Improving Image Representation for Surface Defect Recognition with Small Data

Thai Tieu Phuong [1,2], Duong Duc Tin [1,2], and Le Hong Trang [1,2,*]

[1] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam

[2] Vietnam National University Ho Chi Minh City (VNU-HCM), Ho Chi Minh City, Vietnam

Email: ttphuong.sdh212@hcmut.edu.vn (T.T.P.); ddtin.sdh212@hcmut.edu.vn (D.D.T.); lhtrang@hcmut.edu.vn (L.H.T.)

*Corresponding author

***Abstract*—Automated surface defect detection systems have received much attention for quality control in industrial production. Deep learning techniques are proving their capability in these systems, due to the complexity of defects and inspection requirements. However, in fact, the availability of defective data is a major challenge. It is thus difficult to build an efficient model for a high-accuracy inspection system. In this paper, we present a method to deal with this lack of defective data by using self-contrastive learning to enhance image representations and the margin loss to improve the discriminativeness of defect features. Experiments were performed on the NEU dataset and MixedWWM38 dataset for several data size settings and for the few-shot learning task. The obtained results demonstrate the effectiveness of our proposed method. Particularly, the method achieves an accuracy of 98.83% and 92.27% on NEU dataset and MixedWM38 dataset, respectively, with only 20 training samples per class.**

***Keywords*—image classification, contrastive learning, representative learning, surface defect recognition**

I. INTRODUCTION

Surface defect detection plays a crucial role for quality control in many industries such as manufacturing [1, 2], electronics [3], and fabric productions [4, 5]. Traditionally, this task is carried out manually, which is time-consuming and requires a lot of human effort. Additionally, the human eye has limitations in detecting complex defects. To overcome this difficulty, many companies now use automated defect inspection systems. These systems typically consist of a camera module to acquire and highlight surface defects, if present, and software to detect and identify the defects.

Defect inspection based on visual perception can be mainly classified into two approaches, including traditional image processing-based and machine learning-based methods. The former approach applies image processing techniques to transform and characterize defect features for a given defect detection problem. This is only useful when defect classes are clearly differentiated. Furthermore, defect characteristics within the same class remain consistent. It is strongly dependent on the imaging environment and therefore has poor adaptability. Whereas, in the latter, learning models provide a more flexible approach for the complex defects. In particular, deep learning techniques have recently been proposed in this field. These methods can extract deep features of images via convolution, pooling operators and the attention mechanism. They are capable to generalize important features for class discrimination without requiring feature extraction rules provided by human, which can be designed as an end-to-end framework to integrate into automated inspection systems. However, training deep neural networks requires large amount of labeled training data to tune their parameters and avoid over-fitting. Unfortunately, in industrial scenarios, there are only a few or dozens of defective images that can be provided, posing the challenge of small data.

To solve this problem of deep learning approach, there are currently different solutions [6]: data augmentation, transfer learning, and unsupervised or semi-supervised methods. Data augmentation methods [7–9] consist of applying image processing operators on original images to obtain more samples and fusing individual defects to form defective samples. Transfer learning from pre-trained networks [10–12] is one of the most commonly used methods to boost performance and reduce over-fitting on small datasets. Finally, unsupervised and semi-supervised methods [13–16] can utilize a large number of unlabeled data to train the models.

In this paper, we present a method to improve image representations for surface defect recognition with small data. We choose representation learning approach to address the small sample problem, since its capability to learn better feature representations for class discrimination. Consequently, it performs the defect classification task more effectively with less training data. Our main contributions are listed as follows:

- We propose an end-to-end architecture that enhances feature embeddings of an extraction backbone for surface defect recognition task through data intensive and supervision of a self-contrastive loss and an angular margin loss. We

Manuscript received December 7, 2023; revised December 27, 2023; accepted January 26, 2024; published May 10, 2024.

doi: 10.12720/jait.15.5.572-579

integrate two modules into one pipeline for optimizing these loss functions effectively;
- We design multiple experiments with reduced training set on classification and few-shot learning task to have a comprehensive analysis on the efficiency of combining the loss functions for training;
- We evaluate the method on two surface defect benchmark datasets, NEU [17] and MixedWM38 [18], and achieve better accuracy than current methods in full-data training settings. The experimental results of small-data settings show that the model focus on discriminative features of defect regions and significantly outperform the backbone model for classification.

The rest of the paper is organized as follows: Section II reviews related works. The proposed method is given in Section III. Then, Section IV presents the experiments. Finally, Section V is the conclusion.

II. Literature Review

In this section, we comprehensively review current approaches of surface defect recognition in visual inspection and several representation learning methods in computer vision.

A. Surface Defect Recognition

With the excellent achievement of deep learning methods in computer vision, many pre-trained convolution-based networks on ImageNet [19–22] became the backbone or feature extraction blocks used for image classification task, including industrial defect recognition problems. In 2020, Konovalenko *et al.* [23] used the pre-trained ResNet50 [20] as a classifier for recognizing three classes of flat surface defects in rolled metal. They applied the binary focal loss function to overcome the problem of data sample imbalance and obtained the best accuracy of 96.91%. In 2021, Feng *et al.* [24] proposed a hot rolled steel strip defect dataset called Xsteel Surface Defect Dataset (X-SDD) with 1360 images of seven typical defect types. For defect recognition, they combined the RepVGG algorithm [25] with spatial attention mechanism to achieve promising results of 95.10% on this dataset. However, the performance of the algorithm was not very well on some categories, because the number of samples was not sufficient. In 2022, Li *et al.* [26] introduced a lightweight network based on Coordinate Attention and Self-Interaction (CASI-Net) mechanism to extract image features and locate defect regions for better recognition of steel surface defects in NEU dataset [17]. Despite reducing parameters and computation, it seems to be difficult for this architecture to distinguish some defects with a high degree of “inter class similarity and intra class diversity” [26].

In Defect Pattern Recognition (DPR) of wafer maps, Wang *et al.* [18] published the MixedWM38 dataset, additionally designed a Deformable Convolutional Network (DC-Net) and a multi-label output layer for mixed-type defect classification with average accuracy of 93.2%. By testing on the same data, Nag *et al.* [27] presented an encoder-decoder network called WaferSegClassNet (WSCN) for both classification and segmentation tasks. They achieved an average classification accuracy of 98.2% on all 38 classes. These supervised methods require the training on a substantially large sample size to alleviate over-fitting problem and reach a stable recognition performance. However, this is also the main challenge in the real industrial environment where the number of defect-labeled images is limited.

There are several works applying weakly supervised and few-shot learning method to overcome the key issue of small defective data. In 2019, Liu *et al.* [28] introduced a One-Class Classification (OCC) method based on Generative Adversarial Network (GAN) [29] for steel strip defect detection, which could only detect abnormal samples and cannot recognize defect types. He *et al.* [15] solved this data issue by a semi-supervised learning method based on multi-training of GAN and ResNet18 [20] networks. In particular, GAN was utilized to generate unlabeled samples, then the algorithm integrated both labeled and unlabeled into a multi-training process to acquire higher accuracy of 99.56%. In 2020, Deshpande *et al.* [30] approached the task through applying Siamese neural network to perform one-shot recognition on NEU dataset and achieved 83.22% true predictions without training on new defect categories.

B. Representation Learning

Advanced representation learning techniques introduced various useful loss functions to extract meaningful patterns of images for better recognition. Contrastive Learning (CL) emerged as an effective self-supervised learning method that could reduce the cost of annotating large-scale datasets, by learning embeddings from augmented versions of images. There are several works introduced based on this idea, such as Swapping Assignments between multiple Views of the same images (SwAV) [31], Momentum Contrast (MoCo) [32] and Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [33]. The SimCLR is a typical framework of contrastive learning for image self-representation. SimCLR combines data augmentation operators and provide an efficient contrastive loss to enhance image features for multiple downstream tasks.

In the field of face recognition, researchers recently concentrated loss functions to improve image discrimination. The key idea of these functions is to provide a guidance to the extraction model so that it can minimize the intra-class distance, while maximizing the inter-class distance. Schroff *et al.* [34] introduced the Triplet loss that enforces a margin between positive and negative face pairs via an anchor and Euclidean distance. This thus helps to reinforce the discriminability to other identities. Based on the idea of angular margin [35], ArcFace [36] leveraged the SoftMax loss via adding an angular margin penalty into the geometric interpretation of the function. It then could optimize face class separability and lead to outperforming the state-of-the-art of face recognition. These innovative approaches have played an important role in the field of image representation learning.

They become efficient methods to improve the model capacity of learning discriminative features.

In this work, we propose a learning strategy that combines multiple modified loss functions to improve discriminative feature representation of extraction model, which could lead to solving the challenge of small defect data in the real industrial context. In the next section, we elaborate on the details of our proposed methodology using limited training size for surface defect recognition in visual inspection.

III. Materials and Methods

We propose an end-to-end framework for surface defect recognition with small data through improving image representations. We apply a self-supervised representation learning paradigm to solve the classification task, combining the Contrastive loss and Margin loss functions to enhance discriminative features for defect type separability.

Fig. 1 illustrates the overview of our network architecture with two main modules. The first branch, Self-Contrastive Learning (SCL) module is designed based on the Siamese neural network with the aim of maximizing agreement between two augmented versions of images by using the Contrastive learning technique. The second branch, Angular Margin Penalty (AMP) module utilizes the Margin loss mechanism, which is to force the model learning discriminative features of inter-classes. ResNet-50 [20] is used as the backbone for feature extraction.

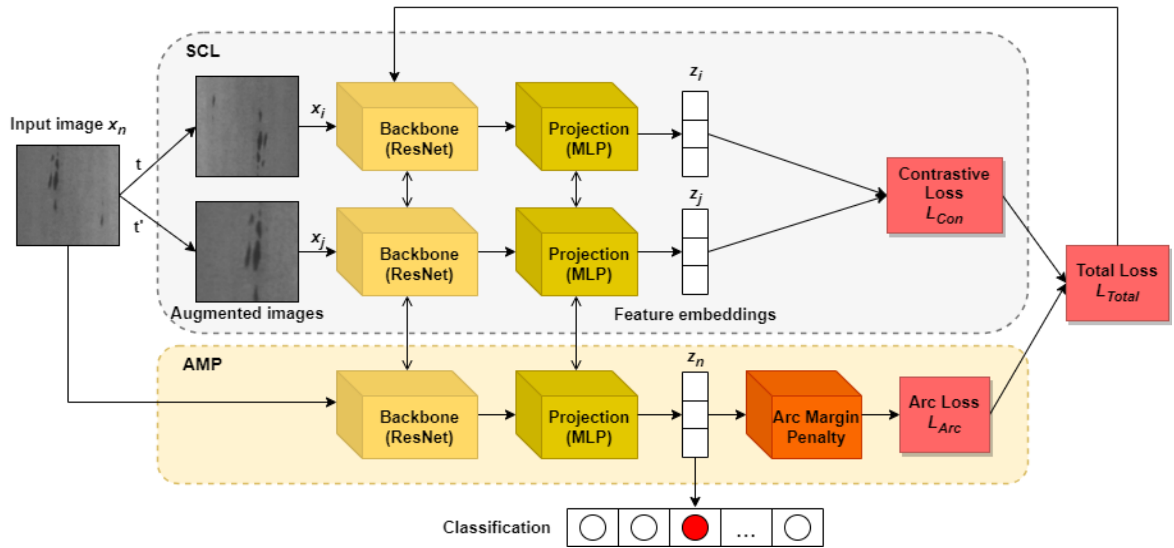

Fig. 1. Our proposed architecture. In SCL, each training image is augmented into two positive views for optimizing the contrastive prediction. In AMP, the feature embedding is interpreted in geometric space and is added a margin by the Arc Margin Penalty block to calculate the Arc loss. We train the Backbone and the Projection supervised by the total loss of both L_{Con} and L_{Arc} to improve features for classification task.

A. Self-Contrastive Learning (SCL) Module

Due to the real-life problem of small defect-labeled data, we aim to increase the amount of available data for training the classification model by using augmentation techniques. To this end, the twin Siamese network for Contrastive learning of visual representation that introduced in SimCLR [33] is used. The key idea is to learn good representations of different augmented versions of the same images. The module enhances the similarity degree of features between these two augmented views by optimizing the self-contrastive loss, with their pseudo labels "positive" or "negative". In particular, this module consists of three components:

- Data augmentation T: Each image x_n in a training mini-batch of N samples is transformed into two views x_i and x_j by different augmentation operators t and t'. These operators are randomly selected from a set T of augmentation methods that we use for manufacturing defective images;
- Base backbone E: In this work, we use the ResNet-50 as the backbone for feature extraction in all experiments. The outputs of the average pooling layer are high dimensional encoded vectors. They are then fed forward to a linear projection head Multi-Layer Perceptron (MLP) to reduce the dimension for calculating Contrastive loss. For every single image, it creates a positive pair x_i and x_j and then extract two feature embeddings z_i and z_j, respectively;
- Contrastive loss L_{Con}: A mini-batch with the size of N samples are stochastically transformed to $2N$ augmented views. The Contrastive loss function applies the cosine similarity metric on these feature vectors $sim(z_i, z_j) = z_i^\top z_j / (\|z_i\| \|z_j\|)$ to calculate the SoftMax loss of all image pairs by Eq. (1).

$$l(x_i, x_j) = -log \frac{e^{sim(z_i, z_j)/\tau}}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} e^{sim(z_i, z_k)/\tau}} \quad (1)$$

where $\tau \in [0,1]$ is a temperature parameter used to help the model learn from the negatives and $\mathbb{I}_{[k \neq i]} \in \{0,1\}$ represents a function that is set to 1 iff $k \neq i$. The contrastive loss of a training mini-

batch is defined to be the average of three losses between the original image x_n and its two views, $l(x_n, x_i)$, $l(x_n, x_j)$ and $l(x_i, x_j)$, respectively, as given in Eq. (2).

$$L_{Con} = \frac{1}{3N} \sum_{n=1}^{N} \left(l(x_n, x_i) + l(x_n, x_j) + l(x_i, x_j) \right) \quad (2)$$

B. Angular Margin Penalty (AMP) Module

The challenge of surface defect recognition is not only small sample issue, but also the specific characteristics of these kinds of images in the real industrial environment. The objects, which mean the defective regions, are often very small, compared to the whole image. Moreover, there is high variance of these defect images in intra classes, while inter classes also have some similar features, especially when multiple defect types appear together on the same surface. This leads to the misclassification of deep learning models.

In this work, we add a module to instruct the backbone model to learn feature representation effectively, by applying the Margin Loss strategy of ArcFace [36]. Normally, a mini-batch of N images in the process of training a n-class classifier updates the model by using the categorical softmax loss, as given in Eq. (3):

$$L = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \quad (3)$$

where $x_i \in \mathbb{R}^d$ denotes the feature of the i-th sample, belonging to the y_i-th class. d is the embedding feature dimension, $W \in \mathbb{R}^{d \times n}$ is the weights and $b_j \in \mathbb{R}^n$ is the bias. The logit $W_j^T x_i + b_j$ can be formulated in cosine geometric space. The bias b_j is set to 0, then the logit of the weight and the feature is transformed to dot product $W_j^T x_i = \|W_j\| \|x_i\| cos\theta_j$. Normalized by l_2, $\|W_j\| = 1$ and scale $\|x_i\| = s$, then the logit is only dependent on the angle θ_j between the normalized weight and the feature. This angle is added by an angular margin penalty m for improving discriminative features of intra-class and inter-class samples, as shown in Eq. (4). We apply L_{Arc} by implementing an Arc Margin Penalty block that gets dense embeddings from the MLP projection and re-calculates the logits for the softmax function, as illustrated in Fig. 1.

$$L_{Arc} = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{s\left(\cos\left(\theta_{y_i} + m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i} + m\right)\right)} + \sum_{j=1, j \neq y_i}^{n} e^{s \cdot cos\theta_j}} \quad (4)$$

C. Network Optimization

The training loss of our end-to-end network consists of two loss functions that are Contrastive loss L_{Con} and Arc loss L_{Arc}. The first term is supervised by self-defined pseudo labels of augmented images, while the second term modified from softmax loss is supervised by class labels for defect recognition. The total loss should be defined as

$$L_{Total} = \alpha \cdot L_{Con} + \beta \cdot L_{Arc} \quad (5)$$

where α and β are hyper-parameters used to balance the two losses, L_{Con} and L_{Arc}. Due to the difference in their loss values, we set $\alpha = 3\beta = 0.75$ during the training phase of all experiments. This joint loss function guides both the base backbone and the projection head to learn good representations. It aims to not only maximize the similarity between augmented versions of an image, but also increase the feature gap between different classes.

IV. Result and Discussion

In this section, we evaluate the proposed method on two benchmark datasets, NEU [17] and MixedWM38 [18], to assess how effective the model recognizes defect types in comparison with several current frameworks. The output feature representations are then used for supervised classification task and few-shot learning recognition task.

A. Implementation Details

In the training phase, to overcome the problem of limited defective images in the industry and facilitate the contrastive prediction task mentioned in Section III, we perform a data augmentation task. We augment surface defect data with simple random cropping (with resizing) to create adjacent, local and global views from the images. To make the contrastive prediction task become harder, we additionally use affine transformations, such as random rotation (with different degrees), horizontal and vertical flip operators. The angle of the rotation is selected stochastically from a set of angles {0, π/2, π, 3π/2} (in radian). This augmentation composition has been proved to improve the quality of image representation [33]. As mentioned, ResNet-50 is used as the backbone for feature representation. Table I presents the hyper-parameter settings.

TABLE I. Hyper-Parameters Used to Train Our Network

Parameter	Value
Batch Size	32
Number of epochs	100
Learning Rate	5×10^{-4}
Optimizer	Adam [37]
Temperature τ in L_{Con}	0.5
Margin m in L_{Arc}	0.5
Embedding Size d	128

In the testing phase, we build a reference database to inference the testing set. The reference images are chosen randomly from the training set, and then extracted to feature vectors for creating the database. For every testing image, we utilize the cosine similarity metric to compare its feature vector to k-shot images of each class. The prediction result is decided via the highest average value.

B. Experiments on NEU

The NEU dataset [17] is a defect dataset that captures the defect images from the surface of hot-rolled steel plates.

This dataset contains 6 typical defect types of steel strip, i.e., Rolled-in Scale (RS), Patches (Pa), Crazing (Cr), Pitted Surface (PS), Inclusion (In), and Scratches (Sc). There are 1,800 images (200×200 dimension), with an equal number of 300 for each class. Fig. 2 shows the patterns of all 6 defect types. It can be observed that PS, RS and Cr classes do not clearly have defective regions in the images, meaning that it is not easy for model to extract discriminative features.

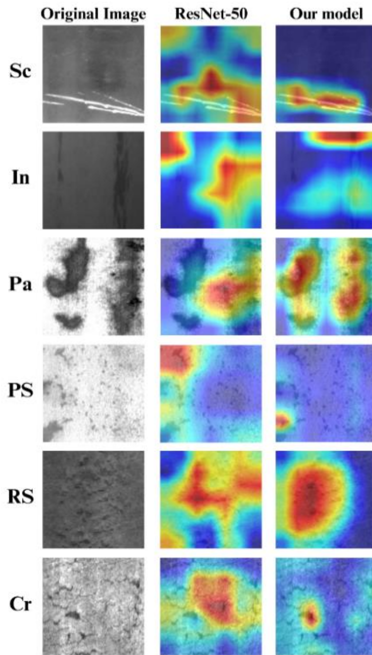

Fig. 2. Gradient-based visualization of feature maps generated by ResNet-50 and our method (both are trained with only 10 images per class) on the samples of 6 defect patterns in NEU.

1) Classification task

Firstly, we compare our accuracy scores with current deep learning models on steel surface defect classification task to evaluate our method. Fig. 2 illustrates the heat-maps of 6 defect images generated by the last layer of the backbone ResNet-50 and our model, based on Gradient-weighted Class Activation Mapping (Grad-CAM) [38]. Both models are trained on the same small-scale data with only 10 images per class. The maps highlight important regions in the image for predicting the class. We can see that our model focus on key features of defect regions for discrimination and reduce noise features from the background. This shows the efficiency of the Contrastive loss and the Arc loss, compared to the conventional Cross-Entropy loss.

For a fair comparison in the number of labeled samples per class used in the training phase, we randomly divide the NEU dataset into training and testing sets with proportions of 80%, 70% and 60% data for training, corresponding to the proportions used in Siamese Neural Network [30], CASI-Net [26] and Secure Sockets Layer, (SSL) [15]. The experimental results in Table II show that within the context of the same training samples, our representation learning strategy can outperform another contrastive learning approach [30], an attention mechanism [26] and a semi-supervised method [15], with the average accuracy of 100%, 99.81%, and 99.58%, respectively.

TABLE II. COMPARISON OF CLASSIFICATION ACCURACY ON NEU

Method	Training Data (%)	Accuracy (%)
Siamese Neural Network [30]	80	92.55
CASI-Net [26]	70	95.98
SSL [15]	60	99.56
Proposed (60%)	60	**99.58**
Proposed (70%)	70	**99.81**
Proposed (80%)	80	**100.00**

As mentioned above, due to the limited quantity of defective samples existing in the industry, we conduct several experiments with very small amounts of training data. Numbers of 10, 20, and 50 samples per each class are chosen randomly from the original NEU training set. We aim to show the efficiency of combining Contrastive loss and Arc loss to optimize the backbone for feature extraction, compared to only using the backbone alone. For each small-data setting, we keep the same training hyper-parameters as the ResNet-50 baseline, and then average the accuracy of 10 different experiments. As showed in Table III, our method has better results than ResNet-50 when all being trained by 20 and 50 samples. However, in case that defective images of a type are as scarce as 10 samples, the method strongly obtains 98.39% of true predictions on the test set. While the original Convolutional Neural Network (CNN) baseline requires a large number of labeled training data to avoid over-fitting, our method overcomes this challenge by utilizing contrastive self-supervised learning algorithm with composition of data augmentations, additionally reinforcing discriminative features with the margin penalty module.

TABLE III. RESULTS OF SMALL-DATA SETTINGS ON NEU

Train Dataset (images/class)	ResNet-50 (Backbone)	Proposed
10	93.89	**98.39**
20	97.72	**98.83**
50	99.06	**99.72**

2) Few-shot learning task

In order to show further the effectiveness of the proposed method, we conduct few-shot learning experiments. In this experiment, it allows the model to learn from all existing samples and then predict unseen defect types with a few representative images called k-shot of each new class. We train our method on full 900 images of 3 classes, including patches, inclusion and rolled-in scales. In the testing phase, 900 images of the remaining classes, consisting of crazing, scratches and pitted-surface, are shown to the model for few-shot recognition.

In Table IV, with only k samples of each new category, for $k = 1$, 3, and 5, we achieve the accuracy of 82.16%, 84.53%, and 86.29%, respectively. This result is competitive with the testing accuracy 83.22% of the one-shot recognition baseline introduced in [30] for steel surface defects.

TABLE IV. RESULTS OF FEW-SHOT LEARNING TASK ON NEU

k-shot	Accuracy (%)
$k = 1$	82.16
$k = 3$	84.53
$k = 5$	**86.29**

C. Experiments on MixedWM38

In order to show further the performance of our proposed network, we conduct this experiment to test the classification task on a much more complicated dataset, i.e., MixedWM38 [18]. The dataset has totally 38 defect patterns of semiconductor wafer surfaces, including 38,015 images (52×52 dimension) with single and multiple defect types appearing on a map. The 8 single defect types are listed as follows: Center (C), Donut (D), Edge-Loc (EL), Edge-Ring (ER), Local type (L), Nearful (NF), Scratch (S) and Random (R). The remaining 29 mix-type patterns have several single defect types presenting together a piece of wafer, which makes the recognition task more complex. Particularly, there are 1 defect-free pattern (C1) and 4 defect groups: 8 different single defects (C2-C9), 13 two mixed-type defects (C10-C22), 12 three mixed-type defects (C23-C34), and 4 four mixed-type defects (C35-C38). Fig. 3 shows some examples of our results. Obtained heat-maps indicate that the proposed method precisely gives representative features for the classification task.

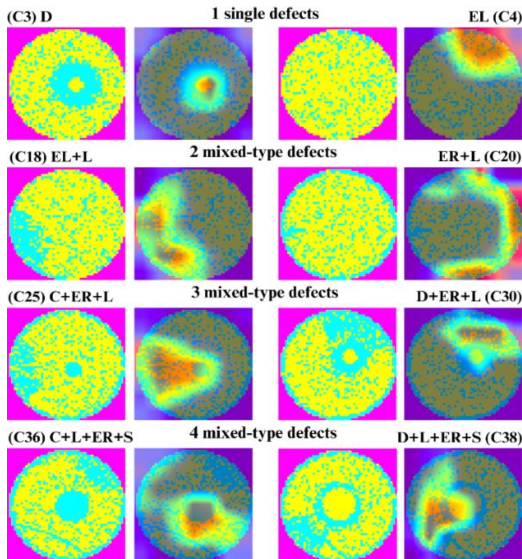

Fig. 3. Gradient-based visualization of feature maps generated by our model (trained with 50 images for each of all 38 classes) on some samples of 4 defect groups in MixedWM38.

As showed in Table V, we compare our classification results with DC-Net [18] and WaferSegClassNet (WSCN) [27] in the same setting of using 80% dataset for training the networks and 20% for validation. We obtain the average classification accuracy of 98.22% on all 38 classes, which is superior to DC-Net (93.2%). The score is also higher than that of the multi-task learning framework WSCN (98.2%) which also used the Contrastive loss. Although our method is only slightly lower than WSCN in some single defects, around 0.14% in average, its recognition accuracy for two mixed-type, three mixed-type and four mixed-type patterns are 98.69%, 97.69%, and 96.88%, respectively, and are better in all multi-defect clusters. This could indicate the effectiveness of learning features in classify complex compositions of wafer defects which are difficult to recognize, even for human eyes.

TABLE V. COMPARISON OF CLASSIFICATION ACCURACY ON EACH OF ALL 38 CLASSES OF MIXEDWM38 (WITH 80\% TRAINING DATA)

Class	DC-Net [18]	WSCN [27]	Proposed
C1 (Normal)	99.70	**100.00**	**100.00**
C2 (C)	97.80	**100.00**	**100.00**
C3 (D)	96.50	**100.00**	99.00
C4 (EL)	94.40	97.00	**99.00**
C5 (ER)	99.80	99.00	**100.00**
C6 (L)	93.80	99.00	**100.00**
C7 (NF)	95.80	**100.00**	96.60
C8 (S)	93.40	**99.00**	99.00
C9 (R)	**100.00**	98.00	96.10
Avg (1 defect classes)	96.80	**99.00**	98.86
C10 (C+EL)	**99.20**	98.00	98.00
C11 (C+ER)	97.90	**100.00**	**100.00**
C12 (C+L)	98.50	99.00	**100.00**
C13 (C+S)	96.70	99.00	**100.00**
C14 (D+EL)	**99.30**	94.00	98.00
C15 (D+ER)	96.10	99.00	**99.50**
C16 (D+L)	**98.30**	95.00	97.50
C17 (D+S)	92.80	**100.00**	99.00
C18 (EL+L)	93.90	**99.00**	**99.00**
C19 (EL+S)	92.30	**97.00**	96.50
C20 (ER+L)	94.60	96.00	**98.00**
C21 (ER+S)	90.70	**100.00**	99.00
C22 (L+S)	90.30	97.00	**98.50**
Avg (2 defect classes)	95.43	97.92	**98.69**
C23 (C+EL+L)	88.90	97.00	**99.00**
C24 (C+EL+S)	89.40	**99.00**	97.80
C25 (C+ER+L)	91.40	97.00	**99.00**
C26 (C+ER+S)	92.50	**100.00**	99.50
C27 (C+L+S)	90.50	97.00	**98.00**
C28 (D+EL+L)	88.30	97.00	**98.50**
C29 (D+EL+S)	90.50	**96.00**	94.00
C30 (D+ER+L)	92.30	**100.00**	99.00
C31 (D+ER+S)	91.50	98.00	**99.00**
C32 (D+L+S)	88.30	97.00	**97.50**
C33 (EL+L+S)	86.20	**96.00**	**96.00**
C34 (ER+L+S)	89.00	**97.00**	95.00
Avg (3 defect classes)	89.90	97.58	**97.69**
C35 (C+L+EL+S)	87.00	**94.00**	93.50
C36 (C+L+ER+S)	90.60	97.00	**98.00**
C37 (D+L+EL+S)	86.40	95.00	**96.50**
C38 (D+L+ER+S)	88.20	95.00	**99.50**
Avg (4 defect classes)	88.05	95.25	**96.88**
Avg (all 38 classes)	93.20	98.20	**98.22**

We further investigate the classification performance of our network with several small-data settings on MixedWM38. Table VI summarizes the comparison results of the ResNet-50 backbone and our model. Both models are only trained with 20, 50, and 100 samples of each defect pattern. The accuracy in overall are much superior to those of the backbone. Especially, we obtain the accuracy of 92.27% for all 7603 testing images (20% of MixedWM38) of 38 classes through training the model with only 20 images per class. Once again, it shows that the Contrastive loss and Arc loss make a significant contribution to the backbone encoder to generalize discriminative features of different classes with very small-scale data, but still avoid the over-fitting issue.

TABLE VI. RESULTS OF SMALL-DATA SETTINGS ON MIXEDWM38

Train Dataset (images/class)	ResNet-50 (Backbone)	Proposed
20	55.36	**92.27**
50	86.72	**96.88**
100	89.73	**97.83**

D. Discussion

We set up the small data experiments to compare our method with the backbone alone optimized by the categorical cross-entropy loss. However, large parameters of the deep learning-based model ResNet-50 make it difficult to converge at an optimal parameter set with a limited training data size. The phenomenon of over-fitting or under-fitting occurs when training the model with a scarce quantity of images, for example, the backbone gets the accuracy of 55.36% on the MixedWM38 test set with only 20 training samples per class. Furthermore, some small defects in the industry appear as regions of low contrast, non-uniform brightness, or irregular shape, which make it hard to generalize features for classification. As shown in Figs. 2 and 3, our model with a joint loss function pays close attention to defective regions, which significantly contributes to the classification performance.

Based on the results from few-shot learning experiments, our model has the potential but not optimal for one-shot recognition task. However, it is easy to adapt this approach with the minimal requirement of labeled data for classifying images of new defect types appearing in the manufacturing environment. With small labeled data, one of directions for future work can be applying the supervised contrastive learning [39] to avoid mis-recognition of the same-class images in a training mini-batch. The network optimization also can combine with a segmentation target for better recognizing and localizing fine-grained defects. In addition, to become better deployed to the actual production line, we will reduce the number of model parameters to improve recognition speed while maintaining accuracy.

V. CONCLUSION

In this paper, we introduced a potential approach to enhance image representation for surface defect recognition task with small data. We address the challenge of limited training data size by designing a training framework for the feature extractor with the supervision of data intensive, self-contrastive loss and an angular margin loss. By conducting extensive experiments, we achieve better accuracy performance compared to several current methods on two benchmark datasets. With few-shot learning and small data settings, this method also shows high capability to capture discriminative regions on the defect images and obtain the accuracy of 98.83% and 92.27% on NEU and MixedWM38, respectively, with only 20 training samples per class. These experimentations indicate that the method can reduce annotation costs and increase the defect recognition performance with representation learning approach. Future research will focus on applying the supervised contrastive learning and combining the segmentation target to further improve image representation.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

All authors proposed and discussed the idea, Thai Tieu Phuong implemented the methodology and wrote the paper. Duong Duc Tin and Le Hong Trang analyzed the data. This paper was revised by Le Hong Trang. All authors had approved the final version.

ACKNOWLEDGMENT

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

[1] Y. Chen, Y. Ding, F. Zhao, E. Zhang, Z. Wu, and L. Shao, "Surface defect detection methods for industrial products: A review," *Applied Sciences*, vol. 11, no. 16, 2021. https://doi.org/10.3390/app11167657

[2] S. B. Jha and R. F. Babiceanu, "Deep CNN-based visual defect detection: Survey of current literature," *Computers in Industry*, vol. 148, 103911, 2023. https://doi.org/10.1016/j.compind.2023.103911

[3] J. Wang, H. Dai, T. Chen, H. Liu, X. Zhang, Q. Zhong, and R. Lu, "Toward surface defect detection in electronics manufacturing by an accurate and lightweight YOLO-style object detector," *Scientific Reports*, vol. 13, 2023. https://doi.org/10.1038/s41598-023-33804-w

[4] C. Li, J. Li, Y. Li, L. He, X. Fu, and J. Chen, "Fabric defect detection in textile manufacturing: A survey of the state of the art," *Security and Communication Networks*, vol. 05, pp. 1–13, 2021. https://doi.org/ 10.1155/2021/9948808

[5] A. Rasheed, B. Zafar, A. Rasheed, N. Ali, M. Sajid, S. Dar, U. Habib, T. Shehryar, and M. Mahmood, "Fabric defect detection using computer vision techniques: A comprehensive review," *Mathematical Problems in Engineering*, vol. 11, 2020. https://doi.org/10. 1155/2020/8189403

[6] Q. Jin and L. Chen, "A survey of surface defect detection of industrial products based on a small number of labeled data," arXiv preprint, arXiv:2203.05733, 2022.

[7] C. Li, Y. Huang, L. Hai, and X. Zhang, "A weak supervision machine vision detection method based on artificial defect simulation," *Knowledge-Based Systems*, vol. 208, 106466, 2020. https://doi.org/10.1016/j.knosys. 2020.106466

[8] L. Liu, D. Cao, Y. Wu, and T. Wei, "Defective samples simulation through adversarial training for automatic surface inspection," *Neurocomputing*, vol. 360, 2019. https://doi.org/10.1016/j.neucom.2019.05.080

[9] M. Haselmann and D. Gruber, "Supervised machine learning based surface inspection by synthetizing artificial defects," in *Proc. 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, Mexico, 2017, pp. 390–395. doi: 10.1109/ICMLA.2017.0-130

[10] M. Ferguson, R. Ak, Y.-T. T. Lee, and K. H. Law, "Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning," arXiv preprint, arXiv:1808.02518, 2018.

[11] Y. Gong, J. Luo, H. Shao, and Z. Li, "A transfer learning object detection model for defects detection in X-ray images of spacecraft composite structures," *Composite Structures*, vol. 284, 115136, 2022. https://doi.org/10.1016/j.compstruct.2021.115136

[12] J. Liu, F. Guo, H. Gao, L. Maoyuan, Y. Zhang, and H. Zhou, "Defect detection of injection molding products on small datasets using transfer learning," *Journal of Manufacturing Processes*, vol.

70, pp. 400–413, 2021. https://doi.org/10.1016/j.jmapro.2021.08.034
[13] H. Di, X. Ke, Z. Peng, and D. Zhou, "Surface defect classification of steels with a new semi-supervised learning method," *Optics and Lasers in Engineering*, 2019. https://api.semanticscholar.org/CorpusID:126642866
[14] Y. Gao, "A semi-supervised convolutional neural network-based method for steel surface defect recognition," *Robotics and Computer-Integrated Manufacturing*, vol 61, 2019. https://doi.org/10.1016/j.rcim.2019.101825
[15] Y. He, K. Song, H. Dong, and Y. Yan, "Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network," *Optics and Lasers in Engineering*, vol. 122, pp. 294–302, 2019. https://doi.org/10.1016/j.optlaseng.2019.06.020
[16] G. Hu, J. Huang, Q.-H. Wang, J.-R. Li, Z. Xu, and X. Huang, "Unsupervised fabric defect detection based on a deep convolutional generative adversarial network," *Textile Research Journal*, vol. 90, 2019. https://doi.org/10.1177/0040517519862880
[17] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Applied Surface Science,* vol. 285, pp. 858–864, 2013. https://doi.org/10.1016/j.apsusc.2013.09.002
[18] J. Wang, C. Xu, Z. Yang, J. Zhang, and X. Li, "Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, no. 4, pp. 587–596, 2020. https://doi.org/10.1109/TSM.2020.3020985
[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. https://doi.org/10.1109/CVPR. 2009.5206848
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv preprint, arXiv:1512.03385, 2015.
[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, arXiv:1409.1556, 2015.
[22] M. Tan and Q.V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," arXiv preprint, arXiv:1905.11946, 2019.
[23] I. Konovalenko, P. Maruschak, J. Brezinová, J. Viňáš, and J. Brezina, "Steel surface defect classification using deep residual neural network," *Metals*, vol. 10, no. 6, 2020. https://doi.org/10.3390/met10060846
[24] 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, 2021. https://doi.org/10.3390/sym13040706
[25] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," arXiv preprint, arXiv:2101.03697, 2021.
[26] Z. Li, C. Wu, Q. Han, M. Hou, G. Chen, and T. Weng, "CASI-Net: A novel and effect steel surface defect classification method based on coordinate attention and self-interaction mechanism," *Mathematics*, vol. 10, no. 6, 2022. https://doi.org/10.3390/math10060963
[27] S. Nag, D. Makwana, S. C. Teja R, S. Mittal, and C. K. Mohan, "WaferSegClassNet—A light-weight network for classification and segmentation of semiconductor wafer defects," *Computers in Industry*, vol. 142, 103720, 2022. https://doi.org/10.1016/j.compind.2022.103720
[28] K. Liu, A. Li, X. Wen, H. Chen, and P. Yang, "Steel surface defect detection using GAN and one-class classifier," in *Proc. the 2019 25th International Conference on Automation and Computing (ICAC)*, 2019, pp. 1–6. https://doi.org/10.23919/IConAC.2019.8895110
[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," arXiv preprint, arXiv:1406.2661, 2014
[30] A. M. Deshpande, A. A. Minai, and M. Kumar, "One-shot recognition of manufacturing defects in steel surfaces," *Procedia Manufacturing*, vol. 48, pp. 1064–1071, 2020. https://doi.org/10.1016/j.promfg.2020.05.146
[31] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," arXiv preprint, arXiv:2006.09882, 2021.
[32] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," arXiv preprint, arXiv:1911.05722, 2020.
[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," arXiv preprint, arXiv:2002.05709, 2020.
[34] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015. https://doi.org/10. 1109/cvpr.2015.7298682
[35] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," arXiv preprint, arXiv:1704.08063, 2018.
[36] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 5962–5979, 2022. https://doi.org/10.1109/tpami.2021.3087709
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint, arXiv:1412.6980, 2017.
[38] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2019. https://doi.org/10.1007/s11263-019-01228-7
[39] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," arXiv preprint, arXiv:2004.1136, 2021.

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.