Contrastive Learning-Based Heterogeneous Network for PolSAR Land Cover Classification

Jianfeng Cai, Zhixi Feng, *Member, IEEE*, Shuyuan Yang, *Senior Member, IEEE*

***Abstract*—Polarimetric synthetic aperture radar (PolSAR) image interpretation is widely used in various fields. Recently, deep learning has made significant progress in PolSAR image classification. However, supervised learning (SL) requires a large amount of labeled PolSAR data with high quality to achieve better performance, and manually labeled data is insufficient. This causes the SL to fail into overfitting and degrades its generalization performance. Furthermore, many features in PolSAR can be used to improve method performance. To solve these problems, this article proposes a Heterogeneous Network based on Contrastive Learning (HCLNet). HCLNet aims to learn high-level representation from unlabeled PolSAR data for few-shot classification according to multi-features. It introduces the heterogeneous network for the first time to utilize different PolSAR features better. Beyond the conventional CL, HCLNet develops two easy-to-use plugins to narrow the domain gap between optics and PolSAR, including beam search and superpixel-based instance discrimination. The pre-trained online network is used for the downstream task by fine-tuning. Experiments demonstrate the superiority of HCLNet on three widely used PolSAR benchmark data sets compared with state-of-the-art methods on few-shot classification. Ablation studies also verify the importance of each component. Besides, this work has implications for how to efficiently utilize the multi-features of PolSAR data to learn better high-level representation in CL and how to construct networks suitable for PolSAR data better.**

***Index Terms*—Contrastive learning (CL), polarimetric synthetic aperture radar (PolSAR) image classification, few-shot learning, superpixel, beam search**

I. Introduction

Polarimetric synthetic aperture radar (PolSAR), an active remote sensing technology, has attracted significant attention due to its ability to obtain richer information than conventional single-polarization synthetic aperture radar (SAR). By using different polarimetric combinations of transmitting and receiving backscattering waves from land covers, PolSAR can observe targets in all-weather and all-time. Therefore, PolSAR image classification [1] [2], which is the most crucial task in PolSAR image interpretation, has been widely used in various fields such as geography [3], agriculture [4], and environmental monitoring [5].

This work was supported by the National Natural Science Foundation of China (Nos. 62171357, 62276205); the Foundation of Key Laboratory of Aerospace Science and Industry Group of CASIC, China; the Foundation of Intelligent Decision and Cognitive Innovation Center of State Administration of Science, Technology and Industry for National Defense, China; the Key Project of Hubei Provincial Natural Science Foundation under Grant 2020CFA001, China. (Corresponding author: Zhixi Feng, Shuyuan Yang.)

All the authors are with the School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: jfcai_1@stu.xidian.edu.cn; zxfeng@xidian.edu.cn; syyang@xidian.edu.cn).

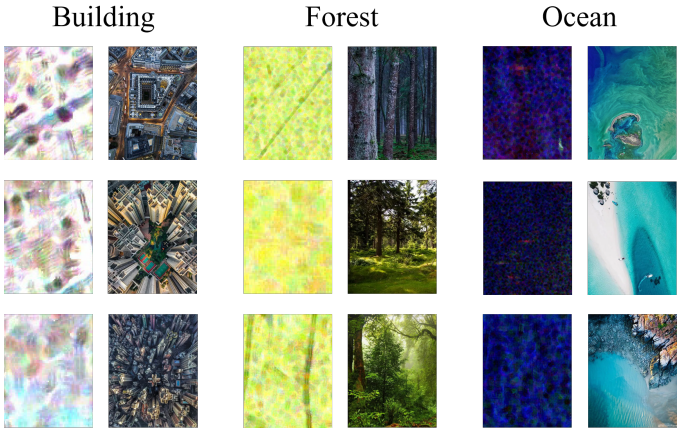


Fig. 1. Visual comparison of instance similarity between PolSAR and optical images, with PolSAR images on the left and optical images on the right.

Numerous researchers have proposed PolSAR classification methods using hand-crafted features. Two primary categories for classifying these features are inherent physical scattering and statistical features. The former is mainly based on target decomposition mechanisms: Freeman decomposition [6] decomposes the pixel into three scattering categories; H/A/α decomposition [7] obtains entropy, anisotropy, alpha angle, and other decomposition methods, including Pauli decomposition [8], Huynen decomposition [9], Cameron decomposition [10], and Krogager decomposition [11]. The latter mainly consists of the coherency and covariance matrix, which follow the complex Wishart distribution. These methods use classifiers such as SVM [12] and MLP [13] to classify PolSAR data. However, the performance of these methods is heavily dependent on the quality of the features, and none of them can fully represent PolSAR data.

Recently, deep learning (DL) methods, especially convolutional neural networks (CNNs), have achieved magnificent success in various fields, including optical image [14]–[16], natural language processing (NLP) [17]–[19], and remote sensing image [20]–[22]. Due to the remarkable results of DL, many researchers have proposed several PolSAR deep learning methods. Zhou et al. [23] first used a CNN to replace conventional methods and achieved breakthrough results. To better adapt to the data structure of PolSAR, complex-valued CNNs (CV-CNNs) [24] were proposed. Subsequently, Wishart deep belief networks (WDBNs) [25], fully convolutional networks [26], and 3D convolution-based networks [27] have been proposed. While supervised CNN-based methods have achieved promising performance, they require a large labeled

training set, which is a significant expense of time and energy. When labeled data are scarce, the trained network can easily result in overfitting, leading to a lack of generalization. It means that supervised learning methods lack robustness in the case of missing labeled data, even with augmentation and regularization techniques [28]–[30].

In contrast to supervised learning, self-supervised learning (SSL) [31], where the data provides supervision, has the advantage of learning general representation from unlabeled data, which is more desirable and meaningful. Contrastive learning (CL) is the popular SSL method in optical images, which constructs simple and easy-to-use frameworks for training. From InstDisc [32], InvaSpread [33], CPC [34], and CMC [35] to MoCo [36], SimCLR [37], BYOL [38], and SimSiam [39], CL has a relatively mature architecture. However, the gap between PolSAR and optical images makes applying optical CL methods to PolSAR directly tricky. Researchers have proposed some PolSAR-tailored CL methods to address this issue: MI-SSL [40] learned the implicit multi-modal representation from unlabeled data. PCLNet [41] developed an instance discrimination proxy objective to learn representation from unlabeled data. SSPRL [42] improves CL, so no negative samples are needed; Cui et al. [43] proposed TCSPANet, the two-staged CL based on attention. However, there are still some challenges with these methods:

- Most of them fail to utilize the multi-features in PolSAR data fully. The diversity of PolSAR features makes it more advantageous in CL and can extract high-level representation through different features. However, as a standard architecture for CL of optical images, they directly utilize the Siamese network to PolSAR, making it hard to exploit these features thoroughly. In contrast, the heterogeneous network can arbitrarily combine different features for better representation learning.
- Some methods attempt to incorporate different features but fail to consider the redundancy between them. The degree of information redundancy varies for different feature combinations, and some features may even hinder model learning. Therefore, feature selection is essential.
- Some of them ignored the high similarity between pixels in PolSAR data. As shown in Fig. 1, generally speaking, in optical images, instances are images with thousands of pixels, and even images of the same class have very different pixel values among themselves. However, in PolSAR, the instance is only the scattering matrix of each pixel. Even the pixels of different classes have certain similarity in the scattering matrix. This makes a good contrastive learning method framework that works in the optical image struggle to differentiate between different PolSAR instances. To overcome this limitation, introducing diversity between instances is necessary to enhance model learning.
- All of their methods do not consider the scattering confusion problem. Due to the scattering mechanism and ground cover types, there are often classification errors between different land covers of PolSAR, that is, the scattering information between two land covers is strongly correlated, so the model cannot classify them well. To solve this problem, the model needs to learn the scattering dissimilarity between different land covers. This requires the model to learn between different scattering features and a large number of different instances.

Based on the preceding analysis, this article introduces a novel approach to PolSAR data classification, named Heterogeneous Contrastive Learning Network (HCLNet). The proposed HCLNet employs two perspectives, physical and statistical, for CL. From the physical perspective, it uses beam search to reduce feature redundancy, while from the statistical perspective, it utilizes the coherency matrix. Additionally, it constructs a heterogeneous network to learn the representation of PolSAR data using the novel superpixel-based Instance Discrimination. This approach effectively utilizes the PolSAR multi-features and addresses the problem of pixel similarity and scattering confusion. Furthermore, it uses two easy-to-use plugins to better adapt to PolSAR data. Specifically, the main contributions of this work can be summarized as follows:

- the *Heterogeneous Network* is proposed to learn the representation of PolSAR data using CL for the first time to effectively addresses the challenge of scattering confusion. The network consists of two sub-networks with different architectures, namely the online network and the target network, where the former is a 2D CNN, and the latter is a 1D CNN. This network can input different features for learning and extract high-level representations hidden between multi-features of PolSAR data without the need for labeled data.
- A novel pretext task, *Superpixel-based Instance Discrimination*, is designed to reduce the similarity between pixels and thus the model can learn representation easier and better. This task utilizes superpixel segmentation to select positive and negative samples for CL, which reduces the occurrence of highly similar pixels being negative samples of each other.
- *Beam search* is utilized to select complementary features and reduce redundancy. It created a classifier as the foundation for beam search, which implements a suitable combination of these features and removes redundant information from multi-features.
- Experimental results demonstrate the superiority of the proposed method. HCLNet is applied to three benchmark PolSAR data sets, and the results indicate that it achieves state-of-the-art classification algorithm, whether in few-shot or full-sample scenarios. Our implementation are given on GitHub [1].

The rest of this article is organized as follows: A brief review of the CL for optical images and PolSAR and PolSAR multi-features are given in Section II. The details of the proposed HCLNet are described in Section III. Section IV shows the experimental results and analysis. Finally, we provide the conclusion and prospects for future research in Section V.

[1]https://github.com/cai-jianfeng/HCLNet

II. Related Work

A. Contrastive Learning

As a prevalent SSL method, CL has a mature and general architecture. It usually has two networks with the same architecture; one is called the online network, which will be sent to downstream tasks for fine-tuning as the main network, and the other is called the target network. They can share parameters. Generally, CL is trained by a proxy objective, called pretext task, and usually chooses Instance Discrimination with InfoNCE loss function. Its main idea is that two representation obtained by the two networks from different perspectives of the same image should be similar and vice versa. InfoNCE loss function is a modified Cross-Entropy Loss that introduces the dot product to compute similarity. The formula is as follows:

$$L(q,k) = -log\frac{exp(q \times k_+/\tau)}{\sum_{i=0}^{K} exp(q \times k_i/\tau)} \tag{1}$$

where q is the output of the online network, k_+ is the output of the target network that q matches and is called the positive sample, k_i ($i = 0 \cdots K$ and $k_i \neq k_+$) is all output of the target network, which is memoried and is called the negative sample, τ is a temperature hyper-parameter that adjusts the uniformity of information distribution [32]. Intuitively, this loss tries to classify q to k_+, and essentially, it is the log loss of a $(K+1)$-way softmax-based classifier.

1) The CL in Optical Images: In optical images, according to the different ways of parameter updating and negative sample selection, CL methods can be divided into different types. Wu et al. [32] first proposed Instance Discrimination with NCE loss function and memory bank to store negative samples. It treats the two sub-images obtained from a cropped image as the positive sample and all other images in the data set as negative samples. While Ye et al. [33] select other samples in the same mini-batch as negative samples. CPC [34] is a more general architecture containing an encoder and an auto-regression model. Positive and negative samples are constructed to train the encoder autoregressively. In addition to cropping, optical images have properties such as depth that can also form positive samples with each other. So Tian et al. [35] proposed CMC using luminance (L channel), chrominance (ab channel) [44], depth, surface normal [45], and semantic labels to construct the positive sample. To make more negative samples and sample representation change smoothly, MoCo [36] introduces a dictionary named queue to increase negative samples and momentum update to update the parameter of the target network. SimCLR [37] uses a larger batch size to achieve better results. It also adds a projection head to the network's end and achieves incredible performance. To eliminate negative samples, BYOL [38] adds a prediction head to the end of the online network, then turns the similarity problem into a prediction problem, which can effectively prevent the model collapse. Chen et al. [39] summarizes the previous work and proposes a simple architecture, SimSiam, which demonstrates the importance of stop gradient.

2) The CL in PolSAR: To use CL based on optical images in the PolSAR domain, some researchers proposed PolSAR-tailored CL methods: MI-SSL [40] uses coherency matrix T and constructs positive samples by visual, physical, and statistical features to learn the implicit multi-modal representation with similarity and difference loss; PCLNet [41] which copy the MoCo technique, selects data set according to stimulation for interclass and intraclass diversity and uses PolSAR image rotating 180 as the positive sample. SSPRL [42] proposes two branches and dynamic convolution (DyConv) layer to improve CL, its basic architecture is similar to BYOL. TCSPANet [43] exploits unsupervised multi-scaled patch-level data sets (UsMsPD) and semi-supervised multi-scaled patch-level data sets (SsMsPD). It also proposes two CL stages in TCNet and adds attention mechanism in SPAE to get better results.

In this article, inspired by SimCLR, we construct a heterogeneous CL network based on superpixel-based instance discrimination, which selects appropriate negative samples to reduce the high similarity between positive and negative samples.

B. Multi-features within PolSAR

1) Physical scattering features: As mentioned in [46], most physical scattering features are based on target decomposition. Different target decompositions, which decomposes target based on Scattering matrix S, have distinct advantages for PolSAR image classification. For example, Freeman decomposition [6] decomposes the scattering matrix S into three scattering categories: surface, volume, and double bounce; entropy, anisotropy, and alpha angle are obtained by H/A/α decomposition [8]; Krogager decomposition [11] decomposed the scattering matrix S into sphere, diplane, and helix components. Other decomposition methods include Yamaguchi, Vanzyl, Neuman, Multiple-Component Scattering Model (MCSM), Huynen, Holm, Barnes, Cloude, Anned, An-Yang, Pauli decomposition, Huynen decomposition, and Cameron decomposition [8] [9] [10] [47]–[51].

2) Statistical features: According to the statistical characteristics, PolSAR data can become the coherency matrix T and the covariance matrix C based on Scattering matrix S, which follows the complex Wishart distance. Scattering matrix S is defined as

$$S = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \tag{2}$$

where S_{XY},X,$Y \in [H,V]$ is the scattering element of horizontal/vertical transmitting/receiving polarization. The covariance matrix C is formed by

$$h = \begin{bmatrix} S_{HH} & \sqrt{2}S_{HV} & S_{VV} \end{bmatrix}^T \tag{3}$$

$$C = hh^{*T} = \begin{bmatrix} |S_{HH}|^2 & \sqrt{2}S_{HH}{S_{HV}}^* & S_{HH}{S_{VV}}^* \\ \sqrt{2}S_{HV}{S_{HH}}^* & 2\,|S_{HV}|^2 & \sqrt{2}S_{HV}{S_{VV}}^* \\ S_{VV}{S_{HH}}^* & \sqrt{2}S_{VV}{S_{HV}}^* & |S_{HV}|^2 \end{bmatrix} \tag{4}$$

where the superscript "T" denotes the conjugate transpose. The coherent matrix T is formed by

$$k_p = \left[\begin{array}{ccc} (S_{HH} + S_{VV})/\sqrt{2} & (S_{HH} - S_{VV})/\sqrt{2} & \sqrt{2}S_{HV} \end{array} \right]^T \quad (5)$$

$$T = < k_p k_p^T > = \left[\begin{array}{ccc} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{array} \right] \quad (6)$$

As aforementioned, these features have different importances in specific scenes. Some features have functional overlap, which leads to feature redundancy. So combining these features properly to learn representation better is necessary. We propose the beam search to obtain a better complementary feature combination that is similar to [46].

3) Feature Selection: Feature selection is an important problem in PolSAR. There are certain complementarity and redundancy between different polarized features, and a good feature selection method can well preserve the complementarity between each feature and eliminate the redundancy. Haddadi G et al. [52] proposed the PolSAR feature selection method using a genetic algorithm. The method used a combination of a genetic algorithm (GA) and am artificial neural network (ANN) to learn feature selection. Yang et al. [46] used the Kullback-Leibler distance (KLD) as a standard, and a 1-D CNN as a selection algorithm to select feature subsets. Huang et al. [53] realized multi-view feature selection via manifold regularization and $l_{2,1}$ sparsity regularization, including polarimetric features and texture features. In addition, many studies also introduced the attention mechanism into feature selection. AFS-CNN [54] is proposed to capture the relationship between input polarimetric features through attention-based architecture to ensure the validity of high-dimensional data classification. In contrast, our beam search feature selection method is similar to [46], but we use a simpler and more intuitive feature selection objective to obtain a better combination of features.

III. Heterogeneous Network based on Contrastive Learning

In this section, the detailed process of the proposed HCLNet is presented. The overall architecture of HCLNet is shown in Fig. 2, which includes three main components. Among them, the first component is Beam Search, which finds the appropriate combination of target decomposition features. The second component is Superpixel-based Instacne Discrimination, which improves the general Instance Discrimination selection of positive and negative samples. Moreover, the final component is the Heterogeneous Network, which is the most important one and learns the high-level representation of PolSAR data. First, all target decomposition features are filtered using the beam search. Then the coherent matrix and the filtered target decomposition features are used as the input of the heterogeneous network. The superpixel-based instance discrimination is used for unsupervised training. The specific details of each component are as follows.

A. Beam Search

Here, we will introduce the beam search in detail. Like [46], we extract many features by the target decompositions mentioned above, which have N in total, and M groups; each target decomposition method generates a group of features. Then, we design a 1-D CNN model to evaluate the performance of different combinations of features. The network's input is an $N \times 1$ vector V, representing the N features of a pixel. During training, we use all N features to predict the label of each pixel for supervised learning with least-squares loss function [51].

After training the 1-D CNN as the classifier namely B_c, we use beam search to select the appropriate combination of features and use classifier accuracy as the selection basis. Unlike [46], we do not introduce additional KLD standard to interfere with the 1D-CNN classifier's choice of the combination of features. In this way, the complementarity of the feature groups selected by the model can be guaranteed to the greatest extent and the redundancy between features can be eliminated as much as possible. Specifically, Step 1, we start with the initial M group of features and choose to remove the first k groups of features that cause the slightest reduction in classification accuracy to form k branches, where one group of features is removed from each branch. Step 2, in each branch, the above steps are repeated such that each branch forms k branches, and the total branches are $k \times k$. Step 3, we choose the first k branches, according to the classification accuracy from high to low. Repeating Step 2 and Step 3 until the feature groups number is reduced to the threshold θ. The process of selecting features by beam search is outlined in Algorithm 1.

Algorithm 1: Beam Search for selecting the appropriate combination of feature groups

Input: feature groups set M_i, the number of feature groups N, threshold θ, branch number k, classifier F
Output: selected feature groups set M_o
set $Q = \{M_i\}$
while $N > \theta$ **do**
  $Q' \leftarrow \{\}$
  for *feature groups* M *in* Q **do**
    for *feature* f *in* M **do**
      remove f from M
      if *len*(Q') == k **then**
        if $F(M) \geq max(F(Q'))$ **then**
          pop Q'(max($F(Q')$))
          push M into Q'
        end
      else
        push M into Q'
      end
    end
  end
  N $\leftarrow$ N-1
  $Q \leftarrow Q'$
end

To ensure the unity of the input dimensions of the classifier,

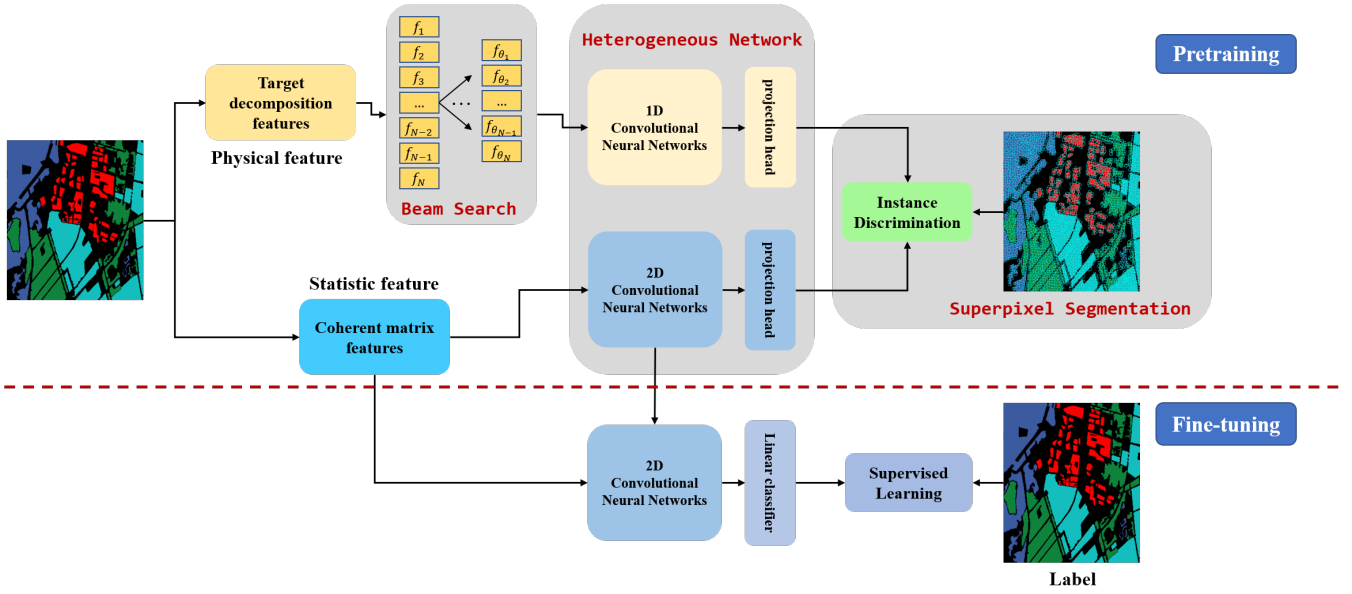


Fig. 2. The overall framework of the proposed HCLNet. It mainly contains two processes: Pretraining and Fine-tuning. In pretraining, it first uses Beam Search to combinate features, then constructs the heterogeneous network and uses Superpixel-based Instance Discrimination to learn the high-level representation. In fine-tuning, it uses the trained online network from pretraining and fine-tunes it with a small number of labeled data to better fit the downstream distribution.

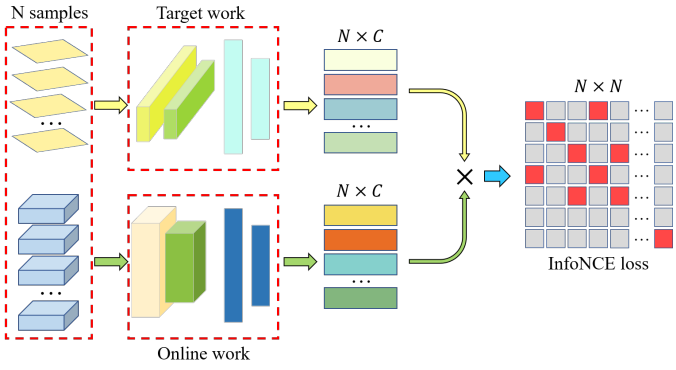


Fig. 3. The architecture of the heterogeneous network in HCLNet. It contains two networks with different architectures and is updated with InfoNCE loss. The output of the target network belonging to different superpixels in the same minibatch will be served as negative samples.

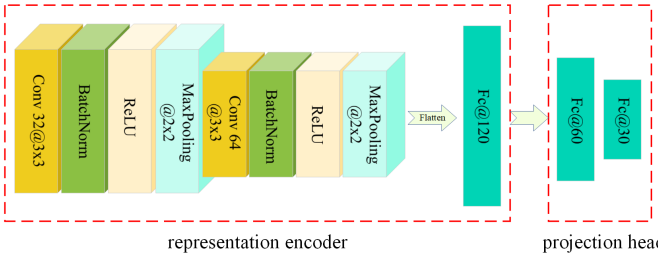


Fig. 4. The architecture of the online network in the heterogeneous network. It contains the representation encoder and the projection head; the former will be used for fine-tuning.

we directly set the value of the features removed each time to 0. Finally, we obtain θ group, θ_N features as the input of the target network, as described in Section III-C.

B. Superpixel-based Instance Discrimination

As mentioned in Section II-A, the general pretext task of the optical images is Instance Discrimination with InfoNCE loss function. The input of CL is usually the whole optical image, with both pixel and semantic features. Significant differences exist between different optical images, even in the same class. So the common Instance Discrimination can perform well. However, PolSAR mainly uses pixels as instances for CL training, which has a considerable similarity between pixels. Therefore, we can no longer treat all other pixels as negative samples of the current pixel. Instead, we should add some prior knowledge to intervene the model to select the pixels with large similarity difference from the original pixels as negative samples. Superpixel segmentation algorithm is a simple and efficient unsupervised algorithm for measuring the similarity between pixels. So to continue to take advantage of Instance Discrimination, we improve the way of selecting negative samples according to superpixel. The details are as follows:

We segment the PolSAR image into some superpixels, as shown in Fig. 5. Specifically, we choose the classical superpixel segmentation method: Simple Linear Iterative Clustering (SLIC) [3] to segment the whole PolSAR images. Compared to the other algorithm, such as turbo or Ncut, it is fast, memory efficient, boundary adherence, and needs to set a few parameters. It used the idea of clustering to cluster part of the pixels into a superpixel. So the similarity of pixels within the same superpixel is high and vice versa. Within the scope of superpixel segmentation, we redefine the positive and negative sample, that is, pixels within different superpixels are defined as negative samples of each other and within the same superpixel as positive samples. And the filtering physical scattering features and statistical features of each pixel remain unchanged. Assumed the size of the PolSAR image is $H \times W$, and we obtain N_s superpixels, N_{s_i} pixels in ith superpixel, the

pixel in ith superpixel has at most $N_{s_i} - 1$ positive sample, $H \times W - N_{s_i}$ negative samples. Then, we can use traditional Instance Discrimination to train the Heterogeneous Network described in Section III-C. In general, for a data sample (that is, a pixel) p_o in a batch B, there are N_+ samples in the same superpixel with p_o and $N_- = B - N_+ - 1$ samples that are not in the same superpixel with p_o. Then the N_+ samples and p_o are positive samples, and the remaining N_- samples are negative samples of p_o. Therefore, its loss function can be rewritten from the form in Section II-A as:

$$L_{p_o}(p_o, \{p_1^+, ..., p_{N_+}^+\}, \{p_1^-, ..., p_{N_-}^-\}) = -\frac{1}{B-1} \times log\frac{\sum_{i=1}^{N_+} exp(p_o \times p_i^+/\tau)}{\sum_{i=1}^{N_+} exp(p_o \times p_i^+/\tau) + \sum_{j=1}^{N_-} exp(p_o \times p_j^-/\tau)} \quad (7)$$

where p_i^+ and p_j^- represent the ith positive and the jth negative samples, respectively. Since all B samples in the batch are selected in turn, the total cost function becomes:

$$L_{HCLNet} = \sum_{i=1}^{B} \frac{L_{p_i}(p_i, \{p_1^+, ..., p_{N_+^i}^+\}, \{p_1^-, ..., p_{N_-^i}^-\})}{B} \quad (8)$$

where N_+^i and N_-^i represent the number of the positive and negative samples of p_i respectively.

C. Heterogeneous Network

Inspired by CLIP [55] in which two networks use different architectures to input different features for fusion learning, PolSAR custom-made Heterogeneous Network is proposed in this article, shown in Fig. 3. It is similar to the traditional CL Network, which has two networks; however, the architecture is different between the two networks. The online network of the heterogeneous network is a 2-D CNN, while the target network of the heterogeneous network is a 1-D CNN. However, different from the traditional CL, each network of Heterogeneous Network inputs different features, which are used to further deepen the model's understanding of the scattering differences of different pixels to solve the scattering confusion problem. The details of the two networks are as follows:

1) Online network: The online network consists of a 2-D CNN with a representation encoder and a projection head, which is similar to the common CL network. Its input is a two-dimensional block of pixels for size $k \times k$, each with the feature value of the he coherent matrix T. Because of its two-dimensional characteristics, it is used to learn the scattering relationship between the current pixel and its neighboring pixels to better learn the scattering similarity between different neighboring pixels. The typical architecture of the representation encoder is shown in Fig. 4. The encoder consists of four main parts: convolution layer, pooling layer, linear embedding layer, and nonlinear activation.

The formulation of convolution is follow

$$y_i^{(l+1)} = \sum_j^J \omega_{ij}^{(l)} \otimes x_j^{(l)} + b_i^{(l+1)} \quad (9)$$

where $x_j^{(l)}$ represents the jth input in the $l+1$th layer and J represents the number of inputs, y_i^{l+} represents the ith output in the l+1th layer, ω represents the kernel matrix and b represents the bias, $\otimes$ represents convolution operation.

The pooling operation is usually a subsampling operation that reduces the input dimension and the computation amount. Moreover, it facilitates the identification of displacement, scaling, and other distortion invariants. It has two common choices: max pooling and average pooling. In this article, we choose max pooling, which selects the maximum value of the specified region.

The linear embedding layer is a linear weighted transformation of the representation in a 1-D space. The formulation is similar to convolution and is as follows

$$y_i^{(l+1)} = \sum_j^J \omega_{ij}^{(l)} * x_j^{(l)} + b_i^{(l+1)} \quad (10)$$

where $x_j^{(l)}$, J, $y_i^{(l+1)}$, ω, b is the same as convolution, $*$ represents vector multiplication.

The nonlinear activation is to improve the nonlinear ability of the network. Avoid using softmax and tanh, which leads to gradient vanishing. We select the ReLU, the nonlinear activation function. The formula of it is as follows

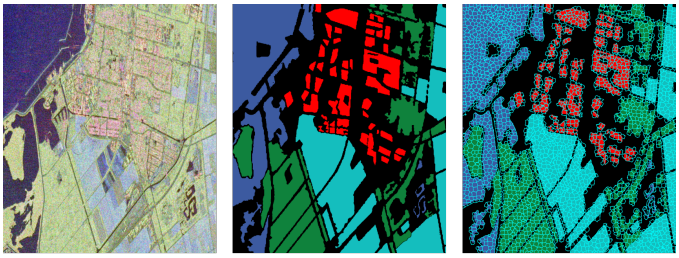
$$f(x) = \max_{kernel}(0, x) \quad (11)$$

where x is the input, $max_{kernel}(\cdot)$ does the kernel size specify the maximum value operation in the region.

By combining the four operations defined above, we obtain the representation encoder $f_e(\cdot)$. Then a projection head is followed by $f_e(\cdot)$, which aims to embed the representation into the more high semantic space and is defined as $g_p(\cdot)$. It is also a linear embedding layer. So we obtain the final online network $g_p(f_e(\cdot))$.

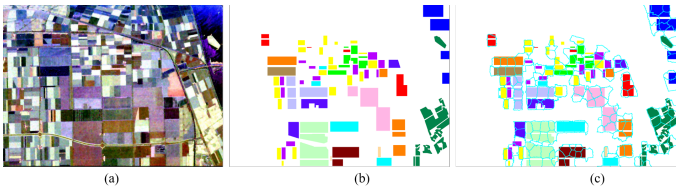
The input of the online network is the coherency matrix, as mentioned in Section II-B. Specifically, for pixel a, we crop out a pixel block of size $k \times k$ with a as the center, and the value of each pixel is represented by the straightened coherency matrix T. Finally, the input dimension is $k \times k \times 9$.

2) Target network: The target network is a 1-D CNN. It's similar in structure to the classifier B_c used in the beam search, but they serve a very different purpose. Its input is the complementary target decomposition multi-features of current pixel filtered by the beam search. By learning the multi-features of different pixels, the model is prompted to learn the scattering difference between each pixel. At the same time, through the combination of online network (scattering similarity), the HCLNet can well learn the scattering differences and relationships between different pixels, which greatly alleviates the problem of scattering confusion. The overall architecture and convolution, pooling, linear embedding operations, and nonlinear activation are similar to 2-D CNN, except they change from two dimensions to one. The input of the target network is the combination of the target decomposition features according to beam search as described in Section III-A. For pixel a, has θ_N features that represent a $\theta_N \times 1$ vector.



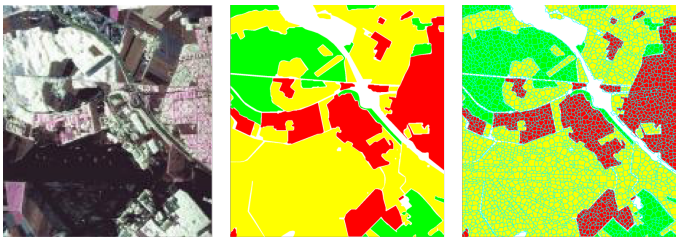
(a) (b) (c)

Fig. 5. RADARSAT-2 Flevoland data. (a) Pauli RGB image. (b) Ground-truth image. (c) Superpixel image.



(a) (b) (c)

Fig. 6. AIRSAR Flevoland data. (a) Pauli RGB image. (b) Ground-truth image. (c) Superpixel image.



(a) (b) (c)

Fig. 7. ESAR Oberpfaffenhofen data. (a) Pauli RGB image. (b) Ground-truth image. (c) Superpixel image.

Finally, for pixel a, the online network inputs the matrix of dimension $k \times k \times 9$, and outputs the $m \times 1$ vector r_o as representation; the target network inputs the vector of dimension $\theta_N \times 1$, and outputs the vector r_{t+} whose dimension is the same as r_o. Then we use the InfoNCE loss function $L(r_o, r)$ to compute the similarity, where $r \in \{r_{t+}, r_{t-}\}$, r_{t-} represents the other target network outputs of the other pixel. After training, we can obtain the final online network to transform PolSAR data into a high-level representation. It can be used as the backbone network for downstream tasks and performs well with fine-tuning.

IV. Experimental Results and Analysis

A. Datasets Description

In this section, we employ three standard PolSAR data sets to verify the superiority of the HCLNet. They include RADARSAT-2 Flevoland, AIRSAR Flevoland, and ESAR Oberpfaffenhofen.

- RADARSAT-2 Flevoland: As shown in Fig. 5. a C-band, fully polarimetric image of the area of Netherland is obtained through the RADARSAT-2 system and was produced in April 2008. The size of the sub-image is 2375 $\times$ 1635. It identifies four types of ground objects: forest, farmland, city, and water area.
- AIRSAR Flevoland: An L-band, full polarimetric PolSAR image of the region of Flevoland, Netherlands, 750 $\times$ 1024, is obtained through the NASA/Jet Propulsion Laboratory AIRSAR. There are 15 labeled objects, including forest, rapeseed, beet, bare soil, grasses, peas, lucerne, barley, buildings, potatoes, water, stembeans, and three kinds of wheat seen in Fig. 6.
- ESAR Oberpfaffenhofen: It covers Oberpfaffenhofen, Germany, which is an L-band, full polarimetric image and is obtained through ESAR airborne platform. The size of the image is 1200 $\times$ 1300. Its ground-truth map can be seen in Fig. 7. It contains three classes: built-up areas, wood land, and open areas.

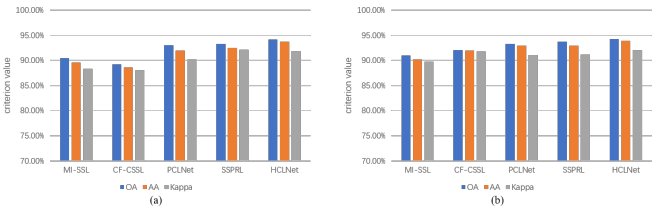


Fig. 8. Comparisons of different methods on the RADARSAT-2 Flevoland data set. (a) Few-shot result. (b) Full-sample result.

TABLE I
ALLOF THE TARGET DECOPOSITION FEATURES

Target Decomposition	Feature Name	Number
Krogager	sphere, diplane, helix	3
TSVM	alpha-s, phi-s, phi, tau-m	4
Neuman	delta, psi, tau	3
Huynen	(T11,T22,T33)dB	3
Holm	Holm1:(T11,T22,T33)dB, Holm2:(T11,T22,T33)dB	6
Freeman	Freeman2:(Vol,Ground)dB, Freeman3:(Odd,Dbl,Vol)dB	5
Cloude	(T11,T22,T33)dB	3
Barnes	Barnes1:(T11, T22, T33)dB, Barnes2:(T11, T22, T33)dB	6
ANNED	(Odd, Dbl, Vol)dB	3
AnYang	AnYang3:(Odd,Dbl,Vol)dB, AnYang4:(Odd, Dbl,Vol, Hlx)dB	7
H/A/α	alpha, anisotropy, beta, delta, entropy, gamma, lambda, combination: HA, (1-H)A, H(1-A), (1-H)(1-A)	11
Yamaguchi	Yamaguchi3:(Odd, Dbl, Vol)dB, Yamaguchi4:(Odd, Dbl, Vol, Hlx)dB	7
Vanzyl	(Odd, Dbl, Vol)dB	3
MCSM	(Odd, Dbl, Vol, Hlx, Dbl-Hlx, Wire)dB	6
SUM		**70**

B. Experimental Settings

- Implement details: The online network, in which the input patch size is 15 $\times$ 15, has two 2-D convolution layers in which the kernel size is 3 $\times$ 3, the padding is 2 $\times$ 2 and 1

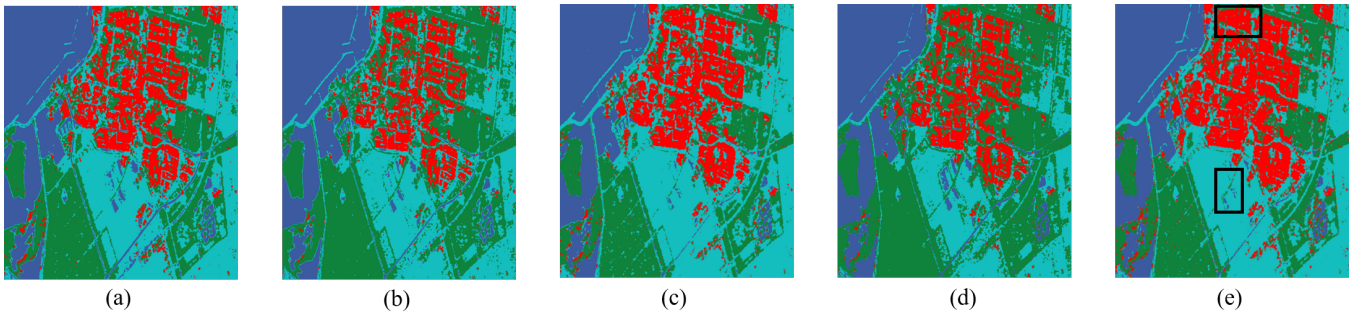


Fig. 9. Few-shot classification results with different methods on the RADARSAT-2 Flevoland data set. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

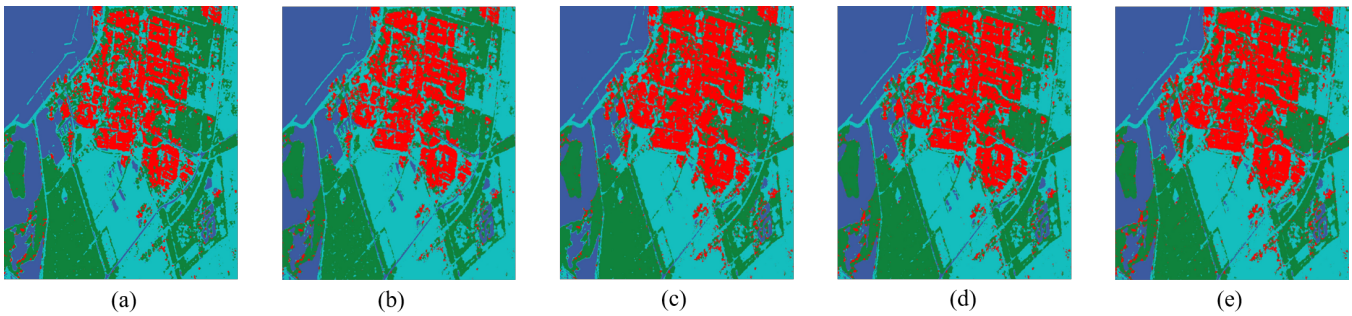


Fig. 10. Full-sample classification results with different methods on the RADARSAT-2 Flevoland data set. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

TABLE II
FEW-SHOT CLASSIFICATION RESULTS (%) ON THE RADARSAT-2 FLEVOLAND WITH DIFFERENT METHODS

method	MI-SSL	CF-CSSL	PCLNet	SSPRL	HCLNet
Forest	79.23	69.70	86.55	72.35	**91.69**
Cropland	98.78	98.33	99.53	**99.21**	99.69
Water	93.02	96.75	92.06	**98.36**	93.87
Urban	91.10	85.35	**95.60**	77.10	94.85
OA	90.42	89.23	93.01	93.25	**94.15**
AA	89.54	88.64	92.00	92.50	**93.74**
Kappa	88.37	88.05	90.20	92.15	**91.84**

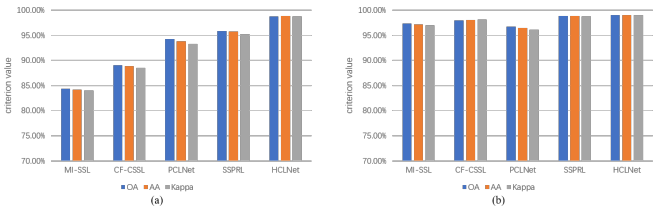


Fig. 11. Comparisons of different methods on the AIRSAR Flevoland data set. (a) Few-shot result. (b) Full-sample result.

$\times$ 1, and two linear embedding layers. The target network has two 1-D convolution layers in which the kernel size is the same as the online network, and the padding is 2 $\times$ 2, and one linear embedding layer. After each convolution layer, the batch norm and max pooling layers are added. Furthermore, the same as MoCo, we use normalization in the outputs of both networks. The optimizer is the SGD which the learning rate is initialized to 0.01 and decreases with a cosine trend with training epoch, the momentum is 0.9, and weight decay is 0.0001. The τ is 0.07. The network is trained for 30 epochs and the minibatch size of 4096. The threshold θ in the beam search is set to 8, and the k is set to 2 in the first three rounds of search and 1 in the subsequent searches. We initialize the number K of superpixels based on the size of each superpixel being roughly 30×30. So the parameter K for the three datasets are 1746, 863, and 1728, respectively. The search range of the center of each superpixel is 3×3. All experiments were conducted independently on a single GeForce 3070 GPU with the PyTorch library.

- Multi-features: The Refined Lee filter with the window size 7 $\times$ 7 is used to preprocess the three PolSAR data sets to reduce the influence of speckles on the result of classification. Then the same as [46], we use 14 groups of target decomposition features and obtain 70 decompositions features as the initial features of the Beam Search.

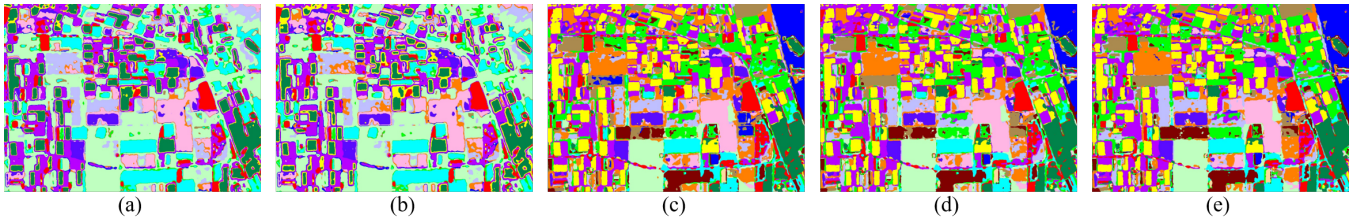


Fig. 12. Few-shot classification results with different methods on the AIRSAR Flevoland data set. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

Fig. 13. Full-sample classification results with different methods on the AIRSAR Flevoland data set. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

TABLE III
FULL-SAMPLE CLASSIFICATION RESULTS (%) ON THE RADARSAT-2 FLEVOLAND WITH DIFFERENT METHODS

method	MI-SSL	CF-CSSL	PCLNet	SSPRL	HCLNet
Forest	82.54	85.03	90.33	86.62	**90.34**
Cropland	97.98	97.28	97.90	**98.68**	98.36
Water	92.37	93.04	94.37	**95.15**	94.24
Urban	91.04	92.11	90.56	91.38	**92.71**
OA	90.95	92.05	93.29	93.70	**94.30**
AA	90.23	91.97	92.91	92.96	**93.91**
Kappa	89.72	91.80	91.06	91.17	**92.04**

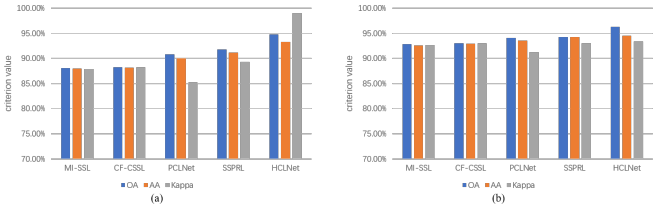


Fig. 14. Comparisons of different methods on the ESAR Oberpfaffenhofen data set. (a) Few-shot result. (b) Full-sample result.

The whole target decompositions are shown in Table I. These features are sufficient to represent PolSAR data.

- Compared method: To evaluate the superiority of the proposed method, we select several supervised and CL methods for comparison. Specifically, two traditional classification methods are chosen, including MI-SSL [40], Coarse-to-Fine CSSL (CF-CSSL) [56] PCLNet [41], and SSPRL [42].

C. Experimental Results and Analysis

1) Classification accuracy: In the experiment, our method is pre-trained with 10% unlabeled data in each data set. To verify the effectiveness of HCLNet, we choose 0.1% and 10% samples per category for training, denoted as few-shot and full-sample classifications. The rest of the samples are used as the test set for evaluation. The overall classification accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa) are used as criteria to assess the performance of all methods.

The results of the three data sets are shown in Tables II to VI, respectively, demonstrating the superiority of HCLNet in the small number of labeled data. Different cases will generally have different results, but the trend is consistent across different data sets.

RADARSAT-2 Flevoland: Specifically, in Tables II and III, for RADARSAT-2 Flevoland, MI-SSL is relatively stable, dropping only 0.53%, 0.69%, and 1.35% from full-sample classification to few-shot classification, but its overall classification accuracy is not high and only 90.95% in full-sample classification. CF-CSSL performs well in full-sample classification; however, its performance drops sharply, even lower than MI-SSL, when the number of data decreases, that is when few-shot classification. Numerically, from full-sample classification to few-shot classification, the OA, AA, and Kappa of CF-CSSL decrease by 2.82%, 3.33%, and 3.75%. The overall number of network parameters of PCLNet is similar to that of HCLNet, which is 1.5 larger than HCLNet. Through the customized task and positive/negative sample selection of PCLNet, its accuracy is much improved compared with CF-CSSL, especially in few-shot classification, in which the

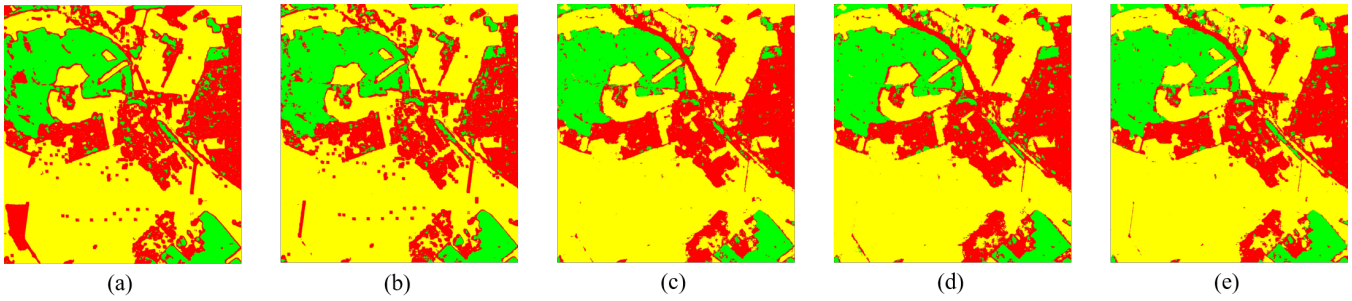


Fig. 15. Few-shot classification results with different methods on the ESAR Oberpfaffenhofen data set. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

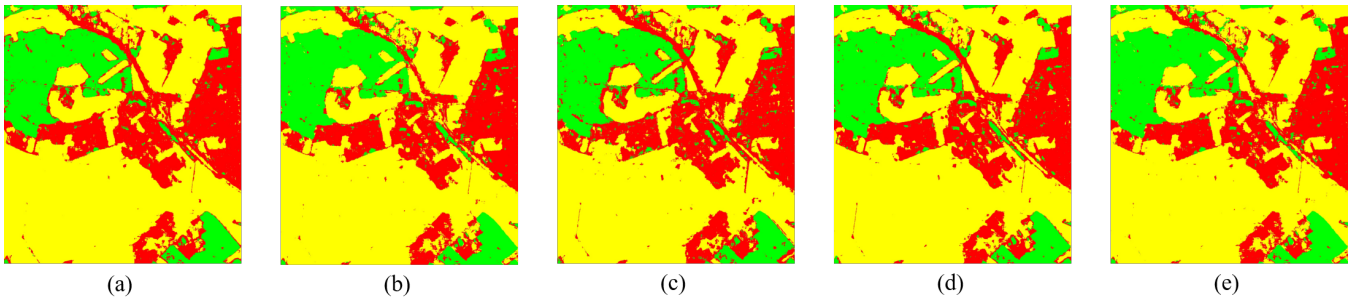


Fig. 16. Full-sample classification results with different methods on the ESAR Oberpfaffenhofen data set. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

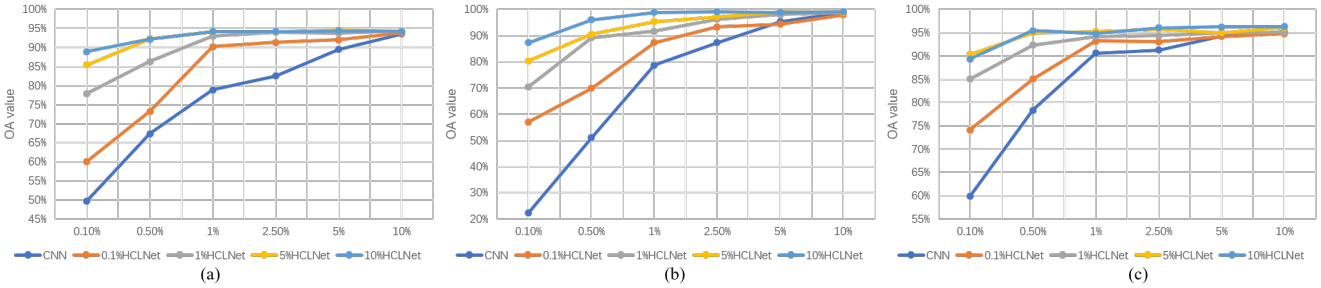


Fig. 17. Comparisons of the performance(OA) with different ratios of unlabeled and labeled samples between CNN and HCLNet on three data sets. (a) RADARSAT-2 Flevoland. (b) AIRSAR Flevoland. (c) ESAR Oberpfaffenhofen.

overall improvements are 3.78%, 3.36%, and 2.15%. However, HCLNet outperforms it in both full-sample classification and few-shot classification. In addition, due to the dual contrastive learning architecture of SSPRL and the unique selection of positive samples, its performance exceeds that of PCLNet. Numerically, the OA, AA, and Kappa of SSPRL are 0.24%, 0.5%, and 1.95% higher than that of PCLNet, respectively. However, the network architecture of SSPRL is too complex, and the number of parameters is too large, which is more than ten times that of HCLNet. Furthermore, its performance fails to surpass HCLNet. The proposed HCLNet obtains the best results in full-sample and few-shots, in which the OA, AA, and Kappa are 94.15%, 93.74%, and 91.84%.

For a more straightforward comparison, Fig. 8 illustrates the results of different methods in few-shot classification and full-sample classification of RADARSAT-2 Flevoland. The result clearly shows that HCLNet comprehensively outperforms all other methods. Furthermore, the classification maps of few-shot and full-sample classification results for different methods are presented in Figs. 9 and 10. It can be observed that the HCLNet achieves the best result of the different landforms. Moreover, as indicated by the black box in Fig. 9, we find many scattered, isolated pixel in the four compared methods. In comparison, our approach can solve this problem well due to the introduction of superpixels to ensure better contextual consistency in the phase of CL.

In order to more intutively show the advantage of HCLNet in few-shot classification, we respectively train HCLNet and CNN with the same architecture as online network of HCLNet according to different numbers of traning samples, and compare the accuray differences between them. Specifically, to explore the importance of the number of unlabeled and labeled samples, we designed comparison experiments between HCLNet and its backbone model using OA as the metric. We use the same online network of HCLNet without the projector head as the backbone model. Its parameters are initialized

TABLE IV
FEW-SHOT CLASSIFICATION RESULTS (%) ON THE AIRSAR FLEVOLAND WITH DIFFERENT METHODS

method	MI-SSL	CF-CSSL	PCLNet	SSPRL	HCLNet
Buildings	96.08	95.69	97.24	98.32	**99.95**
Rapeseed	53.92	37.23	87.32	90.60	**96.91**
Beet	87.23	92.03	98.33	99.02	**99.77**
Stembeans	83.10	69.15	98.18	98.89	**99.93**
Peas	86.54	85.73	95.24	96.32	**99.21**
Forest	85.01	67.23	96.38	97.93	**95.11**
Lucerne	91.99	92.19	95.02	97.28	**99.13**
Potatoes	93.12	93.02	92.37	95.00	**99.84**
Bare soil	94.13	92.95	89.88	92.13	**98.52**
Grass	89.26	83.72	85.42	89.26	**98.47**
Barley	92.19	92.43	95.77	96.23	**97.75**
Water	69.54	64.29	90.28	95.26	**98.87**
Wheat one	92.01	84.07	97.46	97.33	**99.71**
Wheat two	89.60	90.18	93.02	95.40	**99.92**
Wheat three	97.28	77.52	95.38	97.56	**99.32**
OA	84.32	89.02	94.30	95.83	**98.80**
AA	84.21	88.90	93.82	95.77	**98.82**
Kappa	83.98	88.54	93.27	95.23	**98.73**

TABLE V
FULL-SAMPLE CLASSIFICATION RESULTS (%) ON THE AIRSAR FLEVOLAND WITH DIFFERENT METHODS

method	MI-SSL	CF-CSSL	PCLNet	SSPRL	HCLNet
Buildings	99.07	99.53	98.30	99.53	**99.91**
Rapeseed	95.32	95.19	93.24	96.31	**97.14**
Beet	97.03	98.24	99.01	99.02	**99.78**
Stembeans	98.50	**99.99**	99.26	99.55	99.96
Peas	99.02	97.20	98.93	99.08	**99.12**
Forest	85.36	96.23	**97.41**	94.37	96.04
Lucerne	98.05	98.34	96.02	98.99	**99.81**
Potatoes	99.23	99.57	95.44	**99.82**	99.74
Bare soil	97.34	99.00	98.21	99.05	**99.26**
Grass	97.03	95.24	90.43	98.17	**98.49**
Barley	98.62	99.01	98.09	**99.56**	98.77
Water	95.44	98.98	91.52	**99.79**	98.84
Wheat one	98.79	99.26	97.40	**99.93**	99.69
Wheat two	99.91	99.90	95.88	99.84	**99.95**
Wheat three	97.36	96.51	98.31	99.40	**99.52**
OA	97.32	97.95	96.77	98.84	**99.05**
AA	97.14	98.02	96.50	98.83	**99.06**
Kappa	96.98	98.13	96.14	98.79	**98.99**

randomly. The backbone network and HCLNet are evaluated using 0.1%, 0.5%, 1%, 2.5%, 5%, and 10% samples per category. Moreover, for unlabeled samples, we set different ratios of 0.1%, 1%, 5%, and 10% to pre-training HCLNet. Compared to the results, we can find that the gap between the conventional CNN and HCLNet becomes larger with less labeled data, shown in Fig. 17(a) clearly. HCLNet performs similarly to the backbone network with 1% labeled data when trained with 0.1% labeled data and 10% unlabeled data, as shown in Fig. 17(a). When the ratio of labeled data is over 1%, the performance of HCLNet is better than the backbone network in any labeled data ratio. It fully demonstrates the effectiveness of HCLNet, which learns robust high-level representations from unlabeled data.

AIRSAR Flevoland: Similar experimental results for AIRSAR Flevoland are shown in Tables IV and V; the predicted map is shown in Figs. 11 and 12. Since this data set has more categories and fewer data per class, the actual data amount of 1% of training samples is small. It further widens the performance gap between HCLNet and other methods. Since there are only a few superpixels in each class, the difference between samples is tremendous, which makes the training of HCLNet more effective. The OA, AA, and Kappa of HCLNet are 2.97%, 3.05%, and 3.5% higher than the best previous method, demonstrating that HCLNet has a greater advantage over PCLNet and SSPRL in this case. Visually, as shown in Fig. 13, MI-SSL and CF-CSSL misclassified many data due to the lack of training labeled data. The performance is significantly degraded compared to the other methods. As mentioned, HCLNet reduces the scattered, isolated pixels to obtain better contextual consistency in the maps than the other four methods.

Furthermore, as shown in Fig. 17(b), the performance of HCLNet with the ratio of labeled data is 0.5%, and the ratio of unlabeled data is 10% is better than backbone network with the ratio of labeled data is 5%, and achieves the highest result.

TABLE VI
FEW-SHOT CLASSIFICATION RESULTS (%) ON THE ESAR OBERPFAFFENHOFEN WITH DIFFERENT METHODS

method	MI-SSL	CF-CSSL	PCLNet	SSPRL	HCLNet
Built-up areas	84.53	82.37	85.38	**87.79**	87.02
Wood land	89.22	91.05	89.70	90.88	**94.50**
Open areas	88.96	94.13	95.06	94.94	**98.33**
OA	88.05	88.23	90.83	91.78	**94.80**
AA	87.96	88.17	90.05	91.20	**93.29**
Kappa	87.89	88.25	85.21	89.32	**92.99**

TABLE VII
FULL-SAMPLE CLASSIFICATION RESULTS (%) ON THE ESAR OBERPFAFFENHOFEN WITH DIFFERENT METHODS

method	MI-SSL	CF-CSSL	PCLNet	SSPRL	HCLNet
Built-up areas	88.35	85.63	89.08	90.18	**90.34**
Wood land	94.88	95.02	**95.23**	94.27	95.20
Open areas	95.63	96.04	96.30	97.35	**97.92**
OA	92.87	93.02	94.09	94.30	**96.33**
AA	92.56	92.95	93.54	94.27	**94.49**
Kappa	92.63	92.99	91.22	92.99	**93.41**

When only 0.1% of training samples per category are used, the OA gap between the two methods is the largest, reaching 65.34%. In contrast, the backbone network requires at least 2.5% training samples per category to achieve the same result. It demonstrates the great generalization of the representation learned by HCLNet.

ESAR Oberpfaffenhofen: Compared with the first two data sets, ESAR Oberpfaffenhofen has fewer categories and a larger number of each category. It may result in similar data, but our method still works well in this case. As shown in Table VI, Table VII, and Fig. 14, compared to the other two methods, which provide a different strategy to reduce the similarity between data, HCLNet is 3.97%, 3.24%, 7.78% higher than PCLNet and 3.02%, 2.09%, 3.76% higher than SSPRL in the few-shot. It demonstrates the effectiveness of superpixel-based instance discrimination to reduce the similarity between data. It still can keep an excellent contextual consistency shown in Figs. 15 and 16. Moreover, due to the more labeled data, it can be observed that when the ratio of labeled data is 0.5%, HCLNet shows excellent performance, outperforming the backbone network with any ratio of labeled data, as shown in Fig. 17(c).

To sum up, the experimental results on three data sets can confirm the effectiveness of HCLNet in generalization and contextual consistency.

2) Scattering Confusion: In evaluating the effectiveness of a PolSAR classification method, the accuracy is one aspect, but the most important thing is whether the method can solve the scattering confusion problem of the different land covers in PolSAR. So we use the **confusion matrix** to make a qualitative and quantitative analysis on the performance of HCLNet in solving scattering confusion problem. Specifically, we choose AIRSAR Flevoland dataset for the main experimental results presentation, mainly due to its large number of land cover classes, each land cover is more prone to scattering confusion. As show in Fig. 18, it can be seen that the scattering confusion between different land covers classification of MI-SSL and CF-CSSL is very serious. For example, in MI-SSL, there is Rapeseed certain scattering similarity between Forest, Potatoes, and Barley, which leads to the model's misclassification of these types of land covers. And in CF-CSSL, except that it has similar phenomenon to MI-SSL, it also has the scattering confusion problem between Forest and Potatoes leading to model classification errors. The situation of PCLNet and SSPRL is somewhat better, but there is still a partial scattering confusion problem. PCLNet still has some shortcomings in dealing with the scattering similarity of Beet and Wheat two. It cannot learn the difference between their scattering features completely, which leads to the model misclassifying part of Beet. Similarly, SSPRL does not fully learn the scattering difference between each land cover, which leads to some confusion in its classification of Rapeseed and Forest. However, HCLNet learns the connection and difference between different pixles well through the target network's learning of scattering similarity and the online network's learning of scattering difference. There are very few instances where a class is significantly misclassified to another, that is, there is no correlation between the classes. This shows that HCLNet has a great improvement in alleviating the scattering confusion problem.

3) Representation visualization: In the above experiments, HCLNet has demonstrated the powerful ability of representation learning with unlabeled data using supervised-based instance discrimination. In order to further explore the quality of representation, 2-D t-stochastic neighbor embedding(t-SNE) [57] is used to visualize the learned representation. Specifically, MI-SSL, CF-CSSL, PCLNet, SSPRL, and HCLNet utilize 1% labeled samples per category in the map. For each method, the output of the representation encoder is used as the representation without any labeled samples. These visualizations are performed on three benchmark data sets above. The results are shown in Figs. 19, 20, and 21, different

Fig. 18. Confusion matrix of the classification result on the AIRSAR Flevoland data set with different methods. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

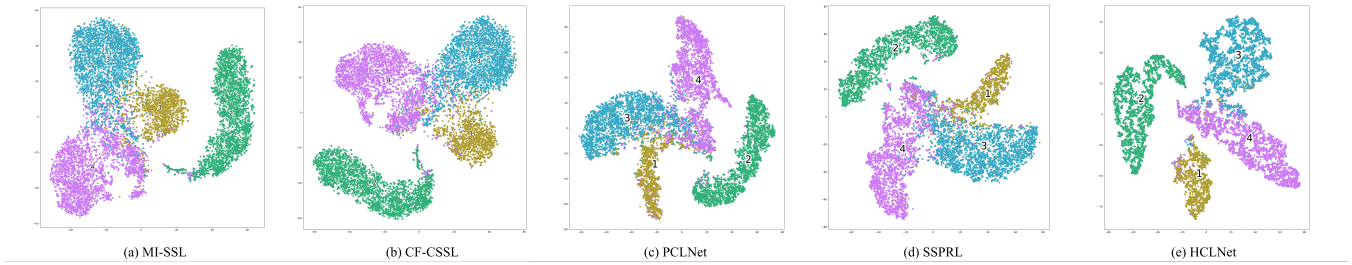


Fig. 19. T-SNE visualization of the representations learned on the RADARSAT-2 Flevoland data set with different methods. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

colors indicate different categories.

From the results, it can be observed that MI-SSL cannot distinguish each category well, with existing multiple categories highly overlapping. Moreover, the closeness is relatively weak. Some features of the same category are distributed in different locations and form multiple disconnected regions.

For CF-CSSL, under the guidance of only a few labeled data, it leads to the poor improvement of closeness overlapping, and some even deteriorate. Benefiting from training on large amounts of unlabeled data, PCLNet and SSPRL drastically reduce the overlapping but still exist some disconnected regions. On the contrary, HCLNet significantly alleviates disconnection and overlapping. It turns out that HCLNet provides a better representation by learning from different perspectives for the downstream task to improve the classification ability.

Through t-SNE, it also illustrates again the significant improvement of HCLNet on scattering confusion problem. The representations obtained by HCLNet distinguish each land cover well, and there are few cases of severe overlap of representations in some classes as with other methods.

4) Model Complexity Analysis: We evaluated the complexity of the models using the number of model parameters and floating-point operations (FLOPs). The quantitative comparison results of all methods are listed in Table VIII. All the contrasting methods follow the setup in the original paper. For instance, CF-CSSL use the the U-Net that follows an encoder-decoder architecture, which the encoder includes 3 convolutional blocks, each of which consists of two 3×3 convolutional layers and one 2×2 maxpooling layer, and the decoder has a symmetric architecture to the encoder, which each block is composed of one up-sampling layer and three convolutional layers. It has a high number of parameters

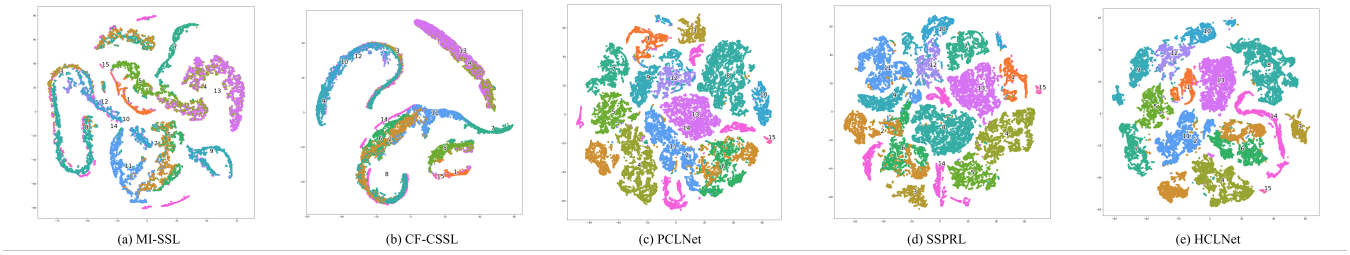


Fig. 20. T-SNE visualization of the representations learned on the AIRSAR Flevoland data set with different methods. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

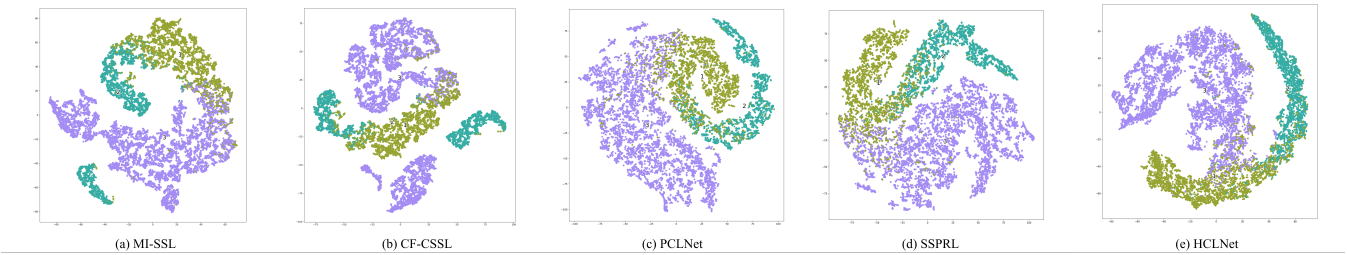


Fig. 21. T-SNE visualization of the representations learned on the ESAR Oberpfaffenhofen data set with different methods. (a) MI-SSL. (b) CF-CSSL. (c) PCLNet. (d) SSPRL. (e) HCLNet.

and FLOPs. MI-SSL has multiple forward computations and multiple contrast computations, so it has high FLOPs. PCLNet and SSPRL have higher FLOPs due to the complex auxiliary modules in their networks. In contrast, our method has a relatively low number of parameters and FLOPs because our network only has fewer layers, and not too many reduntant, complex designs.

5) Ablation study: To better understand the effectiveness of the individual components of HCLNet, we experiment with combinations of different components on the three data sets above and design the three groups of ablation experiments. The ablation results are reported in Table IX. The abbreviations in Table VIII represent the different components: CNN is the Siamese network in which the target network and the online network are the same network architecture, and both of them share parameters; Hnet is the heterogeneous network with the same architecture as HCLNet; SID is the superpixel-based Instance Discrimination; BS is the beam search. The results implicate that each component can improve the classification results. The first two experiments involve that compared with CNN as the architecture, Hnet exhibits the more powerful representation extraction ability and has fewer parameters. When BS is added to remove the redundancy between features, the model can learn better high-level representation, demonstrated in the experiment's third group.

Significantly, the fourth group of experiments demonstrates that the SID is essential for heterogeneous networks. As shown in Fig. 22(a), without SID, the training of HCLNet likely falls into the salt and pepper noise problem. We analyze that the network is challenging to learn the high-level representation when CL without SID to distinguish the differences between pixels and even confuse the representations between pixels

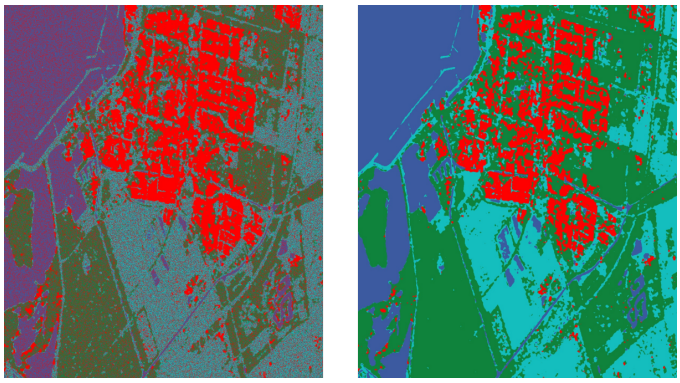


Fig. 22. (a) The salt and pepper noise problem and (b) the improved result.

of different categories. To verify our conjecture, we replace the SID and use pixels of different categories as negative samples and pixels of the same category as positive samples to increase the dissimilarity between negative samples. Finally, it successfully solves the salt and pepper noise problem, as shown in Fig. 21(b). So starting from the direction of SSL, Hnet, and SID may be a match made in heaven.

TABLE VIII
COMPARISON ON THE NUMBER OF PARAMETERS AND FLOPS FOR DIFFERENT METHODS.

Datasets	Complexity	MI-SSL	CF-CSSL	PCLNet	SSPRL	HCLNet
RADARSAT-2 Flevoland	Parameter(M)	1.67	12.35	0.34	5.79	0.28
RADARSAT-2 Flevoland	FLOPs(M)	12.53	4.67	2.90	14.85	2.43
AIRSAR Flevoland	Parameter(M)	2.09	12.32	0.59	8.63	0.28
AIRSAR Flevoland	FLOPs(M)	20.63	7.96	5.12	25.39	4.74
ESAR Oberpfaffenhofen	Parameter(M)	1.44	12.33	0.32	5.26	0.28
ESAR Oberpfaffenhofen	FLOPs(M)	9.89	3.97	2.13	11.92	1.85

TABLE IX
OA (%) ON RADARSAT-2 FLEVOLAND (OA_{RF}), AIRSAR FLEVOLAND (OA_{AF}) AND ESAR OBERPFAFFENHOFEN (OA_{EO}) DATA SETS FOR ABLATION STUDY

method	OA_{RF}	OA_{AF}	OA_{EO}
CNN+SID+BS	90.22	95.35	91.37
Hnet+SID+BS	94.15	98.80	94.80
Hnet+SID	91.47	93.65	92.91
Hnet+BS	64.50	85.32	79.66

V. CONCLUSION

This article proposes a self-supervised learning method based on CL with a heterogeneous network for the first time. We use HCLNet to extract the high-level representation of PolSAR data from the physical and statistical properties and propose two plugins. The beam search is introduced to select the appropriate combination of features to reduce the redundancy of multi-target decomposition features. The superpixel-based Instance Discrimination is proposed to reduce the similarity between pixels and learn better representation. Therefore, with the help of unsupervised pre-training to learn representation, the online network can achieve high results of few-shot PolSAR classification by fine-tuning. Experiments are conducted on three widely used benchmark data sets, and the experimental results demonstrate the performance of HCLNet compared with several mainstream methods in both few-shot and full-sample classification.

Compared with the CL of optical images, PolSAR has more valuable features under different properties, while the heterogeneous network has the natural advantage of entirely using these features. This work creates a precedent for future research on heterogeneous network learning. And we believe that more in-depth and comprehensive research about the heterogeneous network may further improve the PolSAR classifier performance. Our future interest is to explore the problem of positive and negative selection for heterogeneous networks in depth.

REFERENCES

[1] W. Nie, K. Huang, J. Yang, and P. Li, "A deep reinforcement learning-based framework for polsar imagery classification," *IEEE Trans. Geosci. Remote Sens*, vol. 60, pp. 1–15, 2022.

[2] H. Bi, F. Xu, Z. Wei, Y. Xue, and Z. Xu, "An active deep learning approach for minimally supervised polsar image classification," *IEEE Trans. Geosci. Remote Sens*, vol. 57, no. 11, pp. 9378–9395, 2019.

[3] G. Hong, S. Wang, J. Li, and J. Huang, "Fully polarimetric synthetic aperture radar (sar) processing for crop type identification," *Photogramm. Eng. Remote Sens.*, vol. 81, no. 2, pp. 109–117, 2015.

[4] F. T. Ulaby and C. Elachi, "Radar polaritnetry for geoscience applications," 1990.

[5] R. Shirvany, M. Chabert, and J.-Y. Tourneret, "Ship and oil-spill detection using the degree of polarization in linear and hybrid/compact dual-pol sar," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 3, pp. 885–892, 2012.

[6] A. Freeman and S. L. Durden, "A three-component scattering model for polarimetric sar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 3, pp. 963–973, 1998.

[7] S. R. Cloude and E. Pottier, "An entropy based classification scheme for land applications of polarimetric sar," *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 1, pp. 68–78, 1997.

[8] E. Pottier, "Dr. jr huynen's main contributions in the development of polarimetric radar techniques and how the'radar targets phenomenological concept'becomes a theory," in *Proc. SPIE*, vol. 1748. SPIE, 1993, pp. 72–85.

[9] J. R. Huynen, "Physical reality of radar targets," in *Proc. SPIE*, vol. 1748. SPIE, 1993, pp. 86–96.

[10] W. L. Cameron and L. K. Leung, "Feature motivated polarization scattering matrix decomposition," in *Proc. IEEE Int. Conf. Radar*. IEEE, 1990, pp. 549–557.

[11] E. Krogager, "New decomposition of the radar target scattering matrix," *Electron. Lett.*, vol. 18, no. 26, pp. 1525–1527, 1990.

[12] C. Lardeux, P.-L. Frison, C. Tison, J.-C. Souyris, B. Stoll, B. Fruneau, and J.-P. Rudant, "Support vector machine for multifrequency sar polarimetric data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 12, pp. 4143–4152, 2009.

[13] B. Zou, H. Li, and L. Zhang, "Polsar image classification using bp neural network based on quantum clonal evolutionary algorithm," in *2010 IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*. IEEE, 2010, pp. 1573–1576.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 60, no. 6, pp. 84–90, 2017.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[17] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[18] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Trans. Neur. Net. Lear.*, vol. 28, no. 10, pp. 2222–2232, 2016.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[20] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosc. Rem. Sen. M.*, vol. 5, no. 4, pp. 8–36, 2017.

[21] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, 2016.

[22] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogramm.*, vol. 152, pp. 166–177, 2019.

[23] Y. Zhou, H. Wang, F. Xu, and Y.-Q. Jin, "Polarimetric sar image classification using deep convolutional neural networks," *IEEE Geosci. Remote Sens.*, vol. 13, no. 12, pp. 1935–1939, 2016.

[24] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric sar image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7177–7188, 2017.

[25] F. Liu, L. Jiao, B. Hou, and S. Yang, "Pol-sar image classification based on wishart dbn and local spatial information," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3292–3308, 2016.

[26] A. G. Mullissa, C. Persello, and A. Stein, "Polsarnet: A deep fully convolutional network for polarimetric sar image classification," *IEEE J-Stars.*, vol. 12, no. 12, pp. 5300–5309, 2019.

[27] X. Tan, M. Li, P. Zhang, Y. Wu, and W. Song, "Complex-valued 3-d convolutional neural network for polsar image classification," *IEEE Geosci. Remote Sens.*, vol. 17, no. 6, pp. 1022–1026, 2019.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[30] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[31] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, 2020.

[32] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 3733–3742.

[33] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 6210–6219.

[34] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[35] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 776–794.

[36] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 9729–9738.

[37] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2020, pp. 1597–1607.

[38] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, pp. 21 271–21 284, 2020.

[39] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 15 750–15 758.

[40] B. Ren, Y. Zhao, B. Hou, J. Chanussot, and L. Jiao, "A mutual information-based self-supervised learning model for polsar land cover classification," *IEEE Trans. Geosci. Remote Sens*, vol. 59, no. 11, pp. 9224–9237, 2021.

[41] L. Zhang, S. Zhang, B. Zou, and H. Dong, "Unsupervised deep representation learning and few-shot classification of polsar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2020.

[42] W. Zhang, Z. Pan, and Y. Hu, "Exploring polsar images representation via self-supervised learning and its application on few-shot classification," *IEEE Geosci. Remote Sens.*, vol. 19, pp. 1–5, 2022.

[43] Y. Cui, F. Liu, X. Liu, L. Li, and X. Qian, "Tcspanet: two-staged contrastive learning and sub-patch attention based network for polsar image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 14, no. 10, p. 2451, 2022.

[44] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1058–1067.

[45] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 2650–2658.

[46] C. Yang, B. Hou, B. Ren, Y. Hu, and L. Jiao, "Cnn-based polarimetric decomposition feature selection for polsar image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8796–8812, 2019.

[47] S. R. Cloude and E. Pottier, "A review of target decomposition theorems in radar polarimetry," *IEEE Trans. Geosci. Remote Sens.*, vol. 34, no. 2, pp. 498–518, 1996.

[48] Y. Yamaguchi, T. Moriyama, M. Ishido, and H. Yamada, "Four-component scattering model for polarimetric sar image decomposition," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 8, pp. 1699–1706, 2005.

[49] J. J. van Zyl, "Application of cloude's target decomposition theorem to polarimetric imaging radar data," in *Radar polarimetry*, vol. 1748. SPIE, 1993, pp. 184–191.

[50] L. Zhang, B. Zou, H. Cai, and Y. Zhang, "Multiple-component scattering model for polarimetric sar image decomposition," *IEEE Geosci. Remote Sens.*, vol. 5, no. 4, pp. 603–607, 2008.

[51] W. A. Holm and R. M. Barnes, "On radar polarization mixed target state decomposition techniques," in *Proc. IEEE Int. Conf. Radar*. IEEE, 1988, pp. 249–254.

[52] A. Haddadi G, M. Reza Sahebi, and A. Mansourian, "Polarimetric sar feature selection using a genetic algorithm," *Canadian Journal of Remote Sensing*, vol. 37, no. 1, pp. 27–36, 2011.

[53] X. Huang and X. Nie, "Multi-view feature selection for polsar image classification via l, sparsity regularization and manifold regularization," *IEEE Transactions on Image Processing*, vol. 30, pp. 8607–8618, 2021.

[54] H. Dong, L. Zhang, D. Lu, and B. Zou, "Attention-based polarimetric feature selection convolutional network for polsar image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.

[55] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[56] M. Yang, L. Jiao, F. Liu, B. Hou, S. Yang, Y. Zhang, and J. Wang, "Coarse-to-fine contrastive self-supervised feature learning for land-cover classification in sar images with limited labeled data," *IEEE Transactions on Image Processing*, vol. 31, pp. 6502–6516, 2022.

[57] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.