The International Journal of Advanced Manufacturing Technology (2025) 138:2139–2156
https://doi.org/10.1007/s00170-025-15685-7

ORIGINAL ARTICLE

Recognition and ranking using similarity on defective wafer bin maps

YoungWook Kwon[1] · SuMin Oh[1] · HyunJin Kim[1]

Received: 2 November 2024 / Accepted: 5 May 2025 / Published online: 10 May 2025

Abstract
As semiconductor manufacturing technology has been rapidly advanced, conventional approaches cannot classify new wafer defect patterns without retraining. To overcome this, we propose an image matching-based pattern search method to analyze the similarity between the wafer defect patterns using new pre-processing and similarity metrics. The proposed search method finds the correlation of wafer defect patterns to determine the similarity value between wafer bin maps (WBMs). The pre-processing performs the denoising to reduce the effects of non-significant defect patterns. Besides, we propose two metrics for performing robust pattern searches based on the shape, location, and area of defect patterns. Experimental results show that the proposed method is effective on industrial-driven datasets WM-811K and MixedWM38, having significant benefits of the pre-processing and similarity metrics in the search for various defect patterns.

Keywords Deep learning · Image matching · Semiconductor manufacturing · Wafer defect patterns · Mixed pattern detection · Single pattern detection

1 Introduction

After electrical testing in semiconductor manufacturing, defects are visualized on wafer bin maps (WBMs). By analyzing recurring defect patterns on WBMs, engineers can diagnose problematic processes [1]. Furthermore, as semiconductor manufacturing processes become more complex, the importance of accurate defect pattern analysis continues to grow [2]. Conventional methods for defect pattern analysis are categorized into two main approaches: Supervised and unsupervised learning. Supervised learning-based methods achieve high accuracy and have been widely applied in defect detection tasks. Recent studies have explored deep learning-based classification methods [3–8], achieving state-of-the-art performance. However, these methods require large amounts of labeled data, which are time-consuming and labor-intensive. Furthermore, semiconductor defect datasets often suffer from data imbalance, where certain defect patterns are significantly outnumbered by other patterns [9, 10]. This leads to model bias and reduced detection accuracy. Most of all, technological advances increase the diversity of defect patterns, further complicating the labeling process and requiring frequent model retraining to maintain performance.

On the other hand, unsupervised learning-based methods have been proposed to reduce reliance on labeled data [11–14]. These methodologies have demonstrated potential in identifying novel defect patterns and addressing the emergence of diverse defects. However, they rely on statistical distributions which should be predefined using domain knowledge from expertise in the field [15]. In addition, the limited interpretability of unsupervised learning makes it difficult for engineers to validate the detected patterns and integrate them into manufacturing processes.

Recently, similarity-based methods have been widely studied [16–19] to overcome these limitations of traditional methods. By comparing defect patterns using similarity metrics such as Euclidean distance, engineers can search and analyze similar defects from the dataset. In other words, identifying and searching for similar defect patterns enables rapid analysis of recurring defects without relying on labeled data

YoungWook Kwon and SuMin Oh contributed equally to this work.

✉ HyunJin Kim
hyunjin2.kim@gmail.com

YoungWook Kwon
kyw96@naver.com

SuMin Oh
osm040836@gmail.com

[1] Department of Electronics and Electrical Engineering, Dankook University, 152, Jukjeon-ro, Suji-gu, Yongin-si 16890, Gyeonggi-do, Republic of Korea

and domain knowledge. Recent studies have adopted various feature extraction techniques and similarity metrics to improve search accuracy. However, despite their attempts, existing similarity-based methods still face key challenges. They fail to account for variations in defect location and size, which can lead to inaccurate search. Furthermore, previous studies have primarily focused on global defect patterns, which may not effectively capture localized defect patterns in real-world semiconductor manufacturing scenarios.

To address the above issues, this work proposes a novel pattern search method that accounts for variations in the area and location of the defect pattern in the WBM while ensuring robust and accurate pattern searching. The proposed method has three main processes as follows: first, we propose to pre-process the WBMs to refine defect patterns. This step removes non-significant defects, reducing noise and enhancing the robustness of subsequent analysis. By eliminating irrelevant defects, the method improves the accuracy of defect search and similarity calculations. Second, image matching-based defect pattern analysis is performed. After extracting keypoints and descriptors from refined defect regions of WBMs, the matching score (Mscore) is computed between defect patterns. Notably, this process does not require labeled WBMs, allowing the method to generalize to new defect patterns without retraining a neural network model. Third, similarity calculation using the proposed *Confidence* and *match of defects (MoD)* scores is applied. The Confidence score evaluates the reliability of matched defect patterns by assessing the consistency of their feature representations across WBMs. A higher Confidence score indicates a more reliable match, helping to filter out incorrect or ambiguous pattern searching. Meanwhile, the MoD score addresses a key limitation of conventional similarity-based methods, which often overlook variations in defect location and area. By applying a penalty based on the partial distribution and size of matched defect patterns, the MoD score ensures that similarity measurements remain consistent across different WBMs. The main contributions of this work are as follows:

- **Pre-processing for refined defect patterns:** A WBM pre-processing is introduced to remove non-significant defects, reducing noise and improving searching accuracy.
- **Similarity calculation with Confidence and MoD scores:** We propose the Confidence score to assess the reliability of matched patterns and the MoD score to account for defect shape, location, and size, ensuring robust pattern search.
- **Enhanced pattern searching performance:** Experimental evaluations on two industry-driven datasets demonstrate that the proposed method substantially outperforms existing similarity-based techniques in search accuracy.

The rest of the paper is organized as follows. Section 2 provides background information on existing methods for wafer defect pattern analysis. Section 3 introduces the proposed pattern search method, including the preprocessing, keypoint-based image matching, and similarity metrics. Section 4 evaluates the performance of our method using industry-driven datasets, comparing it with existing similarity-based methods. The conclusions are provided in Section 5.

2 Background

2.1 Traditional similarity-based methods for WBM defect analysis

Similarity-based methods use geometrical distance to measure the similarity between a source WBM and target WBMs. By comparing problematic processes of WBMs with higher similarity, root cause analysis can be facilitated. Thus, while classification-based methods existing methods such as supervised- and unsupervised learning-based methods struggle to analyze defect patterns that have the same labels or overlapping defects, similarity-based methods offer greater flexibility.

For similarity measures, template matching has been widely adopted. [20] first adopted template matching for wafer defect analysis to rank WBMs based on similarity. In addition, [16] employed template matching to segment defect patterns in WBMs. More recently, [17] adopted a different type of template matching called best-buddies similarity, which enhances matching accuracy by identifying mutually nearest neighbors between two distributions. Beyond template matching, alternative similarity measures have been explored to improve defect analysis. For instance, [18] introduced a WBM similarity measurement framework that integrates a mountain clustering algorithm and weighted modified Hausdorff distance. Besides, [21] adopted deep learning-based similarity models to learn feature representations of WBMs, allowing for more adaptive and scalable similarity evaluations. These approaches aim to overcome the limitations of purely template matching methods by incorporating spatial information and non-linear feature mappings.

Despite these advances, template matching may suffer from inherent limitations, as it struggles with complex transformations, non-rigid deformations, and ambiguous variations of images [22]. Furthermore, deep learning-based similarity models may face the same challenges as supervised

learning-based methods since they require labeled WBM datasets.

To address these limitations, image matching can serve as a more effective approach. To the best of our knowledge, image matching has not yet been applied to wafer defect analysis. However, its ability to extract distinctive features and establish robust correspondences under geometric variations makes it a promising alternative to template matching. The following subsection introduces key image matching methodologies and demonstrates their potential applicability to WBM similarity analysis.

2.2 Image matching algorithms

Image matching algorithms have been developed for various computer vision tasks (e.g., object recognition and image stitching). Generally, these algorithms identify key features called *keypoints* in images and establish correspondences between them to enable accurate matching and alignment. Specifically, they consist of three key processes: feature detection, description, and matching. In feature detection, classical methods such as scale invariant feature transform (SIFT) [23] and oriented FAST and rotated BRIEF (ORB) [24] detect keypoints by identifying regions with sharp changes in brightness or contrast, relying on handcrafted operations such as difference of Gaussian filtering. In feature description, these methods then generate descriptors based on histograms of gradient orientations, ensuring invariance to rotation and scale. In feature matching, techniques such as brute-force matching or approximate nearest neighbor are commonly used, comparing feature vectors using similarity metrics such as Euclidean distance.

Recent studies have leveraged deep learning-based approaches to enhance the robustness of image matching. Instead of relying on handcrafted operations, they utilize convolutional neural networks (CNNs) to learn keypoint representations from real or synthetic keypoint datasets. For example, [25–27] adopted supervised learning to train detectors and descriptors, while [28, 29] employed self-supervised learning to improve the consistency of keypoint detection and description across varying conditions. In feature matching, graph neural networks (GNNs) and attention mechanisms, which model relationships between keypoints beyond simple pairwise comparisons, have been widely adopted. SuperGlue [30] employed GNNs with self- and cross-attention to refine matching, while local feature matching with transformers [31] adopted transformer architecture [32] to incorporate global context, further improving accuracy in feature matching. More recently, lightweight models such as local feature matching at light speed [33] have been proposed to optimize attention-based feature matching for real-time applications.

As we noted in Section 2.1, template matching struggles with complex transformations and ambiguous shapes of defect patterns, making it less effective for WBM analysis. Similarly, traditional image matching methods can face the same challenges as they rely on handcrafted features and local similarity metrics, leading to inconsistent keypoint detection and false correspondences. Thus, we adopted deep learning-based image matching methods that leverage keypoint detection, descriptor learning, and attention-based feature matching. Depending on hardware resources and search purposes, various image matching algorithms can be utilized for WBM analysis. While lightweight methods may be suitable for real-time searches, transformer-based methods can capture both local and global defect patterns. However, for a strong and stable baseline, we selected SuperPoint and SuperGlue for wafer defect analysis, as they have demonstrated robust performance across various geometric transformations and noise conditions. The following section provides detailed information on the proposed pattern search method and discusses how different image matching algorithms can be integrated into WBM defect analysis.

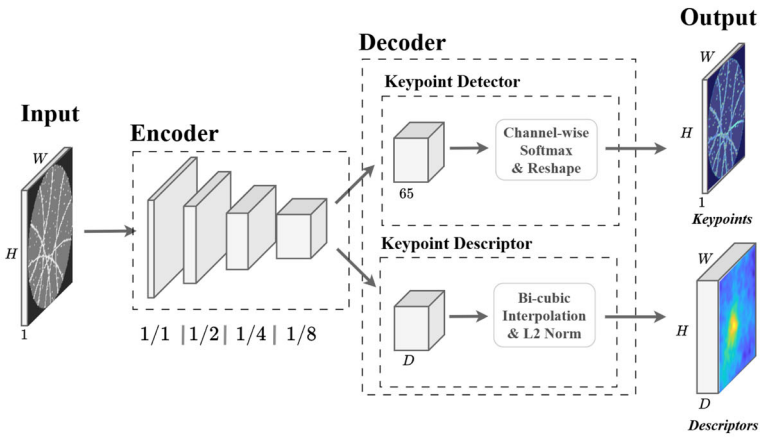
3 Proposed pattern search method

In this section, we propose the pattern search method designed to improve WBM defect analysis. Our approach integrates keypoint-based image matching with a pre-processing step to refine defect patterns. The method consists of three main components: (1) pre-processing, which eliminates non-significant defect patterns to enhance the reliability of similarity calculations; (2) image matching, which extracts keypoints and descriptors from defect patterns and establishes correspondences between source and target WBMs; and (3) similarity measurement, which adjusts Mscores using Confidence and MoD scores to improve robustness against variations in defect shape, location, and area. The following subsections provide details on the key components of the proposed method.

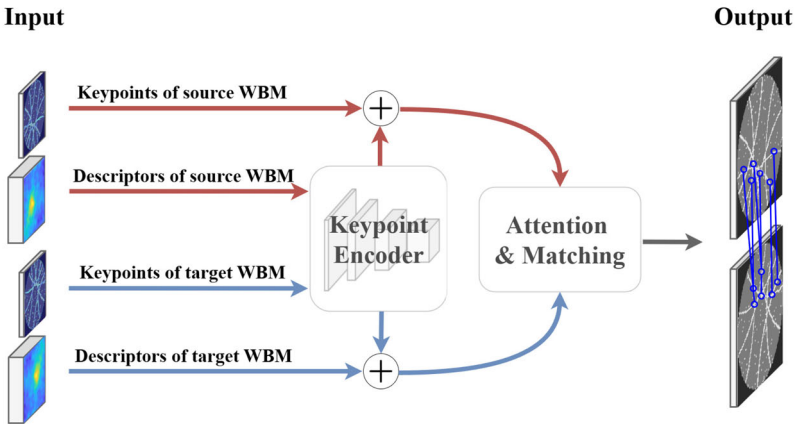
3.1 Image matching for WBM defect analysis

The image matching for the proposed pattern search algorithm in WBMs has four main sub-functions: (1) detection, (2) description, (3) matching, and (4) scoring.

The detection function detects salient pixels called keypoints from the defect patterns in each WBM. Then, the description function extracts descriptors around keypoints to enhance robustness against rotational and scale variations of the defect patterns. Figure 1a shows an example of the keypoint detection and description processes in WBM. On the other hand, the matching function matches keypoints from different defect patterns based on their descriptors. Figure 1b shows an example of the keypoint matching processes in WBM. Finally, the scoring function calculates the match-



(a) Keypoint detection and description in WBM



(b) Keypoint matching between WBMs

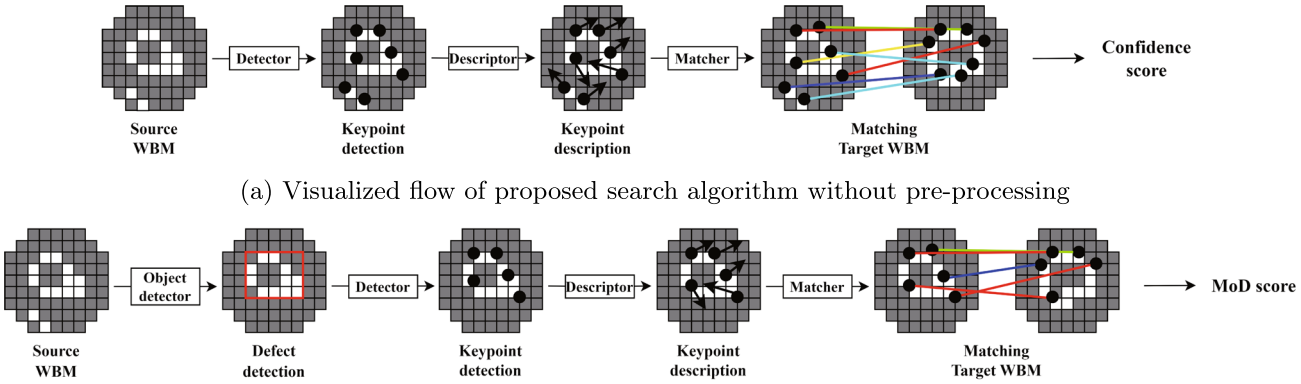
Fig. 1 Keypoint detection, description, and matching of the proposed pattern search method

ing score, which refers to similarity scores, using distance metrics between descriptors, such as dot products and the Euclidean distance. In the scoring function, the proposed two metrics can be applied by incorporating the average Mscore between matched descriptors and the geometric characteristics of defect patterns. The final similarity value is obtained by adjusting the average Mscore using the calculated Confidence or MoD score, ensuring a more accurate comparison between defect patterns in the source WBM and target WBMs.

3.2 Implementation details of the proposed method

The pseudo-code of the proposed search method is shown in Algorithm 1. Let us assume that there are a source WBM denoted as $\mathbf{s}$ and target WBMs denoted as $\mathbf{t}_{k\in 1,\ldots,K}$. Several terms and functions are defined to clarify the pseudo-code:

- $\vec{kp}_{\mathbf{x}}$: Keypoints detected from the defect patterns in WBM $\mathbf{x} \in \{\mathbf{s}, \mathbf{t}_{k\in 1,\ldots,K}\}$.
- $\vec{des}_{\mathbf{x}}$: Descriptors from the keypoints from the defect patterns in WBM $\mathbf{x} \in \{\mathbf{s}, \mathbf{t}_{k\in 1,\ldots,K}\}$.
- Detection $(\mathbf{x}, D_{max}(\vec{kp}_{\mathbf{x}}))$: A function of detecting keypoints $\vec{kp}_{\mathbf{x}} \in \mathbb{R}^{D_{max}(\vec{kp}_{\mathbf{x}})}$ from WBM image $\mathbf{x} \in \{\mathbf{s}, \mathbf{t}_{k\in 1,\ldots,K}\}$, where $D_{max}(\vec{kp}_{\mathbf{x}})$ is the maximum dimension of keypoints $\vec{kp}_{\mathbf{x}}$.
- Description $(\mathbf{x}, \vec{kp}_{\mathbf{x}}, D(\vec{des}_{\mathbf{x}}))$: A function of generating descriptors $\vec{des}_{\mathbf{x}} \in \mathbb{R}^{D(\vec{des}_{\mathbf{x}})}$ from keypoints $\vec{kp}_{\mathbf{x}}$ of defect patterns in WBM $\mathbf{x} \in \{\mathbf{s}, \mathbf{t}_{k\in 1,\ldots,K}\}$, where $D(\vec{des}_{\mathbf{x}})$ the dimension of the descriptors $\vec{des}_{\mathbf{x}}$.
- Pre-processing $(\mathbf{x})$: A function of performing the pre-processing for WBM $\mathbf{x} \in \{\mathbf{s}, \mathbf{t}_{k\in 1,\ldots,K}\}$, as shown in Fig. 2b.
- Matching $(\vec{kp}_{\mathbf{s}}, \vec{des}_{\mathbf{s}}, \vec{kp}_{\mathbf{t}_k}, \vec{des}_{\mathbf{t}_k})$: A function of performing image matching between pairs of $(\vec{kp}_{\mathbf{s}}, \vec{des}_{\mathbf{s}})$ from the defect patterns in WBM $\mathbf{s}$ and $(\vec{kp}_{\mathbf{t}_k}, \vec{des}_{\mathbf{t}_k})$ from the



(a) Visualized flow of proposed search algorithm without pre-processing

(b) Visualized flow of proposed search algorithm with pre-processing

Fig. 2 Visualization of the proposed search algorithm

defect patterns WBM $\mathbf{t}_k$. This function can produce the vector of Mscores $\vec{m}_{des_j \leftrightarrow des_i}$.

- Matched descriptors: In the source WBM $\mathbf{s}$, if any descriptor $des_i \in \vec{des}_{\mathbf{s}}$ is matched with the descriptor of a target WBM denoted as $des_j \in \vec{des}_{\mathbf{t}_k}$ for a target WBM $\mathbf{t}_k$, the matched descriptors are denoted as $des_j \leftrightarrow des_i$ or $des_i \leftrightarrow des_j$. Terms i and j are for the indices of defect patterns from the source WBM and target WBM, respectively.
- Confidence ($\mathbf{s}$, $\vec{m}_{des_j \leftrightarrow des_i}$, $\mathbf{t}_k$): A function of calculating the Confidence score between WBMs $\mathbf{s}$ and $\mathbf{t}_k$ based on the vector of Mscores $\vec{m}_{des_j \leftrightarrow des_i}$ from matching($\vec{kp}_{\mathbf{s}}$, $\vec{des}_{\mathbf{s}}$, $\vec{kp}_{\mathbf{t}_k}$, $\vec{des}_{\mathbf{t}_k}$).
- MoD ($\mathbf{s}$, $\vec{m}_{des_j \leftrightarrow des_i}$, $\mathbf{t}_k$): A function of calculating the MoD score between two WBM images $\mathbf{s}$ and $\mathbf{t}_k$, based on the vector of Mscores $\vec{m}_{des_j \leftrightarrow des_i}$ from matching($\vec{kp}_{\mathbf{s}}$, $\vec{des}_{\mathbf{s}}$, $\vec{kp}_{\mathbf{t}_k}$, $\vec{des}_{\mathbf{t}_k}$).

The description of Algorithm 1 is as follows: source WBM $\mathbf{s}$ and target WBMs $\mathbf{t}_{k\in 1,\ldots K}$ are the inputs of Algorithm 1. The maximum dimension of keypoints and the dimension of descriptors can be given as the inputs, which are not shown for clarity. Term $pre \in \{T, F\}$ indicates whether the pre-processing is performed (T) or not (F). Term $score \in$ {Confidence, MoD} is the name of the function to calculate the similarity between defect patterns of $\mathbf{s}$ and $\mathbf{t}_{k\in 1,\ldots K}$. After extracting keypoints and descriptors from $\mathbf{s}$ and $\mathbf{t}_k$ (Lines 5–12), the vector of Mscores $\vec{m}_{des_j \leftrightarrow des_i}$ is calculated from the matching function (Line 13). Then, the similarity of the two defect patterns from $\mathbf{s}$ and $\mathbf{t}_k$ denoted as sim is calculated, where either Confidence or MoD score is obtained depending on $score$ (Line 14). In other words, the calculated Confidence or MoD score can be the similarity denoted as sim. Finally, the similarity between defect patterns of $\mathbf{s}$ and $\mathbf{t}_{k\in 1,\ldots K}$ is appended into the array of similarities denoted as $\mathbf{sim}_{(\mathbf{s},\mathbf{t}_{1,\ldots k-1})}$ (Line 15). We note that image matching with extracted keypoints and their descriptors can be performed using SuperGlue in the matching function (Line 13). However, any keypoint-based image matching algorithm instead of SuperGlue can calculate the above Mscores (Line 13). It is noted that the Confidence or MoD score in Line 14 is needed to normalize and adjust the average Mscore depending on the location and area of defect patterns. The sub-processes of the proposed algorithm are explained in the following subsections.

Algorithm 1 Pseudo code of the proposed pattern search algorithm.

Input: Source WBM $\mathbf{s}$, Target WBMs $\mathbf{t}_{k\in 1,\ldots K}$, $pre \in \{T, F\}$, $score \in$ {Confidence, MoD}
Output: Array $\mathbf{sim}_{(\mathbf{s},k\in 1,\ldots K)}$

1: $\mathbf{sim}_{(\mathbf{s},\mathbf{t}_{1,\ldots k})} \leftarrow \emptyset$
2: **if** $pre = T$ **then**
3: $\quad \mathbf{s} \leftarrow$ pre-processing($\mathbf{s}$)
4: **end if**
5: $\vec{kp}_{\mathbf{s}} \leftarrow$ detection($\mathbf{s}$, $D_{max}(\vec{kp}_{\mathbf{s}})$)
6: $\vec{des}_{\mathbf{s}} \leftarrow$ description($\mathbf{s}$, $\vec{kp}_{\mathbf{s}}$, $D(\vec{des}_{\mathbf{s}})$)
7: **for** $k \in \{1, \ldots, K\}$ **do**
8: $\quad$ **if** $pre = T$ **then**
9: $\quad\quad \mathbf{t}_k \leftarrow$ pre-processing($\mathbf{t}_k$)
10: $\quad$ **end if**
11: $\quad \vec{kp}_{\mathbf{t}_k} \leftarrow$ detection($\mathbf{t}_k$, $D_{max}(\vec{kp}_{\mathbf{t}_k})$)
12: $\quad \vec{des}_{\mathbf{t}_k} \leftarrow$ description($\mathbf{t}_k$, $\vec{kp}_{\mathbf{t}_k}$, $D(\vec{des}_{\mathbf{t}_k})$)
13: $\quad \vec{m}_{des_j \leftrightarrow des_i} \leftarrow$ matching($\vec{kp}_{\mathbf{s}}$, $\vec{des}_{\mathbf{s}}$, $\vec{kp}_{\mathbf{t}_k}$, $\vec{des}_{\mathbf{t}_k}$))
14: $\quad sim \leftarrow score(\mathbf{s}, \vec{m}_{des_j \leftrightarrow des_i}, \mathbf{t}_k)$
15: $\quad \mathbf{sim}_{(\mathbf{s},\mathbf{t}_{1,\ldots k})} \leftarrow$ Append($\mathbf{sim}_{(\mathbf{s},\mathbf{t}_{1,\ldots k-1})}$, sim)
16: **end for**

3.3 Image pre-processing for pattern search

Pre-processing techniques, such as denoising, have been widely used in image analysis [34]. However, their applica-

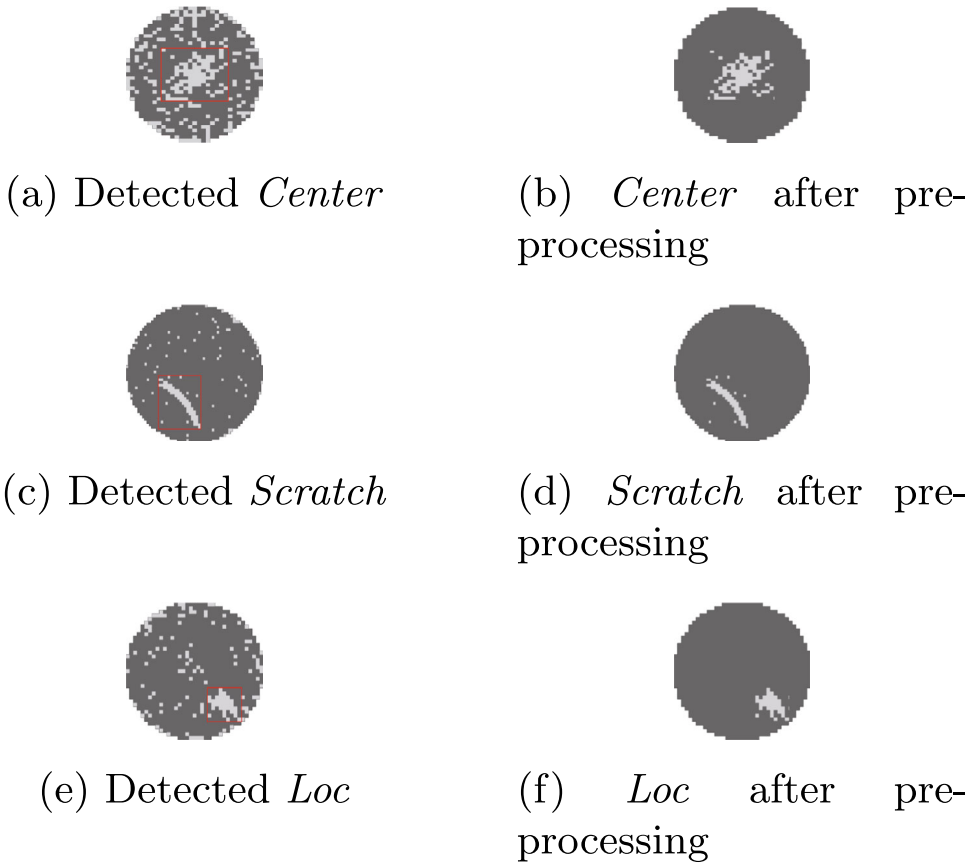
tion to defect pattern analysis requires careful consideration due to the unique characteristics of semiconductor defect patterns, including their spatial distribution, variation in size, and noise from non-significant defects. In [20, 35–37], the location and area of defect patterns affect the matching results of defect patterns. In other words, the repetitive malfunction of the manufacturing process is significantly related to the defect patterns having close locations or similar areas. However, general keypoint-based image matching algorithms have rotation and scale invariant properties. These properties can degrade the results of pattern search, because the location and area of the defect patterns are not considered.

To overcome this problem, we propose a WBM pre-processing method using object detection, which is performed in the pre-processing function in Algorithm 1. By pre-processing, significant defect patterns in WBMs are detected and localized while non-significant noise is filtered out [35]. Our proposed pre-processing method addresses the limitations of supervised learning methods by employing the pre-processing function exclusively for denoising and defect pattern refinement, rather than integrating it as a core component of pattern search or similarity analysis. This ensures that our approach avoids the data dependency and retraining challenges associated with supervised learning-based methods.

The details of the pre-processing function are explained with Fig. 3 as follows: firstly, as shown in Fig. 3a, c, and e, an object detector builds the bounding box (denoted as bbox), which finds the location and area of the significant defect patterns. Secondly, the dies out of bbox are included in non-significant defect patterns called *noise* in this paper. Then, so-called *denoising* is performed to erase the non-significant defect patterns in the WBM in Fig. 3b, d, and f. The pattern search can continue after performing the pre-processed WBMs from the aforementioned two steps.

3.4 Image matching with keypoints and descriptors

The matching function in Line 13 of Algorithm 1 uses the keypoints and descriptors as its inputs. The keypoints and their descriptors can be obtained from classical image matching algorithms such as SIFT and ORB. The generated descriptors capture essential photometric information of keypoints. In general, the image matching using the above classical algorithms calculates Euclidean distance between two descriptors $des_i \in \vec{des}_{\mathbf{s}}$ for source WBM $\mathbf{s}$ and $des_j \in$



(a) Detected *Center* (b) *Center* after pre-processing

(c) Detected *Scratch* (d) *Scratch* after pre-processing

(e) Detected *Loc* (f) *Loc* after pre-processing

Fig. 3 Denoising in image pre-processing

$\vec{des}_{\mathbf{t}_k}$ for target WBM $\mathbf{t}_{k\in 1,\ldots,K}$, which is formulated as $\sqrt{\sum_{z=1}^{d}(des_i^z - des_j^z)^2}$ for $des_i, des_j \in \mathbb{R}^d$. Term d denotes the dimension of each descriptor. To find a matched descriptor $des_j \in \vec{des}_{\mathbf{t}_k}$ with a des_i, the nearest-neighbor search can be applicable, where a matched descriptor des_j with des_i denoted as $des_j \leftrightarrow des_i$ is calculated as follows:

$$des_j \leftrightarrow des_i = \underset{des_j}{\operatorname{argmin}} \sqrt{\sum_{z=1}^{d}(des_i^z - des_j^z)^2} \quad \text{for } des_i, des_j \in \mathbb{R}^d. \tag{1}$$

On the other hand, SuperPoint and SuperGlue can be used in image matching. Unlike classical image matching algorithms, both SuperPoint and SuperGlue are learning-based. We think that the above learning-based approaches can be effective in the pattern search for specific types of defect patterns in WBMs. In other words, the models in SuperPoint and SuperGlue can be trained from WBM datasets. Then, the trained models in SuperPoint and SuperGlue are used in the image matching for defect patterns in WBMs, which is the main difference from the classical algorithms. The pre-trained CNNs of SuperPoint can detect keypoints in real time. On the other hand, the matcher in SuperGlue calculates Mscores from the given keypoints and descriptors from SuperPoint. SuperGlue considers the positional relationships between keypoints based on graph attention networks (GATs), enabling robust matching for both single and mixed defect patterns. The model used in SuperGlue is trained to update the weights of GATs extracted from the same defect patterns. In this process, while each keypoint is represented as a node, the relationships between keypoints are represented as edges. Considering the above advantages, SuperPoint and SuperGlue can be mainly adopted in the proposed pattern searching method. It is noted that SuperGlue does not output the class of matched defect patterns, which is different from the trained models in the classification-based image matching [38].

The detailed process using SuperPoint and SuperGlue is described as follows: for the pattern searching, the image matching algorithm extracts keypoints from SuperPoint for the source WBM **s**. Unnecessary keypoints can be removed by specifying the maximum number of keypoints denoted as $D_{max}(\vec{kp}_{\mathbf{x}})$ in Algorithm 1. Then, descriptors are generated from the extracted keypoint, where each descriptor contains both the position and visual information of a keypoint $kp_{\mathbf{x}}$. All descriptors are embedded into a high-dimensional vector denoted as ${}^{(0)}\vec{x}_i$ through multi-layer perceptron (MLP) as follows:

$${}^{(0)}\vec{x}_i = \vec{des}_i + \mathrm{MLP}(\vec{pos}_i), \tag{2}$$

where the superscript (0) denotes that ${}^{(0)}\vec{x}_i$ is the input for the following attentional aggregation. In Eq. 2, $des_i \in \vec{des}_{\mathbf{s},\mathbf{t}_k}$ and $pos_i \in \vec{pos}_{\mathbf{s},\mathbf{t}_k}$ represents a visualized descriptor and its position, respectively.

The keypoints of the defect patterns in two WBMs are extracted by attentional aggregation between similar source WBM and targets via GATs. The keypoints are then combined to make a so-called message denoted as msg in [39, 40] and connected via *edges*. Edges for all keypoints can be denoted as $j : (i, j) \in \varepsilon$, where i and j indicate the indexes of any keypoints having connections via edges ε. In this case, the aggregated result from all keypoints of the source and target WBMs is denoted by message $msg_{\varepsilon\to i}$. The message is formulated as follows:

$$msg_{\varepsilon\to i} = \sum_{j:(i,j)\in\varepsilon} \alpha_{ij}\mathbf{v}_j, \tag{3}$$

where α_{ij} can be calculated using the Softmax over the key-query similarities [30]. Term α_{ij} is formulated as follows:

$$\alpha_{ij} = Softmax_j(\mathbf{q}_i^\top \mathbf{k}_j). \tag{4}$$

In Eqs. 3 and 4, $\mathbf{v}_j$, $\mathbf{q}_i$, and $\mathbf{k}_j$ denote values, queries, and keys, which are computed as a linear projection of deep features of the GNNs [30]. The matched descriptors for keypoints between target WBM $\mathbf{t}_k$ and source WBM **s** are represented by the following linear projection:

$$f_i^{\mathbf{t}_k} = \mathbf{W}\cdot x_i^{\mathbf{t}_k} + \mathbf{b}. \tag{5}$$

$$f_j^{\mathbf{s}} = \mathbf{W}\cdot x_j^{\mathbf{s}} + \mathbf{b}. \tag{6}$$

A score matrix denoted as $S(i, j) =< f_i^{\mathbf{t}_k}, f_j^{\mathbf{s}} >$ is obtained from the inner product of $f_i^{\mathbf{t}_k}$, $f_j^{\mathbf{s}}$, where operator $< \mathbf{a}, \mathbf{b} >$ denotes the inner product of any **a** and **b** [30]. Each element of the score matrix means Mscore from the matching between two keypoints.

3.5 Refining similarity measurement by adjusting Mscore with Confidence and MoD scores

In the matching function, SuperGlue and several existing image matching algorithms [26, 28, 31] use the average Mscore to measure the similarity between image patterns. As far as we know, it is noted that no metric-based methods have been developed to adjust the average Mscore considering the characteristics of the pattern search for defect patterns in WBMs. Therefore, we propose two similarity metrics for adjusting the matching results between defect patterns in WBMs.

3.5.1 Confidence score

In the pattern search of general image dataset, the scale and dimension of the target images can determine the Mscore between any matched descriptors from keypoints. In the image matching using unsupervised learning, the increasing redundant descriptors can degrade the image matching results between defect and noisy patterns. To address the above problem, we proposed to normalize the Mscore $m \in \vec{m}_{des_j \leftrightarrow des_i}$ in Algorithm 1 to reduce the influence of the noisy patterns. When a m is normalized considering the maximum and minimum values of Mscore denoted as m_{max} and m_{min}, the normalized Mscore of m is denoted as m_{norm}, which is equated in Eq. 7 as follows:

$$m_{norm} = \frac{m - m_{min}}{m_{max} - m_{min}}, \tag{7}$$

where confidence denoted as $conf := m_{norm}$ represents the confidence for the matched descriptors from keypoints. The visualization of $conf$s for all matched keypoints is shown in Fig. 4. We note that because the matched descriptors are generated from the keypoints, the matched keypoints can be visualized. Then, Confidence score denoted as $score_{conf}$ is calculated by averaging all confidences $conf$s from all matches in $\mathbf{x}$ and $\mathbf{t}_k$, which is equated as follows:

$$score_{conf_k} = \frac{\sum_{\forall des_j \leftrightarrow des_i} conf^2}{length(\vec{m}_{des_j \leftrightarrow des_i})}. \tag{8}$$

By averaging all $conf$s, it is concluded that Eq. 8 can adjust the average Mscore from the matching result between defect patterns.

3.5.2 MoD score

In the general image matching algorithm, each Mscore contains the matching result between descriptors for the same-sized patterns. Therefore, in the matching for defect patterns in WBMs, the naive calculation of Mscore does not consider the characteristics of pattern search in WBMs because each naive Mscore cannot consider the location and area of defect patterns detected from an object detector in the pre-processing function. Therefore, we propose a new metric called MoD score, which can be used as similarity after adjusting the average Mscore from the image matching of the denoised defect patterns after pre-processing. In the search process, a bounding box (bbox) with x- and y-axes coordinate values are obtained from the object detector in the pre-processing function. In Eq. 9, the area of the defect pattern is calculated by multiplying the width and height of the bbox as follows:

$$A_{bbox} = W_{bbox} \times H_{bbox}, \tag{9}$$

which is visualized in Fig. 5a. Then, a penalty AP_k depending on the area of each defect pattern is formulated as follows:

$$AP_k = \begin{cases} \frac{A_{\mathbf{t}_k}}{A_{\mathbf{s}}}, \text{ if } & A_{\mathbf{s}} \geq A_{\mathbf{t}_k}, \\ \frac{A_{\mathbf{s}}}{A_{\mathbf{t}_k}}, \text{ if } & A_{\mathbf{s}} < A_{\mathbf{t}_k}, \end{cases} \tag{10}$$

where $A_{\mathbf{s}}$ and $A_{\mathbf{t}_k}$ refer to the defect pattern area of source WBM $\mathbf{s}$ and target WBMs $\mathbf{t_k}$, respectively. As shown in Eq. 10, AP_k can be calculated by one of the following three cases: (1) the area in a source WBM $A_{\mathbf{s}}$ is larger than the area of a target WBM $A_{\mathbf{t}_k}$; (2) $A_{\mathbf{t}_k}$ is larger than $A_{\mathbf{s}}$; (3) the defect pattern is undetected in any region of either $\mathbf{s}$ and $\mathbf{t}_k$. By applying a penalty depending on various cases, the pattern search can be sensitive to the area variations of the defect patterns. A penalty on the location of the defect pattern denoted as LP_k is calculated by Eq. 11 as follows:

$$LP_k = \frac{Region_{\mathbf{s}} \cap \mathbf{t}_k}{Region_{\mathbf{s}}}. \tag{11}$$

While AP_k considers the ratio of $A_{\mathbf{s}}$ to $A_{\mathbf{t}_k}$, LP_k considers the ratio of an overlapped area in each location between $\mathbf{s}$ and $\mathbf{t}_k$. An example of calculating LP_k is visualized in Fig. 5b, where a WBM is divided into multiple cells $\in \{A, \ldots, P\}$. The total area of the cells in the bbox is defined as $Region$. If $Region_{\mathbf{s}} \in \{A, B, E, F\}$ and $Region_{\mathbf{t}_k} \in \{E, F\}$, $LP_k = 0.5$. As a result, a MoD score denoted as $score_{MoD}$ is calculated

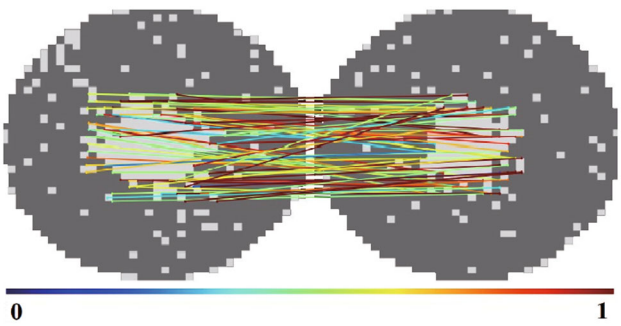


Fig. 4 Visualization of the $conf$s of the all matched keypoints

(a) Area of defect pattern (b) Cells of defect pattern

Fig. 5 Area and cells of a defect pattern

by assigning the above penalty to the average Mscore for all the matched descriptors between $\mathbf{s}$ and $\mathbf{t}_k$, which is equated in Eq. 12 as follows:

$$score_{MoD_k} = LP_k \times AP_k \times \frac{\sum_{\forall des_j \leftrightarrow des_i}^{m}}{length(\vec{m}_{des_j \leftrightarrow des_i})}. \quad (12)$$

Engineers can choose either Confidence score or MoD score as the similarity between defect patterns according to the purpose of their pattern search and the usage of pre-processing. While the normalized image matching results of Confidence score can be useful in pattern searches without using pre-processing, $score_{MoD}$ can be preferred in the pattern search for significant defect patterns after pre-processing. In the following section, the similarity of MoD score after pre-processing shows better performance in the visualized search results for detected significant defect patterns.

4 Experimental results and analysis

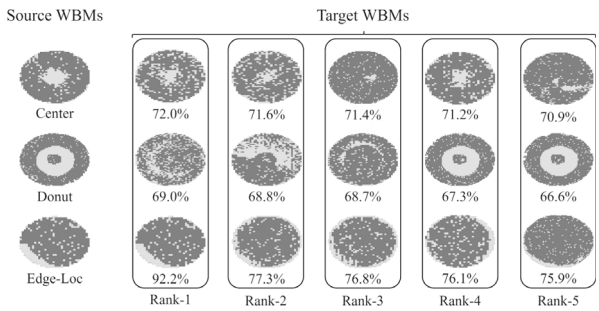
4.1 Experimental setup

We evaluated the proposed search algorithms on industry-driven WM-811K [20] and MixedWM38 [41] datasets. The WM-811K dataset has 811,457 WBMs with 8 single defect pattern classes (*Center*, *Donut*, *Loc*, *Edge-Loc*, *Edge-Ring*, *Scratch*, *Near-full*, and *Random*). The MixedWM38 dataset has 38,015 WBMs with 38 single and mixed defect pattern classes, which includes the above 8 classes from the WM-811K dataset. In [20], it is noted that the WM-811K dataset originated from real wafers of Taiwan Semiconductor Manufacturing Company. The WM-811K and MixedWM38 datasets have a wide range of resolutions from 26 × 26 to 50 × 60.

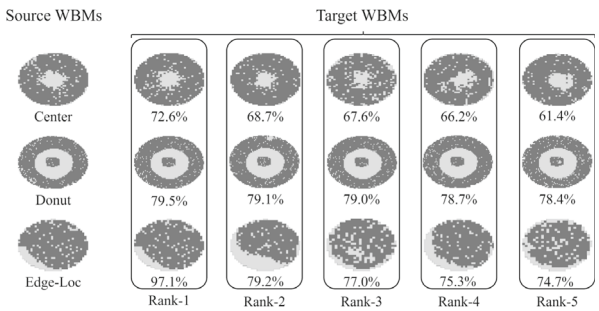
For the pre-processing, we chose you only look once (YOLO) v8 [42] as the object detector. The detailed description for training the YOLOv8 was as follows. We used Adam [43] optimizer, having the *learning rate* of 1e−2 and the number of epochs of 20. We also used a small *weight decay* of 5e−4 to improve generalization performance and avoid overfitting. Note that we manually created an annotation file for the defect patterns of the WBMs used in training. In the proposed pattern search, SuperPoint [28] was used to extract keypoints and generate descriptors. Besides, SuperGlue [30] was used as the matcher to calculate Mscores between matched descriptors. During training, all WBMs were resized to 640 × 480 for SuperPoint and 640 × 640 for SuperGlue. Then, the resized images were converted to grayscale. The Adam optimizer was used, having the *learning rate* of 1e−4, no weight decaying, and 20 training epochs. Since the training is to extract keypoints and perform image matching, we did not perform the pre-processing on WBMs.

4.2 Experiments on single defect patterns

Figure 6 shows single defect patterns for classes *Center*, *Donut*, and *Edge-Loc* on the WM-811K dataset. Figure 6a shows 5 similar WBMs ordered based on each Confidence score without using pre-processing. A term *Rank-i* is assigned in descending order of high similarity. The pattern search for *Center* in a source WBM shows that the defect patterns of *Rank-1* and *Rank-2* had close locations and similar areas with the defect pattern in the source WBM. On the other hand, the pattern search for *Donut* shows that the defect patterns of highly ranked WBMs had different shapes from each other. Instead, the defect patterns of *Rank-4* and *Rank-5* were very similar to the defect pattern in the source WBM. We conclude that the Confidence scores were high due to keypoints extracted from non-significant defect patterns called noise. In the search for *Edge-Loc*, the defect pattern of *Rank-1* is very similar to that of the source WBM. However, *Edge-Loc* in the source WBM differs from the defect patterns from *Rank-*



(a) Visualized results using the proposed method without pre-processing



(b) Visualized results using the proposed method with pre-processing

Fig. 6 Visualized search results for single defect patterns

2 to *Rank-5* in their locations. Therefore, the pattern search without pre-processing did not show acceptable performance due to the noise in WBMs.

The above weakness was improved by the proposed pre-processing in the following. Figure 6b shows several WBMs after performing the image pre-processing and their MoD scores. In the search results of *Center*, high MoD scores were assigned to the defect patterns with close locations and similar areas. Besides, *Donut* looks clearer after denoising from the pre-processing. Figure 6b shows that defect patterns with close locations and similar areas were searched well with high similarity values. In the search for *Center* and *Edge-Loc*, the difference of locations in *Rank-5* showed a significant drop in the MoD score. Besides, in the search for *Edge-Loc*, differences in the area of the defect patterns between source and target WBMs degraded the MoD score in *Rank-2*. Therefore, the cases with lowered MoD scores show that the pre-processing and application in the MoD score adjusted the naive image matching results considering the varying locations and areas of the defect patterns in the proposed pattern search method.

Table 1 summarizes the pattern searches on WM-811K dataset with single defect patterns, where the defect patterns with high similarity between a source WBM and other 1977 target WBMs were searched. When the average Mscore was used as the similarity between defect patterns, the image matching results were not adjusted without applying the penalty of Confidence score or MoD score. Although the pattern search did not classify the defect patterns into pre-determined classes, the class of the searched *Rank-1* defect pattern in a target WBM gave information about how well the searched patterns were similar to the defect pattern of a source WBM. The term *Best* means the similarity of *Rank-1* WBM. Besides, the term *Worst* means the lowest similarity among search results. When using the naive average Mscore and Confidence score, 4 and 3 defect patterns had different labels from those of the defect patterns in source WBM, respectively. On the other hand, with the proposed MoD score, only one defect pattern in a target WBM had a label different from that of the source WBM. Therefore, we note that the similarity using pre-processing and MoD score can be useful in the search for defect patterns on the WM-811K dataset.

Whereas the *Best* values did not refer to outstanding differences among the pattern searches with different similarity metrics, the *Worst* values of Confidence score and MoD score were smaller than those of the average Mscore. Notably, the *Worst* and *Mean* values of MoD score were smaller than those of the average Mscore and Confidence score. We expect that the pattern search using the MoD score as a similarity metric can filter out so-called *vaguely* similar defect patterns by assigning them a small similarity, so that denoised defect patterns in WBMs can be well searched. On the other hand, when a defect pattern appeared in the large area of a WBM such as *Loc*, *Edge-Loc*, and *Scratch*, the pre-processing can detect the defect pattern by showing large bboxes, which proves that the trained YOLOv8 detector can be effective in finding the defect patterns that appear on the entire region of a WBM. In the search for *Near-full*, because the defects were scattered on the entire region, the penalty of the difference in locations and areas in the MoD score could not be applied. However, considering the searched defect patterns and their characteristics in Table 1, the proposed pattern search using MoD score was effective in finding a single defect pattern using pre-processing.

4.3 Experiments on mixed defect patterns

In Fig. 7, we conducted experiments on the MixedWM38 dataset to evaluate the proposed image search for the following mixed defect patterns:

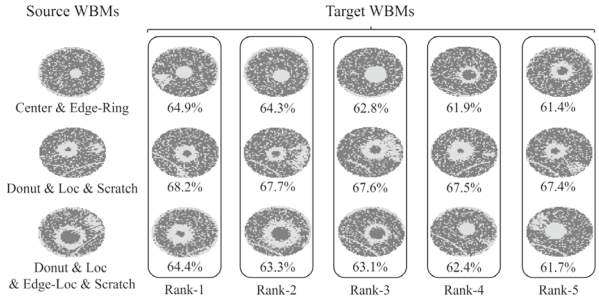
- CaseI: *Center* & *Edge-Ring*
- CaseII: *Donut* & *Loc* & *Scratch*
- CaseIII: *Donut* & *Loc* & *Edge-Loc* & *Scratch*.

Figure 7a visualizes the search results and lists their Confidence scores without using pre-processing. The pattern search for CaseI from *Rank-1* to *Rank-3* shows acceptable visualized results for similar defect patterns without using pre-processing. However, in the search for the defect patterns of *Rank-4* and *Rank-5* with *Donut* shape, the pattern search without using pre-processing seemed to be confused between *Center* and *Donut*. The pattern search of CaseII in Fig. 7a shows low similarity values in the pattern searching for CaseII due to the different locations and areas in the mixed *Loc* and *Scratch*. The pattern search for CaseIII also shows low similarity values in the pattern search for the *Edge-Loc*. However, in the pattern searches for Cases II and III in Fig. 7a, *Donut* was well searched in *Rank-1*, *Rank-2*, and *Rank-3*. Therefore, when pre-processing was not adopted, the effects of noise on defect patterns were significant in mixed defect patterns. Besides, the Confidence scores listed in Fig. 7a were lowered by the difference in locations and areas between pattern searches.

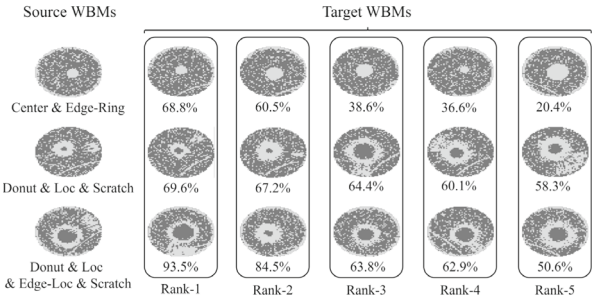
On the other hand, Fig. 7b shows the pattern search using pre-processing. The pattern search for CaseI performed well for all target WBMs having mixed two defect patterns, showing the significant defect patterns in any ranked WBMs. However, in the pattern search for CaseII, because different significant defect patterns were mixed, it is expected that the locations and areas of defect patterns could determine the similarity between defect patterns. The visualizations for *Donut* and *Loc* and *Scratch* in CaseII show that the defect patterns with close locations and similar areas can be well searched. There were many keypoints extracted for *Edge-Loc* because it created a bbox on the entire region of a WBM.

Table 1 Average Mscore, Confidence score, and MoD score in searches for 8 defect patterns

Source WBM	Average Mscore (w/o pre-processing)				Confidence score (w/o pre-processing)				MoD score (using pre-processing)			
	Rank-1	*Best*	*Worst*	*Mean*	*Rank-1*	*Best*	*Worst*	*Mean*	*Rank-1*	*Best*	*Worst*	*Mean*
Center	Center	73.9%	34.8%	67.9%	Center	71.6%	20.3%	67.9%	Center	73.3%	2.4%	6.9%
Donut	Donut	73.7%	24.1%	61.3%	Donut	71.8%	10.0%	56.3%	Donut	73.7%	0.9%	25.9%
Loc	Center	75.4%	30.2%	68.9%	Center	73.8%	14.3%	65.7%	Loc	68.9%	1.3%	17.9%
Edge-Loc	Center	76.9%	27.4%	69.2%	Edge-Loc	74.3%	11.8%	66.0%	Edge-Loc	33.5%	0.8%	10.7%
Edge-Ring	Edge-Ring	76.3%	28.8%	67.8%	Edge-Ring	74.8%	13.9%	64.3%	Edge-Ring	77.5%	0.09%	41.4%
Scratch	Random	71.9%	27.1%	61.4%	Loc	67.5%	10.0%	56.4%	Scratch	42.0%	6.3%	18.8%
Near-full	Center	80.8%	37.4%	74.9%	Random	80.0%	37.4%	72.9%	Random	78.8%	0.05%	32.2%
Random	Random	68.9%	21.2%	47.2%	Random	66.6%	10.0%	37.9%	Random	70.5%	0.2%	35.9%



(a) Visualized results using the proposed method without pre-processing



(b) Visualized results using the proposed method with pre-processing

Fig. 7 Visualized search results for mixed defect patterns

Therefore, in the pattern search for CaseIII, *Edge-Loc* significantly degraded the pattern search results for *Donut* and *Scratch*. Whereas a single *Edge-Loc* in Fig. 6b can be well searched, complex extracted keypoints from *Edge-Loc* can disturb the search for other defect patterns such as *Scratch* and *Loc* in the search for mixed patterns. Therefore, it is concluded that when small defect patterns among mixed patterns were scattered on the entire region of a WBM such as *Edge-Loc*, the scattered defect pattern significantly degraded the pattern search results. As a result, we note that pre-processing significantly improves the accuracy of the search when dealing with mixed defect patterns. While a single defect pattern or significant defect patterns among mixed defect patterns were well matched, the pattern search for scattered defect patterns on the entire region of a WBM remains challenging in the proposed method. Further research should be done to address the weakness of the above case.

4.4 Comparison with state-of-the-art models

To enhance the evaluation and demonstrate the effectiveness of the proposed method, we conducted comparative experiments with state-of-the-art (SOTA) models. While conventional approaches rely on labeled data during training or require labels for testing, our method does not require class labels during training and testing. Therefore, we adopted a metric based on nearest-neighbor similarity to ensure a fair comparison.

During evaluation, we computed the similarity between WBM in the test set and all WBMs in the train set using Confidence and MoD scores. Based on the similarity, we searched the Top-N most similar WBMs in the train set. If the ground-truth label of the test WBM appeared among the labels of Top-N most similar WBMs, the prediction was considered correct. The resulting metric was defined as Top-N accuracy formulated as follows:

$$\text{Top-}N\text{Accuracy} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{1}\left(y_i \in \mathcal{T}_i^{(N)}\right), \tag{13}$$

where $\mathcal{T}_i^{(N)}$ denotes the set of labels from the Top-N most similar training WBMs to the i-th test WBM, and y_i is the ground-truth label of the i-th test WBM. The indicator function $\mathbb{1}$ returns 1 if the ground-truth label y_i is found within the Top-N set, and 0 otherwise. Table 2 shows a performance

Table 2 Comparison of wafer defect classification methods across WM-811K and MixedWM38 datasets

Dataset	Method	Input	Parameters (M)	Accuracy (%)
WM-811K	U-Net+CBAM [3]	256 × 256 × 1	-	96
	Opt-RseDCNN [4]	96 × 96 × 1	-	90
	MFFP-Net [5]	224 × 224 × 3	48.1	97
	Modified VGG16 [6]	64 × 64 × 1	-	73
	ACDDPM-ResNet [8]	48 × 48 × 3	25.5	97
	Proposed w/o pre-proc.	640 × 480 × 1	11.2	90/88
	Proposed w/ pre-proc.	640 × 480 × 1	14.5	94/92
MixedWM38	Modified VGG16	64 × 64 × 1	-	96
	CNN+Augmentation [7]	52 × 52 × 1	-	96
	ACDDPM-ResNet	48 × 48 × 3	25.5	98
	Proposed w/o pre-proc.	640 × 480 × 1	11.2	80/71
	Proposed w/ pre-proc.	640 × 480 × 1	14.5	91/84

comparison of the proposed method with several SOTA models in terms of input resolution, model size, and classification accuracy. To ensure clarity in presentation, the classification accuracy of our method is represented in the format of Top-5 accuracy/Top-1 accuracy.

On the WM-811K dataset, MFFP-Net and ACDDPM-ResNet showed the highest Top-1 accuracy among different models. Besides, MFFP-Net and ACDDPM-ResNet relied on RGB inputs and required approximately 23.7M and 25.5M parameters, respectively. In particular, ACDDPM-ResNet operated on a very low input resolution of 48 × 48, which may lead to loss of critical pattern information when applied in practical industrial applications. In contrast, the proposed method with pre-processing used grayscale inputs with fewer parameters and achieved 92% Top-1 and 94% Top-5 accuracieo. These results demonstrate that the proposed method performs comparably with SOTA models, despite not requiring class labels.

On the MixedWM38 dataset, most models showed competitive classification performance. Most of all, ACDDPM-ResNet achieved the highest Top-1 accuracy of 98%, followed by CNN+Augmentation with 94%. However, ACDDP M-ResNet used 25.5M parameters and RGB inputs, while CNN+Augmentation operated on a very low input resolution of 48 × 48, which may result in the loss of critical information when applied in practical industrial environments. On the other hand, the proposed method achieved 91% Top-5 and 84% Top-1 accuracies using grayscale inputs and 14.5M parameters when pre-processing was applied. Without pre-processing, the performance slightly decreased to 80% Top-5 and 71% Top-1 accuracies. This performance gap indicates that pre-processing improves the consistency of defect pattern representation, which directly contributes to the accuracy of similarity-based search. Notably, although the proposed method was trained without any class labels, it achieved Top-5 accuracy that is comparable to those of SOTA models that adopted supervised learning-based methods. This result suggests that the similarity-based search approach is capable of identifying relevant defect patterns even without explicit class information.

As a result, the experimental results on both WM-811K and MixedWM38 datasets demonstrate that the proposed pattern search method provides comparable performance to the SOTA models, while requiring fewer parameters and operating with lower input resolution. Furthermore, the consistent accuracy across datasets suggests that the method is applicable to industrial scenarios where labeled data is limited and pattern search is preferred over classification.

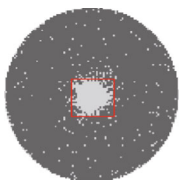
4.5 Visualized evaluation of the proposed pattern search

To evaluate the effectiveness of the proposed pattern search method, additional visualizations and analyzes are provided.

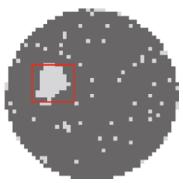
4.5.1 Advantages of image matching over template matching

In template matching, the primary focus is on identifying geometric similarity between defect patterns. Figure 8a presents a centered defect pattern from a source WBM used as the template, while Fig. 8b shows a defect pattern in the target WBM with a shape resembling the template in Fig. 8a. However, when similarity is assessed using Euclidean distance, the spatial difference between the template and the defect pattern in the target WBM becomes a significant challenge. Since Euclidean distance directly measures the displacement between corresponding pixels, even minor location shifts can substantially impact the similarity measurement. These visual examples in Fig. 8 illustrate why naive template matching may not be well-suited for pattern search in WBMs.

Figure 9 further elaborates on the performance of template matching by presenting visualized search results for both single and mixed defect patterns. The datasets adopted for experiments were the same as previously used for single and mixed pattern analysis in Sects. 4.2 and 4.3. In Fig. 9a, the results for single defect patterns demonstrate high sim-

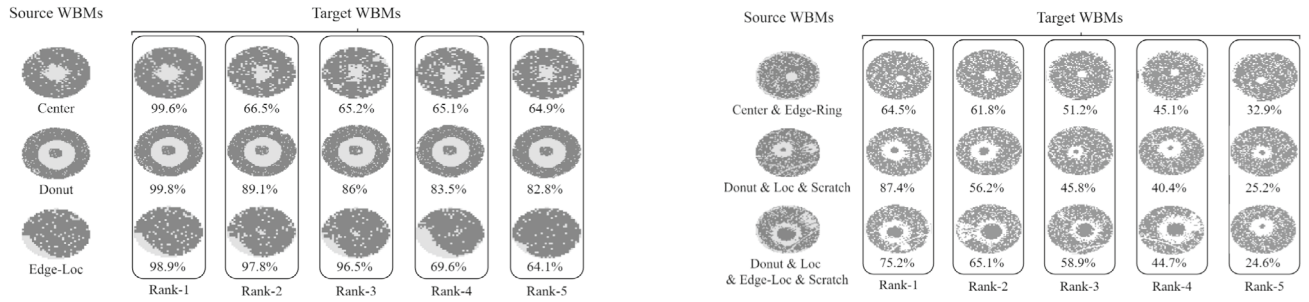


(a) Template of source WBM



(b) Pattern in target WBM

Fig. 8 Matched *Center* using template matching



(a) Visualized results for single defect patterns

(b) Visualized results for mixed defect patterns

Fig. 9 Visualized search results for single and mixed defect patterns using template matching

ilarity scores, ranging from 64.1 to 99.6% across source and target WBMs. It demonstrates that template matching is highly effective for single, well-defined defect patterns, as the geometric similarity between the template and target is consistent, with minimal interference from other patterns. However, Fig. 9b shows the results for mixed defect patterns, where similarity scores drop significantly, ranging from 44.7 to 87.4%. This decline occurs because mixed defect patterns introduce complexity and overlapping features, which confuse the Euclidean distance-based similarity measurement. The presence of multiple, spatially overlapping defects disrupts the precise alignment required for effective template matching, leading to reduced search accuracy and reliability in pattern search for mixed defects.

4.5.2 Advantages of denoising in keypoint-based image matching

The keypoint-based matching algorithm is robust to the varying location and area of defect patterns. In the experiments of keypoint extraction, Fig. 10 visualizes extracted keypoints in red. Figure 10 shows that many keypoints were found around the centered defect pattern. Besides, the defects around the wafer edge also produced the keypoints around the edge of the WBM. In semiconductor manufacturing, defects around the wafer edge are common, which means that almost all WBMs can have keypoints around the edge of WBMs. The common defects around the edge can be mixed with other defect patterns, being any noise in the search for other significant defect patterns. In other words, the common defect patterns could be considered non-significant defect patterns.

The randomly scattered small patterns in Fig. 10a produced the scattered keypoints on the entire region, which can be another non-significant defect pattern. In Fig. 10b, when a significant defect pattern and its keypoints in the center of the WBM are visualized, the keypoints around the edge and scattered on the entire region are filtered out. Whereas many keypoints from the non-significant defect patterns are matched in Fig. 11a, b shows the matching results between source WBM and *Rank-1* after denoising, which proves that the keypoints in source WBM and target WBMs from the significant defect pattern were well matched.

Figures 12 and 13 prove the effectiveness of the proposed denoising with additional visualizations of the searched defect patterns. Additional experiments were conducted on 5 complex single defect patterns: *Loc*, *Donut*, *Edge-Ring*, *Scratch*, and *Random*. The results of pattern searches without denoising are presented in Fig. 12. The experiments showed that except for *Random*, the source WBM and target *Rank-1*s in the pattern search only using Confidence score had the

(a) Without denoising

(b) After denoising

Fig. 10 Impact of pre-processing on keypoint extraction

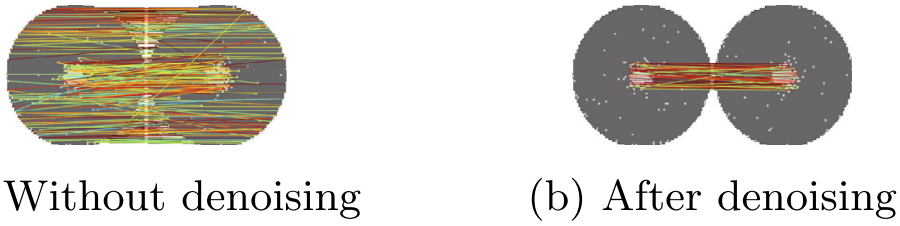


Fig. 11 Matched defect patterns depending on denoising

same labels. However, *Random* in source WBM was matched with *Edge-Ring*, showing that defect patterns without denoising cannot be well searched. In Fig. 13, the searched patterns after denoising had the same labels in all cases, showing that *Random* in source WBM was matched with *Random* in a target WBM.

4.5.3 Advantages of deep learning-based image matching

In Fig. 14, we compared the performance of traditional image matching methods, such as SIFT and ORB, with the adopted deep learning-based image matching method, SuperPoint. As shown in Fig. 14a and b, traditional methods rely on handcrafted keypoint detection, which struggles with detecting defect patterns in WBMs. Due to variations in defect size, shape, and intensity, traditional methods frequently fail to extract keypoints effectively, leading to unreliable keypoint detection results. In contrast, SuperPoint, which leverages a deep learning-based keypoint detection and description process, significantly improves detection performance. By learning feature representations from a synthetic dataset, SuperPoint achieves higher robustness against geometric variations and noise, enabling more consistent keypoint detection in complex defect patterns. The results in Fig. 14c demonstrate that SuperPoint successfully detects keypoints in defect patterns where SIFT and ORB fail, highlighting its advantages in WBM analysis. These findings suggest that deep learning-based image matching algorithms provide a more effective solution for defect pattern analysis in WBMs, making them a suitable choice for semiconductor manufacturing applications.

4.6 Processing time comparison for different WBM resolutions and methods

The speed of the proposed pattern search depends on the resolution of each WBM, so we compared the processing times depending on the resolutions of WBMs. Besides, by comparing with the processing times of template matching [44], the computational overhead of the keypoint-based image matching can be analyzed in the experiments. Figure 15 shows the inference times of the models used in the proposed search depending on 6 different resolutions, where the image matching with two WBMs was performed in each inference. The setup for source WBM and target WBMs was based on the experiments in Table 1. The processing times were estimated on an NVIDIA RTX 4090 and AMD Ryzen Threadripper 16-core PRO 5955WX. In the inference, all WBMs from the WM-811K dataset were resized to one of the 6 resolutions in Fig. 15. Template matching was implemented using the functions provided by OpenCV and their default settings.

In Fig. 15, while template matching was the fastest on low-resolution WBMs, it became dramatically slow on high-resolution WBMs. On the other hand, the proposed pattern search also increased the processing times with resolutions. However, the increase in the ratio of processing time to its resolution was not as steep as in the cases of template matching. In 2560×1920 resolution, the proposed pattern search with-

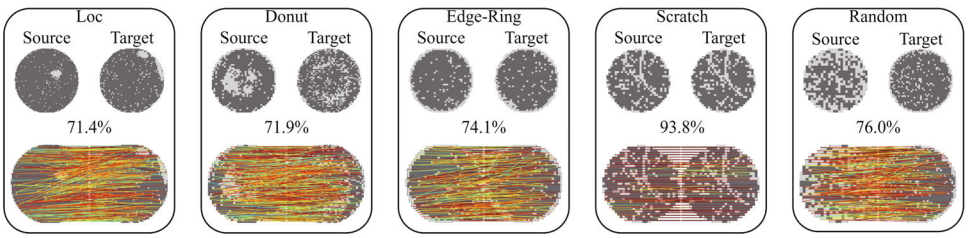


Fig. 12 Searches without denoising

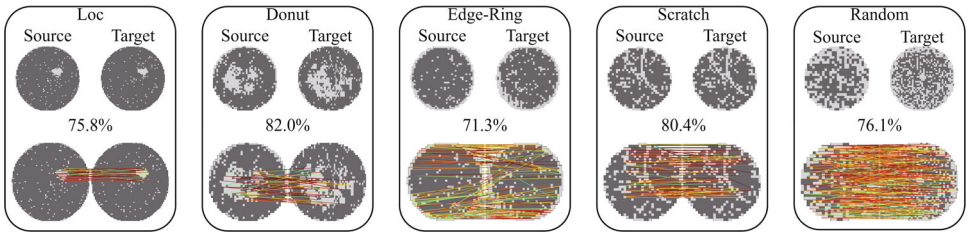
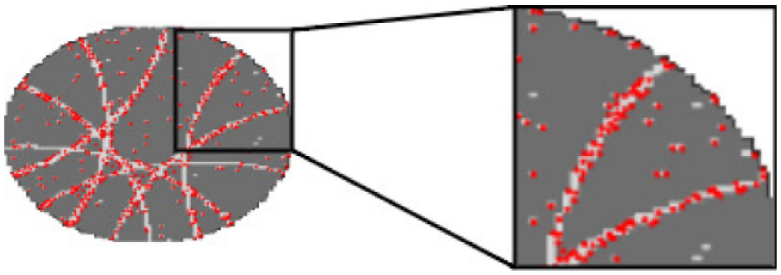
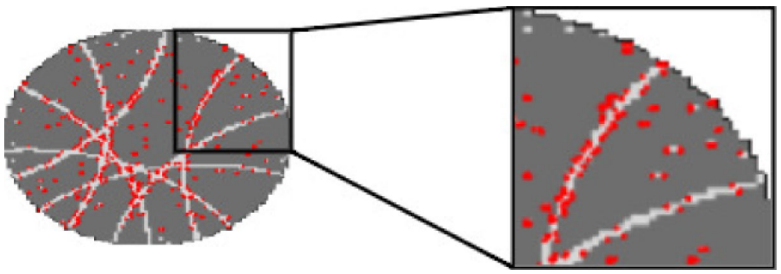


Fig. 13 Searches after denoising

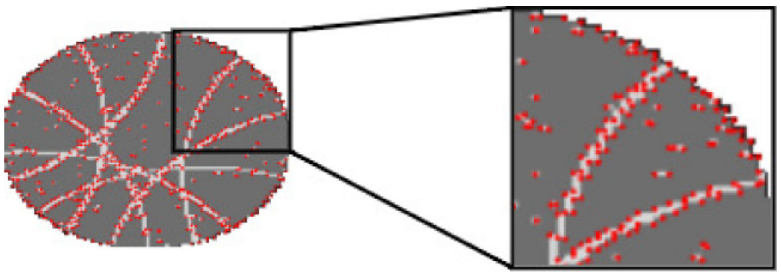
out using pre-processing showed 0.188s inference time for each WBM. Therefore, about 459,574 pattern searches can be performed to calculate the similarity between WBMs in a day. Using the pre-processing, the proposed pattern search required about 0.263 s inference time for each WBM in the same resolution. The above estimations show that the pre-



(a) Keypoint detection using SIFT algorithm



(b) Keypoint detection using ORB algorithm



(c) Keypoint detection using SuperPoint algorithm

Fig. 14 Comparison of different keypoint detectors

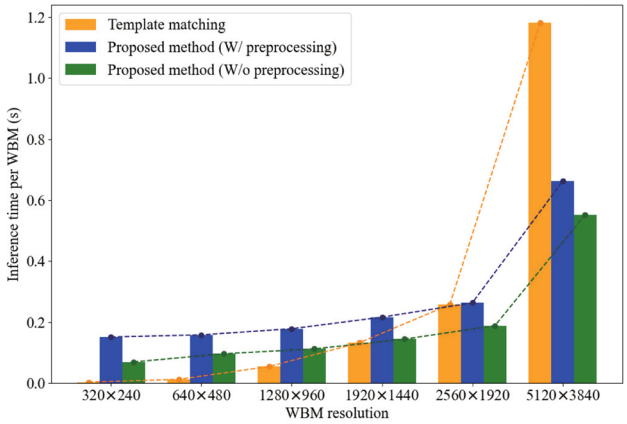


Fig. 15 Processing time comparison of different methods based on resolution

processing can reduce the number of extracted keypoints and the processing times in the pattern search.

5 Conclusion

This paper proposes a pattern search method to analyze the similarity between wafer defect patterns based on the image matching algorithm. To overcome the weakness of existing supervised and unsupervised methods, we propose a new robust pattern search method, including pre-processing, detection, description, and matching functions. Notably, experimental data prove the effectiveness of denoising, which focuses on the significant defect patterns. The similarity metrics called Confidence and MoD score adjust the Mscores of the image matching in WBMs by considering the normalization and the location and area of defect patterns. In experiments on the industry-driven datasets, we verified that the proposed pattern search method performs well in the searches for both single and mixed defect patterns. Considering the experimental data and analysis, the proposed method is useful to enhance the results of pattern searches in WBMs.

Author contribution Conceptualization: YoungWook Kwon, SuMin Oh. Methodology: YoungWook Kwon, SuMin Oh. Formal analysis and investigation: YoungWook Kwon, SuMin Oh, HyunJin Kim. Writing—original draft preparation: YoungWook Kwon, SuMin Oh, HyunJin Kim. Writing—review and editing: YoungWook Kwon, SuMin Oh, HyunJin Kim. Funding acquisition: HyunJin Kim. Supervision: HyunJin Kim

Funding This paper was result of the research project supported by SK hynix Inc. Also, this research was results of a study on the "HPC Support" Project, supported by the "Ministry of Science and ICT" and NIPA.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Taha K (2025) Observational and experimental insights into machine learning-based defect classification in wafers. J Intell Manuf 1–51
2. Geng H, Sun Q, Chen T, Xu Q, Ho T-Y, Yu B (2023) Mixed-type wafer failure pattern recognition (invited paper). In: 2023 28th Asia and South Pacific Design Automation Conference (ASP-DAC). pp 727–732
3. Cha J, Jeong J (2022) Improved U-Net with residual attention block for mixed-defect wafer maps. Appl Sci 12(4):2209
4. Wang F-K, Chou J-H, Amogne ZE (2022) A deep convolutional neural network with residual blocks for wafer map defect pattern recognition. Qual Reliab Eng Int 38(1):343–357
5. Chen Y, Zhao M, Xu Z, Li K, Ji J (2023) Wafer defect recognition method based on multi-scale feature fusion. Front Neurosci 17:1202985
6. Bae Y, Kang S (2023) Supervised contrastive learning for wafer map pattern classification. Eng Appl Artif Intell 126:107154
7. Shim J, Kang S (2023) Learning from single-defect wafer maps to classify mixed-defect wafer maps. Expert Syst Appl 233:120923
8. Li J, Tao R, Chen R, Chen Y, Zhao C, Huang X (2024) Sample-imbalanced wafer map defects classification based on auxiliary classifier denoising diffusion probability model. Comput Ind Eng 192:110209
9. Saqlain M, Abbas Q, Lee JY (2020) A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes. IEEE Trans Semicond Manuf 33(3):436–444
10. Lim C, Hur Y (2024) TripletMatch: wafer map defect detection using semi-supervised learning and triplet loss with mixup. IEEE Access
11. Ezzat AA, Liu S, Hochbaum DS, Ding Y (2021) A graph-theoretic approach for spatial filtering and its impact on mixed-type spatial pattern recognition in wafer bin maps. IEEE Trans Semicond Manuf 34(2):194–206
12. De La Torre J, Kent D, Pivin D, St Pierre E Dimensionality reduction and clustering by yield signatures to identify candidates for failure analysis. In: International symposium for testing and failure analysis, vol 84741. ASM International, pp 1–6
13. Hou X, Qin G, Lu Y, Yi M, Chen S (2024) A defect detection method of mixed wafer map using neighborhood path filtering clustering algorithm. J Electron Test 40(4):419–433
14. Kang M-S, Shin J-S, Lee D-H (2024) Similarity searching for wafer bin maps by measuring shape, location, and size similarities of defect patterns. Comput Ind Eng 196:110486
15. Alawieh MB, Boning D, Pan DZ (2020) Wafer map defect patterns classification using deep selective learning. In: 2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, pp 1–6
16. Barone M (2020) Image wafer inspection based on template matching. Comput Sci Inf Technol
17. Wang R, Wang S (2023) Similarity searching for fault diagnosis of defect patterns in wafer bin maps. Comput Ind Eng 185:109679
18. Hsu C-Y, Chen W-J, Chien J-C (2020) Similarity matching of wafer bin maps for manufacturing intelligence to empower industry 3.5 for semiconductor manufacturing. Comput Ind Eng 142:106358

19. Wang S, Yan S, Shen Q, Luo C, Ai J, Li L, Wang, D, Ding S, Xia Q (2021) Wafer defect map similarity search using deep learning in semiconductor manufacturing. In: 2021 China Semiconductor Technology International Conference (CSTIC). IEEE, pp 1–4
20. Wu M-J, Jang J-SR, 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. https://doi.org/10.1109/TSM.2014.2364237
21. Park S, Jang J, Kim CO (2021) Discriminative feature learning and cluster-based defect label reconstruction for reducing uncertainty in wafer bin map labels. J Intell Manuf 32:251–263
22. Hashemi NS, Aghdam RB, Ghiasi ASB, Fatemi P (2016) Template matching advances and applications in image analysis. arXiv:1610.07231
23. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60:91–110
24. Rublee E, Rabaud V, Konolige K, Bradski G (2011) Orb: an efficient alternative to sift or surf. In: 2011 international conference on computer vision, pp. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544
25. Verdie Y, Yi K, Fua P, Lepetit V (2015) Tilde: a temporally invariant learned detector. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 5279–5288
26. Yi KM, Trulls E, Lepetit V, Fua P (2016) Lift: learned invariant feature transform. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14. Springer, pp 467–483
27. Salti S, Tombari F, Spezialetti R, Di Stefano L (2015) Learning a descriptor-specific 3D keypoint detector. In: Proceedings of the IEEE international conference on computer vision. pp 2318–2326
28. DeTone D, Malisiewicz T, Rabinovich A (2018) Superpoint: self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp 224–236
29. Revaud J, De Souza C, Humenberger M, Weinzaepfel P (2019) R2D2: reliable and repeatable detector and descriptor. Adv Neural Inform Process Syst 32
30. Sarlin P-E, DeTone D, Malisiewicz T, Rabinovich A (2020) Superglue: learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)
31. Sun J, Shen Z, Wang Y, Bao H, Zhou X (2021) LoFTR: detector-free local feature matching with transformers. CVPR
32. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al. (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv:2010.11929
33. Lindenberger P, Sarlin P-E, Pollefeys M (2023) Lightglue: local feature matching at light speed. In: Proceedings of the IEEE/CVF international conference on computer vision. pp 17627–17638
34. Fan L, Zhang F, Fan H, Zhang C (2019) Brief review of image denoising techniques. Visual Comp Ind Biomed Art 2(1):7
35. Shinde PP, Pai PP, Adiga SP (2022) Wafer defect localization and classification using deep learning techniques. IEEE Access. 10:39969–39974. https://doi.org/10.1109/ACCESS.2022.3166512
36. Yuan T, Kuo W, Bae SJ (2011) Detection of spatial defect patterns generated in semiconductor fabrication processes. IEEE Trans Semicond Manuf 24(3):392–403
37. Wang R, Chen N (2022) Detection and recognition of mixed-type defect patterns in wafer bin maps via tensor voting. IEEE Trans Semicond Manuf 35(3):485–494. https://doi.org/10.1109/TSM.2022.3183008
38. Koch G, Zemel R, Salakhutdinov R, et al (2015) Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop, vol 2. Lille, pp 1–30
39. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. In: International conference on machine learning. PMLR, pp 1263–1272
40. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R, et al. (2018) Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261
41. 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. https://doi.org/10.1109/TSM.2020.3020985
42. Jocher G, Chaurasia A, Qiu J YOLO by Ultralytics (2023). https://github.com/ultralytics/ultralytics
43. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
44. Gonzalez R, Woods R (2001) Digital image processing. Addison-Eesle y Longman Publishing Co., Inc, Boston, MA, USA