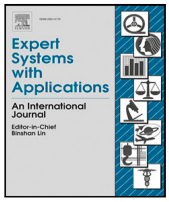
Expert Systems With Applications 238 (2024) 122301

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Semi-supervised imbalanced classification of wafer bin map defects using a Dual-Head CNN

Siyamalan Manivannan

Department of Computer Science, Faculty of Science, University of Jaffna, Sri Lanka

ARTICLE INFO

Keywords:
Wafer bin map classification
Semi-supervised learning
Deep learning
Semiconductor manufacturing

ABSTRACT

Wafer Bin Map (WBM) defect patterns are a critical aspect of identifying the root cause of manufacturing defects in the semiconductor industry. Semi-supervised learning (SSL) approaches have gained popularity for this purpose, as they can leverage both labeled and unlabeled data to improve model performance. However, SSL of WBM defect patterns is challenging due to class imbalance, where some defect classes have many more examples than others. Most of the existing SSL approaches assume a balanced dataset and often fail to provide satisfactory results when applied to imbalanced class problems. To address this issue, this work proposes a novel Dual-Head Convolutional Neural Network (CNN) architecture that contains two classifier heads. One classifier head maximizes overall classification scores, while the other aims to maximize per-class classification scores, providing equal attention to both majority and minority classes. The proposed CNN architecture uses pseudo-labels selected based on the outputs of these two classifiers to expand the labeled training set, which is then used to retrain the CNN. In this way, highly confident pseudo-labels are selected even from the minority classes, leading to better model training. Experiments show that the proposed approach is effective in handling class-imbalanced classification of WBM defect patterns, reporting state-of-the-art classification with an F1 score of 0.918, accuracy of 98.2% and a mean per-class accuracy of 91.7% using a lightweight ResNet-10 model as the backbone on the real-world public WBM dataset, WM-811K. The proposed approach's success suggests that it could be a valuable tool for improving the accuracy and reliability of WBM defect pattern classification in semiconductor manufacturing. The code is available at https://github.com/M-Siyamalan/SSL-DHCNN.

1. Introduction

Semiconductor devices are manufactured on circular slices of semiconductor material, typically silicon, known as wafers. A Wafer Bin Map (WBM) (Fig. 1) is a graphical representation that displays the areas of the wafer containing functional dies that meet specifications, as well as the areas that contain defective dies. The WBM is an important tool for semiconductor manufacturers, as it allows them to identify the root cause of defects in the manufacturing process. By analyzing the WBM, experts can identify patterns in the distribution of good and bad dies and adjust the manufacturing process to reduce the number of defects and increase the yield of good dies. However, manually identifying WBM defect patterns can be time-consuming and prone to errors. Automatically identifying defect patterns in WBM is crucial for the semiconductor manufacturing industry. It enhances accuracy, reduces the time and costs associated with manual inspection, and improves overall efficiency.

Several automated approaches have been proposed for WBM defect classification, most of which are based on fully supervised learning (FSL). These FSL approaches (e.g., Adly, Alhussein et al. (2015), Adly, Yoo et al. (2015), Chen et al. (2023), Chien, Hsu, and Chen (2013), Piao, Jin, Lee, and Byun (2018), Shin, Kahng, and Kim (2022) and Shin and Yoo (2023)) require large amount of labeled data to train, which is usually difficult to obtain as labeling is a tedious and costly process. In contrast, semi-supervised learning (SSL) (e.g., Kahng and Kim (2021), Sohn et al. (2020) and Zhang et al. (2021)) leverages both labeled and unlabeled data for training the system and has been shown to improve performance over FSL. However, most existing SSL approaches, including FixMatch (Sohn et al., 2020), assume a uniform distribution of the number of images from different classes, which is not representative of real-world scenarios. In reality, data often follows a long-tail distribution, with the majority of images belonging to one class and many classes having very few data points, resulting in data imbalance. In such cases, most existing SSL approaches fail to provide good results as the CNN trained on class imbalanced datasets are biased towards the classes which contains the majority of the data points. The classification of WBM is an example of highly imbalanced data classification, where the majority of WBM show no clear defect

E-mail address: siyam@univ.jfn.ac.lk.

https://doi.org/10.1016/j.eswa.2023.122301
Received 26 July 2023; Received in revised form 17 September 2023; Accepted 20 October 2023
Available online 31 October 2023

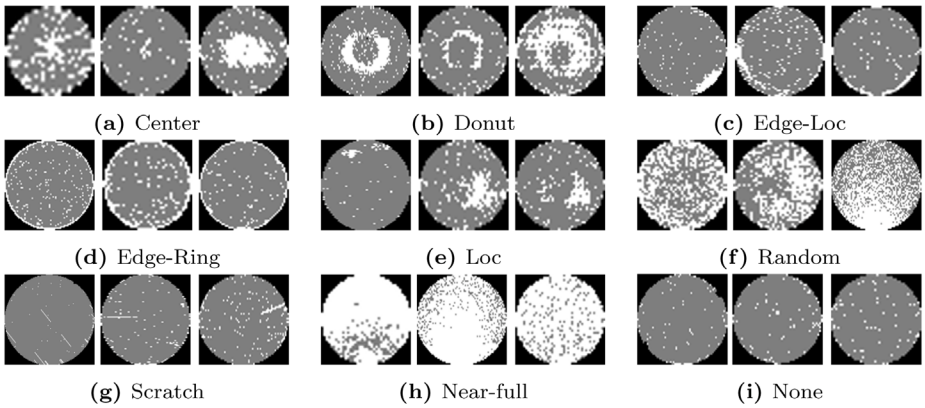


Fig. 1. Example images from different classes of the WM-811K dataset.[1].

patterns, while the remaining ones are distributed among different classes of defect patterns.

This work proposes a novel SSL approach for WBM classification. The proposed approach is based on a Dual-Head CNN architecture that is designed to handle data imbalance effectively. The proposed architecture contains two classifier heads. The first classifier head aims to maximize overall accuracy values, while the second one aims to maximize the average of per-class accuracy values. Initially, the proposed CNN is trained using the available labeled training data. Then, it is used to identify high-confident pseudo-labels from the unlabeled data using the output from both classifier heads. The experiments demonstrate that high confident pseudo-labels can be selected even from the minority classes with high precision and recall values by the proposed approach, leading to better CNN training. The newly selected data is then used to expand the labeled training set, which is subsequently used to retrain the proposed CNN. The proposed SSL approach is shown to be effective, particularly for imbalanced data classification of WBM. Comparative experiments indicate that the proposed approach is the new state-of-the-art for WBM defect classification.

In the following, the work related to WBM defect classification using both FSL and SSL approaches are reviewed in Section 2. The proposed methodology is explained in detail in Section 3. The experiments and results are discussed in detail in Section 4, including the evaluation of the proposed approach against the state-of-the-art methods. Finally Section 5 concludes this work.

2. Related work

Deep learning-based approaches for WBM classification have been widely studied in recent years. This section reviews these approaches under FSL and SSL setting.

2.1. FSL approaches

These approaches use labeled data to train deep learning models. Various deep learning-based FSL approaches have been explored for WBM defect classification, with most of them focusing on applying existing off-the-shelf CNN architectures for this problem. ResNet (Manivannan, 2022; Shin & Yoo, 2023), DenseNet (Manivannan, 2022), MobileNet (Shin & Yoo, 2023; Tsai & Lee, 2020), and AlexNet (Hsu & Chien, 2020) have been explored. For example, in Shin and Yoo (2023), different CNN architectures were compared, and MobileNet-V3 was found to be better in terms of both training speed and accuracy. In Tsai and Lee (2020), a deep learning-based method is proposed for WBM data augmentation and defect classification. In this approach, data augmentation was based on a CNN encoder–decoder architecture, and classification was performed using depthwise separable convolutions to reduce the number of parameters in the network. MixUp (Zhang, Cisse, Dauphin, & Lopez-Paz, 2017) was used with CNN in Shin et al. (2022) to identify mixed-type defective patterns. Ensemble CNN-based approaches usually report better performance than single model-based approaches, so they were also explored for WBM classification (Hsu & Chien, 2020; Misra, Kim, Kim, Shin, & Kim, 2022; Piao & Jin, 2022). For example, in Hsu and Chien (2020), a weighted majority voting is adopted to ensemble CNN models to achieve higher predictive performance. In Misra et al. (2022), features extracted from multiple pre-trained CNN models are concatenated, and a classification layer is trained on this concatenated representation to predict the label. As discussed earlier, FSL approaches require large amount of labeled data which is usually difficult to obtain as the labeling is a costly and time consuming process.

2.2. SSL approaches

SSL approaches are becoming increasingly popular for WBM defect classification as they leverage the available unlabeled data to improve model performance. However, the number of SSL approaches for WBM classification is still much lower compared to FSL approaches. An ensemble based SSL approach was recently proposed in Manivannan (2022) which involves training a set of CNN classifiers to predict the pseudo-label of each unlabeled image. The pseudo-labeled images are then used to update each CNN in the ensemble, resulting in improved performance over FSL baseline.

Self-supervised learning approaches also utilize unlabeled data to improve model's performance. However, they typically consists of two stages: In the first stage, a contrastive loss function is used with the available unlabeled data to pretrain the model to generate rich feature representations. In the second stage, the model is fine-tuned using labeled data for classification. Self-supervised learning based approaches also have been proposed for WBM defect classification in Hu, He, and Li (2021) and Kahng and Kim (2021).

In contrast to the above SSL approaches, the proposed approach is based on a Dual-Head CNN architecture that aims to handle imbalanced class problems effectively. Experiments show that the proposed approach performs significantly better than the above approaches proposed for WBM defect classification.

3. Methodology

The proposed approach is a pseudo labeling-based SSL technique that leverages unlabeled data to train the proposed Dual-Head CNN model. Initially, the CNN model is trained using the labeled data, and then this trained CNN is employed to select high-confident pseudo-labels from the unlabeled set. The selected subset, along with its

[1] https://www.kaggle.com/qingyi/wm811k-wafer-map.

Table 1
Notations used throughout in this manuscript.

Notation	Meaning
$\mathcal{X}$	Labeled set
x_i	A Labeled image
y_i	Label of x_i
N	Number of labeled images
$\mathcal{U}$	Unlabeled set
u_i	An unlabeled image
$\hat{y}_i$	Pseudo-label of u_i
M	Number of unlabeled images
C	Number of classes
H	Classifier head which is trained using the CE loss
H_W	Classifier head which is trained using the class balanced CE loss
w_c	Class weight for the class c
N_c	Number of images from class c
α	Weak augmentation
$\mathcal{A}$	Strong augmentation

pseudo-labels, is then considered as a labeled set and combined with the original labeled set to retrain the model. The overall procedure of the proposed SSL approach is summarized in Algorithm 1. The following section explains the notations used, the proposed Dual-Head CNN model and its motivation for SSL, the loss functions used for training, the proposed method for selecting highly confident pseudo-labeled data from the unlabeled set, and the overall training procedure.

3.1. Problem setting

Assume that the training set consists of both labeled and unlabeled images. $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{N}$ be the labeled set, where each labeled image x_i is associated with a label $y_i \in [0, C-1]$. Here, C represents the total number of classes, and N represents the total number of labeled images. Similarly, the unlabeled set can be represented as $\mathcal{U} = \{(u_i)\}_{i=1}^{M}$, where M represents the total number of unlabeled images. The label of each unlabeled image u_i is unknown. It is also assumed that the distribution of the number of images from different classes follows a long-tail distribution, implying that the number of images in each class is highly imbalanced. The notations used throughout this manuscript are listed in Table 1.

3.2. Dual-head CNN

When training the CNN model using the commonly employed Cross Entropy (CE) loss, without considering the issue of class imbalance, a significant bias toward the majority classes emerges. Consequently, the model tends to incorrectly classify most images as if they belong to the majority class, while selecting only a few or no data from the minority classes. Subsequent training using these erroneously chosen pseudo-labels adversely affects the model's generalization ability, a phenomenon known as "confirmation bias" (Wang, Wu, Lian, & Yu, 2022).

On the contrary, training the model with a class-balanced CE loss, which assigns equal importance to all classes, results in more accurate pseudo-label selection. Pseudo-labels from even the minority classes are chosen in this scenario, proving highly beneficial for the subsequent semi-supervised training of the model. However, it is important to note that training a model with a balanced CE loss might not be suitable if the primary objective is to maximize the overall prediction accuracy.

Therefore, in our proposed approach, a Dual-Head CNN model is introduced to effectively tackle the class imbalance issue, as depicted in Fig. 2. This model comprises two classifier heads, denoted as H and H_W, which share the same backbone network. The pseudo-labels selected from both heads are utilized for semi-supervised learning (SSL). Classifier head H_W is trained using the balanced CE loss to facilitate the selection of pseudo-labels from minority classes. Meanwhile, classifier head H is trained to maximize overall prediction accuracy. During testing, the classification head H can be employed to make predictions that optimizes overall accuracy values.

3.2.1. Classifier head H

The classifier head, H, is focused on classifying images into a set of predefined classes without considering their imbalanced nature. i.e., H is optimized using the CE loss function as follows:

$$\mathcal{L}^{\alpha} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} \mathbb{1}_{y_i=c} \log p(y_i = c|\alpha(x_i), H) \tag{1}$$

where, $\mathbb{1}_{y_i=c}$ is the indicator function which returns 1 if $y_i = c$, and returns 0 otherwise, $\alpha(x_i)$ is the augmented version (e.g., by applying rotation) of the image x_i and $p(y_i = c|\alpha(x_i), H)$ is the predicted probability of the augmented image $\alpha(x_i)$ belonging to the class c by the classifier head H.

The CE loss has gained widespread adoption in SSL, exemplified by its use in prominent approaches like FixMatch (Sohn et al., 2020). It aligns well with the goal of maximizing the likelihood of the true labels given the model's predictions. The choice of the CE loss function in SSL is rooted in its effectiveness for multiclass classification tasks and its suitability for generating pseudo-labels from unlabeled data. By optimizing this loss function on a combination of labeled and pseudo-labeled data, SSL aims to enhance the model's generalization capabilities and overall performance.

However, as explained earlier, this loss function does not account for the imbalanced nature of the classes. Consequently, its predictions may be biased towards the classes which contain majority of the data. This loss function would be more appropriate if the objective is to maximize the overall accuracy or the F1-score. When this classifier is used in SSL to select high confident pseudo-labels, the majority of the selected data would be from the majority classes, resulting in few or no data selected from the minority classes.

3.2.2. Classifier head H_W

The classifier head H_W is designed to address the imbalanced nature of different classes by applying a class weighted CE loss. This loss assigns different weights to each class, giving equal importance to all classes and resulting in an unbiased classifier.

$$\mathcal{L}_W^{\alpha} = -\frac{1}{N'}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c \, \mathbb{1}_{y_i=c} \log p(y_i = c|\alpha(x_i), H_W) \tag{2}$$

where, w_c is the weight applied to class c, and N' is a normalization factor which can be given as $N' = \sum_{i=1}^{N}\sum_{c}^{C} w_c \mathbb{1}_{y_i=c}$. The class weight of the class c, w_c, is the normalized version of the inverse frequency of that class and it can be calculated as:

$$w_c = \frac{\frac{1}{N_c}}{\sum_{l=1}^{C}\frac{1}{N_l}} \tag{3}$$

where, N_c is the number of images labeled as class c and can be given as $N_c = \sum_{i=1}^{N} \mathbb{1}_{y_i=c}$, and in addition, $N = \sum_{c=1}^{C} N_c$.

Eq. (3) assigns greater weight values to the minority classes and smaller weight values to the majority classes. Consequently, by incorporating these weights into Eq. (2), it ensures that each class receives equal importance during model training.

The class-weighted loss function (Eq. (2)) is particularly well-suited for maximizing the mean per-class accuracy (MCA) values, as demonstrated in Section 4.2.1 through experimental validation. This is because it assigns equal importance to each class. Furthermore, this loss function proves to be more compatible with SSL settings, as it aids in selecting highly confident pseudo-labels, even from minority classes, thanks to its equitable treatment of all classes.

3.2.3. Joint training of H and H_W

Using the weighted CE loss alone to train the CNN can maximize MCA values and assist SSL, however, it may result in low overall accuracy values and F1 scores (refer Section 4.2.1 for experimental validation). To address this issue, the dual-head CNN architecture is employed to enhance both SSL and F1 scores.

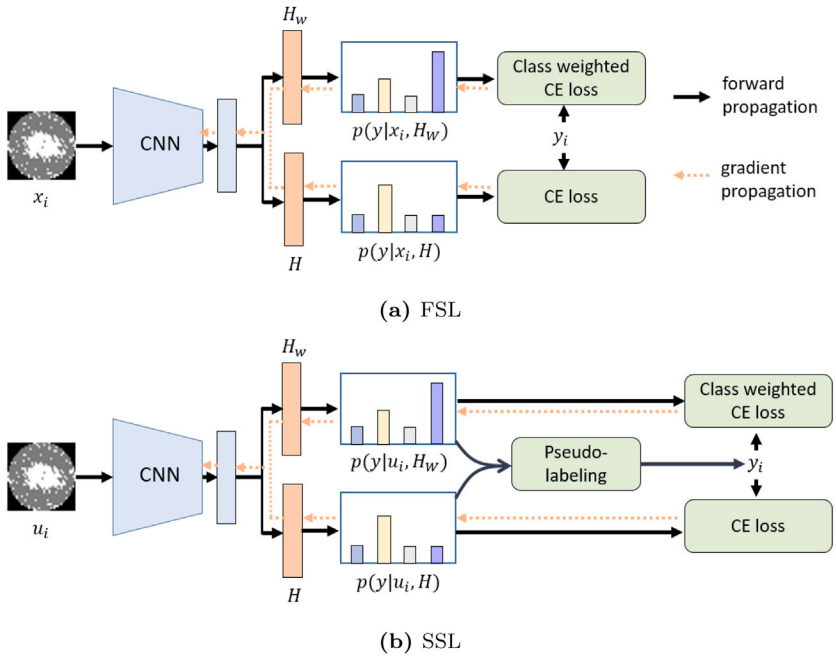


Fig. 2. The proposed Dual-Head CNN architecture under (a) FSL and (b) SSL. H and H_W are classifier heads which maximize the overall accuracy, and the per-class accuracies respectively.

Two types of augmentations, weak (α) and strong ($\mathcal{A}$), are employed as they have been widely used by the recent SSL approaches, e.g., FixMatch (Sohn et al., 2020). The classifier heads, H and H_W, are jointly trained using the following loss function.

$$\mathcal{L}_T = \lambda\left(\mathcal{L}^{\alpha} + \mathcal{L}^{\mathcal{A}}\right) + \lambda_W\left(\mathcal{L}^{\alpha}_W + \mathcal{L}^{\mathcal{A}}_W\right) \tag{4}$$

where, $\mathcal{L}^{\alpha}$ represents the CE loss which uses no class weighting and uses weak augmentation using the classifier head H (Eq. (1)). On the other hand, $\mathcal{L}^{\mathcal{A}}$ is the CE loss which uses strong augmentations. Similarly, $\mathcal{L}^{\alpha}_W$ is the weighted CE loss with weak augmentation and based on the output of the classifier head H_W (Eq. (2)). λ and λ_W are the trade-off parameters.

Weak augmentation typically generates clear, easily classifiable images. In contrast, strong augmentation generates images that are more challenging to classify. Therefore, during training, both weak and strong augmentations are taken into consideration.

3.3. Pseudo-labeling based consistency regularization

The proposed SSL approach makes use of a pseudo-labeling based consistency regularization approach as it has been widely used by the recent state-of-the-art SSL methods such as FixMatch (Sohn et al., 2020) and FlexMatch (Zhang et al., 2021). Consistency regularization enforces that the predictions of the perturbed versions of an input should be similar to each other. In pseudo labeling based consistency regularization, first the pseudo label, $\hat{y}_i$, of an unlabeled image, u_i, is generated based on the weakly augmented version of that image as follows:

$$\hat{y}_i = \arg\max(p(y|\alpha(u_i), H')) \tag{5}$$

That is, the pseudo-label is identified as the class which corresponds to the maximum value in $p(y|\alpha(u_i), H')$. Here, H' could be either H or H_W. This pseudo label is then considered as the true label for the strongly augmented version of that image when updating the CNN model which gives a form of consistency regularization. When updating the model only the high confident pseudo-labels are considered, i.e., pseudo-labels with $\max(p(y|\alpha(u_i))) > \tau$, where, τ is a threshold value.

3.4. Selection of high confident pseudo-labels based on Dual-Head CNN

At each SSL iteration, a set of high confident pseudo-labels are selected from the unlabeled training set and added with the original labeled set to augment the labeled training set. This section explain how a set of high confident pseudo-labels are generated from the Dual-Head CNN.

High-confidence pseudo-labels can be generated from the output of the classifier H. However, as explained earlier, this classifier is biased, and as a result, it may select a large number of pseudo-labeled data points from the majority classes while selecting none from the minority class. In addition, as it is a biased classifier the selected pseudo-labels may be too noisy and may hurt the subsequent training.

On the other hand, the high-confident predictions can be selected based on the output from the class weighted classifier head, H_W, as it is trained to give equal importance to each class. The proposed approach uses both H and H_W to select the high-confident pseudo-labels.

3.4.1. Pseudo-label selection based only on H_W

Based on H_W, an unlabeled image can be identified as high-confident prediction if the corresponding prediction exceeds a threshold τ. i.e., $q_i > \tau$, where, $q_i = \max(p(y|\alpha(u_i), H_W))$. The threshold, τ, is set as the average value of the prediction probabilities of the unlabeled data, i.e., $\tau = \sum_{i=1}^{M} q_i$

3.4.2. Pseudo-label selection based on both H and H_W

Additional predictions can be selected based on considering the predictions made by both classifiers H and H_W. If both of these classifiers H and H_W agree with the pseudo-label of u_i with high confident, u_i can also be added with the labeled data as a high confident prediction. Let $p(y|\alpha(u_i), H)$ and $p(y|\alpha(u_i), H_W)$ represent the probability values obtained from the classifier heads H and H_W respectively for the unlabeled image u_i, and $\hat{y}_i$ and $\hat{y}_i^w$ respectively represent the calculated pseudo-labels from these probabilities. An unlabeled image is considered high-confident if:

$$(\hat{y}_i = \hat{y}_i^w) \wedge \left(\frac{q_i + q_i^w}{2} > \tau'\right) \tag{6}$$

where, $q_i = \max(p(y|\alpha(u_i), H))$, $q_i^w = \max(p(y|\alpha(u_i), H_W))$, $\tau' = \frac{1}{2M}\sum_{i=1}^{M}\left(q_i + q_i^w\right)$ and $\wedge$ is the *logical-and* operator.

Algorithm 1: Dual-Head CNN for SSL

Input: Labeled set D_L, unlabeled set D_U, Dual-Head CNN f_θ with the classifier heads H and H_W
Output: f_θ
Procedure:
for *t=1 to T* **do**
  Train f_θ on D_L using the loss function defined by Eq. (4)
  Generate pseudo-labels for the weakly augmented version of D_U using H and H_W (Section 3.3)
  Form D'_L by selecting high confident pseudo-labels (Section 3.4)
  Expand the labeled set by $D_L = D_L \cup D'_L$
  Update the class weights using Eq. (3)

3.5. The overall procedure for SSL based on the dual-head CNN

The overall procedure of the proposed SSL approach is summarized in Algorithm 1. Initially, the Dual-Head CNN is trained using the available labeled data and with the loss function defined by Eq. (4). Then this trained CNN is used to predict the pseudo-labels of each weakly augmented unlabeled data point u_i separately using the classifier heads H and H_W as explained in Section 3.3. The high confident predictions are then selected together with their pseudo-labels as explained in Section 3.4 and the labeled training set is expanded with the addition of this high confident pseudo labeled data. The class weights defined by Eq. (3) are updated using this expanded labeled data and the CNN model is then retrained using this expanded labeled data. The above steps were iterated for a few times (i.e., $T = 4$).

Note that weak augmentations are used when calculating pseudo-labels, as they typically represent clear images. Consequently, they are often far from the decision boundaries that distinguish different classes, resulting in high-confidence predictions. However, during model training, strongly augmented images play a crucial role. These images are more challenging and tend to be closer to the decision boundaries, which aids in refining those boundaries. It is not advisable to employ strongly augmented images for calculating pseudo-labels because they are situated near the decision boundaries, leading to potentially noisy and less confident predictions. Hence, pseudo-labels are computed using weakly augmented images and then utilized as if they were the actual labels for the strongly augmented images during model training, thereby enforcing consistency regularization and ensures more stable training.

4. Experiments, results and discussion

In this section, the dataset, experimental settings are summarized, and the results are reported with discussion.

4.1. Dataset

The proposed method was evaluated using the publicly accessible WBM dataset, known as the *WM-811K*[2] dataset. This dataset comprises 811,457 images collected from a real-world wafer fabrication process. However, labels are available for only a subset of this dataset, consisting of 172,950 images. Each image in this subset is annotated into one of the nine classes, with each class containing images ranging from 149 to 147,431, resulting in a highly imbalanced class distribution as shown in Table 2.

[2] https://www.kaggle.com/datasets/qingyi/wm811k-wafer-map.

4.2. Training configurations

ImageNet pretrained CNN models are finetuned with a Stochastic Gradient Descent optimizer with a cosine learning rate decay (Loshchilov & Hutter, 2016), which sets the learning rate at the iteration k to $\eta \cos(\frac{7\pi k}{16K})$ where η is the initial learning rate, and K is the total number of iterations. Section 4.2.1 reports the effect of the learning rate k with different CNN architectures and settings. In all the reported experiments K is set to $K = 80$. Following (Manivannan, 2022) a dropout rate of 0.5 was used before the classification layer of each CNN architecture and a label smoothing factor of 0.1 was used to reduce overfitting. Batch size and the learning rate were set to 128 and 0.05, respectively, unless otherwise specified. Variants of Resnet (He, Zhang, Ren, & Sun, 2016) and Densenet-121 (Huang, Liu, van der Maaten, & Weinberger, 2017) CNN architectures were used for comparison as they were widely used for image classification tasks.

All the images were resized to 96×96 pixels, and were normalized to have zero mean and unit standard deviation before feeding them to the CNN. The normalized images were used as the weakly augmented images. Random rotations, random flips (horizontal and vertical), and cutout were used as the strong augmentation. In the cutout augmentation, a random square patch of size $s \times s$ where $s \in [1, 20]$ pixels was selected at a random location within the image, and the pixels within the patch were replaced with zeros. The resized normalized images were used for testing without any other augmentations.

The evaluation measures for the test set were reported as the mean and standard deviations of three repeated experiments. The macro F1 score and the mean of per-class accuracies (MCA) were used as evaluation measures to provide a balanced evaluation of the model's ability to correctly classify instances across all classes. To ensure a fair comparison, the experimental settings used in Manivannan (2022) and Kahng and Kim (2021) were followed, whereby in each experimental run, 20% of the labeled data was randomly selected as the test set, and $p\%$, of the remaining labeled data was randomly sampled as the labeled training set. The value of p was varied across $1\%, 2\%, 5\%, 10\%, 25\%$, and 100%. Table 3 reports the number of images under each class when varying the value of p. Additionally, 200,000 images were randomly selected from the rest of the data as the unlabeled training set for training the proposed SSL approach.

4.2.1. Investigation of FSL under different settings

This section aims to investigate (1) the suitable CNN architecture and the appropriate learning rate for each architecture, (2) the effect of the class-weighted loss function for the classification of WBM, (3) the effect of the amount of labeled training data on the performance, (4) the effect of data augmentation, and (5) the effect of batch size. Two variants of ResNet (ResNet-10 and ResNet-18) and DenseNet-121 were considered for comparison as Resnet (He et al., 2016) and Densenet (Huang et al., 2017) were widely used for image classification. The results are reported in Tables 4 and 5 respectively when 2% and 10% of labeled training data is considered for training the model with and without class weights. Note that a single head CNN model is used in all the experiments reported in this section.

The following are the main observations from the experiments: (1) Simpler model, Resnet-10, performs better for WBM classification particularly when the training data is limited compared to a more complex model. (2) Applying a class-weighted CE loss lead to better MCA as WBM is an imbalanced classification problem; however, a better F1-score was obtained when no class weights were used. (3) Increasing the size of labeled training data leads to better classification performance regardless of the CNN architecture as more labeled data helps to learn the CNN weights better. (4) More data augmentations leads to better performance as it improves the model's ability to generalize well. (5) Small batch sizes (64,128, and 256) leads to better performance than a larger batch size (i.e., 512) as smaller batch sizes improve the

Table 2
Detail of the WM-811K dataset.

Class name	Labeled									Unlabeled
	Near-Full	Donut	Random	Scratch	Loc	Center	Edge-Loc	Edge-Ring	None	
No. of images	149	555	886	1193	3593	4294	5189	9680	147,431	638,507

Table 3
Number of training (labeled) and testing images under different settings.

Class	Training set						Test set
	$p = 1\%$	$p = 2\%$	$p = 5\%$	$p = 10\%$	$p = 25\%$	$p = 100\%$	
Near-Full	1	2	6	12	30	119	30
Donut	4	9	22	44	111	444	111
Random	7	14	35	69	173	693	173
Scratch	10	19	48	95	238	954	239
Loc	29	57	144	288	719	2,875	718
Center	34	69	172	344	859	3,435	859
Edge-Loc	42	83	207	415	1,038	4,151	1,038
Edge-Ring	77	155	387	774	1,936	7,744	1,936
None	1179	2359	5897	11,795	29,486	117,945	29,486
Total	1383	2767	6918	13,836	34,590	138,360	34,590

Table 4
FSL baseline: Classification performance of different CNN architectures with 2% of labeled training data.

lr	ResNet-10		ResNet-18		DenseNet-121	
	MCA	F1	MCA	F1	MCA	F1
Without class weights						
0.001	48.1 ± 0.6	.484 ± .008	62.1 ± 2.7	.653 ± .038	**66.8 ± 1.4**	**.713 ± .015**
0.005	67.1 ± 0.7	.710 ± .012	69.1 ± 1.7	.729 ± .024	62.8 ± 9.3	.653 ± .103
0.01	73.4 ± 3.7	.778 ± .039	**72.1 ± 2.4**	**.759 ± .034**	50.7 ± 3.2	.532 ± .032
0.05	**76.4 ± 1.5**	**.811 ± .017**	11.1 ± 0.0	.102 ± .000	44.8 ± 4.4	.485 ± .049
0.1	**76.4 ± 0.7**	**.811 ± .010**	11.1 ± 0.0	.102 ± .000	44.9 ± 3.2	.489 ± .032
0.5	35.3 ± 17.4	.350 ± .180	11.1 ± 0.0	.102 ± .000	43.1 ± 3.3	.475 ± .043
With class weights						
0.001	82.8 ± 1.5	.761 ± .018	**81.9 ± 0.8**	**.756 ± .019**	**75.5 ± 2.3**	**.664 ± .020**
0.005	**82.9 ± 1.0**	**.776 ± .021**	61.9 ± 5.9	.518 ± .045	71.4 ± 2.5	.571 ± .029
0.01	80.7 ± 0.9	.734 ± .028	42.7 ± 1.8	.377 ± .024	62.8 ± 4.4	.471 ± .071
0.05	46.5 ± 25.1	.377 ± .258	11.1 ± 0.0	.102 ± .000	42.4 ± 5.7	.252 ± .059
0.1	22.3 ± 15.8	.187 ± .120	11.1 ± 0.0	.102 ± .000	30.9 ± 8.1	.144 ± .062

Table 5
FSL baseline: Classification performance of different CNN architectures with 10% of labeled training data.

lr	ResNet-10		ResNet-18		DenseNet-121	
	MCA	F1	MCA	F1	MCA	F1
Without class weights						
0.0005	65.1 ± 4.4	.675 ± .037	80.3 ± 0.1	.831 ± .003	76.8 ± 1.1	.798 ± .002
0.001	74.4 ± 1.2	.759 ± .009	83.6 ± 1.1	.860 ± .008	**77.9 ± 2.2**	**.817 ± .017**
0.005	85.4 ± 1.5	.877 ± .008	**86.4 ± 1.2**	**.878 ± .005**	73.9 ± 2.9	.779 ± .025
0.01	**86.5 ± 1.7**	**.885 ± .010**	**86.4 ± 0.8**	**.878 ± .006**	58.4 ± 4.3	.619 ± .047
0.05	85.7 ± 1.7	.876 ± .008	11.1 ± 0.0	.102 ± .000	72.4 ± 5.6	.748 ± .056
0.1	85.8 ± 0.9	.875 ± .003	11.1 ± 0.0	.102 ± .000	78.0 ± 1.6	.800 ± .009
With class weights						
0.005	90.4 ± 0.9	.812 ± .006	89.9 ± 0.4	.829 ± .004	**87.8 ± 0.3**	**0.788 ± 0.011**
0.001	91.0 ± 0.7	.831 ± .008	**90.3 ± 0.9**	**.832 ± .005**	87.2 ± 1.2	0.782 ± 0.019
0.005	**91.0 ± 1.2**	**.851 ± .004**	87.2 ± 4.3	.763 ± .026	81.5 ± 3.9	0.691 ± 0.040
0.01	90.5 ± 0.9	.841 ± .011	50.6 ± 28.9	.424 ± .236	76.3 ± 4.7	0.622 ± 0.036
0.05	51.9 ± 28.8	.441 ± .240	11.1 ± 0.0	.102 ± .000	51.2 ± 20.9	0.403 ± 0.207
0.1	16.0 ± 6.9	.127 ± .035	11.1 ± 0.0	.102 ± .000	61.1 ± 19.4	0.472 ± 0.204

generalization ability of the network (Masters & Luschi, 2018). In the following these findings are explained in detail.

Small model is better: Tables 4 and 5 demonstrate that when limited amounts of data are used for training, Resnet-10 significantly outperforms the other models. This is due to Resnet-10 having relatively fewer parameters to train than the other considered models, allowing these parameters to be learned with a smaller amount of data. However,

Table 6
Effect of different data augmentation with FSL when $p = 5\%$.

Flip	Rotation	Cutout	MCA	F1
✗	✗	✗	73.3 ± 0.4	.764 ± .006
✓	✗	✗	79.2 ± 2.0	.817 ± .011
✓	✓	✗	**83.4 ± 1.9**	**.853 ± .006**
✓	✓	✓	**83.4 ± 1.4**	**.855 ± .005**

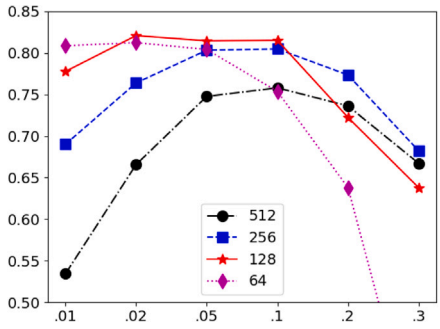


Fig. 3. Effect of batch size: Learning rate (x-axis) vs. F1-scores (y-axis) for different batch sizes with unweighted CE loss.

experiments revealed that further reducing the depth of the Resnet-10 model resulted in a significant drop in performance, and thus these results were not reported in Table 4.

CE loss vs. CE loss with class weights: It can be observed from Tables 4 and 5 that using a class-weighted CE loss (corresponds to $\lambda_W = 1$ and $\lambda = 0$ in Eq. (4)) leads to better MCA scores compared to using an unweighted CE loss (corresponds to $\lambda_W = 0$ and $\lambda = 1$ in Eq. (4)), regardless of the CNN architecture used. This is because the class-weighted CE loss assigns equal importance to each class, maximizing the MCA metric. However, using CE loss without weights results in better F1-scores, indicating that the macro F1-score is influenced by the accuracy of the majority class's prediction.

Better performance with more labels: When increasing the labeled training data classification performance improves significantly regardless of the CNN architecture. For example, when Resnet-10 architecture is considered, an improvement of ∼ 8% in MCA was observed (from 82.9% to 91.0%) when increasing the amount of training data from $p = 2\%$ to $p = 10\%$ (with the class weighted CE loss).

Better results with data augmentation: Table 6 reports the classification performances when different data augmentations are considered with the unweighted CE loss. An improvement of 10% in MCA was obtained when applying all the augmentations (random flipping, rotation and cutout) compared to using only the original images without any augmentations. This improvement can be attributed to the fact that augmentations increase the size of the training set and hence, improve the generalization ability of the network.

Effect of batch size: Fig. 3 illustrates the F1-scores for different batch sizes and learning rates. Batch sizes of 64, 128, and 256 demonstrate nearly identical performances (albeit with varying learning rates) when contrasted with the performance achieved using a batch size of 512. The larger batch size of 512 results in inferior classification performance. This is because the generalization ability of the network is improved by smaller batch sizes (Masters & Luschi, 2018).

Compared to the work of Manivannan (2022) this work reports improved performance by the FSL baseline due to the careful selection

Table 7
Classification performance of the proposed Dual-Head CNN architecture for different percentage of labeled training data with FSL and SSL.

p	FSL/SSL	H_W		H		Both	
		MCA	F1	MCA	F1	MCA	F1
1%	FSL	77.0 ± 3.2	.760 ± .037	67.5 ± 3.4	.729 ± .025	73.5 ± 3.1	.770 ± .029
	SSL	82.3 ± 2.7	.781 ± .013	80.2 ± 3.3	.793 ± .025	81.3 ± 3.1	.791 ± .013
2%	FSL	84.7 ± 0.8	.813 ± .011	76.5 ± 0.2	.814 ± .004	81.1 ± 0.6	.829 ± .006
	SSL	87.7 ± 0.5	.812 ± .012	85.9 ± 0.5	.833 ± .013	86.8 ± 0.4	.831 ± .012
5%	FSL	89.4 ± 1.1	.836 ± .003	83.7 ± 1.2	.860 ± .004	86.7 ± 1.1	.861 ± .003
	SSL	90.7 ± 0.5	.827 ± .008	89.3 ± 0.3	.853 ± .007	89.9 ± 0.3	.842 ± .006
10%	FSL	90.3 ± 1.6	.855 ± .005	85.5 ± 2.1	.879 ± .010	87.9 ± 1.9	.877 ± .009
	SSL	91.8 ± 1.3	.846 ± .003	90.3 ± 1.4	.875 ± .005	91.0 ± 1.2	.865 ± .003
25%	FSL	92.0 ± 0.6	.881 ± .007	87.1 ± 1.2	.897 ± .008	89.5 ± 0.8	.896 ± .005
	SSL	93.0 ± 0.2	.867 ± .011	91.3 ± 0.6	.901 ± .008	92.1 ± 0.6	.896 ± .012
100%	FSL	94.1 ± 0.5	.896 ± .003	90.2 ± 0.9	.918 ± .006	92.3 ± 0.7	.915 ± .006
	SSL	94.2 ± 0.5	.893 ± .005	91.7 ± 1.1	.918 ± .008	93.1 ± 1.0	.915 ± .008

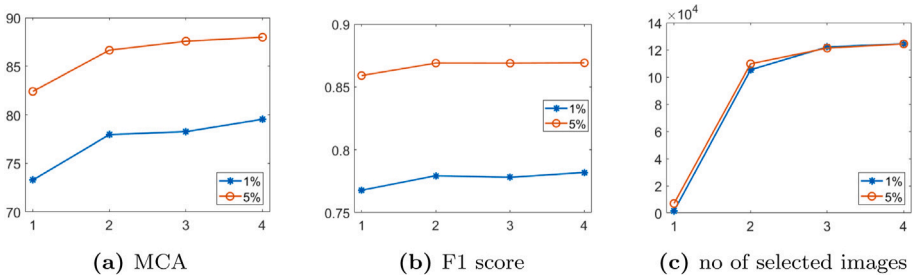


Fig. 4. SSL over iterations: (a) MCA over different iterations (horizontal axis), (b) F1 score over iterations, (c) number of selected images by the SSL over iterations. The first iteration (iteration 1) corresponds to the FSL approach.

of the learning rate and additional data augmentation (rotations and cutout). For example, this work reports a F1-score of .855 (Table 6) compared to the F1-score of 0.760 reported in Manivannan (2022) when 5% of the labeled training data is considered with ResNet-10. Resnet-10 is used in all the below reported experiments as it gives the best classification performance compared to the other architectures considered.

4.2.2. Performance of the proposed dual-head CNN for FSL and SSL

This section examines the classification performance of the proposed dual-head CNN architecture under the FSL and SSL settings and shows that SSL enhances the classification performance significantly compared to FSL.

Table 7 reports the results of the FSL and SSL approaches when different percentages of labeled data are used for training. Here, the results are reported based on the outputs of individual classification heads and their combination, where the combination is obtained by averaging their outputs and then predicting the label based on this average. The results show that H_W provides significantly better MCA compared to H. However, H performs better in terms of F1 score compared to H_W. When the results based on their combination are considered, they provide a reasonable balance between F1 score and MCA. For example, when $p = 5\%$, H_W gives a MCA of 89.4 and a F1 score of .836, and H gives a MCA of 83.7 and a F1 score of .860. On the other hand, the results which was obtained based on the combination of H_W and H gives a MCA of 86.7 and F1 score of .861. SSL improves the classification performance of FSL significantly. For example, when $p = 1\%$ over 5% improvement in MCA, and ~ 0.02 improvement in F1 score was observed from the output of H_W. However, when large amount of labeled data is used for training (e.g., $p = 25\%$) the improvement obtained from SSL is limited as the results started to saturate.

Fig. 4 shows how SSL improves the classification performance over the SSL iterations and the amount of pseudo-labels selected at the end of each iterations.

Based on the experiments reported in Section 4.2.1, the learning rate, λ and λ_W were set to 0.05, 1 and 0.1 respectively for the experiments reported based on the proposed Dual-Head CNN.

4.2.3. Comparison of different methods for pseudo-label selection

Table 8 reports the number of high confident predictions selected for pseudo-labeling from different classes by different approaches together with their Precision and the Recall scores when the system is trained using 2% of labeled data and tested on the test set. The last row of this table reports the total number of pseudo-labels selected and their average Precision and Recall values under different settings.

FixMatch (Sohn et al., 2020) (corresponds the selection based on the classifier head H) selects high confident predictions by applying a fixed threshold value ($\tau = 0.9$) on the predicted probabilities. However, Table 8 shows that out of the total 30,458 images selected as high confidence predictions by FixMatch, the majority (27,465) belonged to the class 'none', with no data selected from the minority class 'near-full'. This indicates the inability of FixMatch in handling imbalanced data and results in bias towards the majority class.

On the other hand, FlexMatch (Zhang et al., 2021), a recently proposed SSL method, addresses this issue by applying per-class adaptive thresholds based on the model's learning status for each class to select high-confident predictions. FlexMatch selects a large number of high confident predictions, including data from the minority classes. However, the selected data, particularly from the minority classes, are inaccurate, with very low precision and recall values (precision of 0.743 for the minority class 'near-full'). Adding this impure data to the labeled data during model training may result in a noisy model with reduced performance.

Similarly, SimiS (Chen et al., 2022), a recent SSL approach, shows improved performance over many SSL approaches, including FixMatch (Sohn et al., 2020) and ReMixMatch (Berthelot et al., 2019), for imbalanced class classification problems in Chen et al. (2022). SimiS uses the difference in class distribution between a particular class and the most frequent class and selects more pseudo-labels from the less frequent classes than the frequent classes. This approach, however, may not be suitable for the extreme imbalance case, as most of the data from the infrequent classes may be selected with noisy pseudo-labels. Table 8 demonstrates that this approach leads to the worst precision and recall values.

In contrast, the proposed approach efficiently handles the class imbalance problem, enabling the selection of highly accurate labeled

Table 8
The number of high confident predictions selected from each class for pseudo-labeling by different approaches and the Precision and Recall scores of the selected pseudo-labels.

Class name	Based on H_W			Based on H_W & H			FixMatch Sohn et al. (2020)			FlexMatch Zhang et al. (2021)			SimiS Chen et al. (2022)		
	#	Prec.	Recall	#	Prec.	Recall	#	Prec.	Recall	#	Prec.	Recall	#	Prec.	Recall
Near-full	13	1.000	0.928	13	1.000	0.928	0	0.000	0.000	35	0.743	0.867	35	0.743	0.867
Donut	70	0.957	0.985	70	0.957	0.985	58	1.000	0.983	120	0.808	0.898	122	0.803	0.883
Random	133	0.872	0.991	133	0.872	0.991	117	0.949	0.982	212	0.741	0.924	248	0.645	0.925
Scratch	71	0.958	0.944	72	0.958	0.945	53	1.000	0.768	158	0.816	0.832	314	0.567	0.745
Loc	236	0.987	0.932	237	0.987	0.925	208	0.990	0.876	554	0.834	0.764	755	0.678	0.713
Center	619	0.982	0.998	624	0.982	0.998	554	0.989	0.989	824	0.929	0.968	943	0.849	0.932
Edge-Loc	460	0.941	0.979	468	0.942	0.978	346	0.974	0.926	921	0.805	0.878	1,282	0.671	0.829
Edge-Ring	1,672	1.000	0.996	1,675	1.000	0.996	1,657	1.000	0.997	1,899	0.974	0.975	1,937	0.962	0.963
None	21,672	0.999	0.998	22,909	0.999	0.998	27,465	0.997	0.999	27,458	0.997	0.993	28,954	0.992	0.974
Total/Avg.	24,946	**0.966**	**0.973**	26,201	**0.966**	**0.972**	30,458	0.878	0.836	32,181	0.850	0.900	34,590	0.768	0.870

-number of confident predictions selected from each class. Prec. - Precision score.

Table 9
Comparison of F1 scores for different percentage of labeled training data with SSL approaches.

Method	1%	2%	5%	10%	25%	100%
Self-supervised representation learning (Kahng & Kim, 2021)	–	–	.815	.839	.864	.897
Ladder networks (Kong & Ni, 2018, 2020)	–	–	.814	.823	.838	.863
SSL (without Ensemble) (Manivannan, 2022)	–	–	.834	.859	.883	.902
SSL Ensemble (Manivannan, 2022)	–	–	.850	.871	.900	.912
Proposed Dual-Head CNN for SSL	**.790**	**.843**	**.861**	**.882**	**.901**	**.918**

Table 10
Comparison with the state-of-the-art approaches.

Method	MCA	F1	Acc
Hand-crafted features + SVM (Wu, Jang, & Chen, 2015)	–	–	94.6
Decision tree ensemble (Piao et al., 2018)	–	–	90.5
Light-weight CNN (Tsai & Lee, 2020)	–	0.800	97.0
MobileNet-V3 (Shin & Yoo, 2023)	–	0.895	98.0
Self-supervised representation learning (Kahng & Kim, 2021)	–	0.897	–
Convolutional Neural Nets (Batool, Shapiai, Fauzi, & Fong, 2020)	–	0.900	98.0
SSL (without Ensemble) (Manivannan, 2022)	–	0.902	–
SSL Ensemble (Manivannan, 2022)	–	0.914	98.2
Proposed Dual-Head CNN for SSL	**91.7**	**0.918**	**98.2**

data, even from the minority classes. The selection based on both H_W and H selects more data than the selection based solely on H_W, without compromising overall precision and recall values. Furthermore, the selected data, even from the minority classes, are highly accurate, demonstrating the effectiveness of the proposed approach for imbalanced class SSL.

4.2.4. Comparison with the state-of-the-art

In this section, a comparison is made between the proposed approach and the current state-of-the-art approaches for WBM classification. The results of the proposed approach show significant improvements over the existing methods and establishes itself as the new state-of-the-art, especially when dealing with limited labeled data during the training phase.

Table 9 presents the results of various SSL approaches proposed for WBM classification and compares them with the proposed approach when different amounts of labeled data are used for training. The proposed approach achieves an F1 score of .79 with only 1% labeled training data, demonstrating its effectiveness in limited data scenarios. Moreover, when 2% of labeled training data is used, the proposed approach outperforms all other single-model-based approaches that use a relatively larger amount of labeled data, i.e., $p = 5\%$. Furthermore, the recently proposed ensemble-based SSL method achieves an F1 score of .850 with 5% labeled training data. In contrast, the proposed approach in this work achieves an F1 score of .861 using a single lightweight model without ensembling.

Table 10 provides a comparison of the proposed approach with other approaches and reports a new state-of-the-art F1 score of .918 when all the labeled training data is used for training.

Note that both Tables 9 and 10 contain gaps in their data due to the absence of results from existing approaches in certain scenarios. For instance, in Table 9, none of the existing methods provide outcomes for training with just 1% and 2% of labeled data. However, considering that the proposed method achieved commendable results even with such limited labeled data, the results have been included for reference.

It is worth noting that none of the existing approaches furnish information on the MCA. Nevertheless, MCA is a valuable metric, particularly for imbalanced data classification scenarios. Consequently, MCA values have been included in Table 10 to facilitate future comparisons and assessments of MCA performance within similar contexts.

5. Conclusion

SSL approaches have gained immense popularity in recent years due to their ability to utilize both labeled and unlabeled data for training deep learning models. However, many existing SSL approaches assume that the training data follows a uniform distribution, where each class consists of approximately an equal number of images. This assumption often results in the failure of these methods to handle highly imbalanced data, where the number of images belonging to one or more classes is significantly less than the others. To address this issue, this work proposes a Dual-Head CNN architecture based SSL approach for imbalanced class WBM defect classification. The proposed approach overcomes the class imbalanced problem by using two classification heads, H and H_W, where H focuses on the overall classification accuracy, and H_W focuses on improving the accuracy of the minority classes by giving equal importance to all the classes. Experimental results demonstrate that the proposed approach outperforms recent SSL learning approaches for imbalanced class WBM defect classification. In particular, the proposed approach achieves a significant improvement over the state-of-the-art ensemble-based SSL approach, with a single lightweight model and without ensembling. Overall, the proposed approach provides a promising solution for addressing the challenges of imbalanced datasets in SSL for deep learning. Future work could focus

on the integration of advanced clustering techniques (e.g., Hu, Chan, Yuan, and Xiong (2019) and Hu, Yang, Tang, He, and Luo (2023)) with the proposed Dual-Head CNN architecture. This integration aims to derive cluster-level labels, thereby reducing the inclusion of noisy labels. Additionally, exploring clustering as a regularization technique to promote smoother decision boundaries and enhance generalization.

CRediT authorship contribution statement

Siyamalan Manivannan: Conceptualization, Methodology, Software, Validation, 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 dataset is available for public use.

References

Adly, F., Alhussein, O., Yoo, P. D., Al-Hammadi, Y., Taha, K., Muhaidat, S., et al. (2015). Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps. *IEEE Transactions on Industrial Informatics*, *11*(6), 1267–1276.

Adly, F., Yoo, P. D., Muhaidat, S., Al-Hammadi, Y., Lee, U., & Ismail, M. (2015). Randomized general regression network for identification of defect patterns in semiconductor wafer maps. *IEEE Transactions on Semiconductor Manufacturing*, *28*(2), 145–152.

Batool, U., Shapiai, M. I., Fauzi, H., & Fong, J. X. (2020). Convolutional neural network for imbalanced data classification of silicon wafer defects. In *IEEE international colloquium on signal processing and its applications* (pp. 230–235).

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., et al. (2019). ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv:1911.09785.

Chen, H., Fan, Y., Wang, Y., Wang, J., Schiele, B., Xie, X., et al. (2022). An embarrassingly simple baseline for imbalanced semi-supervised learning. arXiv preprint arXiv:2211.11086.

Chen, S., Liu, M., Hou, X., Zhu, Z., Huang, Z., & Wang, T. (2023). Wafer map defect pattern detection method based on improved attention mechanism. *Expert Systems with Applications*, *230*, Article 120544.

Chien, C.-F., Hsu, S.-C., & Chen, Y.-J. (2013). A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence. *International Journal of Production Research*, *51*(8), 2324–2338.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hsu, C.-Y., & Chien, J.-C. (2020). Ensemble convolutional neural networks with weighted majority for wafer bin map pattern classification. *Journal of Intelligent Manufacturing*, 1–14.

Hu, L., Chan, K. C., Yuan, X., & Xiong, S. (2019). A variational Bayesian framework for cluster analysis in a complex network. *IEEE Transactions on Knowledge and Data Engineering*, *32*(11), 2115–2128.

Hu, H., He, C., & Li, P. (2021). Semi-supervised wafer map pattern recognition using domain-specific data augmentation and contrastive learning. In *2021 IEEE international test conference (ITC)* (pp. 113–122). IEEE.

Hu, L., Yang, Y., Tang, Z., He, Y., & Luo, X. (2023). FCAN-MOPSO: An improved fuzzy-based graph clustering algorithm for complex networks with multi-objective particle swarm optimization. *IEEE Transactions on Fuzzy Systems*, 1–16.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE conference on computer vision and pattern recognition* (pp. 2261–2269).

Kahng, H., & Kim, S. B. (2021). Self-supervised representation learning for wafer bin map defect pattern classification. *IEEE Transactions on Semiconductor Manufacturing*, *34*(1), 74–86.

Kong, Y., & Ni, D. (2018). Semi-supervised classification of wafer map based on ladder network. In *IEEE international conference on solid-state and integrated circuit technology* (pp. 1–4).

Kong, Y., & Ni, D. (2020). A semi-supervised and incremental modeling framework for wafer map classification. *IEEE Transactions on Semiconductor Manufacturing*, *33*(1), 62–71.

Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983.

Manivannan, S. (2022). An ensemble-based deep semi-supervised learning for the classification of wafer bin maps defect patterns. *Computers & Industrial Engineering*, *172*, Article 108614.

Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. CoRR abs/1804.07612. arXiv:1804.07612.

Misra, S., Kim, D., Kim, J., Shin, W., & Kim, C. (2022). A voting-based ensemble feature network for semiconductor wafer defect classification. *Scientific Reports*, *12*(1), 16254.

Piao, M., & Jin, C. H. (2022). CNN and ensemble learning based wafer map failure pattern recognition based on local property based features. *Journal of Intelligent Manufacturing*, 1–23.

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 Transactions on Semiconductor Manufacturing*, *31*(2), 250–257.

Shin, W., Kahng, H., & Kim, S. B. (2022). Mixup-based classification of mixed-type defect patterns in wafer bin maps. *Computers & Industrial Engineering*, *167*, Article 107996.

Shin, E., & Yoo, C. D. (2023). Efficient convolutional neural networks for semiconductor wafer bin map classification. *Sensors*, *23*(4), 1926.

Sohn, K., Berthelot, D., Li, C., Zhang, Z., Carlini, N., Cubuk, E. D., et al. (2020). FixMatch: Simplifying semi-supervised learning with consistency and confidence. CoRR abs/2001.07685. arXiv:2001.07685.

Tsai, T.-H., & Lee, Y.-C. (2020). A light-weight neural network for wafer map classification based on data augmentation. *IEEE Transactions on Semiconductor Manufacturing*, *33*(4), 663–672.

Wang, X., Wu, Z., Lian, L., & Yu, S. X. (2022). Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14647–14657).

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 Transactions on Semiconductor Manufacturing*, *28*(1), 1–12.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). MixUp: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., et al. (2021). Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, *34*, 18408–18419.