation based on electromagnetic scattering modeling has become an effective way to obtain sufficient samples. Simulated and measured SAR images are nonhomologous data. Due to the fact that the target geometric model of SAR simulation is not inevitably consistent with the real object, the SAR sensor model in SAR simulation may be different from the actual sensor, the background environment of the object is also inevitably different from that of SAR simulation, the error of electromagnetic modeling method itself, and so on. There are inevitable differences between the simulated and measured SAR images, which will affect the recognition performance. To address this problem, an SAR simulation method based on a high-frequency asymptotic technique and a discrete ray tracing technique is proposed in this paper to obtain SAR simulation images of ground vehicle targets. Next, various convolutional neural networks (CNNs) and AugMix data augmentation methods are proposed to train only on simulated data, and then target recognition on MSTAR measured data is performed. The experiments show that all the CNNs can achieve incredible recognition performance on the nonhomologous SAR data, and the RegNetX-3.2GF achieves state-of-the-art performance, up to 84.81%.

## 1. Introduction

Synthetic aperture radar (SAR) is able to perform high-resolution imaging of targets all day and in all weathers, which can provide sufficient target information. Therefore, SAR target recognition has increasingly attracted attention. The existing SAR target recognition algorithms, especially for target recognition algorithms based on deep learning, are mostly verified on the ground vehicle targets data sets collected by the moving and stationary target acquisition and recognition (MSTAR) program of the United States, which have achieved excellent recognition performance [1–8]. In the above literature, the training and test samples belong to the same homologous data. The range of azimuth

coverage is 0–360°, and the pitch angle difference between training and test samples is only 2°. In real target recognition applications of target recognition, it is difficult to obtain full-angle data of targets, especially for noncooperative targets, through actual measurement. Usually, it can only obtain measured data from the individual pitch and azimuth angles. The method of SAR simulation based on electromagnetic scattering modeling is an effective way to obtain sufficient samples [9–15]. It can conveniently obtain a relatively complete data set of targets under different conditions. The ultimate goal of SAR simulation is to replace the actual data as training samples in part or all so as to realize the target recognition of the measured data. However, when SAR simulation image data is directly applied to measured data

target recognition, the problem is obvious. SAR simulation data and measured data are nonhomologous data. Due to the differences in geometric model, sensor model, background environment type, modeling method, and other factors, the SAR simulation image and the measured image are different. It is mainly reflected in two aspects: one is the difference in details of target scattering distribution, and the other is the difference in the scattering of the background environment. These differences are bound to affect recognition performance.

In recent years, deep learning has achieved good results in various fields of pattern recognition; particularly, the convolutional neural networks (CNNs) have made a series of breakthroughs in the field of image classification.

In [16–18] and [19–27], the classification accuracy is far higher than the best level in the past. The application research of CNNs in SAR images mainly focuses on the verification of the target recognition algorithm based on the MSTAR data set. MSTAR training and test data set belong to homologous measured data, and the difference between training and test samples is small. CNNs have achieved excellent recognition performance on the MSTAR homologous data set [5–8]. Taking the advantages of CNNs in the field of image classification and their successful application in the MSTAR homologous data set, we employ various CNNs for SAR target recognition of nonhomologous data in this paper to investigate the recognition performance.

CNNs can achieve high accuracy when the training distribution and test distribution are identically distributed, but the distribution of nonhomologous SAR data has a lot of differences. Currently, data augmentation is an effective technique to eliminate the differences of nonhomologous data and improve the recognition performance of the data shifts encountered during deployment by randomly “augmenting” it [16, 28, 29], for instance, translating the image by a few pixels or flipping the image horizontally. In this paper, we introduce AugMix [30], a data processing method that is simple to implement and helps the model improve the robustness of classification for nonhomologous data during inference. AugMix utilizes stochasticity and diverse augmentations, a formulation to mix multiple augmented images, and a Jensen–Shannon divergence consistency loss to boost the performance of all CNNs employed in this paper. In our experiments, AugMix achieves excellent improvements compared with other common data augmentation methods.

The main contributions of this paper are given as follows:

- (1) The performance limitation of the SAR simulation method based on the high-frequency asymptotic technique and discrete ray tracing technique is analyzed in detail, which provides a theoretical basis for the following nonhomologous SAR target recognition.
- (2) An effective data preprocessing procedure is proposed to reduce the influence of noise inferences in SAR images and to help CNNs extract the main separable features of the target.

- (3) Various CNN-based methods for nonhomologous SAR target recognition are investigated. The experimental results show that the introduced AugMix can alleviate the gap between SAR simulated data and measured data and enable the network to improve the robustness and uncertainty estimates of nonhomologous data classifiers. This paper is an exploration and attempt to apply SAR simulation data to practical target recognition, which has a significant practical application value.

The paper is organized as follows: Section 2 discusses the principle and analysis of electromagnetic scattering modeling. Section 3 presents the theory and implementation steps of nonhomologous target recognition. In Section 4, we present the experiments and analysis. Conclusions are summarized in Section 5.

## 2. Electromagnetic Scattering Modeling

*2.1. Flowchart of Ground Vehicle Target Simulation.* In order to simulate the SAR template images of the ground vehicle targets, the SAR echo signal level simulation based on the high-frequency asymptotic method and discrete ray tracing technology is used for SAR simulation [15], and its flowchart is shown in Figure 1. The complex scene and SAR platform motion are firstly modeled, and then the SAR echo data is simulated combined with the composite electromagnetic calculation method of the target and environment. Finally, the resulting image is obtained by the SAR imaging process. Because the SAR echo signal level simulation method truly simulates the electromagnetic scattering process of SAR to target and environment, it can be used to obtain high similarity SAR simulation images with measured images.

*2.2. Simulation Test.* The geometric models of three vehicle targets (including BMP2, BTR70, and T72) given in Figure 2 are simulated according to the spotlight SAR simulation parameters set in Table 1. SAR simulation images on the grass are obtained, as shown in Figure 3. The parameters of the simulation images are given as follows: X-band, HH polarization, high-resolution spotlight SAR, depression angle of  $17^\circ$ , azimuth angle changing from  $0^\circ$  to  $359^\circ$  with an azimuth interval of  $1^\circ$ , and image size of  $128 \times 128$  with the resolutions of  $0.3 \text{ m} \times 0.3 \text{ m}$ . Figure 3 also shows the measured SAR image of the three vehicles collected by the MSTAR program. The acquisition conditions of the measured images are X-band, HH polarization,  $0.3 \text{ m} \times 0.3 \text{ m}$  high-resolution spotlight SAR, depression angle of  $17^\circ$ , azimuth angle changing from  $0^\circ$  to  $360^\circ$  with an azimuth interval of  $1^\circ \sim 5^\circ$ , and image size of  $128 \times 128$ .

*2.3. Performance Analysis.* By comparing the SAR simulation and the measured images in Figure 3, it can be seen that the two images under the same azimuth are generally consistent in the target contour shape and strong scattering distributions, but there are some details differences and some background environment scattering differences. For

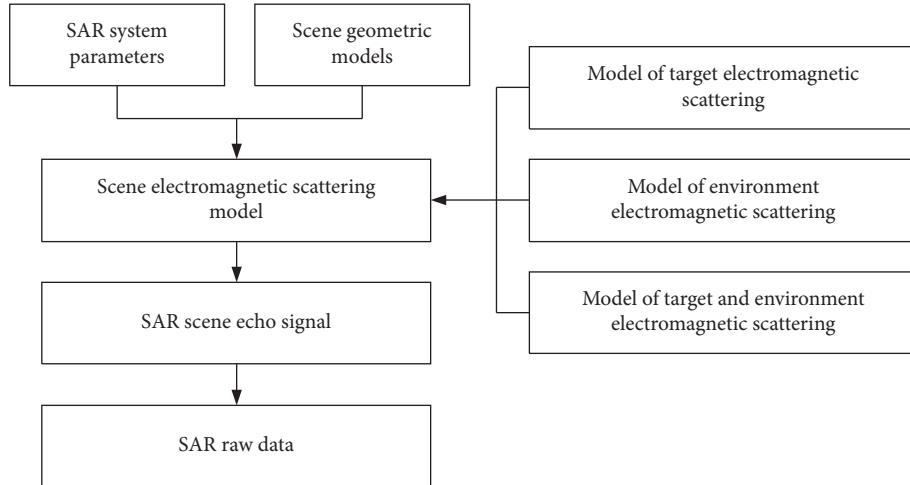


FIGURE 1: SAR simulation process-based electromagnetic scattering modeling.

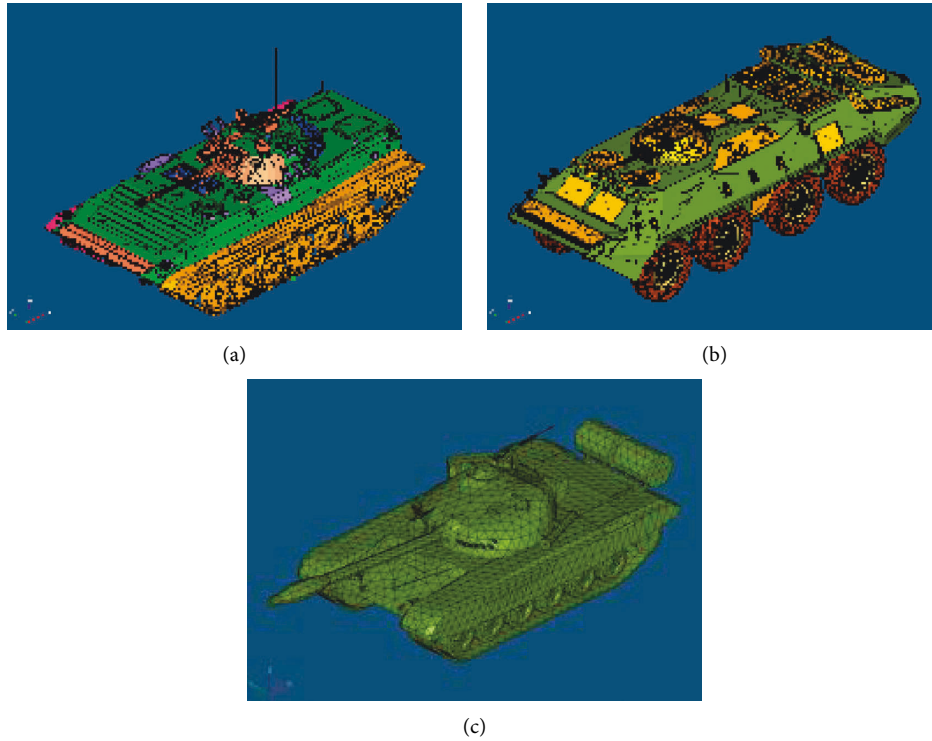


FIGURE 2: Geometric models of the three vehicles. (a) BMP2; (b) BTR70; (c) T72.

TABLE 1: Spotlight SAR simulation parameters of vehicle targets.

Parameter	Value	Parameter	Value
Imaging mode	Spotlight	Pitch angle	17°
Center frequency	9.6 GHz	Signal bandwidth	591 MHz
Azimuthal resolution	0.3 m	PRF	600 Hz
Beam horizontal width	4.293°	Beam pitch width	1.044°
Platform height	400 m	Flight speed	100 m/s
Sample points	512	Sample points	512

the SAR simulation images, although model verification and image evaluation have been carried out [14, 15, 20], there are

still differences between the SAR simulation and measured images due to various reasons. The main reasons are summarized as follows:

- (1) Model differences: there are inevitable differences between the target geometric model of SAR simulation and the real object. The geometric model of the target constructed during SAR simulation and the real object inevitably have some simplification of the detailed structure, the error of the shape structure, the state of some structures on the target (such as the rotating position of the gun barrel), and the state of some extension conditions on the target (such as

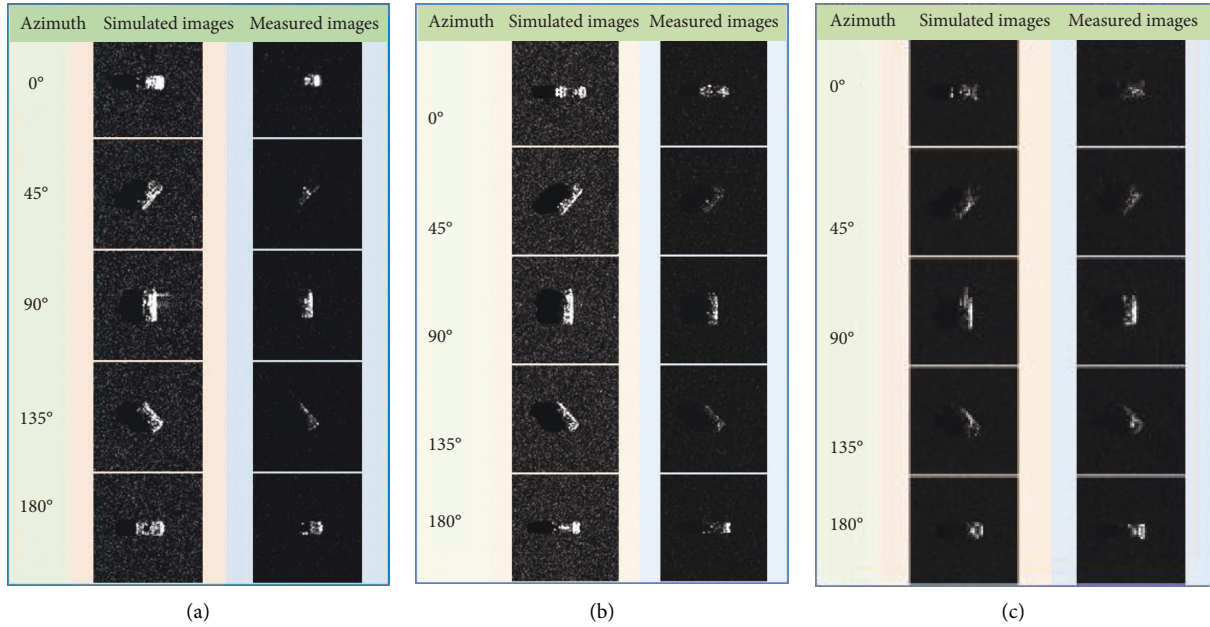


FIGURE 3: SAR simulation images of vehicles on grassland and MSTAR measured SAR images. (a) BMP2; (b) BTR70; (c) T72.

whether the T72 tank is equipped with auxiliary oil tank). These differences lead to the differences between the SAR simulation image and the measured image.

- (2) Sensor difference: there are performance differences between the sensor model of the SAR simulation and the actual sensor. The sensor model constructed during SAR simulation is ideal. For example, the sensor motion model is a steady and uniform linear motion, which is difficult for the actual SAR sensor. The performance of the actual SAR sensor also is different from the theoretical values in terms of signal-to-noise ratio, actual resolution, pitch angle, and azimuth angle due to the limitations of the system hardware.
- (3) Background environment difference: there are differences between the background environment of the SAR simulation and the real object. The ground scene constructed by SAR simulation is a horizontal low grass background without any topographic relief, and the model parameters used are the low grass in the Uraby model [15], which is inevitably different from the real ground background in the measured MSTAR SAR data set. So it results in the inconsistency of the background scattering intensity in the SAR simulation image and the measured SAR image.
- (4) Error of the modeling method: in order to satisfy the engineering needs and improve the calculation efficiency, the high-frequency method is used for target electromagnetic modeling, which will cause calculation errors for some special structures (such as cavities) and targets with multiscale structures. It results in target scattering distribution details differences between the SAR simulation image and the measured image.

The difference between the SAR simulation image and the measured image inevitably affects the target recognition performance of the SAR simulation image directly applied to the measured image and also brings great challenges to nonhomologous SAR target recognition. For nonhomologous SAR target recognition, the influence of background environment difference can be reduced by SAR image preprocessing. So reducing the influence of target scattering distribution difference between the SAR simulation image and the measured image is the key and difficult point of nonhomologous SAR target recognition.

### 3. Nonhomologous Target Recognition

*3.1. SAR Image Preprocessing.* Generally, speckle noise in the original SAR image seriously affects the recognition performance if the original SAR images are directly used for feature extraction and recognition. Therefore, the original SAR images need to be preprocessed as the inputs of CNN or linear and nonlinear feature transformation.

Figure 4 shows the SAR image preprocessing process. Firstly, the original image is transformed by logarithmic transformation. Secondly, the constant false alarm rate (CFAR) method is applied to roughly segment the target. Finally, some filtering operations, masking, and normalization are implemented in sequence.

Assuming that the background clutters obey a negative exponential distribution, the threshold is given as follows:

$$T = -\mu \ln(P_{fa}), \quad (1)$$

where  $P_{fa}$  is the false alarm rate and  $\mu$  is the mean value. For each point  $(i, j)$  in one SAR image, if its pixel value is greater than the threshold  $T$ , it is determined as the target point; otherwise, it is the background point.

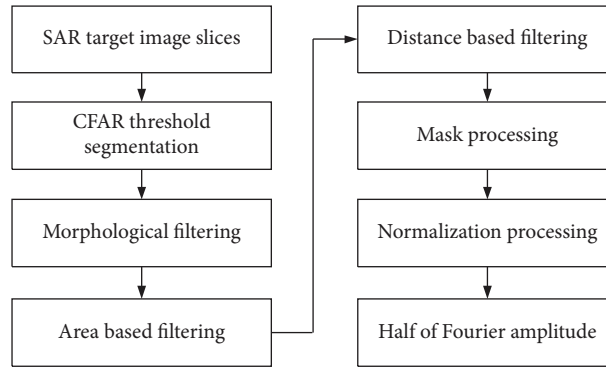


FIGURE 4: Flowchart of the SAR preprocessing method.

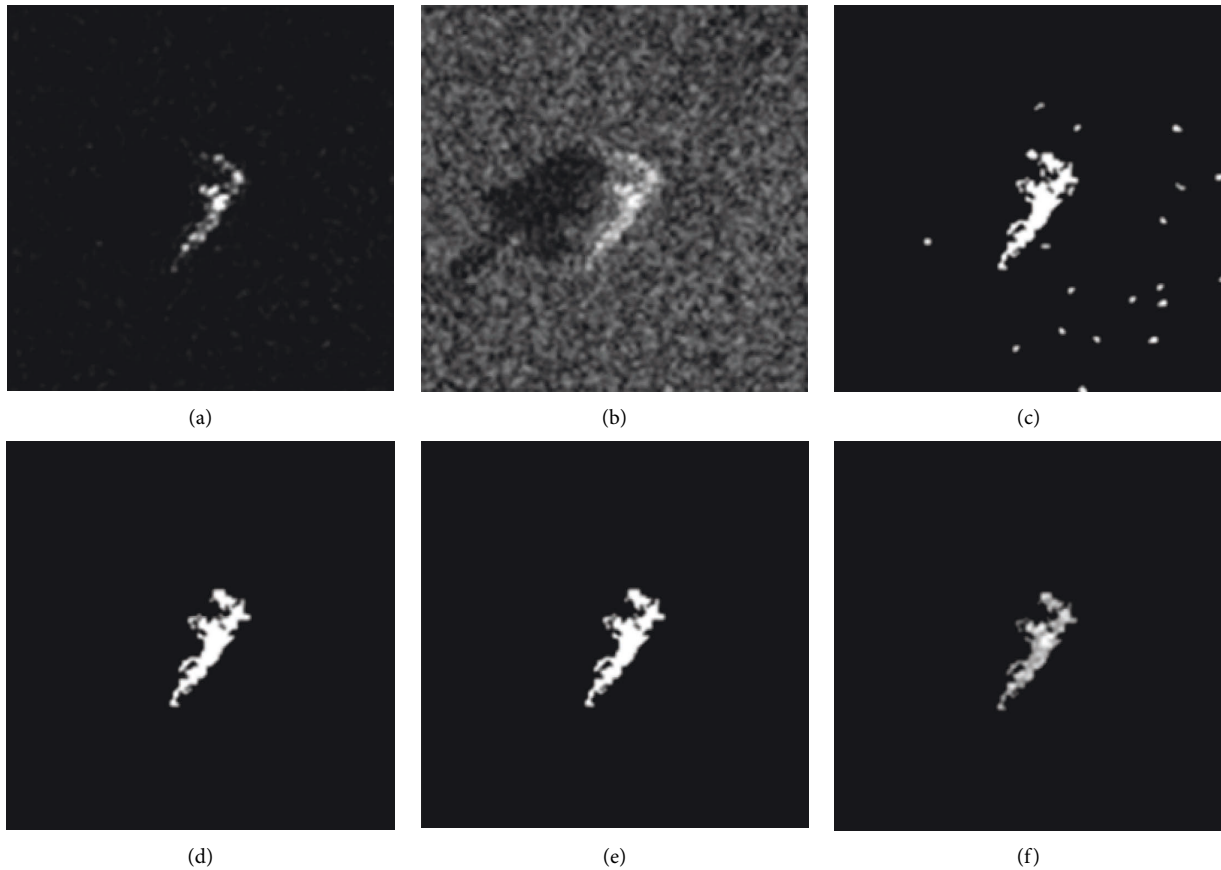


FIGURE 5: Preprocessed SAR images. (a) Original image; (b) logarithmic image; (c) threshold segmented image; (d) area-based filtered image; (e) distance-based filtered image; (f) image after dot multiplication of (e) and (b).

**Morphological filtering:** the purpose of the morphological filtering for the segmented result is to remove nontarget areas, reduce noises, smooth boundaries, remove small holes, and so on.

**Area-based filtering:** for the image processed by morphological filtering, first remove the isolated points and then eliminate the image regions with an area less than  $T_A$ . The parameter  $T_A$  is roughly determined by the size and the resolution of the interested target.

**Distance-based filtering:** for the resulting image of the previous step, first find the largest area and its centroid, then calculate the distance from each area to the largest area, and finally eliminate the area with a distance greater than  $T_D$ . The parameter  $T_D$  is roughly determined by the size and the resolution of the interested target.

**Masking:** in order to obtain the intensity information of the target, the filtered resulting image (binary image) and the logarithmic image do dot multiplication to obtain the final target intensity image as the input of CNN.

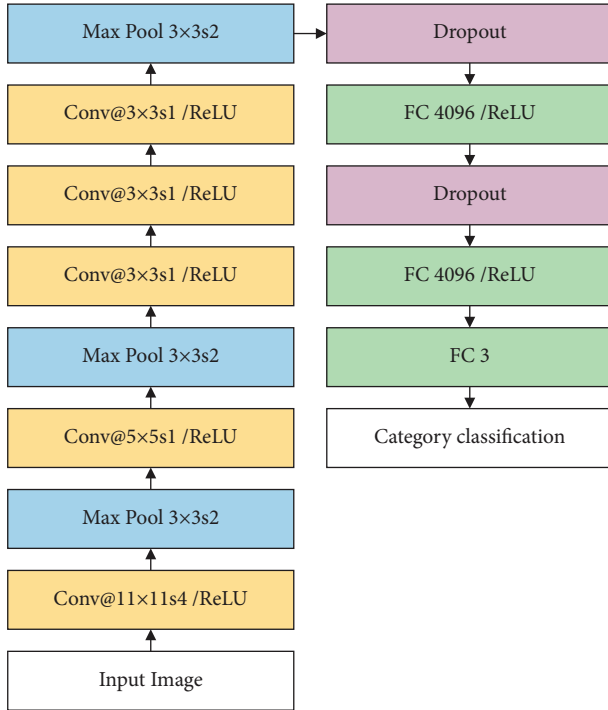


FIGURE 6: CNN structure constructed.

Finally, the target intensity image is normalized, and half of the amplitude after the 2-dimensional Fourier transform is taken as the input of linear and nonlinear feature transform.

As an example, Figure 5 shows the preprocessing result images of T72.

**3.2. Convolutional Neural Networks.** CNNs have achieved successful applications in the field of image classification and have excellent recognition performances on MSTAR homologous SAR data sets [5–8]. In this paper, we attempt to employ it to SAR target recognition on nonhomologous data to investigate the target recognition performance in the case of differences between SAR simulated data and actual measured data.

AlexNet [16] is a classic deep convolutional neural network with a simple structure. It utilizes rectified linear unit (ReLU) activation function, dropout, max pooling, data augmentation, and other technologies, enabling the network to extract more discriminative features on samples, effectively avoid model overfitting, and improve the network robustness. The structure of AlexNet is shown in Figure 6, which includes five convolution layers, three max pooling layers, two dropout layers, and three full connection layers. Moreover, category classification is carried out by the softmax function after the last full connection layer.

VGGNets [19] (proposed by the Visual Geometry Group) inherit the convolutional network architecture of AlexNet and steadily increase the depth of the network by adding more convolutional layers with very small  $3 \times 3$  convolution filters. The main contribution is to demonstrate that the depth of convolutional networks is conducive to the classification accuracy of image recognition task. The architectures of VGGNets

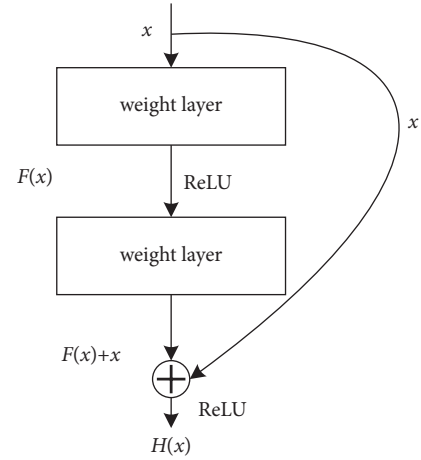


FIGURE 7: A residual block.

are outlined in Table 2, one per column, which are denoted as VGG-11, VGG-16, and VGG-19. All configurations differ only in the depth: from 11 layers in the network VGG-11 (8 convolutional and 3 Fully Connected (FC) layers) to 19 layers in the network VGG-19 (16 convolutional and 3 FC layers).

Network depth is crucial for enhancing feature extraction capabilities. However, with the network depth increasing, the problem of vanishing gradients becomes more severe, and the networks are more difficult to train. As a result, accuracy gets saturated and then degrades rapidly. Residual Networks (ResNets) [20] propose a deep residual learning framework to address the degradation problem. Instead of simply stacking convolutional layers by a one-way flow, ResNets explicitly let these layers fit a residual mapping with residual blocks. Figure 7 shows a residual block example. Formally, denoting the desired underlying mapping as  $H(x)$ , let the stacked nonlinear layers fit another mapping of  $F(x) = H(x) - x$ . The original mapping is recast into  $F(x) + x$ . Table 3 shows the detailed architectures of ResNets.

Dense Convolutional Networks (DenseNets) [21] embrace the “shortcut connections” of ResNets, which connect each layer to every other layer in a feed-forward fashion. For each layer in DenseNets, the feature maps of all preceding layers are used as inputs, and their own feature maps are used as inputs into all subsequent layers. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. Table 4 shows the detailed architectures of DenseNets. The transition layers consist of a batch normalization (BN) layer and a  $1 \times 1$  convolutional layer followed by a  $2 \times 2$  average pooling layer.

Squeeze-and-Excitation Networks (SENet) [22] focus on the channel relationship and propose a new “Squeeze-and-Excitation” (SE) block, as shown in Figure 8. The SE block adaptively recalibrates channelwise feature responses by explicitly modeling interdependencies between channels. As a result, the network can selectively emphasize informative features and suppress less useful ones. The SE block first squeezes global spatial information into a channel descriptor, followed by employing a simple self-gating mechanism to

TABLE 2: Architectures of VGGNets. The added layers are shown in bold, and the ReLU is placed after each weight layer, which is not shown for brevity.

VGG-11	VGG-13	VGG-16	VGG-16	VGG-19
<i>Input image</i>				
Conv@3-64	Conv@3-64 Conv@3-64	Conv@3-64 Conv@3-64	Conv@3-64 Conv@3-64	Conv@3-64 Conv@3-64
<i>Max pool, 2 × 2, s2</i>				
Conv@3-128	Conv@3-128 Conv@3-128	Conv@3-128 Conv@3-128	Conv@3-128 Conv@3-128	Conv@3-128 Conv@3-128
<i>Max pool, 2 × 2, s2</i>				
Conv@3-256	Conv@3-256	Conv@3-256 Conv@3-256	Conv@3-256 Conv@3-256	Conv@3-256 Conv@3-256
Conv@3-256	Conv@3-256	Conv@1-256	Conv@3-256	3-256 Conv@3-256
<i>Max pool, 2 × 2, s2</i>				
Conv@3-512	Conv@3-512	Conv@3-512 Conv@3-512	Conv@3-512 Conv@3-512	Conv@3-512 Conv@3-512
Conv@3-512	Conv@3-512	Conv@1-512	Conv@3-512	3-512 Conv@3-512
<i>Max pool, 2 × 2, s2</i>				
Conv@3-512	Conv@3-512	Conv@3-512 Conv@3-512	Conv@3-512 Conv@3-512	Conv@3-512 Conv@3-512
Conv@3-512	Conv@3-512	Conv@1-512	Conv@3-512	3-512 Conv@3-512
<i>Max pool, 2 × 2, s2</i>				
FC, 4096				
FC, 4096				
FC, 3				
Softmax				

TABLE 3: Architectures of ResNets. Downsampling is performed by the first conv layer in each stage, with a stride of 2.

Layer name	ResNet-18	ResNet-34	ResNet-50	ResNet-101	ResNet-152
Stage 0	7 × 7, 64, stride 2				
Max pool, 3 × 3, stride 2					
Stage 1	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Stage 2	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
Stage 3	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
Stage 4	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
Classification	Global average pool 3-d FC, softmax				

produce weights for channel features. Finally, the feature maps are reweighted to generate the output of the SE block, which can then be fed directly into subsequent layers. SE blocks can be used as a drop-in replacement for the original block at any depth in any architected network. More importantly, SE blocks are sufficiently flexible to be used in ResNets. In this paper, we replace the original convolutional block in ResNets with the SE block, which we term the SE-ResNets. Moreover, we compare

the various variants with the above-mentioned networks on nonhomologous SAR data.

EfficientNets [23] empirically identify that balancing the network depth, width, and resolution can lead to better performance. Based on this observation, they first employ neural architecture search (NAS) to develop a baseline network and consider scaling it up for bigger models. To achieve that, they propose a simple yet effective compound

TABLE 4: Architectures of DenseNets. Note that each “conv” layer shown in the table corresponds to the sequence BN-ReLU-Conv.

Layer name	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-161
Stage 0	7 × 7 conv, stride 2			
	Max pool, 3 × 3, stride 2			
Stage 1	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition layer	1 × 1 conv 1 × 1 average pool, stride 2			
Stage 2	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition layer	1 × 1 conv 1 × 1 average pool, stride 2			
Stage 3	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition layer	1 × 1 conv 1 × 1 average pool, stride 2			
Stage 4	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification	Global average pool 3-d FC, softmax			

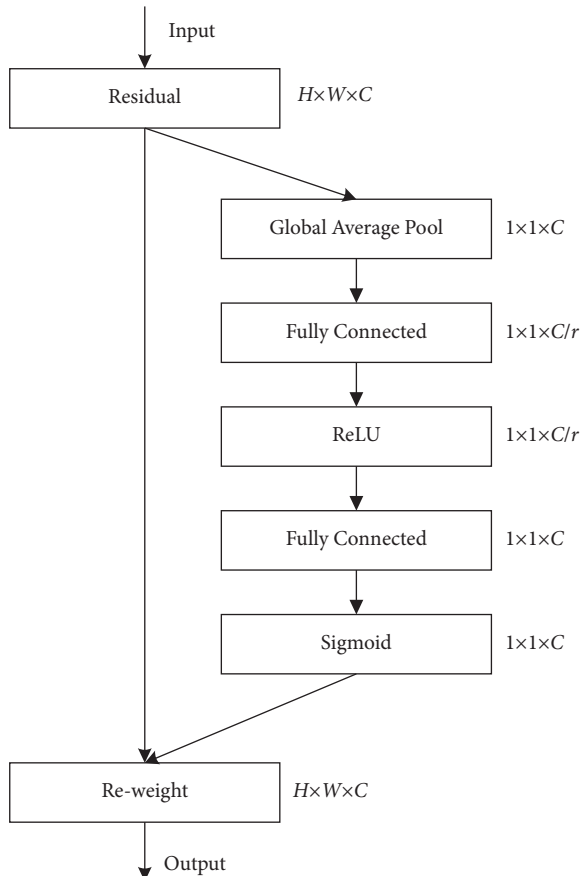


FIGURE 8: A Squeeze-and-Excitation block.

scaling method to uniformly scale network width, depth, and resolution with a set of fixed scaling coefficients and finally obtain a family of models, that is, EfficientNet-B0~B7. Table 5 describes the baseline EfficientNet-B0 designed by NAS. Its main building block is mobile inverted bottleneck MBConv 26,27, to which they also add SE block [22], as shown in Figure 9(a). Starting from the EfficientNet-B0, the compound scaling method uses a compound coefficient  $\phi$  to increase the network depth by  $\alpha^\phi$ , width by  $\beta^\phi$ , and image size by  $\gamma^\phi$ , where  $\alpha$ ,  $\beta$ , and  $\gamma$  are constant coefficients determined by a small grid search.

EfficientNetV2 [24] uses a combination of training-aware NAS and scaling to improve both training speed and parameter efficiency than EfficientNets [23]. They empirically identify that depthwise convolutions in MBConv blocks are slow in early layers. In view of this, they further design a search space enriched with other ops such as Fused-MBConv to optimize training speed, as shown in Figure 9(b). More importantly, EfficientNetV2 introduces a progressive learning strategy to change the image size to speed up training further: in the early training iterations, EfficientNetV2 trains the network with a small image size and weak regularization (e.g., dropout and data augmentation); then, it gradually increases image size and adds stronger regularization. Table 6 shows the searched baseline EfficientNetV2-S. EfficientNetV2-M/L can also be obtained by scaling up EfficientNetV2-S using similar compound scaling as EfficientNet [23].

RegNets [25] propose a new network design paradigm that combines the advantages of manual design and NAS. RegNets focus on designing network design spaces that parametrize populations of networks. The core of the RegNet design space is



TABLE 5: Architectures of EfficientNet-B0 baseline.

Stage	Operator	Resolution	No. of channels	No. of layers
1	Conv3 × 3	224 × 224	32	1
2	MBCConv1, k3 × 3	112 × 112	16	1
3	MBCConv6, k3 × 3	112 × 112	24	2
4	MBCConv6, k5 × 5	56 × 56	40	2
5	MBCConv6, k3 × 3	28 × 28	80	3
6	MBCConv6, k5 × 5	14 × 14	112	3
7	MBCConv6, k5 × 5	14 × 14	192	4
8	MBCConv6, k3 × 3	7 × 7	320	1
9	Conv1 × 1&Pooling&FC	7 × 7	1280	1

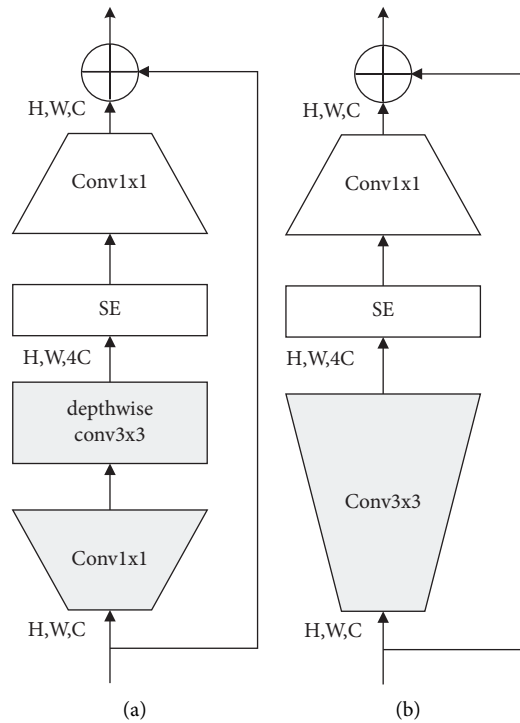


FIGURE 9: Structure of (a) MBCConv and (b) Fused-MBCConv.

TABLE 6: Architectures of EfficientNetV2-S.

Stage	Operator	Stride	No. of channels	No. of layers
0	Conv3 × 3	2	24	1
1	Fused-MBCConv1, k3 × 3	1	24	2
2	Fused-MBCConv4, k3 × 3	2	48	4
3	Fused-MBCConv1, k3 × 3	2	64	4
4	MBCConv4, k3 × 3, SE0.25	2	128	6
5	MBCConv6, k3 × 3, SE0.25	1	160	9
6	MBCConv6, k3 × 3, SE0.25	2	256	15
7	Conv1 × 1&Pooling&FC	—	1280	1

simple: stage widths and depths are determined by a quantized linear function, that is, (1) how to set stage widths; (2) how to set the number of blocks in each stage. RegNets arrive at exciting findings that the depth of the best models is stable across compute regimes and that the best models do not use either a

bottleneck or inverted bottleneck. RegNets design two models, that is, RegNetX and RegNetY ( $Y = X + SE$ ), and also suffix the models with the flop regime, for example, 800MF.

**3.3. AugMix Data Augmentation.** Intuitively, there are significant differences in the distribution of simulated and measured data. When the distributions of training and test samples are nonhomologous and mismatched, accuracy can plummet. In order to improve robustness to nonhomologous data shifts encountered during test time, we employ a simple data processing method, that is, the AugMix [30], to produce high diversity of augmented images, avoiding the CNN, which has a tendency to memorize properties of the specific training samples. AugMix utilizes stochasticity and diverse augmentation, a formulation to mix multiple augmented images, and a Jensen–Shannon divergence consistency loss to achieve data augmentation.

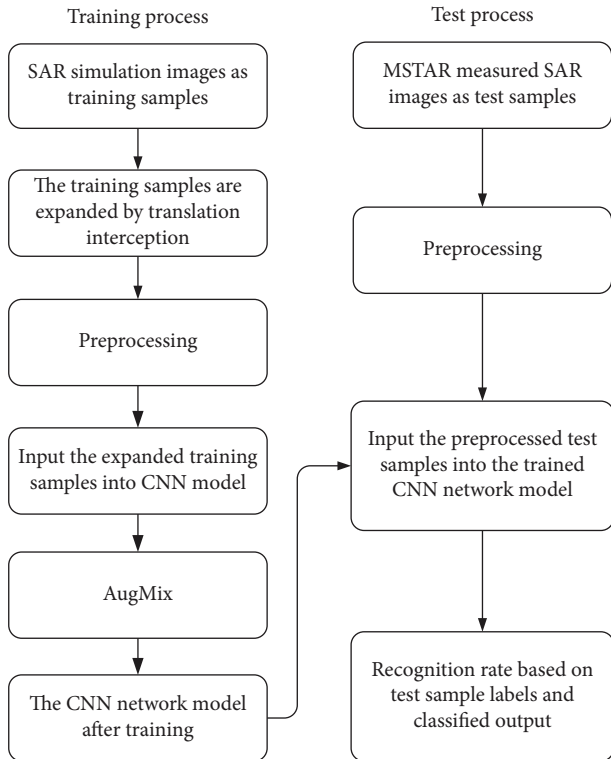


FIGURE 10: Nonhomologous SAR target recognition process based on CNN.

**3.3.1. Stochastic Augmentations.** The augmentation operations used in our paper follow the AutoAugment [29], including the autocontrast, equalize, posterize, rotate, solarize, shear  $X(Y)$ , and translate  $X(Y)$  operations. We randomly sample  $k$  augmentation chains, where  $k=3$  by default. Each augmentation chain is constructed by composing from one to four randomly selected augmentation operations. Randomly sampled operations and their compositions allow us to explore the semantically equivalent input space around an image.

**3.3.2. Mixing.** The resulting images from these augmentation chains are combined by mixing. For simplicity, we weigh each chain for combination. The  $k$ -dimensional vector of weight coefficients is randomly sampled from a Dirichlet( $\alpha, \dots, \alpha$ ) distribution. Once these images are mixed, we combine the mixed image and the original image through a random weight coefficient sampled from a Beta( $\alpha, \alpha$ ) distribution. Finally, mixing these images together produces a new image without veering too far from the original.

**3.3.3. Jensen-Shannon Divergence Consistency Loss.** For each original image  $x$ , we obtain two augmented images  $x_1$  and  $x_2$  through the augmentation and mixing process described above. Since the semantic content of an image is approximately preserved with AugMix, we hope the model embeds  $x$ ,  $x_1$ , and  $x_2$  similarly. Toward this end, we minimize the Jensen-Shannon divergence among the posterior distributions of the original image  $x$  and its augmented images. That is, for  $p = \hat{p}(y|x)$ ,  $p_1 = \hat{p}(y|x_1)$ , and  $p_2 = \hat{p}(y|x_2)$ , we define the model loss:

TABLE 7: Training and test samples used in nonhomologous target recognition experiments.

Number of samples	BMP2	BTR70	T72
Training samples (SAR simulation images)	360	360	360
Training samples (after expansion)	3240	3240	3240
Test samples (MSTAR measured images)	233	233	232

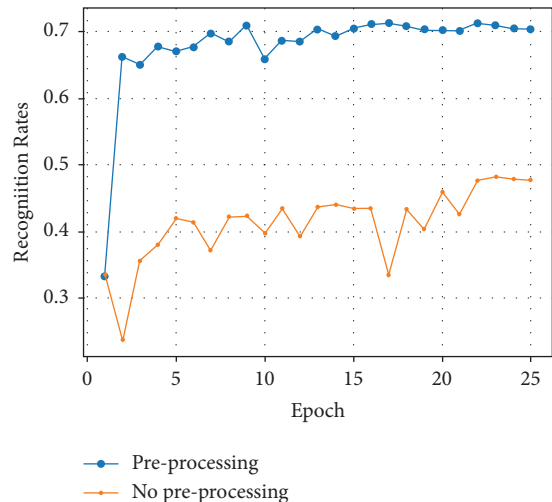


FIGURE 11: Influence of preprocessing on recognition rates.

$$L = L(p, y) + \lambda \text{JS}(p; p_1; p_2), \quad (2)$$

where  $\lambda = 12$  by default and  $y$  is the class label. This loss can be computed by first obtaining  $M = (p + p_1 + p_2)/3$  and then computing

$$\text{JS}(p; p_1; p_2) = \frac{1}{3} (\text{KL}[p||M] + \text{KL}[p_1||M] + \text{KL}[p_2||M]), \quad (3)$$

where KL means KL Divergence. The Jensen-Shannon divergence can be understood to measure the average information that the sample reveals about the identity of the distribution from which it was sampled. The Jensen-Shannon consistency loss impels to model to be stable, consistent, and insensitive across a diverse range of inputs.

To sum up, the process of nonhomologous SAR target recognition based on any CNN is shown in Figure 10. Crucially, the training samples and test samples are preprocessed to reduce the influence of speckle noise in SAR images. Moreover, in order to prevent the overfitting problem in small samples of the CNN, the translation interception of SAR images is used to expand the training samples. AugMix is used in the process of loading training data sets during training.

## 4. Experiments

**4.1. Data Sets and Settings.** In order to investigate the target recognition performance of different methods on nonhomologous SAR data, this paper carries out target recognition experimental verification on nonhomologous SAR data with three types of vehicle targets. The number of training and test samples is shown in Table 7. The training samples are simulated SAR

TABLE 8: Recognition rates of different data augmentation methods on AlexNet and ResNet-34.

Method	Random horizontal flip	Mixup	AugMix	Recognition rate (%)
AlexNet	✓			71.20
	✓	✓		75.35
	✓		✓	74.50
	✓		✓	<b>81.80</b>
ResNet-34	✓			72.63
	✓	✓		75.50
	✓			73.78
	✓		✓	<b>84.52</b>

TABLE 9: Recognition results of different CNN methods.

Method	Recognition rate (%)	Params (M)
AlexNet	81.80	57.02
VGG-16 (with BN)	80.08	134.28
VGG-19 (with BN)	80.94	139.59
ResNet-18	78.51	11.18
ResNet-34	84.52	21.29
ResNet-50	80.66	23.51
DenseNet-121	82.81	6.96
DenseNet-161	84.53	26.48
SE-ResNet-18	78.79	11.27
SE-ResNet-34	82.66	21.44
SE-ResNet-50	80.66	26.03
EfficientNet-B0	77.79	<b>4.01</b>
EfficientNet-B1	73.35	6.52
EfficientNet-B3	70.20	10.70
EfficientNetV2-S	82.23	20.18
EfficientNetV2-M	81.52	52.86
RegNetX-800MF	81.80	6.59
RegNetX-3.2GF	<b>84.81</b>	14.29
RegNetY-800MF	81.66	5.65
RegNetY-3.2GF	81.38	17.93

images for three types of vehicle targets, and each type contains 360 images, with the size being  $128 \times 128$ . The test samples are the MSTAR measured SAR images for three types of vehicle targets, and the size of each image is also  $128 \times 128$ . The translation interception of training samples is expanded by 9 times, and the size of the image is  $88 \times 88$  after center interception, ensuring that each image contains a target. The test samples are only intercepted into the size of  $88 \times 88$  pixels, containing the target.

Our CNN model is trained with adaptive moment estimation (Adam) for 25/40/100 epochs with a total of 16 images per minibatch. The CNN model uses an initial learning rate of 0.0001, which decays following a cosine learning rate. The image size is fixed to  $224 \times 224$  for training and testing. All input images are only preprocessed with standard random horizontal flipping prior to the AugMix augmentation. All the experiments are implemented on the PyTorch 1.8 framework and performed on a NVIDIA RTX 2080 Ti GPU.

## 4.2. Results

**4.2.1. Preprocessing Evaluation.** In order to analyze the influence of data preprocessing on recognition performance in nonhomologous SAR target recognition, we compare methods

with and without data preprocessing based on AlexNet. The comparison recognition rates among training epochs are shown in Figure 11. It can be seen that the data preprocessing method obviously improves the performance of AlexNet by a large margin. This is because the preprocessing method reduces the influence of background interference, enhancing the feature representation of the target. As a result, useful semantic information can be extracted more effectively by the network. We adopt this data preprocessing method in the subsequent CNN methods.

**4.2.2. AugMix Evaluation.** We conduct a series of experiments to demonstrate the effectiveness of the AugMix method in nonhomologous SAR target recognition based on AlexNet and ResNet-34. Specifically, we compare AugMix with two data augmentation methods, including random horizontal flip and Mixup. Mixup trains the neural network on convex combinations of pairs of samples and their labels. The results are shown in Table 8. With a random horizontal flip, AlexNet gains a 4% improvement, and ResNet-34 attains a nearly 3% increase. It is believed that horizontal flipping of images during training is an effective data augmentation method due to the fact that SAR targets are imaged from various angles and possess a certain symmetry. Nevertheless, there is a slight drop when additional Mixup is introduced compared to a purely random horizontal flip. This is because that Mixup is a kind of linear augmentation among minibatch training samples, which may produce images drifting far from the original image and lead to unrealistic images, thus jeopardizing the model performance. More importantly, it can be seen that both AlexNet and ResNet-34 achieve more than 10% higher recognition improvement over their corresponding baseline. This shows that mixing random augmentations and using the Jensen-Shannon loss substantially improve robustness and uncertainty estimates. For nonhomologous data, using the AugMix can ease the differences of their features, enhance the correlation among feature domains, finally improve the recognition performance, and effectively maintain robustness even as the distribution shifts at test time.

**4.2.3. Comparison with Different CNN-Based Methods.** We compare five CNN-based methods, including AlexNet, VGGNets (with batch normalization (BN) after every convolutional layer), ResNets, DenseNets, and SE-ResNets. The recognition results of these methods are listed in Table 9,

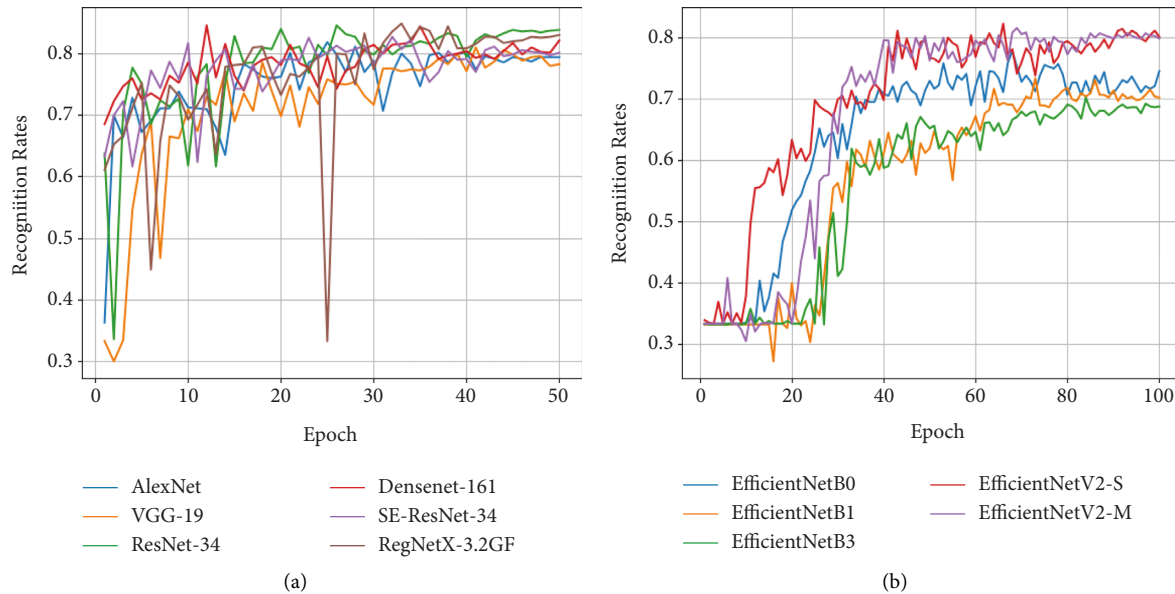


FIGURE 12: (a) Recognition rates of each best CNN method during training. (b) Recognition rates of EfficientNetV1 and V2 series.

and the corresponding accuracy curves of each best CNN method during training (excluding the EfficientNetV1 and V2 series) are displayed in Figure 12(a). The accuracy curves of the EfficientNet series are shown in Figure 12(b). It can be obtained that, in the early training epochs (10~20 epochs), the networks are divergent, and they need more training epochs to converge to stable performance. One of the main reasons for this degradation is the design of stacking many depthwise convolutions in the network blocks, which is not conducive to SAR target feature extraction, leading to slow convergence. Table 9 shows that the overall performance of the EfficientNetV2 series is better than the EfficientNetV1 series because they replace depthwise convolution with general  $3 \times 3$  convolution in a block (see Figure 9). It is also believed that conventional convolutions (e.g.,  $3 \times 3$  convolution) are beneficial in extracting more valuable semantic information in the SAR recognition task where the target is concentrated in the image center.

The results in Table 9 indicate that the CNN-based method is suitable for nonhomologous SAR target recognition. Almost all CNNs can achieve a recognition rate of more than 80%. This is because CNN methods can automatically extract the main features with good target separability and ignore the secondary features with poor separability. With that, various CNNs can alleviate the problem of the detailed difference in target scattering distribution between SAR simulation and measured images and achieve a certain recognition effect of non-homologous SAR targets.

Moreover, Table 9 shows that the deeper 19-layer VGGNet has better recognition performance than the shallower 16-layer. This phenomenon is also shown from ResNet-18 (SE-ResNet-18; DenseNet-121) to ResNet-34 (SE-ResNet-34; DenseNet-161). In fact, deeper networks can extract richer semantic information features and are more likely to enjoy accuracy gains from increased depth.

RegNetX-3.2GF attains the highest recognition rate among all CNN methods, up to 84.81%, which is 3% higher than RegNetX-800MF, which confirms that the characteristic of depth is indeed useful for nonhomologous target classification.

However, simply increasing the depth does not achieve the same high accuracy as always. Compared with other ResNets, the performance of ResNet-34 is the best, not ResNet-50, and the same is true of the 34-layer in SE-ResNets. In addition, a series of EfficientNets shows that performance drops dramatically even when the network is deeper. For instance, the recognition rate drops by 7% from EfficientNet-B3 to EfficientNet-B0 and by 0.7% from EfficientNetV2-M to EfficientNetV2-S. These indicate that the very deep networks are easily disturbed, resulting in redundant feature extraction, which is not conducive to discriminating the nonhomologous targets. In addition, although the general shape and strong scattering distribution of the target are basically the same, there are inevitable differences in detailed features. These differences will induce poor recognition results in the deeper networks.

The confusion matrices of the direct recognition method and each best CNN method are shown in Figure 13. For BMP2, the best recognition is performed by SE-ResNet-34, up to 78.54%. For BTR70, DenseNet and AlexNet both achieve a top-1 accuracy of 94.42%. For T72, ResNet-34 attains the highest recognition rate of 90.95%. Moreover, it can be seen that BMP2 is easily misclassified into T72 in all CNN methods, while BTR70 does not. The main reason is that BMP2 and T72 are tracked vehicles, BTR70 is the wheeled vehicle, and BMP2 and T72 are more similar in structure and easier to be confused.

To sum up, in order to improve the performance of nonhomologous SAR target recognition, in addition to taking measures to improve the simulation accuracy as

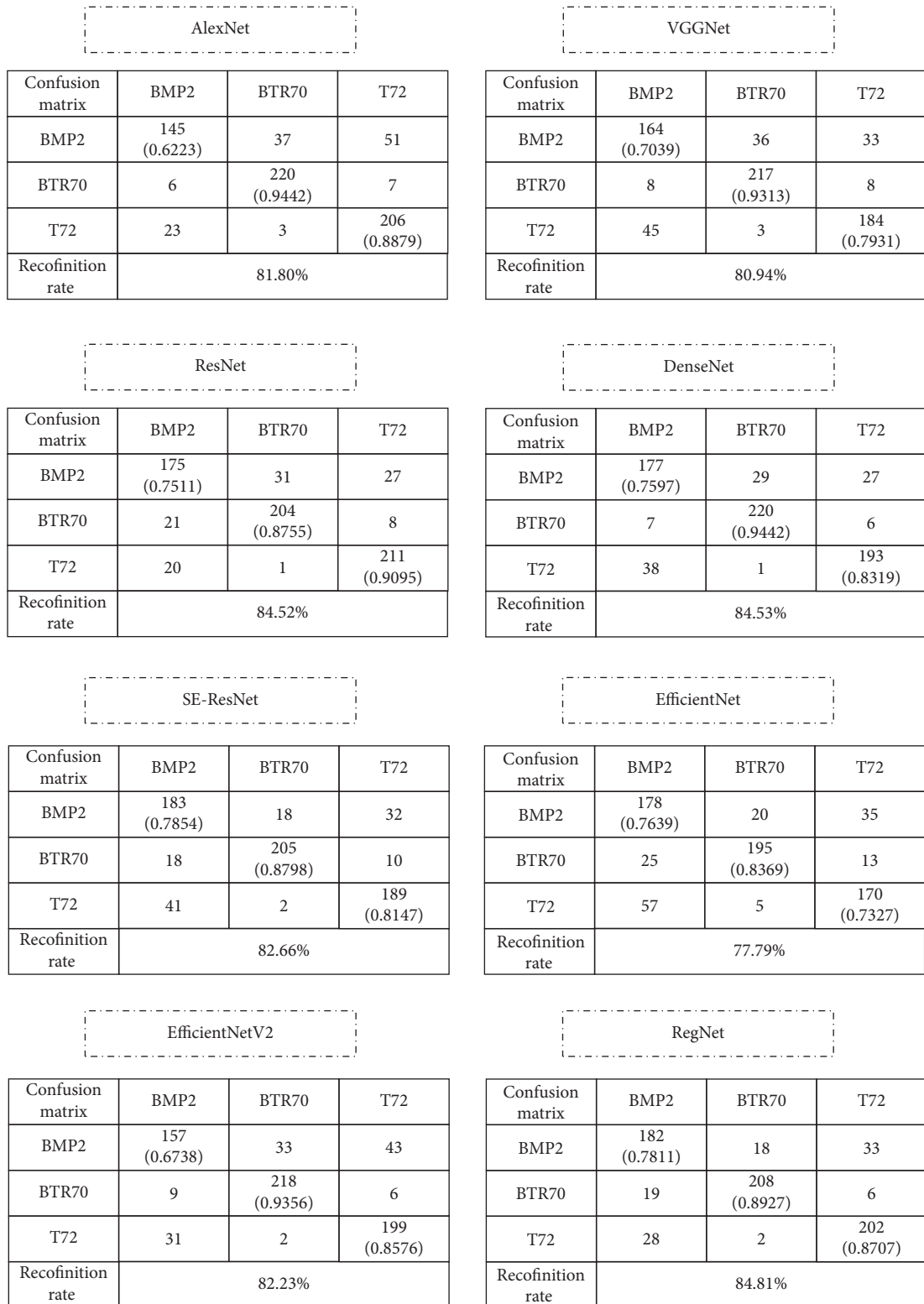


FIGURE 13: Confusion matrices for different CNN methods.

much as possible in the SAR image simulation process based on electromagnetic scattering modeling, extracting the main recognition features of nonhomologous data by CNN-based method is one of the main ways to solve the problem of nonhomologous SAR target recognition.

### 5. Conclusions

Aiming at the problem of unsatisfactory practical target recognition performance of SAR simulation image data, this paper attempts to explore the advantages of different CNN-

based methods to compare the recognition results of non-homologous targets with SAR simulated data as training samples and MSTAR measured SAR data as test samples. Experiments show that, for the nonhomologous SAR target recognition, the incredible recognition performance for the measured SAR data can be achieved by various CNNs. Moreover, the recognition performance can be significantly improved by using the AugMix method to process data. In a word, the extensive experiments verify the feasibility of nonhomologous target recognition based on SAR simulation data. On the other hand, it also demonstrates the research direction of how to improve the recognition performance when SAR simulation data is used for actual target recognition. As a long-term research work, more in-depth researches on CNNs will be carried out in the future in order to further improve the recognition performance of nonhomologous data and improve the application ability of SAR simulation data in actual target recognition.

### Data Availability

Data are not freely available and are subject to commercial confidentiality.

### Conflicts of Interest

The authors declare no conflicts of interest.

### Acknowledgments

This research was funded by Fundamental Strengthening Program-Key Fundamental Research Project, under grant number 2020-JCJQ-ZD-087-00. The authors acknowledge the Defense Advanced Research Project Agency (DAPPA) for providing the MSTAR data publicly available.

### References

- [1] J. I. Park and K. T. Kim, "Modified polar mapping classifier for SAR automatic target recognition," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 1092–1107, 2014.
- [2] J. I. Park, S. H. Park, and K. T. Kim, "New discrimination features for SAR automatic target recognition," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 3, pp. 476–480, 2013.
- [3] L. Hu, H. Liu, and S. Wu, "Novel pre-processing method for SAR image based automatic target recognition," *Journal of Xidian University*, vol. 34, no. 5, pp. 733–737, 2007.
- [4] L. Hu, H. Yin, and B. Chen, "Improved kernel clustering-based discriminant analysis," *Systems Engineering and Electronics*, vol. 33, no. 5, pp. 1176–1181, 2011.
- [5] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 1–5, 2016.
- [6] T. Zhu, *Research on Ground Target Recognition Techniques of Synthetic Aperture Radar Based on Deep Learning*, Harbin Institute of Technology, Harbin, 2017.
- [7] S. Chen, H. Wang, F. Xu, and Y. Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806–4817, 2016.
- [8] F. Xu, H. Wang, and Y. Jin, "Deep learning as applied in SAR target recognition and terrain classification," *Journal of Radars*, vol. 6, no. 2, pp. 136–148, 2017.
- [9] X. Guan, S. Wang, and Y. Su, "Simulation and experiment of electromagnetic scattering characterization for complex metal objects," *Chinese Journal of Radio Science*, vol. 22, no. 3, pp. 463–469, 2007.
- [10] M. Zhou, J. Lu, and G. Gao, "Simulation method of SAR slice images of vehicle targets based on RCS precise prediction," *Application Research of Computers*, vol. 26, no. 7, pp. 2274–2276, 2009.
- [11] R. Li, K. Ji, H. Zou, and S. L. Zhou, "Simulation of SAR imagery of target based on electromagnetic scattering characteristic computation," *Radar Science and Technology*, vol. 8, no. 5, pp. 395–400, 2010.
- [12] K. Ji, A. Zhang, H. Zou, and W. S. Sun, "Simulation and evaluation of SAR imagery of typical ground vehicles," *Radar Science and Technology*, vol. 8, no. 3, pp. 223–228, 2010.
- [13] R. Zhang, J. Hong, and F. Ming, "SAR echo and image simulation of complex targets based on electromagnetic scattering," *Journal of Electronics and Information Technology*, vol. 32, no. 12, pp. 2836–2841, 2011.
- [14] C. z. Dong, H. c. Yin, and C. Wang, "A fast hidden surface removal approach for complex SAR scene based on adaptive ray-tube splitting method," *Journal of Radars*, vol. 1, no. 4, pp. 436–440, 2013.
- [15] C. Dong, L. Hu, G. Zhu, and Y. Hong-cheng, "Efficient simulation method for high quality SAR images of complex ground vehicle," *Journal of Radars*, vol. 4, no. 3, pp. 351–360, 2015.
- [16] A. Krizhevsky, I. Sutskever, and G. Hinton, "Image net classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.
- [17] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, M. A, June 2015.
- [18] G. F. Tzortzis and A. C. Likas, "The global kernel K-means algorithm for clustering in feature space," *IEEE Transactions on Neural Networks*, vol. 20, no. 7, pp. 1181–1194, 2009.
- [19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, CVPR), Las Vegas, NV, USA, June 2016.
- [21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269, Honolulu, HI, USA, July 2017.
- [22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE/CVF Conference on*

- Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.
- [23] M. Tan and Q. Le, “Efficientnet: rethinking model scaling for convolutional neural networks,” in *Proceedings of the International Conference on Machine Learning*, Long Beach, CA, USA, 2019.
- [24] M. Tan and Q. Le, “Efficientnetv2: smaller models and faster training,” in *Proceedings of the International Conference on Machine Learning*, pp. 10096–10106, Vienna, Austria, 2021.
- [25] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, “Designing Network Design Spaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.
- [26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “Mobilenetv2: inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.
- [27] M. Tan, B. Chen, R. Pang et al., “Mnasnet: Platform-Aware Neural Architecture Search for mobile,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019.
- [28] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, “Mixup: beyond empirical risk management,” *6th International Conference Learning Representations*, Vancouver, BC, Canada, 2018.
- [29] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning Augmentation Strategies from Data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019.
- [30] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “Augmix: A Simple Data Processing Method to Improve Robustness and Uncertainty,” 2019, <https://arxiv.org/abs/1912.02781>.