Research on Semantic Segmentation Method of Remote Sensing Image Based on Self-supervised Learning

Wenbo Zhang, Achuan Wang
College of Computer and Control Engineering, Northeast Forestry University, Harbin, China

***Abstract*—To address the challenge of requiring a large amount of manually annotated data for semantic segmentation of remote sensing images using deep learning, a method based on self-supervised learning is proposed. Firstly, to simultaneously learn the global and local features of remote sensing images, a self-supervised learning network structure called TBSNet (Triple-Branch Self-supervised Network) is constructed. This network comprises an image transformation prediction branch, a global contrastive learning branch, and a local contrastive learning branch. The contrastive learning part of the network employs a novel data augmentation method to simulate positive pairs of the same remote sensing images under different weather conditions, enhancing the model's performance. Meanwhile, the model integrates channel attention and spatial attention mechanisms in the projection head structure of the global contrastive learning branch, and replaces a fully connected layer with a convolutional layer in the local contrastive learning branch, thus improving the model's feature extraction ability. Secondly, to mitigate the high computational cost during the pre-training phase, an algorithm optimization strategy is proposed using the TracIn method and sequential optimization theory, which increases the efficiency of pre-training. Lastly, by fine-tuning the model with a small amount of annotated data, effective semantic segmentation of remote sensing images is achieved even with limited annotated data. The experimental results indicate that with only 10% annotated data, the overall accuracy (OA) and recall of this model have improved by 4.60% and 4.88% respectively, compared to the traditional self-supervised model SimCLR (A Simple Framework for Contrastive Learning of Visual Representations). This provides significant application value for tasks such as semantic segmentation in remote sensing imagery and other computer vision domains.**

***Keywords*—Computer vision; deep learning; self-supervised learning; remote sensing image; semantic segmentation**

I. Introduction

With the rapid development of remote sensing satellite technology, remote sensing images are playing an increasingly critical role in various fields such as urban planning, resource exploration, and natural disaster prediction 0. Extracting useful information from the vast wealth of remote sensing geo-information has become a long-standing scientific challenge in remote sensing. Among the methods explored, semantic segmentation [2] has proven to be an effective approach.

In the field of semantic segmentation for remote sensing images, there are two main approaches: traditional methods based on handcrafted feature descriptors and deep learning methods based on Convolutional Neural Networks (CNNs). Due to the complexity of background and scale differences in high-resolution remote sensing images, traditional methods have not been very effective. However, since Long et al. proposed the Fully Convolutional Neural Network (FCN) [3] in 2015, deep learning-based techniques for semantic segmentation in remote sensing images have made significant progress. This has led to the development of post-processing techniques based on probabilistic graphical models [4], global context modeling using multi-scale aggregations [5], and perpixel semantic modeling based on attention mechanisms [6].

For instance, Ronneberger et al. introduced the U-Net model [7], which employs an encoder-decoder architecture with lateral connections, enabling multi-scale recognition and feature fusion in the image. Similarly, Chen et al. proposed the DeepLabV3+ model [8], which utilizes a spatial pyramid structure to gather rich contextual information through pooling operations at various resolutions. Furthermore, it uses the encoder-decoder architecture to achieve precise object boundaries, thereby enhancing segmentation accuracy.

Despite the achievements made in deep learning-based semantic segmentation of remote sensing images in recent years [9][10], these methods all rely on large amounts of manually annotated data to train the neural network. This requirement not only consumes significant human resources but also reduces the efficiency of semantic segmentation. Therefore, the application of self-supervised learning [11] to semantic segmentation of remote sensing images has become a feasible method. Li et al. [12] proposed a multi-task self-supervised learning method for semantic segmentation of remote sensing images, which applied three pretext tasks [13] to self-supervised learning and achieved decent results. However, these pretext tasks only learn the global features of the image, lacking in the learning of local features of the image. Thus, how to effectively use these unannotated remote sensing data has become a major research focus in recent years.

The main contributions of this study are as follows:

1) To tackle the aforementioned challenges, a self-supervised semantic segmentation approach for remote sensing images is introduced, along with the design of a triple-branch self-supervised network named TBSNet. This network

uses an image transformation prediction branch and a global contrastive learning branch to learn global features of images, and a local contrastive learning branch to learn local features.

2) On this basis, the projection head structures in the global contrastive learning branch and the local contrastive learning branch are improved to enhance their performance. Specifically, in the projection head of the global contrastive learning branch, a combination of spatial attention mechanisms [14] and channel attention mechanisms [15] are used to better focus on the important parts of the feature map, thus improving the quality of feature representation. In the local contrastive learning branch, the original first fully connected layer is replaced with a convolutional layer, which enables the learning of richer local features.

3) Considering the heavy computational cost of pre-training, the TracIn method [16] and sequential optimization theory [17] are employed to optimize the model's pre-training process, reducing computational and time costs. Finally, the model is fine-tuned in the downstream task using a small amount of annotated data to achieve the expected semantic segmentation results.

The remaining structure of the article is outlined as follows: Section II introduces the background knowledge and relevant work. Section III describes the implementation details of the proposed method, including the network framework and optimization techniques. Section IV presents the experiments conducted and analyzes the obtained results. Section V provides the conclusions drawn from our experiments.

II. BACKGROUND

A. Self-supervised Learning

Self-supervised learning is a type of unsupervised learning [18] , as shown in Fig. 1. Compared to supervised learning, it utilizes a large amount of unlabeled data through specially designed pretext tasks. This approach relies on pseudo-labels generated by the model itself, enabling it to learn high-level features from the input data. The model can then be further transferred to downstream tasks in actual applications. With a small amount of labeled data, the model can be fine-tuned to achieve, or even surpass, the performance of supervised learning. Generally, self-supervised learning can be divided into generative and contrastive categories [19] [20].

B. Pretext Task

During the self-supervised pre-training phase, different pretext tasks are typically designed to allow the model to more effectively learn the intrinsic features and interrelationships within the samples. By performing these pretext tasks, the model can generate pseudo-labels internally to guide its learning, thus achieving self-supervised learning without the need for labeled data. Classic pretext tasks include image inpainting [21], which uses neural networks to repair missing parts by learning texture features; rotation prediction [22], which allows neural networks to grasp the overall features of an image; and jigsaw puzzles [23], where the neural network needs to learn the relative positional features among different pieces for image stitching. These pretext tasks have achieved good results in instance-level image classification tasks. However, their effectiveness is not ideal for semantic segmentation tasks due to a lack of learning about local features.

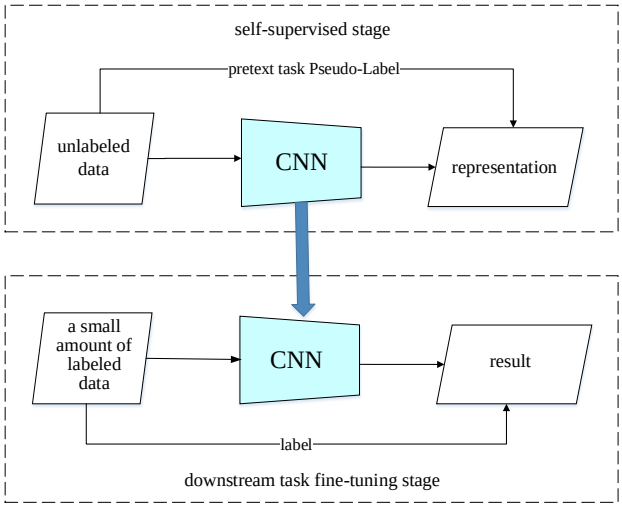

Fig. 1. Schematic diagram of self-supervised model.

C. Contrastive Learning

Contrastive self-supervised learning [24], also referred to as contrastive learning, shows more promising results in the field of remote sensing compared to generative self-supervised learning. The central idea of contrastive learning, a common method of self-supervised learning, is to learn high-level semantic features by contrasting two semantically similar inputs. Specifically, samples are divided into positive and negative pairs, with the aim of drawing positive samples closer while pushing negative samples farther apart, as shown in formula (1):

$$sim\big(f(x), f(x^+)\big) \gg sim\big(f(x), f(x^-)\big) \quad (1)$$

Here, x^+ represents a sample semantically similar to x, thus forming a positive pair with x; x^- is a sample that is different from x, thereby forming a negative pair with x. sim represents the similarity measure between two pairs of features generated by encoding function f.

Classic examples of contrastive learning include Momentum contrast (MoCo) [25] and SimCLR [26]. MoCo introduces momentum contrast for unsupervised visual representation learning, constructing a dynamic dictionary with a queue and a moving average encoder to improve the effects of contrastive learning. SimCLR presents a simple framework where two different data augmentations of the same image x are generated as a positive pair (x_i and x_j), while the augmented image from a different image y serves as a negative sample. A projection head is added after the encoder to achieve significant results. While both of the above-mentioned models have achieved significant accomplishments in self-supervised learning research, they also exhibit notable limitations. This is because both models utilize pairs of images as positive samples, allowing them to effectively learn overall features of images. However, they lack the ability to learn local features.

Recent years have seen extensive research on image processing based on contrastive learning, such as a method proposed by Krishna et al. for medical image semantic segmentation based on global and local features [27]. Research on self-supervised learning for remote sensing images mainly focuses on instance-level remote sensing scene classification [28][29], given the comparatively limited exploration of pixel-level semantic segmentation in the context of remote sensing images, a triple-branch network architecture is introduced. This architecture facilitates the acquisition of both global and local image features, consequently leading to enhanced semantic segmentation outcomes for remote sensing images.

D. TracIn Method

Deep learning requires a large amount of data support, and the quality of data often has a significant impact on model training. An important measure of data quality is influence, but due to the complexity of models and the growing influence of scale features and datasets, it is challenging to quantify influence. The TracIn method captures changes in predictions when accessing individual training examples by tracking the training process and determines the influence of training examples by assigning influence scores to each.

E. Order Optimization Theory

Order Optimization (OO) is an effective strategy widely used in the industry to solve optimization problems, with its specific solution process shown in Fig. 2.

For a given optimization problem, suppose the set of the "truly best" g solutions is G. However, due to computational resource constraints, the set G cannot be solved from the solution space. Using the order optimization idea, a rough model with simple computations is used to select some solutions from G. All solutions are ranked according to some performance evaluation method provided by the rough model, and the best s solutions are chosen to form the solution set S. In the process of using the rough model, we generally only care about how many of the intersecting parts of sets G and S (G∩S) are genuinely good solutions. The order optimization quantifies the probability that the set S obtained based on the rough model corresponds to $|G \cap S| \geq k$, i.e., the alignment probability (AP). In practice, the alignment probability of sets S and G is often much larger than expected, and the amount of data in set S is often several orders of magnitude smaller than the real solution space, so the order optimization method can typically save at least one order of magnitude of performance evaluation times.

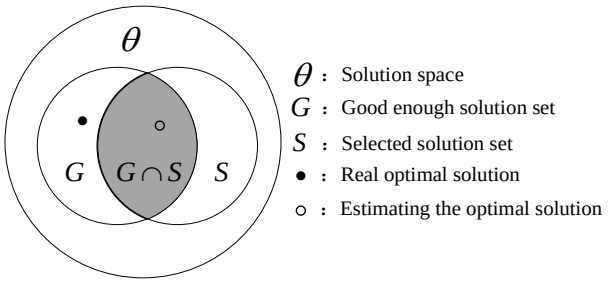

Fig. 2. Schematic diagram of solving sequential optimization theory.

III. Proposed Method

A. Network Architecture Design for Semantic Segmentation of Remote Sensing Images Based on Self-supervised Learning

With the objective of improving the outcomes of semantic segmentation for remote sensing images using a limited quantity of annotated data, as well as intensifying the acquisition of local small-object characteristics, a triple-branch network architecture known as TBSNet is introduced. As shown in Fig. 3, this network structure includes an image transformation prediction branch, a global contrastive learning branch, and a local contrastive learning branch. The image transformation prediction branch and the global contrastive learning branch are used to learn the overall features of the image, while the local contrastive learning branch can learn the local features of the image. Each branch performs self-supervised learning in different ways, and then the losses of each branch are summed up as the total loss for adjusting the network parameters.

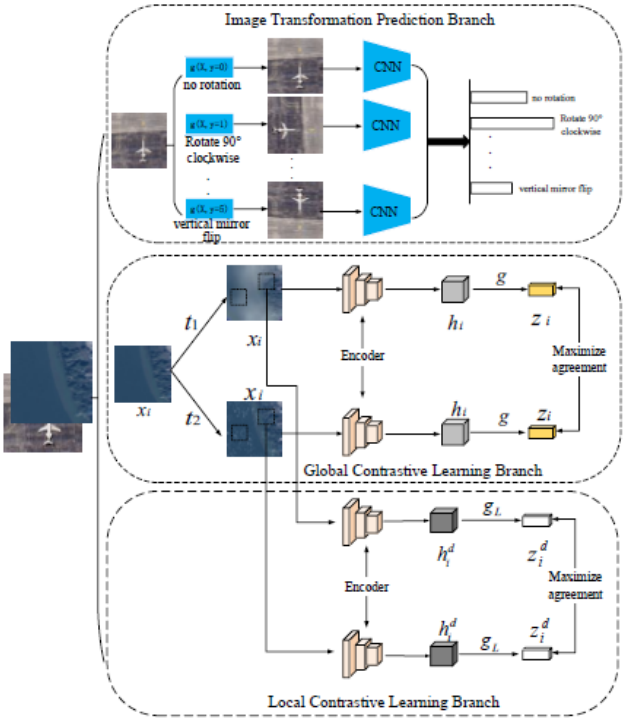

Fig. 3. Schematic diagram of triple-branch network TBSNet.

1) Design of Image Transformation Prediction Branch Learning Strategy: To realize the learning of overall semantic features of images without labels, this branch randomly rotates (e.g., 90°, 180°, 270°) or mirror flips the original image, and feeds the original image and the rotated image into the neural network for transformation type identification. Since remote sensing images have rotation invariance, rotation can help the neural network better understand the concepts described in remote sensing images. Specifically, the aforementioned rotations are defined as a set of discrete geometric transformations $G=\{y_0, y_1, .. y_m\}$. One is randomly selected from G and applied to the input image x to get x', which is

then fed into the network and trained to identify the type of rotation, transforming the image transformation prediction branch into a classification problem. The loss function of the transformation prediction branch can be defined as shown in equation (2):

$$L_\alpha = -\sum_{m=1}^{M} \hat{A}_{(m)} log P_{(m)} \quad (2)$$

Here, $\hat{A}_{(m)} = \{0,1\}$ represents the one-hot encoding of the basic true value class, and P represents the probability of M different types of geometric transformations. In this branch, geometric transformations are divided into six types, which are clockwise rotation of 90°, 180°, 270°, horizontal left-right mirror flipping, vertical top-bottom mirror flipping, and no rotation.

2) Global Contrastive Learning Branch Learning Strategy Design

a) Remote Sensing Data Enhancement Method Based on Weather Conditions: For the same location, remote sensing images under different weather conditions exhibit variations, yet their deep features remain consistent. Consequently, in the global contrastive learning segment, traditional data augmentation approaches for forming positive sample pairs are eschewed. Instead, the data augmentation method is adjusted to mimic diverse weather conditions at the identical location, aligning with the distinct traits of remote sensing images. Any x_i in $\{x_1, x_2, .. x_N\}$ undergoes two different random data augmentations (including simulating clouds, simulating snowflakes, simulating haze, and no augmentation). To simulate cloud layers, this paper adds Perlin noise to the original image and then blurs the cloud layer using a Gaussian filter. To simulate snowflakes, random noise is added to the original image, and then a median filter is used to simulate the snowflake effect. To simulate haze, we utilize a method of overlaying a generated haze layer onto the original image. The haze layer is represented by an array of the same size as the original image. To ensure uniform effect across all color channels, this array is expanded to a three-channel array, with each channel having the same values as the original random array. Each element of the haze layer is multiplied by the haze intensity, and the result is multiplied with each pixel of the original image. This effectively reduces the contrast of the original image in the haze areas. Then, the haze layer is multiplied by the atmospheric brightness and added to the original image, simulating the haze effect. In this paper, the haze intensity is randomly selected between 0.3 and 0.8, and the atmospheric brightness is randomly chosen between 250 and 270.

b) Model design integrating channel and spatial attention mechanisms: The two samples $\tilde{x}_i$ and $\hat{x}_i$ obtained after enhancement are positive samples for each other, and other samples in the same batch are all negative samples. The two positive samples obtained are passed through the encoder-based backbone network f to get the feature vectors $\tilde{h}_i$ and $\hat{h}_i$, which are then mapped to the contrast loss space through an improved MLP projection head $g(\cdot)$ to get $\tilde{z}_i$ and $\hat{z}_i$. As shown in Fig. 4, this work introduce the combination of channel attention mechanism and spatial attention mechanism on the basis of the original projection head, improving the discriminability and expressive power of features. It can also adaptively select important features, which helps to improve the model's generalization ability on different types of remote sensing images. In order to better integrate the channel attention module and the spatial attention module, a convolution layer and ReLU activation function are added. This additional convolution layer can help to further extract features before applying the attention mechanism.

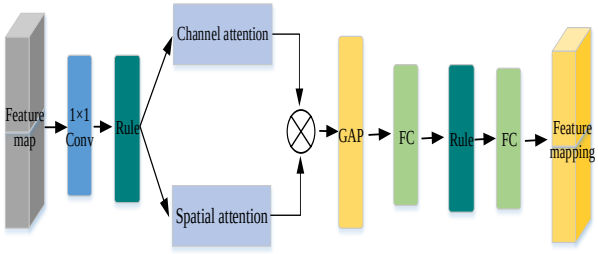

Fig. 4. Schematic diagram of global contrastive learning branch projection head structure.

Lastly, the contrast loss is used to bring positive samples closer, thus learning the geographical features in the image. The contrast loss is defined as follows in equation (3):

$$L_\beta = \frac{1}{2N}\sum_{k=1}^{N}\left(l_\beta(\tilde{x}_i, \hat{x}_i) + l_\beta(\hat{x}_i, \tilde{x}_i)\right) \quad (3)$$

Where N represents the number of samples in the same batch, l_β uses the NT-Xent contrast loss function in SimCLR as shown in equation (4):

$$l_\beta(\tilde{x}_i, \hat{x}_i) = -log\frac{\exp\left(\frac{sim(\tilde{z}_i, \hat{z}_i)}{\tau}\right)}{\sum_{k=1}^{2N} 1_{[k\neq i]}\exp\left(\frac{sim(\tilde{z}_i, z_k)}{\tau}\right)} \quad (4)$$

Here, $1_{[k\neq i]}$ is an indicator function that equals 1 when $k \neq i$, $sim(u,v) = \frac{u^T v}{\|u\|\,\|v\|}$, i.e., it calculates the cosine similarity between u and v. z_k represents the feature vector obtained after the negative sample goes through the projection head, that is, $z_k = g\left(f\left(t(x_k)\right)\right)$. τ is the temperature parameter, which is set to 0.1 in this paper.

3) Local Contrastive Learning Branch Learning Strategy Design: The above two branches can effectively learn the global information of images. However, for semantic segmentation, learning only global features is not enough. A single remote sensing image may contain various objects, and the learning of global features cannot effectively handle small targets. Therefore, the local contrastive learning branch can learn more local information, which is crucial for improving the performance of semantic segmentation. This branch shares the sample after data augmentation with the global contrastive learning branch. It forms positive sample pairs by selecting two local blocks of the same size from the same location in two augmentation images, and takes the local areas of other images in the same batch as negative samples. In this paper, a random selection of a central pixel point and outward expansion method is employed for selecting a local region. To

prevent insufficient edge size, if the selected size is s*s, the central pixel point is chosen within the rectangular region formed by the four points: $\left(\left\lfloor\frac{s}{2}\right\rfloor,\left\lfloor\frac{s}{2}\right\rfloor\right),\left(\left\lfloor\frac{s}{2}\right\rfloor,\left(255-\left\lfloor\frac{s}{2}\right\rfloor\right)\right),\left(\left(255-\left\lfloor\frac{s}{2}\right\rfloor\right),\left\lfloor\frac{s}{2}\right\rfloor\right),\left(\left(255-\left\lfloor\frac{s}{2}\right\rfloor\right),\left(255-\left\lfloor\frac{s}{2}\right\rfloor\right)\right)$. To avoid excessive duplication of selected regions, pixels are only chosen with odd coordinates. Moreover, to prevent selecting the same region multiple times, each pixel point can only be selected once.

Just like the global contrastive learning branch, for each global contrastive learning image, m local areas are selected for contrast learning. Therefore, for the selected two local areas $\widetilde{x_i^d}$ and $\widehat{x_i^d}$ $(d \in (1, m))$, after going through the encoder, we get the feature vectors $\widetilde{h_i^d}$ and $\widehat{h_i^d}$, and finally map the local area features to be $\widetilde{z_i^d}$ and $\widehat{z_i^d}$ through the projection head $g_L(\cdot)$ similar to the one in the global contrastive learning branch. As shown in Fig. 5, in this branch, since the sample image has already been cropped to the local area, the attention mechanism is not used in the projection head. Instead, the first fully connected layer in the original projection head is replaced with a convolution layer to retain spatial information and better capture the features within the local area, thus improving the segmentation effect of the model.

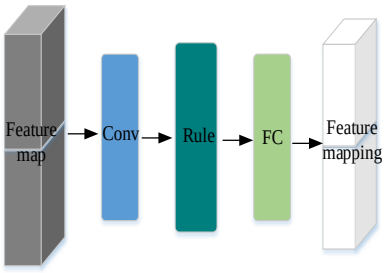

Fig. 5. Schematic diagram of local contrastive learning branch projection head structure.

The contrast loss can be represented as equation (5):

$$L_\gamma = \frac{1}{2N_\gamma}\sum_{i=1}^{N_\gamma}\left(l_c\left(\widetilde{x_i^d},\widehat{x_i^d}\right),l_c\left(\widehat{x_i^d},\widetilde{x_i^d}\right)\right) \quad (5)$$

$$\text{where, } l_c\left(\widetilde{x_i^d},\widehat{x_i^d}\right) = -log\frac{\exp\left(\frac{sim\left(\widetilde{z_i^d},\widehat{z_i^d}\right)}{\tau}\right)}{\sum k_d\in\Lambda_\gamma^-\exp\left(\frac{sim\left(\widetilde{z_i^d},z_{(k_d)}\right)}{\tau}\right)} \quad (6)$$

In this, N_γ represents the number of all local regions in a batch, i.e., $N_\gamma = N \times m$. Λ_γ^- represents the other local regions outside the two local area positive samples.

The total loss of the triple-branch self-supervised network can be represented as shown in equation (7), which is used for the calculation of the TracIn score during optimization.

$$L = L_\alpha + L_\beta + L_\gamma \quad (7)$$

B. Design of Self-supervised Network Architecture Based on Semantic Segmentation of Remote Sensing Images

Since self-supervised pre-training requires a large amount of data as support, but a large amount of data will inevitably increase the calculation amount and consume time cost. Therefore, how to effectively optimize the self-supervised learning algorithm becomes the key to solve the cost problem of self-supervised learning. This paper proposes to optimize the self-supervised learning algorithm by using the TracIn method and sequence optimization theory, achieving the effect of using all data to train the model by only pre-training the model with the top 80% of training points that contribute the most, reducing calculation cost and time cost.

1) Training point score calculation based on TracIn method: The TracIn method identifies the overall impact of training examples by tracking the training process. Its principle is as follows: Z represents the sample space, $\boldsymbol{z}$ and $\boldsymbol{z'}$ respectively represent the training point (training sample) and test point (test sample). Given a set of k training points $\boldsymbol{S} = \{\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots \boldsymbol{z_k} \in \boldsymbol{Z}\}$, train the predictor by finding the parameters $\boldsymbol{\omega}$ that minimize the training loss $\boldsymbol{Loss} = \sum_{i=1}^{k}\boldsymbol{L}(\boldsymbol{\omega}, \boldsymbol{z_i})$ through the iterative optimization process using a training point $\boldsymbol{z_t} \in \boldsymbol{S}$ in iteration $\boldsymbol{t}$, and update the parameter vector from $\boldsymbol{\omega_t}$ to $\boldsymbol{\omega_{t+1}}$. For the training point $\boldsymbol{z} \in \boldsymbol{S}$, the loss reduction caused by the training process for a given test point $\boldsymbol{z'} \in \boldsymbol{Z}$ can be expressed as:

$$\boldsymbol{TracInIdeal}(\boldsymbol{z}, \boldsymbol{z'}) = \sum_{t:z_t=z}\boldsymbol{L}(\boldsymbol{\omega_t}, \boldsymbol{z'}) - \boldsymbol{L}(\boldsymbol{\omega_{t+1}}, \boldsymbol{z'}) \quad (8)$$

By limiting the gradient to a specific gradient descent and substituting the parameter change formula into a first-order approximation and ignoring the high-order term $\boldsymbol{O}(\boldsymbol{\eta_t^2})$, the following first-order approximation of loss change can be obtained:

$$\boldsymbol{L}(\boldsymbol{\omega_t}, \boldsymbol{z'}) - \boldsymbol{L}(\boldsymbol{\omega_{t+1}}, \boldsymbol{z'}) \approx \boldsymbol{\eta_t}\boldsymbol{\nabla}(\boldsymbol{\omega_t}, \boldsymbol{z'}) \cdot \boldsymbol{\nabla}(\boldsymbol{\omega_t}, \boldsymbol{z}) \quad (9)$$

where $\boldsymbol{\eta_t}$ represents the step length in iteration $\boldsymbol{t}$.

For a specific training point z, this approximation can approximate the idealized influence by summing this approximation over all iterations where z is used to update the parameters. This first-order approximation is referred to as $\boldsymbol{TracIn_{TBSNet}}$, as shown in equation (10):

$$\boldsymbol{TracIn}_{\boldsymbol{TBSNet}(\boldsymbol{z},\boldsymbol{z'})} = \sum_{t:z_t=z}\boldsymbol{\eta_t}\boldsymbol{\nabla}\boldsymbol{L}(\boldsymbol{\omega_t}, \boldsymbol{z'}) \cdot \boldsymbol{\nabla}\boldsymbol{L}(\boldsymbol{\omega_t}, \boldsymbol{z}) \quad (10)$$

From the above equation, it can be seen that the score of a training point $\boldsymbol{q_i}$ at a test point $\boldsymbol{q_j'}$ can be expressed as:

$$\boldsymbol{tracin}_{\boldsymbol{score}_{\boldsymbol{q_i},\boldsymbol{q_j'}}} = \boldsymbol{TracIn}_{\boldsymbol{TBSNet}\left(\boldsymbol{q_i},\boldsymbol{q_j'}\right)} \quad (11)$$

In order to verify the difference in contributions among each training point, the scores of each training point on the test point are summed up, and the final score obtained is the total score of this training point, that is:

$$\boldsymbol{Score_i} = \sum_{j=0}^{N}\boldsymbol{tracin}_{\boldsymbol{score}_{\boldsymbol{q_i},\boldsymbol{q_j'}}} \quad (12)$$

where $\boldsymbol{N}$ represents the number of test points.

2) Optimization of TBSNet training process based on sequential optimization: Following the computation of the TracIn score as outlined earlier, scores can be derived for each training point against the identical set of test points. Based on the idea in sequential optimization that "sequence is more useful than ratio," each training point's TracIn score can be seen as abstracting each training point into a sortable value. For the triple-branch network model that applies the TracIn method, the scores of each training point are sorted, and the training points with higher scores are considered to contribute more, while the training points with lower scores are considered to contribute less. Therefore, using only a certain range of higher scoring training points can achieve results comparable to training with all training points. The specific operation process is shown in Fig. 6.

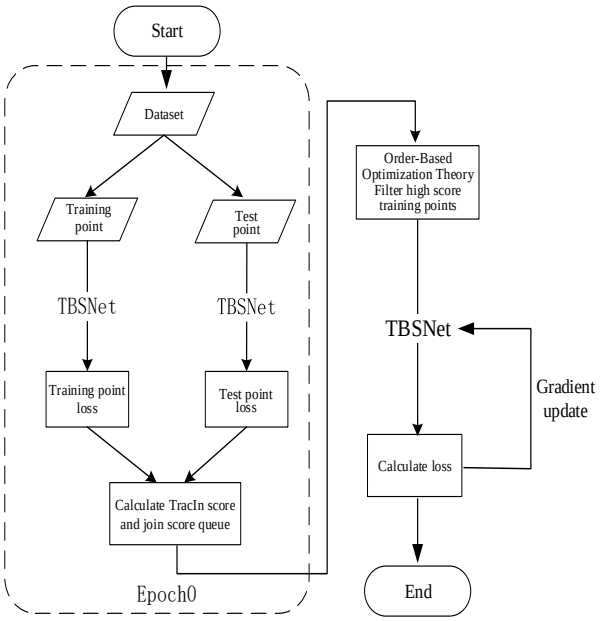

Fig. 6. Schematic diagram of training optimization process.

After the pre-training is completed, the trained neural network is transferred to the downstream task for fine-tuning.

IV. Experimental Results and Analysis

A. Dataset Selection

The dataset utilized for this study is derived from high-resolution satellite remote sensing images from the southern regions of China. To validate the generalization capability of the proposed method, this dataset amalgamates imagery from various locations and different satellite types. These datasets were uniformly re-annotated into five land cover categories: vegetation, buildings, roads, water bodies, and others. In total, it comprises 12 large-scale RGB original images ranging from 4000×4000 to 8000×8000 pixels.

Due to computational resource constraints, the original remote sensing images were segmented into patches of 256×256 pixels. Additionally, to meet the extensive data requirements for pre-training, the experimental dataset was augmented using random noise, Gaussian blur, and color transformations, resulting in an enriched dataset. Ultimately, a training sample consisting of 100,000 images was established.

B. Evaluation Metrics

For validating and evaluating the suggested approach in subsequent tasks related to remote sensing image segmentation, the performance metrics employed are Overall Accuracy (OA) and Recall. These metrics are defined in equations (13) and (14) as provided below:

$$OA = \frac{TP}{N} \tag{13}$$

$$Recall = \frac{TP}{TP+FN} \tag{14}$$

Here, TP represents the correctly predicted pixel count, or true positives. FN signifies the incorrectly predicted pixel count, or false negatives. N denotes the total number of pixels.

C. Experimental Setup and Configuration

The experiment was conducted in a Linux environment, with an Intel(R) Xeon(R) Gold 5218R CPU and NVIDIA GeForce RTX 2080Ti 11Gb GPU. Programming was done in Python 3.8 within the PyTorch framework, with Resnet50 as the backbone network and DeepLabV3+ for segmentation. The initial learning rate during the self-supervised stage was set to 0.01, batch size was 32, region size in the local contrast learning branch was 24×24, and six local areas were selected from each image. The pre-training stage was set to run for 500 epochs using the Adam optimizer. The initial learning rate for the fine-tuning stage was set to 0.005, the fine-tuning epoch was set to 50, and the training and validation sets were split in a 7:3 ratio. The selection of training and testing points for calculating the TracIn score was done per batch, and the ratio of training points to testing points was set at 5:1.

The experiment started with the computation of TracIn scores for each training point during the first training round. Following the idea of sequential optimization, the top 80% of training points with higher TracIn scores were selected for further self-supervised pre-training. Finally, after pre-training, the trained model was fine-tuned on a downstream segmentation task using a small amount of labeled data.

D. Comparative Experiment

To validate the effectiveness of our method, we compared it against several mainstream methods, including MoCo v2 [30], which uses a dynamic dictionary for contrastive learning, SimCLR, which uses data augmentation to create positive pairs for contrastive learning, the classic self-supervised learning pretext task of image inpainting, and supervised pre-training models using ImageNet for segmentation. In this paper, we used 0.5%, 1%, 5%, and 10% of the self-supervised pre-training data to fine-tune the downstream task, as shown in Table I. For different models' semantic segmentation experiments on this dataset, some experimental results are shown in Fig. 7.

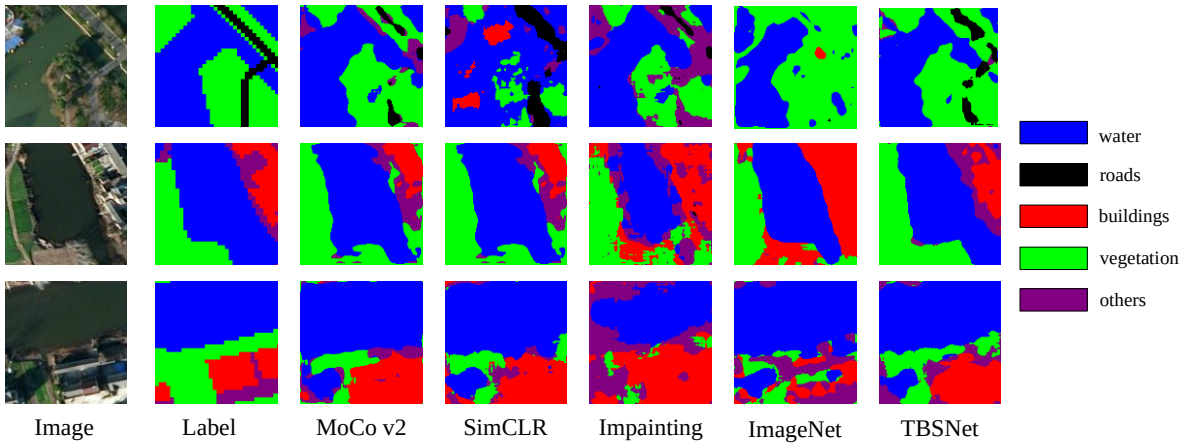

Fig. 7. Comparison experiment effect picture.

TABLE I. COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT MODELS

Fine-tune data volume	0.5%		1%		5%		10%	
Evaluation Metrics	OA	Recall	OA	Recall	OA	Recall	OA	Recall
Inpainting	0.2764	0.2593	0.4375	0.4359	0.5158	0.4783	0.5709	0.5256
SimCLR	0.3809	0.3825	0.4604	0.4289	0.5781	0.5452	0.6679	0.6623
MoCo v2	0.3654	0.3577	0.4432	0.3964	0.5295	0.5114	0.6067	0.6008
ImageNet	0.3848	0.3826	0.4128	0.4049	0.5624	0.5551	0.6792	0.6567
Ours(TBSnet)	**0.3856**	**0.3831**	**0.4721**	**0.4558**	**0.5739**	**0.5584**	**0.7139**	**0.7111**

Our results show that our method is effective for land cover segmentation in remote sensing images. With only 10% of labeled data used for fine-tuning, the overall accuracy (OA) and recall reached 0.7139 and 0.7111 respectively, representing a significant improvement over advanced self-supervised models such as MoCo v2 and SimCLR.

Analysis of the results reveals that since Inpainting mainly predicts missing areas from the image's context, it often lacks precision for complex remote sensing images, thus performing the worst in the experiments. Both MoCo v2 and SimCLR focus on global features, and their effectiveness is limited due to the lack of learning of local features in remote sensing images. Although ImageNet uses millions of natural images for pre-training, the differences in distribution, texture, and color between remote sensing images and natural images make ImageNet pre-training ineffective for downstream remote sensing image segmentation tasks. Our method performs well in learning both global and local features, demonstrating great application potential worthy of further research and exploration.

E. Ablation Experiments

1) Ablation experiments on the three-branch network structure: In the ablation experiments on the triple-branch network structure, 10% of pre-training data was used for fine-tuning. After one round of training, the top 80% of training points based on TracIn scores were selected for training. The following experiments were designed to demonstrate the effectiveness of the proposed method: using only the image rotation prediction branch (Exp1), using only the global contrast learning branch (Exp2), using only the local contrast learning branch (Exp3), using both the global and local contrast learning branches (Exp4), using the image rotation prediction branch and the global contrast learning branch (Exp5), using the image rotation prediction branch and the local contrast learning branch (Exp6), and using the complete triple-branch network (Exp7). The experiment results are shown in Table II.

TABLE II. EXPERIMENTAL RESULTS OF TRIPLE BRANCH NETWORK ABLATION

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
OA	0.5726	0.6593	0.5830	0.6845	0.6507	0.6732	0.7139
Recall	0.5658	0.6494	0.5800	0.6744	0.6457	0.6642	0.7111

The results reveal that a reasonable combination of the three branches is more conducive to the improvement of downstream task performance. The image rotation prediction branch and the global contrast learning branch can learn the overall features of the image. However, due to the lack of local feature learning, they do not achieve the best results, reaching only an overall accuracy of 0.6507 when learning global features only. Without global feature learning, solely learning local features results in an overall accuracy of only 0.5830. The best results are achieved when both global features are learned using the image rotation prediction branch and the global contrast learning branch, and local features are learned using the local contrast learning branch. Compared to the former two scenarios, the overall accuracy is improved by 0.0632 and 0.1309, respectively.

2) Ablation experiments on training process optimization methods: This work conducted training using the top 20%, 50%, 80%, 100% of training points based on TracIn scores, and without sequential optimization, randomly selected 80% of training points (Random 80%), to investigate the impact of the TracIn method and sequential optimization on experiment accuracy. The results are shown in Table III. Simultaneously, we explored the relationship between the time consumed and the overall accuracy when pre-training with different data volumes. The results are shown in Fig. 8.

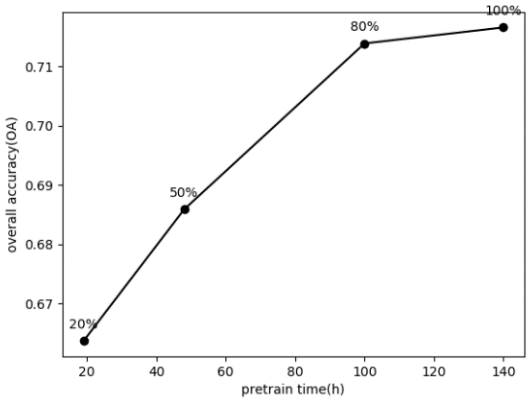

Fig. 8. The impact of different data volumes on time and accuracy.

TABLE III. OPTIMIZATION METHOD ABLATION EXPERIMENTAL RESULTS

	20%	50%	80%	100%	Random 80%
OA	0.6637	0.6859	0.7139	0.7166	0.7048
Recall	0.6599	0.6916	0.7111	0.7183	0.6985

In Fig. 8, with the increase in data volume, both the pre-training time and the overall accuracy increase. However, the ratio of overall accuracy to pre-training time (i.e., the slope of the line) keeps decreasing, indicating that the time required to improve the unit accuracy is increasing. This reflects that those with higher TracIn scores contribute significantly to accuracy improvement. When the data volume reaches the top 80% of TracIn scores, the accuracy is nearly the same as using all pre-training data (i.e., 100%), demonstrating the effectiveness of the optimization method proposed in this paper for reducing data volume. Meanwhile, as shown in Table III, using the top 80% of data based on TracIn scores also improved the results compared to randomly using 80% of the data, verifying the effectiveness of the proposed optimization method.

The experimental results show that applying the training data selected by the TracIn method and sequential optimization to the proposed triple-branch self-supervised network can reduce the data volume by 20% with almost no impact on the experiment accuracy. The reduction in data volume brings about a decrease in time cost. Therefore, for self-supervised learning, the combination of the TracIn method and sequential optimization theory is an optimization format worth considering.

V. CONCLUSION

This study introduces a self-supervised learning-based semantic segmentation technique for remote sensing images. Initially, a triple-branch self-supervised learning network known as TBSNet is developed to capture both global and local features within these images. Subsequently, the TracIn method and sequential optimization theory are employed to enhance the pre-training procedure of the self-supervised learning network, consequently reducing the time and computational resources necessary for pre-training. Ultimately, the pre-trained model is fine-tuned for downstream tasks, culminating in a semantic segmentation model tailored for remote sensing images through self-supervised learning. In comparison to traditional self-supervised models, our experiments reveal varying degrees of enhancement. But there are two notable limitations in this study. First, during the fine-tuning phase, 10% of the labeled data was utilized, thus not achieving a fully unsupervised approach. There remains potential for further reduction in the amount of labeled data used. Secondly, to simultaneously learn global and local features, a triple-branch network collaboration was employed, leading to an increase in the model's size. For future research, under the premise of further reducing labeled data, the goal is to integrate more advanced optimization techniques to develop a lighter self-supervised model and strive to further enhance segmentation accuracy.

REFERENCES

[1] J. A. Richards, and J. A. Richards, Remote sensing digital image analysis: Springer, 2022.

[2] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, “A review of semantic segmentation using deep neural networks,” International journal of multimedia information retrieval, vol. 7, pp. 87-93, 2018.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation." pp. 3431-3440.

[4] M. Pastorino, G. Moser, S. B. Serpico, and J. Zerubia, “Semantic segmentation of remote-sensing images through fully convolutional neural networks and hierarchical probabilistic graphical models,” IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-16, 2022.

[5] J. M. Alvarez, Y. LeCun, T. Gevers, and A. M. Lopez, "Semantic road segmentation via multi-scale ensembles of learned features." pp. 586-595.

[6] Zhao, C. Wang, Y. Gao, Z. Shi, and F. Xie, “Semantic segmentation of remote sensing image based on regional self-attention mechanism,” IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2021.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation." pp. 234-241.

[8] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation." pp. 801-818.

[9] X. Yuan, J. Shi, and L. Gu, “A review of deep learning methods for semantic segmentation of remote sensing imagery,” Expert Systems with Applications, vol. 169, pp. 114417, 2021.

[10] R. Liu, L. Mi, and Z. Chen, “AFNet: Adaptive fusion network for remote sensing image semantic segmentation,” IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 9, pp. 7871-7886, 2020.

[11] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and benchmarking self-supervised visual representation learning." pp. 6391-6400.

[12] W. Li, H. Chen, and Z. Shi, “Semantic segmentation of remote sensing images with self-supervised multitask representation learning,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 6438-6450, 2021.

[13] S. Albelwi, “Survey on self-supervised learning: auxiliary pretext tasks and contrastive learning methods in imaging,” Entropy, vol. 24, no. 4, pp. 551, 2022.

[14] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks." pp. 6688-6697.

[15] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks." pp. 286-301.

[16] Pruthi, F. Liu, S. Kale, and M. Sundararajan, “Estimating training data influence by tracing gradient descent,” Advances in Neural Information Processing Systems, vol. 33, pp. 19920-19930, 2020.

[17] Y.-C. Ho, R. S. Sreenivas, and P. Vakili, “Ordinal optimization of DEDS,” Discrete event dynamic systems, vol. 2, no. 1, pp. 61-88, 1992.

[18] B. Barlow, “Unsupervised learning,” Neural computation, vol. 1, no. 3, pp. 295-311, 1989.

[19] L. Wu, H. Lin, C. Tan, Z. Gao, and S. Z. Li, “Self-supervised learning on graphs: Contrastive, generative, or predictive,” IEEE Transactions on Knowledge and Data Engineering, 2021.

[20] Y. Zheng, M. Jin, Y. Liu, L. Chi, K. T. Phan, and Y.-P. P. Chen, “Generative and contrastive self-supervised learning for graph anomaly detection,” IEEE Transactions on Knowledge and Data Engineering, 2021.

[21] Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting." pp. 2536-2544.

[22] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” arXiv preprint arXiv:1803.07728, 2018.

[23] M. Noroozi, and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles." pp. 69-84.

[24] P. H. Le-Khac, G. Healy, and A. F. Smeaton, “Contrastive representation learning: A framework and review,” Ieee Access, vol. 8, pp. 193907-193934, 2020.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning." pp. 9729-9738.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations." pp. 1597-1607.

[27] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu, “Contrastive learning of global and local features for medical image segmentation with limited annotations,” Advances in neural information processing systems, vol. 33, pp. 12546-12558, 2020.

[28] P. Berg, M.-T. Pham, and N. Courty, “Self-supervised learning for scene classification in remote sensing: Current state of the art and perspectives,” Remote Sensing, vol. 14, no. 16, pp. 3995, 2022.

[29] Z. Zhao, Z. Luo, J. Li, C. Chen, and Y. Piao, “When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework,” Remote Sensing, vol. 12, no. 20, pp. 3276, 2020.

[30] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” arXiv preprint arXiv:2003.04297, 2020.