Few-Shot Ship Classification in Optical Remote Sensing Images Using Nearest Neighbor Prototype Representation

Jiawei Shi, Zhiguo Jiang, Member, IEEE, and Haopeng Zhang, Member, IEEE

Abstract—With the development of ship detection in optical remote sensing images, it is convenient to obtain accurate detection results and ship images. Owing to the superior performance of convolutional neural networks (CNNs), one way to acquire the category of ship is to train a classifier using numerous ship images. However, the classification performance of CNN may degrade in the case of a small number of training samples. To solve this problem, we propose a metric-based few-shot method to generate novel concept (class) representation using nearest neighbor prototype. Different from image-to-image measure in common few-shot methods, we use an image-to-feature measure. We map small number of samples to the feature space through CNN, and generate prototypes by computing nearest neighbor value on each dimension of the feature separately. Our method is validated on patch-level ship image dataset, a reproduced ship classification dataset based on HRSC2016. The experimental results demonstrate the accuracy and robustness of our method for ship classification with a small amount of labeled data.

Index Terms—Convolutional neural network (CNN), few-shot learning, nearest neighbor, remote sensing image (RSI), ship classification.

I. INTRODUCTION

GREAT progress in remote sensing has been made in recent years, which makes it more convenient to get high-resolution and high-quality optical remote sensing images (RSIs). The increasing number of remote sensing data has greatly promoted the research of interpretation of optical RSIs. Recently, ship detection [1]–[4] is playing an important role in military and civilian fields, and has attracted increasing attention. Tang et al. [1] adopted wavelet coefficients to catch compressed domain for ship candidate fast extraction, and exploited convolutional neural networks (CNNs) and extreme leaning machine for final decision. SVDNet [2] was designed based on CNN and singular value decomposition (SVD), where SVD was used to automatically learn convolutional filters from training data. Yang et al. [4] made use of saliency segmentation to extract candidate regions and local binary pattern descriptors to achieve discriminative ship features. Li et al. [3] proposed hierarchical selective filtering layers based on faster R-CNN [5], and from left to right).

Fig. 1 Visualization of intraclass diversity and interclass similarity in ship classification. (a) Intraclass diversity and (b) interclass similarity make ship classification a challenging task. (a) Intraclass diversity: ContainerA (first row) and aircraft carrier (second row). (b) Interclass similarity: WhidbeyIsland versus Austen, ContainerA versus Container, Medical versus Car CarrierB, ContainerA versus Car CarrierB, and WhidbeyIsland versus Sanantonio (from top to bottom and from left to right).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
methods, we propose a novel nearest neighbor prototype representation (NNPR) CNN for few-shot ship classification. First, support images and query images are converted into feature vectors by CNN feature extractor. Then, these feature vectors are compared by nearest neighbor algorithm to obtain prototype representations for each class. After that, prototype representations and the feature vector of the query image are used to calculate image-to-feature measure to determine whether the query image belongs to a corresponding category. In summary, the contributions of our work are four-fold, which are as follows.

1) We propose a novel few-shot ship classification method NNPR. It can efficiently classify ship images even if only one sample is provided per class. It addresses the issue of data scarcity in RSI classification.

2) We develop a new approach to dynamically generate category’s prototypes by computing nearest neighbor value on each dimension between support and query features. The classification is achieved by an image-to-feature measure between the query feature and the prototypes of different categories. It can generate more possible prototypes beyond limited labeled samples.

3) We adopt random sampling to train a unified CNN feature extractor instead of the popular episodic training strategy (ETS), which has significantly improved the classification performance. In this way, the CNN feature extractor needs to be trained only once to adapt to different experimental scenes, which makes it more appreciate to practical applications.

4) We comprehensively evaluate the performance of our proposed NNPR on a public remote sensing ship dataset HRSC2016. Our method outperforms state-of-the-art few-shot methods in classification accuracy by at least 4%.

The remainder of this article is organized as follows. Section II gives a brief review of previous works about ship classification with scarce samples and few-shot methods in general images. Section III presents the proposed method. Experimental results and discussions are included in Section IV. Finally, conclusion is drawn in Section V.

II. RELATED WORKS

A. Ship Classification With Scarce Samples

Previous literature for ship classification with limited training samples generally addresses the issues on the DA [6]–[8] and transfer learning [8]–[12]. Lu and Li [8] performed DA by rotating images and expanding the edges with pixel pads. Li et al. [6] increased the number of data by cropping and flipping. Besides, 3-D ship models are used to synthesize more images [7]. Transfer learning usually conducts fine-tuning (FT) based on a pretrained network, and could achieve a better result than training from scratch. Yosinski et al. [22] have proved that the transfer features outperform the random weights. The literatures [9]–[11] separately pretrained a network from automatic identification system, unlabeled SAR scene images and ImageNet [23], and transferred the knowledge to the target dataset. Rostami et al. [12] obtained a classifier using two encoders with shared embedding space. The classifier trained from optical domain with a large number of labeled samples can generalize well on SAR domain. It should be noticed that Rostami et al. [12] and our method both belong to few-shot methods via metric learning. Rostami et al. [12] used domain adaption and the categories in optical domain require to be the same as the categories in SAR domain. In addition, Rostami et al. [12] can only classify an image into two classes, ship or no-ship. Our approach can recognize specific categories and does not need other domain data.

In general, although the aforementioned methods achieve good results, a certain amount of training data is still necessary. These methods may not work when using an extremely limited number of samples. Comparatively, our NNPR method works even if only one sample is provided.

B. Few-Shot Methods

The first one-shot method [24] was proposed in 2006, which held that one or few labeled samples are helpful for building new concept. The following of this section will introduce two kinds of few-shot methods.

1) Meta-Learning-Based Methods: The works in [16], [18], [20], and [25] acquire a meta-learner model that can generalize to unseen class quickly by accessing scarce samples. In order to give the model proper weights, model-agnostic meta-learning [16] built the meta-learner that was fine-tuned by second derivation on each tasks. Ravi and Larochelle [18] made use of LSTM-based [26] optimizer as a meta-learner and searched for good initial weights. MM-Net [20] built a complicated architecture for few-shot classification.

Meta-learning-based methods can achieve compelling performance. However, their training process and architecture are complex. These methods are limited by the temporally linear hidden state dependence [27] and always need FT. In contrast, our proposed NNPR is a kind of nonparametric method, and does not require FT.

2) Metric-Learning-Based Methods: The works in [14], [15], [17], [19], [21], [28], and [29] use similarity metric to get the category. Koch et al. [14] introduced deep learning and metric learning into the field of few-shot. The Siamese network accepted paired images and determined whether they are the same category by the distance between the pair of samples. Vinyals et al. [15] proposed ETS and full context embeddings to achieve better performance. Subsequent research works generally adopt ETS. Snell et al. [17] calculated the average of feature vector and viewed it as the corresponding prototype. Sung et al. [19] considered the relationship between labeled and unlabeled samples and proposed a learnable classifier to replace fixed metric classifier.

The proposed method NNPR is also metric-based. However, NNPR is different from the aforementioned methods. In NNPR, we propose an improved prototype representation and utilize random sampling to train CNN instead of ETS.
III. METHODOLOGY

A. Problem Definition

Given a query set \( Q \), few-shot ship classification aims to correctly classify unlabeled samples in \( Q \). To achieve this goal, we need support set \( S \), which contains \( C \) image classes and \( N \) labeled samples per class. The category in \( Q \) belongs to \( C \) classes that are the same as in \( S \). This configuration is called \( C \)-way \( N \)-shot. However, the limited samples in \( S \) degrade the classification performance. The related works employed auxiliary set \( A \) with a large number of samples to facilitate model training. The categories with abundant samples are called base classes, and the classes with limited samples are novel classes. The set \( A \) only contains base classes. To meet the requirement of our practical application of ship classification, the support set \( S \) in our method contains both base classes and novel classes and correspondingly both of them are considered in \( Q \).

B. Proposed NNPR Framework

The framework of the proposed NNPR CNN is illustrated in Fig. 2. NNPR consists of two main components, namely CNN feature extractor and prototype metric. In the first component, the features are extracted by the trained CNN model. In the second component, the prototypes were calculated based on the extracted features. Then, we can get the category result by computing image-to-feature similarity.

1) CNN Feature Extractor: In our method, CNN feature extractor is responsible for extracting features of the images in \( S \) and \( Q \) when inferring. It is a feature extractor trained on set \( A \) using random sampling, and we call it as unified CNN feature extractor. ETS is widely used in few-shot methods. However, this strategy requires that the training process strictly matches the test process. That is to say, the CNN feature extractor needs to be retrained on set \( A \) following corresponding test process as the experimental scenes change (detailed in Section IV-B). In our perspective, this way is complicated in practical applications, and it would result in performance degradation. In contrast, random sampling scheme can make full use of data and thereby can improve the classification performance. Furthermore, the unified CNN feature extractor needs to be trained only once on set \( A \) for different settings of the experimental scene, which is convenient and efficient.

2) Prototype Metric: The phase of prototype metric is non-parametric. We assume there exists a central point for each class in feature space and extract the central point as the prototype for each class. The process of prototype metric is shown in Fig. 3. After acquiring the feature vectors, the prototype of each class is composed by nearest neighbor algorithm. For instance, a query image \( \hat{x} \) is encoded by the unified CNN feature extractor as

\[
f_\phi(\hat{x}) = [a_1, a_2, \ldots, a_m]
\]

where \( f_\phi \) is the unified CNN feature extractor and \( \{a_i, i = 1, 2, \ldots, m\} \) are the feature values. The prototype of each category is composed of the nearest neighbor values on each dimension between support features belonging to the same category and the query feature. The prototype of class \( k \) can be described by

\[
p_k = \arg\min_{a \in f_\phi(x_i)} ||f_\phi(\hat{x}) - f_\phi(x_i)|| \quad (x_i, y_i) \in S_k
\]

where \( S_k \) represents a subset of the set \( S \) belonging to the \( k \)th class, \( x_i \) is an image in subset \( S_k \), and \( y_i \) is the label of \( x_i \). It is obvious that different query images or support images correspond to different prototypes, and we can treat it as a dynamic process.

Then, we can get the distance between the query image feature and different prototypes by image-to-feature measure. Since the prototype should be at the central point in cluster for each category, the distance between the prototype and other samples belonging to same category should be minimal. Banerjee et al. [30] have shown that Bregman divergences meet such requirements. We select the Euclidean distance, which is one of the simplest forms of Bregman divergences, as the distance function. The last step of this phase is to use softmax over the Euclidean distance to get the probability of each class as

\[
\text{prob}(\hat{y} = k|\hat{x}) = \frac{e^{-d(p_k, f_\phi(\hat{x}))}}{\sum_k e^{-d(p_k, f_\phi(\hat{x}))}}
\]
Fig. 3. Flowchart of the phase of prototype metric in detail. The query feature is shared by the sparse blue dotted lines. The first gray area enclosed by the dotted lines indicates the operation to calculate the prototype representations using nearest neighbor algorithm, and the second indicates the operation to calculate the distance using Euclidean distance.

### TABLE I

| Dataset       | The number of categories | Categories                                      | The number of Categories about ships | Categories about ships | The number of ships |
|---------------|--------------------------|-------------------------------------------------|--------------------------------------|------------------------|---------------------|
| NWPU VIHR-10  | 10                       | airplane, ship, storage tank, tennis court, harbor, etc. | 1                                    | ship                   | 302                 |
| LEVIR         | 3                        | airplane, ship, oil pot                         | 1                                    | ship                   | 3,025               |
| DOTA          | 15                       | airplane, ship, swimming pool, helicopter, roundabout, bridge, etc. | 1                                    | ship                   | 63,000              |
| HRSC2016-1    | 1                        | ship                                            | 1                                    | ship                   | 2,976               |
| HRSC2016-2    | 3                        | merchant ship, warcraft, aircraft carrier       | 3                                    | merchant ship, warcraft, aircraft carrier | 2,248               |
| HRSC2016-3    | 15                       | Medical boat, Ticonderoga, Nimitz, Perry, Austin, etc. | 15                                   | Medical boat, Ticonderoga, Nimitz, Perry, Austin, etc. | 2010               |

Note: HRSC2016 is listed by level. Italic number is an approximation.

where \( d \) denotes Euclidean distance and \( \text{prob} \) is the possibility that indicates the category \( \hat{y} \) belongs to the \( k \)th class given the query image \( \hat{x} \).

### C. CNN Architecture

CNN architecture refers to CNN feature extractor in Fig. 2. Since the CNN feature extractor is a classification network removed fully connected layers, it can be any classic classification network, such as AlexNet [31], VGG [32], GoogLeNet [33], [34], ResNet [35], DenseNet [36], etc. In this article, we use DenseNet [36] with 121 convolution layers, i.e., the DenseNet-121 structure, as the CNN feature extractor for its good performance in image classification.

Our CNN feature extractor benefits from the powerful feature extraction capability of DenseNet. In Section IV-D3, we also explore the impact of different CNN architecture on few-shot ship classification results.

### IV. EXPERIMENTS

#### A. Dataset and Evaluation Metric

In order to assess our method, we need an optical RSI classification dataset that includes multiple ship categories. To the best of our knowledge, there is currently no such public dataset, so we adopt HRSC2016 dataset [37] to generate experimental data. HRSC2016 is a dedicated dataset for ship detection where all images are captured from six harbors on Google Earth. It contains three-level annotations: ship, ship categories, and ship types. Since HRSC2016 provides sufficient number of type labels, we reproduce the images from the third level in HRSC2016 to construct proper dataset for our experiments. Apart from HRSC2016, another three main public remote sensing datasets, including ships, are NWPU VHR-10 [38], LEVIR [39], and DOTA [40]. As can be seen in Table I, these three datasets contain multiple categories, but the ship is only regarded as one whole category without specific ship category labels. Thus, NWPU
In our experiment, we assume that the patch-level ship images are accurately detected by the upstream ship detection algorithm. The experiments are all conducted based on the ship patch images. The spatial resolution of the patch images ranges from 0.4 to 2 m. The data include a total of 2010 images from 15 categories of ships, i.e., Arleigh Burke, ContainerA, Ticonderoga, Perry, Tarawa, Nimitz, Enterprise, WhidbeyIsland, CommanderA, Austin, Sanantonio, Container, Car CarrierA, Car CarrierB, and Medical boat. Since the two types of aircraft carriers (Nimitz and Enterprise) are very similar, we treat them as a whole class. After merging them into one category, we reproduce a patch-level dataset, including 14 categories of ships, which is named PLSID (patch-level ship image dataset). Fig. 4 is a glance of all the 14 categories. Fig. 5 takes five-shot as an example and shows the distribution of ships in PLSID. The ships in PLSID have different sizes and shapes. Fig. 4 resizes the ship images to similar sizes only for good visualization. We regard the first nine categories as the base classes, and the remaining five categories as the novel classes. In base classes, 70% of them and all novel classes are in set $S$ or $Q$ for the test stage.

Besides, we also use miniImageNet [15] dataset to evaluate our method. MiniImageNet is a subset of ImageNet and widely used to evaluate different methods in the field of few-shot classification, which has 100 categories in total with 600 images per category. The size of the image is $84 \times 84$, and the dataset is divided following Ravi and Larochelle [18].

We employ accuracy (Acc) to quantitatively evaluate the performance. The accuracy is a widely used metric in classification tasks. It is defined by the following formula:

$$\text{Acc} = \frac{T}{T + F}$$

where $T$ indicates the number of correct classification results, and $F$ indicates the number of false classification results. That is to say, $T + F$ indicates the total number of images to be classified. The accuracy is presented in base and novel, where base and novel are the accuracy on base classes and novel classes, respectively. Moreover, in order to make the results more objective, the 95% confidence interval is also reported together with the accuracy.

B. Experiment Setup

In order to comprehensively evaluate the performance, we report the accuracies in four terms. With set $A$ fixed, the accuracy is evaluated when 1 novel class (1-novel) or 5 novel classes (5-novel) is added to $S$, and each novel class has 1 labeled sample (1-shot) or 5 labeled samples (5-shot). We refer to these terms as 1-shot 1-novel, 1-shot 5-novel, 5-shot 1-novel, and 5-shot 5-novel, respectively. Since the base class has 9 categories, 1-novel and 5-novel can be considered as 10-way and 14-way, respectively.

The unified CNN feature extractor is pretrained on ImageNet and utilizes $A$ for FT. We adopt Adam [42] optimizer with an initial learning rate of $1 \times 10^{-4}$, which is halved every 25 epochs. After training, fully connected layers are removed. In the test stage, we randomly select $N$ labeled sample(s) and five test samples for all novel classes and base classes to construct $S$ and $Q$. The reason for using the same number of labeled samples is to avoid performance degradation on novel classes caused by data imbalance. Considering the objectivity, this process is repeated 300 times to get the average accuracy. The accuracy in 1-novel is the average accuracy obtained by 300 trials on 5 different novel classes. Note that all images are resized to $224 \times 224$ for graphics processing unit (GPU) parallel computing. The data in base classes in $S$ does not appear in set $A$.

All experiments were conducted in Python with PyTorch and ran on a computer with two CPUs on Intel (R) Xeon (R) CPU E5-2630 v4 @2.20 GHz and four GPUs of NVIDIA GeForce GTX 1080 Ti.

C. Comparison With Other Methods

We compare our method with six approaches for few-shot ship classification, including the traditional machine learning method (LBP+SVM), FT, deep nearest neighbor neural network.
(DN4) [29], relation network (RN) [19], matching network (MN) [15], and prototypical network (PN) [17]. The last four methods are state-of-the-art metric-based deep learning methods.

Table II shows the comparison results. We can treat PLSID dataset as fine-grained dataset because it contains all kinds of ships. It is well known that fine-grained classification task is more difficult than general task due to huge intraclass diversity and interclass similarity. This makes the performance gap of the comparison methods on PLSID much larger than on Omniglot [43] or miniImageNet [15] dataset.

The first row in Table II is the traditional machine learning approach. We use local binary pattern (LBP) to extract the features and SVM to classify the ships. In order to avoid SVM performance degradation on novel classes when too many base class samples in the auxiliary set A are used to train SVM, the same number of base class and novel class samples are used when training the SVM. This makes the accuracy of base and novel classes similar. The low accuracy indicates that the traditional method may be not suitable for few-shot classification.

The second and third rows are FT results. FT is a popular approach in transfer learning. Generally, it fixes the weights of pretrained model except the last layer and retrains on target task. In our experiments, we use the feature extractor with a fully connected layer for FT. FT has achieved a high accuracy on the base classes because the feature extractor has been well trained on these classes. However, the performance on novel classes is very limited due to the lack of novel class samples. In order to overcome this problem, we perform DA on each novel support sample by randomly rotating 90°, 180°, and 270°, horizontal flipping, vertical flipping and changing their color, brightness, contrast, and sharpness. After DA, the number of novel support samples increases 12-fold. From the third row of the results in Table II named as FT-DA, we can see that the results of the novel classes have improved significantly. Even so, three of the four novel accuracies still lag behind our method in which no DA is adopted. It indicates the obvious superiority of our approach in data scarcity issue.

Compared to other few-shot methods, the proposed NNPR achieves remarkable performance. Taking the second best method PN and novel as examples, NNPR reports a 1-shot 1-novel accuracy of 61.15%, an improvement of 8.47% over 52.68%, and a 5-shot 1-novel accuracy of 70.78%, an improvement of 3.99% over 66.79%. When using 5-novel, the accuracy improvements over PN are 11.71% (32.29% versus 44.00%) and 8.92% (52.28% versus 61.20%) under 1-shot and 5-shot, respectively. These results have demonstrated the effectiveness of our NNPR in few-shot ship classification tasks and the superiority to the other methods. When each class has only one labeled sample, our prototype metric is identical with PN. Under this circumstance, the reason that our approach outperforms PN is that we adopt random sampling instead of ETS. Our unified CNN feature extractor can obtain more informative ship features. Moreover, it can leverage image-to-feature measure to compute the distance between nearest neighbor prototypes and query feature under 5-shot. Different prototype expressions are generated dynamically by composing features according to different support and query samples. This way can generate more possible prototypes and generalize beyond limited labeled samples when some query samples are different from any support samples. The contribution of the unified CNN feature extractor and image-to-feature measure to performance improvement will be analyzed in the ablation study in Section IV-D1.

NNPR also obviously outperforms other three few-shot approaches based on deep learning, i.e., DN4, RN, and MN. It is worth mentioning that although DN4 has achieved state-of-the-art results in general image few-shot classification, it achieves poor performance in our experiments. NNPR improves the 5-shot accuracy on novel by 33.50% (37.28% versus 70.78%) and 34.32% (26.88% versus 61.20%). When using 1-shot, the accuracy improvements over DN4 are 41.16% (19.99% versus 61.15%) and 28.21% (15.79% versus 44.00%). In addition to ETS, another reason for the low accuracy of DN4 is the local invariant features. Different from general images, the targets in RSIs have a variety of directions and aspect ratios. Thus, images with the same category have various local features after resized to $224 \times 224$. In NNPR, CNN feature extractor quantifies the local features to image-level features, which effectively addresses the aforementioned problem.

Besides, we also validate our proposed method NNPR on miniImageNet dataset. The backbone Conv-64F is widely used and described in [15]. The experimental result is shown in...
Table III, and proves that our proposed method is not only applicable to ships, but also to general objects.

D. Analysis and Discussion

1) Ablation Study: Our article contributes two components, i.e., image-to-feature measure (I2F) and unified CNN feature extractor (UFE) trained with random sampling. We perform ablation study to analyze the influence of the two components. The baseline method uses ETS and image-to-image (I2I) measure. I2I measure compares the query image feature to each support image feature and gets the category. I2I can be seen as a strategy based on image matching. We add two contributed components to baseline one by one. It should be noticed that I2F measure is identical with I2I measure under baseline one by one. It should be noticed that I2F measure is identical with I2I measure under baseline one by one. It should be noticed that I2F measure is identical with I2I measure under baseline one by one. It should be noticed that I2F measure is identical with I2I measure under baseline one by one.

As can be seen in Table IV, I2F measure performs much better than I2I measure. For example, the I2F brings 1.37% novel (from 65.07% to 66.44%) and 7.14% base (from 78.23% to 85.37%) accuracy improvements under 5-shot 1-novel when using ETS, indicating the advantage of I2F measure over I2I measure. In addition, the UFE is also superior to ETS. Our NNPR (the last row in Table IV) can gain 4.34% novel (from 66.44% to 70.78%) and 2.72% base (from 85.37% to 88.09%) improvements over ETS under 5-shot 1-novel when using I2F. It demonstrates that the unified CNN feature extractor trained with random sampling is effective in extracting richer and more informative ship features.

2) Robustness Against the Deviations of Detection Boxes: In the previous experiments, we assume that the patch-level ship images are accurately obtained by other ship detection algorithms. The deviation between the detection box and the ground truth is inevitable although the detection algorithms have been achieving more and more compelling detection performance in recent years. In order to verify the robustness against the deviations of detection boxes, four of the best methods in Table II, i.e., NNPR, PN, MN, and RN, are selected for performance comparisons in the presence of deviations. We simulate the original images in our PLSID dataset with offset from 10% to 40%. In addition, each image is randomly selected from eight directions for deviation. Fig. 6 shows an example of one image with 20% offset.

Results of Fig. 7 show that our proposed method NNPR performs the best. Most of the result curves in Fig. 7 keep the performance gap between NNPR and other methods. The novel accuracy in Fig. 7(a) and (b) and base accuracy in Fig. 7(b) and (d) even tends to increase the performance gap. However, Fig. 7(a) and (c) reveal a narrow performance gap on base accuracy between NNPR and MN as the offset increases. When the offset is 40% in Fig. 7(c), MN surpasses NNPR with a slight advantage of 0.41%. In addition, PN is the second best method without deviation. With the occurrence of deviation, PN is exceeded by MN on most indicators, but our method still achieves the best performance. It proves that NNPR is robust against the deviations of detection boxes.

3) Influence of Different Feature Extractors: In this section, we explore the influence of different backbone networks used as the feature extractor. We use other four popular networks (mentioned in Section III-C) as the feature extractor, respectively, and report average results of 300-time repeatable experiments in Table V.

The last column of Table V shows the performance upper-bound, i.e., the test results of the feature extractor on base classes. The performance ranking of these backbones on PLSID dataset is the same as on ImageNet dataset. Similar results are obtained during the few-shot test stage. It is worth noting that during the few-shot test stage, DenseNet-121 has a slight advantage over ResNet-50. This may benefit from its unique way of feature propagation. This is reason we use DenseNet-121 as the feature extractor. Based on the results in Table V, using a more elegant network structure and a deeper network as feature extractor may further improve few-shot ship classification performance.

4) Influence of Different Distance Function: The choice of distance function is a key part. In the previous experiment, we chose Euclidean distance as the distance function. Table VI lists the results of replacing Euclidean distance with cosine distance, and shows that cosine distance has a slight advantage over Euclidean distance. In terms of the performance, cosine distance is a better choice. However, we use the nearest neighbor algorithm to generate prototypes, and the nearest neighbor algorithm is executed by the Euclidean distance between different feature values. We use Euclidean distance as distance function for consistency. Besides that, as we said before, we assume...
TABLE IV
MEAN ACCURACIES (%) ABOUT ABLATION STUDY, WITH 95% CONFIDENCE INTERVALS

| Training Strategy | Metric       | 1-shot 1-novel | 5-shot 1-novel | 1-shot 5-novel | 5-shot 5-novel |
|-------------------|--------------|----------------|----------------|----------------|----------------|
|                   |              | novel          | base           | novel          | base           | novel          | base           | novel          | base           | novel          | base           |
| ETS               | I2I          | 52.68±2.30     | 69.39±2.79     | 65.07±2.50     | 78.23±2.00     | 32.29±2.75     | 61.74±2.77     | 44.51±2.61     | 72.08±2.08     |
| ETS               | I2F          | 52.68±2.30     | 69.39±2.79     | 65.07±2.50     | 78.23±2.00     | 32.29±2.75     | 61.74±2.77     | 44.51±2.61     | 72.08±2.08     |
| UFE               | I2F          | 61.15±3.41     | 80.04±2.30     | 70.78±2.47     | 88.09±1.41     | 44.00±3.27     | 72.33±2.42     | 61.20±2.66     | 82.23±1.61     |

Note: Underlines represent the same accuracies as the same position on the previous row. The ETS and UFE denote episodic training strategy and the unified CNN feature extractor trained with random sampling, the I2I and I2F denote image-to-image measure and image-to-feature measure, respectively.

Bold entities indicate the best results.

![Fig. 7. Result curves with offset. (a) Relationship between accuracy and offset under 1-shot 1-novel. (b) Relationship between accuracy and offset under 5-shot 1-novel. (c) Relationship between accuracy and offset under 1-shot 5-novel. (d) Relationship between accuracy and offset under 5-shot 5-novel. Each set of results contains novel accuracy and base accuracy.](image1)

![Fig. 7. Result curves with offset. (a) Relationship between accuracy and offset under 1-shot 1-novel. (b) Relationship between accuracy and offset under 5-shot 1-novel. (c) Relationship between accuracy and offset under 1-shot 5-novel. (d) Relationship between accuracy and offset under 5-shot 5-novel. Each set of results contains novel accuracy and base accuracy.](image2)

![Fig. 7. Result curves with offset. (a) Relationship between accuracy and offset under 1-shot 1-novel. (b) Relationship between accuracy and offset under 5-shot 1-novel. (c) Relationship between accuracy and offset under 1-shot 5-novel. (d) Relationship between accuracy and offset under 5-shot 5-novel. Each set of results contains novel accuracy and base accuracy.](image3)

![Fig. 7. Result curves with offset. (a) Relationship between accuracy and offset under 1-shot 1-novel. (b) Relationship between accuracy and offset under 5-shot 1-novel. (c) Relationship between accuracy and offset under 1-shot 5-novel. (d) Relationship between accuracy and offset under 5-shot 5-novel. Each set of results contains novel accuracy and base accuracy.](image4)

TABLE V
MEAN ACCURACIES (%) ABOUT THE INFLUENCE OF DIFFERENT BACKBONE NETWORKS ON PERFORMANCE, WITH 95% CONFIDENCE INTERVALS

| Backbone       | 1-shot 1-novel | 5-shot 1-novel | 1-shot 5-novel | 5-shot 5-novel | Performance Upper-Bound |
|----------------|----------------|----------------|----------------|----------------|-------------------------|
|                | novel          | base           | novel          | base           |                         |                         |                         |                         |
| AlexNet        | 37.46±2.89     | 34.82±2.89     | 55.04±2.78     | 54.95±2.65     | 26.67±2.62           | 30.83±2.81           | 43.64±2.71           | 47.99±2.61           | 77.47           |
| VGG-16         | 45.81±1.33     | 62.95±3.08     | 56.33±2.80     | 77.98±2.11     | 28.92±2.77           | 55.76±3.00           | 44.36±2.64           | 71.17±2.10           | 87.73           |
| ResNet-34      | 55.17±3.43     | 76.54±2.52     | 63.53±2.71     | 85.23±1.69     | 37.28±3.11           | 68.39±2.66           | 53.88±2.67           | 80.20±1.78           | 91.76           |
| ResNet-50      | 59.04±3.33     | 79.95±2.34     | 68.91±2.57     | 88.04±1.53     | 42.77±3.17           | 70.95±2.53           | 60.69±2.70           | 82.86±1.70           | 94.69           |
| DenseNet-121   | 61.15±3.41     | 80.04±2.30     | 70.78±2.47     | 88.09±1.41     | 44.00±3.27           | 72.33±2.42           | 61.20±2.66           | 82.23±1.61           | 93.77           |

Note: Performance upper-bound is presented with accuracy using the corresponding classification network on nine base classes.

prototypes are cluster center points for each class in feature space, and they should have minimal distance to other points. Euclidean distance meets the requirement.

5) Error Analysis: We conduct error analysis by confusion matrix. We only show the confusion matrices under 5-shot using our method, as shown in Fig. 8. The entry in the ith row and jth column denotes the rate of query images from the jth class that are classified as the ith class.

From Fig. 8, we can see that misclassification is mutual, i.e., if class A is easily misclassified into class B, class B is also easily misclassified into class A. The confusion matrices and Fig. 1 indicate that misclassification is caused by intraclass diversity and interclass similarity. Fig. 1 shows that the category ContainerA has a large intraclass diversity and interclass similarity, which makes it easy to misclassify into four out of five novel categories (Container, Car CarrierA, Medical, and Car CarrierB). Another novel category Sanantonio is easily confused with WhidbeyIsland. Because the network parameters are not optimized on the novel categories, misclassification is also easily caused between different novel categories, such as Medical and Car carrierB. Even for base categories, WhidbeyIsland and Austen are so similar that it is difficult to distinguish them.
Although aircraft carrier in the first row of Fig. 1(a) has a large intraclass diversity, it is trained on CNN feature extractor and also has large differences from other categories. Therefore, the accuracy of aircraft carrier is very high. The aforementioned partial confusing categories are also shown in Fig. 1(b).

V. CONCLUSION

In this article, we have proposed an effective few-shot ship classification approaches named NNPR. We train the unified CNN feature extractor independently and obtain prototype representation by nearest neighbor algorithm. A series of experiments demonstrates that our proposed approach is more effective than traditional methods and recent state-of-the-art few-shot methods based on deep learning. Our results can be a benchmark for few-shot ship classification, and may encourage other researchers to further study this challenging problem. In future work, we will try to find a better way to represent the prototype and propose an end-to-end approach for the combined task of ship detection and few-shot ship classification.

REFERENCES

[1] J. Tang, C. Deng, G.-B. Huang, and B. Zhao, “Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 3, pp. 1174–1185, Mar. 2015.
[2] Z. Zou and Z. Shi, “Ship detection in spaceborne optical image with SVD networks,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 10, pp. 5832–5845, Oct. 2016.
[3] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, “HSF-Net: Multiscalar deep feature embedding for ship detection in optical remote sensing imagery,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 12, pp. 7147–7161, Dec. 2018.
[4] F. Yang, Q. Xu, and B. Li, “Ship detection from optical satellite images based on saliency segmentation and structure-LBP feature,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 5, pp. 602–606, May 2017.
[5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Neural Inf. Process. Syst., 2015, pp. 91–99.
[6] J. Li, C. Qu, and S. Peng, “Ship classification for unbalanced SAR dataset based on convolutional neural network,” J. Appl. Remote Sens., vol. 12, no. 3, 2018, Art. no. 035010.
[7] C. M. Ward, J. Harguess, and C. Hilton, “Ship classification from overhead imagery using synthetic data and domain adaptation,” in Proc. OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 2018, pp. 1–5.
[8] C. Lu and W. Li, “Ship classification in high-resolution SAR images via transfer learning with small training dataset,” Sensors (Switzerland), vol. 19, no. 1, 2019, Art. no. 63.
[9] Z. Huang, Z. Pan, and B. Lei, “Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data,” Remote Sens., vol. 9, no. 9, 2017, Art. no. 907.
[10] H. Lang, S. Wu, and Y. Xu, “Ship classification in SAR images improved by AIS knowledge transfer,” IEEE Geosci. Remote Sens. Lett., vol. 15, no. 3, pp. 439–443, Mar. 2018.
[11] Y. Wang, C. Wang, and H. Zhang, “Ship classification in high-resolution SAR images using deep learning of small datasets,” Sensors, vol. 18, no. 9, 2018, Art. no. 2929.
[12] M. Rostami, S. Kolouri, E. Eaton, and K. Kim, “Deep transfer learning for few-shot SAR image classification,” Remote Sens., vol. 11, no. 11, 2019, Art. no. 1374.
[13] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, “When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 5, pp. 2811–2821, May 2018.
G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in Proc. 32nd Int. Conf. Mach. Learn., Lille, France, 2015, vol. 37, pp. 1–8.

O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one-shot learning,” in Proc. Neural Inf. Process. Syst., 2016, pp. 3630–3638.

C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 1856–1868.

J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in Proc. Neural Inf. Process. Syst., 2017, pp. 4080–4090.

S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in Proc. Int. Conf. Learn. Representations, 2017, pp. 1–11.

F. Sung, Y. Yang, Z. Li, X. Tao, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1199–1208.

Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei, “Memory matching networks for one-shot image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4080–4088.

A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, “Few-shot learning with global class representations,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9715–9724.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in Proc. Neural Inf. Process. Syst., 2014, pp. 3320–3328.

O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015.

L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 594–611, Apr. 2006.

T. Ramalho and M. Garnelo, “Adaptive posterior learning: Few-shot learning with a surprise-based memory module,” in Proc. Int. Conf. Learn. Representations, 2019, pp. 1–14.

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1741, 1997.

N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, “A simple neural attentive meta-learner,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–7.

V. Garcia and J. Bruna, “Few-shot learning with graph neural networks,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–13.

W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Lao, “Revisiting local descriptor based image-to-class measure for few-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7260–7268.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, “Clustering with Bregman divergences,” J. Mach. Learn. Res., vol. 6, pp. 1705–1749, 2005.

A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Neural Inf. Process. Syst., 2012, vol. 25, pp. 1097–1105.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations, 2015, pp. 1–14.

C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2828.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4700–4708.

Z. Liu, L. Yuan, L. Weng, and Y. Yang, “A high resolution optical satellite image dataset for ship recognition and some new baselines,” in Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods, 2017, vol. 2017, pp. 324–331.

G. Cheng, H. Han, P. Zhou, and L. Guo, “Multi-class geospatial object detection and geographic image classification based on collection of part detectors,” ISPRS J. Photogrammetry Remote Sens., vol. 98, pp. 119–132, 2014.

Z. Zou and Z. Shi, “Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images,” IEEE Trans. Image Process., vol. 27, no. 3, pp. 1100–1111, Mar. 2018.

G.-S. Xia et al., “DOTA: A large-scale dataset for object detection in aerial images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3974–3983.

L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, “Meta-learning with differentiable closed-form solvers,” in Proc. Int. Conf. Learn. Representations, 2019, pp. 1–15.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Representations, 2015, pp. 1–15.

B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, “One shot learning of simple visual concepts,” in Proc. Annu. Meeting Cogn. Sci. Soc., 2011, vol. 33, no. 33, pp. 2568–2573.

Zhiguo Jiang (Member, IEEE) received the B.S. degree in automation from the School of Automation, Beijing University of Posts and Telecommunications, Beijing, China, in 2018. He is currently working toward the M.S. degree in aerospace engineering with the Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics, Beihang University, Beijing, China.

His research interests include computer vision and deep learning applications in remote sensing.

Dr. Jiang currently serves as a standing member of the Executive Council of China Society of Image and Graphics and also serves as a member of the Executive Council of Chinese Society of Astronautics. He is an Editor for the Chinese Journal of Stereology and Image Analysis.

Jiawei Shi received the B.S. degree in automation from the School of Automation, Beijing University of Posts and Telecommunications, Beijing, China, in 2018. He is currently working toward the M.S. degree in aerospace engineering with the Department of Aerospace Information Engineering (Image Processing Center), School of Astronautics, Beihang University, Beijing, China.

His research interests include computer vision and deep learning applications in remote sensing.