



## Article

# An Adaptive Focal Loss Function Based on Transfer Learning for Few-Shot Radar Signal Intra-Pulse Modulation Classification

Zehuan Jing , Peng Li \*, Bin Wu , Shibo Yuan and Yingchao Chen

School of Electronic Engineering, Xidian University, Xi'an 710071, China; jingzehuan@stu.xidian.edu.cn (Z.J.); bwu@xidian.edu.cn (B.W.); yuanshibo@stu.xidian.edu.cn (S.Y.); 19021110339@stu.xidian.edu.cn (Y.C.)

\* Correspondence: penglixid@xidian.edu.cn; Tel.: +86-137-0919-3246

**Abstract:** To solve the difficulty associated with radar signal classification in the case of few-shot signals, we propose an adaptive focus loss algorithm based on transfer learning. Firstly, we trained a one-dimensional convolutional neural network (CNN) with radar signals of three intra-pulse modulation types in the source domain, which were effortlessly obtained and had sufficient samples. Then, we transferred the knowledge obtained by the convolutional layer to nine types of few-shot complex intra-pulse modulation classification tasks in the target domain. We propose an adaptive focal loss function based on the focal loss function, which can estimate the parameters based on the ratio of hard samples to easy samples in the data set. Compared with other existing algorithms, our proposed algorithm makes good use of transfer learning to transfer the acquired prior knowledge to new domains, allowing the CNN model to converge quickly and achieve good recognition performance in case of insufficient samples. The improvement based on the focal loss function allows the model to focus on the hard samples while estimating the focusing parameter adaptively instead of tediously repeating experiments. The experimental results show that the proposed algorithm had the best recognition rate at different sample sizes with an average recognition rate improvement of 4.8%, and the average recognition rate was better than 90% for different signal-to-noise ratios (SNRs). In addition, upon comparing the training processes of different models, the proposed method could converge with the least number of generations and the shortest time under the same experimental conditions.

**Keywords:** intra-pulse modulation classification; convolutional neural network; transfer learning; adaptive focal loss



**Citation:** Jing, Z.; Li, P.; Wu, B.; Yuan, S.; Chen, Y. An Adaptive Focal Loss Function Based on Transfer Learning for Few-Shot Radar Signal Intra-Pulse Modulation Classification. *Remote Sens.* **2022**, *14*, 1950. <https://doi.org/10.3390/rs14081950>

Academic Editor: Danilo Orlando

Received: 8 March 2022

Accepted: 15 April 2022

Published: 18 April 2022

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



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

## 1. Introduction

Intra-pulse modulation classification of radar signals is an essential area within the field of electronic countermeasures (ECM), which determine the system, usage, and type of enemy radar by analyzing the data received from radar reconnaissance systems [1]. Generally, the classification of intra-pulse modulation can be divided into feature-based and data-based classification [2]. Feature-based classification needs to extract features from the radar signal and carefully design classifiers based on these features [3], while data-based classification can fully retain the data of the radar signal, and is the focus of this paper.

Deep learning is a new research direction in the field of machine learning and is a collective term for a class of pattern analysis methods [4]. With the rapid development of deep learning [4], many researchers now use deep learning to study the classification of intra-pulse modulation of radar signals. Compared to traditional feature extraction, deep learning can automatically extract depth features from radar signals. In [5], Z Huang proposed a deep convolutional neural network-based approach that used the amplitude information of single-polarization SAR images as input to automatically extract the hierarchical spatial

characteristics. These features may be more abstract, but they are representative and suitable for classification. Deep learning models can learn all the representation layers together at the same time. By learning the common features, once the model modifies an internal feature, all other features that depend on it will automatically make the corresponding adaptation. To improve the classification accuracy of radar signals, various methods have been proposed. Many researchers convert one-dimensional (1D) radar signals into two-dimensional (2D) time-frequency images for classification. For example, short-time Fourier transformation and CNNs were used to identify six different intra-pulse modulations signals, and the overall classification success rate was over 90% [6]. Ma, XR et al. [3] designed a combination of the short-time Ramanujan Fourier transform and pseudo-Zernike moments invariant feature-based method to recognize different modulation schemes under different parameter variation conditions. To improve classification performance under a low signal-to-noise ratio (SNR), an improved convolutional de-noising automatic encoder [7] was proposed. When the SNR was  $-9$  dB, the encoder could classify 12 kinds of modulated signals, and the classification accuracy was over 95%. Gao, LP et al. [8] proposed an image fusion algorithm using non-multi-scale decomposition to fuse images of a single signal with different time-frequency methods. Liu et al. [9] proposed an algorithm for radar emitter signal recognition transforming raw radio signals into time-frequency image using the Choi–Williams distribution function. Moreover, various studies show that using one-dimensional radar signals for identification makes sense, which is also the focus of this paper. Sun, J et al. [10] designed a novel encoding method to generate high-dimension sequences of equal length as new features in cases of inconsistent features between samples, and proposed a unidimensional convolutional neural network to classify the encoded high-dimension radar signals. Li, X et al. [11] proposed an attention-based approach for radar emitter classification using recurrent neural networks to classify the radar signals. To improve the classification accuracy of radar signals with different SNRs of  $-14\sim 20$  dB, a novel network was proposed, which combines a shallow convolutional neural network (CNN), a long-term memory network (LSTM), and a deep neural network (DNN). Wu, B et al. [12] proposed a novel 1D CNN with an attention mechanism to extract more discriminative features and recognize radar emitter signals. Although methods such as deep CNN can omit feature engineering and automatically extract and learn features from the data [5], at the same time, the number of samples required restricts the development of deep learning in the field of radar signal classification. Most researchers only focus on signal classification under different SNRs. These methods fail to overcome the obstacle of training the depth network with limited radar signals, which is a few-shot recognition problem [13].

To solve these issues, we propose a novel deep network based on transfer learning. Transfer learning is good at applying knowledge or patterns learned in one domain or task to a different but related domain or problem [14]. In general, transfer learning can be classified into three categories: instance-based, feature-based, and shared parameter-based [15]. Instance-based transfer learning studies how to select instances from the source domain that are useful for training in the target domain. For example, an effective assignment of weight to labeled data instances in the source domain can make the distribution of instances in the source and target domains close, so that a reliable learning model with high classification accuracy can be built in the target domain. Dai WY et al. [16] proposed the TrAdaBoost algorithm to improve the classification effect by adjusting the weights of misclassified samples in the source and target domains. Feature-based approaches extract and identify representative features shared between the source and target domains and then use these features to transfer knowledge [17]. Shared parameter-based methods investigate how to find common parameters or prior distributions between the spatial models of source and target data [18]. For few-shot Synthetic Aperture Radar (SAR) image classification, shared parameter-based methods are good at migrating labeled data or learned knowledge structures from related domains with sufficient samples [18–22]. Huang, Z et al. [18] designed an assembled CNN architecture consisting of a classification

pathway and a reconstruction pathway, together with an additional feedback bypass. A novel method, deep memory convolution neural networks, for alleviating the problem of overfitting caused by insufficient SAR image samples was proposed in [19]. Rostami, M et al. [22] proposed a novel deep neural network for classifying SAR images that eliminates the need for a huge labeled dataset.

Different from the classification of 2D SAR images, we recognize the received 1D radar signals instead of converting them into images via time-frequency analysis. First, we trained a 1D deep CNN using three large numbers of simple radar signals with labels as the source dataset, and the source task was to classify the three radar signals as correctly as possible. We could easily obtain an optimal depth CNN, because the number of samples was simple and sufficient, and then discard the classifier and higher convolution layers, leaving only the structure and parameters of the lower layer to transfer to the target domain. The lower convolutional layers of the CNN (those layers closer to the input) extracted more general features, while some higher layers of classifier and convolutional layers were applied to specific features, with the higher layers containing more feature semantics and the lower layers containing less feature semantics but more location information. Meanwhile, we proposed an adaptive focus loss function based on focal loss [23], which can adjust the parameter according to the ratio of hard samples to easy samples in the data set. The experimental results demonstrate that, compared with existing algorithms, the proposed algorithm had significantly improved classification accuracy and convergence speed while using less training data.

The remainder of the paper proceeds as follows. Section 2 is concerned with the methodology used for this study. Detailed experimental procedures and discussion of the results are given in Section 3. Section 4 is the conclusion of this paper.

## 2. Methods

### 2.1. Related Work

#### 2.1.1. Convolutional Neural Network

Deep learning arose rapidly in the first decade of the 21st century as computing power increased. The Convolutional Neural Network (CNN), the exemplar of deep learning, was first established by Y. L. Cun [24], who designed the famous LeNet-5 to classify handwritten numbers, drawing on artificial neurons and visual perception mechanisms. CNNs share many similarities with ordinary neural networks, in that both of them mimic the structure of human nerves and consist of neurons with learnable weights and bias constants [24]. However, CNNs are more widely used because they avoid complex pre-processing of data and can directly input raw data relying on convolution layers to extract feature maps [25]. In the following years, CNNs evolved based on their classical structure. In 2012 Geoffrey and his student Alex designed the Alex network [26], introducing a nonlinear activation function based on LeNet (ReLU) and a method to prevent overfitting (Dropout, Data augmentation). In 2014, K. S et al. [27] proposed VGG-Net, which contains more layers and uses the same size convolutional filter. The Inception structure of GoogLeNet [28] allows the entire network structure to be expanded in both width and depth. ResNet [29] proposes a residual learning framework that reduces the training burden on the network.

#### 2.1.2. Radar Signal Intra-Pulse Module Classification

Radar emitter signal identification, which aims to obtain information concerning radar systems by analyzing the emitter signals, is an important aspect of electronic warfare and has been extensively studied by numerous researchers. All the CNNs mentioned in Section 2.1.1 have achieved good results in many fields, so it is reasonable to use CNNs to learn temporal correlations and deep features from radar signals for classification. Wei, SJ et al. [30] used sequences in the time, frequency, and autocorrelation domains of the original signal as inputs to a shallow CNN, after which the deep features extracted by the CNN were used as the input to an LSTM network, and finally, a DNN, as the classification network, would directly output the modulation type of the signal. This could achieve high

accuracies for four common kinds of measured radar signals. In [9], Z. Liu created a deep CNN using the input of time-frequency spectrums of radar signal intra-pulse modulation to substitute manually constructed features that are time-demanding and neglect delicate characteristics. In [31], Y. Pan et al. used the Hilbert–Huang transform to obtain a wealth of information on the nonlinear and non-stationary properties of radar signals and built a deep residual network to avoid the degradation problem.

### 2.1.3. Transfer Learning

Transfer learning is a machine learning technique that can transfer knowledge learned in the source domain to the target domain for enhancing the learning of the target task. Transfer learning typically includes the following elements: a source domain  $D_S$ , a target domain  $D_T$ , a source learning task  $T_S$ , and a target learning task  $T_T$ . Based on the differences between the source/target domains and tasks, S.J. Pan et al. classified transfer learning into inductive transfer learning, unsupervised transfer learning, and transductive transfer learning [15]. The above three types of transfer learning can be further grouped into four cases: instance transfer learning, feature-representation transfer learning, parameter transfer learning, and relational-knowledge transfer learning [15]. Transfer learning has been successfully adopted in many fields, such as image and video quality, visual categorization, and machinery fault diagnosis. Ling, S et al. [32] surveyed state-of-the-art transfer learning algorithms in visual categorization applications such as object recognition, image classification, and human action recognition. Varga, D et al. [33] pre-trained different types of CNNs based on fusing the decisions of multiple image quality scores that can better characterize authentic image distortion and effectively estimate perceived image quality. In [34], an ImageNet database pre-trained CNN with global average pooling layers was proposed in order to transfer the learned knowledge so that the module can be easily generalized to any input image size and pre-trained CNNs. In [35], Li, C et al. reviews the research progress on deep transfer learning for machinery fault diagnosis in recent years. In the field of radar target classification, Huang, ZL et al. [18] used inductive transfer learning to transfer reconstructed knowledge from convolutional self-encoders to the SAR target classification task. They innovatively used a large number of unlabeled SAR scene images to train the convolutional self-encoder to reconstruct the features well and transferred only the encoder results during the target classification task. Qing, W et al. [36] designed a two-channel CNN combined with bi-directional LSTM architecture to improve the classification performance of the waveforms of cognitive passive radar. They used transductive transfer learning to initialize the target domain classifier with source domain parameters. In this paper, since the source and target domains have different but related distributions, we used parameter transfer learning, in which the source and target tasks share some of the network parameters.

### 2.1.4. The Focal Loss Function

The imbalance of different sample categories in target detection is a critical issue impacting accuracy. Online hard example mining (OHEM) [37], a typical algorithm to deal with class imbalance, increases the weight of misclassified samples but ignores the easy-to-classify samples. Focal loss is proposed in [23] to solve the category imbalance problem, and is obtained by modifying the standard cross-entropy loss. Compared to OHEM, the focal loss function allows the model to focus more on difficult samples by reducing the weights of easy-to-classify samples during the training process [23]. Currently, focal loss is commonly used in different fields. In the area of text detection in computer vision, X. Tian et al. [38] designed a focal text detection network that uses focal loss to train the network well with an uneven number of samples; it could obtain better performance when the number of samples was insufficient. In the field of medical image processing, where sample imbalance is a serious problem, [39] proposed a network framework using residual neural networks (Res-Net) [29] combined with focal loss for determining left ventricle segmentation from cardiac MRI images. For the problem of ship detection in

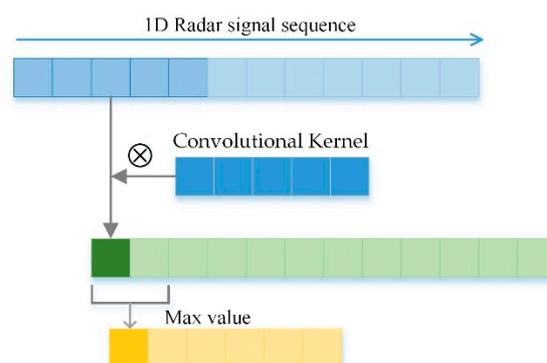
high-resolution SAR images, ref. [40] designed a RetinaNet-Plus method based on the RetinaNet network, which uses focal loss in the training process to resolve class imbalance and reduce the loss weights of easy-to-classify samples.

## 2.2. The Proposed Methods

### 2.2.1. Transfer Learning-Based Convolutional Neural Network

Based on parameter transfer learning, we constructed the network frameworks for the source and target tasks separately, and the source domain used three simple intra-pulse modulation types of radar signals, which were effortlessly obtained and had sufficient samples. The goal was to classify these three radar signals as accurately as possible. In the target domain, we trained with a small number of complex intra-pulse modulation type signals, nine in total, and initialized the convolutional layers of the target domain network using the parameters learned in the source domain instead of random initialization. In the following, we describe the details of the method.

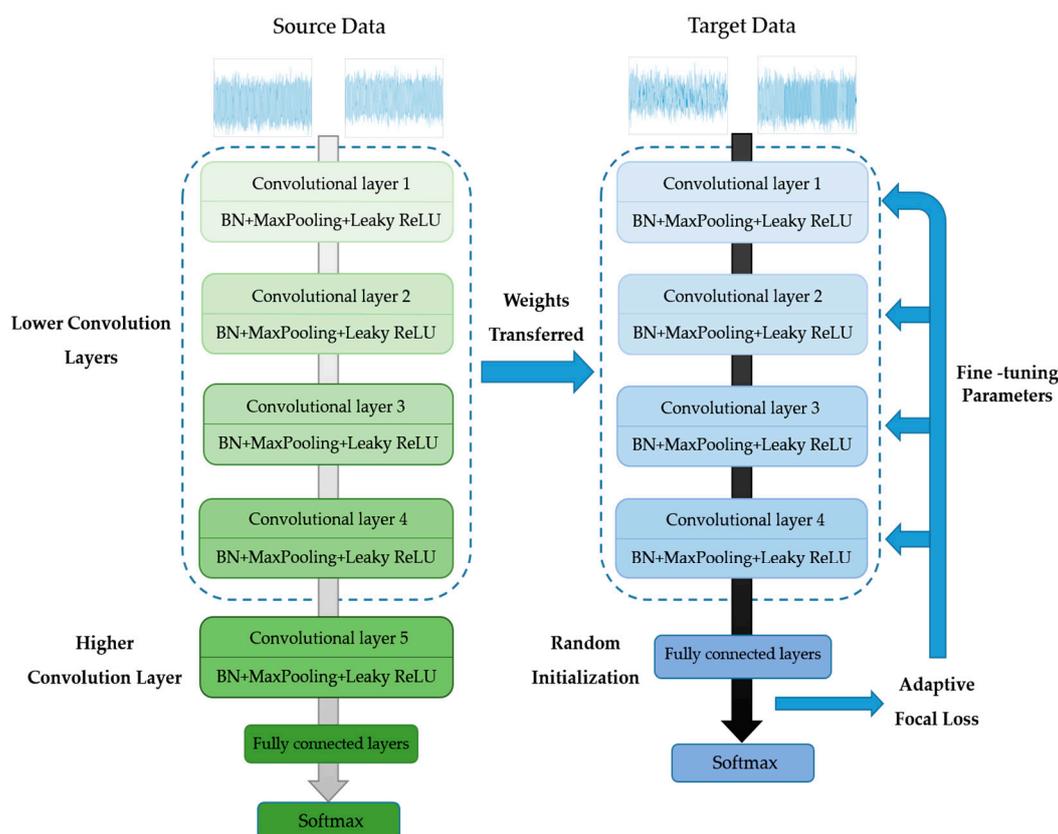
A sequential structure is a convolutional neural network in which the output of each layer is superimposed sequentially as the input of the next layer. Because of its simple structure, it has been widely used as a classical structure. VGG is a typical sequential structured network model, first proposed by K. Simonyan et al. [27]. They added convolutional layers to AlexNet one by one to study the effect of network depth on the recognition effect. Experiments showed that the deeper the network, the better the recognition effect; when the network structure increased to 16 and 19 layers, the effect improved significantly, and therefore these algorithms were called VGG-16 and VGG-19. Therefore, to keep the training simple and the model effortlessly transferred and tuned, we used a sequential structure and simplified the VGG network to build the source and target networks. To ensure that the layer parameters could transfer properly between the source and target domains, we construct a sequential structure with part of the same convolutional layers in the source and target domains. We designed a 1D CNN for the input 1D radar signal intra-pulse modulated sequence; the receptive field of the 1D convolution kernel was continuously translated over the data sequence to observe significant features due to the translation-invariant property of convolution. Each convolution filter of the convolution layer acted iteratively throughout the receptive field to convolve the input sequence, and the convolution result formed a feature map of the input sequence containing the local features of the radar signals. Each convolution filter shared the same parameters, including the same weight matrix and bias term, which were transferred to the target domain after the training of the source domain. The convolved 1D feature map was fed to the pooling layer, and maximum pooling was used for sampling, which was a nonlinear down-sampling method [41]. After acquiring radar signal sequence features by convolution, directly using all the extracted feature data to train the classifier for classification usually expends great computational effort, so the maximum pooling sampling method can be used to down-sample the convolutional features. The convolution and pooling process is shown in Figure 1.



**Figure 1.** Convolution and pooling process.

By training the source domain network, we continuously optimized the feature extraction capability of the convolutional layers for 1D radar signals. Thus, these pre-trained convolutional layers could extract data features well in the face of complex target tasks with few-shot samples and inconsistent sample distribution.

To better assemble and train the network and highlight the effect of transfer learning, we designed a single-input, single-output convolutional network structure based on the above structure, as shown in Figure 2.



**Figure 2.** The architecture of the one-dimensional convolutional neural network with adaptive focal loss function based on transfer learning.

As shown in Figure 2, the source network consisted of five convolutional layers, each followed by batch normalization to prevent gradient disappearance and speed up training. To transfer the parameters properly between networks, we kept the structure of the lower convolutional layers of the source the same as that of the target network, with the same number and size of convolutional kernels. Table 1 shows the number of feature maps and the size of the convolutional kernels for each convolutional layer. We used multiple  $1 \times 3$  (or  $1 \times 5$ ) convolutional kernels instead of large-sized kernels to minimize the number of parameters and amount of computational effort while ensuring that the perceptual field of view was not altered [42]. As the number of layers deepened, we increased the number of convolutional kernels to extract more deep features and then down-sampled the feature maps via maximum pooling to prevent overfitting. This design gave the network an inverted triangular shape, i.e., the closer to the input layer, the smaller the number of parameters, and the closer to the output layer, the larger the number of parameters. Such an inverted triangular structure prevents the neural network from losing gradients too quickly during backpropagation [27]. In terms of activation function selection, the Rectifier Linear Unit (ReLU) is widely used because it can solve the gradient disappearance problem [43], but its sparsity tends to lead to dying ReLU, so we used Leaky

ReLU, which assign a non-zero slope to all negative values after each layer of convolution, to solve this problem [44,45].

**Table 1.** The design scheme for the source and target networks.

	Layer	Kernel Size	Channel	Max Pooling
Source Network	Conv1	$1 \times 5$	32	$1 \times 2$
	Conv2	$1 \times 5$	64	$1 \times 2$
Target Network	Conv3	$1 \times 5$	128	$1 \times 4$
	Conv4	$1 \times 3$	256	$1 \times 4$
Source Network	Conv5	$1 \times 3$	256	$1 \times 4$

In ref. [46], Y, W. et al. proved that different convolutional layers extract different-level feature information, and therefore that during the transfer learning process the appropriate convolutional layer parameters should be selected for transfer, instead of all of them. The lower convolutional layers of CNN (the layers closer to the input) extract more general features, i.e., the lower layers contain few feature semantics but have more location information, while some of the higher classifier and convolutional layers apply to specific features which extract more feature semantics, and the semantic features learned in the last few layers are quite different for different datasets. Therefore, the higher convolutional layers are generally related to task objectives and classification, and the lower convolutional layers are more suitable as feature extractors to extract general features for transfer learning. As a result, in the transfer process, we only kept the first four convolutional layers of the source network and discarded the other layers, which contained more semantic information. These optimized lower convolutional layers were more general and could effectively extract structural and detailed features in the radar signal, even in the face of new data.

### 2.2.2. Adaptive Focus Loss Function (AFL)

To further improve the classification performance of radar signals under few-shot learning, we replaced the original cross-entropy loss with adaptive focal loss, which could automatically adjust its application range depending on the number of samples determined via focal loss.

The focal loss function is widely used and has achieved good results in the field of target detection. The authors proposed focal loss (FL) based on cross-entropy (CE) loss:

$$CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1 - p) & \text{otherwise.} \end{cases} \quad (1)$$

where  $p \in [0, 1]$  is the model's estimated probability and  $y \in \{-1, 1\}$  indicates the ground-truth class. Facing the problem of sample imbalance, the authors added a factor to the CE loss that assigns different weights to the samples:

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t) \quad (2)$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise.} \end{cases} \quad (3)$$

The focusing parameter  $\gamma \geq 0$ . When  $\gamma = 0$ , the focal loss is the traditional cross-entropy loss, and when  $\gamma$  increases, the modulation coefficient also increases.  $\gamma$  smoothly adjusts the proportion of loss accounted for by samples of different difficulties. If a hard sample is misclassified, the  $p_t$  value is small:

$$p_t \rightarrow 0 \quad (4)$$

$$(1 - p_t)^\gamma \rightarrow 1 \quad (5)$$

and focal loss has not changed significantly compared to the original loss. By contrast, when the easily classified samples are correctly classified,  $p_t \rightarrow 0$ , and the contribution

to the total loss is small. Based on this principle, focal loss solves most of the sample imbalance problems very well, but in some scenarios, its effect is not ideal.

Focal loss solves the sample imbalance problem by adjusting the parameter  $\gamma$  to give more weight to the hard samples with poor classification. However, the value of  $\gamma$  is not as large as possible. In ref. [23], the best recognition was achieved when  $\gamma = 2$ , and the performance decreased when  $\gamma > 2$ . For other models, the optimal value of  $\gamma$  needs to be determined through a large number of experiments. We can simplify the determination of the optimal parameters by estimating the range of  $\gamma$ . Therefore, we propose an adaptive focus loss function that estimates the value of  $\gamma$  based on the ratio of hard- to easy-to-classify samples. According to the research [47], there is a huge difference in quantity between easy and hard samples during the training process. First, we trained a base classifier using CE, then we predicted the training set and counted the numbers of easy and hard samples, denoted  $N_e$  and  $N_h$ , respectively (For radar signal intra-pulse modulation classification, we considered  $p_t \leq 0.1$  to be a hard sample and  $p_t \geq 0.9$  to be an easy sample. For different models and problems, one can change the judgment threshold of the hard and easy samples). According to the focal loss function, the loss gap is:

$$\frac{loss_{hard}}{loss_{easy}} = \frac{(1 - 0.1)^\gamma}{(1 - 0.9)^\gamma} = 9^\gamma \quad (6)$$

We defined the difficulty of the training set as the ratio of the number of easy samples to the number of hard ones:

$$r = \frac{N_e}{N_h} \quad (7)$$

The focusing parameter  $\gamma$  was used to adjust the contribution of easy and difficult samples to the overall loss to balance their large quantitative differences. Therefore, the loss gap should not be less than the ratio of the number of easy samples to the number of hard ones, and the value of  $\gamma$  should increase as  $r$  increases, i.e., the more simple samples, the greater the focus on hard samples. Then we derived the estimate of  $\gamma$  as  $\hat{\gamma}$ , which should satisfy the following:

$$9^{\hat{\gamma}} \geq r = \frac{N_e}{N_h} \quad (8)$$

$$\hat{\gamma} \geq \log_9 \frac{N_e}{N_h} \quad (9)$$

In summary, for the multi-classification problem of radar signal intra-pulse modulation, we propose the Adaptive Focus Loss function (AFL) as follows:

$$AFL = -(1 - p_{prediction} \times y_{groundtruth})^{\hat{\gamma}} \log(p_{prediction}) \quad (10)$$

In this paper, we obtained  $\hat{\gamma} = \log_9 \frac{N_e}{N_h}$ , and  $p_{prediction}$  was a  $1 \times 9$  vector, which was our model's estimated probability for the nine radar signals classified.  $y_{groundtruth}$  was the vector of the labels after one-hot encoding.

### 3. Experiments and Results

In this section, we simulated several radar signal datasets with different sample sizes to simulate different small sample cases, which were used to train and test the proposed method and other baseline methods. In all experiments, we used a computer equipped with an Intel 10900K CPU, 64 GB of RAM, and a RTX 3070 GPU.

#### 3.1. Dataset and Parameters Setting

Generally, the typical radar signal is dominated by high-power radio frequency (RF) pulses with a carrier band range from 3 MHz to 100 GHz. The radar receiver in our simulation used a local oscillator to mix with the high-frequency radar signal to reduce the frequency of the received signal, and then output a lower-frequency signal through

the intermediate frequency (IF) amplifier. Specifically, to ensure that the frequency of the received signal was reduced to replicate a radar system operating in a real environment, a mixer was simulated. The mixer multiplied the RF signal by the local oscillator signal to obtain two output frequencies, the summation and the subtraction of the radio frequency  $f_{RF}$  and the local oscillator frequency  $f_{LO}$ , which can be expressed as:

$$f_{RF} + f_{LO} \quad (11)$$

$$f_{RF} - f_{LO} \quad (12)$$

Through using a low-pass filter, the summed frequency  $f_{RF} + f_{LO}$  could be well suppressed, so that we could obtain the subtracted frequency  $f_{RF} - f_{LO}$ , which is the IF signal. In this paper, we simulated the low frequency radar signals from the receiver output and used them to train and test our proposed method. For the source domain dataset, we selected three modulation types of simple and widely obtainable radar signals: single-carrier frequency (SCF) signals, linear frequency modulation (LFM) signals, and sinusoidal frequency modulation (SFM) signals. For the target domain dataset, we used nine different kinds of radar signals with complex modulation types comprising binary phase-shift keying (BPSK) signals, binary frequency-shift keying (BFSK) signals, quadrature frequency-shift keying (QFSK) signals, Frank phase-coded (Frank) signals, even quadratic frequency modulation (EQFM) signals, dual-frequency modulation (DLFM) signals, multiple linear frequency modulation (MLFM) signals, and two kinds of composite modulation (LFM-BPSK, BPSK-BFSK) signals. The sampling frequency was 1 GHz, the pulse width of all radar signals varied from 1  $\mu$ s to 10  $\mu$ s, and other signal parameters are shown in Table 2.

**Table 2.** Parameters of the twelve different intra-pulse modulations of the radar emitter signals.

	Type	Carrier Frequency	Parameter
Source Domain	SCF	50~500 MHz	None
	LFM	100~400 MHz	Bandwidth: 20 MHz~150 MHz
	SFM	100~400 MHz	Bandwidth: 20 MHz~100 MHz
Target Domain	BPSK	100~300 MHz	5,7,11,13-bit Barker code
	BFSK	100~400 MHz	5,7,11,13-bit Barker code
		100~400 MHz	
	QFSK	100~300 MHz	16-bit Frank code
		100~300 MHz	
		100~300 MHz	
		100~300 MHz	
	FRANK	100~400 MHz	Phase number: 4-6
	EQFM	100~300 MHz	Bandwidth: 5 MHz to 100 MHz
	DLFM	100~300 MHz	Bandwidth: 10 MHz to 150 MHz
100~300 MHz			
MLFM	100~300 MHz	Bandwidth: 30 MHz to 100 MHz Bandwidth: 30 MHz to 100 MHz Segment: 20-80%	
	100~300 MHz		
LFM-BPSK	100~300 MHz	Bandwidth: 5 MHz to 150 MHz 5,7,11,13-bit Barker code	
BPSK-BFSK	100~400 MHz	5,7,11,13-bit Barker code	
	100~400 MHz		

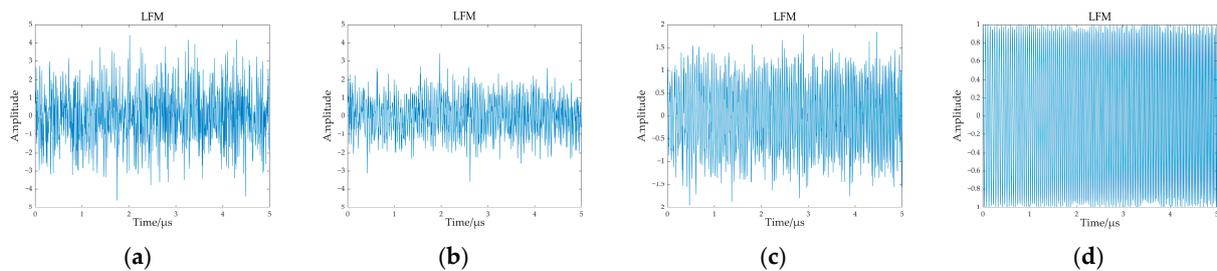
To simulate the real electromagnetic environment, we added additive Gaussian white noise (AWGN) to all signals. The model of the radar signal intercepted by the receiver is given by:

$$x(t) = s(t) + n(t) \quad (13)$$

$n(t)$  is white Gaussian noise, and  $s(t)$  is a radar signal. The SNR is defined as:

$$SNR = 10 \log_{10} \frac{P_s}{P_n} \quad (14)$$

where  $P_s$  represents the effective power of the signal and  $P_n$  is the effective power of the noise. In Figure 3, taking the LFM signal as an example, we simulated the time-domain waveforms of the same signal at  $-5$  dB,  $0$  dB,  $5$  dB, and noiseless, respectively.

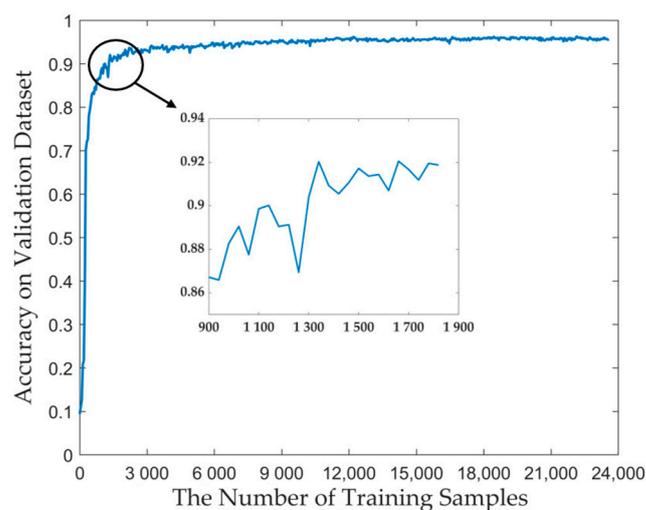


**Figure 3.** Differentiation between LFM signals based on the noise level. (a)  $-5$  dB; (b)  $0$  dB; (c)  $5$  dB; (d) noiseless.

Different research fields require a different number of samples in each field. To better investigate the relationship between the number of training samples and the classification effect of the model in radar intra-pulse modulation classification, we introduced a learning curve [48] to plot the classification accuracy versus the number of the training set. The learning curve equation is as follows:

$$y = 100 + b_1 x^{b_2} \quad (15)$$

where  $y$  is the classification accuracy,  $x$  is the training dataset, and  $b_1$  and  $b_2$  correspond to the learning rate and decay rate, respectively. Figure 4 shows the learning curve of classification accuracy versus number of samples for the nine types of intra-pulse modulated radar signals in the target domain.



**Figure 4.** The learning curve of classification accuracy versus number of samples.

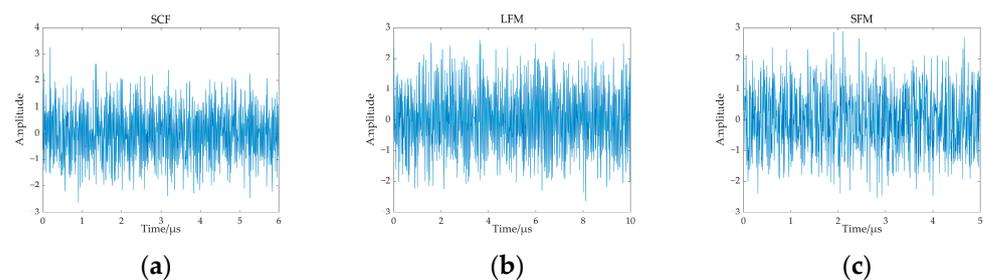
According to the learning curve, the classification accuracy curve reaches smoothness and the model converges when the total number of training samples is 2000, so in this paper, we defined radar signal intra-pulse modulation classification with less than 2000 training samples as small sample learning.

To validate our proposed method on cases of different sample sizes, we randomly generated samples from each type of signal in the target domain with numbers increasing from 50 to 140 at increments of 10, constituting 10 training sets with different sample sizes. For the three types of radar signals in the source domain, the number of samples for each signal was 5000. The number of training sets for each type of radar signal is shown in Table 3.

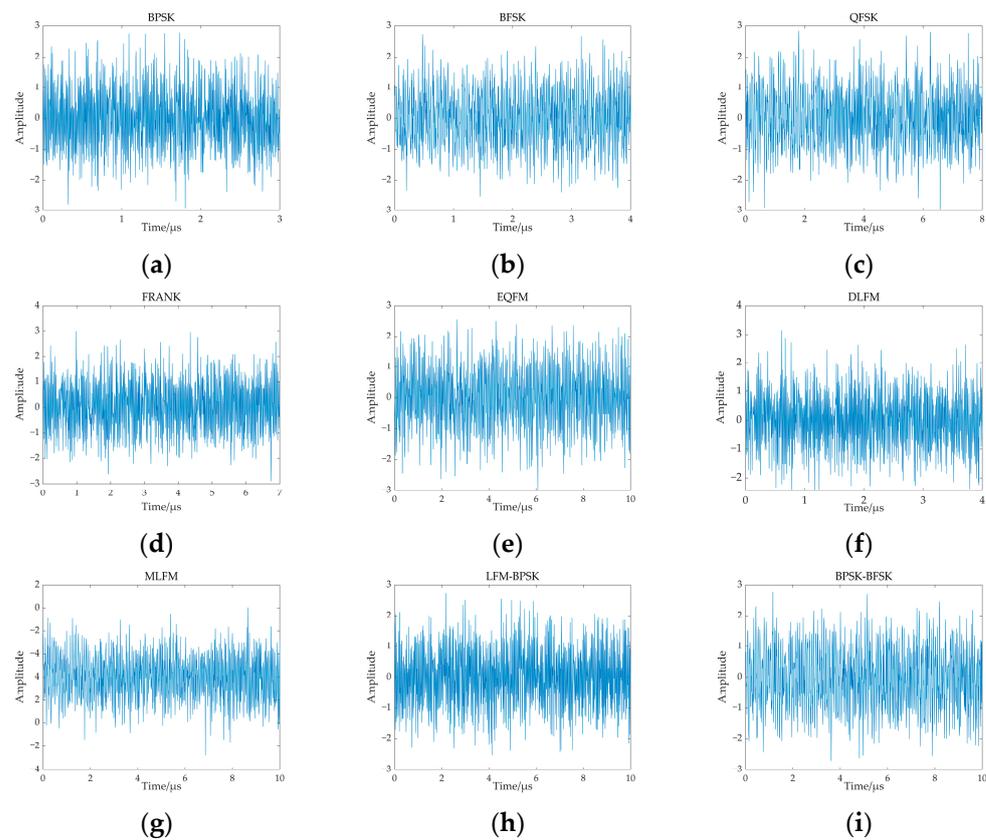
**Table 3.** The number of training sets for each type of radar signal.

	Signal Number	Type
Source Domain	5000 per signal	SCF
		LFM
		SFM
Target Domain	Each type of signal increase from 50 to 140 with a step of 10, constituting 10 training sets with different sample sizes respectively.	BPSK
		BFSK
		QFSK
		FRANK
		EQFM
		DLFM
		MLFM
		LFM-BPSK
BPSK-BFSK		

Each dataset was generated at the same SNR condition ranging from  $-5$  dB to  $5$  dB, with a  $1$  dB interval. An additional set of noise-free signals was generated as a control group to verify the effect of noise on the model performance. For all the above training sets, we produced validation and test sets corresponding to the ratio of  $4:1:1$ . For example, when the target domain training set had 450 samples for each SNR ( $-5\sim 5$  dB and a noise-free dataset for a total of 12 SNRs), the validation set and test set contained 112 samples. Figures 5 and 6 respectively show the waveforms of the source-domain and target-domain intra-pulse modulated radar signals over time when the SNR was  $0$  dB.



**Figure 5.** Different types of intra-pulse modulated radar signals in the source domain. (a) SCF; (b) LFM; (c) SFM.



**Figure 6.** Different types of intra-pulse modulated radar signals in the target domain. (a) BPSK; (b) BFSK; (c) QFSK; (d) FRANK; (e) EQFM; (f) DLFM; (g) MLFM; (h) LFM-BFSK; (i) BPSK-BFSK.

### 3.2. Experiments on 1D-TLAFCNN

#### 3.2.1. Experiments on the Source Domain Network

In this section, the source domain network in Section 2 was trained with only three intra-pulse modulation radar signals. The source task was to classify these three radar signals as accurately as possible to optimize the feature extraction capability of the convolutional layers for 1D radar signals. In this stage of training the source domain network, we added three fully connected layers after the convolutional layer, which contained a hidden layer of 256 neurons and a “Leaky ReLU” activation function. The cross-entropy loss function and Adam optimizer were used with a 0.001 learning rate, the batch size was 64, and the network weights were saved for migration when the validation set had the highest accuracy. The result of the classification task is shown in Table 4.

**Table 4.** The classification accuracy for the source domain at different SNRs.

SNR/dB	−5	−4	−3	−2	−1	0	1	2	3	4	5
SCF	0.9897	0.9923	0.9980	0.9990	0.9969	0.9980	1.0000	1.0000	1.0000	1.0000	1.0000
LFM	0.9840	0.9884	0.9917	0.9915	0.9990	1.0000	0.9990	1.0000	1.0000	1.0000	1.0000
SMF	0.9812	0.9842	0.9863	0.9888	0.9990	0.9969	0.9970	0.9983	1.0000	1.0000	1.0000
Average	0.9850	0.9883	0.9920	0.9931	0.9983	0.9983	0.9987	0.9994	1.0000	1.0000	1.0000

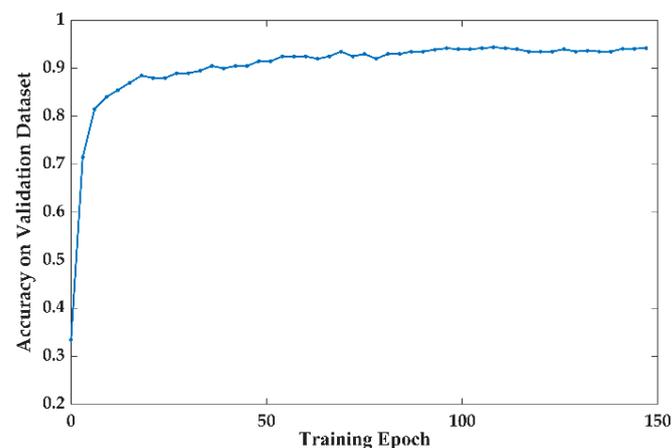
In the subsequent transfer process, we only kept the first four convolutional layers of the source network and discarded the other layers, which contained more semantic information. These optimized lower convolutional layers were more general and could extract structural and detailed features in the radar signal well, even in the face of new data.

### 3.2.2. Experiments on the Target Domain Network

In this section, we transferred the learned weights to the target network and trained the proposed 1D-TLAFLCNN using the different datasets in the target domain generated in Section 3.1, through the following procedure:

1. Initialize the corresponding convolutional layers of the target domain network with the weights learned from the first four convolutional layers of the source domain network, and freeze these weights;
2. Randomly initialize the parameters of the fully connected layers using a Gaussian distribution;
3. Train the classification layers using the target domain dataset;
4. Fine-tune the entire network by unfreezing all convolutional layers and setting a low learning rate (set to 0.0001) to retrain the entire network in order to incrementally fit the pre-trained features to the new data.

When the SNR is 0 dB and the number of training samples is 450 (50 for each signal), the average accuracy value during the training process is shown in Figure 7.



**Figure 7.** The average accuracy value during the training process.

Figure 7 shows that the accuracy of the model stabilized using the validation dataset as the epoch increased, and the accuracy became essentially constant when the epoch reached 90, which indicates that the model converged. Subsequently, we repeated these experiments on training datasets with different SNRs and different sample sizes and tested the classification performance of the proposed model using the highest accuracy weights from the validation dataset. The experiment was repeated 20 times for each case and the average accuracy was taken as the final classification accuracy. The classification accuracies for the nine intra-pulse modulation signals, based on 1D-TLAFLCNN for different cases, are given in Table 5.

As shown in Table 5, we used the training sets of the target domain mentioned in Section 3.1. Each type of signal increased from 50 to 140 with a step of 10, constituting ten training sets with different sample sizes. It can be concluded that the proposed algorithm had good performance in the case of different numbers of small samples, and the average classification accuracy improved as the number of samples increased. Moreover, the proposed method, both in noiseless and noisy environments, had good performance and the classification accuracy steadily improved with increasing SNR. When the SNR was greater than or equal to  $-1$  dB, the classification accuracy of the different data sets was over 90%.

**Table 5.** The classification accuracies of target domain signals, based on 1D-TLAFLCNN for different cases.

SNR \ Size	−5 dB	−4 dB	−3 dB	−2 dB	−1 dB	0 dB	1 dB	2 dB	3 dB	4 dB	5 dB	Noiseless
50	0.8353	0.8444	0.8781	0.8996	0.9031	0.9116	0.9148	0.9227	0.9347	0.9353	0.9365	0.9646
60	0.8400	0.8549	0.8841	0.9059	0.9208	0.9219	0.9171	0.9246	0.9348	0.9372	0.9440	0.9735
70	0.8448	0.8616	0.8966	0.9172	0.9264	0.9287	0.9377	0.9391	0.9441	0.9506	0.9511	0.9746
80	0.8463	0.8637	0.9067	0.9211	0.9273	0.9296	0.9433	0.9444	0.9454	0.9516	0.9512	0.9754
90	0.8480	0.8661	0.9168	0.9238	0.9381	0.9333	0.9452	0.9461	0.9456	0.9529	0.9539	0.9778
100	0.8496	0.8777	0.9201	0.9265	0.9416	0.9405	0.9487	0.9464	0.9468	0.9549	0.9588	0.9798
110	0.8557	0.8804	0.9216	0.9326	0.9423	0.9429	0.9511	0.9466	0.9480	0.9593	0.9594	0.9811
120	0.8581	0.8919	0.9266	0.9397	0.9427	0.9461	0.9525	0.9557	0.9589	0.9639	0.9648	0.9845
130	0.8701	0.9164	0.9278	0.9397	0.9464	0.9524	0.9562	0.9566	0.9642	0.9691	0.9767	0.9863
140	0.8777	0.9206	0.9341	0.9411	0.9514	0.9649	0.9674	0.9709	0.9698	0.9786	0.9854	0.9889

### 3.3. Comparisons with Other Baseline Methods

To show the effectiveness of transfer learning, focal loss, and our proposed adaptive focus loss function (AFL), we constructed five models based on whether or not to add these improvements, while ensuring that they had the same convolutional layer, convolutional kernel size, fully connected layers, batch normalization layer, etc. We used the 1D-CNN as a blank control group, denoted the 1D-CNN with only the focal loss function added (we took the default optimal value as  $\gamma = 2$  [23]) as 1D-FLCNN ( $\gamma = 2$ ), the 1D-CNN with only transfer learning added as 1D-TLCNN, the 1D-CNN with the focal loss function added based on transfer learning as 1D-TLFLCNN ( $\gamma = 2$ ), and the transfer learning-based AFL proposed in this paper as 1D-TLAFLCNN, as shown in Table 6.

**Table 6.** Differences between the proposed method and other baseline methods.

	Focal Loss	Transfer Learning	Adaptive Focus Loss
1D-CNN	no	no	no
1D-FLCNN ( $\gamma = 2$ )	yes	no	no
1D-TLCNN	no	yes	no
1D-TLFLCNN ( $\gamma = 2$ )	yes	yes	no
1D-TLAFLCNN	no	yes	yes

“yes” means that the model uses this improvement, “no” means it is not used.

During the training process, we found that the three methods using transfer learning could have higher classification accuracy on the validation set at the beginning of the iteration, instead of learning from scratch. Transfer learning allows the model to gain some prior knowledge when facing new samples and converge at a faster speed. The accuracy of the five models during the training process at a SNR of 0 dB and 450 training samples (50 for each signal) is shown in Figure 8.

In addition, we compared the number of iterations used to reach convergence and the total time required for the five models in Table 7 and Figure 9.

In addition, we used some representative algorithms as a baseline, including CNN-Qu [7], CNN-Wu [12] and CNN-Wei [30]. All these methods are proposed for intra-pulse modulation classification of radar signals, and have been proved to have good accuracy advantages.

In Table 8 and Figure 10, we compare the classification accuracy of the method proposed in this paper (1D-TLAFLCNN) with other baseline methods at different sample sizes in the 0 dB case and calculate their average accuracy (AA).

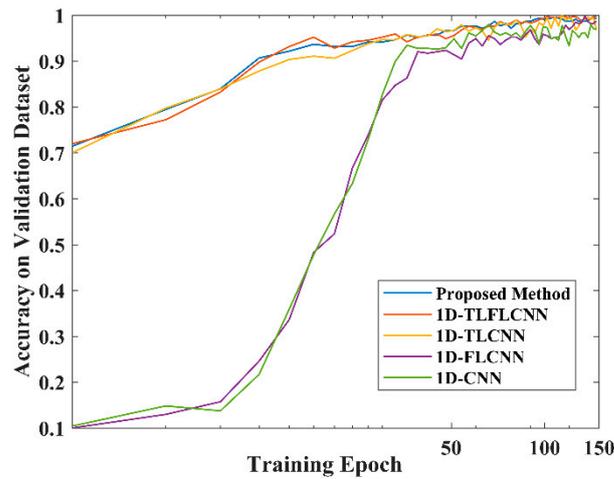


Figure 8. The accuracy of the five models during the training process at a SNR of 0 dB and a number of training samples of 450 (50 for each signal).

Table 7. The number of iterations and time usage for training the model via different methods.

	Number of Epochs	Total Time (RTX3070)
1D-TLAFLCNN	90	258 s
1D-TLFLCNN ( $\gamma = 2$ )	100	284 s
1D-TLCNN	120	355 s
1D-FLCNN ( $\gamma = 2$ )	140	416 s
1D-CNN	160	483 s

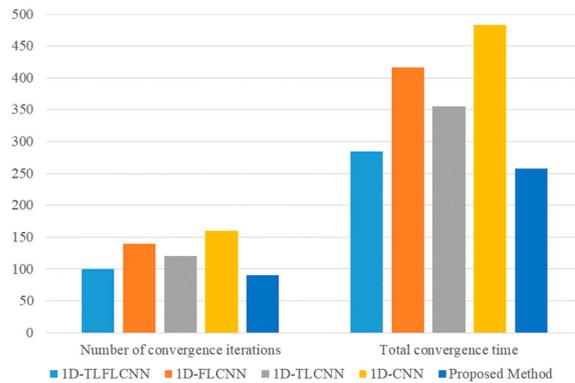
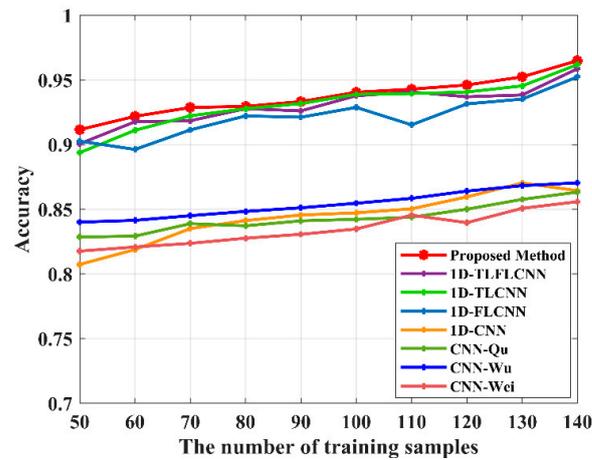


Figure 9. The number of iterations and time usage for training the model in different methods.

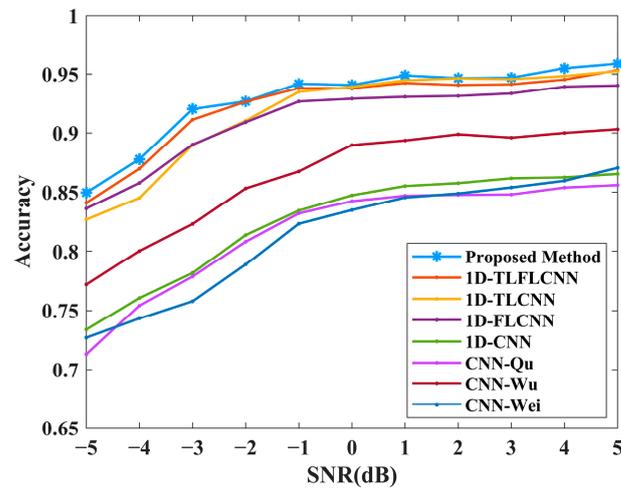
Table 8. The classification accuracy of different methods with different training sets at 0dB.

Datasets	50	60	70	80	90	100	110	120	130	140	AA
1D-TLAFLCNN	0.9116	0.9219	0.9287	0.9296	0.9333	0.9405	0.9429	0.9461	0.9524	0.9649	0.9372
1D-TLFLCNN ( $\gamma = 2$ )	0.9007	0.9177	0.9185	0.9278	0.9261	0.9378	0.9407	0.9370	0.9384	0.9587	0.9303
1D-TLCNN	0.8938	0.9111	0.9223	0.9278	0.9317	0.9389	0.9395	0.9407	0.9454	0.9619	0.9313
1D-FLCNN ( $\gamma = 2$ )	0.9027	0.8963	0.9114	0.9222	0.9212	0.9288	0.9153	0.9315	0.9352	0.9524	0.9217
1D-CNN	0.8074	0.8189	0.8351	0.8413	0.8456	0.8473	0.8504	0.8596	0.8704	0.8644	0.8441
CNN-Qu	0.8286	0.8293	0.8389	0.8373	0.8411	0.8423	0.8439	0.8501	0.8577	0.8633	0.8433
CNN-Wu	0.8751	0.8765	0.8801	0.8834	0.8862	0.8897	0.8935	0.8991	0.9033	0.9055	0.8892
CNN-Wei	0.8177	0.8209	0.8237	0.8277	0.8307	0.8348	0.8455	0.8397	0.8508	0.8559	0.8347



**Figure 10.** The classification accuracy of different methods with different training sets at 0 dB.

In addition, we compared the classification accuracy of all methods on the test set for different SNRs. Figure 11 shows the variation in average accuracy with different SNRs when the sample size was 900 (100 for each signal).



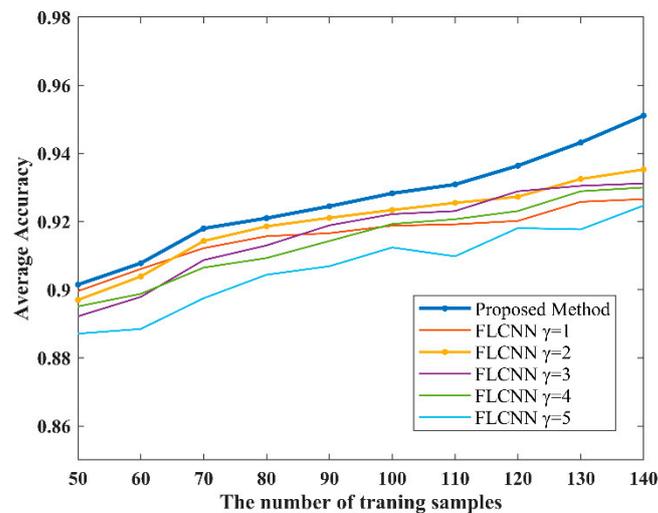
**Figure 11.** The classification accuracy of different methods with different SNRs when the sample size was 900 (100 for each signal).

The experimental results demonstrate that 1D-TLAFLCNN performed best under various SNR conditions, with different magnitudes of improvement compared to other algorithms. The classification effectiveness under different training sets in Table 8 shows that the addition of transfer learning and AFL resulted in a greater improvement than using only transfer learning or FL separately, in which transfer learning could reduce the number of convergence generations required for training and converge faster; AFL could estimate  $\gamma$  adaptively based on FL and improved the accuracy of classification.

#### 4. Discussion

##### 4.1. AFL Compared with Different Values of the Focusing Parameter Based on FL

To investigate the effect of different values of the focusing parameter  $\gamma$  in FL on the classification effect and thus prove the effectiveness of our proposed method, we compared the average accuracy of all SNRs with several different few-shot sample sizes, as shown in Figure 12.



**Figure 12.** The average classification accuracy of all SNRs for different focusing parameters with different sample sizes.

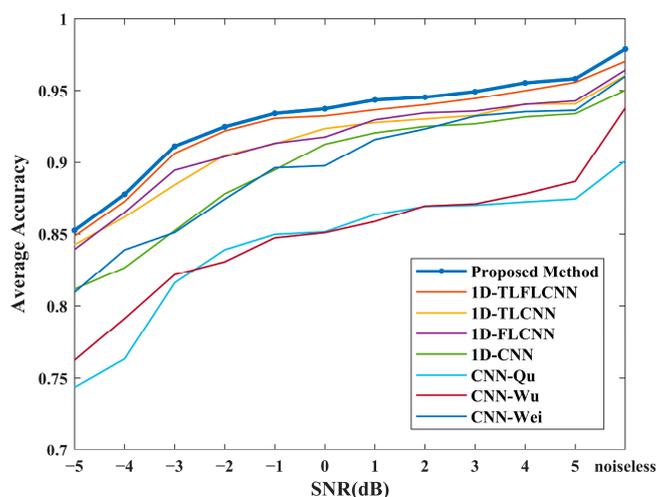
As shown in the figure, for the same classification task, the value of the focusing parameter affected the classification accuracy, as it represents how much attention the model pays to the hard samples. A large value of the focusing parameter (e.g.,  $\gamma=5$ ) tended to over-focus the model on some hard samples and bias the trained model toward these “outliers”, which is often fatal when the training sample is insufficient. In addition, even if the focus parameter took the same value, the classification accuracy kept fluctuating with the sample size. This is because the number of hard and easy samples changed with the sample size, which means that the classification task also changed, and the same value of the focusing parameter could not adjust to all classification tasks. Typically, for new classification tasks, we must conduct quantitative testing to find the appropriate value of the focus parameter  $\gamma$ , and our method is proposed to solve this problem. As we can see in the figure, our proposed method could estimate the range of  $\gamma$  by calculating the proportion of hard and easy samples in the dataset, which was a good improvement compared to other integer values of  $\gamma$ .

#### 4.2. Effect of Different Noise Environments on Experimental Results

The classification performance of the model in noisy environments is one measure of model stability. To this end, we explored the effect of SNRs on the model in both noisy and noiseless environments. We compared the model classification performance of different models for a total of 12 scenarios from  $-5$  dB to  $5$  dB along with pure signals without noise pollution, as shown in Table 9 and Figure 13. The accuracy in the chart is the average classification accuracy after repeating the experiment for ten different sample sets.

**Table 9.** Classification accuracies with different noise environments.

SNR	−5 dB	−4 dB	−3 dB	−2 dB	−1 dB	0 dB	1 dB	2 dB	3 dB	4 dB	5 dB	Noiseless
1D-TLAFLCNN	0.8525	0.8777	0.9112	0.9247	0.9340	0.9372	0.9434	0.9453	0.9492	0.9553	0.9581	0.9786
1D-TLFLCNN ( $\gamma = 2$ )	0.8486	0.8728	0.9063	0.9218	0.9306	0.9323	0.9365	0.9401	0.9444	0.9499	0.9557	0.9702
1D-TLCNN	0.8426	0.8619	0.8843	0.9049	0.9129	0.9235	0.9277	0.9302	0.9326	0.9405	0.9406	0.9605
1D-FLCNN ( $\gamma = 2$ )	0.8393	0.8652	0.8945	0.9039	0.9133	0.9176	0.9296	0.9344	0.9356	0.9403	0.9427	0.9642
1D-CNN	0.8117	0.8267	0.8524	0.8781	0.8947	0.9126	0.9205	0.9250	0.9268	0.9316	0.9338	0.9502
CNN-Qu	0.7436	0.7633	0.8161	0.8391	0.8498	0.8516	0.8636	0.8693	0.8702	0.8725	0.8746	0.9011
CNN-Wu	0.7624	0.7913	0.8218	0.8308	0.8474	0.8510	0.8588	0.8697	0.8711	0.8782	0.8868	0.9377
CNN-Wei	0.8095	0.8390	0.8511	0.8744	0.8962	0.8976	0.9159	0.9232	0.9322	0.9352	0.9361	0.9599



**Figure 13.** Classification accuracies with different noise environments.

It can be seen that our proposed method had the highest average accuracy, 97.86%, in a noise-free environment. The classification performance of all models for noise-free signals was greatly improved compared to noisy environments, with an average accuracy improvement of 2–5%, since the models are more likely to extract features in pure signals, which are often masked in noisy environments. However, even in the  $-5$  dB case, the average accuracy of our proposed algorithm was over 85%. Meanwhile, the model recognition capability steadily improved as the SNR increased, which indicates that our algorithm has certain anti-noise performance.

#### 4.3. Effect of Different Sample Sizes on Experimental Results

As shown in Table 8 and Figure 10, the proposed method had the best accuracy on different few-shot sample datasets, and the average accuracy improved by 8% compared with the traditional CNN algorithm. The comparison between 1D-TLAFLCNN and 1D-TLFLCNN shows that the estimate of  $\gamma$  calculated using AFL had a slightly better classification performance than the default value of taking  $\gamma = 2$ . Overall, the classification results were better when using the transfer learning approach. The method using FL alone (default value  $\gamma = 2$ ) yielded improved results in most cases, but the classification accuracy decreased in some cases, which may be because FL focuses excessively on outliers in the sample and makes the model misclassify. In comparison with other baseline methods, the proposed method had the best classification accuracy for all few-shot sample sizes. The method using TL alone had a flatter accuracy curve. It increased steadily as the sample size increased. Transfer learning enables the model to have some prior knowledge, which makes it have good feature extraction ability and effective performance when facing new few-shot problems.

However, as the number of samples increased, our method also showed certain shortcomings: the improvement effect was not obvious in some sample sets, with the accuracy improvement being less than 0.2%. We analyze that as the sample size increased, the difficulty of model training decreased, and the ratio of hard to easy samples tended to become balanced, which decreased the role of transfer learning and AFL.

#### 4.4. Improvement in Training Model Time Consumption

Besides classification accuracy, fast classification is also necessary for radar reconnaissance systems, which need real-time classification. Shorter training times and faster convergence mean that our models could make predictions faster in real-world applications, helping to speed up our analysis of enemy radar systems.

In real radar reconnaissance systems, real-time training and classification of few-shot unknown signals for a new scene is an important aspect, which requires the model to have

fast convergence capability. In X-band radar, the airborne radar pulse repetition frequency is  $f$  Hz, which emits  $n$  pulses at one wave position; therefore, the radar time-on-target collected in one scan is roughly  $\frac{n}{f}$  seconds. Then, we can calculate the duration of the next received radar echo accordingly, which requires the model to finish converging when the signal is received again. Therefore, considering this practical application, the model convergence epoch and the time spent can be one of the judgments for the performance of the method.

By comparing the number of iterations and the total training time required to reach convergence for the five models in Table 7 and Figure 9, the proposed method converged at 90 generations and 258 s, which reduced the time cost by at least 10%, but this improvement is still far from enough to meet the time requirements for real-time training and testing in practical applications. In future work, we will continue to improve the convergence speed of the model and reduce the training time.

## 5. Conclusions

To solve the problems associated with the difficulty of training deep cellular neural networks and the insufficient training data available for radar signals, which leads to the low classification accuracy of intra-pulse modulation, a 1D-TLAFLCNN method is proposed for the classification of few-shot intra-pulse modulation radar signals. We used a transfer learning method to transfer the knowledge learned from a large number of simple intra-pulse modulated radar signals in the source domain to a complex modulation classification task in the target domain, and estimated the factor  $\gamma$  adaptively based on FL, which ensured that the model could focus more on the hard samples.

The experimental results show that the proposed method could reduce the number of generations to model convergence compared with other baseline methods, and the model converged within 90 generations with the shortest time of 258 s. By comparing the experiments with different values of focusing parameters, AFL had a maximum accuracy improvement of approximately 1.5% based on FL and could reduce repeated experiments by estimating the range of parameters. In addition, the proposed method displayed noise immunity, and the average accuracy of our proposed method was over 85% in the  $-5$  dB case. From further experiments exploring its classification performance on different few-shot datasets, we found that the application of transfer learning could well help the model gain rich prior knowledge and feature extraction ability in few-shot cases, which improved by 8% compared with the traditional CNN algorithm. However, we are also aware of the shortcomings of our method. First, the focusing parameter derived from AFL was a fixed value, but as training proceeded the proportion of hard and easy samples changed, and a fixed focusing parameter could not adapt to the new sample proportion, which made the focal loss function useless. Second, as the number of samples increased, the improvement effect of transfer learning was not obvious. In future work, we hope to develop a method to realize dynamic adjustment of the focusing parameters following changes in the ratio of hard and easy samples.

**Author Contributions:** Conceptualization, Z.J. and P.L.; methodology, Z.J. and P.L.; software, Z.J.; validation, Z.J.; formal analysis, P.L. and Z.J.; investigation, Z.J., S.Y. and Y.C.; resources, P.L. and B.W.; data curation, Z.J.; writing—original draft preparation, Z.J.; writing—review and editing, Z.J.; supervision, P.L. and B.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

**Acknowledgments:** The author would like to show their gratitude to the editors and the reviewers for their insightful comments.

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

## References

1. Wang, S. Research on recognition algorithm for intra pulse modulation of radar signals. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 October 2018; pp. 1092–1096. [\[CrossRef\]](#)
2. Jin, Q.; Wang, H.Y.; Ma, F.F. An overview of radar emitter classification and identification methods. *Telecommun. Eng.* **2019**, *59*, 360–368.
3. Ma, X.R.; Liu, D.; Shan, Y.L. Intra-pulse modulation recognition using short-time ramanujan Fourier transform spectrogram. *EURASIP J. Adv. Signal Process.* **2017**, *1*, 42. [\[CrossRef\]](#)
4. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Huang, Z.L.; Datcu, M.; Pan, Z.X.; Lei, B. Deep SAR-Net: Learning objects from signals. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 179–193. [\[CrossRef\]](#)
6. Wang, X.B.; Huang, G.M.; Zhou, Z.W.; Tian, W.; Yao, J.L.; Gao, J. Radar emitter recognition based on the energy cumulant of short time Fourier transform and reinforced deep belief network. *Sensors* **2018**, *18*, 3103. [\[CrossRef\]](#)
7. Qu, Z.Y.; Wang, W.Y.; Hou, C.B.; Hou, C.F. Radar signal intra-pulse modulation recognition based on convolutional denoising autoencoder and deep convolutional neural network. *IEEE Access* **2019**, *7*, 112339–112347. [\[CrossRef\]](#)
8. Gao, L.P.; Zhang, X.L.; Gao, J.P.; You, S.X. Fusion image based radar signal feature extraction and modulation recognition. *IEEE Access* **2019**, *7*, 13135–13148. [\[CrossRef\]](#)
9. Liu, Z.; Shi, Y.; Zeng, Y.; Gong, Y. Radar emitter signal detection with convolutional neural network. In Proceedings of the 2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT), Jinan, China, 18–20 October 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 48–51.
10. Sun, J.; Xu, G.L.; Ren, W.J.; Yan, Z.Y. Radar emitter classification based on unidimensional convolutional neural network. *IET Radar Sonar Navig.* **2018**, *12*, 862–867. [\[CrossRef\]](#)
11. Li, X.Q.; Liu, Z.M.; Huang, Z.T.; Liu, W.S. Radar emitter classification with attention-based multi-RNNs. *IEEE Commun. Lett.* **2020**, *24*, 2000–2004. [\[CrossRef\]](#)
12. Wu, B.; Yuan, S.B.; Li, P.; Jing, Z.H.; Huang, S.; Zhao, Y.D. Radar emitter signal recognition based on one-dimensional convolutional neural network with attention mechanism. *Sensors* **2020**, *20*, 6350. [\[CrossRef\]](#)
13. Li, F.Z.; Liu, Y.; Wu, P.X.; Dong, F.; Cai, Q.; Wang, Z. A survey on recent advances in meta-learning. *Chin. J. Comput.* **2021**, *44*, 422–446. [\[CrossRef\]](#)
14. Li, Y.; Ding, Z.; Zhang, C.; Wang, Y.; Chen, J. SAR ship detection based on resnet and transfer learning. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1188–1191. [\[CrossRef\]](#)
15. Pan, S.J.L.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [\[CrossRef\]](#)
16. Dai, W.; Yang, Q.; Xue, G.; Yu, Y. Boosting for transfer learning. Machine Learning. In Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, OR, USA, 20–24 June 2007.
17. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring midlevel image representations using convolutional neural networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1717–1724. [\[CrossRef\]](#)
18. Huang, Z.L.; Pan, Z.X.; Lei, B. Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. *Remote Sens.* **2017**, *9*, 907. [\[CrossRef\]](#)
19. Shang, R.H.; Wang, J.M.; Jiao, L.C.; Stolkin, R.; Hou, B.; Li, Y.Y. SAR targets classification based on deep memory convolution neural networks and transfer parameters. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 2834–2846. [\[CrossRef\]](#)
20. Zhang, W.; Zhu, Y.F.; Fu, Q. Deep transfer learning based on generative adversarial networks for SAR target recognition with label limitation. In Proceedings of the 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019; pp. 1–5. [\[CrossRef\]](#)
21. Huang, Z.L.; Dumitru, C.O.; Pan, Z.X.; Lei, B.; Datcu, M. Classification of large-scale high-resolution SAR images with deep transfer learning. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 107–111. [\[CrossRef\]](#)
22. Rostami, M.; Kolouri, S.; Eaton, E.; Kim, K. Deep transfer learning for few-shot SAR image classification. *Remote Sens.* **2019**, *11*, 1374. [\[CrossRef\]](#)
23. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Cun, Y.L.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Habbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* **1990**, *2*, 396–404.
25. Cun, L.Y.; Kavukcuoglu, K.; Faret, C. Convolutional networks and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France, 30 May–2 June 2010; pp. 253–256.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [\[CrossRef\]](#)
27. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *Comput. Sci.* **2014**, *1409*, 1556.

28. Szegedy, C.; Liu, W.; Jia, Y.Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]
29. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
30. Wei, S.J.; Qu, Q.Z.; Su, H.; Wang, M.; Shi, J.; Hao, X.J. Intra-pulse modulation radar signal recognition based on CLDN network. *IET Radar Sonar Navig.* **2020**, *14*, 803–810. [[CrossRef](#)]
31. Pan, Y.W.; Yang, S.H.; Peng, H.; Li, T.Y.; Wang, W.Y. Specific emitter identification based on deep residual networks. *IEEE Access* **2019**, *7*, 54425–54434. [[CrossRef](#)]
32. Shao, L.; Zhu, F.; Li, X.L. Transfer learning for visual categorization: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 1019–1034. [[CrossRef](#)] [[PubMed](#)]
33. Varga, D. No-reference video quality assessment using multi-pooled, saliency weighted deep features and decision fusion. *Sensors* **2022**, *22*, 2209. [[CrossRef](#)] [[PubMed](#)]
34. Varga, D. Multi-pooled inception features for no-reference image quality assessment. *Appl. Sci.* **2020**, *10*, 2186. [[CrossRef](#)]
35. Li, C.; Zhang, S.H.; Qin, Y.; Estupinan, E. A systematic review of deep transfer learning for machinery fault diagnosis. *Neurocomputing* **2020**, *407*, 121–135. [[CrossRef](#)]
36. Wang, Q.; Du, P.F.; Yang, J.Y.; Wang, G.H.; Lei, J.J.; Hou, C.P. Transferred deep learning based waveform recognition for cognitive passive radar. *Signal Process.* **2019**, *155*, 259–267. [[CrossRef](#)]
37. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769. [[CrossRef](#)]
38. Tian, X.W.; Wu, D.; Wang, R.; Cao, X.C. Focal text: An accurate text detection with focal loss. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 2984–2988. [[CrossRef](#)]
39. Chen, M.Q.; Fang, L.; Liu, H.F. FR-NET: Focal loss constrained deep residual networks for segmentation of cardiac MRI. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 764–767. [[CrossRef](#)]
40. Su, H.; Wei, S.J.; Wang, M.K.; Zhou, L.M.; Shi, J.; Zhang, X.L. Ship detection based on retinaNet-Plus for high-resolution SAR imagery. In Proceedings of the 2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), Xiamen, China, 26–29 November 2019; pp. 1–5. [[CrossRef](#)]
41. Nagi, J.; Ducatelle, F.; Caro, G.A.D.; Cireşan, D.; Meier, U.; Giusti, A.; Nagi, F.; Schmidhuber, J.; Gambardella, L.M. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In Proceedings of the 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 16–18 November 2011; pp. 342–347. [[CrossRef](#)]
42. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [[CrossRef](#)]
43. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
44. Xu, B.; Wang, N.Y.; Chen, T.Q.; Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853.
45. Xu, J.; Li, Z.S.; Du, B.W.; Zhang, M.M.; Liu, J. Reluplex made more practical: Leaky ReLU. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7. [[CrossRef](#)]
46. Yu, W.; Yang, K.Y.; Yao, H.X.; Sun, X.S.; Xu, P.F. Exploiting the complementary strengths of multi-layer CNN features for image retrieval. *Neurocomputing* **2017**, *237*, 235–241. [[CrossRef](#)]
47. Li, B.; Liu, Y.; Wang, X. Gradient Harmonized Single-Stage Detector. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8577–8584. [[CrossRef](#)]
48. Figueroa, R.L.; Zeng-Treitler, Q.; Kandula, S.; Ngo, L.H. Predicting sample size required for classification performance. *BMC Med. Inform. Decis. Mak.* **2012**, *12*, 8. [[CrossRef](#)]