drones

MDPI

Article

IRelNet: An Improved Relation Network for Few-Shot Radar Emitter Identification

Zilong Wu, Meng Du, Daping Bi and Jifei Pan *

School of Electronic Countermeasures, National University of Defense Technology, Hefei 230031, China; wuzilong@nudt.edu.cn (Z.W.); dm_csy@nudt.edu.cn (M.D.); bdpeei@163.com (D.B.)
* Correspondence: panjifei17@nudt.edu.cn

Abstract: In future electronic warfare (EW), there will be many unmanned aerial vehicles (UAVs) equipped with electronic support measure (ESM) systems, which will often encounter the challenge of radar emitter identification (REI) with few labeled samples. To address this issue, we propose a novel deep learning network, IRelNet, which could be easily embedded in the computer system of a UAV. This network was designed with channel attention, spatial attention and skip-connect features, and meta-learning technology was applied to solve the REI problem. IRelNet was trained using simulated radar emitter signals and can effectively extract the essential features of samples in a new task, allowing it to accurately predict the class of the emitter to be identified. Furthermore, this work provides a detailed description of how IRelNet embedded in a UAV was applied in the EW scene and verified its effectiveness via experiments. When the signal-to-noise ratio (SNR) was 4 dB, IRelNet achieved an identification accuracy of greater than 90% on the samples in the test task.

Keywords: electronic warfare; UAV; radar emitter identification; meta-learning; relation network; attention mechanism; few-shot

Citation: Wu, Z.; Du, M.; Bi, D.; Pan, J. IRelNet: An Improved Relation Network for Few-Shot Radar Emitter Identification. *Drones* **2023**, *7*, 312. https://doi.org/10.3390/drones7050312

Academic Editor: Gwanggil Jeon

Received: 1 March 2023
Revised: 26 April 2023
Accepted: 7 May 2023
Published: 8 May 2023

1. Introduction

In the field of future electronic warfare (EW), an electronic reconnaissance un-manned aerial vehicle (UAV) must possess the capability to accurately identify radar emitters. The radar emitter identification (REI) capability of UAV troops is essential to their combat efficacy. REI is defined as the process of recognizing various electromagnetic signal carriers in the environment via electronic support measures (ESM) or electronic intelligence reconnaissance (ELINT) systems. If UAV forces were capable of accurately identifying various radar emitters, they would be able to more effectively deduce the operational modes and tactical states of these radar emitters, thus yielding more beneficial intelligence information [1–4]. However, to the best of our knowledge, most of the published papers related to UAVs focus on how to identify UAV signals [5,6] or monitor UAVs [7–9]. Few research papers are aimed at investigating the radar emitter identification (REI) system carried out on the reconnaissance UAV platform, and even fewer research papers are specifically aimed at addressing few-shot REI problems.

The initial methods of REI mainly consisted of two stages: "radar word" extraction and feature parameter matching. Visnevski et al. analyzed the pulse sequence of radar emitters, extracted "radar words" related to the radar state, and matched the extracted "radar words" with those in the database to realize REI [10,11]. The REI system proposed by Visnevski et al. not only provided a comprehensive processing method for radar signals but also introduced a detailed mathematical model of the radar state. Their REI system was highly comprehensive and had pioneering significance in the field of EW. Building on the work by Visnevski et al., Li et al. further improved the method of extracting "radar words". The authors proposed a pulse repetition interval (PRI) analysis method based on the time of arrival (TOA), which converted the one-dimensional TOA sequence into a two-dimensional TOA-PRI, thus effectively reducing the impact of lost pulses and false

pulses [12]. Matuszewski also recognized the importance of accurately identifying radar emitters for EW and introduced an ELINT system in detail [13]. The technical methods proposed in [10–13] generally had high requirements for radar emitter data, as the intercepted radar pulse sequence must be long enough to extract the "radar word" of the radar emitter. Furthermore, the modulation mode of the new generation multifunction radar is highly complex, making it difficult to accurately estimate its characteristic parameters. In conclusion, REI methods based on matching traditional features such as "radar words" are not suitable for modern REI tasks.

To address the aforementioned issues, deep learning technology has been widely adopted by researchers in the field of signal recognition, allowing REI to be achieved using only short radar intrapulse signals. Generally, signal recognition methods based on deep learning consist of three stages: signal domain transformation, automatic feature extraction and signal recognition.

In [14], short-time Fourier transform (STFT) images of radar signals served as the dataset to train a deep learning network. After the network had been trained with numerous images, it was able to automatically extract the characteristics of signals and intelligently identify different radar emitters. In [15], the STFT images of signals were normalized to accelerate the training process of a deep learning network. In [16], STFT images were also employed to train a deep learning network, which was augmented with a residual learning convolutional denoising automatic encoder (RLCDAE) for data denoising and an auxiliary classification network for improved recognition accuracy. Refs. [17,18] fused the smooth pseudo-Wigner–Ville distribution (SPWVD) of a signal with its local binary pattern and used the fused features to train a deep learning network, further improving recognition accuracy. In [18–23], Cohen time–frequency distribution images of signals were employed as the dataset to train a network and preprocessed radar signals via clipping, binarization, denoising, scaling and other methods to enhance the adaptability of REI methods. However, the calculation of the Cohen time–frequency distribution is highly complex and requires many super parameters. In [24], the Hilbert–Huang transform (HHT) feature and bispectral feature of radar emitters were fused, and a convolutional neural network (CNN) was then utilized to automatically extract deep features and accurately identify radar emitters. However, the preprocessing of signals in [24] was computationally expensive, which hindered its ability to rapidly identify radar emitters.

Furthermore, end-to-end REI methods have been gaining traction among researchers due to their high intelligence level and reduced reliance on expert knowledge. However, these methods are highly sensitive to SNRs, and thus, their robustness needs to be further improved [25–27].

In summary, most signal identification methods based on deep learning have effectively addressed the challenge of extracting traditional features from radar signals. However, the level of fusion between the above methods and the EW scene is not high, and all methods require a large amount of training data as the basis. When the radar emitter data are limited, it is often difficult for the existing signal identification methods based on deep learning to accurately identify various radar emitters. However, in actual EW scenarios, it is difficult for an ELINT system to obtain numerous noncooperative radar signals. Thus, when the EW force only has access to a few labeled radar emitter samples, how to accurately identify them remains an issue that warrants further investigation.

To address the issue of intelligent identification with few labeled samples, meta-learning technology has achieved great success in the field of computer vision [28–31], providing a novel solution to the problem of REI with few samples. In particular, metric-based meta-learning technology is not only easy to implement in hardware but also has stable performance in various complex sample identification tasks, rendering it highly suitable for application in the REI field.

Metric-based meta-learning networks mainly include the Siamese network (SN) [32,33], matching network (MN) [34], prototypical network (PN) [35] and relation network (RN) [36]. Professor Hinton took the lead in applying the SN to face verification, indicating that the

deep learning network could be used to resolve the recognition problem with few samples. Then, their work caused an upsurge in research on few-shot identification [37]. The strategy for training the MN is unique. The training data employed in the MN are divided into training tasks and test tasks. Every training task or test task consists of a support set and query set. A task in which the support set contains N classes, and every class contains K samples is referred to as an N-way K-shot task. Every training episode of the MN removes a task from many training tasks, and then the feature representations of all samples in the support set and query set of the task are obtained via a CNN. Then, all the feature representations are input into a long short-term memory network, and cosine distance is used to evaluate the similarity between the samples in the query set and those in the support set. The samples in the support set can be identified according to their similarity. The MN directly stacks feature representations of all samples, which cannot fully utilize the common features of samples belonging to the same class, and it lacks interpretability.

Similar to the MN, the PN also uses many few-shot identification tasks to train a network, but the network learns the ability to extract the class prototype. According to the samples in the support set of the test task, the trained PN can obtain the prototype of every class and then identify the samples in the query set by comparing the distance between the samples to be identified and the prototype of every class.

Compared with the PN, the largest advantage of the RN is that it does not use a fixed metric distance but automatically learns the distances among samples via a relation module. Therefore, when performing different few-shot identification tasks, the RN often has a more stable performance than other meta-learning networks. Additionally, the RN is easier to train. Therefore, the RN is suitable for solving the REI problem with few samples. The original structure of the RN is shown in Figure 1.

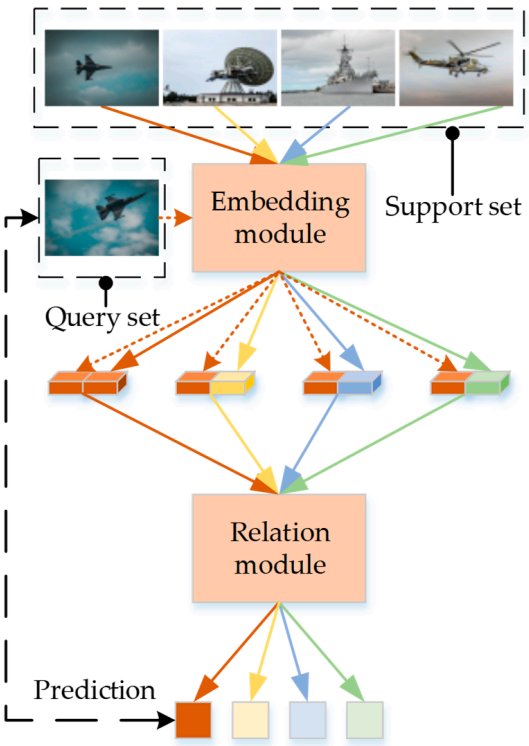

Figure 1. Original structure of RN.

In an RN, the embedding module maps each sample in a support set or query set into a feature, which is then concatenated. The relation module is used to infer to which class the samples in the query set belong based on their similarity to the samples in the support set.

Presently, there are few studies on REI using meta-learning technology. In [38], a network named RepVGG (reparameterization visual geometry group) was used to replace the CNN in the PN, and the improved PN was utilized to identify radar emitters. The improved PN was evaluated on three different datasets containing fourteen classes of signals. However, the method using fixed spatial distance measurement was employed in the above work, which limited the performance of their method to a certain extent.

P. Lang et al. proposed a novel network, named RRSARNet [39], for radar radio source adaptive recognition. RRSARNet can effectively assess the similarity between two samples and then infer the classes of signals to be identified according to a small number of labeled signals. However, in [39], the STFT was utilized for preprocessing radar signals, which reduced the adaptability of their algorithm. In addition, their work added different levels of noise to radar signals to obtain data with more classes, but the differences among the obtained data were not the same as those among the real radar emitters.

Other signal identification methods based on meta-learning also brought great inspiration to our work. The authors of [40] proposed a few-shot learning framework based on foreground segmentation and metric learning. The framework could effectively identify signals with different modulation types using only one labeled sample. In the above work, a double deep image prior (Double-DIP) network was employed for denoising and purifying signals, and feature fusion technology was introduced to improve the signal recognition ability. However, the samples in the training tasks and test task were generated by a set of fixed parameters, so the deep features of various samples would be slightly reduplicative, which was not consistent with the original principle of meta-learning.

In [5], model-agnostic meta-learning (MAML) was introduced into the field of specific emitter recognition and realized recognizing in-phase/quadrature (I/Q) signals by constructing a meta-learning model based on a CNN. When considering the characteristics of signals transmitted via ZigBee devices in the above work, the preamble of these signals was directly utilized as the training data. Thus, the method was not very adaptive to REI. In addition, to the best of our knowledge, the meta-learning network was difficult to train using time domain signals with low SNRs, so the robustness of the above meta-learning network needed to be further improved.

For identifying radar emitters, ref. [41] proposed a combined SN that was composed of several SNs. Every single SN was used to recognize a class of signal, and the recognition results of all single SNs were integrated to determine the class of signal to be identified. However, the above combined SN needed too many training tasks and high computing costs, which made its adaptation limited to certain scenarios. In [42,43], meta-learning was also introduced into the field of signal recognition under unsupervised conditions.

Generally, the identification performance of current REI methods based on meta-learning under low SNRs is still poor, and the adaptability of these methods to different application scenes is not high. Especially in the case of EW, when a UAV can only know a few labeled samples, REI research based on meta-learning is still in the initial exploration stage and needs further in-depth investigation by more scholars.

In this work, an improved RN, IRelNet, is proposed to address the unsatisfactory performance of REI techniques embedded in UAVs for few-shot scenarios. IRelNet was trained using simulated REI tasks that were created from a sizable quantity of simulated radar signals with their respective labels. The trained network was able to assess whether two samples belonged to the same class and to deduce the class of radar emitters to be identified based on the small number of labeled samples. In addition, the method that we propose can also be applied to other platforms, not just UAV platforms. However, in this work, we mainly applied the proposed method to the ESM system of UAVs. The main contributions of this work are listed as follows:

- The proposed IRelNet model, which can be embedded in UAVs, is presented. This network could substantially improve the performance of the REI techniques in few-shot scenarios.

- IRelNet was augmented with both channel attention and spatial attention, which served to augment its capability to extract the deep features of samples.
- To counteract the vanishing gradient problem and improve the stability of IRelNet, a skip connection was integrated into IRelNet.
- The details of how IRelNet embedded in UAVs was utilized to address the real EW scene are presented.

2. Radar Emitter Signal Preprocessing

2.1. WVD

Presently, most radar emitter intelligent identification technologies often need to transform radar signals into their other domain representations to easily extract essential features. In this work, we also need to preprocess radar signals; thus, the features of signals can become more apparent, and the preprocessed signals can be utilized for our proposed IRelNet model. However, a simple signal spectrum has difficulty showing the differences between complicated signals. The calculation costs for the high-order spectrum [44,45], cyclic spectrum [46,47] and cepstrum [48] are high, so it is difficult to rapidly identify radar emitters. Time–frequency transforms can express the differences between radar signals, and their performance is often stable. Therefore, time–frequency transforms have gradually become the popular method to preprocess radar signals.

The STFT is the most basic time–frequency transform; it segments a signal through a time window function and performs a Fourier transform on every time segment to obtain the frequency distribution of the signal. The method leads to a conflict between time and frequency resolution, so its adaptability is poor. The continuous wavelet transform is a linear time–frequency analysis method that has better adaptability than the STFT. However, the performance of the continuous wavelet transform is greatly different due to the different wavelet basis functions. The HHT is a signal analysis method with high adaptability, but its process is too complex and lacks a theoretical basis and interpretability. However, the Wigner–Ville distribution (WVD) has good time–frequency aggregation and adaptability and is often used in the field of signal processing. Therefore, the WVD was selected for preprocessing radar signals in this work. If a radar emitter signal captured with a signal receiver is expressed as $x(m)$, which is a discrete signal, m is the sampling point. The discrete WVD of $x(m)$ is obtained via Equation (1):

$$\mathbf{W}_{n,\omega} = \sum_{m=-\infty}^{\infty} x(n+m)x^*(n-m)e^{-j\omega m} \tag{1}$$

For radar emitter signals with different carrier frequencies, different pulse widths, or different sampling points, the size of the time dimension or frequency dimension of their discrete WVDs are different. The time dimension or frequency dimension differs among various signals' WVDs, which also indicates that the rows or columns of various signals' discrete WVDs are different. To improve the adaptability of the REI model, it is necessary to further standardize the size of the discrete WVDs of various signals. Thus, the discrete WVDs of various signals with different dimensions can be processed using the proposed network whose input size is fixed.

2.2. Bicubic Interpolation

Bicubic interpolation is often employed in the field of image scaling and has better scaling performance than nearest-neighbor interpolation and linear interpolation. By executing bicubic interpolation on discrete WVDs with various sizes, WVDs with the desired fixed size can be obtained [49,50]. The specific steps of executing bicubic interpolation on an initial discrete WVD $\mathbf{W}_{n,\omega}$ of a radar signal are shown in Table 1.

Table 1. Specific steps of executing bicubic interpolation on an initial WVD.

Step	Details
Step 1	The desired size of the initial $\mathbf{W}_{n,\omega}$ is set to (S_i, S_i), and the matrix obtained via bicubic interpolation is represented as $\mathbf{W}'_{n,\omega}$;
Step 2	According to the size of $\mathbf{W}'_{n,\omega}$ and $\mathbf{W}_{n,\omega}$, the scaling factor (K_1, K_2) is calculated via $K_1 = S_i/\text{row}(\mathbf{W}_{n,\omega})$ and $K_2 = S_i/\text{column}(\mathbf{W}_{n,\omega})$, where row() and column() represent the sizes of rows and columns of a matrix, respectively;
Step 3	The bicubic interpolation function, $\text{Bi}(x)$, is expressed as $\text{Bi}(x) = \begin{cases} 2.5\lvert x\rvert^3 - 3.5\lvert x\rvert^2 + 1 & \lvert x\rvert \le 1 \\ 0.5\lvert x\rvert^3 - 2.5\lvert x\rvert^2 + 4\lvert x\rvert - 2 & 1 < \lvert x\rvert < 2 \\ 0 & \text{otherwise} \end{cases}$;
Step 4	(1) Every element $e'_{i\prime,j\prime}$ is obtained from the desired matrix $\mathbf{W}'_{n,\omega}$, where $i\prime$ and $j\prime$ represent the row and the column, respectively, of the element $e'_{i\prime,j\prime}$ in $\mathbf{W}'_{n,\omega}$; (2) The position (i, j) in the initial $\mathbf{W}_{n,\omega}$ corresponding to the position $(i\prime, j\prime)$ is calculated using $i = i\prime/K_1, j = j\prime/K_2$; (3) Sixteen elements $\{e_{m_k,n_l} \mid k, l \in 0, 1, 2, 3\}$ closest to the position (i, j) in $\mathbf{W}_{n,\omega}$, where m_k and n_l represents the row and column, respectively, of the element e_{m_k,n_l} in $\mathbf{W}_{n,\omega}$, are obtained; (4) According to the formula, $e'_{i\prime,j\prime} = \sum_{k=0}^{3}\sum_{l=0}^{3} e_{m_k,n_l}\text{Bi}(m_k - i)\text{Bi}(n_l - j)$, $e'_{i\prime,j\prime}$ is obtained;
Step 5	The process of executing bicubic interpolation on an initial discrete WVD is performed, and the matrix $\mathbf{W}'_{n,\omega}$ with the desired size is obtained.

The preprocessing process of the radar emitter signal was helpful for accelerating the training speed of the network and keeping the network more robust. This work also normalized the bicubic interpolated WVD. If the preprocessed radar emitter signal is denoted as **r**, the general route of radar emitter signal preprocessing is as shown in Figure 2.

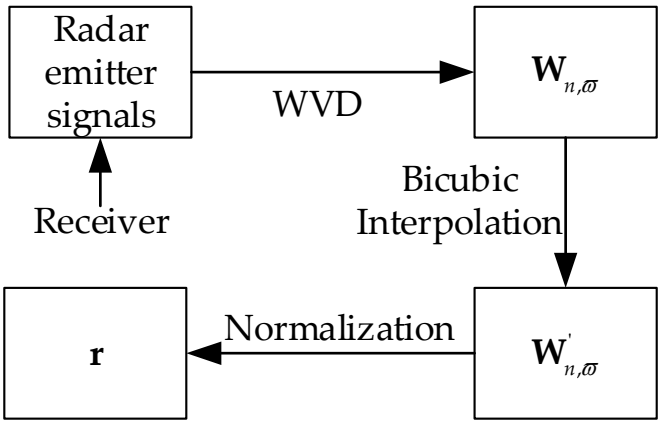

Figure 2. General route for radar emitter signal preprocessing.

3. Method

3.1. REI Scene

In real EW, it was difficult for an electronic reconnaissance UAV to capture much of the wanted radar data with labels, but it was easy to obtain numerous simulated radar data with labels. Using meta-learning technology, we fully utilized these simulated radar data to train the meta-learning network-based REI algorithm in the ESM system of UAVs, so the ESM system could quickly adapt to a new few-shot task. The above provided a new idea for UAVs to address the problem of few-shot REI in a real combat scene, which agreed with the REI scene in real EW. The REI scene assumed in this work is shown in Figure 3.

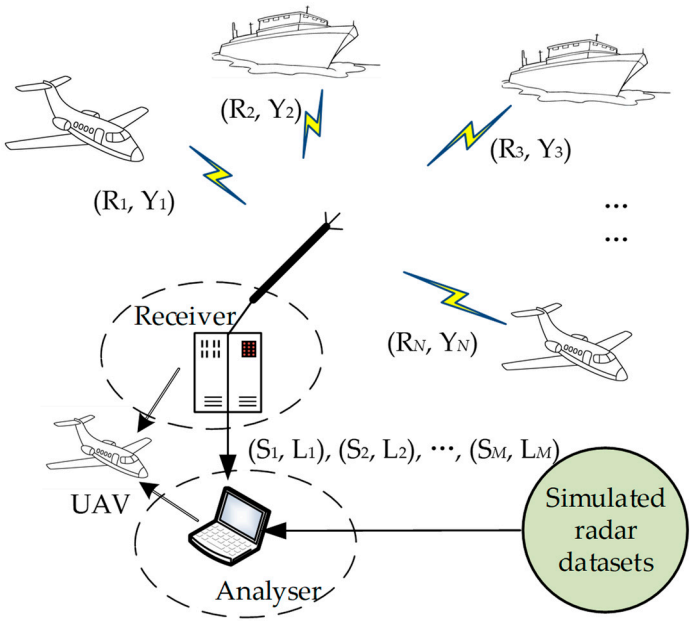

Figure 3. REI scene assumed in this work.

As shown in Figure 3, we have an electronic reconnaissance UAV, whose ESM system includes a signal analyzer and radar signal receiver. First, the REI method based on the meta-learning network in the signal analyzer is trained using simulated radar emitter data. The number of simulated radar emitter data can be large, so many few-shot REI tasks can be constructed using the simulated data. The simulation of the signals is based on common modulation styles, common pulse widths and common carrier frequencies. The meta-learning network in the signal analyzer can learn a feature extraction capability after being trained on the constructed simulated few-shot REI tasks. Furthermore, this feature extraction capability extracts the essential features of a class based on a few samples belonging to the same class. By comparing a sample to be identified with the essential features of all classes, the class of the sample can be determined. Second, the radar signal receiver in the UAV is used to capture data from various radar emitters on the battlefield. These radar emitters can be equipped on aircraft or ships, as shown in Figure 3. All radar emitter data captured by the radar signal receiver are preprocessed into samples that can be inputted into the meta-learning network-based REI method in the signal analyzer. Last, the meta-learning network that was previously trained on many simulated REI tasks is used to identify real radar samples captured with the radar signal receiver in the UAV.

In Figure 3, $S_1, S_2, \ldots$, and S_M are the data corresponding to M classes of simulated radar emitters, and $L_1, L_2, \ldots$, and L_M are their labels. There are many simulated samples in each of $S_1, S_2, \ldots$, and S_M that have been preprocessed using the method in Section 2. $R_1, R_2, \ldots$, and R_N in Figure 3 are the data corresponding to N classes of real radar emitters to be identified, and $Y_1, Y_2, \ldots$, and Y_N are their labels. The radar emitters may be equipped in army airplanes or ships. Additionally, note that M is greater than N.

Only a few samples with labels in each class of $R_1, \ldots$, or R_N are previously known by us, and the goal of the REI algorithm proposed in this work is to infer from which radar emitter a signal is derived when a UAV intercepts a signal from the abovementioned N classes of real radar emitters again.

3.2. REI Algorithm

A flowchart of the proposed REI algorithm in this work, which was used to solve the abovementioned few-shot REI problem, is shown in Figure 4.

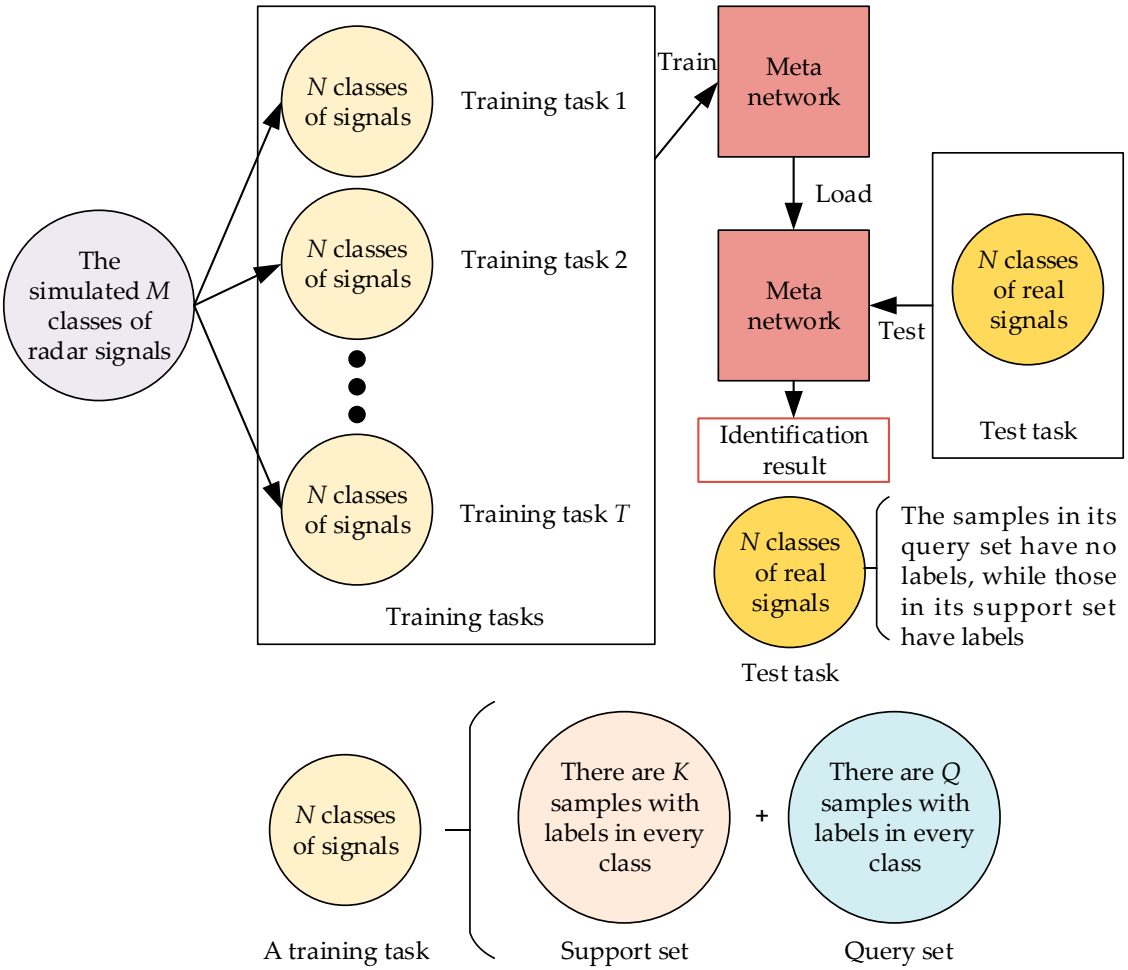

Figure 4. Flowchart of the proposed REI algorithm.

The following section introduces our proposed REI algorithm, as shown in Figure 4.

The first step is to build T training tasks. All training tasks are built with simulated radar signals as follows: First, N classes of signals from the simulated M classes of radar emitter signals are randomly selected. Then, from the selected N classes of simulated signals, K signals and another Q signals from every class are removed. After all the selected signals have been preprocessed, the support set consists of all the K signals belonging to every class, and the query set consists of all the Q preprocessed signals belonging to every class. Therefore, every training task contains a support set and query set, as shown in Figure 4. Hence, the T training tasks denoted as $\{Task_i | i = 1, 2, \ldots, T\}$ are obtained.

The second step is to train the meta-network named IRelNet. First, the network is built, the network weights are initialized, and the training epoch, denoted as $epoch$, is established. Then, IRelNet is trained using T training tasks. The trained IRelNet model with the best performance can be obtained. The trained IRelNet model can extract prototypes for each class based on only a few labeled samples. Therefore, the network can obtain prototypes for each class of real signals based on a small number of captured labeled radar signals.

The third step is to build the test task. First, K labeled signals from each class of radar emitters to be identified are selected. Additionally, N classes of radar emitters need to be identified. After all the $N \times K$ labeled signals have been preprocessed, the support set consists of $N \times K$ samples. Then, after the Q signals without labels have been preprocessed, the query set contains the Q samples from the query set. The labels of the Q samples are those that we want to obtain. The test task composed of the above support set and query set is built; it is denoted as Task$'$.

The fourth step is to identify radar emitters. First, IRelNet trained with the T training tasks is loaded, and the test task is input into IRelNet. IRelNet extracts all class prototypes based on the support set in the test task and then determines to which class a sample in the query set belongs by evaluating the differences between all class prototypes and the sample to be recognized in the query task. The radar emitter signals without labels in the query set of the test task can be identified using the proposed REI algorithm. Additionally, the classes of samples in the training tasks and in the test task do not need to be identical; these samples only need to belong to the same domain, which is supported by the advantage of meta-learning.

3.3. IRelNet

The IRelNet model proposed in this work consists of two parts: the embedding module and the relation module. After preprocessing the radar emitter samples, they were fed into the embedding module to obtain the hidden layer representation of these samples. The hidden layer representation of the samples in the support set and those of the samples in the query set were then concatenated in the relation module. The output of the relation module could then be used to determine to which class each sample in the query set belonged. The structure of IRelNet is illustrated in Figure 5, and the implementation framework was TensorFlow 2.4.0.

The following section provides a detailed description of our proposed network, including Figure 5. At the end of each processing step, we analyzed the changes in feature dimensions by using a sample with a dimension of (128, 128, 1) as the initial input for IRelNet.

(1) The input samples were mapped onto their shallow hidden features. All samples in the support set and test set were passed through a convolutional layer, denoted as 'Conv2D(64, 3, 'same')', a batch normalization layer, denoted as 'BatchNormalization', and an activation layer with the ReLu activation function, denoted as 'ReLu'. 'Conv2D(64, 3, 'same')' indicated that the convolutional layer had 64 convolutional kernels, each with a size of 3×3, and the convolutional operation could be padded with zeros. The purpose of this part of the network was to map the sample onto a hidden feature space and expand the feature dimension of the sample, which facilitates the subsequent network to extract the essential features of the sample. Additionally, if there was an input sample with the dimension (128, 128, 1), then the dimension of the sample's shallow hidden feature obtained via this step would be (128, 128, 64).

(2) The shallow hidden features were mapped onto deep reliable features. The shallow hidden features of all samples obtained in step 1) were then passed through B convolutional block 1 to obtain deep feature representations. Convolutional block 1 was the core component, and its detailed structure is illustrated in Figure 6.

First, the hidden features obtained via the last layer were input into a convolutional layer denoted as 'Conv2D(64, 3, 'same')' in Figure 6. Notably, '64' was the number of convolutional kernels, '3' indicated that the size of all the kernels was 3×3, and 'same' denoted that the convolutional operation could be padded with zeros. Additionally, the dimension of the output of this step was (128, 128, 64).

Second, the output of the convolutional layer was input into two paths, which were the global average pooling layer, denoted as 'GlobalAveragePooling2D' in Figure 6, and the global maximum pooling layer, denoted as 'GlobalMaxPooling2D' in Figure 6. The function of GlobalAveragePooling2D is described as follows: Its input was a feature map with 64 channels, and its output had 64 feature values. Each feature value was equal to the average value of all pixels in one channel of the input feature map. Similarly, the output of GlobalMaxPooling2D had 64 feature values. Each feature value was equal to the maximum value among all pixels in one channel of the input feature map. The two outputs of the global pooling layer were input into a convolutional layer, denoted as 'Conv2D(4, 1, 'same')', an activation layer, denoted as 'ReLu', and another convolutional layer, denoted as 'Conv2D(64, 1, 'same')' in Figure 6. Notably, '4' and '64' were the number of convolutional

kernels, '1' denoted that the size of all the kernels was 1 × 1, 'same' indicated that the convolutional operation could be padded with zeros, and 'ReLu' was the activation function in the activation layer. In addition, after passing through the global pooling layer, each channel of the feature map only had a single feature value with a size of 1 × 1. Hence, we required the usage of 1 × 1 Convolution2D operations. The dimensions of the results of the two paths were (1, 1, 64).

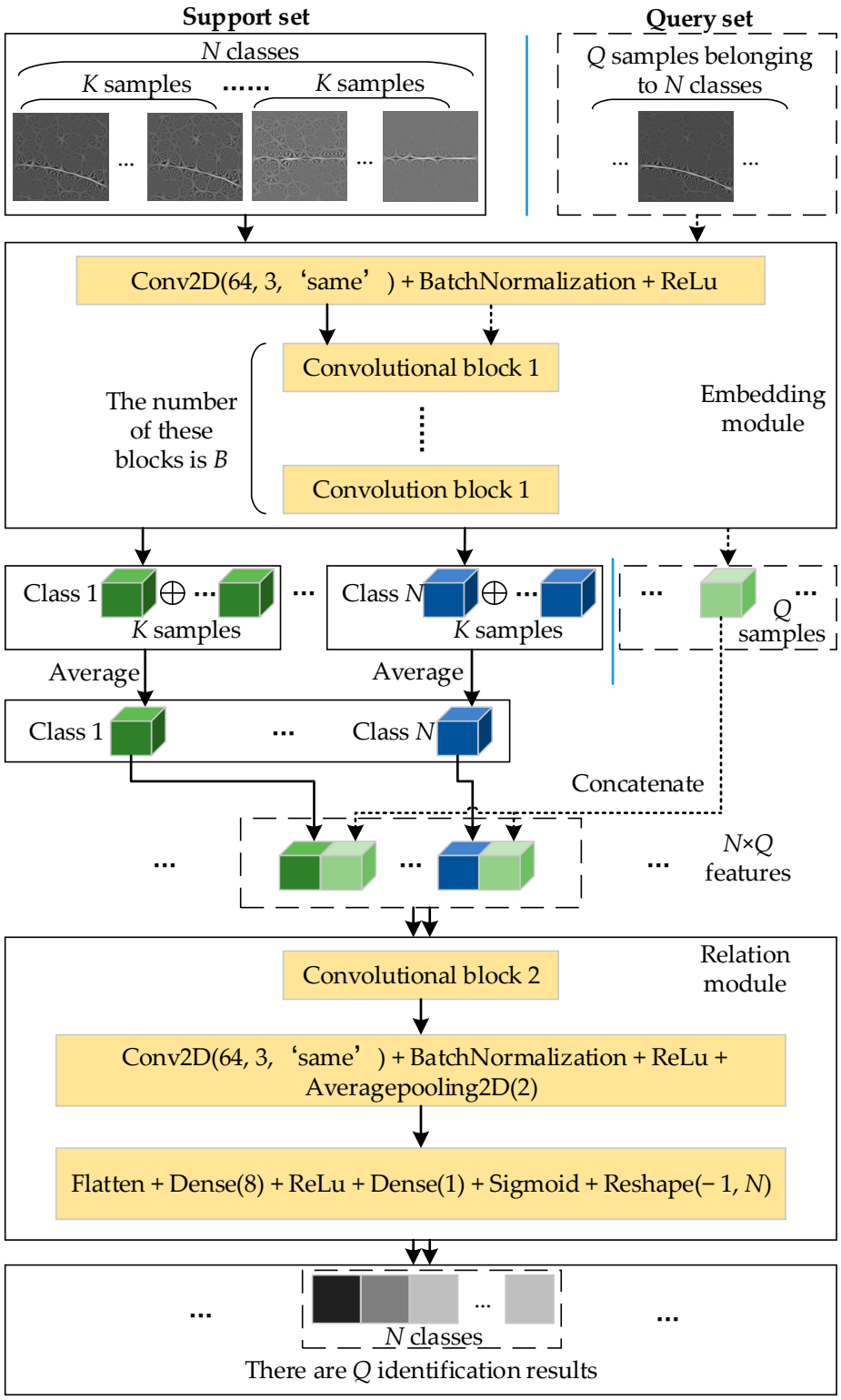

Figure 5. Detailed structure of IRelNet.

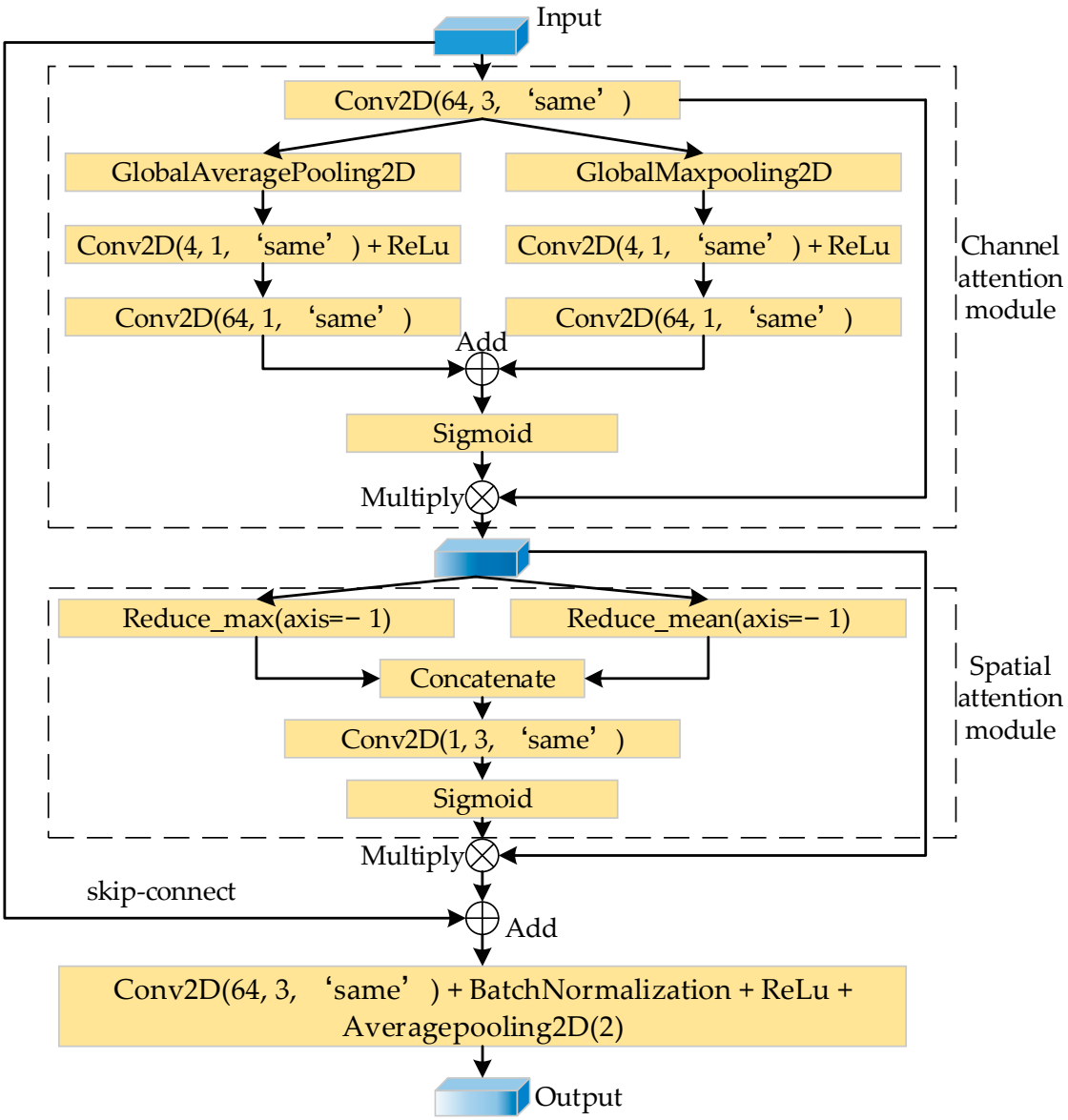

Figure 6. Detailed structure of convolutional block 1.

Third, the result by adding outputs of the two paths was passed into an activation layer, denoted as 'Sigmoid', which was the activation function. The dimension of the result of this step was (1, 1, 64).

Fourth, the output of the first convolutional layer in Figure 6 was multiplied by the output of the last 'Sigmoid' activation layer to obtain the subsequent input. The dimension of the result of this step was (128, 128, 64).

According to the first substep to the fourth substep above, the channel attention mechanism was incorporated into the network, allowing the essential features of samples in different channels to be emphasized differently. This substep enabled the proposed network to easily extract the features that could accurately represent the samples belonging to different classes.

Fifth, the output of the network with the channel attention mechanism was fed into two paths. In the first path, the channels were reduced based on the maximum value, which was denoted as 'Reduce_max(axis = −1)', while in the second path, the channels were reduced based on the mean value, which was denoted as 'Reduce_mean(axis = −1)', as shown in Figure 6. The dimensions of the results of the two paths were (128, 128, 1).

Sixth, the result obtained by concatenating the outputs of the above two paths was then fed into a convolutional layer, denoted as 'Conv2D(1, 3, 'same')', and an activation layer with a sigmoid activation function, denoted as 'Sigmoid'. Notably, '1' indicated the number of convolutional kernels, '3' indicated the size of the kernel was 3×3, and 'same' indicated that the convolutional operation could be padded with zeros. The dimension of the result of this step was (128, 128, 1).

Seventh, the output of the last step was then multiplied by the output of the network with the channel attention mechanism from the fourth step, thus allowing the result to generally represent the essential features of different classes of samples. The dimension of the result of this step was (128, 128, 64).

Now, according to the fifth substep to the seventh substep above, the spatial attention mechanism was incorporated into the network. With the incorporation of the spatial attention mechanism, the essential features of samples in different spatial areas could be highlighted differently, thus allowing the proposed network to easily capture the important features.

Eighth, the output of the network with the channel and spatial attention mechanism was added with the initial input of convolutional block 1, which is denoted as 'skip-connect' in Figure 6. The dimension of the result of this step was (128, 128, 64).

The network with the channel and spatial attention mechanism was slightly lengthy, which could potentially lead to a loss of some features of the samples. However, 'skip-connect' could prevent this loss by directly connecting the output of the network with the attention mechanism to its input.

Last, the output of the network with 'skip-connect' was fed into a convolutional layer denoted as 'Conv2D(64, 3, 'same')', a batch normalization layer denoted as 'BatchNormalization', an activation layer with a ReLu activation function, denoted as 'ReLu', and an average pooling layer, denoted as 'Averagepooling2D(2)'. 'Conv2D(64, 3, 'same')' indicated that the convolutional layer had 64 convolutional kernels, each with a size of 3×3 and that the convolutional operation could be padded with zeros. Notably, '(2)' in 'Averagepooling2D(2)' indicated that the size of the pooling was 2×2. The dimension of the result of this step was (64, 64, 64).

Thus far, the process of how a sample passed through convolution block 1 has been explained. Through B convolutional block 1, the network embedded the sample into its deep reliable feature representations.

(3) The feature representation of a class was calculated. The above 'Embedding module' in Figure 5 could be utilized to obtain the feature representation of each sample in the support set and query set. Then, the feature representation of a class could be calculated by taking the average of the K deep features of the K samples belonging to the class in the support set. In this way, the N feature representations of the N classes could be obtained. If the input sample passed through 4 convolutional blocks 1 in step 3, the dimension of the feature representation of a class was (8, 8, 64).

(4) The N feature representations of the N classes with Q feature representations of the Q samples in the query set were concatenated. The aim of this step was to build the relationship between the unknown samples in the query set and the labeled samples in the support set. Therefore, which class of labeled samples was the most similar to the samples to be identified was determined. The dimension of the concatenated feature representation was (8, 8, 128).

(5) Deeper features of the concatenated $N \times Q$ features were obtained. The concatenated $N \times Q$ features were then fed into convolutional block 2, as shown in Figure 5. Convolutional block 2 was another key component in the proposed network, and its detailed structure is illustrated in Figure 7.

First, one of the $N \times Q$ features was fed into a convolutional layer, denoted as 'Conv2D(64, 3, 'same')', a batch normalization layer, denoted as 'BatchNormalization', and an activation layer with the ReLu activation function, denoted as 'ReLu'. The dimension of the result of this step was (8, 8, 64).

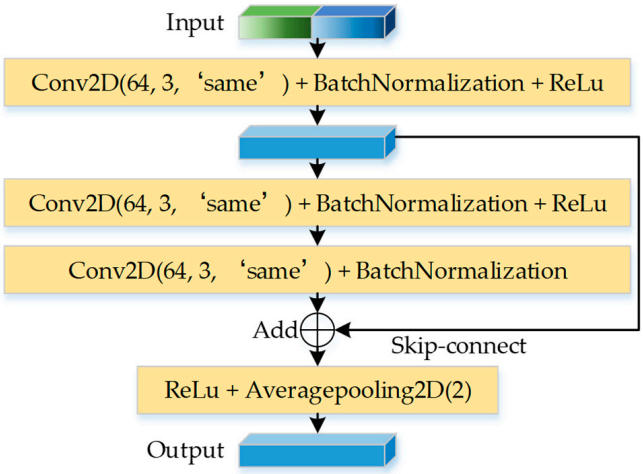

Figure 7. Detailed structure of convolutional block 2.

Second, the output of the first activation layer in Figure 7 was fed into a convolutional layer denoted as 'Conv2D(64, 3, 'same')', a batch normalization layer, denoted as 'BatchNormalization', an activation layer with the ReLu activation function, denoted as 'ReLu', a convolutional layer, denoted as 'Conv2D(64, 3, 'same')', and a batch normalization layer, denoted as 'BatchNormalization'. The dimension of the result of this step was (8, 8, 64).

Third, the output of the above second step was added to the output of the above first step, which was also referred to as 'skip-connect', as shown in Figure 7. This 'skip-connect' allowed the reuse of reliable features obtained from the embedding module, thus avoiding the overfitting of the network during training. The dimension of the result of this step was (8, 8, 64).

Last, the deep feature obtained from the previous step was fed into an activation layer with the ReLu activation function, denoted as 'ReLu', and an average pooling layer, denoted as 'Averagepooling2D(2)'. The dimension of the result of this step was (4, 4, 64). The subsequent processing steps were performed on many samples; therefore, we would no longer show the changes in the feature dimensions of the initial input sample.

At this point, the process of how a concatenated feature passed through convolutional block 2 was explained. The purpose of convolutional block 2 was to extract deeper features while also helping to prevent the network from overfitting and accelerating the training process.

(6) The $N \times Q$ features were scored. The output of convolutional block 2 was fed into a convolutional layer, denoted as 'Conv2D(64, 3, 'same')', a batch normalization layer, denoted as 'BatchNormalization', an activation layer with the ReLu activation function, denoted as 'ReLu', and an average pooling layer, denoted as 'Averagepooling2D(2)'. To score the concatenated $N \times Q$ feature, the output of the pooling layer was then fed into a flattening layer, denoted as 'Flatten', a fully connected layer with 8 neurons, denoted as 'Dense(8)', an activation layer with the ReLu activation function, denoted as 'ReLu', another fully connected layer with 1 neuron, denoted as 'Dense(1)', another activation layer with the sigmoid activation function, denoted as 'Sigmoid', and a reshaping layer, denoted as 'Reshape(-1. N)'. 'Reshape(-1, N)' indicates reshaping the dimension of the deep feature into the shape of $1 \times N$.

(7) The classes of the samples in the query set were deduced. When a sample from the query set is fed into the above network, the output of the reshaping layer contains N elements. The position of the maximum value among these N elements indicated the class of the sample.

In summary, the proposed IRelNet model was a powerful network that could embed a sample into a reliable feature representation and score the relation between a sample in the support set and a sample in the query set. This network enabled drone troops to identify radar emitters with only a few labeled samples. The channel attention mechanism and

spatial attention mechanism were used to enhance the network's ability to extract deep features, while the skip connection was also used to avoid overfitting the network.

4. Experiments

4.1. Datasets

The simulated training dataset included eight classes of radar emitter signals, namely, linear frequency modulation (LFM), nonlinear frequency modulation (NLFM), continuous wave (CW), frequency diversity (FD), binary phase shift keying (BPSK), binary frequency shift keying (BFSK), binary amplitude shift keying (BASK), Barker code and linear frequency hybrid modulation (Baker-LFM). The parameters of these eight classes of simulated radar emitter signals are presented in Table 2.

Table 2. Parameters of these eight classes of simulated radar emitter signals.

Class	Parameters
LFM	Frequency bandwidth ∈ [15, 20] MHz; carrier frequency ∈ [25, 30] MHz
NLFM	Frequency bandwidth ∈ [10, 15] MHz; carrier frequency ∈ [25, 30] MHz
CW	carrier frequency∈ [25, 30] MHz
FD	Frequency F_1 ∈ [5, 10] MHz; frequency F_2 ∈ [15, 20] MHz; frequency F_3 ∈ [25, 30] MHz
BPSK	Phase coding sequence [1, 1, 1, 0, 0, 1, 0]; carrier frequency ∈ [25, 30] MHz
BFSK	Frequency coding sequence [1, 1, 0, 0, 0, 1, 0, 0, 1]; frequency F_1 ∈ [10, 15] MHz; frequency F_2 ∈ [25, 30] MHz
BASK	Amplitude coding sequence [1, 1, 1, 0, 1, 0, 0, 1, 0, 0]; carrier frequency ∈ [25, 30] MHz
Baker-LFM	Barker coding sequence [1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0]; frequency bandwidth ∈ [10, 15] MHz; carrier frequency ∈ [25, 30] MHz
All signals have a PW of 10 μs and a sampling rate of 100 MHz	

Notably, '0' in the phase coding sequence of BPSK indicated that the phase of the modulated radar signal was 0, while '1' indicated that the phase was π. Similarly, '0' in the frequency coding sequence of BFSK indicated that the frequency of the modulated radar signal was F_1, while '1' indicated that the frequency was F_2. The number '0' in the amplitude coding sequence of BASK indicated that the amplitude of the modulated radar signal was 0, while '1' indicated that the amplitude was 1. Last, '0' in the Barker coding sequence of Baker-LFM indicated that the phase of the modulated radar signal was 0, while '1' indicated that the phase was π.

The test dataset in this work contained six classes of radar emitter signals that were generated with our hardware equipment. The parameters of the pulse signal were set via the hardware equipment, and the receiver was used to acquire radar signals under noncooperative conditions with a sampling rate of 50 MHz. The scene of receiving radar signals is shown in Figure 8.

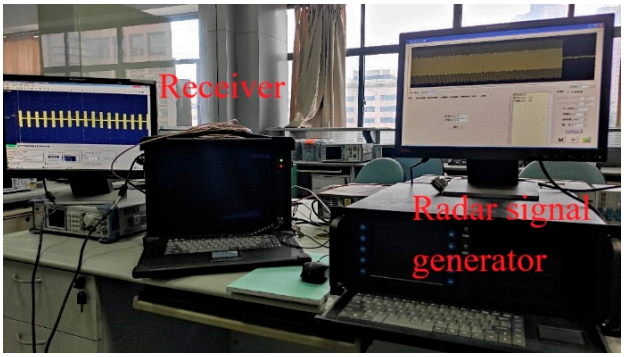

Figure 8. The scene of receiving radar signals.

All the simulated and received radar signals were preprocessed using the method described in Section 2. The dimension of the training dataset was (8, 50, 128, 128, 1), indicating that there were 8 classes of samples, with 50 samples in each class, and the size of each sample was (128, 128, 1). Similarly, the dimension of the test dataset was (6, 30, 128, 128, 1), indicating that there were 6 classes of samples, with 30 samples in each class, and the size of each sample was (128, 128, 1).

As shown in Figure 4 of Section 3.2, the eight classes of simulated radar signals were used to construct the training tasks, and six classes of pseudo-real radar signals generated with our hardware device were used to construct the testing tasks, which is consistent with the application scenario of the proposed REI algorithm. In the application scenario of our REI algorithm, simulated signals are used to train the meta-learning network, and real radar emitters are identified based on the trained meta-learning network. The classes of signals in different training tasks may differ. The training of the meta-learning network requires differences in the training tasks; otherwise, the network will not vary from traditional networks.

4.2. Model Optimization

To obtain the best configurations, the performance of the proposed IRelNet model with different configurations on the 4-way 6-shot test tasks was evaluated. Additionally, '4' indicates that N = 4 and K = 6 in Figure 4. The number of training epochs was set to 7. The results, shown in Table 3, indicate that the number of training tasks, T, and the number of convolutional blocks 1, B, had a significant influence on the performance of the model. The SNR of the signals in the training tasks was in the range of [−4, 0] dB with a step size of 1 dB, represented as [−4:1:0] dB. If there was no special declaration, Gaussian white noise was added to the radar emitter signals to obtain various samples with different SNRs in this work. We suggest that in the field of military radar, the detection distance is typically far, and the received radar signals are usually mixed with Gaussian white noise. The Gaussian distribution's mean was set to 0, and its variance was determined via SNR.

Table 3. Performance of the IRelNet model with different configurations on the test tasks (%).

Configurations		−6 dB	−4 dB	−2 dB	0 dB	2 dB	4 dB
T = 500	B = 3	0.6521	0.6772	0.7171	0.7541	0.7850	0.8152
T = 1000	B = 3	0.7433	0.7246	0.7372	0.7703	0.7903	0.7979
T = 1500	B = 3	0.5628	0.5892	0.6935	0.7294	0.7847	0.7812
T = 500	B = 4	0.5634	0.7028	0.7806	0.7980	0.8568	0.8685
T = 1000	B = 4	0.6550	0.7552	0.8047	0.8575	0.8807	0.9035
T = 1500	B = 4	0.5974	0.7206	0.8001	0.8322	0.8776	0.8768
T = 500	B = 5	0.4332	0.5238	0.5233	0.5643	0.5622	0.5795
T = 1000	B = 5	0.5736	0.6806	0.7135	0.7591	0.7870	0.8006
T = 1500	B = 5	0.6313	0.6678	0.7352	0.7670	0.7707	0.7687

Under various SNR conditions, the performance of the IRelNet model is relatively better when T = 1000 and B = 4.

Table 3 shows that the performance of the IRelNet model did not always improve with an increase in the number of training tasks. When the structure of IRelNet was simpler, such as when B was equal to 3, the identification performance of the IRelNet model decreased with an increase in the number of training tasks. On the other hand, when the structure of IRelNet was more complex, such as when B was equal to 5, the identification performance of the IRelNet model improved with an increase in the number of training tasks. Additionally, Table 3 shows that a larger number of attention modules in IRelNet did not always lead to better performance. The larger B was, the more complex IRelNet was and the more training tasks were needed. This trend would lead to an increase in the time cost required for training IRelNet, which was not desirable for real EW applications.

The experimental results suggest that T could be set to 1000 and B could be set to 4, which could reduce the time cost and improve the stability of IRelNet.

4.3. Influence of SNR

IRelNet was trained using samples with varying SNRs, which could potentially lead to different performance results. To investigate the effect of SNR on the performance of IRelNet, different levels of Gaussian white noise were added to the simulated radar emitter samples. The identification performance of IRelNet trained using samples with varying SNRs is demonstrated in Figure 9 for the 4-way 6-shot test tasks composed of samples with different SNRs.

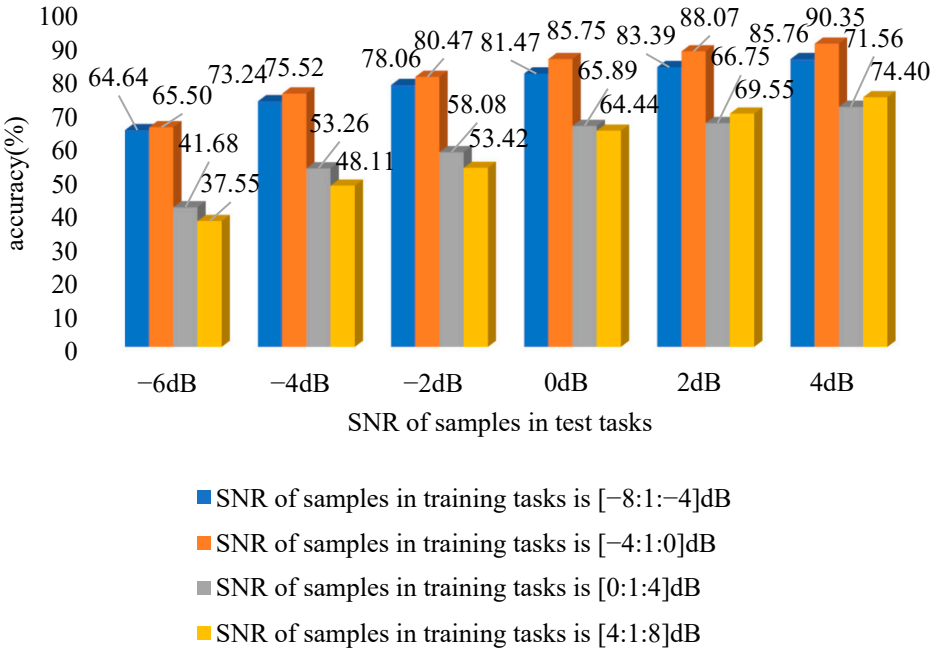

Figure 9. Identification performance of IRelNet trained using samples with varying SNRs.

Figure 9 shows that the trained IRelNet model had the best performance when the SNRs of the samples in the training tasks were within the range of [−4:1:0] dB. The experimental results indicate that if the SNRs of samples in the training tasks were too low, it was not conducive to the fitting of the network and the learning of sample features. On the other hand, if the SNRs of samples in the training tasks were too high, it was not conducive to the robustness of the network and the adaptation of the network to new identification tasks. Thus, the SNRs of samples in the training tasks had a significant impact on the performance of IRelNet. Therefore, simulated samples with SNRs within the range of [−4:1:0] dB should be preferentially employed for training tasks of the network.

In addition, this section further explores the impact of Rayleigh fading channels on the performance of IRelNet via additional experiments. In this experiment, the Rayleigh fading channel had 3 transmission paths with path delays of 0 μs, 0.25 μs and 0.5 μs. The identification accuracy of IRelNet under different path average gain conditions is shown in Table 4.

Table 4. Identification accuracy of IRelNet under different path average gain conditions (%).

Training Tasks \ Test Tasks	0 dB, 1 dB, 2 dB	−2 dB, −1 dB, 0 dB	−4 dB, −3 dB, −2 dB
0 dB, 1 dB, 2 dB	66.53	64.85	62.42
−2 dB, −1 dB, 0 dB	76.80	73.75	75.12
−4 dB, −3 dB, −2 dB	55.75	53.77	43.19

When the path average gain is '0 dB, 1 dB, 2 dB', the performance of the IRelNet model is the best.

Table 4 shows that IRelNet trained with samples passing through 3 channel paths with average gains of −2 dB, −1 dB and 0 dB achieved the best performance. Samples with certain noise levels can improve the robustness of IRelNet. Furthermore, Table 4 indicates that the trained IRelNet model can more easily identify samples passing through channel paths with larger average gains. The larger the average gain is, the greater the distinguishability between two different samples. The experiment demonstrates that IRelNet can adapt to Rayleigh fading channels, which further confirms the applicability of the proposed method to practical scenarios.

In addition, in a real-world scenario, the SNR dynamically varies from different signal sources. To improve the capability of IRelNet to handle such situations, it might be beneficial to construct training tasks using simulated data with a larger SNR range or to increase the number of training tasks used to train IRelNet. However, these improvements cannot evaluate the performance limit of IRelNet, which is the main purpose of this section.

4.4. Influence of Classes N and the Number K

The classes of target radar emitters to be identified and the classes of simulated samples utilized for training tasks should generally be identical. However, the number of classes of radar emitters to be identified with an ESM system of drones often varies in different EW scenes. Therefore, it was necessary to investigate the identification performance of the proposed method with different classes of samples that needed to be identified. Furthermore, the number of samples K belonging to one class that needs to be identified in various EW scenes would also affect the identification performance of the proposed method. When the SNRs of the samples in the training tasks were in the range of [−4:1:0] dB and the SNR of the samples in the test task was 4 dB, the identification performance of IRelNet corresponding to different N-way K-shot tasks was as shown in Figure 10.

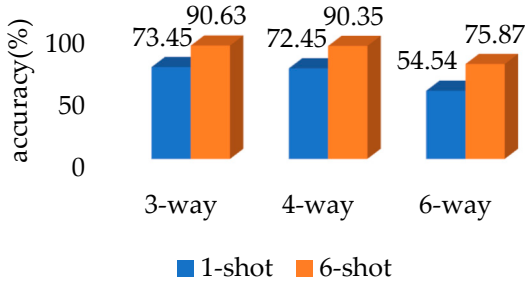

Figure 10. Identification performance of IRelNet corresponding to different N-way K-shot tasks.

Figure 10 shows that when N was fixed and K was greater, the identification accuracy of IRelNet increased. When the number of samples K used to train IRelNet was larger, the network could more accurately extract the essential features of a known class of radar emitter, allowing the features of signals to be identified to better match the deep features of the known class. When K was fixed and N was larger, the identification accuracy of IRelNet decreased. IRelNet was able to more easily extract features of different classes when there were fewer classes. Therefore, it was recommended to train IRelNet with more samples and to learn to identify fewer classes for IRelNet at a time in real-world applications.

4.5. Performance of Different Methods

To the best of our knowledge, there have been a few REI methods that could be used in scenarios with very few labeled samples. Some methods, such as PN, RN, RRSARNet [39] and IRelNet, could be applied to the EW scene that we established. Therefore, this section mainly compares the performances of PN, RN, RRSARNet and IRelNet. The feature embedding network in the PN included 5 convolutional layers. Samples with different SNRs were used to construct 4-way 6-shot test tasks. When the training epoch was 7, the identification performance of different methods on the 4-way 6-shot test tasks was as shown in Figure 11.

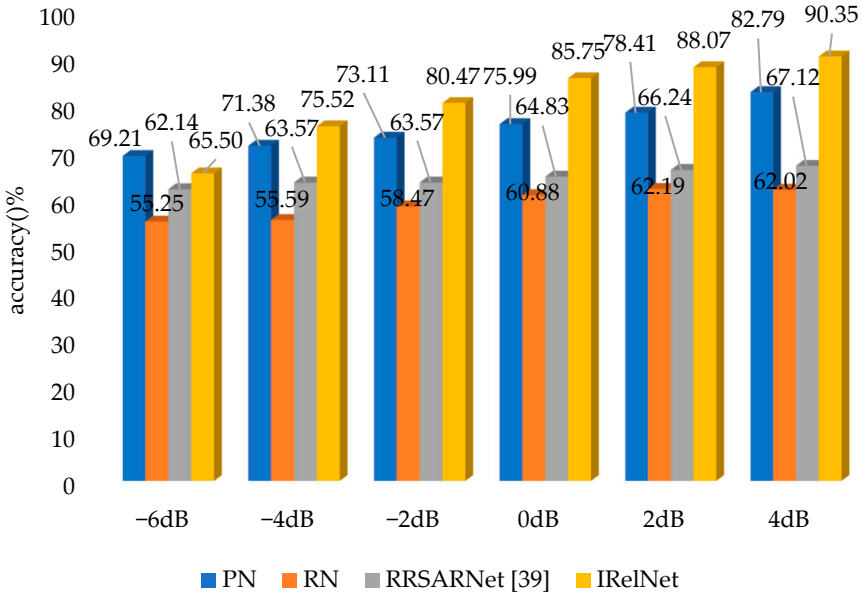

Figure 11. Identification performances of different methods on the 4-way 6-shot test tasks.

Figure 11 shows that the identification performance of IRelNet on the test tasks was higher than that of the RN, indicating that the attention mechanism and skip-connect mechanism introduced in the RN could effectively improve the feature-embedding and matching abilities of the RN. The identification performance of the PN on the test tasks was higher than those of the RN and RRSARNet and lower than that of IRelNet. The PN with a fixed metric distance was easier to train; thus, it might have better performance than the RN when the number of training tasks is fixed.

We analyzed the computational effort of the aforementioned methods. Additionally, the time required for the PN, the RN, RRSARNet and IRelNet to train a few-shot REI task reflects their computational effort. Table 5 displays the corresponding times required for the experiment conducted on a 12th Gen Intel(R) Core(TM) i9-12900KF, NVIDIA GeForce RTX 3090, and TensorFlow 2.4.0. The abovementioned computational resources are the same computational resources used in all experiments in this work.

Table 5. Time required for the PN, the RN, RRSARNet and IRelNet to train a few-shot REI task.

Method	PN	RN	RRSARNet	IRelNet
Time	1.5280 s	2.6440 s	3.5430 s	5.9160 s

Table 5 reveals that the time required for each method to train a few-shot REI task is in the order of seconds, indicating that the training for all methods can be completed in a few hours, thus satisfying the demands of most scenarios. We primarily employed the trained network to identify radar samples, and the network's identification time for a sample did not exceed 1 s. In summary, the time complexity of the proposed IRelNet model satisfies the needs of most few-shot REI scenarios.

Based on all the experiments conducted, IRelNet exhibited the best identification performance on the test tasks. The IRelNet-based method effectively addresses the few-shot REI problem and offers novel technical support to the ESM system of UAVs.

5. Conclusions

In this work, an REI method based on IRelNet was proposed. This method was designed to be embedded in UAVs to enable them to identify radar emitters. The scene of how a UAV employed IRelNet to identify radar emitters was described in detail. Additionally, the attention mechanism and feature reuse mechanism were implemented in IRelNet to

improve its performance. Experiments were conducted to optimize the hyperparameters of IRelNet, and the results demonstrate that the proposed method was capable of effectively solving the problem of REI with few labeled samples. This work presented novel approaches and techniques for REI in the field of electronic reconnaissance UAVs, which could facilitate the further integration of signal processing, meta-learning and UAVs.

However, the primary limitation of IRelNet is that the samples utilized in the training tasks must possess some similarities to those utilized in the test tasks. Without any similarity, it becomes challenging to train a network that can exhibit good performance. IRelNet does not endorse incremental learning and hence is unable to handle radar signals from unknown classes. In the future, we could strive to enhance IRelNet to enable it to identify radar emitters from unknown classes, which would more aptly fulfill the requirements of real-world, electronic reconnaissance UAVs. Using information from received radar signals to dynamically improve the algorithm is also an excellent approach. Furthermore, additional studies could be conducted to evaluate the performance of IRelNet in real-world scenarios to assess its applicability. When IRelNet is embedded in UAVs and shows excellent performance on real signals, further investigating its scalability to larger-scale systems will have significance. Moreover, adversarial attack techniques are likely to exist in future electronic warfare, and, therefore, it is necessary to evaluate the resilience of the meta-learning-based network to such attacks when developing deep-learning-based REI methods. To implement the proposed REI method on actual equipment, it is imperative to address the challenges of managing enormous data flow and signal source deinterleaving.

Author Contributions: Conceptualization, Z.W. and D.B.; methodology, Z.W. and M.D.; validation, Z.W. and J.P.; formal analysis, Z.W. and J.P.; investigation, Z.W.; resources, D.B. and J.P.; data curation, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W. and M.D.; visualization, Z.W.; supervision, D.B.; project administration, J.P. and Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 62071476, and the Postgraduate Scientific Research Innovation Project of Hunan Province under Grant CX20220032 and CX20220014.

Data Availability Statement: Due to the data being generated by specific hardware device and involving our privacy, it is not convenient to disclose it.

Conflicts of Interest: The authors declare that they have no conflict of interest to report regarding the present study.

References

1. Sharma, P.; Sarma, K.K.; Mastorakis, N.E. Artificial intelligence aided electronic warfare systems- recent trends and evolving applications. *IEEE Access* **2020**, *8*, 224761–224780. [CrossRef]
2. Wiley, R.G. *Electronic Intelligence: The Analysis of Radar Signals*, 1st ed.; Artech House: Dedham, MA, USA, 1982.
3. Wilkinson, D.R.; Watson, A.W. Use of metric techniques in ESM data processing. *IEE Proc. F (Commun. Radar Signal Process.)* **1985**, *132*, 229–232. [CrossRef]
4. Grant, P.M.; Collins, J.H. Introduction to electronic warfare. *IEE Proc. F (Commun. Radar Signal Process.)* **1982**, *129*, 113–132. [CrossRef]
5. Yang, N.; Zhang, B.; Ding, G.; Wei, Y.; Wang, J.; Guo, D. Specific emitter identification with limited samples: A model-agnostic meta-learning approach. *IEEE Commun. Lett.* **2021**, *26*, 345–349. [CrossRef]
6. Al-Emadi, S.; Al-Senaid, F. Drone detection approach based on radio-frequency using convolutional neural network. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020.
7. Nemer, I.; Sheltami, T.; Ahmad, I.; Yasar, A.U.-H.; Abdeen, M.A.R. RF-based UAV detection and identification using hierarchical learning approach. *Sensors* **2021**, *21*, 1947. [CrossRef] [PubMed]
8. Gan, J.; Hu, A.; Kang, Z.; Qu, Z.; Yang, Z.; Yang, R.; Wang, Y.; Shao, H.; Zhou, J. SAS-SEINet: A SNR-Aware Adaptive Scalable SEI Neural Network Accelerator Using Algorithm–Hardware Co-Design for High-Accuracy and Power-Efficient UAV Surveillance. *Sensors* **2022**, *22*, 6532. [CrossRef]
9. Al-Sa'd, M.F.; Al-Ali, A.; Mohamed, A.; Khattab, T.; Erbad, A. RF-Based Drone Detection and Identification Using Deep Learning Approaches: An Initiative towards a Large Open Source Drone Database. *Future Gener. Comput. Syst.* **2019**, *100*, 86–97. [CrossRef]

10. Visnevski, N.; Krishnamurthy, V.; Wang, A.; Haykin, S. Syntactic modeling and signal processing of multifunction radars: A stochastic context-free grammar approach. *Proc. IEEE* **2007**, *95*, 1000–1025. [CrossRef]
11. Visnevski, N.; Haykin, S.; Krishnamurthy, V.; Dilkes, F.A.; Lavoie, P. Hidden Markov models for radar pulse train analysis in electronic warfare. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA, 23 March 2005.
12. Li, C.; Wang, W.; Wang, X. A method for extracting radar words of multi-function radar at data level. In Proceedings of the IET International Radar Conference 2013, Xi'an, China, 14–16 April 2013.
13. Matuszewski, J. The analysis of modern radar signals parameters in electronic intelligence system. In Proceedings of the 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, 23–26 February 2016.
14. Cai, J.; Li, C.; Zhang, H. Modulation recognition of radar signal based on an improved CNN model. In Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 19–20 October 2019.
15. Wu, J.; Zhong, Y.; Chen, A. Radio modulation classification using STFT spectrogram and CNN. In Proceedings of the 2021 7th International Conference on Computer and Communications (ICCC), Chengdu, China, 10–13 December 2021.
16. Chen, K.; Zhang, J.; Chen, S.; Zhang, S.; Zhao, H. Automatic modulation classification of radar signals utilizing X-net. *Digit. Signal Process.* **2022**, *123*, 103396. [CrossRef]
17. Hou, C.; Fang, C.; Lin, Y.; Li, Y.; Zhang, J. Implementation of a CNN identifying modulation signals on an embedded SoC. In Proceedings of the 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, 9–12 August 2020.
18. Hou, C.; Li, Y.; Chen, X.; Zhang, J. Automatic modulation classification using KELM with joint features of CNN and LBP. *Phys. Commun.* **2021**, *45*, 101259. [CrossRef]
19. Zhang, M.; Diao, M.; Guo, L. Convolutional neural networks for automatic cognitive radio waveform recognition. *IEEE Access* **2017**, *5*, 11074–11082. [CrossRef]
20. Kong, S.H.; Kim, M.; Hoang, L.M.; Kim, E. Automatic LPI radar waveform recognition using CNN. *IEEE Access* **2018**, *6*, 4207–4219. [CrossRef]
21. Qu, Z.; Mao, X.; Deng, Z. Radar signal intra-pulse modulation recognition based on convolutional neural network. *IEEE Access* **2018**, *6*, 43874–43884. [CrossRef]
22. Tian, X.; Sun, X.; Yu, X.; Li, X. Modulation pattern recognition of communication signals based on fractional low-order Choi-Williams distribution and convolutional neural network in impulsive noise environment. In Proceedings of the 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 16–19 October 2019.
23. Zhu, M.; Li, Y.; Pan, Z.; Yang, J. Automatic modulation recognition of compound signals using a deep multi-label classifier: A case study with radar jamming signals. *Signal Process.* **2020**, *169*, 107393. [CrossRef]
24. Sun, W.; Wang, L.; Sun, S. Radar emitter individual identification based on convolutional neural network learning. *Math. Probl. Eng.* **2021**, *2021*, 5341940. [CrossRef]
25. Peng, L.; Qu, W.; Zhao, Y.; Wu, Y. A multi-level network for radio signal modulation classification. In Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, Sanya, China, 19 December 2019.
26. Shao, G.; Chen, Y.; Wei, Y. Convolutional neural network-based radar jamming signal classification with sufficient and limited samples. *IEEE Access* **2020**, *8*, 80588–80598. [CrossRef]
27. O'Shea, T.J.; West, N.; Vondal, M.; Clancy, T.C. Semi-supervised radio signal identification. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), Pyeongchang-gun, Republic of Korea, 19–22 February 2017.
28. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-Learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5149–5169. [CrossRef]
29. Vanschoren, J. Meta-learning: A survey. *arXiv* **2018**, arXiv:1810.03548.
30. Li, P. Research on radar signal recognition based on automatic machine learning. *Neural Comput. Appl.* **2020**, *32*, 1959–1969. [CrossRef]
31. Finn, C.B. Learning to Learn with Gradients. Ph.D. Dissertation, Department Computer Science, University of California, Berkeley, CA, USA, 2018.
32. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a "Siamese" time delay neural network. In Proceedings of the 6th International Conference on Neural Information Processing Systems, Denver, CO, USA, 29 November 1993.
33. Zagoruyko, S.; Komodakis, N. Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
34. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5 December 2016.
35. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the part of Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017.
36. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.

37. Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010.
38. Huang, J.; Wu, B.; Li, P.; Li, X.; Wang, J. Few-shot learning for radar emitter signal recognition based on improved prototypical network. *Remote Sens.* **2022**, *14*, 1681. [CrossRef]
39. Lang, P.; Fu, X.; Martorella, M.; Dong, J.; Qin, R.; Feng, C.; Zhao, C. RRSARNet: A novel network for radar radio sources adaptive recognition. *IEEE Trans. Veh. Technol.* **2021**, *70*, 11483–11498. [CrossRef]
40. Zhang, Z.; Li, Y.; Zhai, Q.; Li, Y.; Gao, M. Few-shot learning for fine-grained signal modulation recognition based on foreground segmentation. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2281–2292. [CrossRef]
41. Sun, G. RF Transmitter identification using combined Siamese networks. *IEEE Tran. Instrum. Meas.* **2022**, *71*, 8000813. [CrossRef]
42. Dong, Y.; Jiang, X.; Zhou, H.; Lin, Y.; Shi, Q. SR2CNN: Zero-shot learning for signal recognition. *IEEE Trans. Signal Process.* **2021**, *69*, 2316–2329. [CrossRef]
43. Xu, H.; Wang, J.; Li, H.; Ouyang, D.; Shao, J. Unsupervised meta-learning for few-shot learning. *Pattern Recognit.* **2021**, *116*, 107951. [CrossRef]
44. Gong, J.; Xu, X.; Lei, Y. Unsupervised specific emitter identification method using radio-frequency fingerprint embedded InfoGAN. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2898–2913. [CrossRef]
45. Cao, R.; Cao, J.; Mei, J.P.; Yin, C.; Huang, X. Radar emitter identification with bispectrum and hierarchical extreme learning machine. *Multimed. Tools Appl.* **2019**, *78*, 28953–28970. [CrossRef]
46. Yan, X.; Liu, G.; Wu, H.C.; Zhang, G.; Wang, Q.; Wu, Y. Robust modulation classification over α-stable noise using graph-based fractional lower-order cyclic spectrum analysis. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2836–2849. [CrossRef]
47. Yan, X.; Liu, G.; Wu, H.C.; Feng, G. New automatic modulation classifier using cyclic-spectrum graphs with optimal training features. *IEEE Commun. Lett.* **2018**, *22*, 1204–1207. [CrossRef]
48. Sun, L.; Wang, X.; Huang, Z.; Li, B. Radio frequency fingerprint extraction based on feature inhomogeneity. *IEEE Internet Things J.* **2022**, *9*, 17292–17308. [CrossRef]
49. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [CrossRef]
50. Liu, M.; Liao, G.; Zhao, N.; Song, H.; Gong, F. Data-driven deep learning for signal classification in industrial cognitive radio networks. *IEEE Trans. Ind. Inform.* **2021**, *17*, 3412–3421. [CrossRef]