remote sensing

MDPI

Article

Open-Set Recognition Model for SAR Target Based on Capsule Network with the KLD

Chunyun Jiang, Huiqiang Zhang, Ronghui Zhan *, Wenyu Shu and Jun Zhang

National Key Laboratory of Automatic Target Recognition, College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China; jiangchunyun18@nudt.edu.cn (C.J.); zhanghq22@nudt.edu.cn (H.Z.); shuwenyu23@nudt.edu.cn (W.S.); zhangjun@nudt.edu.cn (J.Z.)
* Correspondence: zhanrh@nudt.edu.cn

Abstract: Synthetic aperture radar (SAR) automatic target recognition (ATR) technology has seen significant advancements. Despite these advancements, the majority of research still operates under the closed-set assumption, wherein all test samples belong to classes seen during the training phase. In real-world applications, however, it is common to encounter targets not previously seen during training, posing a significant challenge to the existing methods. Ideally, an ATR system should not only accurately identify known target classes but also effectively reject those belonging to unknown classes, giving rise to the concept of open set recognition (OSR). To address this challenge, we propose a novel approach that leverages the unique capabilities of the Capsule Network and the Kullback-Leibler divergence (KLD) to distinguish unknown classes. This method begins by deeply mining the features of SAR targets using the Capsule Network and enhancing the separability between different features through a specially designed loss function. Subsequently, the KLD of features between a testing sample and the center of each known class is calculated. If the testing sample exhibits a significantly larger KLD compared to all known classes, it is classified as an unknown target. The experimental results of the SAR-ACD dataset demonstrate that our method can maintain a correct identification rate of over 95% for known classes while effectively recognizing unknown classes. Compared to existing techniques, our method exhibits significant improvements.

Keywords: automatic target recognition (ATR); capsule network; Kullback–Leibler divergence (KLD); open-set recognition (OSR)

Citation: Jiang, C.; Zhang, H.; Zhan, R.; Shu, W.; Zhang, J. Open-Set Recognition Model for SAR Target Based on Capsule Network with the KLD. *Remote Sens.* **2024**, *16*, 3141. https://doi.org/10.3390/rs16173141

Academic Editor: Dusan Gleich

Received: 10 July 2024
Revised: 23 August 2024
Accepted: 25 August 2024
Published: 26 August 2024

1. Introduction

Synthetic aperture radar (SAR) is an active radar observation system known for its all-weather, all-day advantages and is widely used in military and civilian applications. In the military domain, a crucial task is to identify targets in SAR images, such as military vehicles, tanks, ships, and aircraft. However, due to its unique imaging mechanism, SAR images are more challenging to interpret compared to optical images. The manual interpretation of SAR images requires significant manpower and material resources; therefore, it is not suitable for applications on unmanned platforms such as aircraft, missiles, and satellites. Therefore, SAR automatic target recognition (ATR) has been proposed. In recent years, SAR ATR has become a hot research area, continuously evolving and achieving promising results.

The methods for SAR ATR can be primarily categorized into traditional approaches and deep learning-based methods. Traditional approaches predominantly adhere to a framework that combines handcrafted features with classifiers. Commonly used features include geometric features [1,2], transform domain features [3–6], and electromagnetic scattering features [7,8]. Among classifiers, support vector machines (SVMs) [9], k-nearest neighbor classifiers (k-NNs) [10], and sparse representation classifiers (SRCs) [11] are widely employed. With the advancement of deep learning technology, it has been extensively

applied to the field of SAR, such as in SAR target recognition [12,13], SAR target detection [14,15], and SAR data augmentation [16,17]. Compared to traditional methods, it offers the advantages of end-to-end implementation, the automatic extraction of target features, and high accuracy, effectively enhancing system performance. Early research integrating various deep learning techniques into SAR ATR has achieved significant success [18,19].

Most of the aforementioned studies are based on the closed-set assumption, where the target types in the test set are included in the target types of the training set. This type of task is referred to as closed-set recognition (CSR). However, in real-world applications, the situation is often complex and dynamic, and new target categories may appear during the testing phase that were not present during the training phase. In such cases, the closed-set assumption no longer holds, leading to what is known as open-set recognition (OSR). SAR ATR methods based on the closed-set assumption often force unknown class targets to be classified as one of the known classes, resulting in unquantifiable errors and risks. A comprehensive SAR ATR system should be able to classify known classes while also effectively rejecting unknown classes.

Therefore, many scholars have focused their attention on OSR of SAR targets in recent years. OSR methods in the SAR domain can generally be categorized into discriminative approaches and generative approaches. Discriminative approaches achieve OSR by designing similarity metrics that capitalize on the differences between known and unknown classes. Scherreik et al. [20] improved SVM, innovatively proposing the W-SVM and POS-SVM methods, and successfully applied them to the OSR of SAR targets. Dang et al. [21] utilized extreme value theory (EVT) to construct closed boundary models for known classes, thereby detecting unknown classes. Wang et al. [22] introduced an entropy-aware meta-learning method that innovatively processes at the feature space level, significantly improving the OSR performance of the system. Ma et al. [23] proposed an OSR method based on the joint training of class-specific sub-dictionary learning, utilizing reconstruction error to identify unknown class targets. Due to the powerful feature extraction capabilities of deep learning, it has also been widely used in SAR OSR. For instance, many researchers have deeply explored the OpenMax method, applying it effectively to the OSR of SAR targets [24,25]. Giusti et al. [26] have differentiated between known and unknown classes by leveraging the proportional similarity between different SAR image categories and setting thresholds. Thomson et al. [27] utilized the Regular Polytope Network (RPN) for SAR OSR, which enhances the separation of target features and is beneficial for recognition performance. Inkawhich et al. [28] developed a training method named AdvOE, significantly enhancing model accuracy. Additionally, Ma et al. [29] divided OSR into two tasks: classification and anomaly detection, implementing the task with generative adversarial networks (GANs) for SAR images. Zhou et al. [30] explored SAR OSR under data-constrained conditions, utilizing graph convolutional network (GCN) to construct distributions and explaining unknown classes through discriminative class similarity.

In contrast to discriminative approaches, generative approaches primarily address the OSR problem by generating instances of unknown classes and training them alongside known classes, thereby transforming the open-set issue into a closed-set issue for resolution. Cui et al. [31] combined a counterfactual framework with SAR OSR, employing counterfactual images generated from test samples to determine whether they belong to known or unknown classes. Geng et al. [32] proposed two innovative generative models: spatial clipping generating (SCG) and weighting generating (WG) models, demonstrating their excellent performance in known class categorization and unknown class detection through a series of experiments. Neal et al. [33] proposed a novel framework called OSRCI, which employs GANs to generate samples that closely resemble known classes but do not belong to any of them, effectively converting the OSR problem into a CSR problem with an additional class. Jo et al. [34] conducted further investigations into the application of GAN for the generation of synthetic data, with the objective of augmenting the robustness of classifiers when encountering unknown classes.

The aforementioned studies have achieved satisfactory outcomes, providing valuable references for future work. Summarizing the aforementioned research findings, the SAR OSR primarily confronts several challenges: First, the diversity and complexity of SAR targets. When the perspective, attitude, and angle of SAR targets change, their imagery undergoes significant alterations, which traditional feature extraction methods may struggle to capture. Second, the presence of noise and interference. Due to the unique imaging mechanism of SAR, it contains a substantial amount of speckle noise, which can affect traditional deep learning models, thereby impacting recognition accuracy. Third, the limitation of data volume. High-quality, large-scale datasets are difficult to obtain due to the high cost of SAR imaging, and the scarcity of samples in existing SAR target datasets exacerbates the issue of insufficient data for model training, while deep learning methods often rely on large datasets. Based on these challenges, this paper employs the Capsule Network as the backbone network for feature extraction [35], which offers the following advantages in SAR OSR: First, the Capsule Network can effectively encode the pose information of targets, making the system more adaptable to the pose variations of SAR targets. Second, the structural design and the dynamic routing mechanism of the Capsule Network provide stronger robustness against noise and interference in SAR targets, enabling more stable extraction of effective features. Third, the Capsule Network excels in addressing the issues of small-sample learning and data scarcity, with its structure helping to enhance the model's generalization capability. Based on the above advantages, we use the Capsule Network as the backbone network of the proposed method.

Essentially, the key step in achieving OSR lies in accurately classifying known classes while simultaneously identifying a metric that can effectively differentiate between known and unknown classes. The accurate classification of known classes can be accomplished through the classification capabilities of the Capsule Network, while the discrimination of unknown classes necessitates the identification of an appropriate metric. Kullback–Leibler divergence (KLD) is employed to measure the degree of difference between two distributions. When the similarity between two distributions is high, the KLD value is small; conversely, when the similarity is low, the KLD value is large. For a specific target, the KLD value between it and the targets within its own class is relatively small, while the KLD value with targets from other classes is relatively large. Therefore, when a sample under examination belongs to a known class, it will exhibit a small KLD with a target from that known class. Conversely, when the sample belongs to an unknown class, it will demonstrate a large KLD with all known class targets. This criterion can thus be effectively utilized to reject unknown classes.

In view of the advantages of the KLD metric and the Capsule Network, in this paper, we integrate the Capsule Network with KLD to propose a novel OSR model for SAR targets. Moreover, due to the characteristics of SAR targets, which exhibit small inter-class differences and large intra-class variations, unknown class targets may be proximally located to known class targets, making them difficult to distinguish. Consequently, this paper also designs the loss function, which makes features within the same class more compact and features between different classes more dispersed. This strategy increases the feature separation between known and unknown classes, thereby enlarging the KLD between them and effectively enhancing the performance of the OSR model. The contributions of this paper are as follows.

(1) We propose an end-to-end novel OSR model based on the Capsule Network for SAR targets. This method combines the Capsule Network with KLD, utilizing the Capsule Network for feature extraction and calculating the KLD between testing samples and known classes. It achieves the dual objectives of classifying known classes and effectively rejecting unknown classes, thereby addressing the OSR problem.
(2) A loss function tailored for OSR has been designed, incorporating margin loss, center loss, and KL loss, which reduces intra-class differences and increases inter-class differences in the extracted features. This leads to an enlarged KLD between known and unknown class features, thereby addressing the OSR problem more effectively.

(3) Tested on actual datasets, the recognition rate reaches over 95%. Compared with other traditional methods, the model proposed in the paper has superior performance.

The remainder of this paper is organized as follows. Section 2 provides an overall framework of the proposed OSR method. Section 3 provides a detailed description of the feature extraction network proposed in this paper. In Section 4, the discriminant model based on KLD is described in detail. Section 5 presents the experiments and test results. Finally, the discussion and conclusion are presented in Sections 6 and 7.

2. Overall Framework

In this section, the overall framework of the proposed OSR method for SAR targets is presented. As shown in Figure 1, the method is divided into three steps: 1. feature extraction based on the Capsule Network; 2. calculating the KLD threshold for known classes.; 3. classifying the sample under test by discriminating based on the threshold. Each method is described in detail below.

Step 1: The Capsule Network is employed to extract features from SAR samples belonging to known classes. The necessary features are derived from the output of the Capsule Network and subsequently stored.

Step 2: Initially, the K-means algorithm is utilized to compute the central feature for each known class. Subsequently, for a given known class, the KLD between the features of all samples within this class and the central feature is calculated to determine the KLD threshold specific to that class. Ultimately, a common threshold for all known classes is established based on these individual KLD thresholds.

Step 3: The sample under examination is input into the Capsule Network to extract its corresponding features. The KLD between the features of this test sample and the central feature of each known class is then computed and compared against the common threshold for the known classes. This comparison facilitates the classification of the test sample.

The foregoing provides an overarching summary of the methodology presented in the paper, as depicted in Figure 1. Detailed specifics and computational formulas are elaborated in Sections 3 and 4.

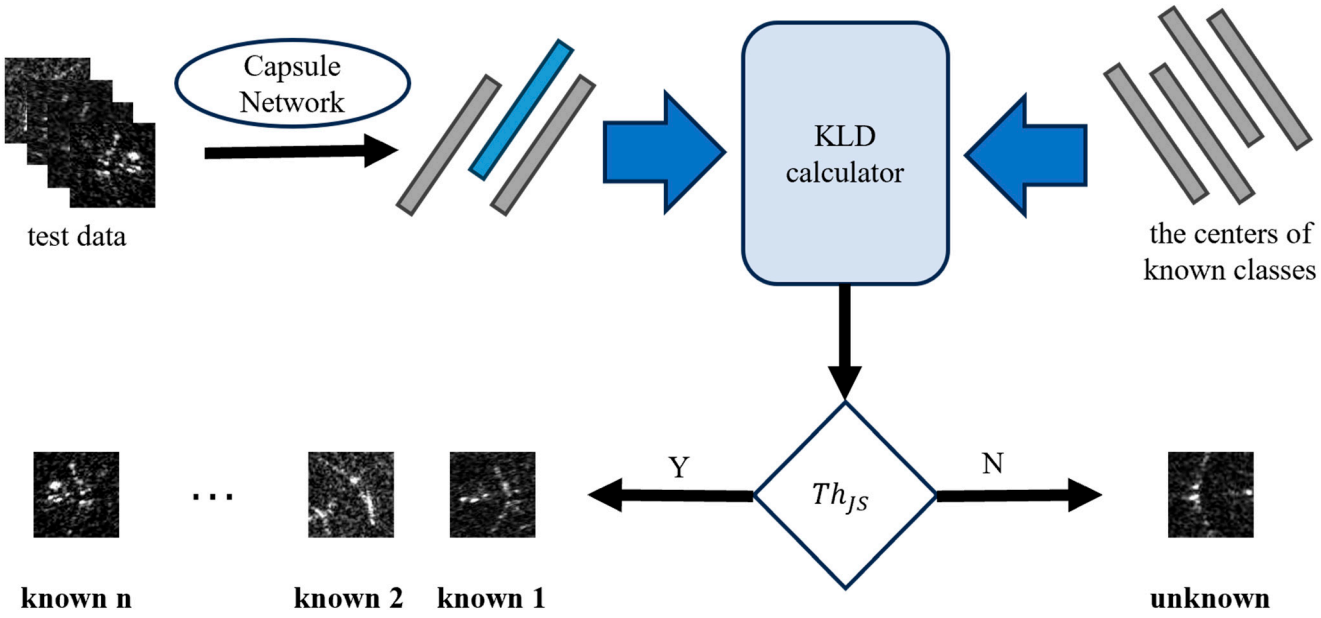

Figure 1. The proposed OSR method.

3. Feature Extraction Based on the Capsule Network

3.1. The Architecture of the Capsule Network

In the method proposed in this paper, the Capsule Network is used for target feature extraction. In the processing of target imagery, traditional neural networks, such as Convolutional Neural Networks (CNNs), employ pooling operations to diminish the dimensions of feature maps. However, this reduction can lead to the disruption of spatial relationships among features, which may render the model insensitive to variations in the target's pose, consequently degrading the model's performance. The Capsule Network offers a solution to this issue by incorporating the concept of "capsules", which preserve a greater amount of information regarding the target's pose through their vectorized outputs. This approach ensures the retention of the target's structural information and spatial relationships to a greater extent, thereby enhancing the model's sensitivity and robustness to changes in the target's pose. Furthermore, CNNs are contingent upon a fixed structure of convolutional kernels, which are incapable of dynamically adjusting to the complex deformations of targets. In contrast, the Capsule Network is equipped with a dynamic routing mechanism that adjusts the connection weights between features, thus providing a more effective adaptation to the intricate deformations of targets. This capability endows the Capsule Network with a superior generalization ability when confronted with targets from various viewpoints. In light of these considerations, we have selected the Capsule Network as the network for feature extraction of targets due to its capacity to maintain spatial hierarchies, its robustness to variations in target pose, and its enhanced adaptability to the complex deformations of targets.

The primary steps involved in data processing by the Capsule Network can be divided into the following stages: the input layer receives data, the convolutional layer extracts image features, the primary capsule layer (denoted as PrimaryCaps layer) encapsulates these features into capsules, and the higher-level capsule layer (denoted as DigitCaps layer) ultimately produces the output. The proposed Capsule Network in this paper comprises two convolutional layers, a PrimaryCaps layer, a DigitCaps layer, and a fully connected layer. Additionally, a multi-scale feature extraction module is introduced. The overall structure of the network and the information pertaining to each component are presented in Table 1.

Table 1. The composition of each layer.

Component Type	Information
Input Layer	-
Convolutional Layer 1	Number of Filters:128 Filter Size: 9×9 Stride: 2
Convolutional Layer 2	Number of Filters: 256 Filter Size: 9×9 Stride: 2
Multi-scale Feature Extraction Module	-
Primary Capsule Layer	Number of Filters: 32 Filter Size: 11×11 Stride: 2
Digital Capsule Layer	-
Fully Connected Layer	-

Given an input image, the processing procedure can be described as follows: Assuming the input image has dimensions of 128×128, it is first fed into the shallow convolutional layers for local feature extraction. Specifically, there are two convolutional layers employed, each with a kernel size of 9×9 and a stride of 2. Following the shallow convolutional

layers, the features are inputted into the multi-scale feature extraction module to extract diverse features. To avoid affecting the original structure of the Capsule Network, the parameters of the multi-scale feature extraction module are designed, thus the dimensions and quantity of features remain unchanged before and after the input. Following this, the PrimaryCaps layer, succeeding the multi-scale feature extraction module, encapsulates the features extracted by the aforementioned convolutional layers into a capsular representation, utilizing 11 × 11 kernels with a stride of 2. Each capsule produced at this stage has a spatial dimension of 8 × 8. The output from the PrimaryCaps layer is transmitted to the DigitCaps layer, where a dynamic routing mechanism iteratively updates the weight matrix, progressively enhancing the degree of alignment between the input and output capsules. The output of the DigitCaps layer comprises several capsules corresponding to the input categories, with the feature dimensionality set at 16. In order to leverage the features extracted by the Capsule Network, a fully connected layer is appended at the end of the network. This layer transforms the output from the DigitCaps layer into a 128-dimensional feature vector. The processing workflow is illustrated in Figure 2.

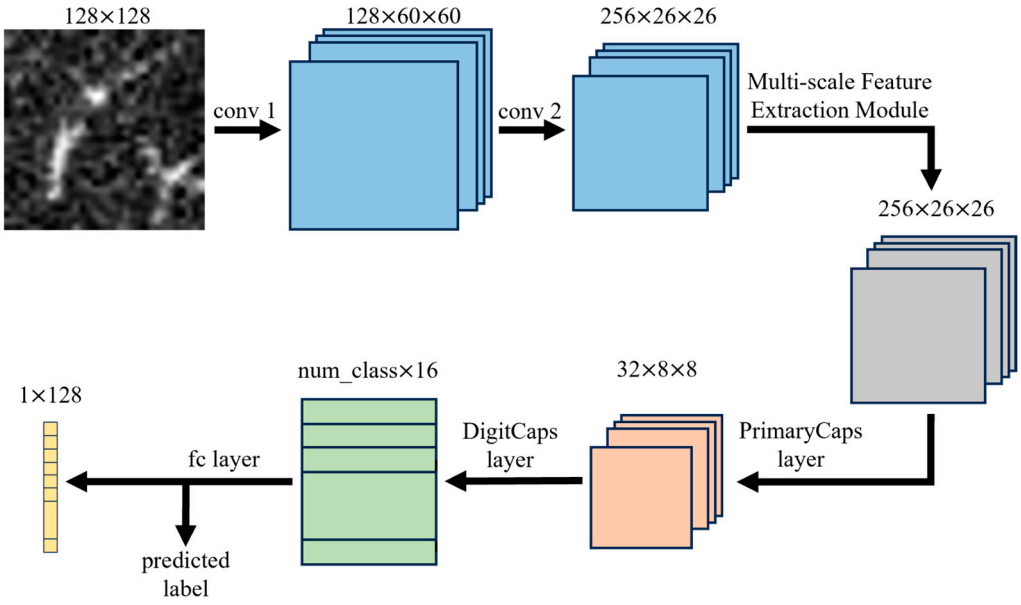

Figure 2. The processing workflow of the proposed network.

3.2. Multi-Scale Feature Extraction Module

The multi-scale feature extraction module is introduced in this article to enhance the robustness of the system. In the context of image processing using deep neural networks, the salient parts of an image may exhibit significant variations in size, posing challenges in selecting the appropriate convolution kernel size for convolution operations. The multi-scale feature extraction module integrates convolution kernels of different sizes and processes them in parallel, enabling the network to simultaneously capture features at various scales. Smaller convolution kernels can capture fine details, while larger kernels can capture more global contextual information. This module allows the network to observe the input data from different perspectives and scales, thus generating richer and more diverse feature representations. Such diverse feature representations contribute to improving the generalization capability and robustness of the model.

The multi-scale feature extraction module is positioned between the shallow convolutional layer and the primary capsule layer in this study, comprising a total of four parallel convolutional modules. The information pertaining to each convolutional module is presented in Table 2.

Table 2. The composition of each parallel layer.

List	Composition	Information
Parallel Module 1	Convolutional Layer 1	Number of Filters: 64 Filter Size: 1 × 1 Stride: 1 Padding: 0
Parallel Module 2	Convolutional Layer 1	Number of Filters: 96 Filter Size: 1 × 1 Stride: 1 Padding: 0
	Convolutional Layer 2	Number of Filters: 128 Filter Size: 3 × 3 Stride: 1 Padding: 1
Parallel Module 3	Convolutional Layer 1	Number of Filters: 64 Filter Size: 1 × 1 Stride: 1 Padding: 0
	Convolutional Layer 2	Number of Filters: 32 Filter Size: 5 × 5 Stride: 1 Padding: 2
Parallel Module 4	Convolutional Layer 1	Number of Filters: 64 Filter Size: 1 × 1 Stride: 1 Padding: 0
	Convolutional Layer 2	Number of Filters: 32 Filter Size: 7 × 7 Stride: 1 Padding: 3

The 1 × 1 and 3 × 3 kernel filters are small convolutional filters used to capture the fine detail features within SAR images, while the 5 × 5 and 7 × 7 kernel filters are larger convolutional filters designed to capture the global features of SAR images. Following the four parallel convolutional modules, the outputs are concatenated along the channel dimension, resulting in a combined feature set. This operation allows the network proposed in the paper to retain detailed features while also extracting more abstract and higher-level information, thereby enhancing the model's representational capacity. It is important to note that this work omits the pooling layers typically found in multi-scale feature extraction networks to prevent the loss of detail information when processing SAR images, which could adversely affect the model's performance. This approach aligns with the concepts underpinning the Capsule Network.

3.3. The Loss Function

For classification tasks, the Capsule Network typically employs the margin loss function as the main loss function to guide the training process. The margin loss is a loss function in deep learning, typically employed to enhance the model's ability to distinguish boundaries between different categories. It serves to augment inter-class variance during training, thereby facilitating better differentiation between categories. The primary role of margin loss is to penalize the distance between samples within the same category while encouraging the model to increase the distance between samples from different categories. Consequently, during training, the model focuses more on inter-class discrimination, thereby increasing inter-class variance and improving the model's classification performance and generalization ability. Specifically, the design of margin loss often incorporates a

margin parameter, which governs the difference between distances among samples within the same category and those from different categories. By adjusting this margin parameter, samples within the same category can be made more compact, while those from different categories can be more dispersed, thereby enhancing inter-class discrimination.

Specifically pertaining to the Capsule Network proposed in this paper, the calculation process of the margin loss can be described as follows: The loss for an input sample is computed for each class individually and then aggregated, ensuring that the length of the capsule output vector for the correct category exceeds a threshold m^+, while the length for incorrect category capsules remains below a threshold m^-. This approach is intended to bolster the confidence of the model in the correct categories whilst diminishing assurance in erroneous ones. Suppose for a given sample x, the Margin Loss L_k for category k can be articulated as:

$$L_k = T_k \max(0, m^+ - |v_k|)^2 + \lambda(1 - T_k)\max(0, |v_k| - m^-)^2 \quad (1)$$

where T_k serves as an indicator, taking the value of 1 if the category k is the correct category, and 0 otherwise; $|v_k|$ denotes the length of the capsule output vector for the sample x corresponding to the category k; and λ represents a scaling parameter utilized to balance the contribution of positive and negative class losses, thereby preventing a disproportionately small contribution of positive class loss to the total loss during the early stages of training.

Assuming the training samples encompass K categories in total, the margin loss for the sample x can be expressed as:

$$L_{margin} = \sum_{k=1}^{K} L_k \quad (2)$$

To reduce intra-class variation in the extracted features, the center loss is introduced. The center loss measures the distance between the sample features and the centroid of their respective classes. By incorporating the center loss, features within the same class can be made more compact. Given the sample x, the center loss can be represented as:

$$L_{center} = |v_x - c_i|_2^2 \quad (3)$$

where c_i is the center of the category i to which sample x belongs; v_x denotes the feature of the sample x extracted by the Capsule Network; and $|\cdot|_2$ represents the Euclidean norm (also known as the L2 norm).

Furthermore, during the training phase, a KLD constraint term is introduced to ensure that the KLD within the same class is minimized and the KLD between different classes is maximized. This is aimed at increasing the inter-class KLD differences while reducing the intra-class KLD differences. Specifically, a regularization term is added to the original loss function, referred to as KL loss. The KL loss penalizes excessive KLD between samples of the same class and rewards increased KLD between samples of different classes.

Due to the requirement of calculating the KLD between different samples, it is not feasible to compute the loss for an individual sample independently. Consequently, a batch is considered as a single processing unit, where the KLDs between all samples within the batch are computed simultaneously. The average of these divergences is then taken to obtain the KL loss for the batch. The specific computational procedure is as follows:

Within a given batch, all sample pairs are exhaustively traversed, with the count of intra-class sample pairs denoted by num_{same} and that of inter-class sample pairs denoted by $num_{different}$.The KLD across all sample pairs is initially computed. The aggregate KLD

for the intra-class and inter-class sample pairs are respectively calculated, and then the mean value is computed for both sets:

$$L_{KL\_same\_ave} = \sum_{i=j} KL(v_i \parallel v_j)/num_{same} \tag{4}$$

$$L_{KL\_different\_ave} = \sum_{i\neq j} KL(v_i \parallel v_j)/num_{different} \tag{5}$$

where $KL(v_i \parallel v_j)$ represents the KLD of v_i from v_j.The KL loss can be expressed as follows:

$$L_{KL} = \lambda(L_{KL\_same\_ave} - L_{KL\_different\_ave}) \tag{6}$$

where λ is a balancing factor that controls the influence of the KLD constraint term.

The final loss function for a batch can be expressed as:

$$L = \alpha \sum_{batch} L_{margin} + \beta \sum_{batch} L_{center} + \gamma L_{KL} \tag{7}$$

where α, β, and γ are the coefficients corresponding to the three loss functions, respectively; and $\sum_{batch}$ denotes the sum of the samples in a batch.

With the designed loss function, the model is capable of maintaining high-precision classification for known classes while simultaneously achieving reduced intra-class variance and increased inter-class variance among the extracted features of known classes. Consequently, this facilitates the distinction of unknown classes from known classes with greater ease.

4. The Discriminant Model Based on KLD

Following the aforementioned training process, the model is successfully obtained, along with the preserved features. The subsequent steps involve contemplating how to effectively utilize the acquired features for discrimination.

4.1. The Concept of KLD

KLD, also known as relative entropy, is utilized in the field of target recognition as a tool for comparing different probability distributions, enabling the quantification of differences between two probability distributions. For the same random variable z, let P and Q be two distinct probability distributions of z defined on the same probability space, then the KLD of P from Q is defined as:

$$\mathrm{KL}(P \parallel Q) = \sum_{j=1}^{n} p(z_j) \log\left(\frac{p(z_j)}{q(z_j)}\right) \tag{8}$$

where n denotes the number of all possible events, and $p(z_j)$ and $q(z_j)$ denote the probability of the event z_j occurring in the corresponding distribution. In the above equation, if P approaches Q, the value is small; conversely, if P diverges from Q, then this value is large. Note that the KLD equals 0 only when P and Q are identical.

Because the logarithmic function is a convex function, the KLD is non-negative:

$$\mathrm{KL}(P \parallel Q) \geq 0 \tag{9}$$

In addition, the KLD is also asymmetric:

$$\mathrm{KL}(P \parallel Q) \neq \mathrm{KL}(Q \parallel P) \tag{10}$$

Based on the aforementioned formulas, it is evident that the KLD can be utilized to measure the dissimilarity between two distributions. This characteristic can be employed for the classification of different targets. Specifically, when the input test sample belongs to

one of the known categories, the KLD between the features obtained through the Capsule Network and the central features of the corresponding known class is small. Conversely, when the test sample x belongs to an unknown class, the KLD is large. Following this principle, by calculating the KLD between the test sample and the centers of known classes and subsequently setting a threshold for the KLD, the identification of unknown classes can be achieved.

4.2. The Calculation of the Threshold for KLD

Initially, the threshold for KLD is determined. The training set is inputted into the Capsule Network to extract features. which are subsequently normalized into probability distributions through the softmax function.

The specific steps are as follows: Assuming a given one-dimensional feature vector v of length n, first, compute the exponential function for each element of the vector. This results in a new vector, $v\prime$, where the ith element is calculated as:

$$v'_i = e^{v_i} \tag{11}$$

where $i = 1, 2, \ldots, n$.

Subsequently, sum all computed exponentials and derive the softmax value for each element by dividing its respective exponential value by the computed sum of all exponentials:

$$\text{softmax}(v_i) = \frac{e^{v_i}}{S} = \frac{e^{v_i}}{\sum_{j=1}^{n} e^{v_j}} \tag{12}$$

The resulting vector $\mathbf{p} = [p_1, p_2, \ldots, p_n]$ represents a probability distribution, where the element p_i denotes the probability $\text{softmax}(v_i)$, corresponding to the respective element in the original feature vector.

The formulations above ensure that all elements are within the range $(0, 1)$ and their sum equals 1, which converts the feature vector into a probability distribution. Following the aforementioned steps, the KLD can be calculated.

For each sample class, the KLD values between all samples in that class and the sample centroid are calculated, with the maximum value selected as the threshold for that class:

$$Th_i = \max_{j=1,\ldots,n} KL_j \tag{13}$$

where i represents the sample class, and KL_j denotes samples in class i.

The final KLD threshold is then determined as the maximum value among all class thresholds, which can be expressed as:

$$Th_{KL} = \lambda \cdot \max_{i=1,\ldots,K} Th_i \tag{14}$$

where λ is the threshold coefficient.

4.3. The Test Procedure

For the test sample, first input it into the Capsule Network for feature extraction and conversion into probability distribution. Then, calculate the KLD values between the distribution and the centers of known categories. Compare the minimum KLD value with the threshold determined earlier. If the KLD value is greater than the threshold, classify it as an unknown target. If it is smaller than the threshold, classify it as a known target and use the predicted category from the Capsule Network as the final prediction result.

5. Experiments and Results

In this section, multiple experiments were conducted using the SAR-ACD dataset [36] to validate the effectiveness of the proposed method. Firstly, an introduction to the SAR-ACD dataset utilized in the experiments was provided. Secondly, by setting three fixed

known classes, experiments were conducted with varying numbers of unknown classes (1, 2, and 3) to validate the effectiveness of the model. Finally, to assess the advancement and robustness of the model, experiments were performed under three scenarios, where the number of known classes ranged from 5 to 3 while the number of unknown classes ranged from 1 to 3, and the results were compared with several methods.

5.1. Experiment Setup

5.1.1. Dataset

In this paper, we conduct experiments using the SAR-ACD dataset, a challenging SAR aircraft category dataset characterized by high scene complexity. This dataset has a resolution of 1m and includes six types of aircraft targets: A220, A320/321, A330, ARJ21, Boeing737, and Boeing787, comprising a total of 3032 images with approximately 500 images for each category. The optical images and corresponding SAR images of six classes of targets in the SAR-ACD dataset are shown in Figure 3.

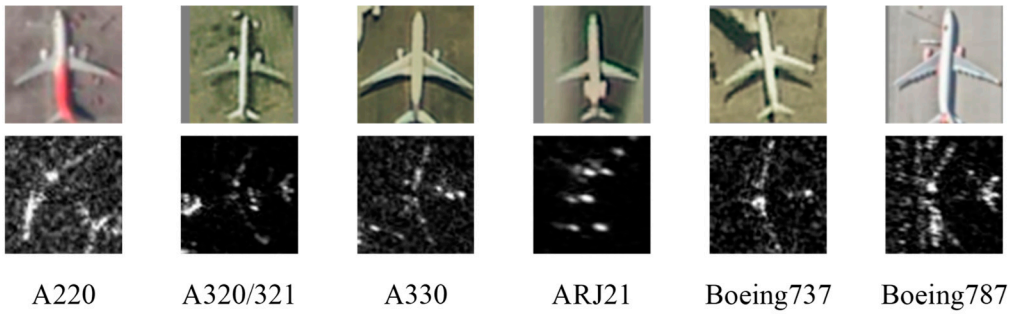

Figure 3. Optical images and corresponding SAR images of six classes of targets in the SAR-ACD dataset.

To facilitate training and testing, images of each type were randomly divided into training and testing sets at a ratio of 6:4. The number of targets in each category and the number of train and test sets after division are shown in Table 3.

Table 3. The sample number of each target category.

class	A220	A320/321	A330	ARJ21	Boeing737	Boeing787
all	464	510	512	514	528	504
train	278	306	307	308	316	302
test	186	204	205	206	212	202

Since the dataset images are of different sizes, they are uniformly resized to 128 × 128 before inputting into the network for the convenience of training and testing.

5.1.2. Metrics

To measure the openness of the OSR issue, this paper introduces the metric Openness [37], which is calculated using the following formula:

$$Openness = 1 - \sqrt{\frac{2 \times |C_{TR}|}{|C_{TR}| + |C_{TE}|}} \tag{15}$$

where $|C_{TR}|$ represents the number of target classes in the training set, and $|C_{TE}|$ denotes the number of target classes in the test set. The larger the value of openness, the greater the proportion of unknown classes in the OSR task.

To qualitatively assess the performance of the proposed OSR method in the SAR ATR task, four evaluation metrics are employed: accuracy, precision, recall, and F1-score. The metrics are calculated as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$precision = \frac{TP}{TP + FP} \tag{17}$$

$$recall = \frac{TP}{TP + FN} \tag{18}$$

$$F1\text{-}score = 2 \times \frac{precision \times recall}{precision + recall} \tag{19}$$

In the aforementioned equations, TP stands for True Positives, representing the number of instances correctly predicted as positive by the model. TN denotes True Negatives, which corresponds to the number of instances correctly predicted as negative by the model. FP signifies False Positives, referring to the number of instances incorrectly predicted as positive by the model. Lastly, FN represents False Negatives, indicating the number of instances incorrectly predicted as negative by the model. Additionally, the confusion matrix is utilized to render the results more intuitive.

The experiments were implemented on a laptop with an Intel Core i5-13500H CPU, a NVIDIA GeForce RTX 3050 4GB Laptop GPU, and 16 GB RAM on the Windows 11 system. In the experiment, the learning rate is set to 0.00003, the batch size is 16, the epoch is 200, and the number of iterations of the dynamic routing mechanism is 4.

5.2. Three-Class Open-Set Recognition

To evaluate the effectiveness of the model proposed in this paper, we conducted experiments under three different scenarios. The known categories (denoted as kn) were fixed at three types: A320/321, ARJ21, and Boeing737. The unknown categories (denoted as un) were set as one category (A330), two categories (A330, Boeing787), and three categories (A220, A330, Boeing787), respectively.

During the training phase, three known categories are input into the Capsule Network, resulting in a trained model along with the corresponding features for each of the three known categories. For each category of features, the KLD between the feature of each sample and the centroid of the sample features for that class is computed. The greatest KLD value across all samples is established as the KLD threshold for that class. Subsequently, the largest threshold among the three category thresholds is selected as the threshold for known classes.

In the test phase, the sample is input into the preserved model, which then outputs the predicted category along with its associated features. The KLD between the features of the test sample and the centroid of the features for each known class is calculated. If the divergence value exceeds the threshold, the sample is determined to be from an unknown class. If not, the predicted category outputted by the network is used as the prediction label. The results obtained are presented in Table 4. The confusion matrices of the three scenarios are in Figure 4.

Table 4. The results in three scenarios.

Scenario	Openness	Accuracy_kn	Accuracy_un	Accuracy_all	Precision	Recall	F1-score
3kn + 1un	0.0742	98.88	96.08	98.19	98.18	98.18	98.18
3kn + 2un	0.1340	98.39	95.07	97.08	97.06	97.57	97.28
3kn + 3un	0.1835	98.88	97.13	98.02	97.69	98.45	98.06

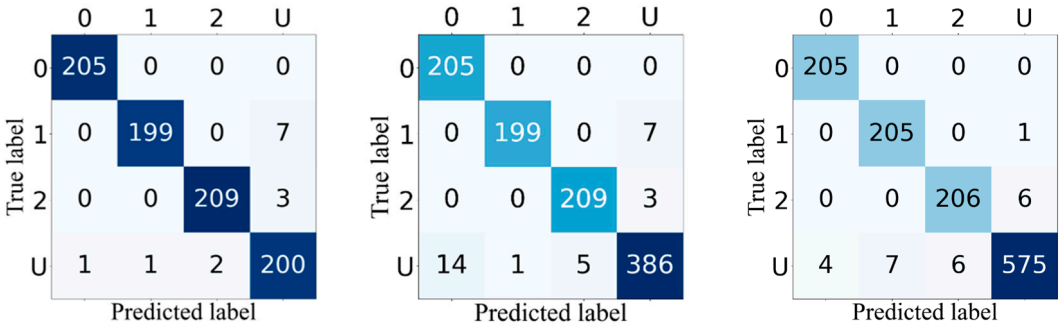

Figure 4. The confusion matrices in three scenarios. (**a**) Three known classes and one unknown class. (**b**) Three known classes and two unknown classes. (**c**) Three known classes and three unknown classes.

The results in Table 3 and Figure 4 indicate that the method proposed in this paper is capable of obtaining a classification accuracy of over 98% for known categories, while also achieving a high recognition rate for unknown categories, with an F1-score maintained above 95%. This demonstrates the effectiveness of the proposed method in accomplishing the OSR task.

5.3. Multiclass Open-Set Recognition

To validate the robustness of the proposed method, three scenarios were established with varying compositions of datasets, where the known categories were set as 5, 4, and 3 classes, respectively, and the corresponding unknown categories were set as 1, 2, and 3 classes. To demonstrate the advancement of this method, it was compared with several other methods, including MGPL [38], OpenMax [39], CROSR [40], CAVECapOSR [41], and CAC [42]. The results obtained are presented in Table 5.

From the results, it can be observed that the method proposed in this paper exhibits significant improvements compared to other methods in terms of accuracy, precision, recall, and F1-score. Especially when unknown targets increase, our method still maintains a high level, while the performances of other methods drop dramatically.

Table 5. The results comparison under three different scenarios.

Scenario	Openness	Method	Accuracy	Precision	Recall	F1-Score
5kn + 1un (A330)	0.0465	MGPL	73.83	83.56	73.91	76.58
		OpenMax	86.75	86.21	86.59	86.30
		CROSR	86.83	86.30	86.68	86.39
		CVAECapOSR	82.30	80.32	84.53	82.15
		CAC	83.95	81.86	86.15	83.95
		Ours	97.37	97.49	97.23	97.29
4kn + 2un (A330, Boeing787)	0.1056	MGPL	73.09	80.99	73.44	75.73
		OpenMax	76.87	78.75	83.81	79.01
		CROSR	77.45	79.14	84.00	79.57
		CVAECapOSR	79.01	76.82	81.19	78.90
		CAC	80.49	78.34	82.71	80.47
		Ours	95.39	96.35	95.02	95.60
3kn + 3un (A220, A330, Boeing787)	0.1835	MGPL	36.68	36.01	54.81	42.15
		OpenMax	70.37	72.57	83.63	72.85
		CROSR	74.57	75.07	85.70	76.59
		CVAECapOSR	74.07	72.04	76.19	74.14
		CAC	79.12	77.05	81.48	79.11
		Ours	98.02	97.69	98.45	98.06

In addition, we compare the above methods with the method in this paper in terms of computational complexity in the third scenario. Computational complexity can be divided into time complexity and space complexity. For the convenience of quantitative analysis, we use the number of floating-point operations (FLOPs) as the indicator of time complexity and use the number of parameters and the size of parameters as the indicators of space complexity [43]. The obtained results are shown in Table 6.

Table 6. The computational complexity of the different methods.

Method	FLOPs	Total Params	Params Size
MGPL	211.797 G	27,042,369	108.3 MB
OpenMax	35.328 G	61,100,840	233 MB
CROSR	32.962 G	35,734,436	136 MB
CVAECapOSR	273.771 G	29,289,473	155 MB
CAC	84.784 G	2,010,502	4.1 MB
Ours	18.713 G	10,601,344	44.7 MB

From the analysis of the results, it can be obtained that our method has the smallest FLOPs, that is, the algorithm has a lower time complexity. The reason for the analysis is that compared with the deep networks in other methods, the number of network layers of the algorithm in this paper is relatively small, so the FLOPs are smaller. However, in terms of the number of parameters and the size of parameters, although the number of network layers of this method is shallower, the number of parameters and the size of parameters are 10 times that of the CAC method and are basically on the same order of magnitude as the other methods. This is mainly because the dynamic routing mechanism needs to be iterated, and this mechanism can effectively improve the performance of the model.

6. Discussion

In this section, we conducted three sets of ablation experiments, where settings were consistent with the third scenario (three known classes and three unknown classes) in Section 5. Additionally, to visually inspect the experimental outcomes more intuitively, the t-SNE visualization method was introduced, plotting the features of all test samples on the same graph.

6.1. Ablation Experiments of Each Module

To analyze the impact of each module on the proposed model, we performed ablation experiments on the Capsule Network module (denoted as Module A), the multi-scale feature extraction module (denoted as Module B), the KLD discriminative module (denoted as Module C), and separately. For the Capsule Network module, a comparison was made with CNN; for the KLD discriminative module, a comparison was made with Softmax; for the multi-scale feature extraction module, the differences between its presence and absence were analyzed. The confusion matrices are in Figure 5. The confusion matrix of the full version is the same as Figure 4c. The results obtained are presented in Table 7.

Table 7. The results of the four kinds of configuration.

Configuration	Accuracy_kn	Accuracy_un	Accuracy_all	Precision	Recall	F1-Score
A + B + C	98.88	97.13	98.02	97.69	98.45	98.06
B + C	90.85	85.47	88.23	88.23	89.50	88.56
A + C	99.04	95.27	97.20	96.50	98.11	97.27
A + B	96.63	92.06	94.40	93.70	95.47	94.52

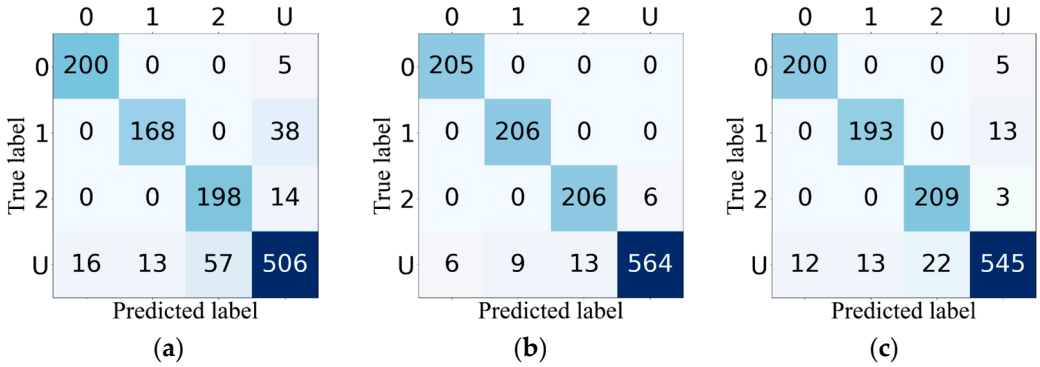

Figure 5. The confusion matrices of different kinds of configuration. (**a**) The configuration without the Capsule Network module; (**b**) The configuration without the multi-scale feature extraction module; (**c**) The configuration without the KLD discriminative module.

Additionally, to provide a more intuitive analysis of the feature extraction effectiveness of the Capsule Network compared to CNN, the t-SNE was employed to compare the features extracted by both networks. The results are illustrated in Figure 6, where the red color represents the unknown class target.

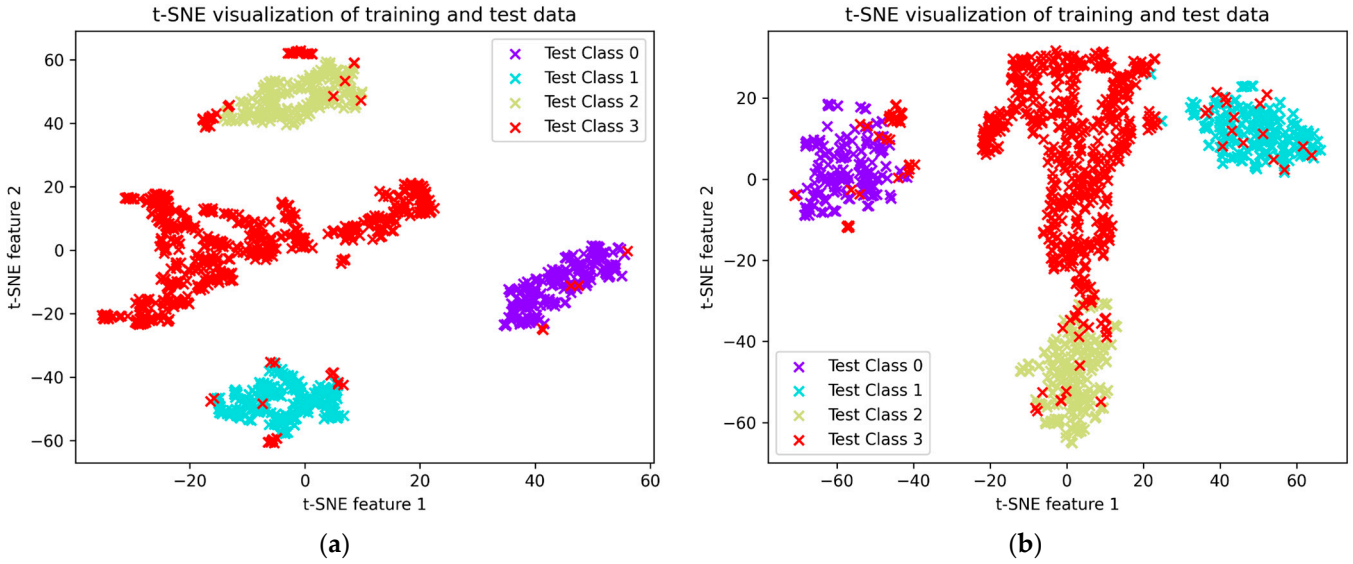

Figure 6. The t-SNE visualization results. (**a**) The feature extracted by the Capsule Network. (**b**) The feature extracted by CNN.

Analysis of the results presented in Table 7 and Figure 5 indicates that the Capsule Network module, the multi-scale feature extraction module, and the KLD discriminative module enhance the performance of the system.

The results in Figure 6 indicate that the features extracted by the Capsule Network exhibit greater inter-class differences compared to those extracted by CNN, which is beneficial for OSR. The implementation of the multi-scale feature extraction module resulted in a slight reduction in the accuracy of the known class; however, it led to improvements across the overall metrics of accuracy, precision, recall, and F1-score.

6.2. Ablation Experiments of the Loss Function

To assess the influence of the center loss and the KL loss in the loss function on the performance of the model, we conducted ablation experiments on the components of the loss function. The margin loss was used as the primary loss, with the coefficient (α) set to 1, the center loss coefficient (β) set to 0.05, and the KL loss coefficient (γ) set to 0.1. The

configuration information is presented in Table 8, where the symbol "√" indicates that the loss exists. The results obtained are presented in Table 9.

Table 8. The configuration information of the four methods.

Method	Margin Loss	Center Loss	KL Loss
a	√	√	√
b	√		√
c	√	√	
d	√		

Table 9. The results of the four methods.

Method	Accuracy_kn	Accuracy_un	Accuracy_all	Precision	Recall	F1-Score
a	98.88	97.13	98.02	97.69	98.45	98.06
b	95.83	92.74	94.32	93.83	95.08	94.42
c	97.91	95.44	96.71	96.31	97.30	96.77
d	88.44	85.14	86.83	86.68	87.67	87.08

The t-SNE visualization results are shown in Figure 7, where the red color represents the unknown class target. The t-SNE visualization result of method a is the same as Figure 6a.

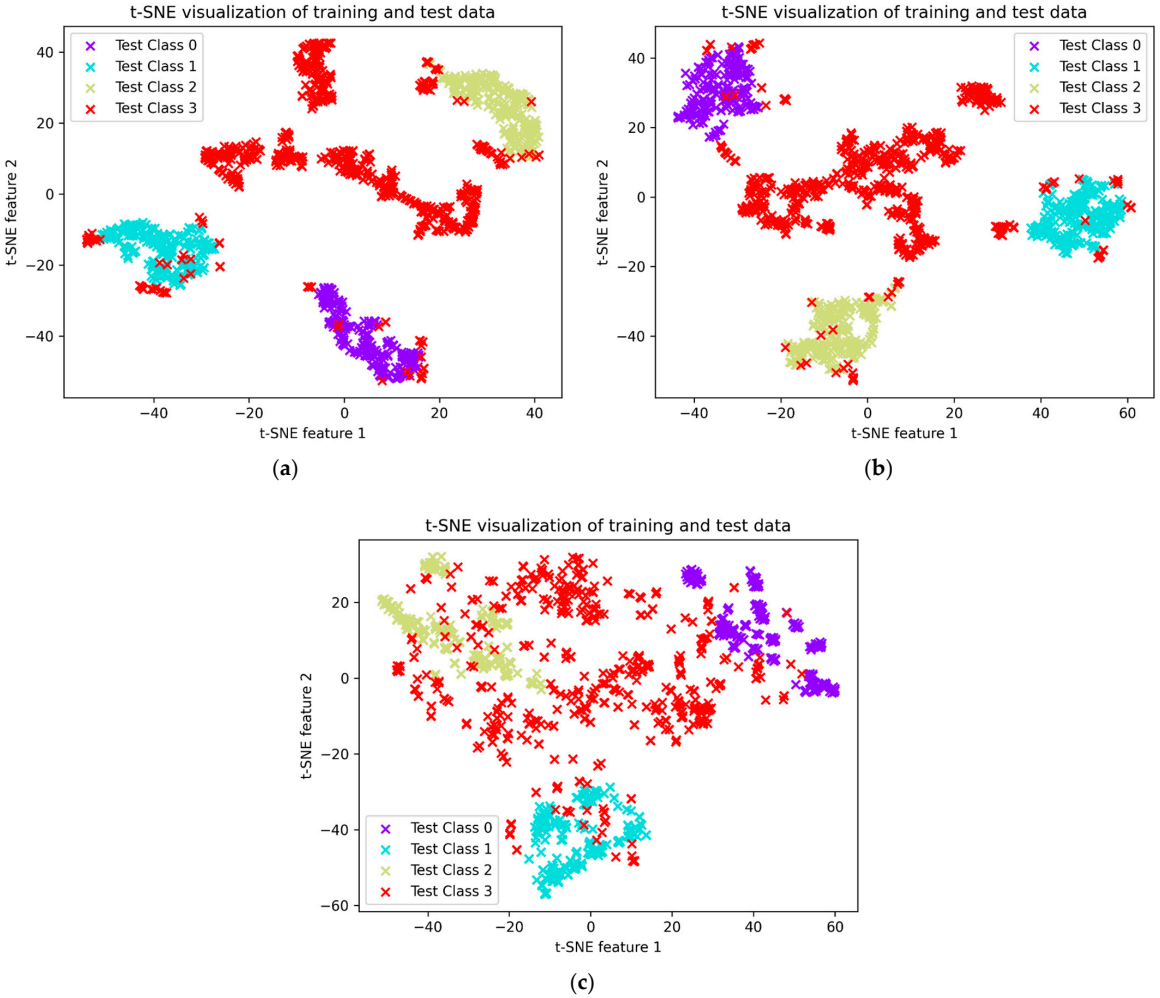

Figure 7. The t-SNE visualization results. (**a**) The loss function without the center loss. (**b**) The loss function without the KL loss. (**c**) The loss function without the center loss and the KL loss.

Analysis of the results presented in Table 8 and Figure 7 indicates that the incorporation of center loss and KL loss significantly enhances the performance of the system. The center loss promotes closer clustering of features within the same class, resulting in a reduced spatial occupation of feature space by known class features, thereby providing additional space for unknown class features. The KL loss ensures that the KLD within classes is minimized, while the KLD between classes is maximized, which is more conducive to threshold determination and selection. Therefore, the method proposed in this study, which integrates the margin loss, the center loss, and the KL loss, effectively improves the performance of the OSR model.

6.3. Ablation Experiments of the Important Hyperparameters

6.3.1. The Coefficients of the Loss Functions

To illustrate the impact of the weight coefficients of the loss function, in addition to the main loss, we conducted ablation experiments on the center loss and KL loss functions, with coefficients set to [0.01, 0.05, 0.1, 0.5, 1], respectively. Since the F1-score better reflects the overall performance of the model, we analyzed the relationship between the coefficients of the two and the F1-score, and the results are shown in Figure 8.

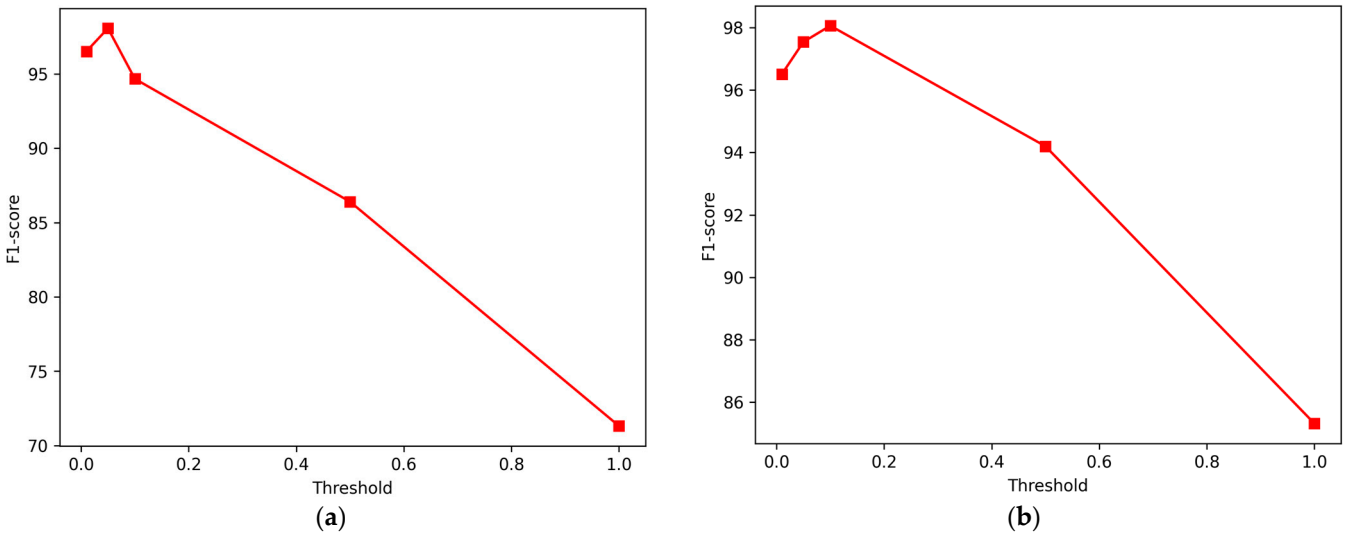

Figure 8. The F1-score results. (**a**) The results of different coefficients of the center loss. (**b**) The results of different coefficients of the KL loss.

From the results, it can be concluded that the optimal coefficient of center loss is 0.05, and the optimal coefficient of KL loss is 0.1. We did not further refine the value of the coefficient because it can bring little improvement. Additionally, the observations reveal that when the coefficients for the components are excessively large, there is a significant degradation in model performance. This decline can be attributed to the imbalance introduced during the training process, which causes the learning of the primary loss to be neglected, thereby resulting in a deterioration of the model's overall performance.

6.3.2. The KL Threshold

The determination of the KLD threshold is related to the overall performance of the model. Therefore, in this scenario, an ablation experiment was conducted on this threshold, which was set to [0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3], and the F1-score was also used as the evaluation criterion. The results are shown in Figure 9.

From the results, it can be concluded that the optimal threshold of the KLD is 1.1. It should be noted that the threshold values in different scenarios are not the same, and fine-tuning is needed to achieve optimal performance. This is also the focus of our next research on how to determine the optimal threshold independently.

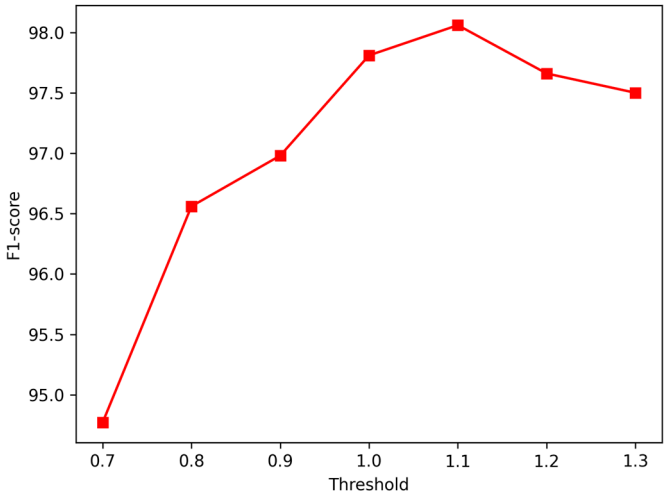

Figure 9. The F1-score results of different KLD thresholds.

7. Conclusions

The SAR ATR task often encounters the OSR issue, where it is crucial to successfully distinguish known classes from unknown classes in the absence of information on the unknown classes. This paper addresses this challenge by exploring the feature level, leveraging the significant differences in feature KLD between different classes. A discriminative model for OSR is constructed using the Capsule Network and KLD. Additionally, a multi-scale feature extraction module is introduced to the original network, effectively enhancing the diversity of extracted features. Furthermore, the loss function is designed. By incorporating the center loss, the features of known classes are more clustered, which facilitates the separation of known and unknown class features, thereby improving recognition performance. The introduction of KL loss reduces the inter-class KLD and increases the intra-class KLD, which is more conducive to threshold selection and discrimination. Experiments conducted on the SAR-ACD dataset have verified the effectiveness and robustness of this method. Compared to other state-of-the-art methods, the proposed evaluation metrics demonstrate optimal performance. Although the computational complexity shows a slight increase compared to methods, it remains within an acceptable range. Ablation experiments indicate that each module effectively enhances the overall performance of the model.

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

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62401581.

Data Availability Statement: This paper adopts the public SAR-ACD dataset, which is available for all researchers on the official website.

Acknowledgments: The authors are thankful to all the participants who dedicated their time and effort to complete this study.

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

References

1. Saghri, J.; Guilas, C. Hausdorff Probabilistic Feature Analysis in SAR Image Recognition. In Proceedings of the Applications of Digital Image Processing XXVIII, San Diego, CA, USA, 31 July–4 August 2005.
2. Zhang, X.; Dong, G.; Xiong, B.; Kuang, G. Refined segmentation of ship target in SAR images based on GVF snake with elliptical constraint. *Remote Sens. Lett.* **2017**, *8*, 791–800. [CrossRef]
3. Mishra, A. Validation of PCA and LDA for SAR ATR. In Proceedings of the TENCON 2008, Hyderabad, India, 19–21 November 2008.
4. Huan, R.; Pan, Y.; Mao, K. SAR image target recognition based on NMF feature extraction and Bayesian decision fusion. In Proceedings of the 2010 Second IITA International Conference on Geoscience and Remote Sensing, Qingdao, China, 28–31 August 2010.
5. Deng, Z.; Jin, J.; Su, J.; Yang, X. Sparse representation of natural image based on Contourlet overcomplete dictionary. In Proceedings of the 2015 International Industrial Informatics and Computer Engineering Conference (IIICEC 2015), Xi'an, China, 10–11 January 2015.
6. Zhou, Z.; Cao, Z.; Pi, Y. Subdictionary-Based Joint Sparse Representation for SAR Target Recognition Using Multilevel Reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6877–6887. [CrossRef]
7. Ding, B.; Wen, G.; Huang, X.; Ma, C.; Yang, X. Target Recognition in Synthetic Aperture Radar Images via Matching of Attributed Scattering Centers. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2017**, *10*, 3334–3347. [CrossRef]
8. Ding, B.; Wen, G. A Region Matching Approach based on 3-D Scattering Center Model with Application to SAR Target Recognition. *IEEE Sens. J.* **2018**, *18*, 4623–4632. [CrossRef]
9. Anagnostopoulos, G.C. SVM-based target recognition from synthetic aperture radar images using target region outline descriptors. *Nonlinear Anal. Theory Methods Appl.* **2009**, *71*, e2934–e2939. [CrossRef]
10. Tian, S.; Yin, K.; Wang, C.; Zhang, H. An SAR ATR Method Based on Scattering Centre Feature and Bipartite Graph Matching. *IETE Tech. Rev.* **2015**, *32*, 364–375. [CrossRef]
11. Zhang, H.; Nasrabadi, N.; Zhang, Y.; Huang, T. Multi-View Automatic Target Recognition using Joint Sparse Representation. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2481–2497. [CrossRef]
12. Huang, Z.; Wu, C.; Yao, X.; Zhao, Z.; Huang, X.; Han, J. Physics inspired hybrid attention for SAR target recognition. *ISPRS J. Photogramm. Remote Sens.* **2024**, *207*, 164–174. [CrossRef]
13. Sun, Z.; Xue, L.; Xu, Y. Recognition of SAR Target Based on multilayer Auto-Encoder And SNN. *Int. J. Innov. Comput. Inf. Control* **2013**, *9*, 4331–4341.
14. Guo, Y.; Chen, S.; Zhan, R.; Wang, W.; Zhang, J. LMSD-YOLO: A lightweight YOLO algorithm for multi-scale SAR ship detection. *Remote Sens.* **2022**, *14*, 4801. [CrossRef]
15. Chen, S.; Zhan, R.; Wang, W.; Zhang, J. Learning Slimming SAR Ship Object Detector Through Network Pruning and Knowledge Distillation. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2020**, *14*, 1267–1282. [CrossRef]
16. Li, J.; Peng, C. Weighted residual network for SAR automatic target recognition with data augmentation. *Front. Neurorob.* **2023**, *17*, 1298653. [CrossRef]
17. Aghababaei, H.; Ferraioli, G.; Stein, A.; Vitale, S. Deep Learning Based Polarimetric Data Augmentation: Dual2Full-pol Extension. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5213016. [CrossRef]
18. Chen, S.; Wang, H.; Xu, F.; Jin, Y. Target Classification Using the Deep Convolutional Networks for SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4806–4817. [CrossRef]
19. Zhang, X.; Feng, S.; Zhao, C.; Sun, Z.; Zhang, S.; Ji, K. MGSFA-Net: Multiscale Global Scattering Feature Association Network for SAR Ship Target Recognition. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2024**, *17*, 4611–4625. [CrossRef]
20. Matthew, S.; Brian, R. Multi-class open set recognition for SAR imagery. In Proceedings of the Automatic Target Recognition XXVI, Baltimore, MD, USA, 18–19 April 2016.
21. Dang, S.; Cao, Z.; Cui, Z.; Pi, Y. Open Set SAR Target Recognition Using Class Boundary Extracting. In Proceedings of the 2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), Xiamen, China, 26–29 November 2019.
22. Wang, C.; Luo, S.; Pei, J.; Liu, X.; Huang, Y.; Zhang, Y.; Yang, J. An Entropy-Awareness Meta-Learning Method for SAR Open-Set ATR. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 4005105. [CrossRef]
23. Ma, X.; Ji, K.; Zhang, L.; Feng, S.; Xiong, B.; Kuang, G. SAR Target Open-Set Recognition Based on Joint Training of Class-Specific Sub-Dictionary Learning. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 3500805. [CrossRef]
24. Oveis, A.H.; Giusti, E.; Ghio, S.; Martorella, M. Open Set Recognition in SAR Images Using the Openmax Approach: Challenges and Extension to Boost the Accuracy and Robustness. In Proceedings of the 14th European Conference on Synthetic Aperture Radar (EUSAR 2022), Leipzig, Germany, 25–27 July 2022.
25. Giusti, E.; Ghio, S.; Oveis, A.H.; Martorella, M. Open Set Recognition in Synthetic Aperture Radar Using the Openmax Classifier. In Proceedings of the 2022 IEEE Radar Conference (RadarConf22), New York, NY, USA, 21–25 March 2022.
26. Giusti, E.; Ghio, S.; Martorella, A.H.O.A. Proportional Similarity-Based Openmax Classifier for Open Set Recognition in SAR Images. *Remote Sens.* **2022**, *14*, 4665. [CrossRef]
27. Thomson, B.; Scherreik, M. Deep Open World SAR Target Recognition with Regular Polytope Networks. In Proceedings of the 2023 IEEE Radar Conference (RadarConf23), San Antonio, TX, USA, 1–5 May 2023.

28. Inkawhich, N.; Davis, E.; Inkawhich, M.J.; Majumder, U.; Chen, Y. Training SAR-ATR Models for Reliable Operation in Open-World Environments. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2021**, *14*, 3954–3966. [CrossRef]
29. Ma, X.; Ji, K.; Zhang, L.; Feng, S.; Xiong, B.; Kuang, G. An Open Set Recognition Method for SAR Targets Based on Multitask Learning. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4014005. [CrossRef]
30. Zhou, X.; Zhang, Y.; Liu, D.; Wei, Q. SAR Target Recognition with Limited Training Samples in Open Set Conditions. *Sensors* **2023**, *23*, 1668. [CrossRef]
31. Cui, Y.; Kuang, G.; Tang, T.; Zhou, X. SAR Open Set Recognition Based on Counterfactual Framework. In Proceedings of the 2022 Photonics and Electromagnetics Research Symposium, Hangzhou, China, 25–27 April 2022.
32. Geng, X.; Dong, G.; Xia, Z.; Liu, H. SAR Target Recognition via Random Sampling Combination in Open-World Environments. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2023**, *16*, 331–343. [CrossRef]
33. Neal, L.; Olson, M.; Fern, X.; Wong, W.; Li, F. Open Set Learning with Counterfactual Images. In Proceedings of the Computer vision—ECCV 2018: 15th European Conference, Munich, Germany, 8–14 September 2018.
34. Jo, I.; Kim, J.; Kang, H.; Kim, Y.; Choi, S. Open Set Recognition by Regularising Classifier with Fake Data Generated by Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018.
35. Sabour, S.; Frosst, N.; Hinton, G. Dynamic routing between capsules. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017.
36. Sun, X.; Lv, Y.; Wang, Z.; Fu, K. SCAN: Scattering Characteristics Analysis Network for Few-Shot Aircraft Classification in High-Resolution SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5226517. [CrossRef]
37. Geng, C.; Huang, S.; Chen, S. Recent advances in open set recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3614–3631. [CrossRef] [PubMed]
38. Liu, J.; Tian, J.; Han, W.; Qin, Z.; Fan, Y.; Shao, J. Learning multiple gaussian prototypes for open-set recognition. *Inf. Sci.* **2023**, *626*, 738–753. [CrossRef]
39. Bendale, A.; Boult, T.E. Towards open set deep networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1563–1572.
40. Yoshihashi, R.; Shao, W.; Kawakami, R.; You, S.; Iida, M.; Naemura, T. Classification-Reconstruction Learning for Open-Set Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
41. Guo, Y.; Camporese, G.; Yang, W.; Sperduti, A.; Ballan, L. Conditional Variational Capsule Network for Open Set Recognition. In Proceedings of the 18th IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
42. Dimity, M.; Niko, S.; Michael, M.; Feras, D. Class Anchor Clustering: A Loss for Distance-based Open Set Recognition. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021.
43. Cheng, X.; Huo, Y.; Lin, S.; Dong, Y.; Zhao, S.; Zhang, M.; Wang, H. Deep Feature Aggregation Network for Hyperspectral Anomaly Detection. *IEEE Trans. Instrum. Meas.* **2024**. *early access*. [CrossRef]