remote sensing

MDPI

Article

Gradual Domain Adaptation with Pseudo-Label Denoising for SAR Target Recognition When Using Only Synthetic Data for Training

Yuanshuang Sun [1], Yinghua Wang [1,*], Hongwei Liu [1], Liping Hu [2], Chen Zhang [1] and Siyuan Wang [1]

[1] National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China
[2] Science and Technology on Electromagnetic Scattering Laboratory, Beijing Institute of Environmental Features, Beijing 100854, China
* Correspondence: yhwang@xidian.edu.cn

Abstract: Because of the high cost of data acquisition in synthetic aperture radar (SAR) target recognition, the application of synthetic (simulated) SAR data is becoming increasingly popular. Our study explores the problems encountered when training fully on synthetic data and testing on measured (real) data, and the distribution gap between synthetic and measured SAR data affects recognition performance under the circumstances. We propose a gradual domain adaptation recognition framework with pseudo-label denoising to solve this problem. As a warm-up, the feature alignment classification network is trained to learn the domain-invariant feature representation and obtain a relatively satisfactory recognition result. Then, we utilize the self-training method for further improvement. Some pseudo-labeled data are selected to fine-tune the network, narrowing the distribution difference between the training data and test data for each category. However, the pseudo-labels are inevitably noisy, and the wrong ones may deteriorate the classifier's performance during fine-tuning iterations. Thus, we conduct pseudo-label denoising to eliminate some noisy pseudo-labels and improve the trained classifier's robustness. We perform pseudo-label denoising based on the image similarity to keep the label consistent between the image and feature domains. We conduct extensive experiments on the newly published SAMPLE dataset, and we design two training scenarios to verify the proposed framework. For Training Scenario I, the framework matches the result of neural architecture searching and achieves 96.46% average accuracy. For Training Scenario II, the framework outperforms the results of other existing methods and achieves 97.36% average accuracy. These results illustrate the superiority of our framework, which can reach state-of-the-art recognition levels with appropriate stability.

Keywords: synthetic aperture radar (SAR); target recognition; domain adaptation; self-training; pseudo-label; synthetic dataset

check for updates

Citation: Sun, Y.; Wang, Y.; Liu, H, Hu, L.; Zhang, C.; Wang, S. Gradual Domain Adaptation with Pseudo-Label Denoising for SAR Target Recognition When Using Only Synthetic Data for Training. *Remote Sens.* **2023**, *15*, 708. https://doi.org/10.3390/rs15030708

Academic Editor: Dusan Gleich

Received: 25 November 2022
Revised: 14 January 2023
Accepted: 20 January 2023
Published: 25 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Synthetic aperture radar (SAR) data have been widely used in many fields, such as military surveillance, disaster warning, and environmental monitoring. SAR automatic target recognition (SAR-ATR) technology, which aims to recognize the target types of SAR images, has been comprehensively studied in the literature. With the popularity of deep learning methods in recent years, they have been effectively used for SAR-ATR, and a large amount of training data are usually needed to ensure the reliability of the trained deep models. However, due to the particularity of SAR images, the cost of acquiring a large amount of labeled SAR data is high, and a lack of training data will influence the recognition performance of the deep models [1–5].

Many research works have been conducted on SAR target recognition regarding cases of insufficient training data or few-shot learning [3–21]. In general, these methods can be divided into the following categories: (1) Data augmentation can be used to increase the

number of training samples. There are traditional data augmentation methods, such as the translation of the target and adding random speckle noise used in [6,7], and deep generative methods, such as generating images by Wasserstein generative adversarial networks with a gradient penalty [12] or by adversarial autoencoder [13]. (2) More robust features can be designed, which helps to improve recognition accuracy. There are traditional features, such as the sparse representation of the monogenic signal [8] and modified polar mapping [9], as well as features extracted by deep networks, such as deep highway unit [10], attribute-guided multi-scale prototypical network [11], angular rotation generative networks [5], and convolutional transformers [16]. (3) Transfer learning methods can be applied [3–5]. In this situation, auxiliary datasets can be used for transferring relevant knowledge [5] or pre-training the network [3,4]. (4) Other methods, such as semi-supervised learning methods [3,15,17–21] or meta-learning methods [11,14], can also be applied to the few-shot learning problem. The above methods can be combined for better results when the training data is scant.

Because the data acquisition of measured (real) SAR data is usually difficult, a growing number of studies [1,2,22–36] have been conducted that utilize synthetic (simulated) SAR data for the recognition problem. The Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset [37] provided by the Air Force Research Laboratory (AFRL), USA, has been widely used in the development of SAR-ATR algorithms over the past two decades. A dataset called the Synthetic and Measured Paired Labeled Experiment (SAMPLE) has also been introduced by AFRL to solve synthetic SAR target recognition problems [24]. In our paper, we conduct experiments on the SAMPLE dataset.

The research works based on the synthetic SAR data for recognition problems can be roughly divided into these main categories: (1) Pre-training: using plentiful synthetic SAR data to pre-train the classifier, and then using a small amount of measured SAR data to fine-tune the pre-trained network [1]; (2) Transformation or augmenting: transforming or augmenting the synthetic data to train the classifier to achieve better recognition results [2,22–26,29,34,35]; (3) Designing features: designing more general features [28,30,32,33,36] for training, such as ensembling traditional features [28], features based on scattering centers [32], or multi-scale deep features [36]. (4) Domain adaptation methods: adding some domain adaptation constraints for optimization, which makes the trained classifier more suitable for measured data [27,36]; (5) Other methods: such as neural architecture searching [31], or exploring the effectiveness of the existing methods and integrating some effective methods [25].

Considering the research works that use synthetic SAR data for recognition, there are two main recognition situations. First recognition situation: large and small amounts of synthetic and measured data, respectively, are labeled for training [1,2,22,29,35]. Second recognition situation: there are only synthetic data with labels for training, and the measured data of the same categories as the synthetic data are used as the test data [23–28,30–34,36]. The second recognition situation is more challenging. In our paper, we also consider the second recognition situation which is training fully on the synthetic SAR data and testing on the measured data, and the main challenge when training fully on the synthetic SAR data and testing on the measured data is the distribution gap between the training and test data.

In this paper, we design a gradual domain adaptation recognition framework with pseudo-label denoising to solve the distribution gap problem and achieve state-of-the-art recognition performance using the SAMPLE dataset. The existing methods [27,36] based on the domain adaptation idea only consider the feature alignment between different domains, while we utilize the self-training method further to narrow the distribution difference of the training data and test data for each category. Figure 1 illustrates the gradual domain adaptation process which contains two stages, i.e., feature alignment classification and self-training. The two stages are explained as follows. For the first stage, there is a domain shift between the original features of the training data and the test data, and feature alignment narrows the feature distance to learn the domain-invariant features. For the second stage of self-training, the classification boundaries are further modified to fit the pseudo-labeled test data, which can also align the distribution of each category between the synthetic

training data and the measured test data, thus enhancing the classification performance. The contributions of this paper can be summarized as follows:

(1) We design the two-stage gradual domain adaptation framework. Warm-up stage: training the feature alignment classification network to obtain the domain-invariant feature representation and a high-precision initialized classification result. Fine-tuning stage: selecting some pseudo-labeled data to fine-tune the classifier, and it actually narrows the distribution difference between the training data and test data for each category.
(2) We propose a pseudo-label denoising method to eliminate some falsely pseudo-labeled data, while falsely pseudo-labeled data negatively influence the fine-tuning stage. We use image similarity to measure the similarity of the pseudo-labeled data, and then keep the label consistent between the images of high similarities. The technique is simple and effective for improving the accuracy of the pseudo-labeled data, thus improving the final classification performance of the whole framework.
(3) We evaluate the whole framework on the newly published SAMPLE data set. The experimental results on the SAMPLE dataset illustrate that our proposed framework is practical and our proposed method can obtain state-of-the-art recognition accuracy.

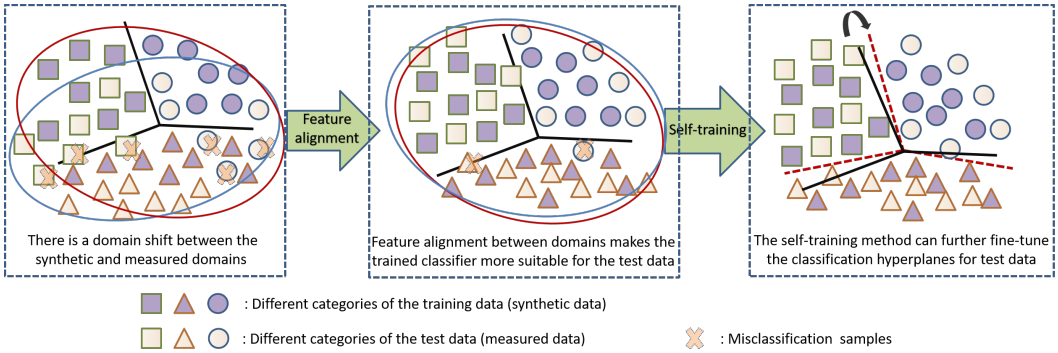


Figure 1. Illustration of the gradual domain adaptation process, which is a feature alignment classification network with self-training.

The rest of our paper is arranged as follows: in Section 2, we briefly review some related works; in Section 3, we introduce the gradual domain adaptation recognition framework and the pseudo-label denoising method; we provide the experimental setups and results in Section 4; finally, in Sections 5 and 6, we give some discussions and draw some conclusions, respectively.

2. Related Work

2.1. Transfer Learning and Unsupervised Domain Adaptation

Transfer learning [38,39] relaxes the hypothesis that the training data must be independent and identically distributed (i.i.d.) with the test data, and therefore, it can be used to solve the problem of insufficient training data. There are two domains for transfer learning problems: source domain D_s and target domain D_t.

Domain adaptation [40,41] is a type of transductive transfer learning, which means that the source task T_s in the source domain and the target task T_t in the target domain are the same, i.e., $T_s = T_t$, while the differences are only caused by domain divergence, $D_s \neq D_t$. Domain adaptation aims to utilize the labeled source data to improve the performance of the target task T_t.

Unsupervised domain adaptation is defined as a case in which labeled source data and unlabeled target data are available [41]. These unsupervised domain adaptation methods include aligning the feature distributions of different domains [42–44], designing different normalized statistics of different domains [45,46], and utilizing ensemble-based methods [47,48] or adversarial training [49,50]. The aforementioned methods can also be combined for the unsupervised domain adaptation problem.

2.2. Semi-Supervised Learning

Only labeled data are used to train the classifier for supervised learning. The classifier's performance depends on the sample size of training data to some extent. However, obtaining abundant labeled data is usually difficult, expensive, or time-consuming. Meanwhile, unlabeled data are relatively easier to obtain. Thus, semi-supervised learning [51,52] is popular because it can make use of unlabeled data.

Some studies [3,15,17–21] have utilized semi-supervised methods for SAR-ATR, mainly in the absence of labeled data. Zhang et al. proposed a semi-supervised method based on generative adversarial networks, which makes it capable of learning the feature representation of unlabeled input data [3]. Gao et al. used the active learning method to label some informative samples. They develop a semi-supervised strategy by adding a new regularization term to the loss functions of convolutional neural networks (CNN) [15]. An end-to-end semi-supervised recognition method based on an attention mechanism and bias-variance decomposition has also been proposed, focusing on unlabeled data screening and assigning pseudo-labels [17]. Wang et al. combined the contrastive learning method with pseudo-labels for the few-shot SAR-ATR [18]. A semi-supervised learning framework, which includes a self-consistent augmentation rule, a mix-up-based mixture, and weighted loss, has also been proposed [20]. Additionally, Liu et al. proposed a semi-supervised conditional generative adversarial network with multi-discriminators, which can discriminate the generated images and predict the labels for unlabeled samples [21].

2.3. Pseudo-Label Denoising

Self-training [53] is also known as the pseudo-label method. The basic idea of self-training is taking the pseudo-labeled samples for training until the prediction results of the classifier no longer change. Self-training performance depends on the correctness of the pseudo-labels provided by the initially trained classifier. When the noisy pseudo-labels are used for fine-tuning, the classifier may make more mistakes during the fine-tuning process. Furthermore, some noisy pseudo-labels will inevitably exist even with an initial high-precision classifier. Thus, pseudo-label denoising is essential for improving the performance of the self-training process.

In recent years, researchers [54–57] have considered how to de-noise false labels for SAR image classification. Wang et al. proposed a loss curve-fitting-based method to identify the noisy labels and effectively train the classification network [54]. Multi-classifiers are combined to keep the label consistent and eliminate noise [55]. Probability transition CNN has also been proposed for patch-level SAR image land-cover classification with noisy labels [56]. Shang et al. [57] proposed a SAR image segmentation method using region smoothing and label correction to correct the misclassified pixels in the image.

3. Gradual Domain Adaptation Recognition Framework

We propose a gradual domain adaptation recognition framework with pseudo-label denoising to solve the distribution gap problem and make the classifier trained by the training data more applicable to the test data. The flowchart of the whole algorithm is illustrated in Figure 2 below. Synthetic dataset X_L is in the source domain and measured dataset X_U is in the target domain. The recognition task is training on the X_L and then testing on the X_U. Since there is a distribution difference between the synthetic data and the measured data, we use a feature alignment classification network $\mathcal{F}$ to learn the domain-invariant feature representation for classification. Then, some pseudo-labeled data X_{deno} from X_U are selected by pre-set threshold and pseudo-labeled denoising method. $X_L \cup X_{deno}$ are used to fine-tune the $\mathcal{F}$ for a better result.

In this section, we first introduce a basic recognition network; then, we introduce the gradual domain adaptation recognition framework, which contains two domain adaptation stages: the feature alignment classification network and the self-training method; finally, we also present the proposed pseudo-label denoising method, which we carry out in the self-training process. The detailed introductions are given below.

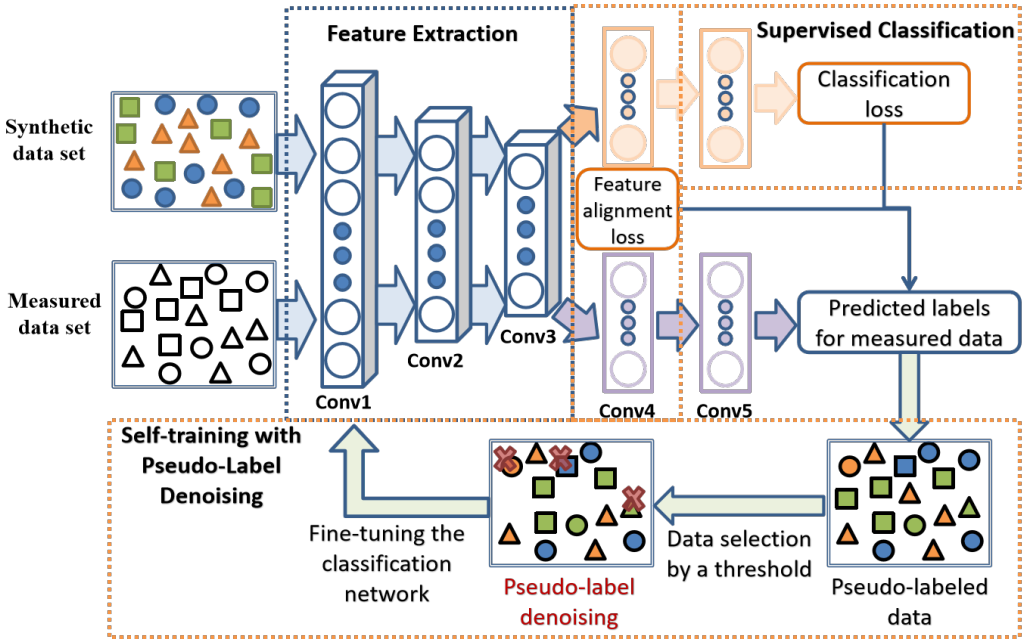


Figure 2. Flowchart of the training process of the whole algorithm.

3.1. Basic Recognition Network

A previous study [31] demonstrated that too deep network structures might not generalize well when there is a domain shift between the training and test data. Thus, we think that the five-layer network structure is sufficient to extract the deep distinguishing features because A-ConvNets have achieved state-of-the-art recognition accuracy for the ten-class MSTAR recognition problem [6]. As shown in Figure 3, we refer to A-ConvNets when designing the basic recognition network and modify the kernel sizes to fit the image sizes.

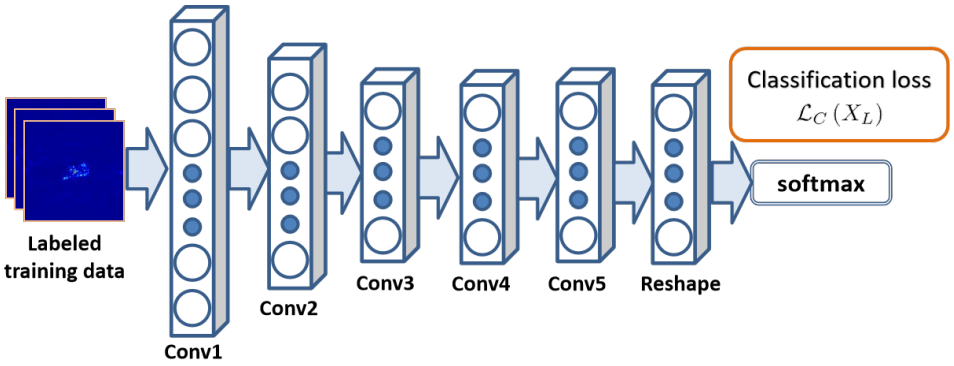


Figure 3. Basic recognition network.

Here, we introduce the architecture of the basic recognition network. The network has a five-layer all-convolutional architecture, and there is a dropout [58,59] operation after the fourth convolutional layer. The kernel size of the first four layers is 5×5, and the kernel size of the fifth layer is 4×4. We parameterize the 5-layer basic recognition network as $\phi(\cdot; \boldsymbol{\vartheta})$, and $\boldsymbol{\vartheta}$ represents the corresponding parameters.

For an input sample $\mathbf{x}$ with the real category label $\mathbf{y}$ of the one-hot form, the predicted label of the recognition network can be expressed as a 10-dimensional vector $\hat{\mathbf{y}} = \phi(\mathbf{x}; \boldsymbol{\vartheta})$. The specific expression is given as follows:

$$\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_K] \tag{1}$$

where $\hat{y}_i$ is the ith-dimensional value of vector $\hat{\mathbf{y}}$, which indicates the probability of the sample $\mathbf{x}$ belonging to the i-th category. K is the number of categories, and $K = 10$ in our experiment.

The classification loss $\mathcal{L}_C(X_L)$, which is actually cross entropy loss [60], is calculated on the labeled synthetic dataset X_L:

$$\mathcal{L}_C(X_L) = E_{X_L}[\sum_{i=1}^{K} -y_i \log \hat{y}_i] \tag{2}$$

where y_i is the ith-dimensional value of the real category vector $\mathbf{y}$.

3.2. Stage One: Feature Alignment Classification Network

As we can see in Figure 3, a recognition network $\phi(\cdot;\vartheta)$ is usually trained on the labeled training dataset X_L. Thus, when there is a distribution gap between the training and test data, the trained network may not fit well with the test dataset X_U.

We design a feature alignment classification network $\mathcal{F}$ to learn a domain-invariant feature representation for a better recognition result. Figure 4 gives the illustration of the feature alignment classification network, and an earlier version is introduced in Ref. [36]. However, while the earlier version utilizes a multi-scale deep network as the feature extractor, we use the all-convolutional architecture as the feature extractor.

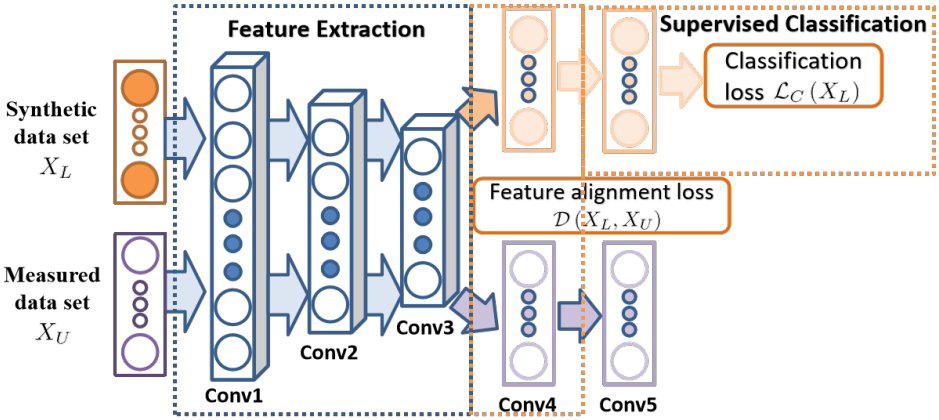


Figure 4. Feature alignment classification network.

Figure 4 shows that the network parameters are shared for the labeled synthetic dataset X_L and unlabeled measured dataset X_U. The classification loss $\mathcal{L}_C(X_L)$ is calculated on the labeled synthetic dataset, and the feature alignment restriction $\mathcal{D}(X_L, X_U)$ is calculated between the fourth-layer features of X_L and X_U. $\mathbf{x}_l$ and $\mathbf{x}_u$ represent the corresponding samples in the labeled synthetic data set X_L and unlabeled measured data set X_U, respectively.

The deep features of the fourth layer are expressed as $f(\mathbf{x}_l)$ and $f(\mathbf{x}_u)$. The feature alignment loss between the features of the synthetic and measured datasets can be simply expressed as follows [36,42]:

$$\mathcal{D}(X_L, X_U) = \|E_{X_L}[f(\mathbf{x}_l)] - E_{X_U}[f(\mathbf{x}_u)]\| \tag{3}$$

Thus, the total loss function of the feature alignment classification network can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_C(X_L) + \lambda \mathcal{D}(X_L, X_U) \tag{4}$$

where λ determines the proportion of feature alignment loss in the total loss $\mathcal{L}$.

As a warm-up, the feature alignment classification network $\mathcal{F}$ is trained to optimize the total loss function to learn the domain-invariant feature representation. The optimization of classification loss ensures an effective recognition performance on the synthetic training dataset X_L, while the optimization of the feature alignment loss can narrow the feature difference between the training and test data, thereby improving the classification result on the measured test dataset X_U.

3.3. Stage Two: Self-Training

Self-training is a common semi-supervised learning method, and it can also be utilized for further domain adaptation to narrow the distribution difference between the training and test data for each category. Figure 5 shows the self-training process.

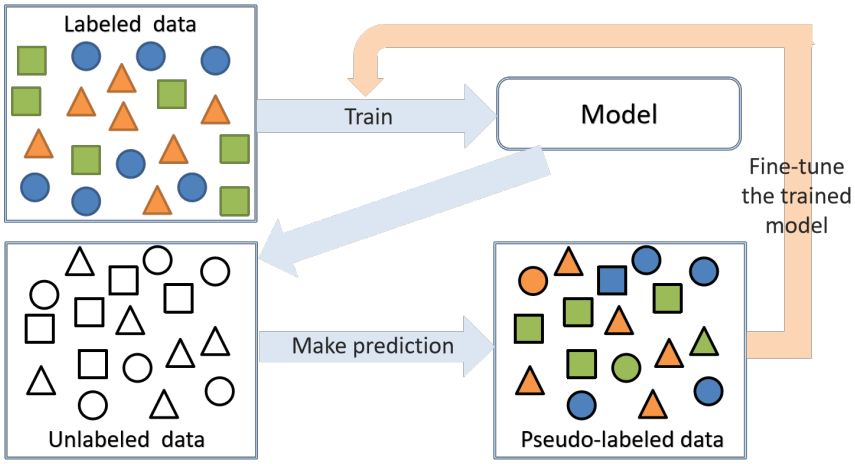


Figure 5. The illustration of the self-training method. The model trained by the labeled data is fine-tuned by the pseudo-labeled data selected from the unlabeled data.

A classifier is initially trained on the labeled dataset X_L to obtain the prediction on the unlabeled dataset X_U. Then, some pseudo-labeled measured data X_{pre} are selected, and in the self-training process, the measured data with the pseudo-labels and labeled synthetic data $X_{pre} \cup X_L$ are used to train the classifier together. Since the classifier has seen the measured data with the pseudo-labels and labeled synthetic data, the boundaries of the trained classifier are modified to fit the measured data better, which can also align the feature of the same categories in synthetic and measured domains.

When the feature alignment classification network $\mathcal{F}$ has been well trained and given an unlabeled measured sample $\mathbf{x}_u$, the predicted pseudo-label $\hat{\mathbf{y}}_u$ can be obtained:

$$\hat{\mathbf{y}}_u = \phi(\mathbf{x}_u; \boldsymbol{\vartheta}) = [\hat{y}_{u1}, \hat{y}_{u2}, \ldots, \hat{y}_{uK}] \tag{5}$$

$$p_{pred} = max\{\hat{y}_{u1}, \hat{y}_{u2}, \ldots, \hat{y}_{uK}\} \tag{6}$$

p_{pred} indicates the probability of the sample $\mathbf{x}_u$ belonging to the predicted category.

The threshold value γ is set and some pseudo-labeled data are selected as the X_{pre} to fine-tune the whole network. For a pseudo-labeled sample ($\mathbf{x}_u$, $\hat{\mathbf{y}}_u$), if $p_{pred} > \gamma$, it is selected; otherwise, it is not selected. We use the predicted soft labels as the pseudo-labels for fine-tuning, which means that the classification loss $\mathcal{L}_C$ calculated on the pseudo-labeled dataset X_{pre} can be expressed as follows:

$$\mathcal{L}_C(X_{pre}) = E_{X_{pre}}[\sum_{i=1}^{K} -\hat{y}_{ui} \log \widetilde{y}_{ui}] \tag{7}$$

where $\widetilde{\mathbf{y}}_u = [\widetilde{y}_{u1}, \widetilde{y}_{u2}, \ldots, \widetilde{y}_{uK}]$ is the newly predicted label for sample $\mathbf{x}_u$ after fine-tuning. The pseudo-labels are constantly updated.

In the fine-tuning process, the total loss function can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_C(X_L) + \lambda \mathcal{D}(X_L, X_U) + \mathcal{L}_C(X_{pre}) \tag{8}$$

we can see that in the self-training process, the classification loss is calculated on the X_L and X_{pre}.

The performance of the self-training method mainly depends on the initially trained classifier. We design the feature alignment classification network $\mathcal{F}$ as the initially trained

classifier for a better recognition result. However, the pseudo-labeled dataset selected by the threshold γ is not entirely accurate and reliable. The correctness of pseudo-labels is also related to the recognition performance of the test data. Since the recognition result of the test data is not perfectly correct, the pseudo-labels of the test data are also not perfectly accurate. In addition to the pre-set threshold γ, we consider a new criterion when selecting pseudo-labeled data for fine-tuning.

3.4. Pseudo-Label Denoising Based on Image Similarity

In the fine-tuning process, we consider combining the image information to select reliable pseudo-labels. Inspired by Ref. [28], which combines the CNN feature and image similarity for classification, and Ref. [55], which also considers pseudo-label denoising to keep the predicted labels consistent for the K nearest neighbors in the pixel-wise classification, in our study, we use the image similarity and k-Nearest Neighbor (KNN) [60] algorithm to eliminate some pseudo-labeled data, and the process is shown in Figure 6.

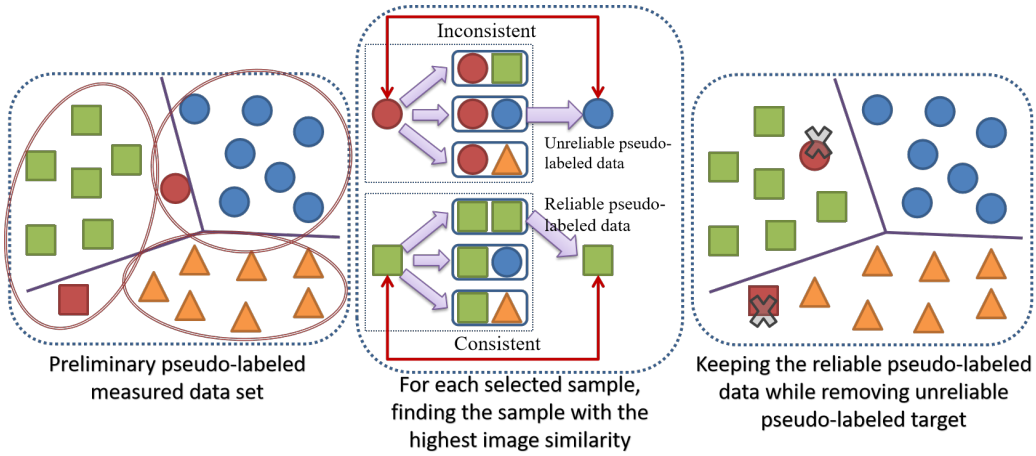


Figure 6. Illustration of pseudo-label denoising method. Different colors (except for red) are predicted categories according to the learned classification boundaries. Different shapes represent real categories. The red color means mispredicted categories, while the samples in the same circle represent the relationship with relatively higher similarity.

The distribution difference between the synthetic and measured data may be reflected in background distribution or different features, such as target shape, gray value, etc. Thus, different aspects of image features can be reflected in different similarity measures, and the measured samples with high similarities are more likely to have the same labels. According to this assumption, we design the pseudo-label denoising method to keep the label consistent between the measured test samples with high similarities.

Figure 6 (left) shows that there are some mispredicted samples (of red color) in the preliminary pseudo-labeled dataset X_{pre}, and the samples in the same circle represent the relationship with relatively higher similarity by similarity measure. Figure 6 (middle) shows that the predicted pseudo-labels are guaranteed to be consistent between the two samples with the highest similarity (i.e., the sample with its KNNs with k = 1). If consistent, keep the pseudo-labeled sample and else eliminate it. Thus, the mispredicted samples can be eliminated according to the comparisons of the similarities and their predicted pseudo-labels as shown in Figure 6 (right). The pseudo-label denoising method is introduced as follows (see Algorithm 1).

There are five image similarities used in Ref. [28]; however, in our paper, we only use three because the other two depend on the results of image segmentation. Thus, we use cosine similarity (COSS), normalized mutual information similarity (NMIS), and structural similarity (SSIM). We briefly introduce them as follows. More details about the similarities can be found in Ref. [28]. x_1 and x_2 represent the two images.

Algorithm 1 Pseudo-label denoising method procedure.

Input: The threshold value γ.
Output: The pseudo-labeled dataset X_{deno} after denoising.
Step 1: The preliminary pseudo-label dataset X_{pre} can be obtained according to the pre-set threshold γ. Initialize X_{deno} as $\emptyset$.
Step 2: Calculate the similarity matrix S of every two samples in X_{pre}.

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$

where n is the number of samples in the X_{pre}.
Step 3:
for t =1 to n **do**
1) Make $s_{tt} = 0$;
2) The pseudo-label is represented as y_t for the sample x_t. For x_t, the pseudo-label of the sample with the highest image similarity in the similarity matrix is found and represented as y_{tmax} according to S;
3) Check whether the pseudo-labels y_{tmax} and y_t are consistent;
if consistent **then**, add the tth sample into the denoised data set X_{deno};
end if
end for
Step 4: The denoised pseudo-labeled dataset X_{deno} and its corresponding labels are obtained after checking all the samples in the X_{pre}.

(1) COSS:

$$\mathrm{COSS}(x_1, x_2) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} \times b_{ij}}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} (a_{ij})^2} \times \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} (b_{ij})^2}} \tag{9}$$

where a_{ij} and b_{ij} represent the pixel values of x_1 and x_2 at coordinate (i, j), respectively.

(2) NMIS:

$$\mathrm{NMIS}(x_1, x_2) = \frac{H(x_1) + H(x_2)}{2 \times H(x_1, x_2)} \tag{10}$$

where $H(\cdot)$ and $H(\cdot, \cdot)$ represent information and joint information entropy, respectively.

(3) SSIM:

$$\mathrm{SSIM}(x_1, x_2) = \frac{2\mu_1\mu_2 + C_1}{\mu_1^2 + \mu_2^2 + C_1} \cdot \frac{2\delta_1\delta_2 + C_2}{\delta_1^2 + \delta_2^2 + C_2} \cdot \frac{\delta_{12} + C_3}{\delta_1\delta_2 + C_3} \tag{11}$$

where μ_1 or μ_2, δ_1^2 or δ_2^2, and δ_{12} represent the mean, variance, and covariance of the two images, respectively. C_1, C_2, and C_3 are the constants set according to Ref. [28].

4. Experimental Results

4.1. Experimental Dataset

The SAMPLE dataset contains a total of 10 types of data, each of which contains pairs of measured and synthetic data. The original image size is 128 × 128, and the resolution is 0.3 m × 0.3 m. The depression angles of the synthetic and measured data are from 14° to 17°. More details can be found in Ref. [24]. The data types and numbers of the original SAMPLE dataset are introduced in Table 1, and the synthetic samples are one-to-one matched with the measured samples. This experimental setting is called "Training Scenario I".

Table 1. Training Scenario I: data types and numbers of samples in original SAMPLE dataset.

	Synthetic (Training)	**Measured (Testing)**
Depression Angles / **Types**	14°–17°	14°–17°
2S1	174	174
BMP2	107	107
BTR70	92	92
M1	129	129
M2	128	128
M35	129	129
M548	128	128
M60	176	176
T72	108	108
ZSU23	174	174
Total	1345	1345

Figure 7 shows one (measured, synthetic) pair from each type. We can clearly see that the synthetic images lack background clutter, which is also mentioned in the literature [23,24]. Additionally, the strong scattering centers are somewhat different for the synthetic and measured data.

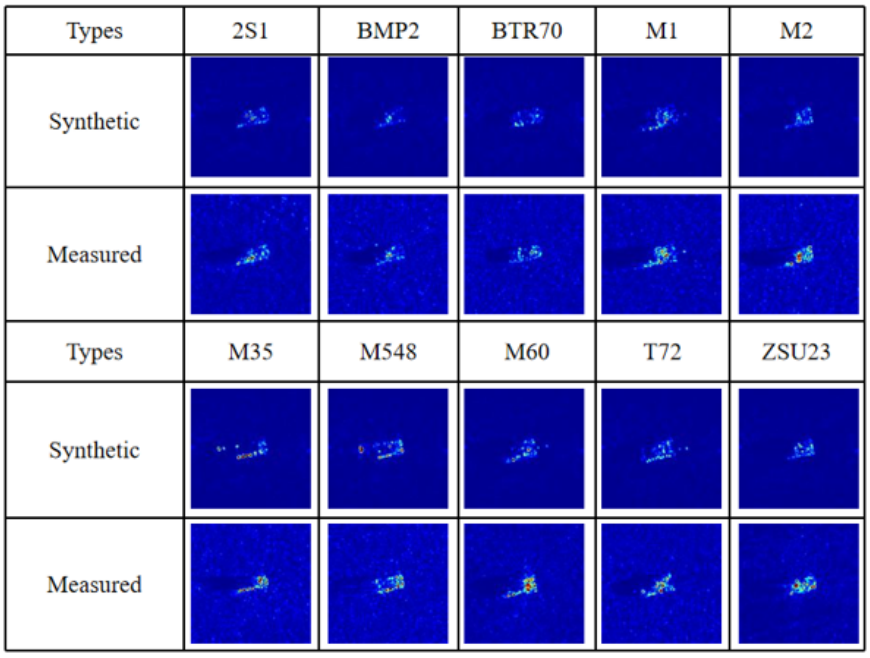


Figure 7. Synthetic data and corresponding measured data of SAMPLE dataset.

We also consider the experimental setting in Ref. [25] in our paper. The depression angles of the synthetic data range from 14° to 16° and the depression angle of the measured data is 17°. Compared with "Training Scenario I", there is a difference in depression angle for synthetic and measured data. Table 2 shows the specific data types and numbers of samples for training and testing, and this experimental setting is called "Training Scenario II".

The software and hardware environment of the experimental model training and testing is Intel(R) Xeon(R) CPU E5-2643 v4 @3.40 GHz, 128 GB memory, Nvidia GeForce GTX 1080ti with 11G RAM. The operating system is Ubuntu 16.04.1 and the development environment is Python 2.7.12 and TensorFlow 1.8.0.

Table 2. Training Scenario II: data types and numbers of samples for training and testing in [25].

Types \ Depression Angles	Synthetic (Training) 14°–16°	Measured (Testing) 17°
2S1	116	58
BMP2	55	52
BTR70	43	49
M1	78	51
M2	75	53
M35	76	53
M548	75	53
M60	116	60
T72	56	52
ZSU23	116	58
Total	806	539

Our work explores the problem of training fully on the synthetic data and testing on the measured data. Therefore, training data refers to synthetic data and test data refers to measured data. We explore the effectiveness of our framework under two training scenarios. In the following experiments:

(1) for the image preprocessing, we crop the data to the size of 64 × 64 to exclude the influence of the background, and then we normalize them by min-max scaling to the range [0, 255].
(2) for the network parameters, we use the grid search [61] method to explore the parameters. We provide a list of candidate values for the key parameters, and their final values are determined through experiments. The final parameter combination has a batch size value of 256, a dropout value of 0.2, and a λ value of 10. We set the total epoch number to 500. We use the Adam [62] optimizer with a learning rate of 10^{-4}.
(3) as for the presented experimental results, we carry out each group of experiments by 20 times, and we also provide the lowest and highest recognition rates, represented by "Min" and "Max", respectively. "Ave" provides the average recognition rates of 20 experiments with the standard deviation.

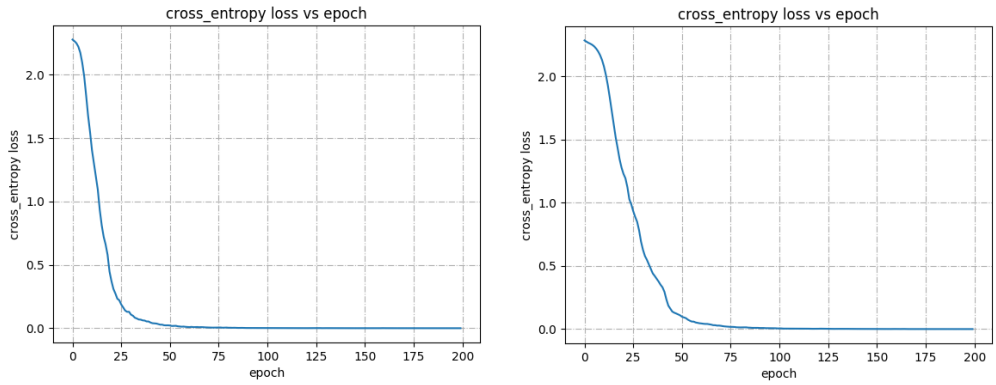
4.2. Recognition Results of Basic Recognition Network

We perform the experiments that utilize the basic recognition network introduced in Section 3.1 for the two training scenarios without feature alignment and self-training. Table 3 shows the experimental results. Furthermore, to check the convergence property of our designed basic recognition network structure under two training scenarios, the curves of the loss value versus the number of epochs are shown in Figure 8, and the epoch number is set to 200 when training the basic recognition network.

When we carry out the experiments under Training Scenario II, the average recognition results are lower and display more apparent fluctuation. As mentioned before, the training and test data of Training Scenario I are paired, and the quantities of training data and test data under Training Scenario I are larger. Thus, it is not suitable to directly compare the recognition results of these two recognition scenarios. As shown in Figure 8, the curve of Figure 8a is smoother, and there is almost no oscillation compared to Figure 8b, and the difference to obtain Figure 8a,b is the training data, which also illustrates that Training Scenario II is more complicated. Nevertheless, the final convergence values of Figure 8a,b are similar, and the experimental results are similar under both training scenarios. The following validation experiments are carried out under Training Scenario II since the experiments are faster with a smaller amount of training data.

Table 3. Recognition results under different training scenarios (%).

Training Scenarios	Min	Max	Ave
Training Scenario I	77.84	81.78	79.83 ± 1.23
Training Scenario II	75.70	81.82	79.32 ± 1.79



(**a**) (**b**)

Figure 8. Vertical and horizontal axes denote the classification loss value and the number of iterations for training, respectively. (**a**) Training Scenario I. (**b**) Training Scenario II.

Then, we observe the experimental results under different data augmentation methods, namely by adding Gaussian noise [25], speckled noise [7], and translational augmentation [6,7]. Adding noise may address the distribution gap problem because the synthetic data lack background clutter.

In Ref. [25], to add Gaussian noise $\mathcal{N}$ to a training image x, a standard deviation (std) α is set, and the augmented image can be expressed as $x_{gaus} = x + \mathcal{N}$ (mean = 0, std = α). In our paper, we set the α range as $\alpha = [0.1, 0.2, 0.3, 0.3, 0.4, 0.4, 0.5, 0.5, 0.6]$, and thus the training data is augmented by 9 times. In Ref. [7], to apply speckle noise to a training image x, the truncated exponential distribution s, which imposes that the maximum intensity of noise samples does not exceed a given parameter a, is used, and the augmented image can be expressed as $x_{spec} = \tilde{x} \times s$. $\tilde{x}$ is the image after applying a median filter on the x. The a range is set as $a = [0.5, 0.5, 0.5, 1.0, 1.0, 1.0, 1.5, 1.5, 1.5]$, and thus the training data is also augmented by 9 times. For translational augmentation, given a training image x, the translated image x' can be written as $x' = x_{u,v}$, which illustrates that all pixels in x are shifted left by u pixels and shifted down by v pixels. x' has the same size as x. To guarantee that the target does not exceed the image boundary, the absolute values of u and v are smaller than 20. Figure 9 gives an illustration of the three data augmentation operations.

We also conduct the experiments on the basic recognition network introduced in Section 3.1 under Training Scenario II, and we augment the training data using the three data augmentation methods. Table 4 provides the recognition results.

Speckle and Gaussian noise augmentation are both helpful for improving recognition rates. The improvement in recognition performance after adding speckle noise is better than after adding Gaussian noise, with a higher average recognition rate and a minor standard deviation. The literature [63] also mentions that adding speckle noise for augmentation is more effective than adding Gaussian noise. Moreover, we can see that translational augmentation does not help to improve recognition accuracy. The literature [25] also considers rotation augmentation; however, the experimental results show that rotation augmentation is not helpful for improving recognition performance [25]. This type of augmentation method, which includes translation, rotation, or their combination, may not be suitable in this scenario, and may even lead to an overfitting problem with unsatisfactory results.

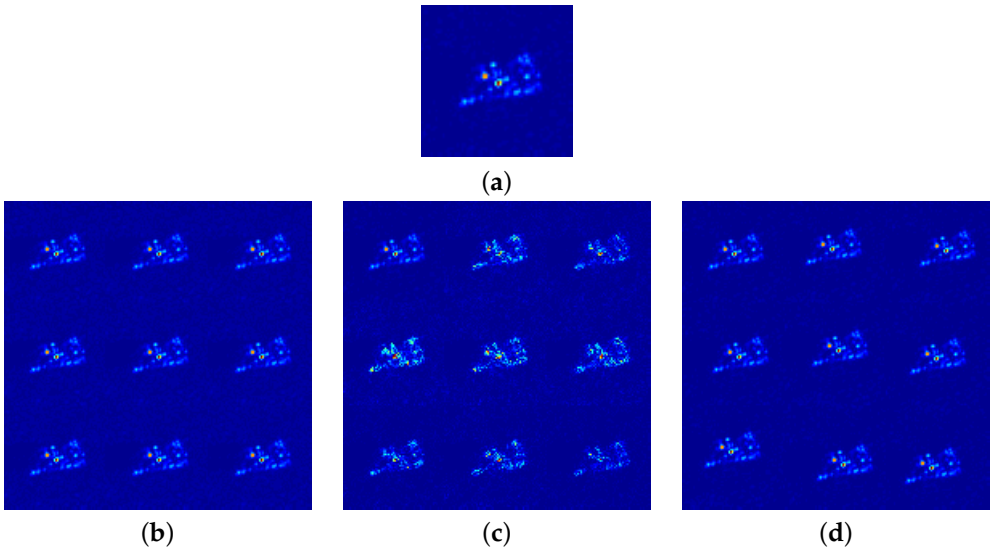


Figure 9. Illustration of data augmentation operations. (**a**) Image of 2S1 category. (**b**) Gaussian noise augmentation. (**c**) Speckle noise augmentation. (**d**) Translation augmentation.

Table 4. Training Scenario II: recognition results of different data augmentation methods (%).

Augmentation Methods	Min	Max	Ave
Without augmentation	75.70	81.82	79.32 ± 1.79
Gaussian noise	76.62	84.60	81.36 ± 2.09
Speckle noise	79.78	86.27	82.29 ± 1.89
Translation augmentation	61.97	71.61	66.53 ± 2.63

Therefore, we consider using speckle noise augmentation in our following experiments, which provides better results.

4.3. Results of Gradual Domain Adaptation Recognition Framework

This subsection discusses the recognition results of our proposed gradual domain adaptation recognition framework. It is worth mentioning that pseudo-label denoising is not carried out in this subsection. Because the gradual domain adaptation recognition framework is a two-stage training process, we discuss the effectiveness of both the feature alignment and self-training, represented by "feature alignment" and "self-training", respectively. "Baseline" gives the result of the basic recognition network introduced in Section 3.1 with speckle noise augmentation of the training data. Tables 5 and 6 show the experiment results under Training scenarios I and II, respectively.

Tables 5 and 6 show that the proposed gradual domain adaptation recognition framework is applicable under Training scenarios I and II settings. Furthermore, the recognition results show incremental improvements. We use the t-distributed stochastic neighbor embedding (t-SNE) [64] method, which can visualize high-dimensional data by mapping them into low-dimensional space, to further illustrate the effectiveness of our framework and compare the feature distributions extracted by the gradual domain adaptation classification network. We map the original vectorized images and features into a 2D space, and Figure 10 shows the visualization results.

Table 5. Training Scenario I: recognition results in different stages of gradual domain adaptation recognition framework (%).

Methods	Min	Max	Ave
Baseline	78.07	85.80	81.57 ± 2.19
Baseline + feature alignment	91.00	94.42	92.65 ± 1.11
Baseline + feature alignment + self-training	92.05	96.73	94.94 ± 1.46

Table 6. Training Scenario II: recognition results in different stages of gradual domain adaptation recognition framework (%).

Methods	Min	Max	Ave
Baseline	79.78	85.16	82.29 ± 1.89
Baseline + feature alignment	88.68	92.58	90.09 ± 1.10
Baseline + feature alignment + self-training	92.76	98.70	95.71 ± 1.93

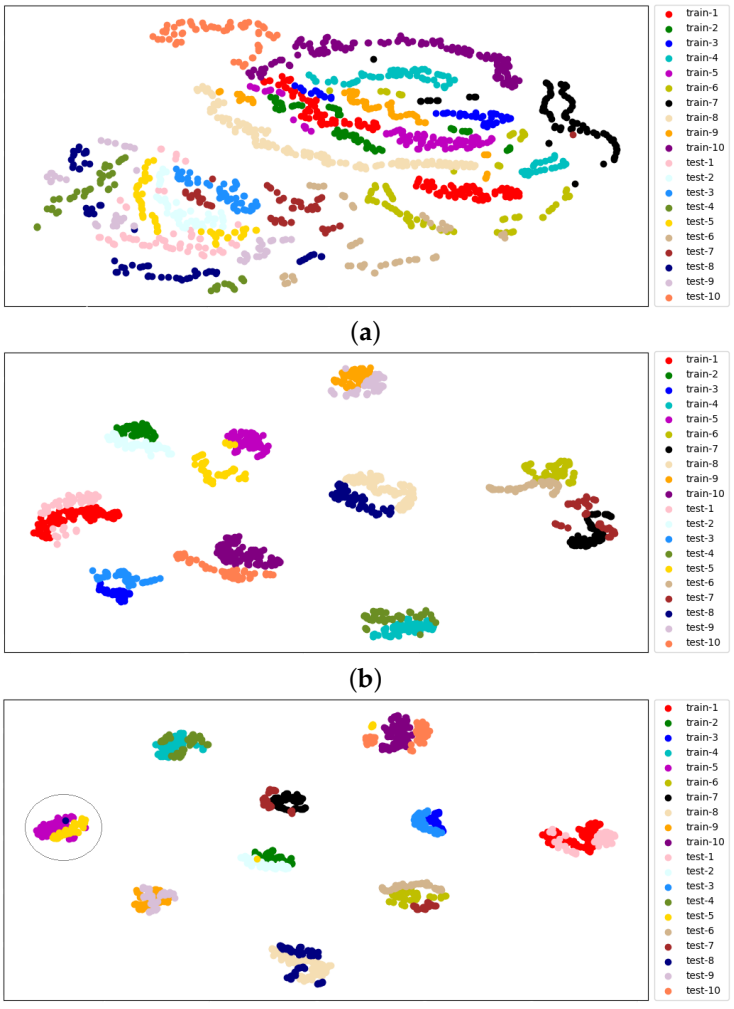


Figure 10. T-SNE visualization results of data distribution under Training Scenario II. (**a**) Original data distribution. (**b**) Feature distribution after feature alignment classification. (**c**) Feature distribution after feature alignment and self-training.

The t-SNE visualization results also illustrate that our proposed gradual domain adaptation framework is effective. Figure 10a shows that there is a distribution gap between the original measured and synthetic data. Most of the training data are already distinguishable after the feature alignment classification, as shown in Figure 10b. However, some categories are not well separated, and the feature distribution of some categories is not compact enough. As shown in Figure 10c, after feature alignment and self-training, the feature distribution of different categories is more compact than we expect. Still, we can see that some samples are in the clusters of other classes (a sample of the test-8 category is in the train-5 cluster, which is shown in the grey circle of Figure 10c).

Thus, in the following subsection, pseudo-label denoising is considered in the self-training process for a better recognition result.

4.4. Effectiveness of Pseudo-Label Denoising Method

We performed pseudo-label denoising in the self-training process. In this subsection, we use three similarity measures, namely COSS, NMIS, and SSIM, for pseudo-label denoising according to Algorithm 1. Tables 7 and 8 show the experimental results of the gradual domain adaptation recognition framework with pseudo-label denoising under Training scenarios I and II, respectively.

Table 7. Training Scenario I: recognition results of gradual domain adaptation recognition framework with pseudo-label denoising using different similarities (%).

Similarity	Min	Max	Ave
COSS	94.28	97.92	96.25 ± 0.78
NMIS	92.93	96.95	95.69 ± 1.30
SSIM	92.04	99.18	96.46 ± 1.14

Table 8. Training Scenario II: recognition results of gradual domain adaptation recognition framework with pseudo-label denoising using different similarities (%).

Similarity	Min	Max	Ave
COSS	92.57	100	96.59 ± 2.36
NMIS	91.84	100	96.87 ± 2.20
SSIM	95.73	98.70	97.36 ± 0.87

Using different similarity measures for pseudo-label denoising is helpful for improvements in recognition performance. SSIM performs better with higher average recognition rates in Training scenarios I and II. Here, we also present the confusion matrices of the pseudo-labeled dataset before and after denoising to illustrate the effectiveness of pseudo-label denoising, which are shown in Figure 11.

We can see that in the pseudo-labeled dataset without denoising, there are some mispredicted samples obtained by the feature alignment classification network. Figure 11b–d show that after the pseudo-label denoising, the correctness of the pseudo-labeled data has improved, and all the false pseudo-labeled data are eliminated when using the SSIM similarity for denoising. Mispredicted pseudo-labeled samples are the main factors that affect the further improvement of recognition rates, and label denoising can improve recognition performance. These results are also consistent with the experimental results in Tables 7 and 8, which illustrate that SSIM performs better. Furthermore, in the following experiments, the final results are shown using pseudo-label denoising using SSIM.

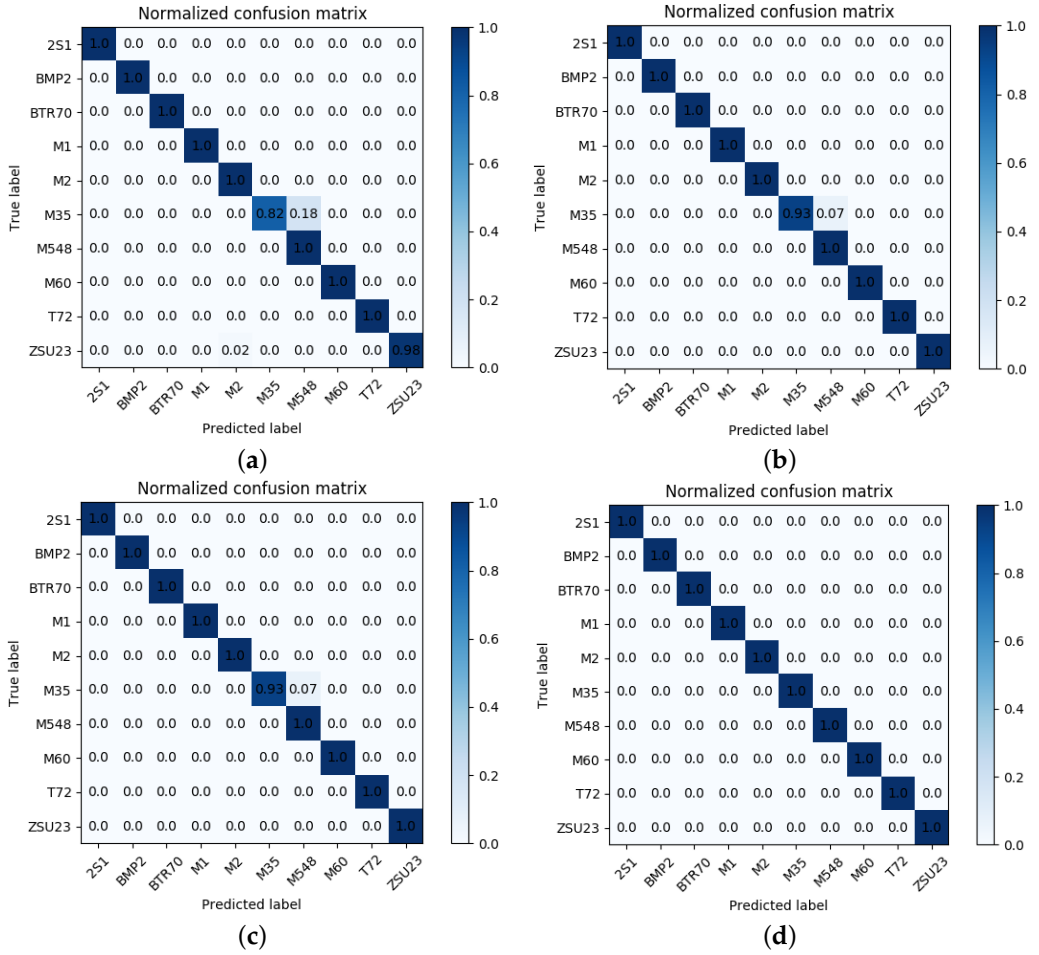


Figure 11. Confusion matrices of the pseudo-labeled dataset. (**a**) Before label denoising. (**b**) Label denoising using COSS. (**c**) Label denoising using NMIS. (**d**) Label denoising using SSIM.

4.5. Comparisons with Other Methods

In this subsection, we compare our framework with other existing methods using the SAMPLE dataset under the two experimental settings, given in Tables 9 and 10, respectively. More experimental results of different CNN architectures can be found in [25,31], and we only present the best recognition results achieved in [25,31] as the comparison.

Table 9. Training Scenario I: comparisons with other methods (%).

Methods	Min	Max	Ave
Lewis et al. (2019) [65]	-	84.8	82.05
Araujo et al. (2022) [32]	-	-	91.3
Dong et al. (2021) [30]	-	-	93.66
Melzer et al. (2021) [31]	96.88	96.88	96.88
Our framework	92.04	99.18	96.46 $\pm$ 1.14

Currently, more studies are carried out under Training Scenario I, and the best recognition accuracy is achieved by neural architecture searching [31]. However, neural architecture searching is computationally expensive and usually difficult to achieve in real-world systems. The recognition results of our framework under Training Scenario I reach a relatively high level of accuracy. In addition, under Training Scenario II, our framework exceeds the results presented in the literature [25], which utilized an ensemble of five models. Other research works use the SAMPLE dataset for recognition with different experimental settings. For

example, in Ref. [29], except for the labeled synthetic data, one or two measured samples per class are also used for training, and their proposed augmentation technique can achieve 95.1% classification accuracy, and our experimental results still outperform it. Thus, our proposed method has achieved the state-of-the-art level.

Table 10. Training Scenario II: comparisons with other methods (%).

Methods	Min	Max	Ave
Lewis et al. (2019) [24]	-	-	24.97
Scarnati et al. (2019) [23]	-	-	55.62 [1]
Inkawhich et al. (2021) [25]	93.32	96.66	95.06 ± 0.87
Our framework	95.73	98.70	97.36 ± 0.87

[1] The exact recognition result of [23] is not presented in the original paper. We refer to [32] and give the experimental result here.

We further illustrate the effectiveness of the whole framework through the confusion matrices of the test data in Figure 12, which are obtained at different training stages under Training Scenario II. Figure 12a,b gives the results of the basic recognition network with or without data augmentation operation. Figure 12c gives the result of the feature alignment classification network with speckle noise augmentation. Moreover, Figure 12d gives the final result of the whole framework with pseudo-label denoising.

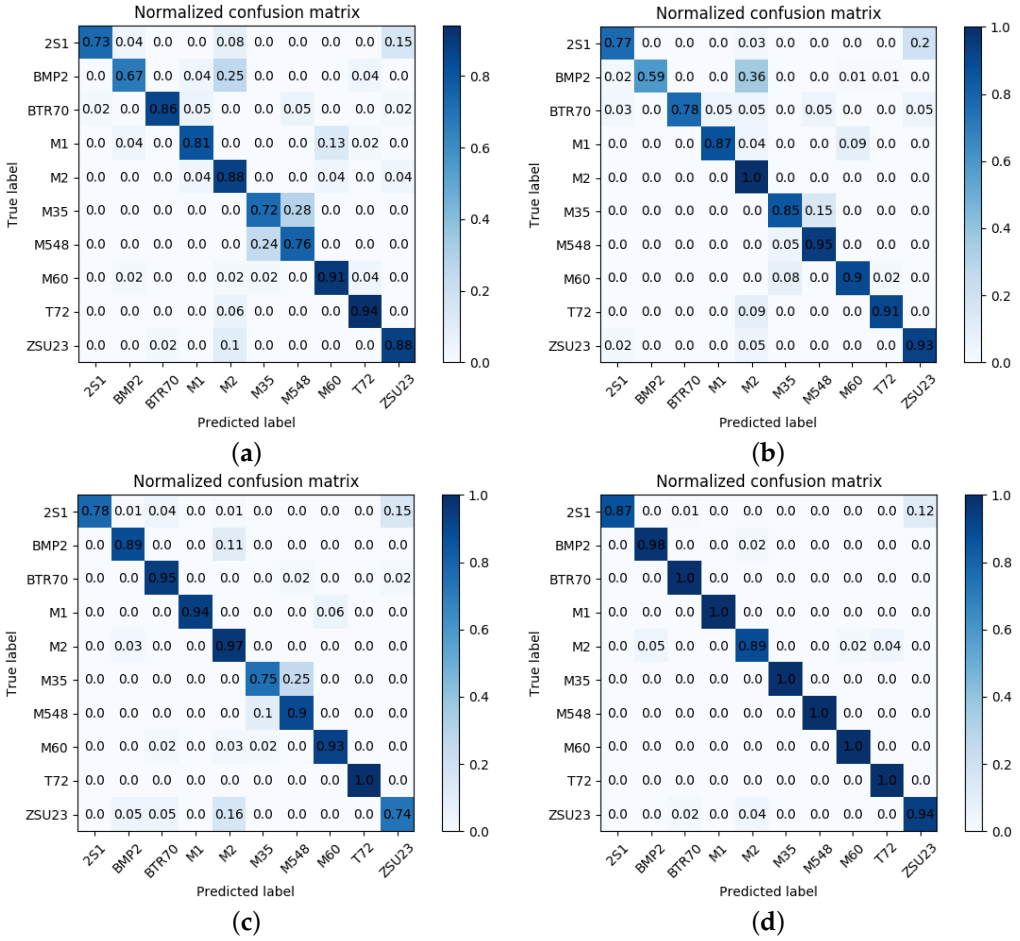


Figure 12. Confusion matrices of test data under Training Scenario II. (**a**) Without data augmentation. (**b**) With speckle noise augmentation. (**c**) With speckle noise augmentation and feature alignment. (**d**) With speckle noise augmentation, feature alignment, self-training, and pseudo-label denoising (SSIM similarity).

Firstly, we can see that the recognition results are progressively better. Secondly, as seen in Figure 12a–c, the M35, 2S1, and M548 categories are hard to recognize. Thirdly, comparing Figure 12b,c, feature alignment may cause the features of M35 and M548 to become closer. Fourthly, comparing Figure 12c,d, we can see that the denoised pseudo-labeled dataset improves the recognition results effectively. Therefore, all the methods of our framework work together to obtain the final results.

4.6. Ablation Analysis of Our Proposed Method

Because our overall framework combines many different kinds of methods, such as data augmentation, feature alignment between domains, self-training, and the pseudo-label denoising method, we consider the ablation experiments in this subsection to verify the effectiveness of these methods (Table 11). We perform the ablation analysis under Training Scenario II.

Table 11. Ablation analysis (%).

Speckle Noise Augmentation	Feature Alignment	Self-Training	Pseudo-Label Denoising	Ave
✓				82.29 ± 1.89
	✓			82.52 ± 1.22
		✓		87.87 ± 2.37
✓	✓			90.09 ± 1.10
✓		✓		82.81 ± 2.85
	✓	✓		86.60 ± 4.06
✓	✓	✓		95.71 ± 1.93
✓	✓	✓	✓	97.36 ± 0.87

By ablation analysis, we can observe that when any method of our designed framework is removed, the performance degrades, verifying that (1) these methods, which are speckle noise augmentation, feature alignment, and self-training are all effective for improving the performance; (2) the utilization of self-training usually leads to a higher standard deviation since the effectiveness of self-training methods mainly depends on the initially trained classifier and the size of the original training dataset, and the larger size of the original dataset makes the pseudo-labeled test data a smaller proportion in the fine-tuning process. Thus, we use the feature alignment classification network obtained by the speckle noise augmented training data, which can provide more stable classification performance; (3) pseudo-label denoising scheme can lead to a better result, with a higher and more stable recognition rate. We can see that the whole framework works together to obtain the final result.

5. Discussion

We also discuss some limitations of the proposed method here. For the problem of training fully on the synthetic data and testing on the measured data considered in this paper, we need to know the categories of synthetic and real data in advance. However, in the real-world recognition scenario, it may be difficult to know the categories of real data in advance. Considering the proposed framework, self-training is a crucial step in our proposed framework, and it relies on the initially trained classifier. We have tried different basic recognition network structures, based on which the proposed feature alignment and self-training with pseudo-label denoising strategies can also improve recognition performance, but the final experimental results are not so good. Therefore, we proposed a feature alignment classification network for a better-initialized recognition result and a pseudo-label denoising method to eliminate the falsely pseudo-labeled data. In addition, there is still room for improving the pseudo-label denoising method, and some pre-set parameters are manually designed, which can be optimized. As a whole, seen from the experimental results, the proposed gradual domain adaptation framework has indeed achieved incremental improvements.

6. Conclusions

In our paper, we design a gradual domain adaptation recognition framework with a pseudo-label denoising method when using only synthetic data for training. The experiments are carried out based on the newly published SAMPLE dataset. Two training scenarios are considered to eliminate the controversy over the completeness of the experimental dataset. The experimental results demonstrate the effectiveness of our framework. For Training Scenario I, the framework matches the result of neural architecture searching and achieves 96.46% average accuracy. For Training Scenario II, the framework outperforms the results of other existing methods and achieves 97.36% average accuracy. Moreover, pseudo-label denoising effectively reduces the fluctuation of experimental results to gain more stable results. In the future, when training fully on the synthetic SAR data and testing on the real SAR data, more complex recognition scenarios can be considered, for example, reducing the amount of training data. We will evaluate the performance of our proposed method based on other SAR datasets and consider further improving the stableness of recognition results. While the pseudo-label denoising method can also be optimized and more similarity measures or combinations can be considered for pseudo-label denoising. To conclude, the recognition methods when training fully on the synthetic SAR data and testing on the real SAR data are worth further exploration. In the real-world recognition system, the real SAR data is demanding to obtain, and it has the application potential to utilize synthetic data for recognition.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S.; software, Y.S.; validation, Y.S., Y.W., C.Z. and S.W.; investigation, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, Y.S. and Y.W.; visualization, Y.S.; supervision, Y.W., H.L. and L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61671354, in part by the stabilization support of National Radar Signal Processing Laboratory under Grant KGJ202206, and in part by the 111 Project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank all reviewers and editors for their comments on this paper.

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

References

1. Malmgren-Hansen, D.; Kusk, A.; Dall, J.; Nielsen, A.; Engholm, R.; Skriver, H. Improving SAR automatic target recognition models with transfer learning from simulated data. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 1484–1488. [CrossRef]
2. Cha, M.; Majumdar, A.; Kung, H.; Barber, J. Improving SAR automatic target recognition using simulated images under deep residual refinements. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15 April 2018; pp. 2606–2610.
3. Zhang, W.; Zhu, Y.; Fu, Q. Semi-supervised deep transfer learning-based on adversarial feature learning for label limited SAR target recognition. *IEEE Access* **2019**, *7*, 152412–152420. [CrossRef]
4. Huang, Z.; Pan, Z.; Lei, B. Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. *Remote Sens.* **2019**, *9*, 907. [CrossRef]
5. Sun, Y.; Wang, Y.; Liu, H.; Wang. N; Wang. J. SAR target recognition with limited training data based on angular rotation generative network. *IEEE Geosci. Remote. Sens. Lett.* **2020**, *17*, 1928–1932. [CrossRef]
6. Chen, S.; Wang, H.; Xu, F.; Jin, Y.Q. Target classification using the deep convolutional networks for SAR images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. [CrossRef]
7. Ding, J; Chen, B.; Liu, H.; Huang, M. Convolutional neural network with data augmentation for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. [CrossRef]
8. Dong, G.; Wang, N.; Kuang, G. Sparse representation of monogenic signal: With application to target recognition in SAR images. *IEEE Signal Process. Lett.* **2014**, *13*, 952–956.
9. Park, J.I.; Kim, K.T. Modified polar mapping classifier for SAR automatic target recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 1092–1107. [CrossRef]

10. Lin, Z.; Ji, K.; Kang, M.; Leng, X.; Zou, H. Deep convolutional highway unit network for SAR target classification with limited labeled training data. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 1091–1095. [CrossRef]
11. Wang, S.; Wang, Y.; Liu, H.; Sun. Y. Attribute-guided multi-scale prototypical network for few-shot SAR target classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 12224–12245. [CrossRef]
12. Cui, Z.; Zhang, M.; Cao, Z.; Cao, C. Image data augmentation for SAR sensor via generative adversarial nets. *IEEE Access* **2019**, *7*, 42255–42268. [CrossRef]
13. Guo, Q.; Xu, F. A deep feature transformation method based on differential vector for few-shot learning. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Brussels, Belgium, 11–16 July 2021; pp. 407–410.
14. Fu, K.; Zhang, T.; Zhang, Y.; Wang, Z.; Sun, X. Few-shot SAR target classification via metalearning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
15. Gao, F.; Yue, Z.; Wang, J.; Sun, J.; Yang E.; Zhou, H. A novel active semisupervised convolutional neural network algorithm for SAR image recognition. *Comput. Intel. Neurosc.* **2017**, *2017*, 3105053. [CrossRef] [PubMed]
16. Wang, C.; Huang, Y.; Liu, X.; Pei, J.; Zhang, Y.; Yang, J. Global in Local: A convolutional transformer for SAR ATR FSL. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
17. Gao, F.; Shi, W.; Wang, J.; Hussain, A.; Zhou, H. A semi-supervised synthetic aperture radar image recognition algorithm based on an attention mechanism and bias-variance decomposition. *IEEE Access* **2019**, *7*, 108617–108632. [CrossRef]
18. Wang, C.; Gu, H.; Su, W. SAR image classification using contrastive learning and pseudo-labels with limited data. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
19. Zhou, Y.; Jiang, X.; Li, Z.; Liu, X. SAR target classification with limited data via data driven active learning. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Kuala Lumpur, Malaysia, 17–21 July 2022; pp. 2475–2478.
20. Wang, C.; Shi, J.; Zhou, Y.; Yang, X.; Zhou, Z.; Wei, S.; Zhang, X. Semisupervised learning-based SAR ATR via self-consistent augmentation. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 4862–4873. [CrossRef]
21. Liu, X.; Huang, Y.; Wang, C.; Pei, J.; Huo, W.; Zhang, Y.; Yang, J. Semi-supervised SAR ATR via conditional generative adversarial network with multi-discriminator. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Brussels, Belgium, 11–16 July 2021; pp. 2361-2364.
22. LLiu; Pan, Z.; Qiu, X.; Peng, L. SAR target classification with CycleGAN transferred simulated samples. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Valencia, Spain, 22–25 July 2018; pp. 4411–4414.
23. Scarnati, T.; Lewis, B. A deep learning approach to the synthetic and measured paired and labeled experiment (SAMPLE) challenge problem. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXVI, Baltimore, MD, USA, 14–18 May 2019; Volume 10987, pp. 29–38.
24. Lewis, B.; Scarnati, T.; Sudkamp, E.; Nehrbass, J.; Rosencrantz, S.; Zelnio, E. A SAR dataset for ATR development: the synthetic and measured paired labeled experiment (SAMPLE). In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXVI, Baltimore, MD, USA, 14–18 May 2019; Volume 10987, pp. 39–54.
25. Inkawhich, N.; Inkawhich, M.; Davis, E.; Majumder, U.; Tripp, E.; Capraro, C.; Chen, Y. Bridging a gap in SAR-ATR: training on fully synthetic and testing on measured data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2942–2955. [CrossRef]
26. Lei, Y.; Xia, W.; Liu, Z. Synthetic images augmentation for robust SAR target recognition. In Proceedings of the International Conference on Video and Image Processing (ICVIP), Guangzhou, China, 22 December 2021; pp. 19–25.
27. He, Q.; Zhao, L.; Ji, K.; Kuang, G. SAR target recognition based on task-driven domain adaptation using simulated data. *IEEE Geosci. Remote. Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
28. Zhang, C.; Wang, Y.; Liu, H.; Sun, Y.; Hu, L. SAR target recognition using only simulated data for training by hierarchically combining CNN and image similarity. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
29. Sellers, S.; Collins, P.; Jackson, J. Augmenting simulations for SAR ATR neural network training. In Proceedings of the IEEE International Radar Conference (RADAR), Washington DC, USA, 27 April–1 May 2020; pp. 309–314.
30. Dong, G.; Liu, H. A hierarchical receptive network oriented to target recognition in SAR images. *Pattern Recognit.* **2022**, *126*, 108558. [CrossRef]
31. Melzer, R.; Severa, W.; Plagge, M.; Plagge, M.; Vineyard, C. Exploring characteristics of neural network architecture computation for enabling SAR ATR. In Proceedings of the Automatic Target Recognition XXXI, Online Only, 12 April 2021; Volume 11729, pp. 1–20.
32. Araujo, G.; Machado, R.; Pettersson, M. Non-cooperative SAR automatic target recognition based on scattering centers models. *Sensors* **2022**, *22*, 1293. [CrossRef] [PubMed]
33. Leng, L.Z.X.; Feng, S.; Ma, X.; Ji, K.; Kuang, G.; Liu, L. Domain knowledge powered two-stream deep network for few-shot SAR vehicle recognition. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]
34. Camus, B.; Monteux, E.; Vermet, M. Refining simulated SAR images with conditional GAN to train ATR algorithms. In Actes de la Conférence on Artificial Intelligence for Defense (CAID), Rennes, France, 17–19 November 2020; pp. 160–167.
35. Song, Q.; Chen, H.; Xu, F.; Cun, T. EM simulation-aided zero-shot learning for SAR automatic target recognition. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1092–1096. [CrossRef]
36. Sun, Y.; Wang, Y.; Hu, L.; Liu, H. SAR target recognition using simulated data by an ensemble multi-scale deep domain adaption recognition framework. In Proceedings of the CIE International Conference on Radar, Haikou, China, 15 December 2021.

37. The Air Force Moving and Stationary Target Recognition Database. Available online: https://www.sdms.afrl.af.mil/datasets/mstar/ (accessed on 10 March 2011).
38. Pan, S.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**; *22*, 1345–1359. [CrossRef]
39. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the International Conference on Artificial Neural Networks, Kuala Lumpur, Malaysia, 11 November 2018; pp. 270–279.
40. Wang, M.; Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing* **2018**, *312*, 135–153. [CrossRef]
41. Wilson, G.; Cook, D. A survey of unsupervised deep domain adaptation. *ACM Trans. Intell. Syst. Technol* **2020**, *11*, 46. [CrossRef]
42. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:1412.3474.
43. Sun, B.; Saenko, K. Deep CORAL: correlation alignment for deep domain adaptation. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8 October 2016; pp. 443–450.
44. Long, M.; Cao, Y.; Wang, J.; Jordan, M. Learning transferable features with deep adaptation networks. In Proceedings of the International Conference on Machine Learning (ICML), Lile, France, 6–11 July 2015; pp. 97–105.
45. Li, Y.; Wang, N.; Shi, J.; Hou, X.; Liu, J. Adaptive batch normalization for practical domain adaptation. *Pattern Recognit.* **2018**, *80*, 109–117. [CrossRef]
46. Carlucci, F.; Porzi, L.; Caputo, B.; Ricci, E.; Bulo, S. AutoDIAL: Automatic domain alignment layers. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5077–5085.
47. Saito, K.; Ushiku, Y.; Harada, T. 2017. Asymmetric tri-training for unsupervised domain adaptation. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6 August 2017; pp. 2988–2997.
48. French, G.; Mackiewicz, M.; Fisher, M. Self-ensembling for visual domain adaptation. *arXiv* **2018**, arXiv:1706.05208.
49. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *JMLR* **2016**, *17*, 1–35.
50. Shu, R.; Bui, H.; Narui, H.; Ermon, S. A DIRT-T approach to unsupervised domain adaptation. In Proceedings of the International Conference of Learning Representation (ICLR), Vancouver, BC, Canada, 30 April 2018.
51. Schmarje, L.; Santarossa, M.; Schröder, S.; Koch, R. A survey on semi-, self-and unsupervised learning for image classification. *IEEE Access* **2021**, *9*, 82146–82168. [CrossRef]
52. Jing, L.; Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell* **2021**, *43*, 4037–4058. [CrossRef]
53. Pise, N.; Kulkarni, P. A survey of semi-supervised learning methods. In Proceedings of the 2008 International Conference on Computational Intelligence and Security, Washington, DC, USA, 13–17 December 2008; Volume 2, pp. 30–34.
54. Wang, C.; Shi, J.; Zhou, Y.; Li, L. Label noise modeling and correction via loss curve fitting for SAR ATR. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–10. [CrossRef]
55. Hua, W.; Wang, S.; Hou, B. Semi-supervised learning for classification of polarimetric SAR images based on SVM-wishart. *J. Radars* **2015**, *4*, 93–98.
56. Zhao, J.; Guo, W.; Liu, B.; Zhang, Z.; Yu, W.; Cui, S. Preliminary exploration of SAR image land cover classification with noisy labels. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 3274–3277.
57. Shang, R.; Lin, J.; Jiao, L.; Li, Y. SAR image segmentation using region smoothing and label correction. *Remote Sens. Mar.* **2020**, *12*, 803.
58. Hinton, G.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
59. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* **2014**, *15*, 1929–1958.
60. Harrington, P. *Machine Learning in Action*; Manning Publications: Greenwich, CT, USA, 2012; ISBN 9781638352457.
61. Larochelle, H.; Erhan, D.; Courville, A.; Bergstra, J.; Bengio, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In Proceedings of the International Conference on Machine Learning (ICML), Corvallis, OR, USA, 20–24 June 2007; pp. 473–480.
62. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
63. Zhu, X.; Montazeri, S.; Ali, M.; Hua, Y.; Bamler, R. Deep learning meets SAR: Concepts, models, pitfalls, and perspectives. *IEEE Geosc. Rem. Sen. M.* **2021**, *9*, 143–172. [CrossRef]
64. Maaten, L.v.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
65. Lewis, B.; DeGuchy, O.; Sebastian, J.; Kaminski, J. Realistic SAR data augmentation using machine learning techniques. In Proceedings of the Algorithms for Synthetic Aperture Radar Imagery XXVI, Baltimore, MA, USA, 14 May 2019; Volume 10987, pp. 12–28.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.