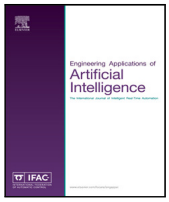
Engineering Applications of Artificial Intelligence 126 (2023) 107154

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Supervised contrastive learning for wafer map pattern classification

Youngjae Bae, Seokho Kang *

Department of Industrial Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea

ARTICLE INFO

Keywords:
Semiconductor manufacturing
Wafer map pattern classification
Convolutional neural network
Supervised contrastive learning

ABSTRACT

In the semiconductor manufacturing process, analyzing the defect patterns on a wafer map is crucial for identifying the causes of the defects. The advent of convolutional neural networks (CNNs) has significantly increased the accuracy of automated wafer map pattern classification. Generally, the use of a larger training dataset results in higher classification accuracy. However, collecting a large number of wafer maps and labeling them with their defect categories is expensive and time-consuming. In this paper, we present an improved training method under data insufficiency for wafer map pattern classification. We apply supervised contrastive learning to train a CNN by exploiting the rotational-invariant characteristic of wafer map labeling. The CNN is trained by simultaneously minimizing two loss functions: classification loss and contrastive loss. The first loss function is to classify the rotational variants of wafer maps accurately. The second loss function is to align the representation vectors for the rotational variants of wafer maps with similar labels to be close to each other. Using two benchmark datasets, WM-811K and MixedWM38, we demonstrate that the proposed method enhances classification accuracy compared with existing methods, particularly when the training dataset is small.

1. Introduction

In the semiconductor manufacturing process, each die in a wafer is electrically tested after fabrication to determine if it functions as designed. The locations of defective dies in the wafer are visualized using a wafer map. The spatial defect pattern of a wafer map provides clues for inferring potential problems during the manufacturing process (Hansen et al., 1997). For example, problems in etching and mechanical operations may cause ring-shaped and linear scratch-shaped patterns, respectively (Wang et al., 2006). Thus, wafer map pattern classification plays a substantial role in improving the yield of the manufacturing process.

Owing to the increasing complexity and scale of the semiconductor manufacturing process, the significance of prompt and accurate analysis of defect causes has been emphasized. In the past, wafer map pattern classification was performed manually by repetitive labor of process engineers. Thus, it takes a longer time to classify a large number of wafer maps. In addition, the classification accuracy relies heavily on the proficiency of engineers (Huang and Pan, 2015).

Numerous researchers have attempted to automate wafer map pattern classification tasks using machine learning. Early studies developed classification models based on manual feature extraction. A wafer map was transformed into a vector of meaningful handcrafted features based on domain knowledge, such as density, geometry, and radon features. Subsequently, an off-the-shelf classification model, such as a support vector machine or decision tree, was built on top of the vector representation (Li and Huang, 2009; Wu et al., 2015; Piao et al., 2018). More recently, the use of convolutional neural networks (CNNs), which directly operate on a wafer map without manual feature extraction, has resulted in significant performance improvements (Nakazawa and Kulkarni, 2018; Kyeong and Kim, 2018; Yu et al., 2021). Owing to their effectiveness, CNNs have become the mainstream classification models for wafer map pattern classification.

To deploy a classification model into a real-world manufacturing process, the model should be as accurate as a professional engineer. This necessitates the acquisition of a large training dataset to establish the model, which is difficult in practice owing to time and cost constraints. Various research attempts have been made to address this difficulty, including data augmentation (Kang, 2020; Wang and Chen, 2020; Shin et al., 2022; Shawon et al., 2019; Wang et al., 2019), unsupervised pre-training (Hu et al., 2021; Kahng and Kim, 2021), semi-supervised learning (Kong and Ni, 2020; Lee and Kim, 2020), and active learning (Kong and Ni, 2020; Shim et al., 2020, 2021). As in previous studies, the primary purpose of our study is to establish a more accurate classification model by better learning from an insufficient training dataset.

In this paper, we present a supervised contrastive learning approach to achieve this purpose. To adapt supervised contrastive learning to

* Corresponding author.
E-mail addresses: qodudwos@skku.edu (Y. Bae), s.kang@skku.edu (S. Kang).

https://doi.org/10.1016/j.engappai.2023.107154
Received 3 November 2022; Received in revised form 25 July 2023; Accepted 12 September 2023
Available online 22 September 2023

wafer map pattern classification, we exploit the rotation-invariant characteristics of wafer map labeling and introduce a label similarity weighting scheme that can be generalized to both multi-class and multi-label settings. Given a training dataset containing wafer maps labeled with their defect categories, we train a CNN by minimizing the learning objective which involves two loss functions. The first loss is the classification loss to accurately classify the rotational variants of the wafer maps into their defect categories. The second loss is the contrastive loss to align the representation vectors of the rotational variants of wafer maps with similar labels to be close. This is achieved by introducing a label similarity weight based on Jaccard similarity coefficient. The proposed method can be applied to any type of CNN architecture with the addition of an auxiliary projection head. We validated the effectiveness of the proposed method experimentally using multi-class and multi-label wafer map datasets.

The remainder of this paper is organized as follows. Section 2 reviews the related studies on wafer map pattern classification and contrastive learning. The proposed method is described in Section 3. Section 4 presents the experimental results. Finally, in Section 5, conclusions and future work are presented.

2. Related work

2.1. Wafer map pattern classification

The application of CNNs to wafer map pattern classification has improved classification accuracy. CNNs benefit from feature learning on raw wafer maps using a stack of multiple convolutional layers (LeCun et al., 2015). Existing studies have primarily focused on training a CNN in a multi-class setting where each wafer map is assumed to have a single defect pattern (Nakazawa and Kulkarni, 2018; Kang, 2020; Kahng and Kim, 2021; Ishida et al., 2019; Yu et al., 2019, 2021). Some studies have addressed a multi-label setting where each wafer map can have a mixed-type defect pattern and thus belong to more than one defect category (Kyeong and Kim, 2018; Kong and Ni, 2019; Shin et al., 2022; Wang et al., 2020).

As mentioned previously, a primary challenge is to make a CNN as accurate as possible by using an insufficient training dataset because collecting and labeling a large number of wafer maps is time-consuming and costly. There have been various research attempts to address the data insufficiency problem, which we categorize into the following four research directions.

- **Data augmentation**: Data augmentation creates synthetic instances using the original training dataset without additional data collection or annotation costs (Shorten and Khoshgoftaar, 2019). The synthetic instances are further used to train a CNN with better classification accuracy. One strategy of data augmentation is to create different views of original wafer maps in the training dataset by applying label-preserving transformations. Kang (2020) created new wafer maps by randomly rotating and flipping an existing wafer map about its center. Wang and Chen (2020) mapped a wafer map into a matrix representation by polar coordinates mapping and produced new matrices by randomly shifting the angle dimension in the matrix, which had the same effect as rotating the wafer map. Shin et al. (2022) combined two or three wafer maps using mixup to synthesize a wafer map with a mixed-type pattern. Chiu and Chen (2021) combined two randomly rotated distinct wafer maps to generate a mixed-type pattern wafer map. Shim and Kang (2023) synthesized mixed-type pattern wafer maps using mixup, random rotation, and noise filtering. Another strategy is to train a generative model using the original training dataset and to employ the model to produce synthetic wafer maps. Shawon et al. (2019) and Wang et al. (2019) employed a convolutional autoencoder and generative adversarial network, respectively, for the purpose.
- **Unsupervised pre-training**: Unsupervised pre-training of a CNN helps to better learn useful features from the training dataset for the target task. Hu et al. (2021) and Kahng and Kim (2021) adopted contrastive learning with data augmentation to pre-train a CNN in an unsupervised manner without using label information. Shon et al. (2021) employed a convolutional variational autoencoder for unsupervised pre-training of a CNN. Following pre-training, the CNN was fine-tuned in a supervised manner for wafer map pattern classification.
- **Semi-supervised learning**: The semi-supervised learning approach utilizes both labeled and unlabeled instances to train the CNN (Van Engelen and Hoos, 2020). This is useful when there are abundant wafer maps; however, only a few of them are labeled with their defect categories owing to the high annotation cost. Compared to the supervised learning approach, which utilizes only labeled wafer maps for training, the classification accuracy can be further enhanced by utilizing the information of unlabeled wafer maps. Kong and Ni (2020) employed a ladder network (Rasmus et al., 2015) and semi-supervised variational autoencoder (Kingma et al., 2014) to train a CNN in a semi-supervised manner. Moreover, they utilized pseudo-labeling to increase the number of labeled wafer maps. Kang and Kang (2023) presented a semi-supervised representation learning method to acquire rotational-invariant representations of wafer maps by learning from a partially-labeled dataset. Lee and Kim (2020) extended a semi-supervised variational autoencoder to classify wafer maps with mixed-type defect patterns.
- **Active learning**: Active learning interactively selects unlabeled instances to be queried for their labels to improve the classification accuracy of a CNN (Ren et al., 2021). When unlabeled wafer maps are abundant, some of them can be labeled at a cost. The selective labeling of informative unlabeled wafer maps helps reduce the annotation cost required to achieve the desired classification accuracy. Kong and Ni (2020) selected the wafer maps with the highest information entropy for querying labels. Shim et al. (2020) compared various classification uncertainty measures that are applicable to active learning of a CNN. Shim et al. (2021) enhanced the cost efficiency of active learning by utilizing cluster-level annotations. A cluster of similar wafer maps instead of a single wafer map was queried for the label in each interaction.

This study aims to improve the training of a CNN with an insufficient training dataset to mitigate the difficulties arising from the high cost of acquiring and labeling a large training dataset. We extend the idea of an unsupervised pre-training approach. While existing unsupervised pre-training methods (Hu et al., 2021; Kahng and Kim, 2021; Shon et al., 2021) sequentially performed unsupervised pre-training and supervised fine-tuning, the proposed method trains a CNN by minimizing the learning objective that simultaneously incorporates contrastive learning and supervised fine-tuning. As baselines, we employ representative methods in data augmentation and unsupervised pre-training approaches.

2.2. Contrastive learning

Contrastive learning aims to learn the representations of data by contrasting similar and dissimilar instances (Chen et al., 2020b). A model that maps the input to a representation vector is built. For the training dataset, positive and negative pairs of instances are defined according to their contextual similarity. The model is trained to make representation vectors between positive pairs closer and negative pairs farther.

It is crucial to use a proper strategy to define the positive and negative pairs of instances based on data characteristics (Le-Khac et al., 2020). Widely used strategies can be categorized as follows.

- **Data augmentation**: This strategy is typically used for self-supervised representation learning. Transform operations that preserve the original context of an instance are used to define positive and negative pairs. Generally, a positive pair comprises two different views of an instance, whereas a negative pair comprises the views of two different instances (Chen et al., 2020b; Misra and Maaten, 2020). Chen et al. (2020b) proposed a simple contrastive learning framework named SimCLR to train a CNN for visual-representation learning. Chen et al. (2020c) improved SimCLR using a larger CNN architecture and knowledge distillation. Grill et al. (2020) presented bootstrap your own latent (BYOL), which trains two CNNs that interact and learn from each other without using negative pairs. Misra and Maaten (2020) proposed pre-text invariant representation learning (PIRL) that uses a memory bank of negative instances to define negative pairs. He et al. (2020) proposed momentum contrast (MoCo), which uses a dictionary of negative instances that are dynamically updated with momentum values to define negative pairs. Hu et al. (2021) and Kahng and Kim (2021) adapted the data augmentation strategy to pre-train a CNN for wafer map pattern classification.
- **Label information**: If the label information of the training dataset is available, the labels can be used to define the positive and negative pairs of instances (Khosla et al., 2020). A positive pair of instances belongs to the same class, whereas a negative pair belongs to different classes. This strategy is referred to as supervised contrastive learning. Khosla et al. (2020) and Małkiński and Mańdziuk (2022) extended SimCLR to utilize label information for the pre-training of a CNN under multi-class and multi-label settings, respectively. Following pre-training, the CNN was fine-tuned to perform downstream classification task. Li et al. (2022) presented selective supervised contrastive learning that only uses confident pairs of instances to alleviate the side effects of noisy labels. Supervised contrastive learning methods benefit from the use of label information for better representation learning and can be useful when strong restrictions are imposed on data augmentation (Jaiswal et al., 2021).
- **Similarity measure**: Positive and negative pairs of instances can be defined using a similarity measure. Li et al. (2021) derived clusters of data instances by performing clustering on the learned embedding space based on an expectation–maximization algorithm and used the instances in the same cluster as positive pairs. Zheng et al. (2021) employed instances whose embeddings were close to each other to form positive pairs.
- **Heterogeneous observations**: Different types of observations for the same instance can be used as positive pairs. Sermanet et al. (2017) used multiple cameras in different positions to capture the target object and used different views of the same object as positive pairs and views of different objectives as negative pairs. This strategy is related to cross-modal contrastive learning. Several methods used a pair of images and text representing the same object as a positive pair and those from different objects as a negative pair for vision-language representation learning (Wen et al., 2021; Zhang et al., 2021; Zolfaghari et al., 2021).

In this study, we utilize the first and second strategies for supervised contrastive learning of a CNN. To leverage the rotational-invariant characteristics and label information of wafer maps, we define positive pairs as rotational variants of wafer maps with similar labels. To enhance classification performance under the insufficiency of the training dataset, we introduce a learning objective that encourages the model to accurately classify wafer maps and closely align the representation vectors of positive pairs at the same time.

3. Proposed method

3.1. Overview

Depending on the type of labels for wafer maps, we formulate the problem of wafer map pattern classification as either a multi-class or multi-label classification task. In a multi-class classification task, each wafer map is assumed to have a single defect pattern and thus belongs to a single defect category. In a multi-label classification task, each wafer map can have mixed-type defect patterns and thus can belong to more than one defect category.

The training dataset $\mathcal{D} = \{(\mathbf{X}_n, \mathbf{y}_n)\}_{n=1}^{N}$ comprises of N wafer maps and their respective class labels. The wafer map $\mathbf{X}_n \in \{0,1\}^{p\times q}$ is represented as a matrix, in which each element has a value of 1 if the corresponding die is defective and 0 if it is non-defective or outside the wafer region. The label $\mathbf{y}_n = [y_{n,1}, \ldots, y_{n,C}] \in \{0,1\}^C$ is a one-hot or multi-hot vector that indicates the defect categories to which the wafer map belongs.

The goal is to develop a classification model that predicts the defect categories of a wafer map. The proposed method adapts supervised contrastive learning with rotation-based data augmentation to train the model. The model is trained by simultaneously minimizing two loss functions: classification and contrastive losses. Fig. 1 illustrates the framework of the proposed method. The architecture of the model and learning objectives are described in the following subsections.

3.2. Model architecture

The proposed model comprises three components: encoder h, projection head g, and classification head f. The encoder h is a CNN that maps the wafer map $\mathbf{X}$ into a d-dimensional representation vector $\mathbf{r}$ as follows:

$$\mathbf{r} = h(\mathbf{X}), \quad \mathbf{r} \in \mathbb{R}^d. \tag{1}$$

The representation vector $\mathbf{r}$ is used as the input for projection head g and classification head f. The projection head g is a fully-connected layer that reduces the dimensionality of the representation vector $\mathbf{r}$ to d' as follows:

$$\mathbf{z} = g(\mathbf{r}), \quad \mathbf{z} \in \mathbb{R}^{d'}. \tag{2}$$

The classification head f is a fully-connected layer that outputs the prediction for wafer map $\mathbf{X}$ in the form of a C-dimensional response vector $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = [\hat{y}_1, \ldots, \hat{y}_C] = f(\mathbf{r}), \quad \hat{\mathbf{y}} \in [0,1]^C. \tag{3}$$

In the case of multi-class classification, we use softmax activation for the classification head f such that the summation of the elements in $\hat{\mathbf{y}}$ equals 1. In the case of multi-label classification, sigmoid activation is used such that many elements can have values close to 1.

3.3. Training

To train the model, we introduce the learning objective $\mathcal{J}$ that involves the classification loss function $\mathcal{L}_{\text{cls}}$ and contrastive loss function $\mathcal{L}_{\text{con}}$. The classification loss function $\mathcal{L}_{\text{cls}}$ is used to make the classification head f accurately predict the class labels of wafer maps. The contrastive loss function $\mathcal{L}_{\text{con}}$ is used to make the projection head g align the projected representations for wafer maps having similar labels to be close to each other.

For a mini-batch $\mathcal{S} = \{(\mathbf{X}_m, \mathbf{y}_m)\}_{m=1}^{M} \subset \mathcal{D}$ at each training iteration, we apply rotation-based data augmentation to create rotational variants of wafer maps in the mini-batch $\mathcal{S}$. For each wafer map $(\mathbf{X}_m, \mathbf{y}_m) \in \mathcal{S}$, we use transformations t_{2m-1}, t_{2m} randomly sampled from augmentation policy $\mathcal{A}$ to generate two different views $(\tilde{\mathbf{X}}_{2m-1}, \tilde{\mathbf{y}}_{2m-1})$ and $(\tilde{\mathbf{X}}_{2m}, \tilde{\mathbf{y}}_{2m})$. Owing to the rotation-invariant characteristic of wafer map labeling, the rotation operation is a label-preserving transformation of the wafer

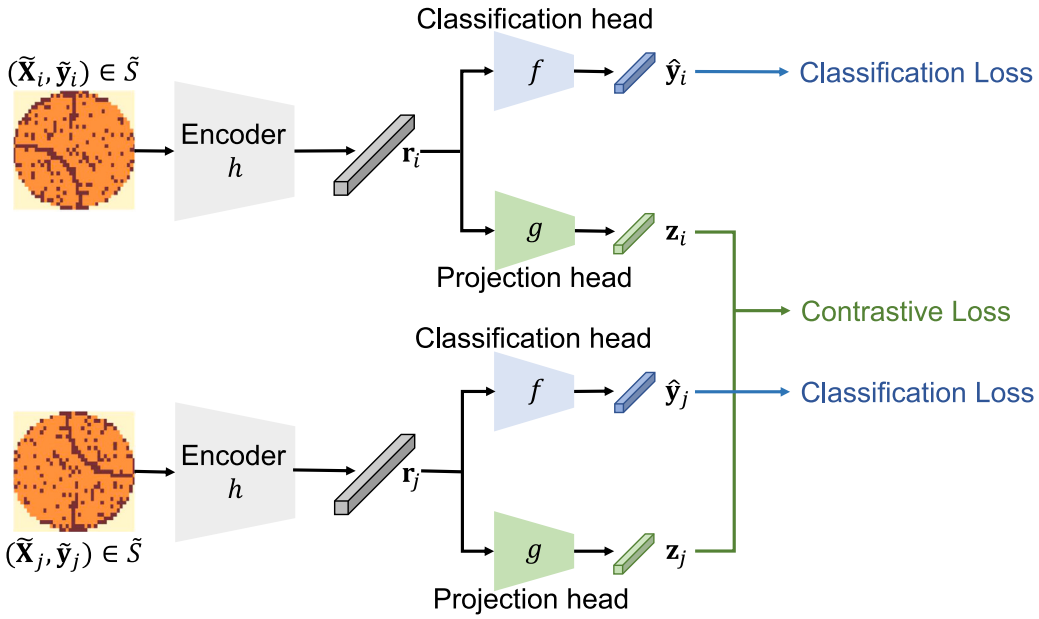


Fig. 1. Framework of the proposed method.

maps. Thus, the labels of the two views, $\tilde{\mathbf{y}}_{2m-1}$ and $\tilde{\mathbf{y}}_{2m}$, are equivalent to $\mathbf{y}_m$.

$$\begin{aligned} \tilde{\mathbf{X}}_{2m-1} &= t_{2m-1}(\mathbf{X}_m); \qquad \tilde{\mathbf{y}}_{2m-1} = \mathbf{y}_m; \\ \tilde{\mathbf{X}}_{2m} &= t_{2m}(\mathbf{X}_m); \qquad \tilde{\mathbf{y}}_{2m} = \mathbf{y}_m. \end{aligned} \tag{4}$$

Following data augmentation, we acquire an augmented mini-batch $\tilde{S} = \{(\tilde{\mathbf{X}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^{2M}$, using which the learning objective $\mathcal{J}$ is computed.

The classification loss function $\mathcal{L}_{\text{cls}}$ evaluates the difference between the ground-truth label $\tilde{\mathbf{y}}_i$ and the prediction $\hat{\mathbf{y}}_i = f(h(\tilde{\mathbf{X}}_i))$ for the wafer map $\tilde{\mathbf{X}}_i$. This differs between multi-class and multi-label classification settings. Under the multi-class setting, the categorical cross-entropy function is used:

$$\mathcal{L}_{\text{cls}}(\tilde{\mathbf{y}}_i, \hat{\mathbf{y}}_i) = \sum_{i=1}^{2M} \sum_{c=1}^{C} \left[-\tilde{y}_{i,c} \log \hat{y}_{i,c} \right]. \tag{5}$$

Under the multi-label setting, the binary cross-entropy function is used:

$$\mathcal{L}_{\text{cls}}(\tilde{\mathbf{y}}_i, \hat{\mathbf{y}}_i) = \sum_{i=1}^{2M} \sum_{c=1}^{C} \left[-\tilde{y}_{i,c} \log \hat{y}_{i,c} - (1 - \tilde{y}_{i,c}) \log(1 - \hat{y}_{i,c}) \right]. \tag{6}$$

These two functions are commonly used as learning objectives for supervised learning in multi-class and multi-label classification tasks, respectively. A smaller value of each function indicates that the CNN more accurately predicts the label $\tilde{\mathbf{y}}_i$ of a given wafer map $\tilde{\mathbf{X}}_i$.

The contrastive loss function $\mathcal{L}_{\text{con}}$ evaluates the pairwise dissimilarity, in the embedding space, between the wafer map $\tilde{\mathbf{X}}_i$ and other wafer maps in $\tilde{S}$ whose labels are similar to $\tilde{\mathbf{y}}_i$. Let $\mathbf{z}_i$ denote the embedded vector obtained using the projection head g, *i.e.*, $\mathbf{z}_i = g(h(\tilde{\mathbf{X}}_i))$. The loss function $\mathcal{L}_{\text{con}}$ for the wafer map $(\tilde{\mathbf{X}}_i, \tilde{\mathbf{y}}_i)$ is calculated as follows:

$$\begin{aligned} &\mathcal{L}_{\text{con}}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{y}}_i; \tilde{S}) \\ &= -\frac{1}{\sum_{j=1}^{2M} \mathbf{1}(i \neq j) w_{i,j}} \log \left(\sum_{j=1}^{2M} \frac{\mathbf{1}(i \neq j) w_{i,j} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2M} \mathbf{1}(i \neq k) \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \right), \end{aligned} \tag{7}$$

where sim denotes the cosine similarity and τ indicates the temperature scaling hyperparameter. A smaller value of $\mathcal{L}_{\text{con}}$ indicates that the wafer maps with similar labels and their rotational variants are closely aligned in the embedding space.

The label similarity weight $w_{i,j}$ quantifies the similarity between two labels $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_j$ using the Jaccard similarity coefficient as follows:

$$w_{i,j} = \begin{cases} 1, & \text{if } \tilde{\mathbf{y}}_i = \tilde{\mathbf{y}}_j = \mathbf{0}; \\ \frac{\tilde{\mathbf{y}}_i \cdot \tilde{\mathbf{y}}_j}{\|\tilde{\mathbf{y}}_i\|_1 + \|\tilde{\mathbf{y}}_j\|_1 - \tilde{\mathbf{y}}_i \cdot \tilde{\mathbf{y}}_j}, & \text{otherwise.} \end{cases} \tag{8}$$

The introduction of $w_{i,j}$ in the contrastive loss enables the evaluation of the similarity between labels under both multi-class and multi-label settings. Under the multi-class setting, $w_{i,j}$ has a value of 1 if the labels $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_j$ exactly match, meaning that the wafer maps $\tilde{\mathbf{X}}_i$ and $\tilde{\mathbf{X}}_j$ belong to the same defect category, and a value of 0 otherwise. Thus, $\mathcal{L}_{\text{con}}$ becomes equivalent to the contrastive loss function used in the work of Khosla et al. (2020). Under the multi-label setting, $w_{i,j}$ is an intersection of $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_j$ over a union of $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_j$. Unlike the conventional Jaccard similarity coefficient, if both labels are zero vectors, then $w_{i,j}$ is forced to 1. The label similarity weight $w_{i,j}$ allows a pair of wafer maps with partially overlapping labels to be handled appropriately in the multi-label setting.

Finally, the learning objective $\mathcal{J}$ represents the weighted sum of the two loss functions $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{con}}$ averaged over the augmented mini-batch $\tilde{S}$:

$$\mathcal{J} = \frac{1}{2M} \sum_{(\tilde{\mathbf{X}}_i, \tilde{\mathbf{y}}_i) \in \tilde{S}} \left[\mathcal{L}_{\text{cls}}(\tilde{\mathbf{y}}_i, \hat{\mathbf{y}}_i) + \gamma \cdot \mathcal{L}_{\text{con}}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{y}}_i; \tilde{S}) \right], \tag{9}$$

where $\gamma > 0$ is a hyperparameter that adjusts the trade-off between the two losses. We train the model components f, g, and h to minimize the learning objective $\mathcal{J}$.

3.4. Inference

Once the model is trained, it can be used to predict the defect categories of the unseen wafer maps. At inference, encoder h and classification head f are used to make a prediction, whereas the projection head g is not used.

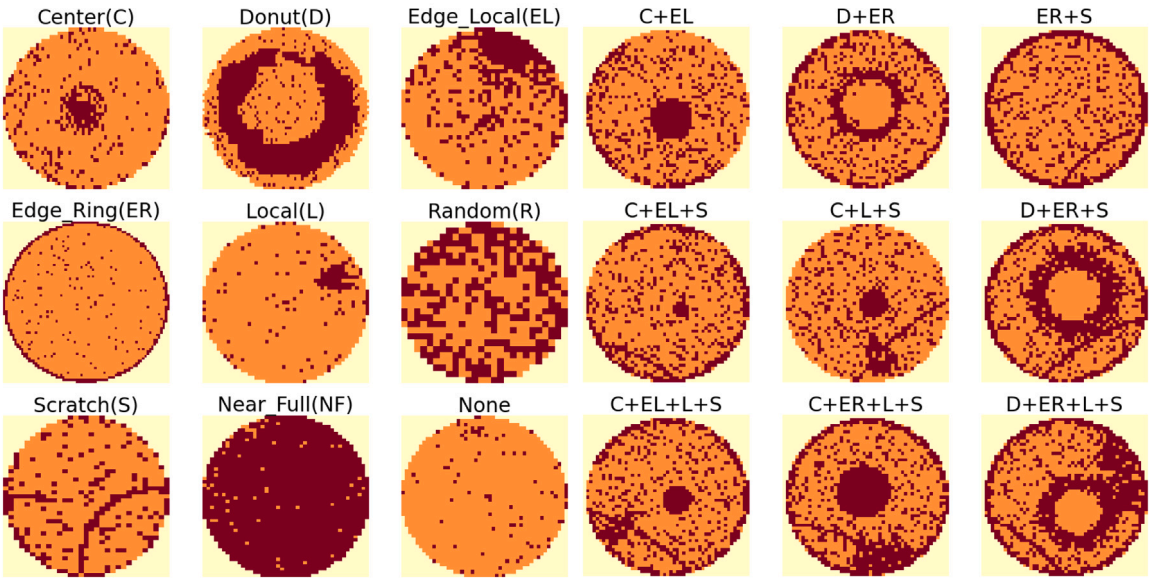
Given a query wafer map $\mathbf{X}_*$, we obtain the response vector $\hat{\mathbf{y}}_*$ by passing $\mathbf{X}_*$ through encoder h and classification head f:

$$\hat{\mathbf{y}}_* = [\hat{y}_{*,1}, \ldots, \hat{y}_{*,C}] = f(h(\mathbf{X}_*)), \tag{10}$$

where each element $\hat{y}_{*,j}$ can be interpreted as the probability estimate that the query wafer map $\mathbf{X}_*$ has a corresponding defect pattern.

Table 1
Distribution of the defect categories in benchmark datasets.

Category	WM-811K		MixedWM38	
	No. Wafer Maps	Proportion	No. Wafer Maps	Proportion
Center	4,294	2.48%	13,000	34.20%
Donut	555	0.32%	12,000	31.57%
Edge-Local	5,189	5.60%	13,000	34.20%
Edge-Ring	9,680	3.00%	12,000	31.57%
Local	3,593	2.08%	18,000	47.35%
Random	866	0.50%	866	2.28%
Scratch	1,193	0.69%	19,000	49.98%
Near-Full	149	0.09%	149	0.39%
None	147,427	85.24%	1,000	2.63%
Total	172,946	100.00%	38,015	100.00%



(a) WM-811K dataset (b) MixedWM38 dataset

Fig. 2. Examples of wafer maps and the corresponding defect categories.

Vector $\hat{\mathbf{y}}_*$ is further processed to determine which defect categories the query wafer map $\mathbf{X}_*$ is predicted to belong to. In the case of multi-class classification, the predicted defect category for $\mathbf{X}_*$ is obtained as $\text{argmax}_c\ \hat{y}_{*,c}$. In the case of multi-label classification, the set of predicted defect categories for $\mathbf{X}_*$ is obtained as $\{c|\hat{y}_{*,c} > 0.5\}$.

4. Experiments

4.1. Data description

To investigate the effectiveness of the proposed method, we conducted experiments using two benchmark datasets: WM-811K (Wu et al., 2015) and MixedWM38 (Wang et al., 2020). Fig. 2 illustrates the examples of wafer maps and their defect categories in the datasets. Table 1 lists the distribution of defect categories in the datasets.

- **WM-811K** (Wu et al., 2015) is a multi-class dataset containing 811,457 wafer maps collected from a semiconductor manufacturer. We excluded all unlabeled wafer maps and four abnormal wafer maps having less than 100 dies. Because the sizes of wafer maps varied from 6 × 21 to 300 × 201, we resized each wafer map to 64 × 64 by applying bilinear interpolation. After preprocessing, we obtained 172,946 wafer maps along with their corresponding defect categories. Each wafer map had a single defect pattern belonging to one of the following defect categories: *Center*, *Donut*, *Edge-Local*, *Edge-Ring*, *Local*, *Random*, *Scratch*, *Near-Full*, and *None*. We represented the label of each wafer map as a 9-dimensional one-hot vector. The distribution of defect categories was extremely imbalanced. Approximately 85.25% of the wafer maps belonged to the *None* category. We constructed a multi-class classification model using this dataset.
- **MixedWM38** (Wang et al., 2020) is a multi-label dataset containing 38,015 wafer maps with mixed-type defect patterns for nine defect categories: *Center*, *Donut*, *Edge-Local*, *Edge-Ring*, *Local*, *Random*, *Scratch*, *Near-Full*, and *None*. The dataset comprises wafer maps collected from a semiconductor manufacturer, as well as wafer maps synthesized by a generative adversarial network. All the wafer maps in this dataset had the same shape of 52 × 52, and thus, additional resizing was not necessary. We represented the label of each wafer map as an 8-dimensional multi-hot vector. Each wafer map was labeled with up to four categories. The label with zero vector indicated *None* category. In the distribution of defect categories, *Near-Full*, *Random*, and *None* categories were relatively rare compared with the other categories. We developed a multi-label classification model using this dataset.

4.2. Experimental settings

In the experiments, we randomly split each dataset in an 8:2 ratio. The former was used to construct the training set and the latter as the test set. To simulate various levels of data insufficiency, the size of the training set N was varied by random sampling as 100, 200, 500, 1,000, 2,000, 5,000, 10,000, and 20,000, respectively.

For the proposed method, the model was configured as follows. For the encoder h, we used a modified version of VGG16 (Simonyan and Zisserman, 2014), as it has exhibited performance superior or at least comparable to other CNN architectures in related studies (Shin et al., 2022; Ishida et al., 2019; Kang and Kang, 2021). Because the wafer maps were single-channeled, we reduced the number of input channels to 1. We removed all the fully-connected layers and replaced the flatten operation with global average pooling to output a 512-dimensional representation vector. The projection head g was a single fully-connected layer with a dimensionality of 128. The classification head f was also a single fully-connected layer with a dimensionality of 9 and softmax activation for the WM-811K dataset and dimensionality of 8 and sigmoid activation for the MixedWM38 dataset.

The following configurations were used to train the model. For the data augmentation policy $\mathcal{A}$, we used a random rotation operation with an angle sampled from Uniform$(-\pi, \pi)$ and a random flip operation with a probability of 0.5. The trade-off hyperparameter γ in the learning objective was set to 0.1. The temperature-scaling hyperparameter τ was set to 0.1 (Khosla et al., 2020). We used 80% of the training set to update the model parameters and the remaining 20% was used for validation. For parameter updating, we used the Adam optimizer with a learning rate of 10^{-5} and a mini-batch size of 128. Training was terminated if the validation performance did not improve over 50 consecutive epochs or the number of training epochs reached 500.

To evaluate the classification performance, we calculated Micro-F1 and Macro-F1 scores on the test set. F1 score is the harmonic mean of precision and recall. The terms Micro- and Macro- indicate how metrics are averaged across multiple categories. Micro-F1 is calculated by averaging the instance-wise performances as below:

$$\text{Micro-P} = \frac{\sum_j \text{TP}^j}{\sum_j \text{TP}^j + \sum_j \text{FP}^j}; \tag{11}$$

$$\text{Micro-R} = \frac{\sum_j \text{TP}^j}{\sum_j \text{TP}^j + \sum_j \text{FN}^j}; \tag{12}$$

$$\text{Micro-F1} = \frac{2 \times \text{Micro-P} \times \text{Micro-R}}{\text{Micro-P} + \text{Micro-R}}, \tag{13}$$

where TP^j, FP^j, FN^j, P^j, and R^j denote the true positive, false positive, false negative, precision, and recall, respectively, for the jth category. Macro-F1 is calculated by averaging the category-wise performances as below:

$$\text{P}^j = \frac{\text{TP}^j}{\text{TP}^j + \text{FP}^j}; \tag{14}$$

$$\text{R}^j = \frac{\text{TP}^j}{\text{TP}^j + \text{FN}^j}; \tag{15}$$

$$\text{Macro-F1} = \frac{1}{C} \sum_j \frac{2 \times \text{P}^j \times \text{R}^j}{\text{P}^j + \text{R}^j}. \tag{16}$$

Micro-F1 is heavily influenced by the majority categories, whereas Macro-F1 allows each category to contribute equally to the overall score. We also calculated category-level F1 scores for individual defect categories:

$$\text{F1}^j = \frac{2 \times \text{P}^j \times \text{R}^j}{\text{P}^j + \text{R}^j}. \tag{17}$$

We performed 10 repetitions of the experiments using different random seeds. The means and standard deviations of the results are presented. The PyTorch implementation of the proposed method is available online at https://github.com/YoungJaeBae/supcon_wmpc.

4.3. Compared methods

The classification performance of the proposed method was compared with the following baselines. For each method, all experimental settings not specified below were set the same as those in Section 4.2.

- **Supervised Learning based on Manual Feature Extraction (SL-MFE)** (Saqlain et al., 2019; Kang and Kang, 2021): Following Kang and Kang (2021)'s study, we extracted a 59-dimensional feature vector including density, geometry, and radon features from each wafer map. As the classification model, we used a fully-connected neural network (FNN) consisting of two hidden layers with 128 tanh units. The FNN takes the feature vector of a wafer map as input to predict the defect categories.
- **Supervised Learning using CNN without Data Augmentation (SL-CNN)**: A CNN was trained to minimize the classification loss using the training set without data augmentation.
- **Supervised Learning using CNN with Data Augmentation (SL-CNN-DA)**: Further from **SL-CNN**, data augmentation was applied during the training of the CNN.
- **Supervised Learning using Stacking Ensemble of MFE and CNN-DA (SL-Stacking)** (Kang and Kang, 2021): We constructed a stacking ensemble of two classification models, **SL-MFE** and **SL-CNN-DA**. Following Kang and Kang (2021)'s study, we employed a multi-response linear regression model as the meta-classifier. The meta-classifier utilizes the predictions from the base classifiers to make the final prediction.
- **Self-Supervised Contrastive Learning based on SimCLR (SSCL-SimCLR)** (Chen et al., 2020b; Hu et al., 2021; Kahng and Kim, 2021): A CNN was pre-trained using SimCLR (Chen et al., 2020b) without using label information. Subsequently, it was fine-tuned by supervised learning using label information. We applied data augmentation during fine-tuning of the CNN.
- **Self-Supervised Contrastive Learning based on MoCo (SSCL-MoCo)** (Chen et al., 2020a): It transitioned from **SSCL-SimCLR** by replacing the pre-training method from SimCLR to MoCo (He et al., 2020), while maintaining all other configurations unchanged.
- **Supervised Contrastive Learning with Binary Weighting (SCL-BW)** (Małkiński and Mańdziuk, 2022): From the proposed method, we replaced the label similarity weighting scheme $w_{i,j}$ as follows, by referring to Małkiński and Mańdziuk (2022)'s study:

$$w_{i,j} = \begin{cases} 1, & \text{if } \tilde{\mathbf{y}}_i = \tilde{\mathbf{y}}_j = \mathbf{0} \text{ or } \tilde{\mathbf{y}}_i \cdot \tilde{\mathbf{y}}_j > 0 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

  The weight is 1 if any common defect pattern exists in the two wafer maps and 0 otherwise. **SCL-BW** is identical to the proposed method in a multi-class classification setting.

4.4. Results and discussion

Table 2 and Table 3 compare the classification performance of the baseline and proposed methods on the WM-811K and MixedWM38 datasets, respectively, with varying training set sizes. The row-wise best results are highlighted in bold. Each best result is presented with an asterisk (*) if the p-value obtained by the paired t-test against the second-best result was less than 0.05. Fig. 3 plots the Micro-F1 and Macro-F1 against the training set size N for the datasets. In addition, Table 4 and Table 5 compare the per-category F1 scores on the WM-811K and MixedWM38 datasets, respectively, when the training set size N was set to 5,000. Analyzing the per-category F1 scores of the baseline and proposed methods provides further insight into the classification performance across specific defect categories.

Overall, the proposed method successfully improved the classification performance on both datasets. In particular, there was a substantial performance improvement when the size of the training set was small. However, as the training set size N increased, the outperformance of the proposed method became less pronounced. Among the baseline methods, **SL-CNN**, which trained a CNN without data augmentation, consistently exhibited poor classification performance and was even inferior to **SL-MFE**. This highlights the importance of data augmentation in training a CNN for wafer map pattern classification.

Table 2
Comparison of Micro-F1 and Macro-F1 scores of the WM-811K dataset (mean±standard deviation).

Metric	Size (N)	Supervised learning				Self-Supervised contrastive learning		Supervised contrastive learning
		SL-MFE	SL-CNN	SL-CNN-DA	SL-Stacking	SSCL-SimCLR	SSCL-MoCo	SCL-Proposed
Micro-F1	100	0.6965 ± 0.0477	0.8631 ± 0.0120	0.9152 ± 0.0064	0.9133 ± 0.0121	0.9112 ± 0.0080	0.9105 ± 0.0149	**0.9170 ± 0.0055***
	200	0.7612 ± 0.0532	0.8976 ± 0.0212	0.9295 ± 0.0096	0.9270 ± 0.0112	0.9258 ± 0.0069	0.9222 ± 0.0080	**0.9317 ± 0.0063**
	500	0.9121 ± 0.0064	0.9199 ± 0.0046	0.9408 ± 0.0037	0.9422 ± 0.0077	0.9407 ± 0.0024	0.9330 ± 0.0072	**0.9441 ± 0.0031**
	1,000	0.9303 ± 0.0040	0.9290 ± 0.0036	0.9508 ± 0.0024	0.9522 ± 0.0029	0.9501 ± 0.0028	0.9494 ± 0.0050	**0.9527 ± 0.0020**
	2,000	0.9389 ± 0.0017	0.9357 ± 0.0043	0.9357 ± 0.0043	0.9603 ± 0.0007	0.9580 ± 0.0016	0.9594 ± 0.0019	**0.9620 ± 0.0016**
	5,000	0.9509 ± 0.0007	0.9479 ± 0.0017	0.9674 ± 0.0009	0.9684 ± 0.0012	0.9649 ± 0.0016	0.9679 ± 0.0013	**0.9685 ± 0.0016**
	10,000	0.9566 ± 0.0007	0.9540 ± 0.0013	0.9728 ± 0.0006	0.9729 ± 0.0011	0.9684 ± 0.0015	0.9728 ± 0.0008	**0.9732 ± 0.0009**
	20,000	0.9601 ± 0.0004	0.9601 ± 0.0016	0.9760 ± 0.0009	0.9764 ± 0.0005	0.9725 ± 0.0008	0.9760 ± 0.0005	**0.9767 ± 0.0005**
Macro-F1	100	0.3144 ± 0.0388	0.2453 ± 0.0623	0.4901 ± 0.0313	0.4546 ± 0.0675	0.4780 ± 0.0433	0.4285 ± 0.0700	**0.5022 ± 0.0268***
	200	0.2721 ± 0.0374	0.3218 ± 0.0657	0.5121 ± 0.0738	0.5024 ± 0.0742	0.4957 ± 0.0541	0.4729 ± 0.0436	**0.5204 ± 0.0515**
	500	0.3567 ± 0.0248	0.4194 ± 0.0302	0.5683 ± 0.0406	0.5870 ± 0.0551	0.5699 ± 0.0236	0.5143 ± 0.0387	**0.5909 ± 0.0400**
	1,000	0.3417 ± 0.0245	0.4779 ± 0.0503	0.6430 ± 0.0270	**0.6643 ± 0.0349**	0.6485 ± 0.0297	0.6405 ± 0.0270	**0.6643 ± 0.0173**
	2,000	0.4107 ± 0.0219	0.5089 ± 0.0503	0.7130 ± 0.0406	**0.7294 ± 0.0250**	0.7068 ± 0.0217	0.7049 ± 0.0246	0.7287 ± 0.0355
	5,000	0.6073 ± 0.0313	0.6387 ± 0.0294	0.7964 ± 0.0171	**0.8011 ± 0.0123**	0.7726 ± 0.0171	0.7915 ± 0.0214	0.7998 ± 0.0188
	10,000	0.7096 ± 0.0100	0.6889 ± 0.0106	**0.8484 ± 0.0048**	0.8419 ± 0.0126	0.8014 ± 0.0114	0.8406 ± 0.0113	0.8469 ± 0.0059
	20,000	0.7481 ± 0.0060	0.7324 ± 0.0130	0.8680 ± 0.0102	0.8695 ± 0.0092	0.8364 ± 0.0102	0.8648 ± 0.0076	**0.8720 ± 0.0070**

Table 3
Comparison of Micro-F1 and Macro-F1 scores of the MixedWM38 dataset (mean±standard deviation).

Metric	Size (N)	Supervised learning				Self-Supervised contrastive learning		Supervised contrastive learning	
		SL-MFE	SL-CNN	SL-CNN-DA	SL-Stacking	SSCL-SimCLR	SSCL-MoCo	SCL-BW	SCL-Proposed
Micro-F1	100	0.6318 ± 0.0326	0.3389 ± 0.1114	0.7772 ± 0.0306	0.7977 ± 0.0147	0.7946 ± 0.0389	0.7301 ± 0.0658	0.7997 ± 0.0378	**0.8013 ± 0.0369***
	200	0.6920 ± 0.0100	0.4676 ± 0.0531	0.8559 ± 0.0131	0.8716 ± 0.0056	0.8720 ± 0.0126	0.8282 ± 0.0110	0.8777 ± 0.0106	**0.8785 ± 0.0105***
	500	0.7312 ± 0.0174	0.6172 ± 0.0428	0.9168 ± 0.0048	0.9226 ± 0.0024	**0.9306 ± 0.0048**	0.9149 ± 0.0063	0.9288 ± 0.0059	0.9300 ± 0.0065
	1,000	0.7639 ± 0.0110	0.6801 ± 0.0359	0.9516 ± 0.0038	0.9519 ± 0.0048	0.9563 ± 0.0019	0.9466 ± 0.0052	0.9566 ± 0.0032	**0.9567 ± 0.0031**
	2,000	0.8153 ± 0.0102	0.7437 ± 0.0317	0.9666 ± 0.0039	0.9694 ± 0.0018	0.9703 ± 0.0028	0.9670 ± 0.0028	**0.9707 ± 0.0019**	0.9703 ± 0.0022
	5,000	0.8817 ± 0.0014	0.8099 ± 0.0258	0.9821 ± 0.0014	0.9830 ± 0.0007	0.9837 ± 0.0009	0.9814 ± 0.0017	0.9836 ± 0.0012	**0.9843 ± 0.0011**
	10,000	0.9127 ± 0.0012	0.8857 ± 0.0248	0.9881 ± 0.0009	0.9885 ± 0.0008	**0.9891 ± 0.0005**	0.9876 ± 0.0011	0.9886 ± 0.0004	0.9889 ± 0.0009
	20,000	0.9368 ± 0.0010	0.9415 ± 0.0099	0.9912 ± 0.0007	0.9916 ± 0.0005	**0.9922 ± 0.0006**	0.9919 ± 0.0007	**0.9922 ± 0.0004**	0.9920 ± 0.0006
Macro-F1	100	0.4600 ± 0.0495	0.2050 ± 0.0729	0.6894 ± 0.0482	0.6737 ± 0.0352	0.7108 ± 0.0516	0.5850 ± 0.0888	0.7136 ± 0.0506	**0.7152 ± 0.0500***
	200	0.5269 ± 0.0355	0.3078 ± 0.0471	0.7981 ± 0.0465	0.8220 ± 0.0630	0.8184 ± 0.0486	0.7532 ± 0.0736	0.8221 ± 0.0448	**0.8230 ± 0.0452***
	500	0.6276 ± 0.0669	0.4561 ± 0.0493	0.8759 ± 0.0424	0.8893 ± 0.0430	**0.8985 ± 0.0398***	0.8797 ± 0.0450	0.8905 ± 0.0450	0.8895 ± 0.0458
	1,000	0.6202 ± 0.3778	0.5240 ± 0.0564	0.9288 ± 0.0345	0.9276 ± 0.0412	0.9330 ± 0.0339	0.9132 ± 0.0409	0.9347 ± 0.0360	**0.9360 ± 0.0339**
	2,000	0.6677 ± 0.0158	0.6361 ± 0.0586	0.9640 ± 0.0047	0.9659 ± 0.0019	0.9643 ± 0.0045	0.9625 ± 0.0039	**0.9664 ± 0.0032**	0.9660 ± 0.0036
	5,000	0.7778 ± 0.0040	0.7725 ± 0.0307	0.9766 ± 0.0039	0.9771 ± 0.0017	0.9768 ± 0.0022	0.9755 ± 0.0038	0.9761 ± 0.0037	**0.9770 ± 0.0042**
	10,000	0.8238 ± 0.0121	0.8834 ± 0.0283	0.9804 ± 0.0026	0.9807 ± 0.0020	0.9809 ± 0.0033	0.9786 ± 0.0026	0.9810 ± 0.0036	**0.9811 ± 0.0027**
	20,000	0.9154 ± 0.0317	0.9443 ± 0.0097	0.9823 ± 0.0020	0.9835 ± 0.0030	**0.9852 ± 0.0029***	0.9834 ± 0.0027	0.9826 ± 0.0018	0.9818 ± 0.0020

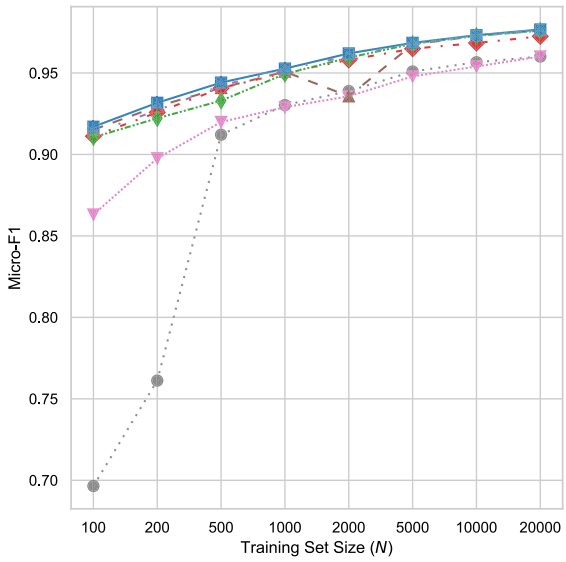
Table 4
Comparison of class-wise F1 scores of the WM-811K dataset when the training set size $N = 5,000$ (mean±standard deviation).

Category	Supervised learning				Self-Supervised contrastive learning		Supervised contrastive learning
	SL-MFE	SL-CNN	SL-CNN-DA	SL-Stacking	SSCL-SimCLR	SSCL-MoCo	SCL-Proposed
Center	0.8483 ± 0.0091	0.8322 ± 0.0132	0.9068 ± 0.0075	0.9042 ± 0.0086	0.8969 ± 0.0100	0.9032 ± 0.0109	**0.9100 ± 0.0075**
Donut	0.3702 ± 0.1027	0.4263 ± 0.1442	0.7377 ± 0.0443	0.7469 ± 0.0545	0.6977 ± 0.0264	**0.7563 ± 0.0364**	0.7517 ± 0.0394
Edge-Local	0.6674 ± 0.0105	0.5499 ± 0.0283	0.7681 ± 0.0074	0.7747 ± 0.0122	0.7497 ± 0.0159	0.7712 ± 0.0096	**0.7763 ± 0.0088**
Edge-Ring	0.9343 ± 0.0018	0.9184 ± 0.0047	0.9600 ± 0.0050	**0.9619 ± 0.0045**	0.9520 ± 0.0053	0.9617 ± 0.0036	0.9609 ± 0.0060
Local	0.4802 ± 0.0150	0.4846 ± 0.0301	0.6997 ± 0.0269	0.7002 ± 0.0158	0.6835 ± 0.0208	0.7038 ± 0.0200	**0.7183 ± 0.0156**
Random	0.6957 ± 0.0503	0.8144 ± 0.0298	0.8205 ± 0.0370	0.8401 ± 0.0225	0.8360 ± 0.0273	0.8348 ± 0.0218	**0.8417 ± 0.0199**
Scratch	0.0072 ± 0.0082	0.0074 ± 0.0090	**0.4743 ± 0.0526**	0.4726 ± 0.0531	0.3209 ± 0.0785	0.4038 ± 0.1593	0.4364 ± 0.1347
Near-Full	0.4829 ± 0.2606	0.7355 ± 0.1073	0.8127 ± 0.0943	0.8213 ± 0.0969	**0.8296 ± 0.0532**	0.8006 ± 0.0603	0.8149 ± 0.0829
None	0.9798 ± 0.0004	0.9797 ± 0.0011	0.9879 ± 0.0004	0.9879 ± 0.0007	0.9875 ± 0.0007	0.9876 ± 0.0007	**0.9881 ± 0.0006**

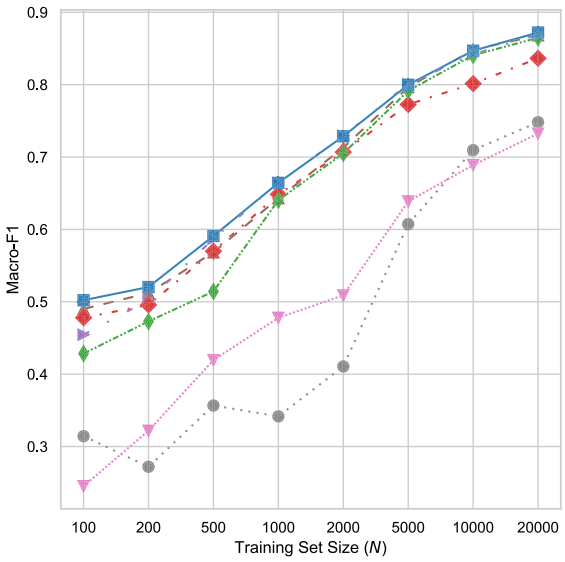
Table 5
Comparison of class-wise F1 scores of the MixedWM38 dataset when the training set size $N = 5,000$ (mean±standard deviation).

Category	Supervised learning				Self-Supervised contrastive learning		Supervised contrastive learning	
	SL-MFE	SL-CNN	SL-CNN-DA	SL-Stacking	SSCL-SimCLR	SSCL-MoCo	SCL-BW	SCL-Proposed
Center	0.8714 ± 0.0025	0.8819 ± 0.0379	0.9998 ± 0.0002	0.9998 ± 0.0001	**0.9999 ± 0.0001**	0.9998 ± 0.0001	**0.9999 ± 0.0001**	**0.9999 ± 0.0001**
Donut	0.9170 ± 0.0022	0.9762 ± 0.0090	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9999 ± 0.0001	0.9999 ± 0.0001	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
Edge-Local	0.9232 ± 0.0017	0.3961 ± 0.0466	0.9680 ± 0.0022	0.9685 ± 0.0020	0.9702 ± 0.0016	0.9663 ± 0.0028	0.9696 ± 0.0018	**0.9705 ± 0.0021**
Edge-Ring	0.9739 ± 0.0014	0.8809 ± 0.0068	0.9824 ± 0.0017	0.9832 ± 0.0021	0.9840 ± 0.0010	0.9820 ± 0.0011	0.9840 ± 0.0015	**0.9843 ± 0.0015**
Local	0.9339 ± 0.0007	0.8521 ± 0.0306	0.9885 ± 0.0011	0.9885 ± 0.0007	0.9890 ± 0.0007	0.9884 ± 0.0012	**0.9893 ± 0.0006**	0.9892 ± 0.0008
Random	0.8373 ± 0.0292	0.8685 ± 0.0350	**0.9861 ± 0.0036**	0.9851 ± 0.0040	0.9831 ± 0.0041	0.9838 ± 0.0055	0.9840 ± 0.0052	0.9837 ± 0.0061
Scratch	0.7457 ± 0.0076	0.8005 ± 0.0259	0.9615 ± 0.0051	0.9648 ± 0.0030	0.9662 ± 0.0030	0.9597 ± 0.0065	0.9656 ± 0.0046	**0.9685 ± 0.0040**
Near-Full	0.0000 ± 0.0000	0.6624 ± 0.1729	**0.9149 ± 0.0300**	0.9145 ± 0.0120	0.9117 ± 0.0189	0.9098 ± 0.0259	0.9074 ± 0.0261	0.9115 ± 0.0300
None	0.7977 ± 0.0085	0.6339 ± 0.0651	0.9882 ± 0.0052	**0.9896 ± 0.0028**	0.9868 ± 0.0054	0.9895 ± 0.0047	0.9855 ± 0.0046	0.9853 ± 0.0034

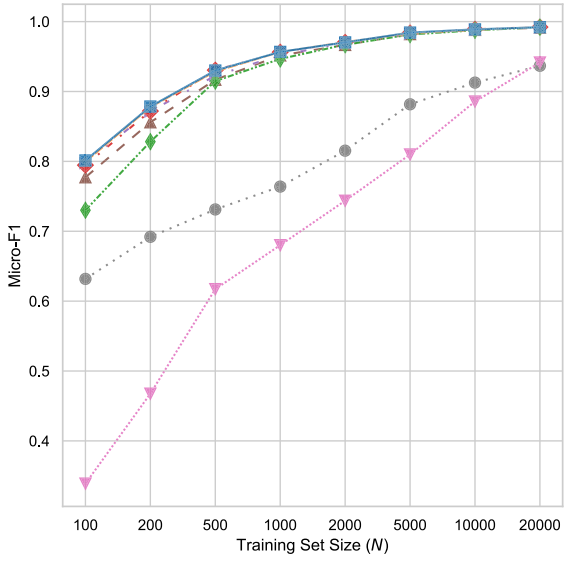
For the multi-class classification task on the WM-811K dataset, the proposed method outperformed all the baseline methods regardless of the training set size N in terms of Micro-F1. With respect to Macro-F1, the proposed method outperformed the baselines when the training set size N was smaller than 2000 and showed performance comparable to the best method when the size N was larger. For class-wise F1 scores when $N = 5,000$, the proposed method performed best on most defect categories. However, it performed worse than the best baseline for *Edge-Ring*, *Near-Full*, and *Scratch*, which are relatively minority categories in the dataset.



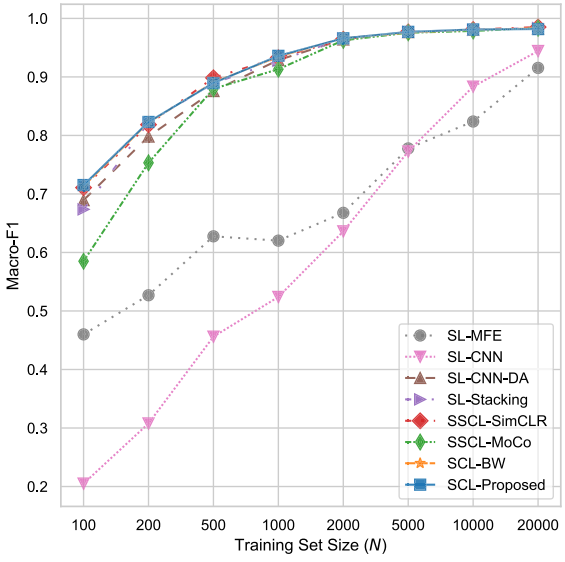
(a) Micro-F1 on WM-811K dataset



(b) Macro-F1 on WM-811K dataset



(c) Micro-F1 on MixedWM38 dataset



(d) Macro-F1 on MixedWM38 dataset

Fig. 3. Comparison of Micro-F1 and Macro-F1 scores for the baselines and proposed methods across different training set sizes N: (a) Micro-F1 on WM-811K dataset, (b) Macro-F1 on WM-811K dataset, (c) Micro-F1 on MixedWM38 dataset and (d) Macro-F1 on MixedWM38 dataset.

For the multi-label classification task on the MixedWM38 dataset, the proposed method significantly outperformed the baselines in terms of both Micro- and Macro-F1 when the training set was small. In particular, the superiority of the proposed method over **SCL-BW** indicates that the label similarity weighting scheme contributed to improving the classification performance in the presence of wafer maps containing mixed-type defect patterns. As the training set size increased, the performance gap against the best baseline became negligible. For class-wise F1 scores when $N = 5,000$, the proposed method exhibited performance superior to or at least comparable to the baseline methods for all defect categories.

5. Conclusion

In this paper, we presented a supervised contrastive learning method for automated wafer map pattern classification. We achieved the objective by training a CNN model to accurately predict the defect categories for rotation variants of wafer maps and to closely align the representation vectors for rotation variants of wafer maps having similar labels. We introduced a label similarity weighting scheme that can be generalized to both multi-class and multi-label settings of supervised contrastive learning. By conducting experiments on two wafer map datasets, WM-811K and MixedWM38, we demonstrated that the proposed method improved the classification performance on both multi-class and multi-label classification tasks, particularly when the training dataset was insufficient.

To further enhance the performance of wafer map pattern classification, there are two important considerations to be taken into account. The first is to address the challenge of handling wafer maps of various shapes. For example, the shapes of the wafer maps in the WM-811K dataset range from $(6, 21)$ and $(300, 202)$. In this study, we resized all wafer maps to the same shape before feeding them into the CNN. However, extreme resizing of a wafer map may not preserve the main defect patterns on it, thus negatively affecting the classification performance. As a future research direction, we plan to investigate methods to build

a classification model that can accurately classify arbitrarily-shaped wafer maps without resizing them. The second is to prioritize the classification performance specifically for the minority defect categories. Although the proposed method has shown improvements in overall classification performance, certain minority defect categories, such as *Scratch* and *Near-Full*, still suffer from low accuracy. In real-world applications, misclassifying wafer maps belonging to these minority defect categories can incur high costs. As a future direction for enhancing the performance for these specific defect categories, we plan to adopt class-adaptive and learnable data augmentation strategies.

CRediT authorship contribution statement

Youngjae Bae: Conceptualization, Methodology, Software, Writing – original draft. **Seokho Kang:** Conceptualization, Methodology, Investigation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The datasets used in this study are publicly available.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT; Ministry of Science and ICT) (No. RS-2023-00207903).

References

Chen, X., Fan, H., Girshick, R.B., He, K., 2020a. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020b. A simple framework for contrastive learning of visual representations. In: Proceedings of International Conference on Machine Learning. pp. 1597–1607.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E., 2020c. Big self-supervised models are strong semi-supervised learners. In: Advances in Neural Information Processing Systems. pp. 22243–22255.

Chiu, M.C., Chen, T.M., 2021. Applying data augmentation and mask R-CNN-based instance segmentation method for mixed-type wafer maps defect patterns classification. IEEE Trans. Semicond. Manuf. 34 (4), 455–463.

Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., Valko, M., 2020. Bootstrap your own latent - a new approach to self-supervised learning. In: Advances in Neural Information Processing Systems. pp. 21271–21284.

Hansen, M.H., Nair, V.N., Friedman, D.J., 1997. Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects. Technometrics 39 (3), 241–253.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738.

Hu, H., He, C., Li, P., 2021. Semi-supervised wafer map pattern recognition using domain-specific data augmentation and contrastive learning. In: Proceedings of IEEE International Test Conference. pp. 113–122.

Huang, S.H., Pan, Y.C., 2015. Automated visual inspection in the semiconductor industry: A survey. Comput. Ind. 66, 1–10.

Ishida, T., Nitta, I., Fukuda, D., Kanazawa, Y., 2019. Deep learning-based wafer-map failure pattern recognition framework. In: Proceedings of International Symposium on Quality Electronic Design. pp. 291–297.

Jaiswal, A., Li, T., Zander, C., Han, Y., Rousseau, J.F., Peng, Y., Ding, Y., 2021. SCALP-supervised contrastive learning for cardiopulmonary disease classification and localization in chest X-rays using patient metadata. In: Proceedings of IEEE International Conference on Data Mining. pp. 1132–1137.

Kahng, H., Kim, S.B., 2021. Self-supervised representation learning for wafer bin map defect pattern classification. IEEE Trans. Semicond. Manuf. 34 (1), 74–86.

Kang, S., 2020. Rotation-invariant wafer map pattern classification with convolutional neural networks. IEEE Access 8, 170650–170658.

Kang, H., Kang, S., 2021. A stacking ensemble classifier with handcrafted and convolutional features for wafer map pattern classification. Comput. Ind. 129, 103450.

Kang, H., Kang, S., 2023. Semi-supervised rotation-invariant representation learning for wafer map pattern analysis. Eng. Appl. Artif. Intell. 120, 105864.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised contrastive learning. In: Advances in Neural Information Processing Systems. pp. 18661–18673.

Kingma, D.P., Mohamed, S., Jimenez Rezende, D., Welling, M., 2014. Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems. pp. 3581–3589.

Kong, Y., Ni, D., 2019. Recognition and location of mixed-type patterns in wafer bin maps. In: Proceedings of IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering. pp. 4–8.

Kong, Y., Ni, D., 2020. A semi-supervised and incremental modeling framework for wafer map classification. IEEE Trans. Semicond. Manuf. 33 (1), 62–71.

Kyeong, K., Kim, H., 2018. Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. IEEE Trans. Semicond. Manuf. 31 (3), 395–402.

Le-Khac, P.H., Healy, G., Smeaton, A.F., 2020. Contrastive representation learning: A framework and review. IEEE Access 8, 193907–193934.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lee, H., Kim, H., 2020. Semi-supervised multi-label learning for classification of wafer bin maps with mixed-type defect patterns. IEEE Trans. Semicond. Manuf. 33 (4), 653–662.

Li, T.S., Huang, C.L., 2009. Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing. Expert Syst. Appl. 36 (1), 374–385.

Li, S., Xia, X., Ge, S., Liu, T., 2022. Selective-supervised contrastive learning with noisy labels. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 316–325.

Li, J., Zhou, P., Xiong, C., Hoi, S., 2021. Prototypical contrastive learning of unsupervised representations. In: Proceedings of International Conference on Learning Representations.

Małkiński, M., Mańdziuk, J., 2022. Multi-label contrastive learning for abstract visual reasoning. IEEE Trans. Neural Netw. Learn. Syst.

Misra, I., Maaten, L.v.d., 2020. Self-supervised learning of pretext-invariant representations. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6707–6717.

Nakazawa, T., Kulkarni, D.V., 2018. Wafer map defect pattern classification and image retrieval using convolutional neural network. IEEE Trans. Semicond. Manuf. 31 (2), 309–314.

Piao, M., Jin, C.H., Lee, J.Y., Byun, J.Y., 2018. Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features. IEEE Trans. Semicond. Manuf. 31 (2), 250–257.

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T., 2015. Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems. pp. 3546–3554.

Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Gupta, B.B., Chen, X., Wang, X., 2021. A survey of deep active learning. ACM Comput. Surv. 54 (9), 180.

Saqlain, M., Jargalsaikhan, B., Lee, J.Y., 2019. A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. IEEE Trans. Semicond. Manuf. 32 (2), 171–182.

Sermanet, P., Lynch, C., Hsu, J., Levine, S., 2017. Time-contrastive networks: Self-supervised learning from multi-view observation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 486–487.

Shawon, A., Faruk, M.O., Habib, M.B., Khan, A.M., 2019. Silicon wafer map defect classification using deep convolutional neural network with data augmentation. In: Proceedings of IEEE International Conference on Computer and Communications. pp. 1995–1999.

Shim, J., Kang, S., 2023. Learning from single-defect wafer maps to classify mixed-defect wafer maps. Expert Syst. Appl. 233, 120923.

Shim, J., Kang, S., Cho, S., 2020. Active learning of convolutional neural network for cost-effective wafer map pattern classification. IEEE Trans. Semicond. Manuf. 33 (2), 258–266.

Shim, J., Kang, S., Cho, S., 2021. Active cluster annotation for wafer map pattern classification in semiconductor manufacturing. Expert Syst. Appl. 183, 115429.

Shin, W., Kahng, H., Kim, S.B., 2022. Mixup-based classification of mixed-type defect patterns in wafer bin maps. Comput. Ind. Eng. 167, 107996.

Shon, H.S., Batbaatar, E., Cho, W.S., Choi, S.G., 2021. Unsupervised pre-training of imbalanced data for identification of wafer map defect patterns. IEEE Access 9, 52352–52363.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6, 60.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Van Engelen, J.E., Hoos, H.H., 2020. A survey on semi-supervised learning. Mach. Learn. 109 (2), 373–440.

Wang, R., Chen, N., 2020. Defect pattern recognition on wafers using convolutional neural networks. Qual. Reliab. Eng. Int. 36 (4), 1245–1257.

Wang, C.H., Kuo, W., Bensmail, H., 2006. Detection and classification of defect patterns on semiconductor wafers. IIE Trans. 38 (12), 1059–1068.

Wang, J., Xu, C., Yang, Z., Zhang, J., Li, X., 2020. Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition. IEEE Trans. Semicond. Manuf. 33 (4), 587–596.

Wang, J., Yang, Z., Zhang, J., Zhang, Q., Chien, W.-T.K., 2019. AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition. IEEE Trans. Semicond. Manuf. 32 (3), 310–319.

Wen, K., Xia, J., Huang, Y., Li, L., Xu, J., Shao, J., 2021. COOKIE: Contrastive cross-modal knowledge sharing pre-training for vision-language representation. In: Proceedings of IEEE/CVF International Conference on Computer Vision. pp. 2208–2217.

Wu, M.J., Jang, J.S.R., Chen, J.L., 2015. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. IEEE Trans. Semicond. Manuf. 28 (1), 1–12.

Yu, J., Shen, Z., Wang, S., 2021. Wafer map defect recognition based on deep transfer learning-based densely connected convolutional network and deep forest. Eng. Appl. Artif. Intell. 105, 104387.

Yu, N., Xu, Q., Wang, H., 2019. Wafer defect pattern recognition and analysis based on convolutional neural network. IEEE Trans. Semicond. Manuf. 32 (4), 566–573.

Zhang, H., Koh, J.Y., Baldridge, J., Lee, H., Yang, Y., 2021. Cross-modal contrastive learning for text-to-image generation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 833–842.

Zheng, M., Wang, F., You, S., Qian, C., Zhang, C., Wang, X., Xu, C., 2021. Weakly supervised contrastive learning. In: Proceedings of IEEE/CVF International Conference on Computer Vision. pp. 10042–10051.

Zolfaghari, M., Zhu, Y., Gehler, P., Brox, T., 2021. CrossCLR: Cross-modal contrastive learning for multi-modal video representations. In: Proceedings of IEEE/CVF International Conference on Computer Vision. pp. 1450–1459.