Contrastive Learning for Fine-grained Ship Classification in Remote Sensing Images

Jianqi Chen, Keyan Chen, Hao Chen, Wenyuan Li, Zhengxia Zou*, and Zhenwei Shi, *Member, IEEE*

***Abstract*—Fine-grained image classification can be considered as a discriminative learning process where images of different subclasses are separated from each other while the same subclass images are clustered. Most existing methods perform synchronous discriminative learning in their approaches. Although achieving promising results in fine-grained visual classification (FGVC) in natural images, these methods may fail in fine-grained ship classification (FGSC) problem in remote sensing (RS) images due to the highly "imbalanced fineness" and "imbalanced appearances" of ships among subclasses. To tackle the issue, we propose an asynchronous contrastive learning-based method for effective FGSC. The proposed method, which we refer to as "Push-and-Pull Network (P^{2}Net)", includes a "push-out stage" and a "pull-in stage", where the first stage forces all the instances to be de-correlated and then the second one groups them into each subclass. A dual-branch network is designed to separate/de-correlate the images with each other, while an Integration Module is designed to aggregate the de-correlated images into their corresponding subclass together with a Proxy-based Module designed for acceleration. In this way, the correlation between subclasses can be decoupled, which in turn makes the final classification much easier. Our method can be trained end-to-end and requires no additional annotations other than category information. Extensive experiments are conducted on two large-scale FGSC datasets (FGSC-23 and FGSCR-42). Our method outperforms other state-of-the-art approaches. Ablation experiments also suggest the effectiveness of our design. Our code is available at https://github.com/WindVChen/Push-and-Pull-Network.**

***Index Terms*—Fine-grained classification, contrastive learning, ship classification, remote sensing.**

I. INTRODUCTION

FINE-GRAINED ship classification (FGSC) task in remote sensing (RS) images aims at differentiating subclasses of the main ship category. FGSC has great application perspectives in both civil and military fields. In addition, the recent development of many high-performance object detection methods [1]–[3] also provides important research foundations for the downstream FGSC task. Compared with coarse classification tasks that only differentiate higher-level classes (*e.g.*, ships, airplanes, cars, etc), the fine-grained classification task is much more challenging. The cues to distinguish different subclasses of ships are subtle, while the differences intra-class are significant due to different imaging conditions and different ship appearances. Fig. 1 shows an example of low intra-class similarity and high inter-class similarity in ship classification.

Fig. 1. The challenge of high inter-class similarity and low intra-class similarity in FGSC task in RS. In the first row, ships of different subclasses are similar in appearance, while in the second row, ship appearances of the same subclass are quite different.

FGSC has attracted increasing attention in RS field [4]–[8]. Existing methods of this topic can be roughly divided into two groups. One group focuses on hand-crafted features [4], [5] and to some extent combines with the image features of convolution neural networks (CNN) [6]. The other group makes use of the powerful deep learning (DL) method and takes advantage of the few sample learning [7] and data augmentation [8] to ease the problem of lacking enough data. Despite the promising results achieved, attention has been paid more to learning with few samples, and less to fine-grained classification itself. Thus previous methods may not be well adapted to the above-mentioned inter-class and intra-class problems. In other words, the power of DL has not been fully utilized due to insufficient data support.

The work was supported by the National Natural Science Foundation of China under the Grant 62125102, and the Fundamental Research Funds for the Central Universities. (Corresponding author: Zhengxia Zou (zhengxia-zou@buaa.edu.cn))

Jianqi Chen, Keyan Chen, Hao Chen, Wenyuan Li and Zhenwei Shi are with the Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with the Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China.

Zhengxia Zou is with Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China.

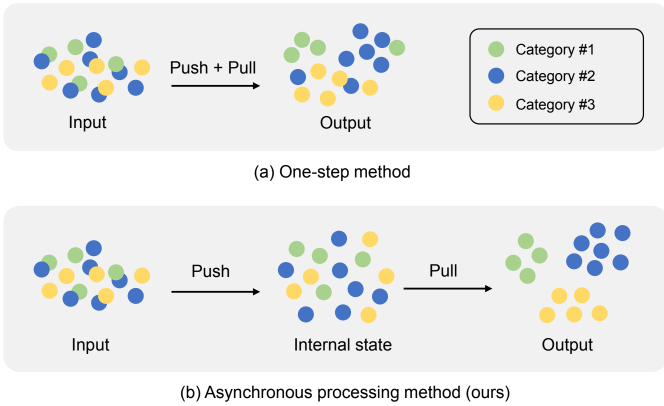


Fig. 2. The main idea behind our method. (a) The synchronous push-pull process of previous methods (one-step). (b) The asynchronous push-pull process of the proposed P^2Net (two-step). Different colors denote different subclasses.

Benefit from two large-scale FGSC datasets [9], [10] proposed in the last year, the limited data problem has been alleviated to some degree, and a very natural idea is to apply the latest DL methods [11]–[25] in FGVC to FGSC. In the FGVC task, each subclass is almost at the same level of fineness, and corresponds to one specific type of object (*e.g.*, black-footed albatross in CUB_200_2011 dataset [26], 2012 BMW M3 coupe in Standford Cars dataset [27]). However, in the FGSC task in RS, the fineness of subclasses is not that uniform. For example, the subclasses of aircraft carrier and destroyer usually attract more attention and are finer than the subclasses of fishing boat. Moreover, there is also an imbalance of ship appearances among subclasses due to the limitations of reality. In subclasses such as aircraft carrier, ship appearances are just that few worldwide, yet there are many in fishing boat subclass. These two phenomena in the FGSC task, which we name "imbalanced fineness" and "imbalanced appearances", bring more challenges to the FGSC task.

The fine-grained classification problem is commonly viewed as a push-and-pull process, i.e., images of one subclass are encouraged to get closer to their corresponding subclass while getting far away from other subclasses. Some FGVC methods [23], [25] design additional loss terms for the push and pull purposes respectively, while the others [19]–[22] only employ the final classification loss function (often CrossEntropyLoss). However, most of these methods share the same characteristic that the loss terms are attached all together at the output end of the classifier. In other words, they perform the push and pull processes synchronously, which we refer to as one-step. However, due to the aforementioned two imbalanced problems, the synchronous discriminative learning is hard to sufficiently separate the similar images of different subclasses before aggregating images of the same subclass.

To address the above issues, we propose an asynchronous contrastive learning-based method for effective FGSC. We look at the push-and-pull process from a novel perspective, and the two processes in our method are conducted asynchronously. Given a set of training images, our method first takes each image as an individual one and forces them to be pushed far away and de-correlated from each other. Then, the pull process is applied to enforce the separated images back to their corresponding subclass clusters. We refer to our method as "Push-and-Pull Network (P^2Net)". Fig. 2 shows a basic idea of our method. In the push-out stage, each input image is regarded as an individual class and to be dispersed from each other as much as possible. We leverage the recent advances in contrastive learning (CL) and design a dual-branch network for the data separation process. Then in the pull-in stage, an Integration Module is proposed to cluster the dispersed images. We also propose a Proxy-based Module to accelerate the image clustering process. Previous FGVC methods usually set single [28], [29] explicit or implicit proxy [30], and force the images to get close to their corresponding proxies. Different from all these approaches and by taking into account the challenges of FGSC, we set multiple explicit proxies to represent each subclass.

Although we perform the push-and-pull process asynchronously, our network can be trained in an end-to-end fashion and only require image-level annotation for training (weakly-supervised). We conduct experiments on two large-scale FGSC datasets: FGSC-23 and FGSCR-42. Ablation studies and visualization are further conducted to illustrate the idea of our method. The results demonstrate that the proposed method achieves higher accuracy compared with other state-of-the-art methods.

Our contributions can be summarized as follows:

1) We introduce a new method for the RS image FGSC task. The proposed method takes an asynchronous push-and-pull strategy to tackle the challenges in the FGSC task, whereas previous methods follow a synchronous discriminative learning process and thus suffer from the "imbalanced fineness" and "imbalanced appearances" of RS objects.
2) We take advantage of the recent advances in CL and propose a novel push-and-pull network (P^2Net). The network consists of a Push-out part and a Pull-in part, which are carefully designed to make the classification both efficient and effective.
3) The proposed P^2Net can be trained end-to-end and requires only image-level annotations. The experimental results on the FGSC-23 and FGSCR-42 datasets validate the superiority of the P^2Net on the FGSC task compared with the existing methods.

The rest of the paper are organized as follows. The related work is described in Section II. The details of our network are given in Section III. Experimental results and visualization are conducted in Section IV, and the conclusions are drawn in Section V.

II. Related Work

In this part, we briefly review the recent progress in RS image FGSC. We then review the natural image FGVC methods studied in the field of computer vision. We also review the recent advances in CL, which are related to our network.

A. FGSC in Optical RS Images

Fine-grained RS object classification has raised increasing attention in recent years [31]–[36]. Sumbul *et al.* [31] focused on fine-grained street tree classification and proposed a zero-shot learning method. They then explored to use of multi-source data for the classification task [33]. Ni *et al.* [32] introduced the adaptive density discrimination into fine-grained terrain classification. Nie *et al.* [36] proposed a classifier-adaptive earth mover's distance for classification of few-sample fine-grained aircraft. These methods all achieve some good results. However, the study on the FGSC task is still in its infancy.

Previous methods of RS image FGSC are mainly based on hand-crafted features and data utilization due to the lack of large-scale FGSC datasets. Shi *et al.* [5] combined the Fourier transform with CNN to classify the ships, and they then fused more hand-craft features in their later work [6]. Qi *et al.* [8] studied data augmentation in FGSC. Shi *et al.* [7] proposed to utilize few-shot learning to solve FGSC problems.

Recently, two large-scale FGSC datasets (FGSC-23 [9] and FGSCR-42 [10]) are proposed. With the datasets, Zhang *et al.* [9] proposed a DL-based method using extra attribute annotations. Zhao *et al.* [37] focused on the low-resolution FGSC task and proposed a feature balance strategy with the use of both super-resolution and low-resolution images. Chen *et al.* [38] introduced a hierarchy and exclusion graph to model the label hierarchy. Although achieving some good results, these methods depend on extra annotations and inputs, thus limiting the possibility of their large-scale application. In our method, we focus on weakly-supervised approaches where we assume only image-level annotations are available. Also, compared with some recent methods that focused on few samples [39] and the interpretability of the network [40], here we aim to address the two issues of "imbalanced fineness" and "imbalanced appearances", which may bring the FGSC task challenges.

B. FGVC in Natural Images

FGVC also has drawn increasing attention in the computer vision field recently. Methods in the FGVC task can be divided into three categories. The first category mainly focuses on the representation learning ability of the network. Lin *et al.* proposed B-CNN [11] that exploited the discriminative feature by a bilinear pooling on two local parts of an input image. Yu *et al.* [13] proposed a hierarchical bilinear pooling network based on B-CNN. Although these methods can improve the classification accuracy, they usually have complex structures and the computational cost is intensive.

Methods of the second category focus on fine-grained annotations among subclasses and attract the most attention currently. The early methods in this direction usually utilize auxiliary annotations as supervisions. Some research has been done on the utilization of object's key part [15] and image's key point [41] annotations. The latest research in this category focuses on how to exploit image-level weakly-supervised information for fine-grained classification. Zheng *et al.* [19] designed a trilinear attention sampling method to detect object key parts. Ding *et al.* [22] proposed to use pyramid structure to determine key regions. The study found that using the key part location information can effectively improve the accuracy of fine-grained classification. However, many approaches [20], [22], [42] in this direction often set pre-defined bounding boxes like anchors in their pre-processing stage which requires prior knowledge and is not flexible.

The last category of methods is based on metric learning, in which similar research problems have been explored in face identification tasks [43], [44]. Sun *et al.* [23] proposed constrained pair-based loss where different features of the input image pairs are pushed and pulled. Xu *et al.* [25] exploited the discriminative feature by using both the pair-based loss and proxy-based loss. The metric learning methods usually bring no extra computational cost, as it mainly focuses on the design of loss functions. Our P^{2}Net can also be classified as a metric learning method. However, different from the above methods that perform push and pull processes synchronously, we propose a two-step way that can achieve better results in the FGSC task.

C. Contrastive Learning

Contrastive learning is a recently emerged research topic in unsupervised/self-supervised representation learning [45]–[47]. In CL, the input batched images are first transformed into two views by using different augmentation strategies. The views of the same image are considered the positive samples, while the views of different images are considered negative samples. After that, the augmented images are encoded by an encoder network and then mapped to a feature space by a designed projection network, where a carefully designed contrastive loss is applied. The contrastive loss aims to repulse negative samples while attracting the positive ones. Thus the features of positive samples can be clustered while that of the negative ones can be dispersed. Bachman *et al.* [48] argue that the CL can maximize the mutual information among latent representations of different unlabeled images, which makes CL successful as an unsupervised learning way.

Recently, many effective CL methods [49]–[51] are proposed. He *et al.* [49] propose MoCo (Momentum Contrast) method to reduce the memory cost while fusing more information by designing a dynamic queue. Chen *et al.* [51] propose a much simpler siamese network by introducing a Stop Gradient strategy. Using these CL methods, the need for expensive image labeling can be alleviated, and a large number of unlabeled images can be leveraged, enabling better initialization/pretraining for neural networks. In our P^{2}Net, we also design a CL-like structure in the Push-out part, consisting of a custom-designed contrastive loss and a projection structure. However, the goal of the Push-out part is quite different from the common CL methods. The current CL methods aim to address the insufficiency of labeled images and explore a better-pretrained model, while we introduce the CL idea into our method to tackle the "imbalanced fineness" and "imbalanced appearances" issues by separating the input images at an image level. What else, most CL methods are implemented in a two-step way (pretrain first, then finetune) in applications, whereas the P^{2}Net in an end-to-end way.

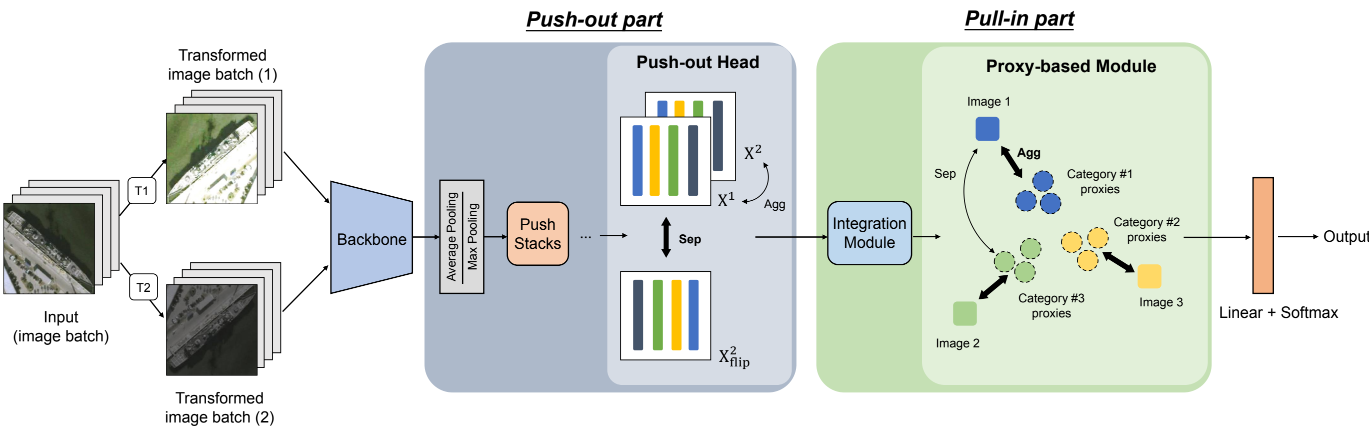


Fig. 3. The detailed structure of P[2]Net. The *Agg* and *Sep* are the abbreviations of Aggregation and Separation operations. In the Push-out part, different colors denote different input images at an image level, while in the Pull-in part, colors denote different subclasses at a class level.

III. Methodology

In this section, we introduce the details of the proposed P[2]Net, including the pipeline, network architecture, and loss functions.

A. Overview

In the proposed P[2]Net, we have two processing stages: a push-out stage and a pull-in stage. Detailed structures of the P[2]Net are shown in Fig. 3. In the training phase, the P[2]Net takes in two randomly augmented views T^1 and T^2 from input image batch I and extracts the corresponding features through its network backbone. Then the features are processed by the push-out part first and then by the pull-in part. In the push-out part, input images are regarded as individual classes of their own. Views of the same image are forced to get close while views of different images are forced to be far away from each other. The Push-out part will enforce the images to be projected to a feature space where the adhesion/correlation between images is reduced. Then, in the Pull-in part, each image again belongs to its corresponding subclass and is forced to be close to a set of proxies which represent each subclass. In the inference phase, the Push-out part and the Pull-in part can be mostly removed, thus the network has almost no additional computational cost. Note that our method is flexible in different backbone structures such as ResNet [52], DenseNet [53], and so on. In this paper, we take ResNet50 as the default backbone of our architecture. The details of the two parts are given in the following.

B. Details of Push-out Part

As we mentioned in Section I, the "imbalanced fineness" and "imbalanced appearances" intensify the difficulty of the FGSC task. Subclasses in RS objects are usually not at the same level of fineness and the problem of the intra-class dissimilarity and inter-class similarity is further exacerbated, which leads to the ineffectiveness of the existing one-step methods. Our Push-out part is to alleviate the above challenges by de-correlating the images. The design of the Push-out part is inspired by recent CL methods [49]–[51] which have achieved amazing representation results by exploring the relation between unlabeled images.

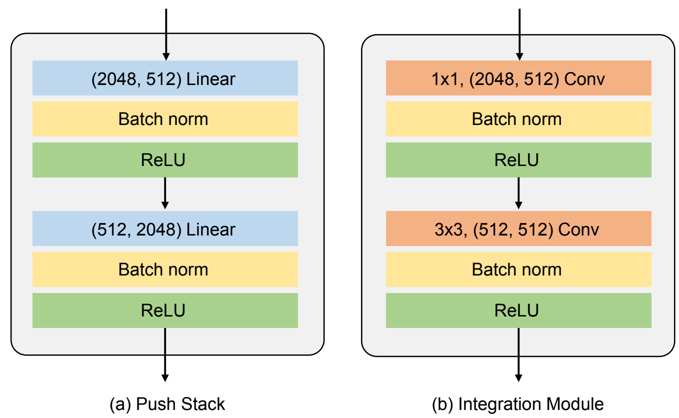


Fig. 4. The details of the Push Stack and Integration Module. (a) Push Stack. (b) Integration Module. Note that the structures displayed here are based on ResNet50.

The Push-out part takes the features extracted by the backbone as input. These features are first processed by a global average/max pooling operation in the spatial dimension and then projected to a new feature space by several designed Push Stacks. We refer to the representations of input images after Push Stack as $X^i = f(T^i)$, $i \in \{1, 2\}$. We design our Push Stack as a set of fully connected layers with batch normalization and relu activations. The detailed architectures are displayed in Fig. 4. The features X^1, X^2 will be input to the Push-out head where the views of different images are separated.

To clearly elaborate the operation of the Push-out head, we provide a pseudo code in Algorithm 1 to illustrate the process. For the features $X^1, X^2 \in \mathbb{R}^{B\times C}$, B, C are the batch size and channel number respectively, we unfold them into feature vectors: $X^{\{1,2\}} = \{x_1^{\{1,2\}}, x_2^{\{1,2\}}, x_3^{\{1,2\}}, ..., x_B^{\{1,2\}}\}, x \in \mathbb{R}^C$. The feature vectors at the same location in X^1 and X^2 correspond to different augmented views of the same input image, and the different locations correspond to different input images. We flip X^1 or X^2 at their sample dimension, in which

Algorithm 1: The process of the Push-out head

Input: $X^{\{1,2\}} = \{x^{\{1,2\}}_{1,2,...,B}\}$, features of different augmented views after Push Stack

Input: $X^{\{1,2\}}_{flip} = \{x^{\{1,2\}}_{B,B-1,...,1}\}$, the flipped version of feature X

Define: $Aggregate(\cdot)$, a loss function to make the augmented views of the same image close (See in Section III-D1)

Define: $Separate(\cdot)$, a loss function to make the augmented views of different images separated (See in Section III-D1)

// m denotes different input images
for m *in* $1 : B$ **do**
    // The augmented views of the same image are forced to aggregate
    $Aggregate(x^1_m, x^2_m)$

    // The augmented views of different images are forced to separate
    $Separate(x^1_m, x^2_{B-m+1})$
end

way we can get $X^{\{1,2\}}_{flip} = \{x^{\{1,2\}}_B, x^{\{1,2\}}_{B-1}, x^{\{1,2\}}_{B-2}, ..., x^{\{1,2\}}_1\}$. When the batch size is even, features at the same location in the flipped one and the other correspond to different input images. Therefore, we force the flipped one and the other to be far away. At the same time, we force features from the same image in X^1 and X^2 to get close. To achieve the effect of the aggregation and separation in the Push-out head, we design two loss functions, the details of which can be found in Section III-D1.

Compared with the existing CL methods [49]–[51], [54], the Push-out part explores a more efficient contrastive loss and a more suitable projection structure for the FGSC task. The final Push-out loss led by our flipped operation (See details in Section III-D) and the Push Stack designed can save more memory space, and reduce more calculations while leading to higher classification accuracy, which can be verified in Section IV-E. Also should be noticed that the standing point for the designed Push-out part is different from the current CL methods, where the Push-out part is for tackling the "imbalanced fineness" and "imbalanced appearances" issues we discussed above, while the current CL methods are usually for addressing the insufficiency of labeled data and model pre-training.

C. Details of Pull-in Part

After the Push-out ope, the input images are de-correlated in their feature space. The Pull-in part includes an Integration Module and a Proxy-based Module. The Integration module's structure is displayed in Fig. 4. The first 1×1 convolution layer is used for reducing the number of features from the channel dimension. The second convolution layer is used for information fusion from a spatial dimension with a 3×3 kernel. Through the Integration module, the network can better process the potential relation of the previous separated features, thus speeding up the pull process.

To further enforce the separated images to be pulled back into their corresponding subclasses, the Proxy-based Module is designed after the Integration Module. We use $Z^i = g(T^i)$ to denote the features of different augmented views after the Integration module. Different from the description in Section III-B, we unfold $Z^{\{1,2\}}$ in a new form: $Z^{\{1,2\}} = \{z^i_m, y_m\}_B, i \in \{1,2\}, m \in \{1 : B\}, y_m \in \{1 : K\}$, where B and K denote batch size and number of subclasses respectively. Here, we no longer look at each input image at the image level but at a class level, as we assign the subclass label attribute y_m to each z^i_m vector. That means the feature vectors at different locations (denoted by the subscript m) can share the same subclass attribute (denoted by y_m). Then, we construct a set of learnable proxy vectors whose number is positively related to the subclass number and size identical to z^i_m. We force each feature vector to get close to the proxies of its corresponding subclass. As we mentioned in Section I, the subclasses in the FGSC task are not at the same level of fineness as that in the FGVC task, and also consist of different numbers of appearances. Therefore, the previous proxy-based methods [28], [29] that just select one proxy vector to represent each subclass are not suitable for FGSC. Different from the previous methods, for each subclass, we select multiple proxies. Suppose $P = \{p_k | k \in \{1, 2, ..., NK\}\}, p_k \in \mathbb{R}^C$ represents the proxy vectors, where C, N, K denote the channel number, the number of each subclass's proxies and the subclasses number respectively.

We force z^i_m to be close to $\{p_k | k \in \{N(y_m - 1) + 1 : Ny_m\}\}$, while far away from $\{p_k | k \notin \{N(y_m - 1) + 1 : Ny_m\}\}$. What else, to prevent proxies of the same subclass to be identical, we also force proxies to get away from each other. Note that these proxies are explicitly set and will not add additional cost like the implicit way [30]. The detailed implementation of the aggregation and separation in the Proxy-based Module can be found in Section III-D2. The pseudo code of our Proxy-based Module is displayed in Algorithm 2.

D. Objective Function

The objective functions for training can be divided into three parts: (1) the loss of the Push-out part, (2) the loss of the Pull-in part, and (3) the loss of the final classifier. (For simplicity, some variables such as B, N, K in the following, if not specified, correspond to the same meaning mentioned before in Section III.)

1) Push-out Loss: As mentioned above, in the Push-out part, each input image is enforced to be separated from each other, and aggregated with augmented views of the same origin. Here we use cosine similarity to measure the degree of separation and aggregation as follows:

$$\begin{aligned} \mathrm{Sep}(a, b) &= \mathcal{D}(a, b) \\ \mathrm{Agg}(a, b) &= -\mathcal{D}(a, b), \end{aligned} \tag{1}$$

where $\mathcal{D}(a, b) = a \cdot b / (\|a\|_2 \cdot \|b\|_2)$ denotes the cosine similarity between the two input representation vectors a, b.

For the representation x^i_m of each image in each view after Push Stack, we force it to be far away from x^j_n, where $n =$

Algorithm 2: The process of the Proxy-based Module in the Pull-in part

Input: $Z^{\{1,2\}} = \{z_m^i, y_m\}_B$, features of different augmented views after Integration Module
Input: $P = \{p_k\}_{NK}$, proxies of subclasses
Define: $Aggregate(\cdot)$, a loss function to make the inputs closed (See in Section III-D2)
Define: $Separate(\cdot)$, a loss function to make the inputs separated (See in Section III-D2)

/*Pull the input features into corresponding subclass*/
// i denotes different augmented views
for i *in* $\{1, 2\}$ **do**
    // m denotes different input images
    for m *in* $1 : B$ **do**
        // k denotes different proxies
        for k *in* $1 : NK$ **do**
            if $k \in \{N(y_m - 1) + 1 : Ny_m\}$ **then**
                // Force z closed with the proxies of the same subclass
                $Aggregate(z_m^i, p_k)$
            else
                // Force z away from the proxies of the different subclasses
                $Separate(z_m^i, p_k)$
            end
        end
    end
    /*Force the proxies not be identical*/
    for k *in* $1 : NK$ **do**
        for k' *in* $1 : NK$ **do**
            if $k' \neq k$ **then**
                $Separate(p_k, p_{k'})$
            else
                continue
            end
        end
    end
end

$B + 1 - m$ and $j \neq i$. The representation x_n^j corresponds to the image at the same position after flipping the batch. The separation loss for one batch can be formulated as:

$$\mathcal{L}_{sep} = \sum_{m=1}^{B} \text{Sep}(x_m^1, sg(x_n^2)) + \text{Sep}(x_n^1, sg(x_m^2)), \quad (2)$$

where $sg(\cdot)$ denotes a stop-gradient operation and a symmetrical structure is adopted, which are proven to be effective in preventing model collapse [51].

To constrain similarity between the two views of the same image, we define the aggregation loss as:

$$\mathcal{L}_{agg} = \sum_{m=1}^{B} \text{Agg}(x_m^1, sg(x_m^2)) + \text{Agg}(x_m^2, sg(x_m^1)), \quad (3)$$

Our push-out loss is the average between the separation loss and the aggregation loss: $\mathcal{L}_{Push} = \frac{1}{2B}(\mathcal{L}_{sep} + \mathcal{L}_{agg})$. Note that the Push-out part is a dual-branch network (one branch starting with average pooling and one branch starting with max pooling), we calculate the push-out loss for each branch and then sum them up.

2) Pull-in Loss: In the Pull-in part, the previously separated images are again pulled into corresponding subclass proxies. Here, the objective function includes three parts: pull the images with proxies that represent the same subclass, push the images out of proxies that represent different subclasses, and separate the proxies from each other.

We use the class label y_m of each sample to guide its representation z_m^i close to proxy vectors belonging to the same category y_m, while far away from that belonging to different categories.

The aggregation and separation loss between the image feature and the proxies are defined as:

$$\mathcal{L}_{ap} = \frac{1}{2BN} \sum_{i=1}^{2} \sum_{m=1}^{B} \sum_{k=N(y_m-1)+1}^{Ny_m} \text{Agg}(z_m^i, p_k) \quad (4)$$

$$\mathcal{L}_{sp} = \frac{1}{2BN(K-1)} \sum_{i=1}^{2} \sum_{m=1}^{B} \sum_{k=1, k \notin S}^{NK} \text{Sep}(z_m^i, p_k) \quad (5)$$

where S denotes the set $\{N(y_m - 1) + 1 : Ny_m\}$.

As each category is represented by multiple proxies, to prevent these proxies to be identical, we further define a loss to separate the proxies as:

$$\mathcal{L}_{p} = \frac{1}{2NK(NK-1)} \sum_{k=1}^{NK} \sum_{k'=1, k' \neq k}^{NK} \text{Sep}(p_k, p_{k'}) \quad (6)$$

Our overall pull-in loss is $\mathcal{L}_{Pull} = \mathcal{L}_{ap} + \mathcal{L}_{sp} + \mathcal{L}_{p}$.

3) Classifier Loss: We choose the standard cross-entropy loss (abbreviated as CE) as the classification loss:

$$\mathcal{L}_{Cls} = \frac{1}{2B} \sum_{i=1}^{2} \sum_{m=1}^{B} \text{CE}(\hat{y}_m^i, y_m) \quad (7)$$

where $\hat{y}$ is the final network prediction, y is the label of the input.

By considering all the above losses, the final loss function for training is written as:

$$L = \alpha L_{Push} + \beta L_{Pull} + \gamma L_{Cls} \quad (8)$$

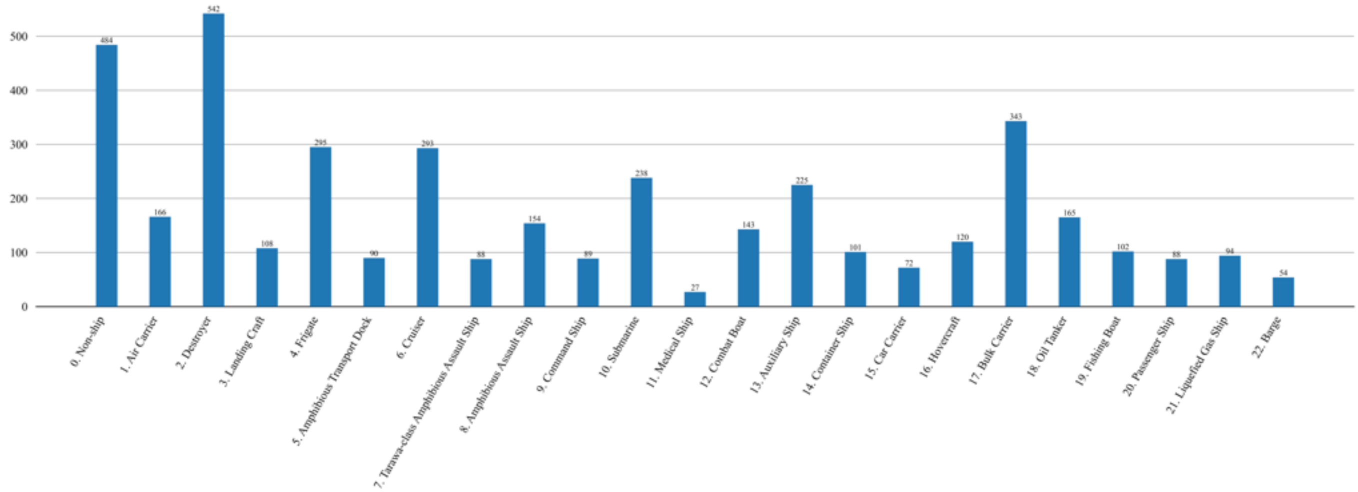
where α, β, and γ are the balancing weights of the three loss terms and we set them all to 1 in our work.

IV. Experimental Results and Discussions

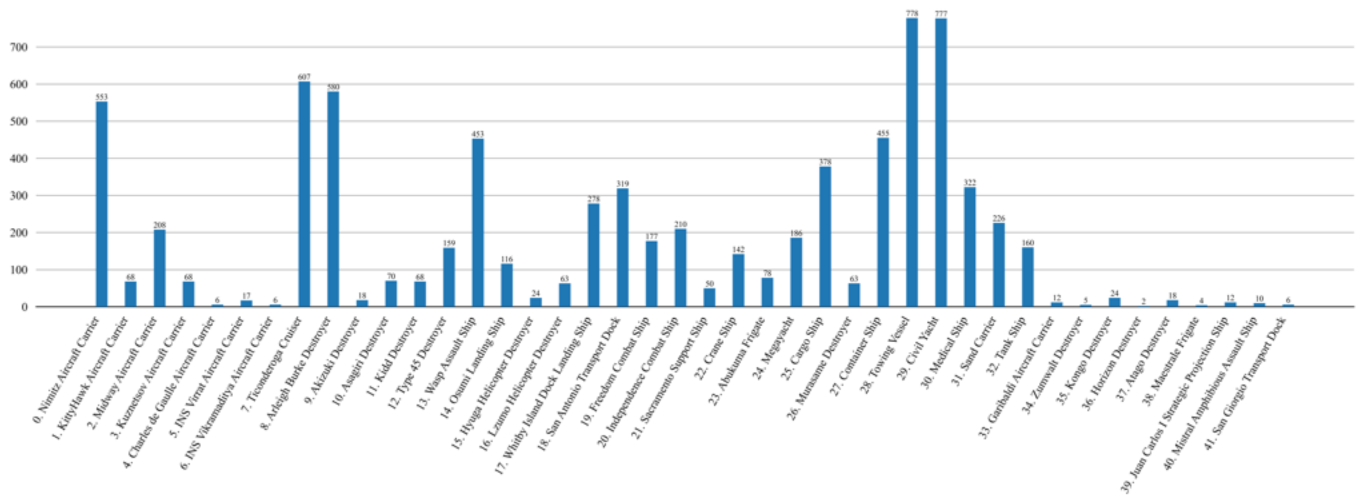
In this section, the FGSC datasets we adopt are first introduced. The second part gives the implementation details of our method. We present the evaluation protocol in the third part, and then our experimental results are given.

A. Datasets

We experiment on the datasets FGSC-23 [9] and FGSCR-42 [10]. In the following, we give a brief summary of these two datasets.



(a) Instances per category for the FGSC-23 dataset



(b) Instances per category for the FGSCR-42 dataset

Fig. 5. Number of annotated instances per category for (a) FGSC-23 and (b) FGSCR-42. The number in the front of the horizontal axis label represents the index of the category.

1) FGSC-23: has about 4,081 images from 23 subcategories. The images are mainly from Google Earth and GF-1 Satellite. Among the subcategories, there are aircraft carrier, destroyer, oil tanker, fishing boat, and so on.

2) FGSCR-42: has about 7,776 images from 42 subcategories. The images are mainly collected from Google Earth and the previous datasets (*e.g.*, DOTA [55]). In FGSCR-42, the military ships are further divided, *e.g.*, the aircraft carrier is divided into Nimitz-class, KittyHawk-class, and so on.

To better illustrate the existing “imbalanced fineness” and “imbalanced appearances” issues discussed in Section I, we present more details in Fig. 5. We can see that the military ships are usually divided into finer subclasses compared with the civilian ships, which corresponds to the “imbalanced fineness”. What else, considering that some categories like aircraft carriers have so few ships around the world, it leads to the problem of “imbalanced appearances”.

In our experiments, the datasets are divided into a training set, valid set, and test set in a 3:1:1 ratio. Since the occurrence frequency is not equal between military and civil subclasses, both datasets have an imbalanced sample problem (FGSC-23 is subtle while FGSCR-42 more serious). Therefore, we perform augmentation to the fewer-sample subclasses in the training set. The augmented operations include random crop and scale, random gaussian blur, random rotation, and random horizontal and vertical flip.

B. Implementation Details

We implement the backbone of our network based on ResNet50 pretrained on ImageNet. The Push-out part is designed as a dual-branch network, where the two branches start with average and max pooling respectively, and are followed by the same Push Stack structure. Through the branch with average pooling, the input images can be separated at a global level, while at a local level through the max-pooling branch.

TABLE I
COMPARISON RESULTS OF DIFFERENT APPROACHES ON FGSC-23 AND FGSCR-42 DATASETS. ALL THE METHODS ARE IMPLEMENTED BASED ON RESNET50. THE BEST RESULTS ARE MARKED IN BOLD, AND THE SECOND-BEST ONES ARE UNDERLINED.

Method	Params (M)	FLOPs (G)	AA	
			FGSC-23	FGSCR-42
ResNet50	23.6	4.12	86.92	91.62
HBPNet (ECCV 18 [13])	74.9	6.59	87.72	91.32
DCL (CVPR 19 [56])	23.8	4.12	85.35	90.24
TASN (CVPR 19 [19])	34.8	18.7	87.03	91.85
GFNet (NIPS 20 [21])	56.5	4.59	87.13	<u>92.03</u>
API-Net (AAAI 20 [24])	23.6	4.12	<u>87.78</u>	91.47
ProtoTree (CVPR 21 [57])	108.8	20.7	84.17	89.92
P^2Net (ours)	26.9	4.23	**88.99**	**93.21**

TABLE II
THE COMPARISONS WITH DIFFERENT PROJECTION STRUCTURES ON THE FGSC-23 DATASET. THE BEST ONES ARE MARKED IN BOLD.

Projection Structure	AA
SimCLR [50] projection	87.44
SimSiam [51] projection	87.01
Push Stack (ours)	**88.99**

TABLE III
THE COMPARISON RESULTS AMONG CONTRASTIVE LOSS OF SIMCLR, SIMSIAM, AND OUR P^2NET ON THE FGSC-23 DATASET. THE BEST ONES ARE MARKED IN BOLD.

Push-out Loss	AA
Loss in SimCLR [50]	88.21
ours (remove stopgrad)	**88.63**
Loss in SimSiam [51]	87.64
ours	**88.99**

We design the Push Stack as a bottleneck-like structure that can fuse much information while economical.

The category label of the image is the only annotation used for training. In the training phase, the network takes two views (T_1, T_2) transformed from the input images. The transformations are almost the same as the augmentations in the preprocess, except for replacing the operation of random crop and scale with a customized resize operation to ensure 224×224×3 input size. We argue that the standard resize operation that directly stretches the image to fulfill a specific size can lose a lot of information when the input image has a large aspect ratio, which is very common to the ship images in RS. Considering that, our resize operation stretches the image while keeping its original ratio, and then pads the rest space with all-zero pixel values.

We implement our method with Pytorch and train all the models on a single RTX 3090 GPU card. We adopt Stochastic Gradient Descent (SGD) optimizer for training with momentum of 0.9 and weight decay of 1e-4. The initial learning rate (lr) of the backbone is set to 0.01 and decays in a Cosine annealing strategy, while the lr of other network parts is positively related to that of the backbone (See details in Section IV-H). We also run 10 epochs warm-up to stabilize the training phase. All the models are trained for 100 epochs with a minibatch size of 64. The number of Push Stack and proxies in each subclass are set to 2 and 3 respectively if not specified, and the commonly used Xavier Uniform [58] scheme is adopted to initialize the proxies. Note that in the inference phase, Push-out and Pull-in parts are mostly removed, only the Integration Module being kept.

C. Evaluation Protocol

Considering the sample imbalance in the FGSC datasets, different from the existing methods that choose overall accuracy (OA) to evaluate the network performance, we adopt average accuracy (AA) which is more reasonable. The AA metric is defined as:

$$AA = \frac{1}{C}\sum_{c=1}^{C} Acc(c) \tag{9}$$

where C denotes the number of categories, and $Acc(\cdot)$ denotes the accuracy of each category.

We also make use of the accuracy rate (AR) and misclassification rate (MR) of each category to display the network's detailed performance, and confusion matrix (CM) for more intuitive visual perception (For brevity, we embed AR and MR into CM for display.). Floating Point Operations (FLOPs) and Model Parameters (Params) are also adopted to illustrate the computational complexity of the network in the inference phase.

D. Comparisons with State-of-the-Art Methods

Table I shows the performance evaluation of different approaches to FGSC-23 and FGSCR-42 datasets. Since there are still few effective weakly supervised FGSC in RS, we here choose some state-of-the-art methods [13], [19], [21], [24], [56], [57] in the FGVC for a fair comparison. Considering different fineness degrees of subclasses, we set the number of proxies to 3 in the FGSC-23 dataset and 2 in the FGSCR-42. From the results, we can see that our method achieves the best

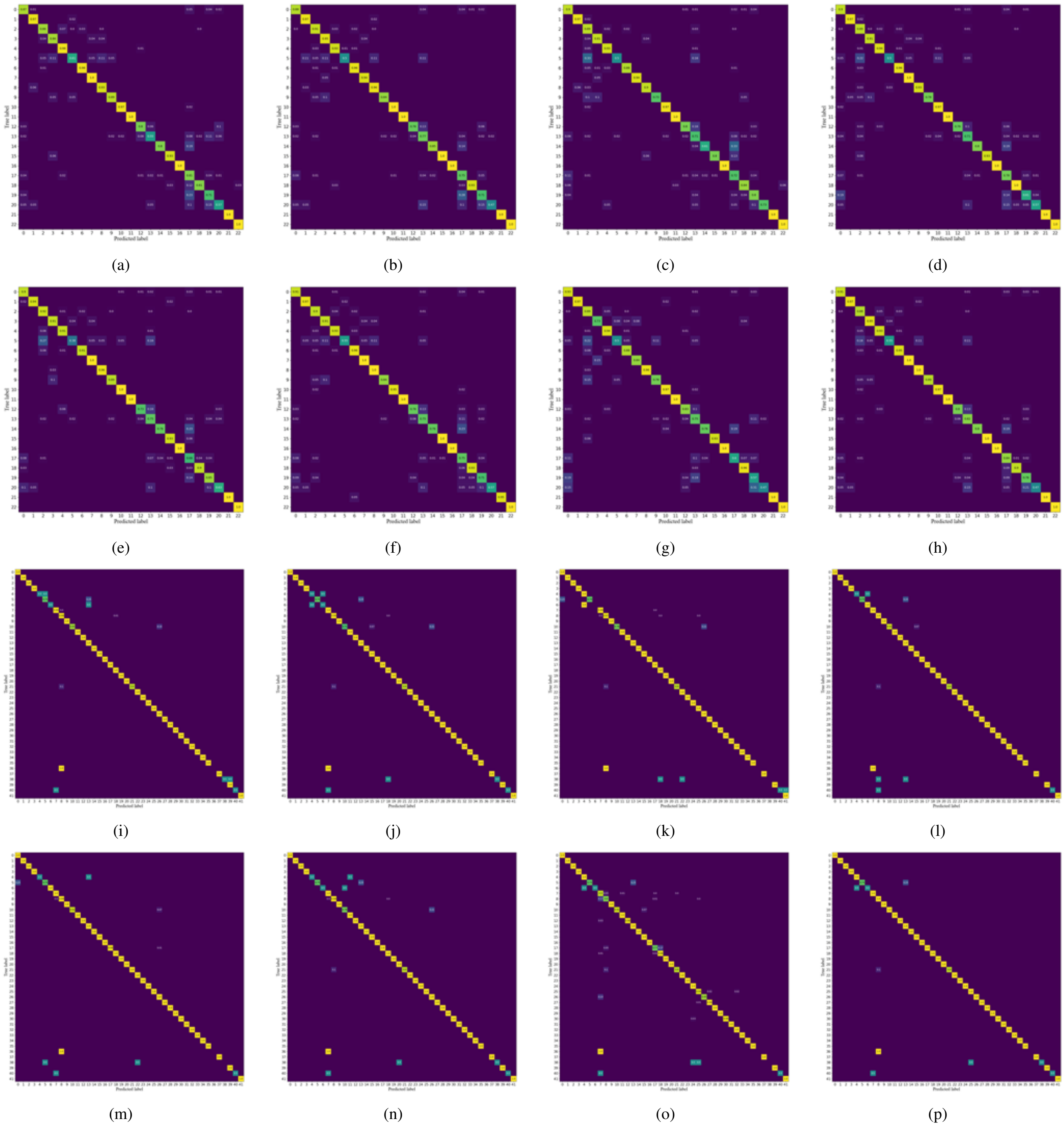


Fig. 6. Confusion matrixes of different methods on the FGSC-23 and FGSCR-42 datasets. The horizontal and vertical coordinates are the category index (See details in Fig. 5). (a)-(h) are respectively the results of ResNet50, HBPNet, DCL, TASN, GFNet, API-Net, ProtoTree, and our P^2Net on the FGSC-23 dataset. While (i)-(p) are the results on the FGSCR-42 dataset. For brevity, we filter out the values of zero in CM.

AA on both FGSC-23 and FGSCR-42 datasets (2.07% and 1.59% higher than baseline ResNet50 respectively). Moreover, our method has almost the same computation as the baseline, except for the limited amount of computation introduced by the Integration Module, and has far less computation than the methods like GFNet. To further verify that our AA improvement is not from the additional cost, we conduct ablation experiments in Section IV-F, where we can see that there will be no accuracy gain if we only have Integration Module applied.

We further display visually the CM results in Fig. 6, which detail the performance of different methods on each category of the FGSC-23 and FGSCR-42 dataset. The values on the CM's diagonal are the AR for each subclass, while the values

TABLE IV
THE ABLATION EXPERIMENTS OF THE DIFFERENT PARTS OF P^{2}NET ON THE FGSC-23 DATASET. THE COARSE ABLATION FOCUSES ON THE PUSH-OUT AND PULL-IN PARTS, WHILE THE FINE ABLATION FOCUSES ON THE MODULES IN EACH PART.

Ablation		Push-out Part		Pull-in Part		AA
		Average Pooling Branch	Max Pooling Branch	Integration Module	Proxy-based Module	
ResNet50		-	-	-	-	86.92
Coarse Ablation		+	+	-	-	86.35
		-	-	+	+	88.07
		+	+	+	+	**88.99**
Fine Ablation	Push	+	-	-	-	87.38
		-	+	-	-	86.82
		+	+	-	-	86.35
		-	+	+	+	88.23
		+	-	+	+	88.78
		+	+	+	+	**88.99**
	Pull	-	-	-	+	87.95
		-	-	+	-	86.98
		-	-	+	+	88.07
		+	+	-	+	87.32
		+	+	+	-	88.5
		+	+	+	+	**88.99**

of the other regions are the MR. Compared with the other methods, it can be seen that for many of the ship subclasses, the P^{2}Net achieves the best AR, while for the others, the P^{2}Net also achieves a relatively good result. In a more intuitive way, we can observe in both datasets (FGSCR-42 is more obvious) that the CM of P^{2}Net has fewer and darker valued regions off the diagonal, which corresponds to a smaller MR to other classes and again verifies our method's superiority over the others.

E. Comparisons with Contrastive Learning Methods

We also make comparisons with other CL methods to verify the effectiveness of the proposed projection structure and contrastive loss of our Push-out part. All experiments below are conducted on the FGSC-23 dataset [9] if not specified.

Effectiveness of Projection Structure. Previous CL methods like SimCLR [50] and SimSiam [51] do not pay specific attention to the projection structure, leading the structure to be somewhat too complex (SimSiam) or too simple (SimCLR). We thus propose the Push Stack to search for an optimal projection structure. Table II shows the performance of our Push Stack compared with the structure used in SimCLR [50] and SimSiam [51]. Compared with previous projection structures, our Push Stack achieves a much better result (1.55% and 1.98% higher than SimCLR and SimSiam respectively).

Effectiveness of Contrastive Loss. We also mentioned in Section III-B that our contrastive loss, i.e., Push-out loss with the flipped operation can achieve better results than previous CL methods. In Table III, we show the results compared with SimCLR [50] and SimSiam [51]. For a fair comparison, when compared with SimCLR, we remove the stopgrad operation borrowed from SimSiam. The results (0.42% better than SimCLR and 1.35% than SimSiam in AA) suggest the effectiveness of our Push-out loss and our flipped operation. We argue that the decay of SimSiam is because it only takes into account the aggregation of different augmented views of the same origin, but ignores the separation, while the decay of SimCLR is more probably caused by too much separation, as it not only separates the inter-view images but also the intra-view ones.

F. Ablation Studies

In this subsection, we conduct ablation studies to evaluate the contributions of each part in the proposed P^{2}Net.

We investigate the ablation of different parts in two ways: coarse and fine. The coarse ablation refers to the experiments where we focus on the effect of the Push-out part and the Pull-in part, while in the fine ablation, we further explore each component in the above two parts. As shown in Table IV, the results are divided into four parts. The first is the baseline ResNet50, the second is the coarse ablation, and the last two are the fine ablation of the Push-out and the Pull-in part. The results can be summarized as follows:

- In the coarse ablation experiment, we can see when only applying the Push-out part to the backbone, AA has a 0.57% decay, which is mainly because just the final linear layer has not enough capacity to pull the separated images back. When the Pull-in part is also applied, a noticeable AA increase (2.64%) can be observed. This again verifies the effectiveness of our idea in P^{2}Net, that is, the fine-grained classification task will be much easier if we do the push and pull asynchronously.
- In the fine ablation results of the Push-out part, the idea of the global and local level push branch talked in Section IV-B suggests effectiveness. When adding the Average Pooling Branch (APB) on the Pull-in part, there is a

TABLE V
THE EFFECTS OF DIFFERENT NUMBERS OF PUSH STACK ON THE FGSC-23 DATASET. THE BEST ONE IS MARKED IN BOLD.

Push Stack Number	AA
1	88.18
2	**88.99**
3	87.60

TABLE VI
THE EFFECTS OF DIFFERENT NUMBERS OF PROXIES IN EACH SUBCLASS ON THE FGSC-23 DATASET. THE BEST ONES ARE MARKED IN BOLD.

proxies of Each Subclass	AA
1	87.77
2	88.04
3	**88.99**
4	**88.99**
5	88.67

noticeable 0.71% increase in AA, and so is the Max Pooling Branch (MPB), though not that obvious. We may also note that when only APB is applied, there is an increase (0.46%) in AA, yet a little drop when MPB is applied and a big drop when both two applied. The reason behind this phenomenon may come from the balance and imbalance between the power of push and pull. The push power from APB can match with the pull power from the last linear layer, thus it can get a good result. As the MPB performs push power at a local level with its max-pooling operation, this local push power will diminish after the average pooling layer behind, thus it has little impact on the final result. However, when we accumulate the push power from APB and MPB, the power actually overwhelms that from the last linear layer, leading to an imbalance and an obvious deterioration in AA. That is also the reason why we design the Pull-in part to keep a balance.

- The fine ablation results of the Pull-in part further illustrate the essence of our P^{2}Net. As we add the dual-branch Push-out part, we can see a decrease in AA. Then when we again add the Integration Module and Proxy-based Module of the Pull-in part in turn, there is a gradual increase in AA. The effectiveness of our Integration Module and Proxy-based Module is thus verified in this part. The results also show that the accuracy has just little improvement (0.06%) when only the Integration Module applied. This further verifies that the accuracy increase in the inference phase is not from the additional cost.

From the above analyses, the contributions of each part in the P^{2}Net can be recognized, and the thought behind our method is further proved effective.

G. Controlled Experiments

Here, we investigate the effect of different settings in the Push-out part and the Pull-in part.

TABLE VII
THE PERFORMANCE OF P^{2}NET ON DIFFERENT BACKBONES ON THE FGSC-23 DATASET. THE BEST ONES ARE MARKED IN BOLD.

Backbone	Whether with P^{2}Net	AA
ResNet50 [52]	w/o	86.92
	w	**88.99**
DenseNet121 [53]	w/o	87.36
	w	**88.96**
Xception [59]	w/o	85.73
	w	**89.73**
Mobilev3_large [60]	w/o	85.86
	w	**87.34**
InceptionV4 [61]	w/o	87.74
	w	**88.89**

TABLE VIII
THE PERFORMANCE OF DIFFERENT TRAINING STRATEGIES ON THE FGSC-23 DATASET. THE BEST ONE IS MARKED IN BOLD.

Strategy	AA
Three inputs: two transformed and one original	85.5
Pretrain, then fine tune	87.5
End-to-end train (ours)	**88.99**

1) Different Push Stack Number: Table V shows the results with different numbers of Push Stacks. It can be seen there are both decays when using only a single Push Stack and three Push Stacks. We argue the decays are mainly caused by the imbalance of the push power and the pull power. On one hand, when using a single Push Stack, the backbone receives more impact from the Push-out part and the images can be farther away from each other and not easy to pull back. On the other hand, when using three Push Stacks, the adhesion among images can not be sufficiently removed, thus similar images of different subclasses cannot be divided that easily. Therefore, we set the number to 2.

2) Different Number of proxies in Each Subclass: As we argued in Section III-C, the subclasses in FGSC are unequal to each other, *e.g.*, more types of ships in the civil subclass while fewer in the military one. Thus the existing proxy-based methods [28], [29] that choose only one proxy for each subclass are not that useful in the FGSC task. In our work, more than one proxy is chosen to represent each subclass. The results in Table VI prove it effective to represent each subclass with several proxies. It can be observed that compared with only one proxy, when we set the number to 3 or 4, there is 1.22% increase in AA. It also should be noticed that proxies should not be set too many, as it may deviate the network training from our original purpose of pulling in the separated images (there is a decay after we set the number to 5).

H. Generalization, Training Strategy and lr Setting Studies

In this subsection, we investigate the generalization of our P^{2}Net on different backbone structures, the effects of different

TABLE IX
THE EXPERIMENTAL RESULTS OF DIFFERENT LR SETTINGS ON FGSC-23 DATASET, WHERE DETACH DENOTES THAT LR IN DIFFERENT NETWORK PARTS IS SPLIT, SCALE DENOTES HOW MANY TIMES THE LR OF OTHER PARTS IS BACKBONE'S. THE BEST ONES ARE MARKED IN BOLD.

lr (Backbone)	Whether detach	Scale (n*lr)		AA
		Modules other than Proxy-based	Proxy-based Module	
0.001	False	-	-	85.69
0.01		-	-	**88.17**
0.05		-	-	85.89
0.01	True	1	10	87.97
		5		**88.99**
		10		87.86
		5	5	87.5
			10	**88.99**
			15	87.84

training strategies and lr settings.

1) Generalization: We explore the generalization performance of P^{2}Net on some popular backbone structures, *e.g.*, DenseNet [53], Xception [59] and so on. From the results shown in Table VII, we can see when applied to other backbones, our P^{2}Net can also achieve a significant AA increase (4% especially on Xception), which proves a good generalization ability.

2) Different Training Strategies: Since we combine the idea of CL, a natural question is what if we first pretrain the Push-out part and then finetune the whole P^{2}Net. Moreover, we also care about whether the network can perform better when it takes three inputs: two augmented views and one original one (the formal two for Push-out and Pull-in part, the latter for classifier). With these questions, we conduct the experiments in Table VIII. First, we can see the pretrained training strategy fails to work well, we think this is because the Push-out part and the Pull-in part actually complement each other and guide each other in joint training to achieve good results. Second, the three-input strategy also fails (more than 3% decay in AA). We argue that it is the inconsistency of the inputs targeted by the loss functions, which may lead to ambiguity during network optimization, that causes the deterioration.

3) Different Settings of learning rate (lr): As mentioned in Section IV-B, we train our P^{2}Net with the backbone pretrained on ImageNet. However, there is no pretraining on other parts of P^{2}Net except the backbone part. Therefore, it is not suitable to set an lr shared with the whole network, and the non-pretrained parts should be trained with a larger lr. Moreover, we argue the lr of the Proxy-based module be set to a further larger value, as the network will not be able to complete the Pull-in work well if the sync update of the proxies is lagging. The experimental results are displayed in Table IX. As we can see, when we set the lr of the Proxy-based module and the other modules to 10 and 5 times of backbone's, we achieve the best result.

I. Visualization

To further understand the working mechanism behind our P^{2}Net, we visualize the feature distribution of the input images in Fig. 7 using the t-SNE method [62]. From figure (b), we can see when only the Push-out part is applied, images are separated successfully from each other. When the classifier is also applied, the separated images are pulled into the corresponding subclass, which is shown in figure (c). However, since the last linear layer does not have enough pull power, the final aggregation result is not as good as the baseline ResNet50 in figure (a). Then, we design a much powerful Pull-in part. From figure (d), we can see when the Pull-in part and classifier are combined, the images of each subclass gather much closer. After adding the Push-out part, it can be seen from the last two figures (d) and (e) that the several subclasses, which are originally concentrated, are now separated. Compared with the result of the baseline ResNet50 in figure (a), we can see that subclasses are much farther away from each other, while the images of the same subclass get much closer.

J. Effect on Challenge of Imbalanced Samples

As our method mainly focuses on the "imbalanced fineness" and "imbalanced appearances" issues, in the previous experiments, we try to remove the effect of the imbalanced samples in the FGSC-23 and FGSCR-42 datasets by augmenting the categories with few samples as mentioned in Section IV-B.

Here, we further explore whether our method can benefit the issue of imbalanced samples. In Table X, we display the comparisons of the performance with other methods on the original FGSC-23 and FGSCR-42 datasets, that is, without pre-augmentations. From the results, we can see that the P^{2}Net keeps outperforming other methods when facing imbalanced datasets. In the experiments on the original FGSC dataset, all the methods encounter a reduction of accuracy compared with the results in TABLE IV-D. Compared with the big drop of other methods, the P^{2}Net has just a bit reduction and still achieves high accuracy. While on the original FGSCR-42 dataset, we can observe a general increase of accuracy except for the ProtoTree method (it may verify the direct augmentations are not that suitable for alleviating the effect of the imbalanced samples), and the P^{2}Net has a relatively big boost (about 1%), still better than the other methods. Detailed classification results on each category are further displayed

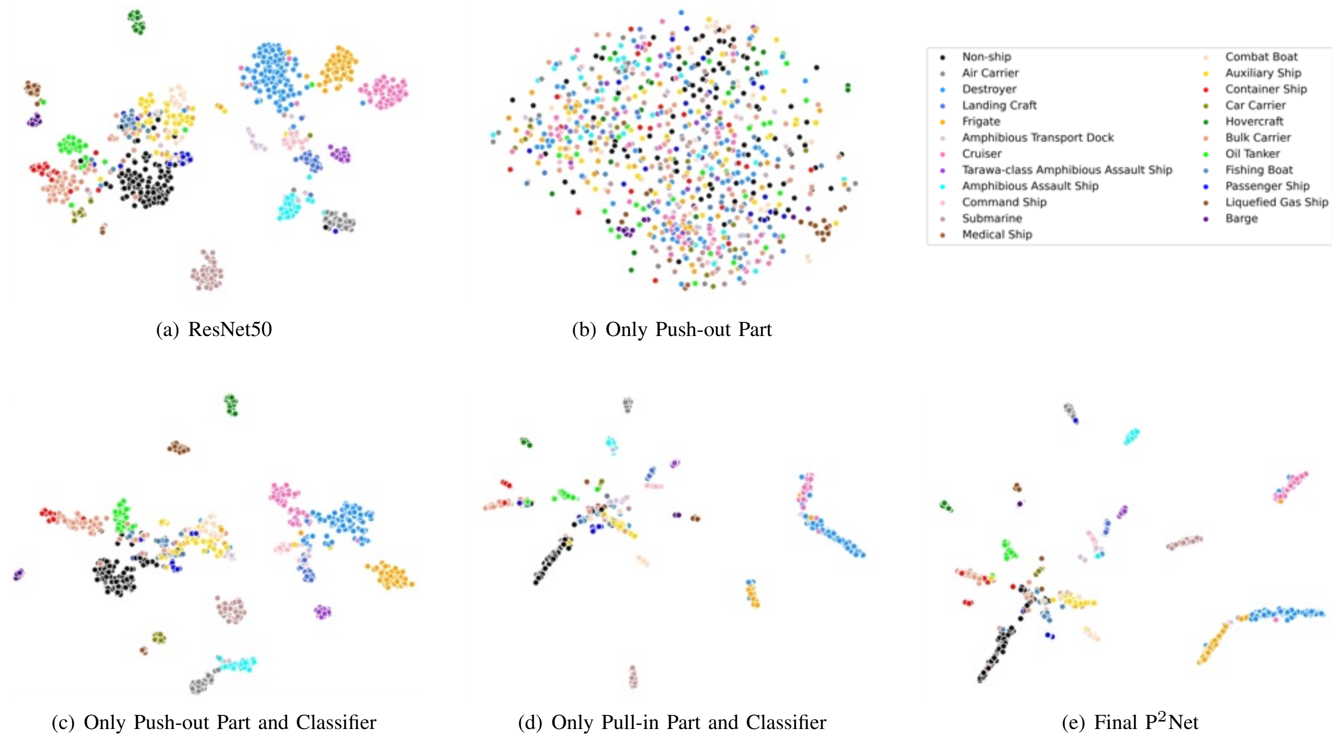


Fig. 7. The feature distribution when applied different parts of P^{2}Net on FGSC-23 dataset. Different colors represent different subclasses.

TABLE X
COMPARISON RESULTS OF DIFFERENT APPROACHES ON THE ORIGINAL FGSC-23 AND FGSCR-42 DATASETS. THE DATASETS ARE NOT PRE-AUGMENTED. THE BEST RESULTS ARE MARKED IN BOLD.

Method	AA	
	FGSC-23	FGSCR-42
ResNet50	85.68	91.85
HBPNet (ECCV 18 [13])	86.09	92.09
DCL (CVPR 19 [56])	84.31	90.65
TASN (CVPR 19 [19])	86.11	92.87
GFNet (NIPS 20 [21])	85.37	92.85
API-Net (AAAI 20 [24])	85.32	91.92
ProtoTree (CVPR 21 [57])	80.46	79.14
P^{2}Net (ours)	**88.56**	**94.19**

in Fig. 8, where we can observe that the CM of the P^{2}Net has fewer and darker valued regions off the diagonal, which corresponds to a smaller MR to other classes and verifies the effectiveness of our method.

V. CONCLUSION

In this paper, we propose a new method called P^{2}Net for the FGSC task. We define two challenging phenomena in the FGSC task, "imbalanced fineness" and "imbalanced appearances", which may cause many difficulties for current state-of-the-art methods. Different from existing methods that perform synchronous discriminative learning, our method introduces an asynchronous push-and-pull strategy where input images are first de-correlated with each other and then aggregated into subclasses. The proposed P^{2}Net leverages the idea of CL and consists of a Push-out part and a Pull-in part that perform the above two processes respectively. With the two parts, our network can decouple the subclasses, and thus make the classification much easier. All network components in our method do not require extra annotations and can be trained in an end-to-end fashion. Extensive experiments on two public datasets (FGSC-23 and FGSCR-42) demonstrate the superiority of our method.

REFERENCES

[1] Z. Zou and Z. Shi, "Ship detection in spaceborne optical image with svd networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 5832–5845, 2016.

[2] L. Li, Z. Zhou, B. Wang, L. Miao, and H. Zong, "A novel cnn-based method for accurate ship detection in hr optical remote sensing images via rotated bounding box," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 686–699, 2021.

[3] X. Zhang, G. Wang, P. Zhu, T. Zhang, C. Li, and L. Jiao, "Grs-det: An anchor-free rotation ship detector based on gaussian-mask in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 3518–3531, 2021.

[4] Q. Oliveau and H. Sahbi, "Learning attribute representations for remote sensing ship category classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2830–2840, 2017.

[5] Q. Shi, W. Li, and R. Tao, "2d-dfrft based deep network for ship classification in remote sensing imagery," in *2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*. IEEE, 2018, pp. 1–5.

[6] Q. Shi, W. Li, R. Tao, X. Sun, and L. Gao, "Ship classification based on multifeature ensemble with convolutional neural network," *Remote Sensing*, vol. 11, no. 4, p. 419, 2019.

[7] J. Shi, Z. Jiang, and H. Zhang, "Few-shot ship classification in optical remote sensing images using nearest neighbor prototype representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3581–3590, 2021.

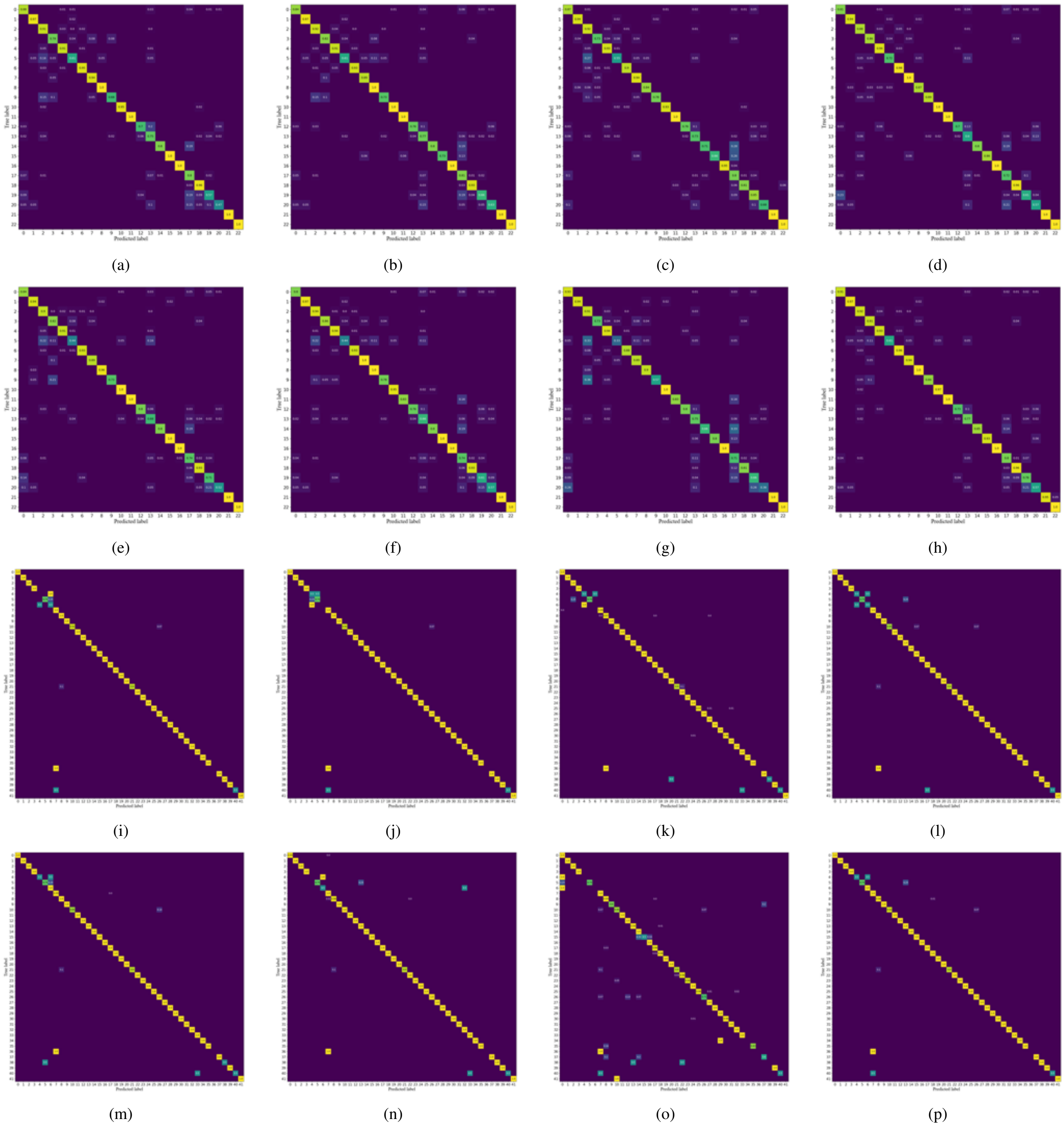


Fig. 8. Confusion matrixes of different methods on the original FGSC-23 and FGSCR-42 datasets without pre-augmentations. The horizontal and vertical coordinates are the category index (See details in Fig. 5). (a)-(h) are respectively the results of ResNet50, HBPNet, DCL, TASN, GFNet, API-Net, ProtoTree, and our P^2Net on the original FGSC-23 dataset. While (i)-(p) are the results on the original FGSCR-42 dataset. For brevity, we filter out the values of zero in CM.

[8] Q. Xiao, B. Liu, Z. Li, W. Ni, Z. Yang, and L. Li, "Progressive data augmentation method for remote sensing ship image classification based on imaging simulation system and neural style transfer," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9176–9186, 2021.

[9] X. Zhang, Y. Lv, L. Yao, W. Xiong, and C. Fu, "A new benchmark and an attribute-guided multilevel feature representation network for fine-grained ship classification in optical remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1271–1285, 2020.

[10] Y. Di, Z. Jiang, and H. Zhang, "A public dataset for fine-grained ship classification in optical remote sensing images," *Remote Sensing*, vol. 13, no. 4, p. 747, 2021.

[11] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1449–1457.

[12] S. Cai, W. Zuo, and L. Zhang, "Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 511–520.

[13] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 574–589.

[14] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1713–1720.

[15] C. Goring, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2489–2496.

[16] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[17] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M. N. Do, "Weakly supervised fine-grained categorization with part-based image representation," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1713–1725, 2016.

[18] J. Han, X. Yao, G. Cheng, X. Feng, and D. Xu, "P-cnn: Part-based convolutional neural networks for fine-grained visual categorization," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[19] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5012–5021.

[20] W. Ge, X. Lin, and Y. Yu, "Weakly supervised complementary parts models for fine-grained image classification from the bottom up," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3034–3043.

[21] Y. Wang, K. Lv, R. Huang, S. Song, L. Yang, and G. Huang, "Glance and focus: a dynamic approach to reducing spatial redundancy in image classification," *arXiv preprint arXiv:2010.05300*, 2020.

[22] Y. Ding, Z. Ma, S. Wen, J. Xie, D. Chang, Z. Si, M. Wu, and H. Ling, "Ap-cnn: weakly supervised attention pyramid convolutional neural network for fine-grained visual classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 2826–2836, 2021.

[23] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 805–821.

[24] P. Zhuang, Y. Wang, and Y. Qiao, "Learning attentive pairwise interaction for fine-grained classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 130–13 137.

[25] F. Xu, M. Wang, W. Zhang, Y. Cheng, and W. Chu, "Discrimination-aware mechanism for fine-grained representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 813–822.

[26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[27] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[28] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 360–368.

[29] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3238–3247.

[30] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, "Softtriple loss: Deep metric learning without triplet sampling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6450–6458.

[31] G. Sumbul, R. G. Cinbis, and S. Aksoy, "Fine-grained object recognition and zero-shot learning in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 770–779, 2017.

[32] J. Ni, Y. Jia, Q. Yin, Y. Zhou, and F. Zhang, "Metric learning based fine-grained classification for polsar imagery," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 716–719.

[33] G. Sumbul, R. G. Cinbis, and S. Aksoy, "Multisource region attention network for fine-grained object recognition in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4929–4937, 2019.

[34] Z. He and D. He, "Bilinear squeeze-and-excitation network for fine-grained classification of tree species," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 7, pp. 1139–1143, 2020.

[35] L. Bai, Q. Liu, C. Li, Z. Ye, M. Hui, and X. Jia, "Remote sensing image scene classification using multiscale feature fusion covariance network with octave convolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[36] Y. Nie, C. Bian, and L. Li, "Adap-emd: Adaptive emd for aircraft fine-grained classification in remote sensing," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[37] W. Zhao, T. Tong, L. Yao, Y. Liu, C. Xu, Y. He, and H. Lu, "Feature balance for fine-grained object classification in aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[38] J. Chen and Y. Qian, "Hierarchical multilabel ship classification in remote sensing images using label relation graphs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.

[39] Y. Li and C. Bian, "Few-shot fine-grained ship classification with a foreground-aware feature map reconstruction network," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[40] W. Xiong, Z. Xiong, and Y. Cui, "An explainable attention network for fine-grained ship classification using remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[41] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," *arXiv preprint arXiv:1406.2952*, 2014.

[42] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1487–1500, 2017.

[43] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[44] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[45] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1920–1929.

[46] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *arXiv preprint arXiv:2006.09882*, 2020.

[47] X. Zhan, J. Xie, Z. Liu, Y.-S. Ong, and C. C. Loy, "Online deep clustering for unsupervised representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6688–6697.

[48] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019.

[49] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[50] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[51] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[53] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[54] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

[55] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3974–3983.

[56] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5157–5166.

[57] M. Nauta, R. van Bree, and C. Seifert, "Neural prototype trees for interpretable fine-grained image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 933–14 943.

[58] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[59] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[60] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[61] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[62] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

Jianqi Chen received his B.S. degree from the Image Processing Center, School of Astronautics, Beihang University in 2021. He is currently pursuing his M.S. degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include deep learning, object detection and artificial intelligence safety.

Keyan Chen received the B.S. degree from the School of Astronautics, Beihang University, Beijing, China, in 2019. He is currently working toward the M.S. degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include image processing, machine learning and pattern recognition.

Hao Chen received his B.S. degree from the Image Processing Center School of Astronautics, Beihang University in 2017. He is currently pursuing his doctorate degree in the Image Processing Center, School of Astronautics, Beihang University.His research interests include machine learning, deep learning and semantic segmentation.

Wenyuan Li received his B.S. degree from North China Electric Power University, Beijing, China in 2017. He is currently working toward his doctorate degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests include self-supervised learning and remote sensing image processing.

Zhengxia Zou received his BS degree and his Ph.D. degree from Beihang University in 2013 and 2018. He is currently an Associate Professor at the School of Astronautics, Beihang University. During 2018-2021, he was a postdoc research fellow at the University of Michigan, Ann Arbor. His research interests include computer vision and related problems in remote sensing. He has published more than 20 peer-reviewed papers in top-tier journals and conferences, including TPAMI, TIP, TGRS, CVPR, ICCV, AAAI. His research was featured in more than 30 global tech media and was adopted by a number of application platforms with over 50 million users worldwide. His personal website is https://zhengxiazou.github.io/.

Zhenwei Shi (Member, IEEE) received the Ph.D. degree in mathematics from the Dalian University of Technology, Dalian, China, in 2005.

He was a Post-Doctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, from 2013 to 2014. He is currently a Professor and the Dean of the Image Processing Center, School of Astronautics, Beihang University, Beijing. He has authored or coauthored over 200 scientific papers in refereed journals and proceedings, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Geoscience and Remote Sensing Letters, the IEEE Conference on Computer Vision and Pattern Recognition, and the IEEE International Conference on Computer Vision. His research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Dr. Shi also serves as an Editor for the Pattern Recognition, the ISPRS Journal of Photogrammetry and Remote Sensing, the Infrared Physics and Technology, and so on. His personal website is http://levir.buaa.edu.cn/.