Few-Shot Steel Surface Defect Recognition via Self-Supervised Teacher-Student Model with Min-Max Instances Similarity

Tianlei Wang*, Zeliang Li*, Ying Xu*, Jiacong Chen, Angelo Genovese, *Member, IEEE,* Vincenzo Piuri, *Fellow, IEEE,* Fabio Scotti, *Senior Member, IEEE*

***Abstract*—Steel Surface Defect (SSD) recognition plays a critical role in guaranteeing the quality and economic benefits of industrial metal products. At present, deep learning-based recognition methods have achieved excellent performance within related fileds. However, these methods usually depend on the availability of extensive training data and label information. For SSD recognition, the SSD data is often insufficient and labels are scarce, making it hard to train a supervised deep model. In this article, we proposed a novel few-shot SSD recognition method based on self-supervised constrastive learning (Self-SCL), which can effectively learn data representation with unlabeled data in the pretraining stage and learn categories from a few labeled samples in the fine-tuning stage. In the method, we proposed a information-guided Teacher-Student (TS) pretrained model with Min-Max Instances Similarity (MMIS) regularization. In the pretraining stage, we utilized the multi-crop mechanism to build a Teacher Encoder (TE) with multi-view information, and leveraged the TE to guide the Student Encoder (SE) with information. Meanwhile, the SE also provided the TE with "questions" during the update process, which the SE also shared updated information with the TE for further refinement. Specially, we introduced MMIS to address the problem of ambiguity between intra-class and inter-class distances in SSD data. During the fine-tuning phase, we fine-tuned the pretrained model obtained by TSMMIS with a few labeled data, and tested the SSD recognition accuracy in four SSD datasets. As the experimental results indicate, our method significantly outperforms other state-of-the-art methods on the four SSD datasets.**

***Index Terms*—Deep learning, few-shot, self-supervised constrastive learning, steel surface defect recognition.**

I. Introduction

THE assessment of surface quality in steel products is crucial for various industries, including manufacturing, military equipment and daily life. Anomalies during production or operation can lead to defects like scratches and pollution on steel surface, impacting product quality, durability and economic viability. Thus, accurate recognition of surface defects is vital for the manufacturing processes. Notably, the recognition of Steel Surface Defect (SSD) is a repetitive, high-speed and demanding task that places excessive pressure on quality inspectors. Furthermore, the subjective consciousness and emotional state of inspectors can lead to inevitable misjudgments. Hence, it is highly significant to develop intelligent SSD recognition systems as an alternative to manual recognition.

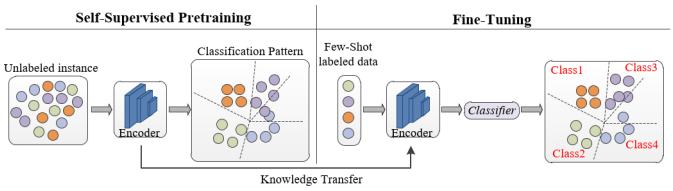

Fig. 1. Sketch of the Self-SCL framework. Firstly, the unlabeled instance was used to pretrain the encoder with self-supervised, so that the encoder had a "Classification Pattern". Secondly, the "Clssification Pattern" was fine-tuned with few-shot labeled data (only 1 to 4 labels per class), making it suit real labels, as well as the specific recognition of category was tested.

Recently, deep learning has been attracting much attention in industrial inspection and recognition research, particularly in the domain of SSD quality inspection [1], [2]. The availability of large annotated datasets and advancements in computing equipment have contributed to the progress in this field. Previous methods have focused on supervised learning with abundant labeled data [3], [4]. They all take advantage of the inductive bias of Convolutional Neural Networks (CNNs) [5], [6] for efficient representation learning. However, the likelihood of machine failure is low, and there are diverse types of defective products that require a substantial amount of expert annotations. Consequently, the SSD data volume tends to be constrained, accompanied by a scarcity of labeled samples. It presents a challenging scenario for few-shot SSD recognition, which relies on only a few labeled data in per class. In this case, over-fitting caused by the scarcity of supervision information [7] hinders the broad applications of deep learning methods in SSD recognition. Hence, developing a algorithm based on unlabeled data is crucial to SSD recognition task with few-shot labeled samples. Fortunately, scholars have tried to study this method and have made progress. For example, Semi-supervised learning (Semi-SL) [8] can perform consistent learning based on unlabeled and partially labeled samples. Self-supervised contrastive learning (Self-SCL) [9] can perform a pretraining based on unlabeled data to obtain reliable data representation ability that can be transferred to few-shot target tasks. Unfortunately, the transferability of

Tianlei Wang, Zeliang Li, Ying Xu, Jiacong Chen are with the Department of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China (e-mail: tianlei.wang@aliyun.com; zeliang0li@163.com; xuying117@163.com; jiacongchen7@163.com). Corresponding author: Ying Xu.

Angelo Genovese, Vincenzo Piuri and Fabio Scotti are with Depatimento di Information, Università , Degli Studi di Milano, via Celoria 18, 20133 Milano (MI), Italy (e-mail: angelo.genovese@unimi.it; vincenzo.piuri@unimi.it; fabio.scotti@unimi.it).

*Contributed to this work equally.

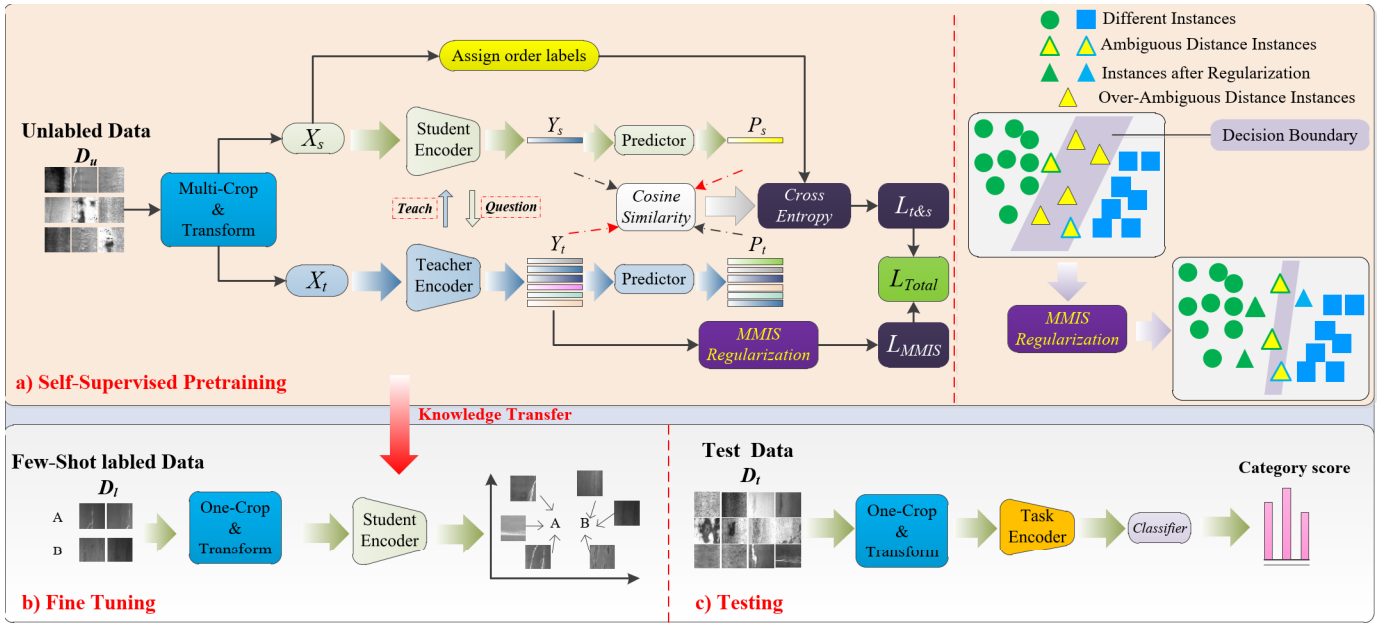

Fig. 2. Flowchart of our method. X_s and X_t represents one crop and K crops input image, respectively. Y_s and Y_t represents the output of encoder. P_s and P_t were obtained by Y_s and Y_t through several linear layers. The CosineSimilarity function was used to calculate the similarity values between Y_s and P_t, P_s and Y_t respectively. $L_{t\&s}$ is obtained by computing the CrossEntropy by assigning ordered labels to the obtained similarity values. L_{MMIS} enhanced the dissimilarity between instances by minimizing the maximum product of pairwise multiplication in Y_t. The fine-tuning and testing stage were carried out simultaneously, where the pretrained model was fine-tuned with few-shot labeled data and the recognition performance of the model was verified.

model knowledge is normally weakened in other based on labeled pretrained models due to their reliance on limited label information [10]. However, Self-SCL is more suitable for few-shot SSD recognition tasks compared to Semi-SL, as it requires a smaller amount of labeled data and is label-independent during training [11]. Moreover, Self-SCL facilitates the exploration of comprehensive data information from multiple perspectives, empowering the model to autonomously learn a data-specific "classification pattern" and transfer it to the target task [12], as shown in Fig.1. Yet it is still a challenging task to transfer Self-SCL to the field of SSD recognition from natural image related domains [13]. On the one hand, existing Self-SCL is based on representation learning with large volume and diverse datasets [14], benefiting from more instance-level information. However, the limited data and indistinguishability of SSD data hinder the ability of model to acquire ample instance-level information, making Self-SCL ineffective in establishing a reliable knowledge base during pretraining (In Fig.1. left). Although data generation-based methods [15], [16] have addressed the problem of scarce instance-level information in few-shot recognition tasks and achieved impressive results, they focus on data generation rather than leveraging existing data structures. Additionally, [16] relies heavily on manual intervention to ensure data quality during generation. In contrast, this paper aims to investigate a stable Self-SCL method solely based on limited unlabeled data. On the other hand, the ambiguity in the distances between intra-class and inter-class defects poses a challenge due to material characteristics [17]. However, Self-SCL relies on the similarity in intra-class and the difference in inter-class [12], which makes it difficult to deal with partial defect data with indistinguishable distances. Consequently, further research is needed to generalize Self-SCL methods for SSD recognition.

To solve the problems discussed above, inspired by the Teacher-Student structure in [18], we proposed a Self-SCL method **T**eacher-**S**tudent with **M**in-**M**ax **I**nstances **S**imilarity (TSMMIS) for few-shot SSD recognition without any data generation. The Teacher Encoder (TE), a large model described in [18], serves as a guide for the small model Student Encoder (SE) to achieve high-performance results. Nevertheless, our Teacher-Student (TS) module not only maintains the simple and consistent structure of TE and SE, but also incorporates TE to instruct SE with multi-view information, thereby enhancing SE data representation capability. We define this form as information-guided TS, rather than model-guided TS. For our approach, the framework is shown in Fig.2. Firstly, we built the TS module using the multi-crop mechanism. Multi-crop added the enhanced information of multiple views to the TE, while the SE only kept the main view information of a single view. Since different transformations can be seen as an independent category [19], multi-crop is able to provide diverse category information for TE, which "teaches" SE updates based on this information. In this way, the problem of limited instance-level information was solved. In addition, the SE fed back its own "question" to the TE by parameter sharing. Secondly, we proposed a regularization method, called MMIS, to optimize the TE. MMIS encourages greater differences between sample features from different views, thereby increasing the learning complexity of the model, reducing the model decision boundary, and alleviating the issue of intra-class and inter-class distance ambiguity in SSD data. The main contributions of the article are summarized as follows.

1) We proposed a few-shot SSD recognition method without any data generation via information-guided Teacher-

Student (TS) module and a regularization (MMIS), namely TSMMIS. According to our knowledge, our work is the first attempt in SDD recognition tasks.
2) We utilized multi-crop mechanism to augment the instance-level information in the limited unlabeled SSD data, and construct the TS module to learn this information in a Self-SCL manner to obtain robust data representation ability.
3) To better deal with the problem of distance ambiguity within intra-class and inter-class in SSD data, we proposed MMIS, which encourages enhancing the differences between instances and narrowing the decision boundaries of the model.
4) We tested our method on four SSD datasets for few-shot recognition and compared it against several state-of-the-art (SOTA) algorithms. TSMMIS achieved competitive performance. Especially, TSMMIS has more outstanding performance in the case of lowest few-shot.

The following sections of this article include: Section II presents related work. Section III describes the details of the method. We present the results of experiments in Section IV. Section V concludes the article.

II. Related Works

In this section, we will review the SSD recognition methods based on deep learning models in recent years, and briefly introduce Semi-SL and Self-SCL.

A. Recognition of steel surface defect

In early studies [20], [21], researchers used CNNs with fully supervision information for SSD recognition tasks. Soukup et al. [22] applied deep CNN to classify SSD and explored regularization in CNNs. Hao et al. [23] used deformable convolution to empower the feature extraction capability of CNN, and then improved the performance of SSD recognition. However, these methods often do not work well when faced with few labeled data. Furthermore, some scholars have tried to improve the feature utilization efficiency of CNN to resist the problem of a small number of labels. They adopted feature fusion [24], [25] to make the limited label information fully utilized by CNN, so as to prevent model overfitting. However, the method of feature fusion not only increases the complexity of the model, but also requires at least 20+ labeled data for each type of defect during training. Apparently, the method based on full supervision, either through the general network [5], [6] or the fusion network, is not suitable for the few-shot SSD recognition tasks.

Furthermore, in order to solve the few-shot problem of SSD, researchers have proposed to recognize SSD with small samples by generating data from the original data. Wen et al. [26] proposed CGAN to generate new data for classifying surface defects with small samples, and used the generated new data to train models. While generating data, CGAN uses the attention mechanism to alleviate the problem of intra-class and inter-class distance ambiguity of surface defect data. Yang et al. [27] used GAN to generate new metal defect samples and added these samples to the training and fine-tuning of the model. He et al. [28] used GAN to generate new SSD data and used them as pseudo-labels needed for semi-supervised learning. By combining the attention mechanism with GAN, Hao et al. [29] generated steel strip surface defect samples, and used the general network to classify all defect samples including the original samples. Although the GAN-based data augmentation method provides a feasible solution to solve the problem of few-shot SSD recognition, this method is cumbersome and unstable. In the work of [26]–[29], the data generation of GAN is outside the model training, preventing the model from being trained end-to-end. In addition, it is challenging to control the quality of data generated by GAN, and it is prone to collapse [30], which will increase the burden of model design. Even if [16] has achieved excellent few-shot SSD recognition performance by using GAN and Self-SCL, it still cannot get rid of the problems of manual intervention and unstable quality of generated data. Therefore, this paper focuses on few-shot SSD recognition method without adding any additional models and additional data, which is only based on limited data and scarce label information, and this few-shot SSD recognition method is also more promising in engineering application.

B. Semi-supervised learning

Semi-SL alleviates the need for labeled data by providing a way to leverage unlabeled data. In much previous Semi-SL work, the learning of a model usually encouraged the model to predict confidently at the output [31]. In much recent work, researchers use the idea of consistency regularization to construct Semi-SL, which maintains the consistency of different views of the same data. This method was first introduced by Berthelot et al. [31], who constructed a Semi-SL framework called MixMatch. In MixMatch, the pseudo-label was composed of the unlabeled data augmented by K different data, and the prediction results of the pseudo-label and the real label were compared to realize the representation learning of the model. Next, Berthelot explored the alignment criteria of data distribution and proposed ReMixMatch et al. [32]. In 2020, Sohn et al. [33] further simplified MixMatch and proposed FixMatch. FixMatch once again sets the standard for Semi-SL in the general domain. FixMatch only makes strong enhancement and weak enhancement on the data, and uses the weak enhanced data as the pseudo-label and calculates the cross-entropy of the strong enhanced data and the pseudo-label. In 2021, MPL proposed by Pham et al. [34] once again refreshed the semi-supervised optimal performance. Based on Semi-SL framework, it combines the idea of Teacher-Student. MPL uses the output of the consistent TE and SE as the loss function of the model. In 2022, Zheng et al. [35] proposed SimMatch, which considers both semantic level and instance level consistency, and instantiates a memory bank for storing labeled data, making full use of the limited labeled data. For the task of few-shot SSD recognition, Semi-SL is better than GAN-based methods. Semi-SL can not only realize end-to-end training, but also does not rely on additional data. However, Semi-SL is susceptible to the interference of pseudo-labeled

data during the training process, which makes it difficult for the model to balance the relationship between pseudo-labels and real labels when learning the data representation ability.

C. Self-supervised contrastive learning

Self-SCL has gradually become a research hotspot in various fields [36], [37]. Self-SCL can pretrain the model based on unlabeled data, and obtain great performance by fine-tuning with few-shot labeled samples. Compared with Semi-SL, Self-SCL has received much interest in the field of small sample size problem because it needs fewer labeled data. The first Self-SCL framework is MoCo proposed by He et al. [38]. MoCo stores the negative examples generated by the momentum encoder through a queue, considers the output of the basic encoder as positive examples, and calculates the loss between positive examples and negative examples through Info-NCE [39]. Meanwhile, Chen et al. [40] proposed SimCLR. Different from MoCo, SimCLR uses a huge update batch to ensure the number of negative examples, and proposes the NT-Xent loss function as the model update target. Caron et al. [41] proposed SWAV, which uses the consistency information between comparison instance clusters for self-supervised training of the model. In 2020, Gril et al. [42] proposed BYOL, which no longer uses negative examples to provide comparative information, but instead performs Self-SCL tasks by comparing the output of two encoders with shared parameters, one of which is stopped from propagating gradients. In 2021, Chen et al. [43] proposed SimSiam model, which similarly no longer uses the information of negative samples, but uses the output generated by two shared parameter encoders to construct a symmetric prediction loss. However, Self-SCL cannot be transferred effectively in the face of few-shot SSD scenarios since there is not enough instance level information available. Furthermore, the aforementioned Self-SCL methods primarily concentrate on capturing instance similarities, neglecting the significance of instance differences. Consequently, they fail to address the challenge of distinguishing between intra-class and inter-class distances in SSD data, leading to ambiguity. Even so, the idea of Self-SCL using unlabeled data for pretraining and rarely labeled data for fine-tuning is specific to the few-shot problem. Therefore, this paper will explore and use a novel Self-SCL method to well solve the problem of few-shot SSD recognition tasks.

III. METHOD

This study proposed a novel Self-SCL pretraining method for few-shot SDD recognition, named TSMMIS. In TSMMIS, ResNet18 [6] was used to construct the encoder of TS module, and the MMIS regularization module was designed to further improve model recognition performance. In the TS module, the TE obtained the information of multiple views through the multi-crop mechanism and used the multi-view information to guide the SE. The SE then feed back the "problem" to the TE. In the MMIS module, starting from the characteristics of human recognition of objects, we encouraged difference of different views of the same image, and then constrained the strength discrimination ability of the model. We strive to develop a self-supervised pretraining method to obtain sufficiently reliable "classification patterns" in unlabeled limited data. Therefore, we mainly discuss how TSMMIS is constructed.

TABLE I
ENCODER DETAILS OF TSMMIS

Stage	Type	Output
-	7x7, 64, stride 2	224x224
-	3x3 max pool, stride 2	112x112
R1	[3x3, 64; 3x3, 64] x 2	56x56
R2	[3x3, 128; 3x3,128] x 2	28x28
R3	[3x3, 256; 3x3, 256] x 2	14x14
R4	[3x3, 512; 3x3, 512] x 2	7x7
-	average pool, fc, softmax	1x1

A. Information-based Teacher-Student

We chose ResNet18 as our encoder, which is the most robust among the general network models. The ResNet of networks can maximally prevent the loss of features in deep learning through a special residual mechanism. In particular, our encoder $f(x;\theta)$ module mainly selects ResNet18 as the backbone network, and the details are set in Table I, which is introduced from [10].

Based on the selected encoder, Self-SCL utilizes various enhanced views of unlabeled images to conduct contrastive learning during the self-supervised pretraining stage, enabling the model to acquire robust data representation capabilities and obtaining a pretrained model capable of effectively differentiating instances [44]. However, the existing Self-SCL methods are developed based on large unlabeled datasets, and the diverse data provides sufficient instance-level information for the pretrained model of Self-SCL, which is difficult to achieve in SSD data. To capture sufficient instance-level information from the limited unlabeled data in SSD without imposing additional data burden during the pretraining process, we proposed an information-guided Teacher-Student framework. The construction process is outlined as follows:

Firstly, we used the "Multi-Crop & Transform" module to obtain one main view crop and multiple views crops by cropping the unlabeled images with different cropping rates and different transformative methods. Different crop ratios allow for effective similarities and inevitablee differences between different views of the same image. It is worth noting that image transformation using multi-crop mechanism does not add any data burden to the task, and it is based on the existing data. While the problem of instance diversity has been addressed, the utilization of this instance information for self-supervised pretraining remains crucial. To tackle this, we incorporated multi-view crops information into TE and single crop information into SE. We leveraged the multi-view information of TE to guide the update of SE, facilitating the transfer of knowledge from the knowledgeable Teacher to the Student. Additionally, parameter sharing enables TE to identify any "problems" in SE. It is important to note that SSD data exhibit inherent scale variability and spatial sparsity due to variations in defect sizes and their limited spatial presence

within the entire image. To address this challenge, we opted for cosine similarity as the feature measurement method. Cosine similarity is selected for its inherent scale invariance and robustness to spatial sparsity [45], making it well-suited for the characteristics of the SSD data. We utilize $CosineSimilarity$ (CS) and $CrossEntropy$ (CE) function to constitute the loss function of the TS stage, so as to realize the interactive update of TE and SE (see Fig.2). In particular, we specify that the input to SE is $X_s = \{x_s^0\} \in \mathbb{R}^{B\times 1}$ and the input to TE is $X_t = \{x_t^k, k \in [1, K]\} \in \mathbb{R}^{B\times 1}$ with *K* views, where *B* represents a batchsize. Inputting batchs of images from X_s and X_t into $f(x;\theta)$, we can obtain:

$$\left[\begin{array}{cc} Y_s & P_s \end{array}\right] = \left[\begin{array}{cc} y_s^0 & p_s^0 \end{array}\right] = f_s(x_s^0;\theta_s) \tag{1}$$

$$\left[\begin{array}{cc} Y_t & P_t \end{array}\right] = \left[\begin{array}{cc} y_t^1 & p_t^1 \\ y_t^2 & p_t^2 \\ \vdots & \vdots \\ y_t^k & p_t^k \end{array}\right] = f_t(x_t^k;\theta_t) \tag{2}$$

where $f_s({x_s}^0;\theta_s)$ and $f_t({x_t}^k;\theta_t)$ stand for TE and SE respectively, as well as θ_s and θ_t are the updatable parameters of the encoder, which actually share the weight with each other. $Y_s = \{y_s^0\} \in \mathbb{R}^{B\times C}$ and $Y_t = \{y_t^k, k \in [1, K]\} \in \mathbb{R}^{B\times C}$ represent the output of TE and SE, respectively, while $P_s = \{p_s^0\} \in \mathbb{R}^{B\times C}$ and $P_t = \{p_t^k, k \in [1, K]\} \in \mathbb{R}^{B\times C}$ represent the *Predictor* output. C is the number of target classes.

Secondly, this paper uses CS function to calculate the similarity information of Y_t to P_s and Y_s to P_t. Then, two similarity values, T_{t2s}^k and S_{s2t}^k, can be obtained. Remarkably, when performing CS, all P are transposing, so as to easily obtain the positive similarity and the negative similarity. $T_{t2s}^k \in \mathbb{R}^{B\times B}$ and $S_{s2t}^k \in \mathbb{R}^{B\times B}$ can be written as follows.

$$T_{t2s}^k = -\frac{p_s^0}{||p_s^0||} \cdot \frac{y_t^k}{||y_t^k||} \tag{3}$$

$$S_{s2t}^k = -\frac{p_t^k}{||p_t^k||} \cdot \frac{y_s}{||y_s||} \tag{4}$$

among them, k in T_{t2s}^k and S_{s2t}^k represents the similarity value matrix between the k^{th} view and the main view. The diagonal of the matrix is the positive similarity, and the other positions are the negative similarity.

Finally, based on the obtained T_{t2s}^k and S_{s2t}^k, we used the CE function to construct the loss function of TS. The purpose of using CE function is to make the proposed method have better instance discrimination ability by way of prediction. In the prediction process, the goal of CE is to maximize the positive similarity and minimize the negative similarity, so that the model can group the instances with the same characteristics into one cluster and obtain a "classification pattern". Notably, the labels used by the CE loss function are sequential numbers generated based on the batch size. For instance, if the batchsize is 64, the labels assigned to the samples would be [0, 1, 2, ..., 63]. The specific CE loss can be expressed by Equations (5) - (6).

$$L_{t2s} = -\sum_{k=1}^{K} \log\Big(\sum_{i=1}^{B} \frac{\exp(t_{ii}^k)}{\exp(t_{ii}^k) + \sum\limits_{j=1}^{B} \mathbb{1}_{[i\neq j]} \exp(t_{ij}^k)}\Big) \tag{5}$$

$$L_{s2t} = -\sum_{k=1}^{K} \log\Big(\sum_{i=1}^{B} \frac{\exp(s_{ii}^k)}{\exp(s_{ii}^k) + \sum\limits_{j=1}^{B} \mathbb{1}_{[i\neq j]} \exp(s_{ij}^k)}\Big) \tag{6}$$

where $\mathbb{1}_{[i\neq j]} \in \{0, 1\}$ is an indicator function evaluating to 1 if $i \neq j$. The t_{ij}^k and s_{ij}^k reprensent the element of T_{t2s}^k and S_{s2t}^k, respectively, the subscript i and j mean the corresponding positions in the matrix. In particular, t_{ii}^k and s_{ii}^k mean to get the diagonal values in the matrix.

Inspired by SimSiam [43], We obtained $L_{t\&s}$ by adding L_{t2s} with L_{s2t}, which is a symmetric form of loss function. It was worth noting that we canceled all gradients with respect to Y. By removing the gradient, L_{t2s} allowed the TE output to guide the SE output, since the value Y_t in the Teacher was just a constant when the model was updated. This process can be likened to a teacher imparting fixed knowledge, while the students need to learn and grasp this knowledge step by step. Similarly, L_{s2t} will provide the results produced by the SE to the TE so that the Teacher can understand the "problem" of the Student and improve the model further.

$$L_{t\&s} = \frac{1}{K}(L_{t2s} + L_{s2t}) \tag{7}$$

B. Min-Max Instances Similarity

After the TS stage, the instance discrimination ability of the model has been fully exploited. At the moment, although the model has the basic ability of instance discrimination, the problem of indistinct intra-class and inter-class distance of defect data is not well treated. In order to balance the instance discrimination ability obtained in TS stage, this paper proposed a regularization strategy of Min-Max Instance Similarity (see Fig.2 "MMIS" module), which can best utilize the difference information between different crops of the same image and limit excessive instance discrimination ability. It alleviates the impact of the distance ambiguity of defect data within and between classes on performance.

Firstly, the Y_t generated by TE were multiplied one by one in order (${y_t}^1$ was multiplied with ${y_t}^k$), and the matrix multiplication is simply batch matrix multiplication (bmm). Taking ${y_t}^1$ and ${y_t}^2$ as an examples, the details are shown in Equation (8) :

$$M^1 = y_t^1 \otimes y_t^2 = \left[m_1^1 m_2^1 \cdots m_n^1\right] \tag{8}$$

where ${m_n}^1$ denotes the n^{th} value in group 1, and such a multiplication form will have K groups. Secondly, the maximum position sequence number of M^1 was taken out according to the row vector direction and labeled.

$$\hat{y}^1 = \arg\max\left(\left[m_1^1 m_2^1 \cdots m_n^1\right]\right) \tag{9}$$

the function argmax$(*)$ is to extract the position sequence number of the maximum value, and $\hat{y}^1$ represents the label

of the first group. Based on the label $\hat{y}^1$, the loss function L^1_{MMIS} is obtained using CE.

$$L^1_{MMIS} = -\log\left(\frac{\exp(m^1_{\max})}{\exp(m^1_{\max}) + \sum_{j=1}^{n} \mathbb{1}_{[j\neq \max]} \exp(m^1_j)}\right) \tag{10}$$

where $m_{max}{}^1$ represents the maximum value in M^1. In particular, the whole L_{MMIS} can be obtained according to K crops with negative sign "-".

$$L_{MMIS} = -\frac{1}{K}\sum_{\mathrm{k}=1}^{K} L^{\mathrm{k}}_{MMIS} \tag{11}$$

Finally, in order to achieve the purpose of constrining the data representation ability of the model in the TS stage, this paper combined L_{MMIS} with $L_{t\&s}$ to form the final loss L_{total}. It is obvious from L_{total} that the "Min" feature of L_{MMIS} is provided by the combined "-".

$$L_{total} = L_{t\&s} + \lambda \times e \times L_{MMIS} \tag{12}$$

where λ is a tunable hyperparameter less than 1 and greater than 0, and e represents the number of epochs of the model update. In this paper, the influence of λ and e on the model performance will be discussed in the ablation experiment.

Especially, the L_{MMIS} is inspired by human intuition. According to human intuition, there is little similarity between different crops of the same image, Fig.3∼Fig.6 show the situation of different regions in the same image. Since different crops are obtained by random cropping, the size of crops and the image information contained in them are not the same, so in the human eyes, each crop may be a different category. Taking advantage of this, we encouraged each crop from TE to have more different information, so as to improve the model for further recognition of each instance. In this paper, λ, L_{MMIS}, and e were used to construct regularization terms with stronger constraint ability as the model was updated. The reason why a dynamic form of regularization is established is that the initial model urgently needs to establish the similarity information between instances, and the use of too strong regularization at this time will lead to the model unable to learn effectively. In the later stage of model update, the strength discrimination ability has been well established. Then, strong regularization is required to constrain the model, so that it can distinguish the intra-class and inter-class distance ambiguity data.

Furthermore, we try to further verify the feasibility of MMIS through some mathematical theories. It is not difficult to obtain from Equation (10) that when updating the maximum similarity $m^1_{\max}$, its derivative can be expressed as follows.

$$\frac{\partial L^1_{MMIS}}{\partial m^1_{\max}} = -1 + \frac{\exp(m_{\max})}{\sum_{j=1}^{n} \mathbb{1}_{[j\neq \max]} \exp(m^1_j)} \tag{13}$$

$$\nabla\theta_{m^1_{\max}} = p^1_{\max} - 1, p^1_{\max} = \frac{\exp(m^1_{\max})}{\sum_{j=1}^{n} \mathbb{1}_{[j\neq \max]} \exp(m^1_j)} \tag{14}$$

here, $p^1_{\max}$ is the probability value that $m^1_{\max}$ occupies in all m^1. According to the law of back propagation, $f(x;\theta)$ performance of the update depends on the gradient values imposed by each parameter. The gradient in the parameter $m^1_{\max}$ can be expressed by Equation (14). It can be seen that when $m^1_{\max}$ is larger, $p^1_{\max}$ will also be larger, thus improving the importance of $m^1_{\max}$ in the gradient. This makes the model tend to make $m^1_{\max}$ larger, so that L^1_{MMIS} converges faster and better. However, we added the constraint of "-" for MMIS in the L_{MMIS} of TSMMIS, which makes $m^{\mathrm{k}}_{\max}$ as small as possible, and then let the update of L_{MMIS} develop in the direction of suppressing $m^{\mathrm{k}}_{\max}$.

C. Summary of Method

In this section, we present the encoder used in our approach, the construction of the TS structure based on information guidance and the design of regularization of MMIS and its discussion. Meanwhile, **Algorithm1** provides the pseudo-code of TSMMIS for pretraining.

Algorithm 1 Pseud-code of TSMMIS

Require: TE $f_t(*;\theta_t)$, SE $f_s(*;\theta_s)$, $CS(*,*)$, $CE(*,*)$, Unlabeled training set D_u, K=5, epochs=300, Multi-Crop&Transform and Dataloader are imported from the Pytorch framework.

Ensure: L_{total}

for $e = 1, 2, \ldots,$epochs **do**
  transform_img = Multi-Crop&Transform(D_u,K)
  loader = Dataloader(*transform_img*)
  for $i = 1, 2, \ldots, len(loader)$ **do**
    y^0_s = $f_s(loader[i][0];\theta_s)$
    p^0_s = *Predictor*(y^0_s)
    for $k = 1, 2, \ldots, K$ **do**
      y^k_t = $f_t(loader[i][k];\theta_t)$
      p^k_t = *Predictor*(y^k_t)
      T^k_{t2s} = $CS(p^0_s, y^k_t.detach())$
      S^k_{s2t} = $CS(p^k_t, y^0_s.detach())$
      L_{t2s} += $CE(T^k_{t2s}, [0, 1, \ldots, 63])$
      L_{s2t} += $CE(S^k_{s2t}, [0, 1, \ldots, 63])$
      if $k < K-1$ **then**
        q_1 = y^k_t,q_2=y^{k+1}_t
      else
        q_1 = y^1_t,q_2=y^k_t
      end if
      M^k = bmm(q_1,q_2)
      $\hat{y}^k$ = argmax(M^k)
      L^k_{MMIS} += $CE(M^k, \hat{y}^k)$
    end for
    $L_{t\&s}$ = $\frac{1}{K}(L_{t2s} + L_{s2t})$
    L_{MMIS} = $-\frac{1}{K}L^k_{MMIS}$
    L_{total} = $L_{t\&s} + \lambda \times e \times L_{MMIS}$
  end for
  return L_{total}
end for

bmm: batch matrix multiplication; detach: no gradient.

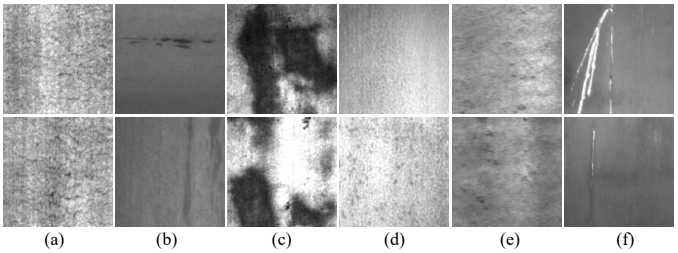

Fig. 3. NEU-CLS defect example demonstration. (a) Cr. (b) In. (c) Pa. (d) Ps. (e) Rs. (f) Sc.

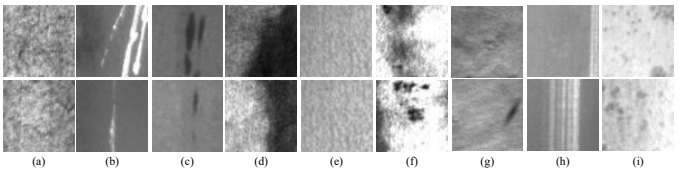

Fig. 4. NEU-CLS64 defect example demonstration. (a) Cr. (b) Gg. (c) In. (d) Pa. (e) Ps. (f) Rd. (g) Rs. (h) Sc. (i) Sp.

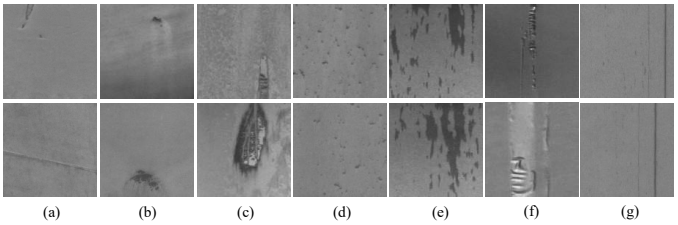

Fig. 5. X-SDD-CLS defect example demonstration. (a) finishing roll printing. (b) iron sheet ash. (c) oxide scale of plate system. (d) oxide scale of temperature system. (e) red iron. (f) slag inclusion. (g) surface scratch.

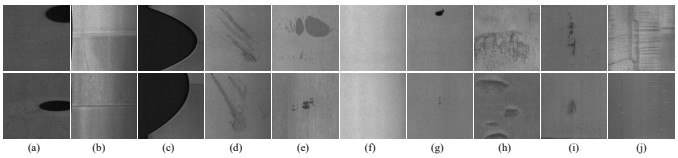

Fig. 6. GC10-CLS defect example demonstration. (a) Pu. (b) Wl. (c) Cg. (d) Ws. (e) Os. (f) Ss. (g) In. (h) Rp. (i) Cr. (j) Wf.

IV. Experiments

In this section, TSMMIS will conduct comparative experiments with several SOTA models to validate the effectiveness of our method in recognizing few-shot SDD. In addition, the contributions of each module of TSMMIS will be discussed in the ablation study. At the same time, this paper will also compare the influence of more factors changes on the recognition performance of TSMMIS, such as hyperparameters, iteration times, backbone network and other factors.

A. Task setup

1) Datasets: For the verification of TSMMIS performance in SSD recognition, we collected four SSD datasets, namely NEU-CLS [21], NEU-CLS64 [21], X-SDD-CLS [46], and GC10-CLS [47].

NEU-CLS dataset is extensively investigated in SSD recognition. It encompasses six distinct defect categories, namely crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled in scale (Rs), and scratches (Sc). Each category contains 300 images of size 200×200. See Fig.3 for details.

NEU-CLS64 dataset differs from NEU-CLS in terms of image size, with a smaller size of 64x64 pixels. Additionally, NEU-CLS64 encompasses a wider range of defect categories, including crazing (Cr), grooves and gouges (Gg), inclusion (In), patches (Pa), pitted surface (Ps), rolling dust (Rd), rolled-in scale (Rs), scratches (Sc), and spots (Sp). The number of these nine defective samples is 1589, 1210, 1148, 797, 775, 773, 438, 296 and 200, respectively. See Fig.4 for details.

X-SDD-CLS dataset specifically focuses on surface defects found in hot rolled steel strips. It encompasses various types of defects such as finish rolling printing, iron plate ash, plate system oxidation scale, temperature system oxidation scale, red iron, inclusions, and scratches. The number of these seven defective samples is 203, 122, 63, 203, 397, 238 and 134, respectively. See Fig.5 for details.

GC10-CLS dataset contains data sets of 10 kinds of steel plate surface defects, including punching (Pu), welding line (Wl), crescent gap (Cg), water spot (Ws), oil spot (Os), silk spot (Ss), inclusion (In), rolled pit (Rp), crease (Cr) and waist folding (Wf). The number of these 10 defective samples is 329, 513, 265, 354, 569, 884, 347, 85, 74 and 143, respectively. See Fig.6 for details.

We will delimit NEU-CLS, NEU-CLS64, X-SDD-CLS, GC10-CLS respectively:

Unlabeled training set $D_u = \{x_i\}_{i=1}^{n}$, will be used for Self-SCL training of the model to obtain transferable knowledge. 60 percent of NEU-CLS, NEU-CLS64, and GC10-CLS were randomly selected, and 70 percent of X-SDD-CLS were randomly selected as D_u for each dataset, and the rest were all used as test set. The n is the number of unlabeled training samples.

Few-shot dataset $D_l = \{(x_i, y_i)\}_{i=1}^{s}$, will be used to perform fine-tuning and linear evaluation. There are only one to four images per class, or 1-label $\sim$ 4-label. The D_l is randomly sampled from the D_u and given the true label. The s is the number of few-shot samples.

Test set $D_t = \{(x_i, y_i)\}_{i=1}^{z}$, will be used to test the final performance of the model. This set does not have any intersection with $D_u = \{x_i\}_{i=1}^{n}$ or $D_l = \{(x_i, y_i)\}_{i=1}^{s}$. The z is the number of test samples.

2) Evaluation Metric: The common methods to evaluate the performance of Self-SCL are linear evaluation and fine-tuning [40]. By fixing the weights of the backbone network and updating only the linear classification layer, the linear evaluation can estimate the correlation between the learned features of the model in the pretrained stage and the downstream tasks. Fine-tuning is done by updating the weights of the backbone network and the linear classification layer to reflect the potential of the pretrained model to perform downstream tasks. Both linear evaluation and fine-tuning are based on the pretrained model obtained in the self-supervised learning phase. For evaluating our model performance, we used a consistent metric: accuracy $\pm$ standard deviations. It can effectively show the overall recognition performance of the model.

3) Implementation details: We adopted a reliable data augmentation method from a previous task [16] that utilized Self-SCL for SSD recognition, and we kept the same data augmentation method in this paper.

In TSMMIS, the multi-crop mechanism was employed, generating crops through different crop ratios. Crop ratios is made up of random values ranging from 0.1 to 1. In TSMMIS, the hyperparameters were set as follows: K = 5, λ = 0.001, BatchSize = 64, and the ResNet18 backbone network was used. During the pretraining stage, we employed the SGD optimizer with a learning rate of 0.003 and momentum of 0.9 for 300 epochs. The model weights from the last epoch were then used for fine-tuning and linear evaluation. Because using only 1-label $\sim$ 4-label data per class resulted in limited authenticity and universality in the fine-tuning and linear evaluation, we conducted multiple experiments to ensure reliable results. Similar to [16], we conducted 100 random fine-tuning or linear evaluation experiments for each method in the Section IV-B and Section IV-C, ranging from 1-label $\sim$ 4-label. The results of these 100 experiments were then averaged. Considering the trade-off between computational resources and experimental effectiveness, 50 randomized fine-tuning experiments were performed for each group in the Section IV-D. By averaging the results of these 50 experiments, we obtained realistic and universal experimental outcomes. The experiments were all conducted using the PyTorch framework with an Nvidia GeForce RTX2080Ti 11G GPU.

B. Comparison with Other State-of-the-Art Methods

We have compared TSMMIS with Self-SCL models from recent SOTA levels, including MoCov2, SimCLR, SimSiam, BYOL, SwAV. At the same time, we also compared two SOTA unsupervised learning (Un-SL) models, Rotation [48] and InstFeat [49]. In particular, we compared the performance gap between TSMMIS and Gan-based FiCo (G-FiCo) as well as Fico [16] on NEU-CLS dataset. In the other datasets, we only compared TSMMIS with G-FiCo. We used these methods for pretraining, and as with TSMMIS, we conducted fine-tuning and linear evaluation experiments to show the performance. In addition, the data augmentation methods used remain consistent across all methods. Concretely, we first qualitatively compared the "classification pattern" of the pretrained models obtained by TSMMIS in different datasets, and then quantitatively analyzed the fine-tuning and linear evaluation performance of TSMMIS and other methods in four datasets.

1) Qualitative Analysis for TSMMIS: To gain a comprehensive understanding of the recognition behavior of the pretrained TSMMIS model and evaluate its clustering performance on different datasets, we employed TSNE technology [50] to visually analyze the instance discrimination ability learned during labelless pretraining. It can be seen from Fig.7 that the "classification pattern" obtained by labelless pretraining in NEU-CLS is the best, and only a small number of instances overlap. The worst is the "classification pattern" in GC10-CLS, where multiple classes overlap and the classes seem to be spatially close to each other. The underlying reason for this situation is the significant data imbalance across

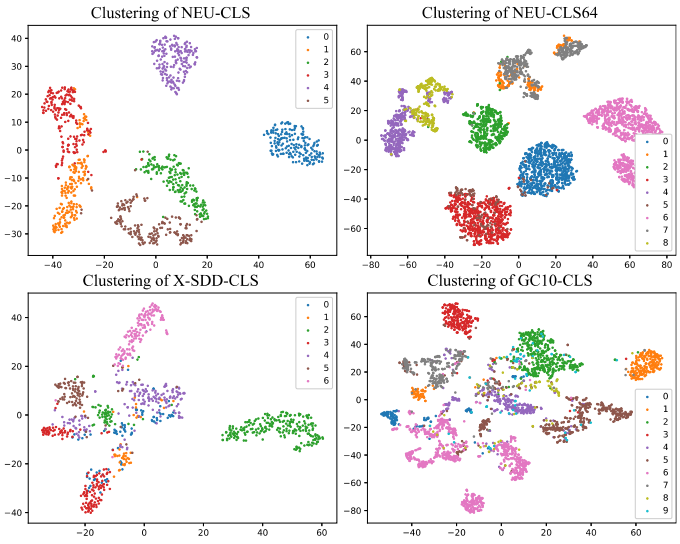

Fig. 7. Clustering performance of TSMMIS for self-supervised pretrained models in each dataset. Each color represents a different class, and since this is label-free pretraining, it is not clear what the specific label is.

various defect types in both the X-SDD-CLS dataset and the GC10-CLS dataset. Specifically, GC10-CLS shows a more drastic imbalance, with the minimum defect (Rp) and the maximum defect (Ss) differing by 810 instances. Fortunately, TSMMIS is able to cluster the unlabeled data relatively correctly even in such datasets, which shows its potential in dealing with the label imbalance problem. Although the "classification pattern" obtained by TSMMIS was proved to be effective in multiple datasets, further quantitative analysis was needed to prove the superiority of TSMMIS.

2) Quantitative Analysis for TSMMIS: To quantitatively access the performance of our method, we conducted fine-tuning and linear evaluation experiments comparing TSMMIS with other methods on four datasets. The process involved using D_u for label-free pretraining of each method, D_l for fine-tuning or linear evaluation of each pretrained model, and D_t for performance testing.

As shown in Table II to Table IX, TSMMIS achieves the highest level of recognition performance among SOTA methods in terms of both fine-tuning and linear evaluation. Especially in the 1-label case, TSMMIS shows a relatively striking performance regardless of the dataset. To be specific:

Recognition on NEU-CLS Dataset. The values in Table II show that, our method can achieve 93.77%, 94.27%, 95.20% and 95.51% classification accuracy in 1-label $\sim$ 4-label, respectively, and maintain the lowest standard deviation. This is significantly superior to other SOTA methods. The accuracy is 18.75%, 9.05%, 6.49%, 4.22% higher than the best SOTA method in four few-shot datasets respectively. In particular, compared with FiCo, our performance is 4.40%, 1.27%, 1.73%, 0.86% higher. However, Our method only surpasses G-FiCo by 2.34% in the 1-label case, while for the remaining cases, our method falls short compared to G-FiCo. Based on the analysis of the aforementioned results, TSMMIS exhibits the most outstanding data representation capability without the need for data generation. However,

in contrast to Gan-based methods, TSMMIS faces certain drawbacks as it does not augment the original data volume through data generation, which is a characteristic of G-FiCo. In Table VI, a comparable accuracy is also achieved in the linear evaluation performance of our method. These results show that TSMMIS is more able to obtain useful "classification patterns" without data generation from unlabeled samples and capture useful information more accurately, as well as shows the best performance in few-shot datasets.

Recognition on NEU-CLS64 Dataset. As can be seen in Table III and Table VII. Since NEU-CLS64 has smaller size images and more defect categories, the performance in this dataset is generally worse than that in NEU-CLS. Even so, our method can obtain the best performance among all compared algorithms, and achieve 84.51%, 87.65%, 89.27%, 89.43% defect recognition accuracy in 1-label $\sim$ 4-label, respectively. The accuracy of the proposed method is 7.39%, 6.71%, 5.88%, 4.71% higher than the best SOTA method in four few-shot datasets. Especially, TSMMIS demonstrates superior performance compared to G-FiCo, with performance improvements of 2.95%, 2.04%, 1.9%, and 1.13%. This indicates that TSMMIS exhibits enhanced robustness when dealing with small-sized datasets. Conversely, G-FiCo performance is hindered by the interference caused by the varying quality of generated data. Moreover, there is almost no gap between the fine-tuning performance and the linear evaluation performance. As these experimental results indicate, our method presents competitive performance in both normal-scale datasets and smaller-scale datasets.

Recognition on X-SDD-CLS Dataset. As shown in Table IV and Table VIII, the proposed method achieves the best performance in almost all the SOTA methods, especially in 1-label case. The X-SDD-CLS dataset has the characteristics of small defect regions and unbalanced defect classes. Therefore, it is more challenging to identify each defect sample in this dataset. In the fine-tuning process, our method obtains 72.68%, 78.93%, 83.22%, 84.63% in 1-label $\sim$ 4-label, respectively. Importantly, TSMMIS demonstrates superior performance in handling extremely limited labeled data compared to Rotation, achieving a remarkable 19.83% higher accuracy than Rotation in the 1-label scenario. Unfortunately, TSMMIS performs worse than G-FiCo except for the 1-label case. This is because data generation in G-FiCo potentially enhances the availability of richer feature information. Despite TSMMIS having a smaller amount of data for pretraining, its recognition performance is comparable to that of G-FiCo. Therefore, TSMMIS demonstrates greater robustness.

Recognition on GC10-CLS Dataset. As shown in Table V and Table IX, the proposed method achieves the best performance. GC10-CLS contains more data with imbalanced defect categories and intensity. In the case of fine-tuning, our method obtains 58.50%, 67.03%, 72.51%, 74.24% in 1-label $\sim$ 4-label, respectively, which is the best performance among all comparison algorithms. Compared with X-SDD-CLS, GC10-CLS has more obvious defect characteristics and more data, which makes TSMMIS perform better. In the linear evaluation case, the proposed method also achieves the best performance. It can be seen that the proposed method has better data scalability ability in the face of more defect categories. And it also has considerable performance in the data with imbalanced density.

From the above experiments, it is known that our proposed method outperforms the SOTA methods in four SSD datasets. Despite G-FiCo having a larger pretraining data volume, TSMMIS outperforms it in terms of recognition performance on both datasets (NEU-CLS64 and GC10-CLS). These benefit from two aspects. On the one hand, the TS module based on information guidance in the proposed method can effectively use the limited unlabeled data to pretrain the model without any data generation, and propose enough reliable data representation from the unlabeled data. On the other hand, thanks to the proposed MMIS module, it can encourage the difference between instances by restricting the similarity between instances, and then help TS module to establish a more excellent representation ability. For the SOTA methods, Self-SCL does not fully consider the problem of limited instance-level information, and in the whole pretraining phase, their learning style actively encourages the similarity information between instances to be close, while the difference information is treated negatively or even not considered. Such a practice is undoubtedly disastrous for the recognition of SSD. Un-SL methods lack implicit guidance information of defect categories in the pretraining stage, so they cannot establish a good data representation ability, resulting in that the model obtained by Un-SL pretraining cannot adapt to the target task. Furthermore, the extensive manual intervention required by G-FiCo to ensure the quality of data generation makes it impractical for industrial applications seeking efficient and straightforward solutions.

C. Comparison with Different Recognition Networks

This section, we will compare the TSMMIS with general Recognition networks [5], [6], [51]–[54]. We also conducted small sample experiments in four SSD datasets, as detailed in Fig.8. The results indicates that our method presents overwhelming performance. Conventional Recognition networks lack a specific supervision mechanism and primarily rely on fully supervised learning for data feature extraction. However, in the context of few-shot SSD recognition with limited supervision information, the fully supervised learning method is ineffective in addressing the few-shot problem. To overcome this limitation, TSMMIS leveraged the limited unlabeled samples to extract category information, established an effective data representation ability, and notably enhanced the recognition performance in the few-shot labeled sample scenario. Meanwhile, it also shows that the structure of the backbone network is not the key factor affecting the performance of the model in few-shot SDD recognition.

D. Ablation Results and Discussion

We have further discussed each module's impact of TSMMIS in the ablation experiments. Next, we will analyze in detail the relationship between the ablation results and the contribution points. It can be seen from the above experiments that the fine-tuning performance of the pretrained model

TABLE II
FINE-TUNING EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON NEU-CLS DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	74.67±8.28	83.05±5.01	86.48±3.82	88.52±3.48
	SimCLR [40]	75.02±7.42	85.22±4.60	88.71±4.16	91.29±3.29
	SimSiam [43]	49.00±9.00	62.76±7.38	70.38±5.60	73.91±5.66
	BYOL [42]	56.37±9.87	71.83±8.39	78.58±6.54	82.44±5.21
	SwAV [41]	48.61±7.73	59.72±7.36	66.69±6.85	71.02±5.82
	FiCo [16]	89.37±5.91	93.00±2.96	93.47±2.64	94.65±1.66
	G-FiCo [16]	91.43±6.04	95.35±2.90	96.04±1.88	96.48±1.42
Un-SL	Rotation [48]	57.59±6.67	68.40±5.95	73.83±4.97	77.78±4.59
	InstFeat [49]	56.40±7.30	70.67±6.58	78.00±5.30	82.64±4.21
Our	**TSMMIS**	**93.77±3.59**	**94.27±2.16**	**95.20±1.55**	**95.51±1.37**

TABLE III
FINE-TUNING EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON NEU-CLS64 DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	77.12±5.68	80.94±3.47	83.39±2.77	84.72±2.50
	SimCLR [40]	55.04±9.11	64.36±6.03	68.96±4.71	71.96±4.24
	SimSiam [43]	51.75±8.12	62.57±5.80	68.69±4.38	70.96±4.13
	BYOL [42]	59.44±7.73	74.58±5.84	80.68±4.27	84.34±2.74
	SwAV [41]	36.14±7.59	46.14±5.60	53.31±5.43	57.86±4.89
	FiCo [16]	-	-	-	-
	G-FiCo [16]	81.56±5.47	85.61±3.92	87.37±3.36	88.30±2.93
Un-SL	Rotation [48]	18.04±5.70	21.70±5.90	25.47±6.41	26.76±4.94
	InstFeat [49]	68.05±7.04	75.25±4.57	79.60±3.58	82.20±3.12
Our	**TSMMIS**	**84.51±3.33**	**87.65±2.55**	**89.27±2.27**	**89.43±5.01**

TABLE IV
FINE-TUNING EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON X-SDD-CLS DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	48.23±7.50	59.30±6.82	64.81±5.74	68.00±5.06
	SimCLR [40]	46.93±7.04	62.32±6.40	70.52±5.30	75.59±4.95
	SimSiam [43]	31.56±7.30	40.17±6.22	44.11±6.78	47.88±8.00
	BYOL [42]	43.00±7.04	57.28±6.70	66.64±6.66	71.76±5.87
	SwAV [41]	41.55±7.17	52.09±6.46	59.82±5.34	64.23±5.37
	FiCo [16]	-	-	-	-
	G-FiCo [16]	71.42±8.60	81.59±5.55	87.37±3.36	88.46±3.70
Un-SL	Rotation [48]	52.85±8.50	64.78±5.53	71.32±4.55	75.96±3.87
	InstFeat [49]	44.10±6.28	57.27±5.69	65.86±4.91	71.18±4.90
Our	**TSMMIS**	**72.68±9.87**	**78.93±6.48**	**83.22±2.89**	**84.62±1.97**

TABLE V
FINE-TUNING EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON GC10-CLS DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	50.73±5.52	58.72±4.75	62.81±4.60	66.17±3.99
	SimCLR [40]	43.67±6.44	53.53±5.63	59.01±3.95	61.85±3.70
	SimSiam [43]	21.91±2.32	24.41±2.45	25.26±2.76	25.58±2.72
	BYOL [42]	37.60±6.08	47.77±5.68	55.17±5.83	60.13±5.23
	SwAV [41]	36.06+6.70	44.13±6.07	47.88±5.42	51.50±3.85
	FiCo [16]	-	-	-	-
	G-FiCo [16]	58.47±5.35	65.81±4.79	70.66±4.29	73.40±3.78
Un-SL	Rotation [48]	32.08±4.83	42.55±4.06	48.20±4.57	52.58±4.20
	InstFeat [49]	38.39±5.98	47.81±5.26	54.82±5.02	58.78±4.58
Our	**TSMMIS**	**58.50±5.20**	**67.10±5.05**	**72.51±5.25**	**74.24±5.01**

TABLE VI
LINEAR EVALUATION EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON NEU-CLS DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	74.49±7.35	84.43±4.82	88.20±3.71	90.37±2.57
	SimCLR [40]	74.33±7.53	85.46±4.83	88.83±4.13	91.24±3.03
	SimSiam [43]	50.35±8.80	63.75±7.51	71.34±5.55	75.30±5.03
	BYOL [42]	63.52±7.68	78.31±6.49	84.44±4.38	87.35±3.73
	SwAV [41]	48.67±7.29	58.99±631	64.47±5.01	68.60±4.34
	FiCo [16]	89.37±5.91	93.00±2.96	93.47±2.64	94.65±1.66
	G-FiCo [16]	91.43±6.04	95.35±2.90	96.05±1.87	96.47±1.42
Un-SL	Rotation [48]	33.64±3.99	37.40±3.97	40.69±4.38	43.38±4.00
	InstFeat [49]	46.75±6.28	56.89±5.34	62.84±4.55	67.52±4.93
Our	**TSMMIS**	**93.77+5.68**	**94.56+2.39**	**95.33+2.69**	**95.78+1.71**

TABLE VII
LINEAR EVALUATION EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON NEU-CLS64 DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	76.10±6.55	81.14±3.35	83.83±2.52	85.25±2.54
	SimCLR [40]	55.39±9.05	64.68±6.12	69.32±4.54	72.38±3.88
	SimSiam [43]	60.11±7.01	68.31±5.36	72.66±4.31	74.91±3.67
	BYOL [42]	61.75±6.88	76.04±5.20	82.21±3.69	85.12±2.82
	SwAV [41]	35.16±7.24	43.90±5.24	50.94±5.36	54.88±4.84
	FiCo [16]	-	-	-	-
	G-FiCo [16]	81.55±5.56	85.62±3.92	87.37±3.42	88.28±2.96
Un-SL	Rotation [48]	12.59±2.32	12.79±2.46	13.24±2.48	13.61±1.97
	InstFeat [49]	66.56±7.18	75.81±4.51	80.84±3.16	83.63±2.73
Our	**TSMMIS**	**84.38±4.23**	**87.78±2.16**	**88.81±1.58**	**89.18+1.61**

obtained by TSMMIS is very similar to the linear evaluation performance. Therefore, all ablation stduies will be fine-tuning experiments based on NEU-CLS dataset.

1) Ablation of Multi-Crop Mechanism: The multi-crop mechanism can crop the input image in different ways, and combine K cropping degrees into K groups respectively. The processing of multi-crop data not only enriches the data structure information but also introduces additional category information [19]. We analyzed the effect of the number of crops on the model performance by varying the value of K. Notably, K contains a main view crop as input to SE. Table X shows that the performance of both 1-label and 4-label models significantly improves with the increase of the number of crops. Especially when K = 3, the obtained performance similarly exceeds other SOTA models. However, increasing the number of crops does not necessarily result in better performance. We observed that when transitioning from K = 5 to K = 6, the model performance deteriorated. This effect was particularly prominent in the 1-label case, where the model discrimination ability was influenced by other views, making the model difficult to judge. This problem arises from an excessive amount of crop information, causing the model to focus on marginalized details and blur the decision boundary. Therefore, in the subsequent ablation experiments, we all adopted the number of crops with K = 5, so that the optimal performance of the model and the appropriate amount of computation were considered simultaneously.

In addition to the number of crops, the selection of crop ratios also influences the model's performance. While crop ratios are randomly chosen from the range of 0.1 to 1, the specific range of selection is a crucial factor. We used *K* = 5 and specified crop ratio limits: upper limits [1.0, 1.0, 0.86, 0.715, 0.571] and lower for four groups—A: [1.0, 0.6, 0.514, 0.443, 0.413], B: [1.0, 0.5, 0.414, 0.343, 0.313], C: [1.0, 0.2, 0.172, 0.143, 0.114], D: [1.0, 0.1, 0.072, 0.043, 0.014]. Fig.9 illustrates the various impacts of different crop ratios and crop ranges on the model. Notably, the 1-label scenario is significantly affected, possibly because diverse

TABLE VIII
LINEAR EVALUATION EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON X-SDD-CLS DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	45.67±7.21	61.50±6.85	68.38±5.15	72.31±5.18
	SimCLR [40]	48.20±7.62	59.07±6.85	64.94±5.92	67.63±5.23
	SimSiam [43]	31.13±7.39	40.24±6.67	44.39±5.21	54.38±4.47
	BYOL [42]	43.57±7.05	60.36±6.43	68.55±5.27	73.80±5.27
	SwAV [41]	40.29±7.24	50.30±6.25	55.88±5.60	60.02±5.47
	FiCo [16]	-	-	-	-
	G-FiCo [16]	71.42±8.62	81.58±5.57	86.40±4.08	88.48±3.70
Un-SL	Rotation [48]	31.66±7.94	39.73±7.31	44.67±6.99	48.36±5.64
	InstFeat [49]	40.77±6.25	57.80±5.66	65.65±4.49	70.67±4.42
Our	**TSMMIS**	**73.89±7.77**	**79.17±6.89**	**84.00±2.27**	**84.37±2.32**

TABLE IX
LINEAR EVALUATION EXPERIMENT COMPARISON WITH SOTA ALGORITHM BASED ON GC10-CLS DATASET

Mode	Methods	1-label	2-label	3-label	4-label
Self-SCL	MoCov2 [38]	52.98±5.16	61.09±4.98	65.69±4.80	68.84±4.15
	SimCLR [40]	43.76±6.43	53.37±5.78	58.80±4.19	61.77±3.66
	SimSiam [43]	21.87±2.81	24.15±2.78	25.32±2.59	26.23±2.61
	BYOL [42]	45.31±5.93	57.83±5.34	65.43±4.53	69.71±4.32
	SwAV [41]	36.59±5.90	43.96±5.41	47.92±4.14	51.19±3.66
	FiCo [16]	-	-	-	-
	G-FiCo [16]	58.47±5.35	65.82±4.80	70.69±4.29	73.43±3.78
Un-SL	Rotation [48]	21.85±3.99	25.54±3.68	27.40±3.15	29.52±2.66
	InstFeat [49]	36.91±5.82	48.00±4.98	55.99±4.48	59.94±3.95
Our	**TSMMIS**	**58.49±5.79**	**67.03±5.93**	**73.47±4.17**	**74.65±4.46**

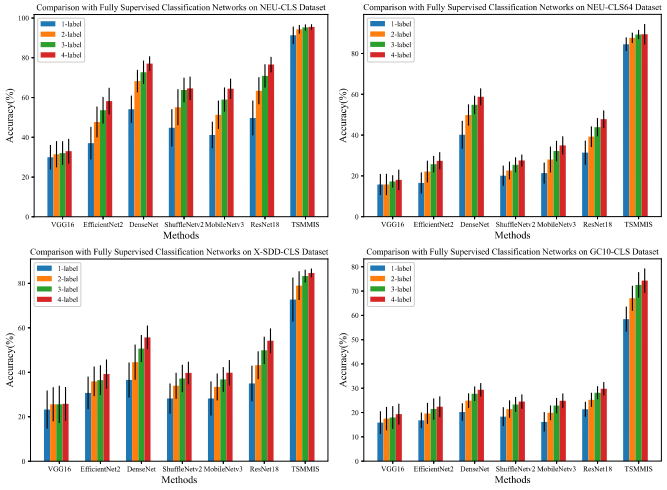

Fig. 8. Comparison with Different Recognition Networks on four Datasets. The vertical bar on the histogram represents the standard deviation.

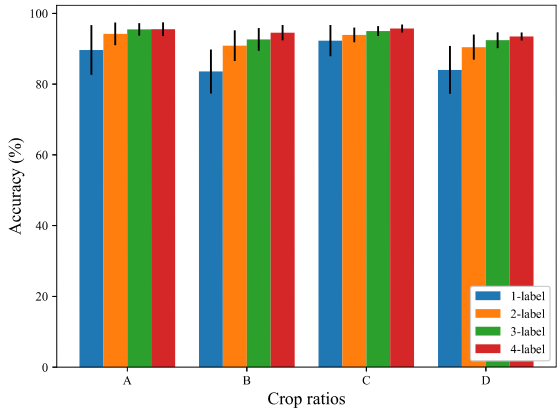

Fig. 9. Effects of different crop rates and different crop ranges on performance.

TABLE X
PERFORMANCE ACHIEVED BY DIFFERENT K

K	1-label	2-label	3-label	4-label
6	84.71±6.47	91.19±3.57	93.66±2.24	93.29±2.48
5	**92.26±4.41**	**93.87±2.08**	**94.96±1.39**	**95.69±1.13**
4	88.52±5.64	93.45±2.41	94.08±1.87	94.47±1.24
3	84.90±5.69	91.23±3.25	92.64±2.98	93.65±2.47

cropping generates finer distinctions, which is challenging to fine-tune using limited supervised data. Conversely, the 4-label performance appears less affected by cropping methods, likely due to the increased supervisory information. Therefore, selecting an appropriate crop method is pivotal, and group C emerges as the optimal choice. Simultaneously, the experiment highlighted an acceptable range for the cropping extent. An excessively large cropping extent would encompass redundant information, introducing model learning bias. Conversely, if the cropping extent is too small, the defect target might not be adequately captured, hampering the model's ability to acquire task-relevant information.

2) Ablation of Different Level Encoder: Various encoder depths offer distinct levels of data feature extraction capabilities to the model. The objective of conducting encoder ablation experiments is to assess the contribution of different depths in the network to the task performance. In this study, we selected different variations of ResNet, namely 1-1-1-1, ResNet18, ResNet34, ResNet50, and ResNet101, to examine their impact on the proposed method. The 1-1-1-1 configuration represents the number of basis blocks in ResNet set to [1, 1, 1, 1]. We investigated the performance and trends of the TSMMIS pretrained model in scenarios characterized by limited labeled samples, particularly focusing on the 1-label and 4-label cases. Our objective is to analyze how the network layers affect the performance of the TSMMIS model and observe how the model performance changes as network layers increase.

TABLE XI
PERFORMANCE AND TREND ACHIEVED BY DIFFERENT LEVELS OF ENCODER

Mode	Backbone	1-label	4-label
Residual series	1-1-1-1	88.90±4.86	94.79±1.16
	ResNet18	**92.26±4.41**	**95.69±1.13**
	ResNet34	89.58±5.04	95.89±1.98
	ResNet50	89.17±4.67	94.57±1.60
	ResNet101	87.26±5.60	93.46±2.03

It can be intuitively understood from Table XI that the recognition performance gradually improves as network layers increase. For the 1-label case, the performance improvement stops at ResNet18. The reason is that the network with more levels is prone to overfitting for fewer labeled samples, and there is mismatch between the number of parameters and the

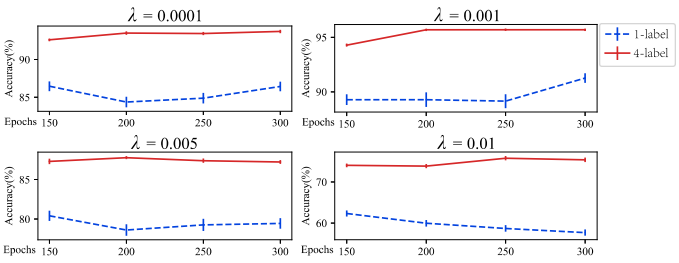

Fig. 10. Effect of λ and e (Epochs) on model recognition performance. Four different levels of λ are set: 0.01, 0.001, 0.005, and 0.0001. The vertical bars present in the broken lines indicate the range of standard deviation variation.

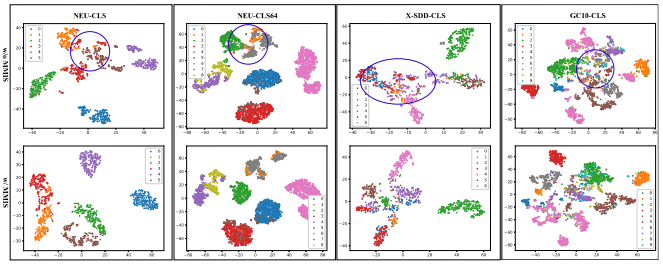

Fig. 11. Clustering performance of w/o and w/ MMIS in four datasets.

supervision information. However, at 4-label, the recognition performance growth is cut off at ResNet50, which is because the number of parameters just matches the supervision information, and the performance shows a decreasing trend when the level increases. ResNet18 can obtain the best performance in 1-label and dominate ResNet34 in 4-label, so this paper chose ResNet18 as the backbone network for the rest of the experiments.

3) Ablation of MMIS: Different hyperparameters such as λ and e (Epochs), affect the model's recognition performance simultaneously. Therefore, we plotted the 1-label and 4-label curves to observe the trend of performance in both cases.

Fig.10 shows that when λ = 0.001 (top right), the model can always maintain the best performance, the upper limit of the accuracy performance of recognition has exceeded 95%, and the fluctuation range of the standard deviation is also small. During epochs from 150 to 200, the 1-label performance tends to decrease for all levels except for λ = 0.001, while the performance for λ = 0.001 does not change much during this period, and the recognition performance is always close to 90%. During epochs from 200 to 300, the 1-label performance of TSMMIS gradually increases for λ = 0.001 or 0.0001. For the 4-label case, the four levels of λ can make TSMMIS maintain the trend of gradually enhancing performance with the increase epochs. However, with λ = 0.01 (bottom right), the performance of TSMMIS continues to degrade as the epochs increases in the 1-label case. The observed downward trend in the three curves corresponding to λ = 0.0001, λ = 0.005, and λ = 0.01 in the 1-label case can primarily be attributed to the effect of excessive regularization, which has the potential to cause model collapse. In particular, except for λ = 0.01, the model shows an increasing trend in the 1-label case, which also indicates the potential of the proposed method in dealing with fewer labeled data.

However, MMIS are essential. In order to understand more intuitively that suitable MMIS can provide reliable constraints for the model, which is more conducive to TSMMIS to obtain better "classification patterns" in self-supervised pretraining, we used TSNE technology [50] to visualize the clustering performance of the pretrained model of w/ MMIS and w/o MMIS on four SSD datasets. See Fig.11 for details.

From Fig.11, we have marked some of the deficiencies of w/o MMIS by selecting the blue oval boxes. In the NEU-CLS dataset, the w/o MMIS led to challenges in effectively distinguishing instances across multiple classes. Similarly, in the NEU-CLS64 dataset, the lack of MMIS resulted in a blurred representation of multi-class data. To address these issues, the w/ MMIS significantly improved the clustering of ambiguous instances. Notably, in the X-SDD-CLS dataset, the absence of MMIS exacerbated the problem of data representation with distance ambiguity, while its presence (w/ MMIS) effectively discerned instances. Unfortunately, for the GC10-CLS dataset, the performance of w/ MMIS and w/o MMIS was comparable, yet the clustering effect achieved by w/ MMIS remains more distinct. It should be noted that visualizing the clustering ability of a pretrained model via the TSNE [50] technique is merely a qualitative analysis. Mapping high-dimensional spatial data to low-dimensional spatial representation to visualize the performance of the model does not absolutely represent the true data representation ability. Therefore, quantitative analysis is also necessary. Furthermore, from the fine-tuning experiments, the MMIS regularization is effective and necessary for exploring the performance of w/o MMIS and w/ MMIS. See TableXII for details.

As can be seen in Table XII, MMIS consistently outperforms w/o MMIS by a substantial margin in all datasets, particularly evident in the 1-label cases, with performance improvements in other cases (2-4-label). Since w/o MMIS can not deal with the problem of fuzzy intra-class and inter-class distances well in the pre-training stage, this results in a model with a large decision boundary. In the case where only 1-label is available for fine-tuning, the scarcity of supervision data makes it challenging for the model to accurately classify ambiguous instances with real supervision. However, TSMMIS addresses this issue by mitigating "intra-class and inter-class distance ambiguity" through the introduced MMIS approach. By imposing similarity constraints to enforce dissimilarity between distinct views, we effectively narrow the decision boundary, enhancing the model performance. This improvement is particularly noticeable in cases involving fewer labeled samples, notably in the 1-label scenario. In particular, the performance improvement provided by MMIS is more pronounced when handling the X-SDD-CLS dataset. As depicted in Fig.5, the X-SDD-CLS dataset exhibits non-uniform defects of small area. MMIS counteracts these challenges by accentuating differences between instances, prompting the model to focus on image characteristics rather than background elements.

4) Ablation of Distance Function of MMIS: In the previous subsection, we discussed the effect of MMIS regularization on the model. Next, we will discuss the impact of different distance loss functions in MMIS on the pretraining perfor-

TABLE XII
PERFORMANCE OF W/O MMIS AND W/ MMIS IN FOUR DATASETS

Datasets	Mode	1-label	2-label	3-label	4-label
NEU-CLS	w/o MMIS	86.52±7.56	93.34±2.96	94.52±1.82	95.13±0.97
	w/ MMIS	**92.26±4.41**	**93.87±2.08**	**94.96±1.39**	**95.69±1.13**
NEU-CLS64	w/o MMIS	81.91±4.38	86.06±2.98	88.13±1.93	88.27±1.76
	w/ MMIS	**83.42±4.66**	**87.29±2.80**	**89.25±1.76**	**89.90±1.79**
X-SDD-CLS	w/o MMIS	66.78±7.72	74.59±4.53	78.35±2.72	79.08±3.09
	w/ MMIS	**73.73±7.64**	**79.17±6.89**	**84.00±2.27**	**84.37±2.32**
GC10-CLS	w/o MMIS	56.13±7.37	67.04±6.42	73.54±6.03	74.31±4.55
	w/ MMIS	**58.06±4.79**	**67.37±5.65**	**73.94±6.04**	**75.00±4.53**

TABLE XIII
PERFORMANCE OF DIFFERENT DISTANCE FUNCTION OF MMIS

Method	Distance Function	1-label	4-label
MMIS	L1-norm	89.09±5.05	95.39±1.84
	L2-norm	85.64±5.00	93.61±1.46
	SmoothL1-norm	90.06±5.06	96.86±1.49
	bmm	**92.26±4.41**	**95.69±1.13**

TABLE XIV
CLASSIFICATION SCORES OF DIFFERENT CLASSIFIERS USING THE PRETRAINED TSMMIS MODEL

Method	150 epochs	200 epochs	250 epochs	300 epochs
KNN	0.97	0.97	0.97	0.98
SVM	0.95	0.96	0.97	0.97
RF	0.95	0.95	0.96	0.96
LR	0.93	0.95	0.96	0.96

TABLE XV
MODEL PERFORMANCE WITH DIFFERENT BLUR LEVELS

Blur Level	1-label	4-label
0	**92.26±4.41**	**95.69±1.13**
0.1,2.0	90.08±4.75	95.56±1.27
1.0,2.0	88.67±6.52	95.22±1.39
1.5,2.0	89.46±4.82	94.68±1.70
2.0,3.0	88.20±4.85	92.93±1.72

mance of the model. Since the constructed MMIS aims at further improving the recognition performance of the model by regulating the relationship between instances, the distance relationship between instances is the focus of MMIS. We introduced L1-norm, L2-norm, and SmoothL1-norm, and the simple inter-vector batch matrix multiplication (bmm) was used in MMIS in this paper. It is worth noting that bmm actually computes the similarity between vectors, and the norm is a measure of the distance between vectors. We discussed the performance difference between 1-label and 4-label.

Table XIII shows that when L1-norm is used in MMIS, the recognition performance obtained is also competitive, and it also has performance improvement compared to TS without adding MMIS. However, when L2-norm is used, the model performance becomes worse. This is because L2-norm has a square operation, which will increase the proportion of L_{MMIS} in the whole L_{total}, leading to the block of the active update of the model. The model performance is also improved by using SmoothL1-norm, especially in the 4-label case, where the recognition performance is higher than the 1.17% accuracy with the use of bbm. This is due to the fact that Smoot-L1-norm has an optional distance measure, which makes its decision boundary partition more friendly when more labels are used. However, MMIS using bmm performs best when the 1-label is used. Both SmoothL1-norm and bmm aim to enhance the difference information between instances. According to Equation (12), SmoothL1-norm actually minimizes the maximum distance, and implicitly increases the proportion of minimum distance and sub-small distance in the backpropagation of MMIS, thus increasing the diversity information. On the contrary, bbm directly enhances the difference information by reducing the similarity between instances. Based on the above discussion, we have explored more bbm-based MMIS.

5) Ablation of Different Classifier for TSMMIS: Additionally, we incorporated four different classifiers, namely KNN, SVM, Random Forest (RF), and Logistic Regression (LR), to evaluate the reliability of the "classification patterns" learned by TSMMIS. The pretrained model generated by TSMMIS was utilized to assess the classification performance on $D_t = \{(x_i, y_i)\}_{i=1}^{z}$ using these four classifiers. No further parameter training was conducted at this stage. The performance of each classifier is quantified using a score, with higher scores indicating more reliable "classification patterns" learned by the pretrained model of TSMMIS. Based on the results presented in Table XIV, the pretrained model derived from TSMMIS demonstrates favorable "classification patterns". Notably, the KNN classifier achieves the highest score of 0.98, indicating its superior performance.

6) Ablation of Different Blur Lever: In the actual working condition, the vibration of the body often leads to fuzzy sampling of the steel image, which will bring great trouble to the defect recognition. For evaluating the robustness of our proposed method in the presence of sampling blur, experiments were conducted using images with varying degrees of blur. The fine-tuned model was employed for recognizing the blurred images, with the important note that no blurred images were included in the fine-tuning process. Only the blurred images were utilized for testing purposes. The degree of blur was classified into four levels: [0.1, 2.0], [0.5, 2.0], [1.5, 2.0], and [2.0, 3.0]. Each level had an equal probability of 50% occurrence.

The results presented in Tabel XV indicate that the addition of fuzzy noise relatively minimal affects the model performance. The highest recognition performance is achieved when the fuzzy level is randomly selected within the range of [0.1, 2.0], which can be attributed to the inclusion of smaller levels of blur within this range. As the blur level increases, the performance decreases for both the 1-label and 4-label cases. Although the 1-label performance for the fuzzy level [1.5, 2.0] reaches 89.46%, it is characterized by considerable randomness, and the overall trend still shows a decline in performance. The findings demonstrate that the TSMMIS pretrained model excels in handling both non-blurred and blurred images, exhibiting remarkable robustness in diverse blur conditions.

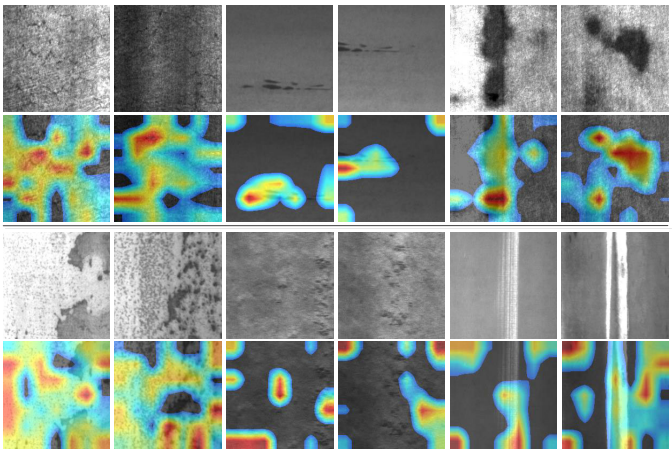

Fig. 12. Visualization of regions of interest in TSMMIS pretrained model. The first and third rows display the original images, while the second and fourth rows depict the corresponding regions of interest. The redder areas indicating higher interest.

7) Visualize Regions of Interest: The TSMMIS pretrained model capability to identify different defect regions in unlabeled data significantly influences the quality of the self-supervised pretraining. To evaluate this, we employed Grad-Cam [55] to visualize the model regions of interest after TSMMIS self-supervised pretraining. Notably, the model lacks label information and relies solely on the weights obtained from TSMMIS pretraining on unlabeled data. Fig.12 illustrates this visualization.

Fig.12 demonstrates the effective self-learning capability of TSMMIS. The pretrained model, obtained through TSMMIS, exhibits the ability to accurately focus on each defect area and provide reliable recognition of individual defect types, as observed in the correct recognition of the last two defects in the first row. However, notably, the pretrained model faces challenges in capturing defect regions in the presence of a blurred background and low contrast, as evident in the third and fourth defects in the third row. Therefore, the model after TSMMIS label-free pretraining has more accurate attention for most types of defects, and can recognize local defect regions even for some difficult samples.

V. Conclusion

In this paper, we proposed a Self-SCL method (TSMMIS) for few-shot SSD recognition. Our model consists of two modules. In the information-based TS module, which involves updating the single-view Student Encoder (SE) by incorporating information from the multi-view Teacher Encoder (TE), while the SE also shares updated information with the TE for further refinement. In the MMIS module, which promotes diversity among different views of the same image, thereby increasing the model decision complexity and narrowing the decision boundary for samples with ambiguous intra-class and inter-class distances. Notably, MMIS exhibits a more pronounced constraint effect in scenarios with fewer samples (1-label). Experimental results indicate that our TSMMIS method performs well and exhibits competitive performance in the SSD recognition task. Moreover, our method holds promise for industrial applications due to its ability to be implemented using simple and straightforward networks. However, the proposed method has limitations when dealing with unbalanced label data. On the other hand, the performance of the proposed method in other general industrial datasets needs to be verified. In the future work, we will fully consider the problem of label imbalance and the generalization of the method in the industrial field. Furthermore, the incorporation of generative methods and the exploration of diverse task patterns (detection and segmentation) should be considered.

Acknowledgments

Guangdong Basic and Applied Basic Research Foundation (No.2019A1515010716, No.2021A1515011576, No.2017KCXTD015); This work was supported by Key Research Projects for Universities of Guangdong Provincial Education Department (No.2020ZDZX3031, No.2022ZDZX1032); Jiangmen Basic and Applied Basic Research Key Project (2021030103230006670), Jiangmen Science and Technology Plan Project (2220002000246, 2022JC01021, 2022JC01027), Jiangmen major science and technology project (202101009002624), and Key Laboratory of Public Big Data in Guizhou Province (No.2019BDKFJJ015); Guangdong, Hong Kong, Macao and the Greater Bay Area International Science and Technology Innovation Cooperation Project (No.2021A050530080, No.2021A0505060011); Research on the Scheme of Intelligent Sensing Personnel Entering and Exiting Stations (HX22185), and Development of a Container Intelligent Panoramic Trademark Quality Inspection System Based on Machine Vision (HX22105).

References

[1] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, 2019.

[2] X. Zhou, H. Fang, Z. Liu *et al.*, "Dense attention-guided cascaded network for salient object detection of strip steel surface defects," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2021.

[3] X. Cheng and J. Yu, "Retinanet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2020.

[4] J. Yang, G. Fu, W. Zhu *et al.*, "A deep learning-based surface defect inspection system using multiscale and channel-compressed features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8032–8042, 2020.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

[7] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.

[8] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, pp. 1–20, 2022.

[9] K. He, H. Fan, Y. Wu, and et al., "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF CVPR*, 2020, pp. 9729–9738.

[10] Z. Liu, Y. Song, R. Tang, and et al., "Few-shot defect recognition of metal surfaces via attention-embedding and self-supervised learning," *J. Intell. Manuf.*, pp. 1–15, 2022.

[11] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 19 290–19 301, 2020.

[12] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, 2020.
[13] Q. Luo, X. Fang, J. Su, and et al., "Automated visual defect classification for flat steel surface: a survey," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9329–9349, 2020.
[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[15] Y. He, K. Song, H. Dong, and et al., "Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network," *Opt. Lasers Eng.*, vol. 122, pp. 294–302, 2019.
[16] Y. Xu, J. Chen, Y. Liang, and et al., "Flexible and diverse contrastive learning for steel surface defect recognition with few labeled samples," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–14, 2023.
[17] Q. Luo, Y. Sun, P. Li, and et al., "Generalized completed local binary patterns for time-efficient steel surface defect classification," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 3, pp. 667–679, 2018.
[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
[19] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.
[20] J. Wang, Y. Ma, L. Zhang, and et al., "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, 2018.
[21] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Appl. Surf. Sci.*, vol. 285, pp. 858–864, 2013.
[22] D. Soukup and R. Huber-Mörk, "Convolutional neural networks for steel surface defect detection from photometric stereo images," in *Int. Symp. Vis. Comput.* Springer, 2014, pp. 668–677.
[23] R. Hao, B. Lu, Y. Cheng *et al.*, "A steel surface defect inspection approach towards smart industrial monitoring," *J. Intell. Manuf.*, vol. 32, no. 7, pp. 1833–1843, 2021.
[24] Y. Xu, Z. Li, Z. Jiang *et al.*, "Defect recognition method based on fusion learning of multi-layer image features," in *2022 IEEE 4th Int. Conf. Power, Intell. Comput. Syst. (ICPICS)*. IEEE, 2022, pp. 342–345.
[25] Y. Gao, L. Gao, X. Li, and et al., "A multilevel information fusion-based deep learning method for vision-based defect recognition," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 3980–3991, 2019.
[26] L. Wen, Y. Wang, and X. Li, "A new cycle-consistent adversarial networks with attention mechanism for surface defect classification with small samples," *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8988–8998, 2022.
[27] B. Yang, Z. Liu, G. Duan, and et al., "Mask2defect: A prior knowledge-based data augmentation method for metal surface defect inspection," *IEEE Trans. Ind. Inform.*, vol. 18, no. 10, pp. 6743–6755, 2021.
[28] Y. He, K. Song, H. Dong, and et al., "Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network," *Opt. Lasers Eng.*, vol. 122, pp. 294–302, 2019.
[29] Z. Hao, Z. Li, F. Ren, and et al., "Strip steel surface defects classification based on generative adversarial network and attention mechanism," *Metals*, vol. 12, no. 2, p. 311, 2022.
[30] A. Creswell, T. White, V. Dumoulin, and et al., "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, 2018.
[31] D. Berthelot, N. Carlini, I. Goodfellow, and et al., "Mixmatch: A holistic approach to semi-supervised learning," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
[32] D. Berthelot, N. Carlini, E. D. Cubuk, and et al., "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *arXiv preprint arXiv:1911.09785*, 2019.
[33] K. Sohn, D. Berthelot, N. Carlini, and et al., "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 596–608, 2020.
[34] H. Pham, Z. Dai, Q. Xie, and et al., "Meta pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11 557–11 568.
[35] M. Zheng, S. You, L. Huang, and et al., "Simmatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14 471–14 481.
[36] J. Feng, N. Zhao, R. Shang, and et al., "Self-supervised divide-and-conquer generative adversarial network for classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.
[37] Z. Liu, Y. Song, R. Tang, and et al., "Few-label defect recognition of metal surfaces via attention-embedding and self-supervised learning," *J. Intell. Manuf.*, pp. 1–15, 2022.
[38] K. He, H. Fan, Y. Wu, and et al., "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[39] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Stat.* JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.
[40] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2020, pp. 1597–1607.
[41] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 9912–9924, 2020.
[42] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P.-H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21 271–21 284, 2020.
[43] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 750–15 758.
[44] A. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
[45] P. Xia, L. Zhang, and F. Li, "Learning similarity with cosine similarity ensemble," *Inf. Sci.*, vol. 307, pp. 39–52, 2015.
[46] X. Feng, X. Gao, and L. Luo, "X-sdd: a new benchmark for hot rolled steel strip surface defects detection," *Symmetry*, vol. 13, no. 4, p. 706, 2021.
[47] X. Lv, F. Duan, J. Jiang, and et al., "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, 2020.
[48] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.
[49] M. Ye, X. Zhang, P. C. Yuen *et al.*, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6210–6219.
[50] D. M. Chan, R. Rao, F. Huang *et al.*, "t-SNE-CUDA: Gpu-accelerated t-SNE and its applications to modern data," in *2018 30th International Symposium on Computer Architecture and High Performance Computing*. IEEE, 2018, pp. 330–338.
[51] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *ICML*, 2021, pp. 10 096–10 106.
[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
[53] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *ECCV*, 2018, pp. 116–131.
[54] A. Howard, M. Sandler, G. Chu, and et al., "Searching for mobilenetv3," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
[55] R. R. Selvaraju, M. Cogswell, A. Das, and et al., "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 618–626.

Tianlei Wang received academic training in Electrical Engineering (B.S.E) from Beijing University of Posts and Telecommunications, Signal Processing (M.S.) from the South China University of technology, and Safety engineering (Ph.D) from Beijing Jiaotong University. His research interests lie at the intelligent systems, control theory and pattern recognition.

Zeliang Li received the B.S. degree in communication engineering from Wuyi University, Jiangmen, China, in 2021, where he is currently pursuing the M.S. degree in electronic information engineering with the Department of Intelligence manufacturing.

His current research interests include deep learning and pattern recognition.

Ying Xu received the BS. and MS. degrees in automation, and control theory and control engineering from the Wuhan University of Science and Technology, Wuhan, China, in 2004 and 2008, respectively, and the Ph.D. degree in control theory and control engineering from the South China University of Science and Engineering, Guangzhou, China, in 2013.

She is currently an Associate Professor with the School of Department of Intelligence Manufacturing, Wuyi University, Jiangmen, Guangdong, China. Her current research interests include image processing, deep learning, and biometric extraction.

Jiacong Chen received the B.S. degree in communication engineering from Wuyi University, Jiangmen, China, in 2021, where he is currently receiving the M.S. degree in electronic information engineering with the Department of Intelligence manufacturing.

His current research interests include image processing and pattern recognition.

Angelo Genovese (Member, IEEE) received the Ph.D. degree in computer science from the Università degli Studi di Milano, Milan, ltaly, in 2014. He has been a Post-Doctoral Research Fellow incomputer science with the Università degli Studidi Milano, since 2014. He has been a Visiting Researcher with the University of Toronto, Toronto, ON, Canada. Original results have been publishedin over 30 articles in international journals, proceedings of international conferences, books, and book chapters. His current research interests include signal and image processing, three-dimensional reconstruction, computational intelligence technologies for biometric systems, industrial and environmental monitoring systems, and design methodologies and algorithms for self-adapting systems.

Dr.Genovese is an Associate Editor of the *Journal of Ambient Intelligence-and Humanized Computing* (Springer).

Vincenzo Piuri (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer engineering from Politecnico di Milan, Milan, Italy, in 1984 and 1988, respectively.

He was the Department Chair with the University of Milan, Milan, from 2007 to 2012, where he has been a Full Professor since 2000. He was an Associate Professor with Politecnico di Milan from 1992 to 2000, a Visiting Professor with The University of Texas at Austin, Austin, TX, USA from 1996 to 1999, and a Visiting Researcher with George Mason University, Fairfax, VA, USA, from 2012 to 2016. He founded a startup company, Sensure srl, Bergamo, Italy, in the area of intelligent systems for industrial applications (leading it from 2007 to 2010) and was active in industrial research projects with several companies. His main research and industrial application interests are intelligent systems, computational intelligence, pattern analysis and recognition, machine learning, signal and image processing, biometrics, intelligent measurement systems, industrial applications, distributed processing systems, Internet-of-Things, cloud computing, fault tolerance, application-specific digital processing architectures, and arithmetic architectures.

Dr. Piuri is an ACM Fellow.

Fabio Scotti (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the Politecnico di Milan, Milan, Italy, in 2003.

He has been an Associate Professor of computer science with the Universitá degli Studi di Milan, Crema, Italy, since 2015. His original results have been published in more than 100 articles in international journals, proceedings of international conferences, books, book chapters, and patents. His research interests include biometric systems, machine learning and computational intelligence, signal and image processing, theory and applications of neural networks, 3-D reconstruction, industrial applications, intelligent measurement systems, and high-level system design.

Dr. Scotti is an Associate Editor with the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS and *Soft Computing* (Springer). He has been an Associate Editor with the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and a Guest Coeditor for the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.