IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 November 2023, accepted 24 December 2023, date of publication 26 December 2023,
date of current version 11 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347634

RESEARCH ARTICLE

Improved Metric-Learning-Based Recognition Method for Rail Surface State With Small-Sample Data

HUIJUN YU[1], CIBING PENG[1], JIANHUA LIU[1], JINSHENG ZHANG[1], AND LILI LIU[2]
[1]School of Rail Transportation, Hunan University of Technology, Zhuzhou 412007, China
[2]School of Intelligent Control, Hunan Railway Professional Technology College, Zhuzhou 412012, China

Corresponding author: Jianhua Liu (jhliu@hut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFF0501101, in part by the National Natural Science Foundation of China under Grant 52272347, in part by the Natural Science Foundation of Hunan Province under Grant 2022JJ50095 and Grant 2021JJ30217, and in part by the Research Foundation of Education Bureau of Hunan Province under Grant 20A162.

ABSTRACT The accurate identification of rail surface states, especially the third media state, is crucial for enhancing traction and braking capabilities of heavy haul trains, ensuring their safe operation, and maintaining heavy haul railways. Few-shot learning is commonly used to recognize rail surface states, effectively addressing the overfitting issue caused by limited sample data. However, in practical rail surface state data situations, few-shot learning faces challenges such as insufficient extraction of crucial feature information and a tendency to lose distinguishing degree information. To address these challenges, this paper proposes a rail surface state recognition model based on improved metric learning. The proposed method incorporates a pyramid-splitting attention mechanism in the feature extraction network. This allows for the extraction of multi-scale spatial information from the feature map, facilitating cross-dimensional channel attention and interaction between spatial attention features. This addresses the issue of inadequate key feature information extraction caused by a limited number of orbital surface state samples. Additionally, a deep local description concatenator splices the local features of the query set and various support set feature maps in pairs, replacing the global feature splicing in traditional metric learning. This enables the filtering of interference information, such as background, while retaining feature information with significant differentiation to a larger extent. The proposed method was evaluated using a small-sample rail surface state dataset that we constructed. According to the experimental results, the proposed method outperforms existing methods in terms of recognition accuracy, precision, and recall.

INDEX TERMS Rail surface state, metric learning, pyramid-splitting attention, local description concatenator.

I. INTRODUCTION

The performance of traction/braking controls in rail transit trains depends on the behavior of the wheel–rail contact, which is affected by the state of the rail surface [1], [2], [3]. Accurately identifying the rail surface state is crucial in order to ensure high-performance train control [4], [5]. Most of the current rail surface state recognition methods are judged using human experience, but these face problems such as low recognition efficiency and poor real-time performance. The introduction of deep learning methods based on big data is expected to improve the accuracy of rail surface state identification. However, due to the limitations imposed by the collection line, the collected samples exhibit minimal variation, and a significant portion of the samples are repetitive, resulting in a reduced number of available samples. Moreover, certain rare states, such as oily, occur infrequently during the collection process, further limiting

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez.

the number of samples representing these states. Additionally, weather conditions significantly impact the duration of data collection. In sunny weather, the number of wet samples is relatively small, whereas during rainy days, the number of dry samples is inadequately represented. Obtaining railway collection data is subject to the approval from relevant authorities and railway operators, along with the consideration of limited railway operation hours.

Consequently, limited time for data collection poses challenges in obtaining a sufficient amount of data. Furthermore, environmental factors such as lighting, climate, and the constraints of the collection equipment contribute to the acquisition of low-quality images, which further restrict the available sample size. In summary, these factors collectively result in an insufficient number of available rail surface samples, presenting challenges in achieving accurate deep learning-based recognition of rail surface states. Therefore, addressing the issue of accurate image recognition of rail surface states with small sample data is a difficult problem that requires immediate attention.

When dealing with the image recognition problem in the context of small sample data, there are typically two potential solutions: data augmentation and small-sample learning. Data augmentation achieves the expansion of the sample size by transforming and expanding the existing rail surface state image to generate new rail surface state data [6], [7]. However, this method is not only expensive but also generates some unreal data, which has a serious impact on the model recognition performance [8]. Few-shot learning primarily encompasses methodologies such as meta-learning, transfer learning, contrastive learning, and metric learning [9], [10], [11]. Meta-learning is a method by which to quickly adapt to new tasks using learned information, but it relies too much on prior knowledge and has limited adaptability to new tasks [12], [13], [14]. Transfer learning uses a pre-trained model and fine-tunes model parameters to achieve small-sample recognition tasks. However, its performance is limited by the source domain dataset, and the recognition of unknown tasks cannot be achieved [15], [16]. Contrastive learning, a self-supervised learning method, aims to bring similar samples closer and separate dissimilar samples for effective classification. However, it exhibits high sensitivity to noise and abnormal samples, and incurs considerable computational costs [17], [18]. Metric learning measures the distance between the support set and query set samples to obtain the similarity score between the two, and compares the similarity scores to complete the recognition of small samples [19], [20], [21], [22]. Compared to meta-learning, transfer learning, and comparative learning, metric learning demonstrates superior data modeling ability. It effectively captures the internal relationships and structures among samples, enabling more accurate expression and comparison. Furthermore, metric learning exhibits strong adaptability and scalability, allowing for flexible adjustments and expansions tailored to specific tasks and problems. This enhances the model's generalization performance. Additionally, metric learning showcases excellent reasoning ability, enabling comparison and classification of new samples without the need for global re-training. This ensures real-time performance while maintaining accuracy [23]. Based on the advantages of the metric learning framework mentioned above, and considering the characteristics of the rail surface sample dataset, it can be concluded that metric learning is a more suitable learning framework for small-sample rail surface state recognition tasks.

With the rapid development of few-shot learning, research on metric learning has received extensive attention from scholars at home and abroad. Koch et al. [24] proposed a Siamese network that utilizes an embedding function to map inputs to the target space and employs the Euclidean distance function for similarity calculation. However, this method is primarily applicable in scenarios involving numerous categories and limited samples per category. Its performance is not satisfactory in cases where the number of categories is smaller. Hao et al. [25] introduced a semantic alignment metric learning method, which utilizes a relationship matrix to capture the distances between the query set and the support set. Subsequently, a multi-layer perceptron (MLP) is employed for similarity calculations to enable classification. However, the utilization of a significant number of MLP parameters increases the risk of overfitting and leads to model instability. Li and Ralescu [26] proposed a supervised metric learning method that concurrently learns distances in geometric and probabilistic spaces. However, the approach heavily relies on a substantial amount of annotated training data, leading to high acquisition costs. Consequently, its performance may be limited when applied to specific scenarios with small sample sizes. Snell et al. [27] proposed a prototype network that uses the mean value of each category mapped to the feature space to represent the prototype of the category and uses the Euclidean distance formula to measure the distance between the query sample and the prototype to achieve sample category prediction. Sung et al. [28] proposed a relational network based on the prototype network, using a CNN instead of the fixed distance function to obtain the best distance metric method for category judgment. The network reduces the computational complexity of the model and improves the general performance. Li et al. [29] proposed an end-to-end covaryance metric network that realizes small-sample recognition through covaryance representation and covaryance metric based on distribution consistency. The approach reduces computational requirements without sacrificing accuracy. However, these three methods have two main issues. Firstly, the high dimensionality of the feature extraction network in the initial stage hinders effective extraction of available feature information. Secondly, during feature splicing, the abstract feature maps are transformed into position-insensitive vectors, resulting in a significant loss of distinguishing information. When employing the aforementioned methods for small-sample rail surface state

identification, it is common for existing defects to undergo amplification, leading to a notable decline in the performance of identification.

To overcome these limitations and enhance model recognition performance, this paper presents an rail surface state recognition method based on improved metric learning tailored for small sample scenarios. Traditional metric learning methods typically comprise three components: feature extraction (embedding module), feature concatenation, and metric module. The feature extraction component commonly utilizes convolutional neural networks to extract features, while feature concatenation involves directly adding global features for splicing. The metric module selects a suitable fixed distance formula for metric based on different scenarios. This article aims to improve all three components. In the feature extraction network part, the pyramid-splitting attention mechanism [30] is introduced to enrich the feature space by capturing the spatial information of different scales, and establishing long-distance channel dependence on local area information, so that richer multi-scale features can be extracted. In the feature splicing part, the local description concatenator (LDC) [31] is used to splice the local features of the sample set and the query set in pairs. This method removes background and interference information from feature maps while retaining their significant distinguishing information to a large extent. In the metric part, the convolutional neural net-work is used to replace the fixed metric formula to realize the fitting metric of the combined feature map. Finally, the proposed method is applied to the self-built small sample rail surface state data set, and experiments are conducted to verify and analyze the results.

II. RELATED WORK

A. RAIL SURFACE STATUS IDENTIFICATION

With the rapid development of heavy-haul railways, concerns regarding the reliability, safety, and stability of railway rail systems have escalated. Accurate monitoring of rail surface states is a crucial factor in ensuring train safety, braking efficiency, and operational effectiveness within the rail transit system. Particularly in complex meteorological and environmental conditions, the state of the rail surface becomes pivotal as it directly influences the adhesion characteristics of the wheel-rail contact area, significantly affecting train traction and braking performance. The presence of third-party media, such as water or oil, on the rail surface is closely linked to the adhesion of the wheel-rail contact area. These extraneous substances substantially reduce adhesion, leading to safety risks like sliding and derailment. Consequently, real-time and accurate identification of rail surface states becomes paramount in maintaining the safe operation of trains [2], [3], [4], [5].

Both domestic and foreign scholars have conducted research on rail surface identification, particularly in the detection and recognition of rail surface defects, resulting in significant achievements. Dubey and Jaffery [35] proposed a visual inspection method using the Maximum Stable Extreme Region (MSER) technology. This approach effectively rails changes in railway rail images, identifies geometric features of defective areas, and visualizes them. Ni et al. [36] developed an algorithm that detects rail surface defects using Partitioned Edge Features (PEF), which effectively mitigates the impact of uneven lighting. He et al. [37] proposed an improved feature pyramid network and metric learning method, employing deformable convolution and convolutional block attention modules to enhance the transformation of FPN and improve the accuracy of detecting defects at various levels. Additionally, Yu et al. [38] adopted a coarse-to-fine strategy for rail surface defect detection. Their method involves using a coarse extractor to approximate the location of defects in rail surface images and a fine extractor to finely classify and identify these potential outliers. However, the aforementioned research methods primarily focus on rail surface defect detection and are not applicable to identifying the state of third-party media on the rail surface. Zhang et al. [39] proposed a rail surface state identification method based on the BP-Adaboost algorithm using real-time adhesion state data. Nevertheless, this method relies heavily on large amounts of data for support. In comparison, the method we propose does not necessitate extensive data support and offers a better solution for identifying the state of third-party media on the rail surface within a small-sample data scenario.

B. METRIC LEARNING

In recent years, researchers have conducted many studies on small-sample image recognition tasks based on metric learning and achieved abundant results [19], [20], [21], [22], [24], [25], [26], [27], [28], [29]. The goal of metric learning is to map similar samples to close distances and dissimilar samples to farther distances by learning an appropriate metric or similarity function. In small-sample image recognition, the number of samples in the data set is limited, and there are often problems of small differences within classes and large differences between classes, which can easily lead to poor final recognition results. Metric learning can better solve this problem by clustering similar samples together.

Metric learning methods can generally be categorized into two types: prototype-based and distance-based metric learning. The former involves learning prototypes or representative samples within a dataset, and employs the nearest neighbor principle for classification or regression tasks to cope with limited-sample scenarios. The latter, on the other hand, focuses on learning an optimal distance metric by assessing the distance or similarity between pairs of samples for similar small-sample challenges. In metric learning-based small-sample image recognition, researchers strive to refine the metric learning algorithms to effectively elevate the performance of small-sample recognition models. For instance, Gao et al. [19] introduced a multi-distance metric network (MDM-Net), which accounts for shallow features and employs a multi-output embedding network to map samples

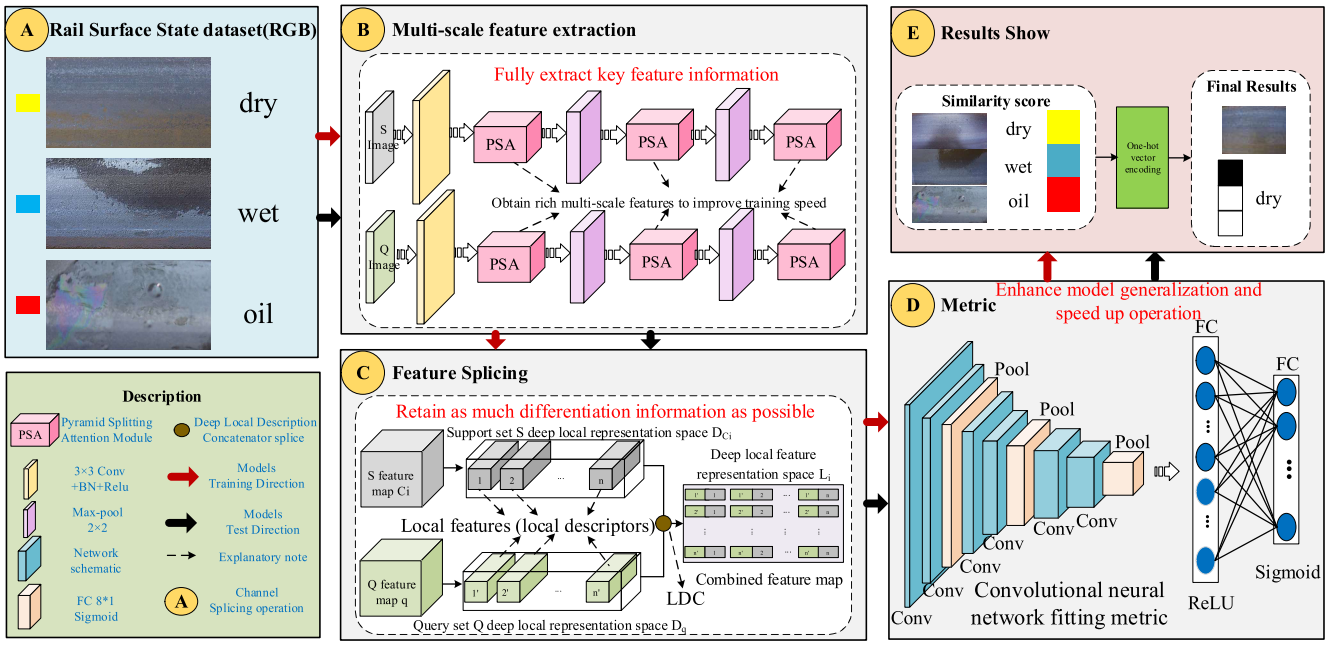

FIGURE 1. Block diagram of improved metric-learning-based rail surface condition recognition model with small-sample data.

across various feature spaces. Huang et al. [40] developed a local multi-prototype network (LMPNet) utilizing local descriptors to characterize images and mitigate the influence of sample variations on prototypes through channel squeeze and spatial excitation (SSE) attention modules to address the issues of uncertainty. Meanwhile, Wu et al. [41] presented Position-Aware Relation Networks (PARN), which incorporate the attention mechanism not only to assess the correlation matrix between the class representation and the query sample but also to consider the autocorrelation matrix. They concatenate all correlation matrices for input into a network that learns the ultimate similarity measurements.

In addition, Nguyen et al. [42] proposed the use of the Euclidean distance and the square root of the sum of normed distances as the distance measurement function for calculating the distance between the query set sample features and the class prototype for classification prediction. Li et al. [31] introduced the DN4 model, which employs the original local descriptor set to represent the query image and support classes, and then employs cosine similarity to measure the similarity between the images. Zhang et al. [43] proposed an Earth Mover's Distance (EMD) metric function that applies different weights to different positions of the image and calculates the best matching method between each patch in the support set and query set to represent the similarity between the two. However, specific challenges are encountered when applying the aforementioned methods to the task of identifying small-sample rail surface state images. Firstly, the intra-class differences in rail surface state images are minimal, whereas the inter-class differences are significant, which leads to the extracted features being less distinguishable and subsequently affects the recognition accuracy. Secondly, the cited methods predominantly employ fixed techniques for distance measurement that lack adaptability for this particular task, further diminishing the model's recognition capabilities. In contrast, our study enhances the model's performance by integrating the pyramid-splitting attention (PSA) mechanism into a convolutional neural network (CNN), thereby achieving efficient extraction of multi-scale features from rail surface images. Regarding the distance metric, our approach utilizes a CNN model for adaptive fitting rather than the fixed distance metric formula, significantly improving the rail surface state recognition model's performance.

III. RAIL SURFACE STATE RECOGNITION FRAMEWORK

Considering the influence of weather conditions and train operating environment, as well as the specific conditions of a particular section of the railway, we categorize the condition of the rail surface into three states: dry, wet, and oily. In this paper, we focus on small-sample rail surface state image data, and we construct a rail surface state recognition model based on improved metric learning, as shown in Fig. 1.

Fig.1 shows a model framework consisting of modules A to E. Module A represents the rail surface state dataset, which is used for model training and testing. Module B is a multi-scale feature extraction network based on the fusion of convolutional network and pyramid split attention mechanism. By introducing the pyramid splitting attention mechanism, module B can effectively learn multi-scale features, and realize multi-scale high-dimensional feature information extraction of query set and support set input images. Module C is a feature splicing module, which regards the feature map generated by the feature extraction network as a combination of multiple local descriptors. Then, by using

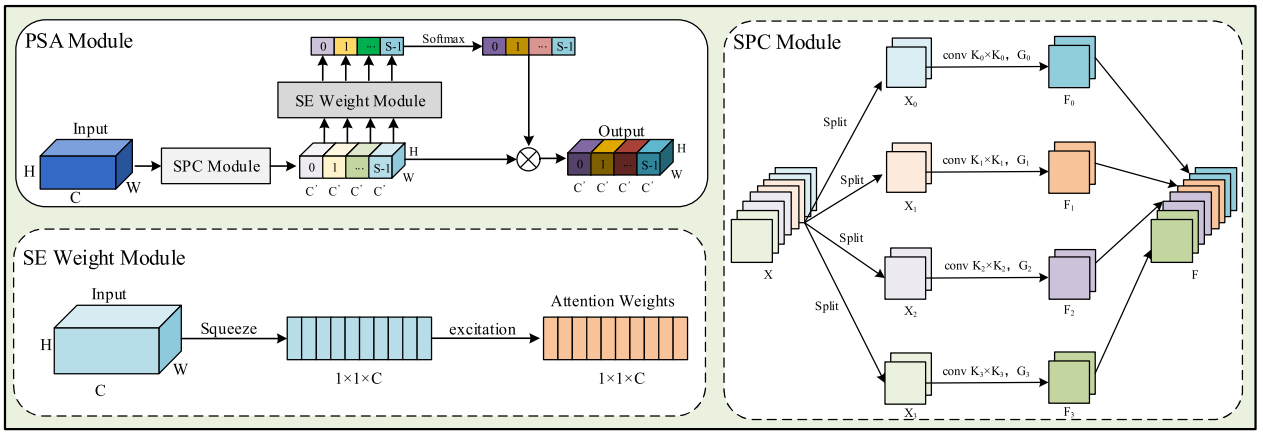

FIGURE 2. PSA module network structure.

the deep local feature concatenator LDC, the local descriptors of the corresponding positions in the query set and the C-type support set feature map are spliced, and the combined feature map of the query set and the support set is obtained. Module D is a metric module, which obtains the similarity score between the query set and the class C support set by inputting the combined feature map into the convolutional neural network for fitting metric. Finally, module E is a result display module, which realizes the output of rail surface status recognition results by one-hot encoding of similarity scores.

A. MULTI-SCALE FEATURE EXTRACTION NETWORK

The rail surface state dataset has the characteristics of a small number of samples and high similarity of samples of some categories. Using the traditional metric-learning 4-layer convolution block for feature extraction may result in limited effective feature information extraction, leading to reduced model recognition performance. In this paper, based on the traditional metric learning feature extraction network, a pyramid-splitting attention (PSA) network mechanism is introduced. This enables the feature extraction network to learn richer multi-scale features and perform adaptive feature recalibration on multi-dimensional channel attention weights, improving the model performance and training speed. The last three layers of convolutional blocks in the traditional metric learning feature extraction network are replaced by PSA modules. Namely, the multi-scale feature extraction network consists of one layer of convolutional blocks, three layers of PSA modules, and two layers of maximum pooling layers. Among them, the PSA module learns the correlation between different regions in the feature map and captures the spatial information of different scales. This enriches the feature space and results in a multi-scale feature map with richer feature information. The network structure of the PSA module is shown in Fig. 2.

The PSA module network structure shown in Fig. 2 includes three parts: split and concatenate (SPC) module, SE weight module, and softmax module. The segmentation operation of the SPC module divides the rail plane state feature graph X (size of C×H×W) into four parts.C is the number of channels, H is the height, and W is the width. Therefore, the subfeature diagram X_i of four orbital plane states is obtained, $X_i \in R^{C' \times H \times W}$, i = 0, 1, 2, 3, and the number of channels is C′, $C' = C/4$. The grouping convolution of the multi-scale convolution kernel is used to extract the multi-scale spatial feature information of the sub-feature map X_i, and the extraction formula is as follows:

$$F_i = Conv\left(K_i \times K_i, G_i\right)\left(X_i\right), i = 0, 1, 2, 3 \tag{1}$$

where K_i is the size of the convolution kernel, $K_i = 2 \times (i + 1)$, G_i is the size of the grouping, $G_i = 2^{(K_i-1)/2}$, and F_i is the feature map of different scales obtained after multi-scale convolution kernel extraction, $F_i \in R^{C' \times H \times W}$.

After the splicing operation of the SPC module, the feature map F_i is multi-scale fused to obtain the multi-scale fusion feature map F in the channel direction:

$$F = Cat\left(\left[F_0, F_1, F_2, F_3\right]\right) \in R^{C' \times H \times W} \tag{2}$$

The channel attention weights of different scale feature maps F_i are extracted through the SE weight module, and the attention weights Z_i in each channel direction are obtained:

$$Z_i = SEWeight\left(F_i\right) \in R^{C' \times 1 \times 1}, i = 0, 1, 2, 3 \tag{3}$$

Cascading operations are performed on Z_i in each channel direction to obtain the multi-scale channel attention weight vector Z:

$$Z = Z_0 \oplus Z_1 \oplus Z_2 \oplus Z_3 \tag{4}$$

Then, the softmax module recalibrates the attention weight Z_i in each channel direction and establishes the long-distance channel attention dependence relationship. This realizes the information interaction between channel multi-scale attention and obtains the recalibration

attention weight att_i:

$$att_i = Soft\ \max(Z_i) = \frac{\exp(Z_i)}{\sum_{i=0}^{3} \exp(Z_i)} \tag{5}$$

A dot-product operation is performed on channels between the different-scale feature maps F_i obtained from the SPC and att_i to obtain the multi-scale channel attention-weighted feature maps Y_i:

$$Y_i = F_i \odot att_i, i = 0, 1, 2, 3 \tag{6}$$

Finally, the stitching of the multi-scale channel attention-weighted feature map Y_i is obtained via the dimension splicing operation, resulting in a multi-scale refined feature map Y with richer information:

$$Y = Cat([Y_0, Y_1, Y_2, Y_3]) \tag{7}$$

B. FEATURE SPLICING MODULE

Due to the different sampling positions of the same category of rail surface state images, the local area features of the images are quite different. Using the global feature vector stitching method in traditional metric learning not only results in a loss of discriminative information but also leads to the problem of intra-class differences and background confusion. To address these issues, we employ local descriptors, also known as local features, instead of global feature vectors. A deep local description concatenator is introduced to perform deep local splicing on the local descriptors of the support set and query set feature maps. This approach helps to retain characteristic information with a significant degree of distinction to a large extent.

The rail surface state samples of the support set and the query set pass through the feature extraction network to obtain feature maps Ci and q of size h×w×c (where h is height, w is width, and c is the number of channels). The two feature maps can be viewed as a collection of n local descriptors, where n is equal to the product of the height (h) and width (w) of the maps. Each local descriptor corresponds to a local feature and has feature dimension c. Its form is as follows:

$$D = [d_1, d_2, \ldots, d_n] \in R^{c \times n} \tag{8}$$

where d_i represents the i-th local descriptor, and D represents the deep local feature representation space. The rail surface state samples in the support set or query set are subjected to the feature extraction network to obtain a feature map with a size of 64 × 11×11. After the above operations, 121 local descriptors with a dimension of 64 are obtained. The deep local description concatenator $\psi(\cdot)$ is used to splice the local descriptors of the query set samples and the support set samples in pairs, resulting in the creation of a deep local descriptor connection space L_i. This space essentially represents a combined feature map. The splicing relationship can be seen as follows:

$$L_i = \psi\left(D_{C_i}, D_q\right) = D_{C_i}^T \otimes D_q, i \in [1, 2, 3] \tag{9}$$

where D_{C_i} and D_q represent the deep local feature representation space of the support set and query set, respectively. The size of the connection space L_i is n×n×1, and each element in L_i is the outer product of the transposed matrix of the local descriptor of the rail surface state sample in the support set and the corresponding matrix of the local descriptor of the rail surface state sample in the query set. The connection space L contains the splicing results of every two local descriptors of the query set sample and the support set sample, and it shows the relationship between the query set and the support set sample in units of local features.

C. METRICS MODULE

The metrics module utilizes a CNN to achieve adaptive fiting metrics for the combined feature map of the rail surface state, rather than relying on traditional fixed distance metrics functions. This approach not only eliminates the need for selecting fixed functions but also allows the rail surface state recognition model to obtain a more appropriate metrics function. Ultimately, the approach improves the model's generalization ability to some extent. The module consists of 6 convolutional blocks, 3 pooling layers, and 2 fully connected layers. Each convolutional block includes a convolutional layer, a batch normalization layer, and a ReLU linear activetion layer. The convolutional layer includes a 64-channel 3∗3 convolution kernel with a step size of 1 and zero padding of 0. The pooling layer is implemented using the maximum pooling method, the pooling window size is 2∗2, the step size is 2, and the zero padding is 0. The two fully connected layers are 8-dimensional and 1-dimensional; the 8-dimensional fully connected layer contains The ReLU activation function and the 1-dimensional fully connected layer contains the sigmoid activation function. After undergoing convolution pooling, the combination feature map of the rail surface state produces a similarity score that ranges from 0 to 1 through the fully connected layer. By comparing these similarity scores, the recognition result can be determined.

D. LOSS FUNCTION

After the connection space is processed by the metric module, a similarity score ranging from 0 to 1 is obtained. This score can be considered as a training objective similar to regression problems. To train the model, we choose SmoothL1 as the loss function. The SmoothL1 loss function is effective in avoiding the gradient explosion phenomenon and improving the stability of the rail surface state model. The expression for SmoothL1 is as follows:

$$SmoothL1 = \frac{1}{n} \sum_{i=1}^{n} g_i \tag{10}$$

$$g_i = \begin{cases} 0.5\left(f\left(x_i\right) - y_i\right)^2, & \left|f\left(x_i\right) - y_i\right| < 1 \\ \left|f\left(x_i\right) - y_i\right| - 0.5, & otherwise \end{cases} \tag{11}$$

Among them, y_i represents the real value of the input sample, while $f(x_i)$ represents the predicted value.

IV. EXPERIMENTS

In this we evaluate the feasibility of the proposed model in recognizing the small-sample rail surface state. The section consists of three main parts: experimental conditions (includeing the dataset used, experimental environment, and evaluation indicators), model training, and experimental analysis.

A. EXPERIMENTAL CONDITIONS

1) RAIL SURFACE STATE DATASET

The rail surface state dataset used in this experiment comes from a certain railway section, including 141 images of three kinds of rail surface state images under different working conditions: dry, wet, and oil. Among these images, 52 were dry samples, 46 were wet samples, and 43 were oil samples. Due to the interference of man-made noise and environmental factors during the acquisition of rail surface state images, it is difficult to guarantee image quality. To eliminate the influence of these disturbances, preprocessing operations such as cropping, denoising, geometric correction, and morphological processing were performed on the images. This was done to obtain rail surface state data with better image quality. To ensure consistency in the dataset, the image size was adjusted to 84×84, resulting in the final self-built rail surface state dataset. Some typical samples in the rail surface status dataset are illustrated in Fig. 3.

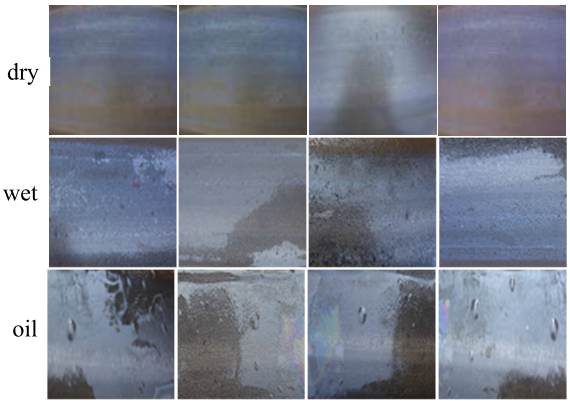

FIGURE 3. Samples of rail images in the rail surface status dataset.

To evaluate the effectiveness of the proposed model, we divided the dataset into training and test sets at a ratio of 3:2. This resulted in 85 training set samples and 56 test set samples. The training and test sets were further divided into 3 categories: dry, wet, and oil. The dry category had 30 training set samples and 19 test set samples, the wet category had 27 training set samples and 18 test set samples, and the oil category had 26 training set samples and 17 test set samples.

2) EXPERIMENTAL ENVIRONMENT

Experiments were conducted using the Windows 10 operating system. The CPU was configured with an Intel(R)-Core(TM) i5-12490F 3GHz, while the GPU was configured with an NVIDIA GeForce RTX-2080Ti. The code-running environment was Python 3.6 with CUDA = 10.0 and cuDNN = 7.6.0.64. The code-running framework used was PyTorch 1.5.0. To address the limited data samples available for the rail surface state, we utilized pre-training models in all modules of the training process. The initial learning rate was set at 10-3, with cosine attenuation used for learning rate attenuation. The optimizer of choice was Adam, and the model was iteratively trained 200,000 times across 1000 epochs. Additionally, the model was tested 600 times, with input images sized at 84×84.

3) EVALUATION INDICATORS

To evaluate the recognition performance of the proposed model, we utilized several evaluation indicators, including accuracy rate (Acc), precision (P), recall rate (R), and F1 value. The accuracy rate (Acc) provides an overall measure of how well the model performs. Precision (P) measures the accuracy of the model's recognition, while recall rate (R) measures the model's ability to correctly identify all relevant instances. The F1 value is a combined metric that takes both precisions and recalls into account, providing an overall measure of the model's performance. These metrics are calculated as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$P = \frac{TP}{TP + FP} \tag{13}$$

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

$$F1 = \frac{2 * P * R}{P + R} \tag{15}$$

where TP refers to true positive, FP refers to false positive, TN refers to true negative, and FN refers to false negative. TP+TN indicates the number of correctly predicted instances, while TP+TN+FP+FN indicates the total number of instances. TP+FP represents the number of predicted positive instances, while TP+FN represents the total number of actual positive instances.

B. MODEL TRAINING

We utilized the metric learning framework and trained the model using the N-way K-shot scenario task training strategy. Here, N represents the number of sample categories in the support set and K represents the number of samples of each type in the support set. The training set for the rail surface state was split into a support set, S, and a query set, Q. Both S and Q contained three sample categories. Namely, K images of the rail surface state were selected at random to form the support set, while Q images were randomly chosen from the remaining samples to create the query set. According to the sample category and the number of samples in S, we used a 3-way K-shot to train the model.

The model uses the support set S and query set Q as inputs. Then, the feature extraction network is used to obtain the

TABLE 1. Effect of pre-training on model performance.

Method	Pre-training	Acc(%)	P(%)	R(%)	F1 (%)
MSML [20]	Y	85.26	85.38	85.42	85.40
	N	74.18	74.29	74.21	74.25
MAML[32]	Y	73.28	73.98	73.16	73.57
	N	58.89	60.54	59.68	60.11
PN[27]	Y	81.67	81.71	82.45	82.08
	N	72.41	71.38	72.32	71.85
RN[28]	Y	84.73	85.54	84.89	85.26
	N	73.92	75.51	73.87	74.69
DN4 [31]	Y	87.85	87.76	86.86	87.16
	N	76.33	76.96	75.34	76.15
PHR [33]	Y	88.76	88.81	88.57	88.69
	S	78.29	78.43	77.86	78.14
LLSTN [34]	Y	90.27	90.86	90.13	90.49
	N	80.20	80.94	80.52	80.73
Proposal-method	Y	92.46	92.99	92.35	92.67
	N	82.98	83.96	82.71	83.33

feature map C_i (where i=1,2,3 for this study) and q. In the 3-way 1-shot scenario, the local descriptors of C_i and q are concatenated using the deep local description concatenator $\psi(\cdot)$ to obtain the deep local descriptor connection space L_i. Specifically, $L_i = \psi(C_i, q)$. To obtain the similarity score θ_{iq} of the feature space L_i, the connection space L_i is input into the metrics module $m_\phi(\cdot)$, which results in $\theta_{iq} = m_\phi(L_i)$. Finally, one-hot vector encoding is performed on the similarity score θ_{iq} to obtain the recognition result of the query set image. In the case of a 3-way K-shot (where K is not equal to 1), the feature maps obtained after all samples of different categories in the support set (S) pass through the feature extraction network are summed by class to obtain the superimposed feature maps (C1 to C3) of the three classes. The remaining steps are consistent with a 3-way 1-shot.

C. EXPERIMENTAL ANALYSIS

1) EFFECT OF PRE-TRAINING ON MODEL PERFORMANCE

To assess the influence of pre-trained model parameters on the performance of small-sample rail surface state recognition models, we carried out comparative experiments using a self-constructed rail image dataset. The experimental subjects comprised the proposed model and commonly employed small-sample learning models, including the Prototype network [27], MAML model [32], Relationship network [28], PHR model [33], LLSTN model [34], MSML model [20], and DN4 network [31]. The models were trained separately using pre-trained parameters and random parameters (non-pretrained parameters), and the experimental results were recorded, as detailed in Table 1. The pre-trained parameters were acquired through training on the miniImageNet dataset, under the experimental condition of 3way-1shot.

Table 1 illustrates a notable enhancement in recognition accuracy, precision, recall rate, and F1 value among all models utilizing pre-training weights, when compared to the non-pre-training model. This suggests that the pre-training model outperforms the non-pre-training model in recognizing the rail surface state. Our model has shownsignificant improvements in recognition accuracy, precision, recall rate, and F1 value when pre-training weights are utilized. Specifically, the pre-training weights resulted in a 9.48% increase in recognition accuracy, a 9.03% increase in precision, a 9.64% increase in recall rate, and a 9.34% increase in F1 value compared to models without pre-training. In the case of a small sample size for rail surface state data, the limited amount of data used in the learning process can lead to over-fitting and a subsequent local optimal problem. The aforementioned results demonstrate that utilizing pre-training weights can mitigate this issue and enhance the model's efficacy.

2) COMPARISON EXPERIMENTS FOR DIFFERENT ATTENTION MECHANISMS

In order to validate the effectiveness and superiority of the pyramid split attention (PSA) module in the rail surface state recognition model, the convolutional neural network (CNN) of the feature extraction module was enhanced by incorporating the PSA mechanism. A comparative experiment was conducted with a CNN network lacking the attention mechanism to ascertain the effectiveness of the PSA module. Furthermore, various attention mechanisms, including SE [44], CBAM [45], SA [46], NL [47], and ECA [48], were introduced into the CNN network of the feature extraction module. The impact of these attention mechanisms on model performance was evaluated to determine the superiority of the PSA module. The experimental results are presented in Table 2. Among them, all the datasets utilized in the experiments are selfconstructed rail surface state datasets, while the experimental conditions were set as 3way-5shot.

Based on the findings presented in Table 2, the inclusion of the PSA module has been observed to yield a substantial enhancement in model performance, in comparison to the scenario where no attention module is incorporated. This outcome serves as empirical evidence to validate the effectiveness of the PSA attention module. Moreover, when compared to other attention modules, the introduction of the PSA module yields improved performance across all

TABLE 2. Comparison experiment of attention module.

Method	Acc(%)	P(%)	R(%)	F1 (%)
-	90.05	90.29	90.43	90.36
CBAM[45]	95.67	95.87	95.49	95.68
SA[46]	91.94	93.43	92.34	92.86
NL[47]	93.24	94.01	93.82	93.77
ECA[48]	94.63	95.15	94.78	94.96
SE[44]	92.95	93.69	91.38	92.92
PSA	**97.96**	**98.61**	**98.07**	**98.34**

indicators in the model. This clearly demonstrates the superiority of the PSA attention module. The rail surface state recognition model utilizes the PSA module, resulting in increases in recognition accuracy, precision, recall rate, and F1 value indicators by 2.29%, 2.74%, 2.58%, and 2.66% respectively, when compared to the CBAM module. Additionally, these indicators are 6.02%, 5.18%, 5.77%, and 5.48% higher than those of the SA module; 4.72%, 4.61%, 4.25%, and 4.57% higher than those of the NL module; and 3.33%, 3.46%, 3.29%, and 3.38% higher than those of the ECA module. Furthermore, the indicators are 5.01%, 4.92%, 6.69%, and 5.44% higher than those of the SE module. After comparing the results between the spatial-channel hybrid attention module (CBAM, PSA) and the non-hybrid attention module (SE, SA, NL, ECA), we found that the hybrid performance indicators were generally higher than those of the non-hybrid module. This suggests that the spatial-channel hybrid attention mechanism has a better recognition effect on the model than a single attention mechanism.

3) COMPARATIVE EXPERIMENTS FOR DIFFERENT FEATURE SPLICING METHODS

To validate the superiority of the proposed deep local splicing method, we conducted comparative experiments with add1, add2, and add3. Add1 involves the direct addition of the global feature vector of the query set and the global feature vector of the support set. Add2 combines the global feature vector of the query set and the global feature vector of the support set to form a complex feature vector. Add3 adds the local feature vector of the query set and the local feature vector of the support set directly. Table 3 presents the experimental results under the 3-way 5-shot condition.

TABLE 3. Comparative experiments for different feature splicing methods.

Method	Acc(%)	P(%)	R(%)	F1 (%)
add1	85.23	85.94	86.08	86.01
add2	80.92	81.22	80.57	80.89
add3	90.88	91.34	90.99	91.16
Proposal -method	**97.96**	**98.61**	**98.07**	**98.34**

Table 3 clearly demonstrates that the deep local splicing method proposed in this paper outperforms other feature splicing methods in terms of the rail surface state recognition model. The model's indicators are superior to those of the compared methods, thus confirming the superiority of the deep local splicing method. The model's recognition accuracy, precision, recall rate, and F1 value have all shown improvement when compared to add1, add2, and add3. Specifically, there was a 12.73% increase in recognition accuracy, a 12.67% increase in precision, an 11.99% increase in recall rate, and a 12.33% increase in F1 value when compared to add1. When compared to add2, there was a 17.04% increase in recognition accuracy, a 17.39% increase in precision, a 17.50% increase in recall rate, and a 17.45% increase in F1 value. Finally, when compared to add3, there was a 7.08% increase in recognition accuracy, a 7.27% increase in precision, a 7.08% increase in recall rate, and a 7.18% increase in F1 value.

4) COMPARISON EXPERIMENTS FOR METRIC METHODS

To test the superiority of the proposed convolutional neural network fitting metrics method, we conducted comparative experiments using six commonly used metric methods: EBD [49], Euclidean distance [50], Mahalanobis distance [51], cosine similarity [52], Manhattan distance [53] and UDML [54]. The experimental results under the 3-way 5-shot condition are presented in Table 4.

TABLE 4. Comparison experiments for metric methods.

Method	Acc(%)	P(%)	R(%)	F1 (%)
EBD[49]	96.18	96.88	96.62	96.75
Euclidean [50]	95.02	95.18	94.96	95.07
Mahalanobis [51]	86.33	86.52	86.56	86.54
Cosine similarity [52]	81.95	82.98	82.81	82.89
Manhattan [53]	93.58	93.75	93.64	93.70
UDML[54]	93.66	93.72	93.83	93.77
Proposal -method	**97.96**	**98.61**	**98.07**	**98.34**

Table 4 demonstrates that the convolutional neural network fitting metrics method outperforms other methods when applied to the rail surface state recognition model. The model's indicators are superior to those of the comparison metric methods, thereby confirming the superiority of the convolutional neural network fitting metrics method in our model. Compared to the Euclidean distance, the model exhibited improvements in recognition accuracy, precision, recall rate, and F1 value indicators by 2.94%, 3.43%, 3.11%, and 3.11%, respectively. Further, when compared to the Mahalanobis distance, the indicators demonstrated increases of 3.27%, 11.63%, 12.09%, 11.51%, and 11.8%, respectively. Similarly, compared to the cosine similarity, the indicators showed improvements of 16.01%, 15.63%, 15.26%, and 15.45%, respectively. Additionally, in comparison to the Manhattan distance, the indicators saw enhancements of 4.38%, 4.86%, 4.43%, and 4.64%, respectively. In contrast to the UDML, the indicators displayed improvements of 4.30%, 4.89%, 4.24%, and 4.57%, respectively. Lastly, when compared to the EBD, the indicators exhibited improvements of 2.30%, 2.89%, 2.26%, and 2.58%.

5) COMPARISON EXPERIMENTS

To validate the effectiveness and superiority of the model proposed in this paper for detecting the rail surface state, we conducted comparative experiments with some popular small-sample learning methods (MAML, Prototype Network, Relational Network, PHR, DN4 network,MSML,LLSTN) on a self-built rail surface state dataset. The experiments were conducted under the 3-way 5-shot condition, and the results are presented in Table 5 and Fig. 4.

TABLE 5. Comparison experiments of mainstream methods.

Method	Acc(%)	P(%)	R(%)	F1 (%)
MAML[32]	81.93	83.49	82.78	83.13
PN[27]	88.19	88.17	87.61	87.89
RN[28]	89.98	90.34	89.85	90.09
MSML [20]	90.74	90.81	90.53	90.67
DN4 [31]	92.21	92.78	92.12	92.45
PHR [33]	93.32	93.60	93.75	93.68
LLSTN [34]	95.27	95.69	95.35	95.51
Proposal-method	**97.96**	**98.61**	**98.07**	**98.34**

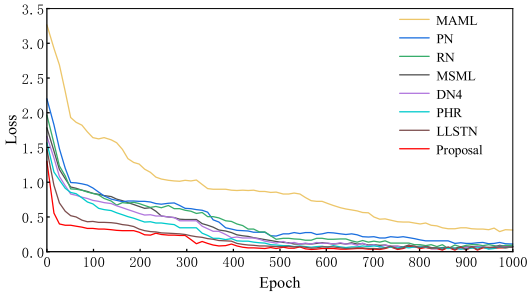

FIGURE 4. Loss value of each model training process.

Table 5 shows that the proposed model achieved better performance on the self-built rail surface state dataset under the 3-way 5-shot condition compared to other small-sample learning methods. The model has the best rail surface state recognition effect, with recognition accuracy, precision, recall rate, and F1 value indicators of 97.96%, 98.61%, 98.07%, and 98.34%, respectively. Compared to the LLSTN model, which exhibits higher performance, the model demonstrates an increase of 2.69%, 2.92%, 2.72%, and 2.83% across various indicators. Additionally, when compared to the PHR model, the model shows an increase of 4.64 %, 5.01%, 4.32%, and 4.66% across the same indicators. Similarly, compared to the DN4 network, the model presents an increase of 5.75%, 5.83%, 5.95%, and 5.89%. Furthermore, when compared to the MSML model, the model demonstrates a consistent increase of 7.22%, 7.80%, 7.54%, and 7.67% across the indicators. Moreover, in comparison to the RN network, the model exhibits an increase of 7.98%, 8.27%, 8.22%, and 8.25%. Additionally, compared to the PN network, the model shows an increase of 9.77%, 10.44%, 10.46%, and 10.45% across the indicators. Lastly, for the MAML model, the model demonstrates an increase of 16.03%, 15.12%, 15.29%, and 15.21% acrossvarious indicators. The results presented in Fig. 4 demonstrate that our model outperforms other small-sample learning methods in terms of training speed, initial loss value, fluctuation, and the final convergence value. Additionally, the model's loss values are consistently lower than those of other models across different iteration cycles. The above experimental results verify the effectiveness and superiority of the model proposed in this paper in the small-sample rail surface state identification task. This means that by introducing the pyramid splitting attention mechanism for feature extraction, using deep local splicing symbols for feature splicing, and designing convolutional network metrics, the representation ability of the model in rail surface state recognition can be significantly improved, thereby stably improving the model's performance aspects such as accuracy and recognition speed.

V. CONCLUSION

In this paper, we proposed a rail surface state recognition model based on the metric learning framework. The model addresses the issue of poor recognition accuracy in actual rail surface state recognition tasks with limited samples and also improves efficiency. To address the issue of inadequate feature extraction, we proposed the use of a pyramid-splitting attention mechanism in the feature extraction network. This mechanism can effectively extract detailed spatial information at multiple scales from rail surface state data samples and establish long-distance channel dependencies. As a result, the feature extraction network can learn more comprehensive multi-scale features, thereby enhancing the model recognition accuracy and improving the training speed. To address the issue of losing key features during the feature splicing process, we proposed the use of a deep local description concatenator. This concatenator splices the local descriptors of the query set feature map and the support set feature map, reducing the influence of irrelevant information such as background, while retaining local features that have obvious distinctions. This approach leads to improved accuracy in model recognition and overall performance. Finally, the proposed model was compared with the current mainstream small-sample learning methods, and the results confirm its effectiveness.

REFERENCES

[1] P. Pichlík and J. Bauer, "Adhesion characteristic slope estimation for wheel slip control purpose based on UKF," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4303–4311, May 2021.

[2] X. Ni, Z. Ma, J. Liu, B. Shi, and H. Liu, "Attention network for rail surface defect detection via consistency of intersection-over-union (IoU)-guided center-point estimation," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1694–1705, Mar. 2022.

[3] J. He, G. Liu, J. Liu, C. Zhang, and X. Cheng, "Identification of a nonlinear wheel/rail adhesion model for heavy-duty locomotives," *IEEE Access*, vol. 6, pp. 50424–50432, 2018.

[4] L. Zhuang, H. Qi, and Z. Zhang, "The automatic rail surface multi-flaw identification based on a deep learning powered framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12133–12143, Aug. 2022.

[5] J. Liu, L. Liu, J. He, C. Zhang, and K. Zhao, "Wheel/rail adhesion state identification of heavy-haul locomotive based on particle swarm optimization and kernel extreme learning machine," *J. Adv. Transp.*, vol. 2020, pp. 1–6, Jan. 2020.

[6] X. Jiang and Z. Ge, "Data augmentation classifier for imbalanced fault classification," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1206–1217, Jul. 2021.

[7] X. Jia, X. Zhong, M. Ye, W. Liu, and W. Huang, "Complementary data augmentation for cloth-changing person re-identification," *IEEE Trans. Image Process.*, vol. 31, pp. 4227–4239, 2022.

[8] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung, "On data augmentation for GAN training," *IEEE Trans. Image Process.*, vol. 30, pp. 1882–1897, 2021.

[9] J. Shin, Y. Kang, S. Jung, and J. Choi, "Active instance selection for few-shot classification," *IEEE Access*, vol. 10, pp. 133186–133195, 2022.

[10] N. Lai, M. Kan, C. Han, X. Song, and S. Shan, "Learning to learn adaptive classifier-predictor for few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3458–3470, Aug. 2021.

[11] X. Li, X. Yang, Z. Ma, and J.-H. Xue, "Deep metric learning for few-shot image classification: A review of recent developments," *Pattern Recognit.*, vol. 138, Jun. 2023, Art. no. 109381.

[12] J. Li, B. Chiu, S. Feng, and H. Wang, "Few-shot named entity recognition via meta-learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4245–4256, Sep. 2022.

[13] J. Chen, W. Hu, D. Cao, Z. Zhang, Z. Chen, and F. Blaabjerg, "A meta-learning method for electric machine bearing fault diagnosis under varying working conditions with limited data," *IEEE Trans. Ind. Informat.*, vol. 19, no. 3, pp. 2552–2564, Mar. 2023.

[14] M. Cheng, H. Wang, and Y. Long, "Meta-learning-based incremental few-shot object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2158–2169, Apr. 2022.

[15] Q. Sun, Y. Liu, Z. Chen, T.-S. Chua, and B. Schiele, "Meta-transfer learning through hard tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1443–1456, Mar. 2022.

[16] Y. Wang, J. Yan, X. Ye, Q. Jing, J. Wang, and Y. Geng, "Few-shot transfer learning with attention mechanism for high-voltage circuit breaker fault diagnosis," *IEEE Trans. Ind. Appl.*, vol. 58, no. 3, pp. 3353–3360, May 2022.

[17] Z. Li and A. Ralescu, "Generalized self-supervised contrastive learning with Bregman divergence for image recognition," *Pattern Recognit. Lett.*, vol. 171, pp. 155–161, Jul. 2023.

[18] J. Y. Lim, K. M. Lim, C. P. Lee, and Y. X. Tan, "SCL: Self-supervised contrastive learning for few-shot image classification," *Neural Netw.*, vol. 165, pp. 19–30, Aug. 2023.

[19] F. Gao, L. Cai, Z. Yang, S. Song, and C. Wu, "Multi-distance metric network for few-shot learning," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 9, pp. 2495–2506, Sep. 2022.

[20] W. Jiang, K. Huang, J. Geng, and X. Deng, "Multi-scale metric learning for few-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1091–1102, Mar. 2021.

[21] X. Li, X. Yang, Z. Ma, and J.-H. Xue, "Deep metric learning for few-shot image classification: A review of recent developments," *Pattern Recognit.*, vol. 138, Jun. 2023, Art. no. 109381.

[22] N. Nadagouda, A. Xu, and M. A. Davenport, "Active metric learning and classification using similarity queries," 2022, *arXiv:2202.01953*.

[23] P. Li and A. Tuzhilin, "Dual metric learning for effective and efficient cross-domain recommendations," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 321–334, Jan. 2023.

[24] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, 2015, pp. 1–30.

[25] F. Hao, F. He, J. Cheng, L. Wang, J. Cao, and D. Tao, "Collect and select: Semantic alignment metric learning for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8459–8468.

[26] Z. Li and A. Ralescu, "Learning generalized hybrid proximity representation for image recognition," in *Proc. IEEE 34th Int. Conf. Tools Artif. Intell. (ICTAI)*, Oct. 2022, pp. 901–908.

[27] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[29] W. Li, J. Xu, J. Huo, L. Wang, Y. Gao, and J. Luo, "Distribution consistency based covariance metric networks for few-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 8642–8649.

[30] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, "EPSANet: An efficient pyramid squeeze attention block on convolutional neural network," in *Proc. Asi. Conf. Conput. Vis. (ACCV)*, 2022, pp. 1161–1177.

[31] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7253–7260.

[32] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, D. Precup and Y. W. Teh, Eds. Aug. 2017, pp. 1126–1135.

[33] Y. Zhou, Y. Guo, S. Hao, and R. Hong, "Hierarchical prototype refinement with progressive inter-categorical discrimination maximization for few-shot learning," *IEEE Trans. Image Process.*, vol. 31, pp. 3414–3429, 2022.

[34] J. Wu and J. Hu, "Learning a latent space with triplet network for few-shot image classification," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 5038–5044.

[35] A. K. Dubey and Z. A. Jaffery, "Maximally stable extremal region marking-based railway track surface defect sensing," *IEEE Sensors J.*, vol. 16, no. 24, pp. 9047–9052, Dec. 2016.

[36] X. Ni, H. Liu, Z. Ma, C. Wang, and J. Liu, "Detection for rail surface defects via partitioned edge feature," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5806–5822, Jun. 2022.

[37] Z. He, S. Ge, Y. He, J. Liu, and X. An, "An improved feature pyramid network and metric learning approach for rail surface defect detection," *Appl. Sci.*, vol. 13, no. 10, p. 6047, May 2023.

[38] H. Yu, Q. Li, Y. Tan, J. Gan, J. Wang, Y.-A. Geng, and L. Jia, "A coarse-to-fine model for rail surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 3, pp. 656–666, Mar. 2019.

[39] C. Zhang, S. Liu, and J. He, "The study of surface state identification based on BP_AdaBoost algorithm," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 5865–5870.

[40] H. Huang, Z. Wu, W. Li, J. Huo, and Y. Gao, "Local descriptor-based multi-prototype network for few-shot learning," *Pattern Recognit.*, vol. 116, Aug. 2021, Art. no. 107935.

[41] Z. Wu, Y. Li, L. Guo, and K. Jia, "PARN: Position-aware relation networks for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6658–6666.

[42] V. N. Nguyen, S. Lokse, K. Wickstrom, M. Kampffmeyer, D. Roverso, and R. Jenssen, "SEN: A novel feature normalization dissimilarity measure for prototypical few-shot learning networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 118–134.

[43] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable Earth Mover's distance and structured classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12200–12210.

[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[45] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and L. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 6000–6010.

[47] F. Zhu, C. Fang, and K.-K. Ma, "PNEN: Pyramid non-local enhanced networks," *IEEE Trans. Image Process.*, vol. 29, pp. 8831–8841, 2020.

[48] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.

[49] Z. Li, Z. Liu, A. Zou, and A. L. Ralescu, "Learning empirical Bregman divergence for uncertain distance representation," 2023, *arXiv:2304.07689*.

[50] L. Wang, X. Bai, C. Gong, and F. Zhou, "Hybrid inference network for few-shot SAR automatic target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9257–9269, Nov. 2021.

[51] D. Das and C. S. G. Lee, "A two-stage approach to few-shot learning for image recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 3336–3350, 2020.

[52] K. Nguyen and S. Todorovic, "Feature weighting and boosting for few-shot segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 622–631.
[53] Z. Yu, K. Wang, S. Xie, Y. Zhong, and Z. Lv, "Prototypical network based on Manhattan distance," *Comput. Model. Eng. Sci.*, vol. 131, no. 2, pp. 655–675, 2022.
[54] U. K. Dutta, M. Harandi, and C. C. Sekhar, "Unsupervised deep metric learning via orthogonality based probabilistic loss," *IEEE Trans. Artif. Intell.*, vol. 1, no. 1, pp. 74–84, Aug. 2020.

HUIJUN YU received the Ph.D. degree in engineering from Central South University, Changsha, China, in 2018. He is currently the Vice President of the Hunan University of Technology, Zhuzhou, China. His current research interests include intelligent control of rail transit, image processing, and fault diagnosis.

CIBING PENG received the B.S. degree from the Hunan University of Arts and Science, Changde, China, in 2020. He is currently pursuing the M.S. degree with the School of Rail Transportation, Hunan University of Technology, Zhuzhou, China. His main research interests include few-shot learning, intelligent of rail transit, and image processing.

JIANHUA LIU received the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2013. He is currently the Dean of the School of Rail Transportation, Hunan University of Technology, Zhuzhou, China. His current research interests include intelligent operation and maintenance of rail transportation, few-shot learning, deep learning, and image recognition.

JINSHENG ZHANG received the B.S. degree from the Dalian Jiaotong University, Dalian, China, in 2019. He is currently pursuing the M.S. degree with the College of Railway Transportation, Hunan University of Technology, Zhuzhou, China. His main research interests include deep learning and image processing.

LILI LIU received the M.S. degree in traffic information engineering and control from Central South University, Changsha, China, in 2009. She is currently an Assistant Professor with the Hunan Railway Professional Technology College. Her main research interests include image recognition, deep learning, and process control.