Journal Paper

by Lavanya R

Submission date: 18-Apr-2024 12:15PM (UTC+0530)
Submission ID: 2353642257
File name: journal_paper.doc (1.17M)
Word count: 8576
Character count: 49895

ENSEMBLE LEARNING FOR WAFER CLASSIFICATION WITH MINIMUM PRE-LABELLED DATA

S. Rajesh[1], R. Lavanya[2]

[1,2]Department of Information Technology, Mepco Schlenk Engineering college, Sivakasi, Tamil Nādu.

***Abstract*— Data quality during training is crucial for building accurate and reliable machine learning and deep learning models. Obtaining annotations relies heavily on the expertise of domain professionals. Prioritizing manual data labelling is crucial for capturing expert knowledge in machine learning, despite its laborious and time-intensive nature in supervised learning. However, Manual annotation is susceptible to distractions, leading to errors and mislabelling and posing significant disadvantages for data labelling processes in supervised learning. This paper elaborates on the pattern recognition of wafer defect map and the ultimate goal of the paper is to train the convolutional neural network. This deep learning model exhibits confidence in its predictions while requiring minimal pre-labelled data for accurately classifying wafer patterns. It is also capable of pseudo-labelling the wafer maps. For this process, deep model is used to perform the proposed method which involves a series of CNN layers. The paper presents hybrid ensembling as a distinctive method to improve wafer classification efficiency. This approach aims to minimize the need for manual annotation compared to existing methodologies.**

Index Terms— **Deep learning, wafer defect map, data labelling, pseudo labelling, pre-labelled data, hybrid ensemble.**

1. Introduction

The semiconductor industry has experienced significant growth over the years, driven by advancements in technology, increasing demand for electronic devices, and the proliferation of emerging applications such as artificial intelligence. Over the years, the semiconductor industry has strived to uphold Moore's Law by continually shrinking the size of transistors and increasing their density on microchips, which has led to advancements in computing performance, energy efficiency, and cost reduction. The concept of multiple layers of active components is increasing in the new era. It refers to the integration of different semiconductor devices or components within a single chip or package, often referred to as system-on-chip (SoC) or multi-chip module (MCM) designs. It achieves space efficiency, improved performance, enhanced functionality and so on. Inline defect detection is a critical aspect of semiconductor manufacturing, as it allows manufacturers to identify and address issues promptly, preventing further propagation of defects and minimizing the impact on yield and product quality. Inline defect detection systems may produce false positives or false negatives. Fine-tuning the system to minimize these errors can be challenging and may require extensive testing and validation [1], [2]. Traditional data analysis methods and platforms struggle to provide practitioners with a comprehensive understanding of the immense volume of data generated online. With the use of modern process control systems, there has been a noticeable shift in the semiconductor manufacturing industry from reactive and predictive analytics to proactive analytics. Due to its widespread use and availability, the WM-811 dataset has become a benchmark dataset for wafer pattern classification by using deep learning-based approaches such as convolutional neural networks. There are nine different kinds of wafer defect patterns among the 811,457 defect maps in the WM-811K dataset [3]. Center, Donut, Edge-Loc, Edge-Ring, Loc, Near-full, Random, Scratch, and None are some of these patterns. Nevertheless, only 172,950 of the 811,457 defect maps in the WM-811K dataset have been carefully annotated by subject matter experts. Furthermore, the amount of sample maps varies significantly throughout defect classes. Wafer maps without labels comprise the remaining 638,507 samples. It's worth noting that recent literature over the past three years has primarily concentrated on the 172,950 labelled maps to enhance overall classification accuracy. For instance, in [4], [5], [6], [7], [8], the training data comprises a range of 31,407 to 124,524 labelled defect maps. However, due to the presence of nine imbalanced classes, effectively performing classification tasks for each defect class poses a significant challenge. In order to offset the impacts of data imbalance, certain research efforts, as [6], have employed data augmentation approaches. With the help of data augmentation, augmentation strategies can boost classification accuracy to a minimum of 95% and as high as 99.73% [6]. It's important to note that the None class is included in these accuracy measures. 85.2% of the initial labeled data for wafer defect patterns are in the None class. When classifying data, machine learning classifiers usually show a bias in favor of the majority class over the minority class. Therefore, an imbalance in the data can have a major effect on how well the minority class performs in categorization. In data labeling, human annotators or automated systems assign labels to data points based on predetermined criteria or guidelines. The quality and accuracy of data labeling directly impact the performance and reliability of machine learning algorithms. Due to advancements in technology in recent years, unlabeled data has become abundant, posing a challenge for data scientists when associating input variables with output variables. To mitigate such imbalances in variable modeling, transitioning to semi-supervised learning offers a promising solution. This approach leverages both labeled and unlabelled data to enhance model training and improve predictive performance. Semi-supervised learning shows great promise to assist in data annotation by taking a small amount of data in combination with a huge amount of unlabeled data, to build equally competitive classifiers as the ones trained by fully supervised learning [9], [10]. Semi-supervised learning utilizes both labeled and unlabeled data to train machine learning models. By incorporating unlabeled data along with the labeled data, semi-supervised learning methods aim to improve model performance and

generalization. The objective of this study is to introduce a novel approach to wafer defect pattern classification, aiming to minimize the requirement for manually labeled data. Using this technology, a deep convolutional neural network (CNN) model is trained with a small number of labeled data examples, allowing the data scientist to automatically annotate unlabeled data. This method greatly minimizes the work needed for data annotation without sacrificing accuracy by guaranteeing the caliber of automatic labeling of wafer fault patterns. About the concept of "pseudo labels" discussed earlier, there is a possibility of erroneous labeling. This study presents an adaptive sequential ensemble strategy that incorporates many shallow CNN classifiers to address this problem. The goal is to recognize and remove any possibly inaccurate labels that the automatic labeling procedure may have produced. The suggested approach seeks to improve the effectiveness of confident deep learning in this way. The study also presents a brand-new method for classifying wafer defects dubbed the hybrid ensemble technique, which emphasizes the employment of several models or ensemble techniques to increase classification accuracy and durability.

Researchers look into ensemble systems that, specifically designed for wafer defect classification tasks, include many classifiers using techniques like bagging, boosting, or stacking. To increase classification performance, various classifier types such as decision trees, support vector machines, convolutional neural networks, and rule-based systems can be combined. The design, implementation, and assessment of hybrid ensembling techniques for more dependable and efficient wafer defect classification systems in semiconductor manufacturing processes are greatly aided by these studies. Wafer defect classification through hybrid ensembling entails a systematic examination of model diversity, ensemble strategies, as well as deep learning integration.

The 9 defect classes are shown in Fig.1. At first, samples in the "None" classes are removed from the original dataset because they comprise the vast majority of wafer defect maps. It is well known that deceptive classification accuracy frequently arises from an unbalanced class distribution. Therefore, in line with best practices in data science, the decision is made to exclude the "None" class from the outset. The remaining 8 defect classes are considered for the experiment.

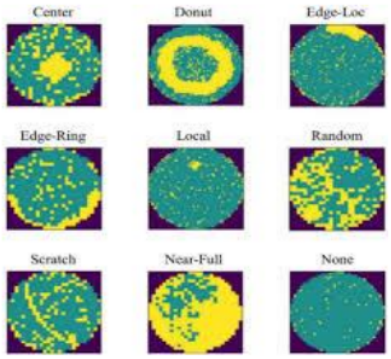


Figure 1: Nine wafer map patterns.

2. Related Work

Wafer defect pattern labeling and recognition using semi-supervised learning is a fascinating field within semiconductor manufacturing. Semi-supervised learning techniques are particularly useful in scenarios where labeled data is scarce or expensive to obtain, which is often the case with wafer defect analysis. With the advent of deep learning, Convolutional Neural Networks have become the dominant approach for wafer defect pattern recognition. CNNs are adept at learning hierarchical representations directly from raw data, making them well-suited for image-based tasks like defect detection. However, deep learning models typically require large amounts of labeled data to achieve good performance. Semi-supervised learning techniques aim to leverage both labeled and unlabeled data during training. This is particularly advantageous in scenarios where obtaining labeled data is costly or time-consuming. models. In the early era, data annotation for tasks such as wafer defect pattern labeling and recognition was primarily performed manually by human experts. This process involved inspecting individual images of wafers and manually labeling instances of defects. However, with the advancements in machine learning, particularly in the fields of generative modeling and unsupervised learning, alternative approaches to data annotation have emerged. Generative models such as Generative Adversarial Networks (GANs) and variational auto-encoders (VAEs) have been utilized for tasks like data augmentation and synthetic data generation. These models can generate realistic-looking synthetic data, which can be used to supplement the limited labeled data available for training. However, it's important to note that while these techniques can assist in data augmentation and generation, they have not entirely replaced the need for labeled data in tasks like wafer defect pattern recognition. Labeled data remains crucial for training supervised models, and semi-supervised learning approaches are employed to leverage both labeled and unlabeled data effectively, thereby optimizing model performance and generalization. A semi-supervised learning method to annotate unlabeled data in the WM-811K dataset was proposed in [10]. The proposed semi-supervised learning method involved training the teacher model through a sequence of feature extraction procedures. This training utilized a Bootstrap aggregating algorithm, where an ensemble classifier consisting of 1,000 decision trees was employed. As part of an ensemble learning process, many models are trained independently on various subsets of the training data, usually acquired using bootstrap sampling. This process is known as bagging or bootstrap aggregating. Each model in the ensemble learns to make predictions based on its subset of the data. Instead of using a single decision tree, the bagging algorithm in this paper employs an ensemble of 1,000 decision trees. Each decision tree in the ensemble is trained on a bootstrapped sample of the training data, ensuring diversity among the trees. This diversity helps reduce over fitting and improves the overall robustness of the model. By using a bagging algorithm with an ensemble of decision trees, the paper [10] likely aims to harness the power of ensemble learning to improve the accuracy and robustness of the defect pattern labeling and recognition system for wafer maps. This paper highlights the importance of accurate and efficient defect detection for ensuring product quality and yield. Detecting defects with precision is not merely a matter of maintaining high standards; it's a cornerstone of reliability and customer satisfaction. It uses WM-811K dataset as mentioned earlier. Cleaning and pre-processing the dataset to remove noise, normalize features, and prepare the data for further analysis is performed. This may involve techniques such as data cleaning, feature scaling, and dimensionality reduction. Then feature extraction is done by extracting relevant features from the raw data to represent each wafer defect pattern. Then, the semi-supervised learning algorithm is used to train a model using both labeled and unlabelled data. This may involve techniques such as self-training, co-training,

graph-based methods, or generative models. Next comes the Training the semi-supervised learning model on the labeled and unlabelled data to learn patterns and relationships between input features and defect labels. This may involve iterative training processes, where the model is updated based on predictions made on unlabeled data [10]. Finally, the performance of the trained model is performed ing validation data or cross-validation techniques. Performance metrics such as accuracy, precision, recall, and F1-score may be calculated to assess the model's ability to correctly label and recognize wafer defect patterns. The aim of the proposed methodology [10] is to improve defect detection and quality assura in semiconductor manufacturing processes. The paper [11] likely serves as a valuable resource for researchers and practitioners seeking to navigate the complexities of small data challenges in the era of big data, offering insights into state-of-the-art methods and best practices for leveraging unlabeled data effectively. The paper [11] likely provides an introduction to unsupervised and semi-supervised learning methods as potential solutions to the sm data challenge. Unsupervised learning techniques aim to identify patterns and structures in unlabeled data, while semi-supervised learning approaches leverage both labeled and unlabeled data to improve model performance. It discusses the eal-world applications and case studies where unsupervised and semi-supervised learning methods have been successfully applied to address small data challenges. Examples could include image recognition, natural language processing, and anomaly detection in cyber security. By harnessing the collective knowledge of a diverse set of decision trees, the paper [10] likely seeks to develop a defect pattern labeling and recognition system that delivers reliable results, thereby supporting the semiconductor manufacturing process in achieving higher yields and improved product quality.

3. Objective

The ultimate objective of the paper is to develop a deep learning framework capable of accurately classifying wafer patterns while minimizing the need for labelled data during the training process. By leveraging self-assured learning techniques, the aim is to create a model that can confidently identify patterns and classify wafer defects even in scenarios where labelled data is scarce or costly to obtain. This approach not only addresses the challenge of limited labelled data but also enhances the efficiency and scalability of defect classification systems in semiconductor manufacturing. By reducing reliance on manual labelling efforts and leveraging the inherent structure within the data, the proposed framework seeks to improve the overall accuracy and reliability of wafer pattern classification, thereby enhancing product quality and manufacturing yield. In semiconductor fabrication, accurately identifying and classifying wafer defects is critical for ensuring product quality and optimizing yield. On the other hand, obtaining labeled data for classification model training can be extremely costly and time-consuming. By presenting a novel deep learning architecture that makes use of self-assured learning methodologies, this research aims to address these issues. Relying less on pre-labelled samples, self-assured learning allows the model to learn efficiently from both labelled and unlabeled data. The model gains confidence in its classification conclusions even in the lack of labelled instances by iteratively updating its predictions and confidence estimates based on the available data. By minimizing the need for extensive manual labelling efforts, the proposed framework not only streamlines the training process but also enhances the scalability and adaptability of wafer pattern classification systems. The model's ability to confidently classify unlabeled data points expands the effective training dataset, improving the robustness and generalization capabilities of the classification model. Moreover, by harnessing the power of deep learning, the framework can capture complex patterns and relationships within the wafer data, enabling more accurate and reliable defect detection. The deep learning technique employed in the paper provides a robust framework for addressing the challenges inherent in wafer pattern classification, particularly when labelled data is scarce. By leveraging deep neural networks, the model autonomously learns complex hierarchical representations of wafer maps, eliminating the need for manual feature engineering and pre-processing. Overall, the deep learning technique offers a powerful and versatile solution for wafer pattern classification, promising accurate and efficient defect detection systems that can significantly improve product quality manufacturing yield in semiconductor production environments. The dataset used in the paper is WM-811K dataset. Wafer datasets are typically used in the field of semiconductor manufacturing for developing and evaluating algorithms for defect detection, classification, and pattern recognition on semiconductor wafers. The percentage of label, no-label, pattern and non-pattern are given in Fig.2. The frequency distribution of different failure types within the pattern wafers, helping to understand the prevalence of each failure type in the semiconductor dataset and is shown in the Fif.3. It provides insights into the dataset's characteristics, which can be crucial for data exploration and analysis in semiconductor manufacturing applications.

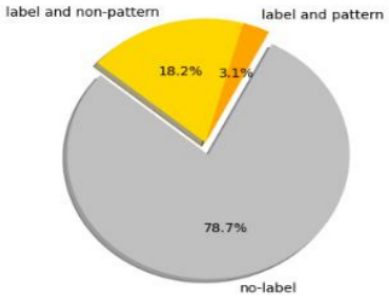


Fig.2. Visualization of the distribution of wafer labels.

4. Process Flow

The generalized process flow can be proceeded as follows.

I. Problem definition and dataset acquisition
II. Data pre-processing and data splitting
III. Model development
IV. Training the model
V. Evaluation

I. Problem definition and dataset acquisition

The Wafer Pattern Classification Minimum Pre-Labeled Data is the main source of contention. the difficulty of correctly categorizing wafer patterns in the absence of tagged data. With wafer patterns playing a crucial role in semiconductor manufacturing processes, the need for robust classification models is paramount. However, manual labelling of wafer patterns is often laborious and time-consuming, leading to a shortage of annotated data for model training. To address this challenge, the paper aims to develop deep learning approaches that can effectively utilize limited labelled data. The overarching goal

is to develop efficient and effective classification models that can accurately identify wafer pattern defects and anomalies, thereby contributing to improved quality control in semiconductor manufacturing processes.

II. Data Pre-processing and data splitting

Initially, the dataset containing wafer pattern images and their corresponding labels is loaded. Subsequently, data augmentation techniques are applied to diversify the dataset, including operations like rotation, flipping, and scaling of images, enhancing the model's ability to generalize. Normalizing pixel values of images to a standardized range aids in stabilizing the training process and accelerating convergence. Furthermore, the dataset is partitioned into training, validation, and test sets to facilitate model evaluation and selection. The distribution of wafer index is shown in Fig.3. Techniques to address class imbalance, if present, are implemented to ensure balanced learning. Through these pre-processing steps, the wafer pattern images are suitably prepared for training deep learning models, ultimately enhancing classification accuracy and robustness.

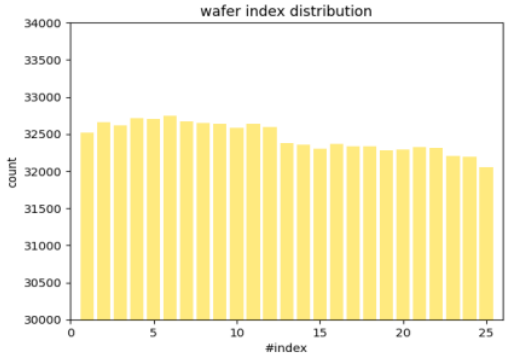


Fig.3. The distribution of wafer index.

III. Model development

The model development encompasses a meticulous process tailored to the intricacies of wafer pattern classification. Initially, appropriate deep learning architectures, such as convolutional neural networks are selected to effectively capture intricate patterns within the wafer images. Hyper parameter tuning becomes pivotal, involving iterative experimentation to optimize parameters like learning rates and batch sizes, ensuring efficient model training. Through careful feature extraction, the chosen architectures are fine-tuned to extract relevant features from the input data, facilitating accurate classification. Techniques like dropout and batch normalization are integrated into the models to prevent over fitting and enhance generalization. Furthermore, exploring ensemble learning methodologies like sequential boosting ensemble enhances model diversity and performance. Training strategies leverage both labelled and unlabeled data, with techniques such as semi-supervised learning utilized to effectively utilize the limited labelled samples.

IV. Training the model

The training of the model involves a two-step approach utilizing pre-trained VGG16 for label generation and Ada-CNN for label discrimination. Initially, the VGG model, pre-trained on a large-scale image dataset, is employed to extract features from the wafer pattern images. These features are then used to generate pseudo-labels for the unlabeled data samples. The structure of vgg-16 is shown in Fig.4. Subsequently, the Ada-CNN model, which incorporates the conventional Ada Boost algorithm adapted for CNNs, serves as a label discriminator. It iteratively learns to identify mislabelled samples by assigning higher weights to difficult-to-classify samples during training. The flowchart for Ada Boost CNN is shown in Fig.5. This iterative boosting process enhances the model's ability to discern accurate labels from the pseudo-labels generated by VGG16. By combining the strengths of both models, the training process aims to leverage the discriminative power of Ada-CNN while benefiting from the feature extraction capabilities of VGG16, ultimately leading to improved performance in wafer pattern classification tasks with limited labelled data.

A. Training the VGG-16 model for label generation:

i. Import the VGG16 model architecture along with pre-trained weights, either from the Keras library or using a pre-trained model file.
ii. To prevent the pre-trained weights from being updated during training, freeze all layers in the VGG16 model.
iii. Add additional layers on top of the VGG16 base to adapt it to your specific classification task. These layers will be responsible for generating labels based on the features extracted by the VGG16 base.
iv. Compile the model with an appropriate optimizer, loss function, and evaluation metric for your specific classification task. The optimizer used here is Adam's optimizer. It is widely used in deep learning due to its effectiveness and efficiency in training neural networks.
v. Train the model on your labelled dataset for multiple epochs. Iterate over the dataset and adjust the learning rate if necessary.
vi. After training, evaluate the model's performance on a separate validation dataset to assess its accuracy and generalization ability.

B. Training the Ada Boost CNN for label discrimination:

i. Define the architecture of the Ada Boost-CNN model, incorporating layers suitable for discriminative learning based on the generated labels. This may involve a combination of convolutional layers, pooling layers, and fully connected layers.
ii. Compile the model with a suitable optimizer, loss function, and evaluation metric. Choose optimizer parameters and hyper parameters based on the nature of the problem and the dataset.
iii. Prepare your dataset for training the model. This involves combining the original images with the labels generated by the VGG16 model.
iv. Train the model on the combined dataset for multiple epochs. Iterate over the dataset.
v. Evaluate the performance of the on a separate validation set to monitor its training progress.

V. Evaluation

A crucial stage in evaluating the effectiveness of the trained models is evaluation. It pertains to the Ada-Boost CNN model for label

discrimination and the VGG16 model for label creation. This entails comparing the generated labels with the actual labels, if available, and computing measures like accuracy, precision, recall, and F1-score. Examining label distributions among various classes and the display of performance via confusion matrices provide additional insight into the model's effectiveness. On the other hand, assessing the Ada-Boost CNN model's label discrimination performance means examining how well it can differentiate between real labels and pseudo labels generated by the VGG16 model. For the predicted labels, metrics including accuracy, precision, recall, and F1-score are computed, and discrimination performance is analyzed employing metrics plots and confusion matrices.

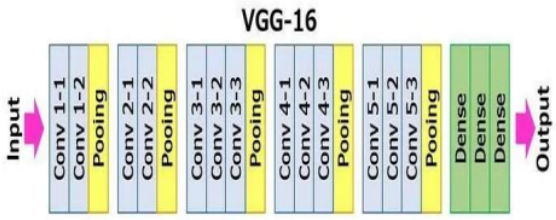


Fig.4. The structure of VGG-16 model.

A comparison between the Ada-CNN and VGG16 models demonstrates the improvements rendered feasible by label discrimination, which facilitates iterations and further model refining.

5. *Need for understanding data distribution*

The wafer map image dataset WM-811K is made up of 811,457 defect maps, each of which shows a different kind of wafer pattern defect. Just 172,950 of these maps have been annotated by domain experts, leaving a sizable part un annotated. Because of this discrepancy in annotation, the dataset is unbalanced, with a smaller fraction of samples labelled than there are in total.

Stakeholders can obtain important insights about the content of the dataset and possible areas of concern in semiconductor production operations by charting the frequency of each failure category. The frequency distribution of wafer maps is shown in Fig. 6. This visualization facilitates the identification of common failure types, which may point to persistent problems that need to be addressed as part of quality improvement initiatives.

Additionally, it aids in resource and action prioritizing by concentrating on resolving the most frequent failures first. Visualizing the frequency of failure types also makes it easier for engineers, data scientists, and decision-makers to communicate and work together. This promotes strategic planning and well-informed decision-making to reduce risks and improve manufacturing processes. All things considered, this visualization is essential for improving comprehension, seeing trends, and motivating action to raise the standard and reliability of semiconductor products. Nine varieties of wafer map patterns are found in the 172,950 images. These are eight different kinds of particular defect patterns and a sort of wafer map known as None, which lacks a defect pattern (Center, Donut,

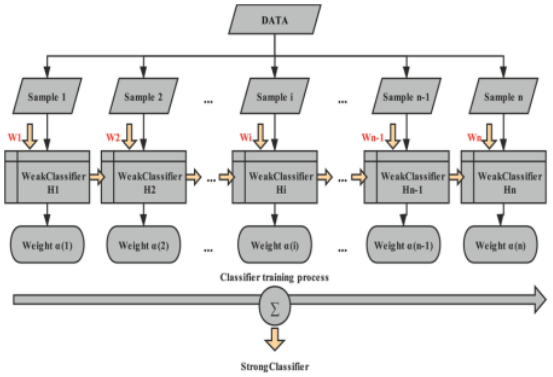


Fig.5. The flowchart for Ada Boost CNN Algorithm.

Edge-Loc, Edge-Ring, Loc, Near-full, Random, and Scratch). Class None makes up 85.2% of the total, and the remaining 8 wafer defect patterns are classified using their distribution. 638,507 wafer maps remain unlabeled till the moment. Wafer maps are composed of three color bands: 0, 1, or 2, where 0 represents no wafer grains, 1 represents non-defect grains, and 2 represents defective grains (highlighted in yellow). It's interesting to see that class None and class Random appear almost exactly the same at first glance. Meanwhile, it seems that a single wafer map has two different fault kinds. To apply the suggested self-assured labeling technique, the class None in the WM-811K dataset needs to be manually removed first. To accurately reflect this option, the following dichotomous classification experiment is conducted between the None Pattern and the remaining 8 defect patterns. Initially, the dataset is split into three groups: 60%, 15%, and 25% of each class are set aside for training, validation, and testing, respectively. Class None improperly classifies less than 5% of the maps corresponding to the remaining 8 defect patterns; all class None maps are perfectly identifiable. Due to their extraordinarily good classification performance, the wafer maps of class None can be easily classified using the traditional 2-class CNN model, allowing for their early elimination.

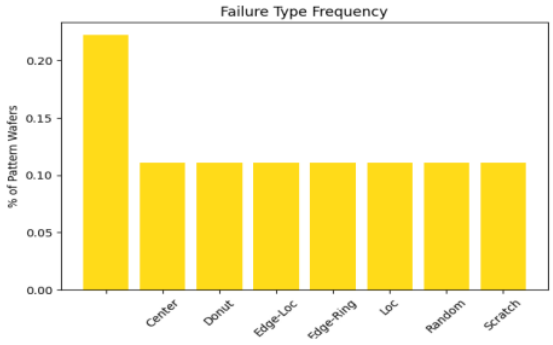


Fig.6. The distribution of failure types across wafer maps.

6. *Proposed model*

Let's examine the two distinct CNN models—one deep and one shallow—that are covered in this section. The shallow CNN model is designed as a label-discriminating neural network, while the deep CNN model is, as its name implies, built as a label-generating neural network. These two models will be expanded in a bit.
Additionally, a preliminary experiment on parameter tuning for the suggested automatic labeling process is done. The following are the stages for the suggested model. Fig.7. shows the proposed model flow.
The subsequent procedures are carried out using a small group of manually labeled samples as their foundation:
(i) Teach the deep CNN how to generate labels.
(ii) Use the trained deep CNN to categorize samples without annotations.
(iii) To eliminate samples that were incorrectly tagged, apply the adaptive sequential ensemble with shallow CNNs.
(iv) Retrain the deep CNN model with the manually labeled data from the first stage and the pseudo-labeled examples that were kept from (iii).

A. VGG16-Based Deep Label Generator (DLG)

There are various reasons why VGG16 could be preferred over Res Network and Efficient Net. First off, the simpler design of VGG16 which consists of stacked convolutional layers which makes it easier to comprehend and apply, which is useful for customization and experimentation. The more complex topologies provided by Res Network and Efficient Net with skip connections and compound scaling, respectively may not always result in better performance for the given application. Furthermore, the track record of success that VGG16 has in picture classification tasks and its compatibility with transfer learning offer a solid basis for model building.
Furthermore, VGG16's somewhat smaller parameter set than Res Network and Efficient Net may provide computational benefits without appreciably compromising performance in situations when computational resources are scarce. 172,950 annotated wafer maps from the WM-811K dataset are used in this example. Samples from nine distinct classes were chosen at random based on three partitions: 60% were used for training (103,770 images), 15% were used for validation (25,943 images), and 25% were used for testing (43,237 images). To be more precise, the number of epochs for the convergence analysis was set to 100, the loss function was calculated using categorical cross-entropy, and the network weights were optimized using the Adam approach. Each iteration round consisted of 200 batches, each with a batch size of 128. To avoid over fitting, the dropout probability value for the models was set to 0.25. The learning rate of 0.0001 was utilized. The combination of the Adam optimizer with cross-entropy loss provides an effective framework for training neural networks in classification problems. This framework allows for flexible learning rates and an appropriate loss function for optimizing model parameters. To train a neural network using the Adam optimizer with cross-entropy loss, the model calculates the cross-entropy loss between the true labels and the projected probabilities in each training iteration, also known as a batch. Next, using back propagation, the Adam optimizer determines the gradients of the loss with respect to the model parameters (weights and biases).
Adam uses adaptive learning rates to update the parameters based on these gradients, accounting for the gradients' first and second moments. In order to reduce the cross-entropy loss and enhance the model's performance on the classification task, this procedure iterates across a number of epochs, progressively changing the parameters. Compared to Efficient Net, VGG16 has a far greater popularity in literature citations, that's why it is used as the DLG in this work. It's also important to emphasize that the primary goal of our paper is to propose a unique deep learning technique for wafer defect pattern classification with the least quantity of pre-labeled data. As a result, the adaptive sequential ensemble will be explained shortly.
Selecting a suitable image size for research is another aspect of the experimental study. In a different experiment, four distinct image sizes—200 x 200, 100 x 100, 75 x 75, and 50 x 50 pixels—were investigated. For the ensuing tests, a picture size of 75 by 75 is used, taking into consideration the computing cost and classification accuracy. The original WM-811K dataset has 632 different image sizes, spanning from 6x21 to 300x202 pixels. In the paper, photos of different sizes in the WM811K dataset were synchronized using nearest-neighbor interpolation, also known as proximal interpolation.

B. Ada-CNN as a Shallow Label Discriminator (SLD)

In this work, the DLG model via VGG16 is used to create wafer-map labels. The deep CNN model is limited to the training set of examples it can learn from. It is possible that the test samples that have not been inspected will not have accurate labels produced by the trained deep CNN model. A specific sequential ensemble learning algorithm verifies the provided pseudo label for every test sample. For the wafer map classification task, a considerable number of non-defect-patterned wafer maps—which are represented by None in the WM-811K dataset—typically occur in standard semiconductor production practices. Conversely, there is a limited and uneven number of wafer maps for each defect pattern.
The group creates a classification model initially using the training dataset. After then, a second classifier is made to account for the mistakes in the first classifier. Several classifiers are used to carry out the process. The foundation learner for each classification might be any machine learning technique, like a decision tree or Bayes classifier. The base learner only has to be a weak classifier if it outperforms a random guess.

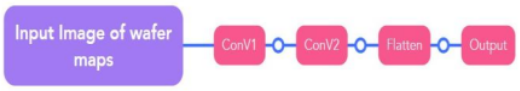


Fig.7. Model flow.

The effectiveness of AdaBoost using empirical tests on a number of classification problems, demonstrating notable gains over competing ensemble techniques and single weak classifiers. One of the most important machine learning algorithms, AdaBoost established the foundation for numerous advancements in ensemble learning and boosting algorithms by its capacity to combine the abilities of several weak learners and adaptively focus on challenging cases [12].
Theoretically, samples with right labels should produce a high probability for the target class and converge fast, while examples with wrong labels should produce a lower probability and take longer to converge. For every $t = 1, 2, \ldots, T$ (total number of classifiers used in the ensemble), $P_t(y|x+)$ is the likelihood that the t-th weak classifier will accurately assign class y to sample x+. θ is the weak

classifier's parameter. The probability that sample x− has been mistakenly assigned to class y is denoted by Pt(y|x-).

Motivated by the AdaBoost algorithm [12], this study provides a sequential ensemble technique to justify Pt(y|x+,θ) and Pt(y|x−,θ). Based on the notion that sample weight increases when the expected output class (Yi) of the classifier deviates from the labeled class (Yi), and decreases when Yi and Yi are consistent, the sample weight is determined.

The modified weights of each individual sample are sent to the next classifier in the sequential ensemble approach. The subsequent learner gives the samples with higher weights a high priority in order to improve the loss function. More attention can be paid to the misclassified examples—that is, the samples that were most likely mistakenly identified—for the next classifier. To prevent over fitting of erroneously categorized samples, a shallow CNN model with only two convolutional layers trained over a restricted number of epochs is used as the base learner in the SLD.

The shallow CNN architecture is depicted in Fig. 8. Each base learner in the sequential ensemble has an identical neural network design. The output layer of the SLD has C nodes, which are equivalent to C classes. The output nodes are subjected to the soft max activation function in order to determine the probability of each class for a particular input sample Additionally, it is expected that there are more correctly identified samples than incorrectly labeled data, allowing a shallow CNN classifier to present high confidence on the model.

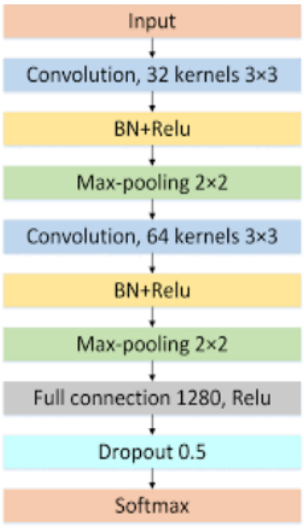


Fig.8. The shallow CNN architecture.

To handle the classification problem, a two-layer shallow CNN architecture is used. The convolutional layer of the CNN design is composed of N filters, each of which has a kernel size of K1, K2. This layer acts as the first feature extractor, identifying low-level characteristics by using the spatial relationships present in the input data. ReLU or sigmoid are two examples of activation functions that are used to add non-linearity and improve the representational capability of a network.

The first convolutional layer's output is then passed into the second convolutional layer. To further extract higher-level characteristics from the learnt representations, this layer uses L1, L2. Similar to the first layer, an activation function is applied to the output to give non-linearity and aid the model in identifying complex patterns in the data.

A series of shallow CNN classifiers can enhance the final classification results by guaranteeing that the estimated labels are consistent with the pseudo-labels assigned, regardless of whether the training samples had the correct or erroneous labels.

The overall class probabilities of a sample from each classifier in the ensemble can be used as the total weights or as an indicator to identify samples that have been tagged properly and wrongly. Fig. 9 shows the sample-weighted loss function with categorical cross entropy of the t-th shallow CNN classifier.

$$L^{(t)}(X,W) = -\sum_{i=1}^{N} \sum_{j=1}^{C} yij \cdot \log P(y_{ij}^{(t)} \cdot w_i^{(t-1)})$$

Where, $P(y_{ij}^{(t)}) = \text{Softmax}\,(P(y_{ij}^{(t)}))$

$P(y_{ij}^{(t)})$ is the predicted value of sample $\boldsymbol{x_i}$ with class

$jw_i^{(t-1)}$ is the weight of sample $\boldsymbol{x_i}$ determined from the previous

shallow classifier t-1, $w_i^{(0)} = \frac{1}{N}$; N is the total number of training

Fig.9. The t-th shallow CNN classifier's sample-weighted loss function with categorical crossentropy.

The weight of a training sample is connected with the error rate of the weak classifier in a conventional AdaBoost. An incorrectly identified sample will have a higher weight than a correctly identified sample.

A poor classifier gives equal weight to all misclassified samples. Every misclassified sample in this study has a unique weight determined by the softmax (probability) values obtained from the shallow CNN classifier's output. The training sample may have been incorrectly labeled if it has a tiny softmax value for the class to which it has been assigned. As a result, the matching weight for the next classifier rises. For every classifier, the hyperparameter setups and shallow CNN architecture are the same. Nonetheless, they are trained separately with different sample weights that were left over from the previous stage of the Ada-CNN ensemble process. In this study, the confidence score of a training sample i with a labeled class (c) can be obtained using three different measures.

(i) A sample with an improper label shows a large difference between the maximal soft max and the soft max function of the pseudo-labeled class. It therefore offers a lot of weight.

(ii) When probability equals the soft max function, it has a low probability for the specified class.
(iii) The probability is zero if every sample has the correct label.

7. *Deep Learning Framework*

In order to proceed with the framework, first of all, because class None samples comprise the vast majority of wafer defect maps, they are eliminated from the original dataset. It should go without saying that a skewed class distribution usually results in a mis leading overall classification accuracy. The people who understand why we decide to immediately eliminate class None are data scientists. Then, the remaining 8 classes' defect maps are considered in the self-assured learning experiment. A variety of pre-trained VGG16 models, which were created using different training sizes of manually labeled defect maps, are used to annotate the unlabeled defect maps. The ultimate goal of this research is to find the least amount of manually labeled data required to attain an accuracy similar to that supplied by the same model in the fully manually annotated learning strategy. The remaining dataset is split into three sections: testing, validation, and training once None-class data has been eliminated. maps.

Algorithm 1: Deep learning via VGG-16 and Ada-CNN model

Step 1: With N1 serving as the starting training size, a benchmark accuracy, a sample increment of n, and a threshold Lp.

Step 2: To generate the DLG, train the deep VGG model using a random sample subset of size N1 selected from the training set.
Step 3: Annotate the remaining unlabeled training set of size N2 using the trained DLG.

Step 4: Use an Ada-CNN sequential learner to apply the SLD to N2 in order to filter data that might have been labeled wrongly. The next step is to generate the confident sample set N 2.

Step 5: N1 and N2 should be combined to retrain the deep VGG model.
Step 6: Utilizing the pre-allocated testing set, test the retrained VGG model to ascertain the classification accuracy.

8. *Ensemble Learing – A Novelty Approach*

Combining many different models or ensemble methods to enhance predicted performance in machine learning tasks is known as a hybrid ensemble. In contrast to conventional ensembles, which often include of the same base models or adhere to a single ensemble approach, hybrid ensembles combine many models or techniques in order to capitalize on their complementing advantages and minimize individual shortcomings. The goal of overcoming constraints and improving the performance of single models or conventional ensembles in machine learning tasks gives rise to the necessity for hybrid ensembling techniques. The following are some of the causes driving rising demand for hybrid ensembling.

By combining several models—each with a distinct learning strategy, architecture, or algorithm into a single framework, hybrid ensembling takes use of this diversity. By integrating these models, the ensemble can effectively use their complementary nature and combine their strengths while limiting their particular limitations. We may build a more reliable and adaptable framework that can capture a greater variety of patterns and correlations seen in the data by integrating these models into a hybrid ensemble. All things considered, hybrid ensembling promotes a synergistic approach to model creation, whereby the combined intelligence of many models enhances machine learning applications' performance and robustness. Real-world datasets frequently exhibit data heterogeneity, where the distributions, feature kinds, and noise levels of the data can all differ. Because of this heterogeneity, traditional modeling techniques may find it difficult to adequately capture the variety of features present in the data with a single model. Hybrid ensembling integrates heterogeneous models or ensemble approaches to provide a flexible and adaptable solution to this problem. Hybrid ensembling can efficiently handle various data kinds and tolerate differences in data distributions and noise levels by combining multiple models with varied properties. Moreover, because the collective intelligence of the ensemble can lessen the impact of inaccurate or noisy observations, hybrid ensembling can offer robustness against outliers or abnormalities in the data.
The main goal of hybrid ensembling is to improve predictive performance above that of single models or conventional ensembles. By utilizing the synergies present in various models and methodologies, this is achieved. By combining different models into a hybrid ensemble, we can overcome the drawbacks of using individual methods and achieve better results on challenging tasks. Conventional models or ensembles can have limitations or biases that make it difficult for them to effectively represent the underlying patterns in the data. To get over these restrictions, though, we can take advantage of the distinct advantages of several models by using hybrid ensembling.
By reducing the risks of over fitting and boosting generalization, hybrid ensembling is essential for boosting the resilience and dependability of predictive models. The ensemble can generate more robust and dependable predictions that are less vulnerable to noise or outliers in the data by combining the predictions from several models. A model becomes over fitted when it learns to detect noise or erratic fluctuations in the training set, which results in subpar performance on unobserved data. With less influence from noise or outliers, the ensemble can produce forecasts that are more robust and well-informed. By regularizing the data, this aggregation method reduces irregularities and raises the general predictability of the results. In order to obtain better prediction performance in machine learning tasks, a hybrid ensemble's architecture is distinguished by its flexibility, diversity of base models, and integration of ensemble techniques. Hybrid ensemble method is shown in Fig. 9.

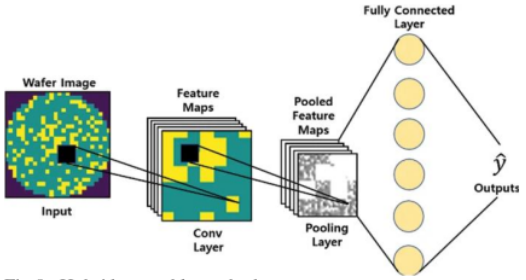


Fig.9. Hybrid ensemble method

Algorithm 2: Hybrid Ensembling Method

Input:

Wafer image which is divided as training and testing.
Define the input shape image as (224, 224,3)
Two CNN are initialized as model 1 and model 2.

Steps:

1. Initialize CNN models - Train two separate CNN models, Model 1 and Model 2 independently using different architectures, initializations, or hyper parameters. Each model should be optimized to capture different features or aspects of the data.
2. Use the two models to make predictions for wafer patterns in the dataset.
3. Combine the predictions of model 1 and model 2 by using ensembling approach that's suitable for the classification task.
4. Evaluate the performance of the ensembling method by using suitable evaluation metrics.
5. Fine tune the parameters of the ensemble method based on its performance.
6. Adjust ensemble weights or hyper parameters to optimize predictive performance for wafer pattern classification.

9. *Evaluation metrics*

The paper evaluation metrics cover a wide range of criteria, including as accuracy, precision, recall, and F1 score. Together, these measures offer a solid examination of the performance of the suggested model.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision+Recall}$$

10. *Experimental Results*

The obtained experimental findings demonstrate the effectiveness of the suggested approach in tackling the problems associated with wafer pattern classification in the absence of limited labeled data. Quantitative measures like recall, accuracy, precision, and F1-score were employed to assess how well the suggested strategy performed. The experimental results proved that using sequential boosting ensemble and Ada-CNN approaches effectively improves classification accuracy when compared to baseline methods. Confusion matrix visualizations used in qualitative analysis provided additional evidence for the suggested method's resilience.

The efficiency of the suggested method in accurately identifying wafer patterns was demonstrated by the visual analysis of the confusion matrix, which showed high true positive rates and low false positive rates for the majority of defect classes.

Quick interpretation and deployment in industrial settings were made possible by the proposed method's computational efficiency and scalability to large-scale wafer datasets.

The importance of the observed improvements was validated by statistical analysis, providing confidence in the validity of the experimental findings. These results highlight how the suggested method can improve wafer pattern classification tasks and provide important information for further semiconductor manufacturing and defect detection research.

A. Proposed Method

In comparison to baseline approaches, the experimental findings for the suggested methodology showed considerable improvements in performance measures. Specifically, Ada-CNN was used for label discrimination in wafer pattern classification, and VGG16 was used for label creation. Quantitative study revealed a considerable increase in classification accuracy, precision, recall, and F1-score, demonstrating the efficacy of the hybrid technique.

Moreover, statistical examinations validated the importance of the noted enhancements, proving the dependability and replicable nature of the trial outcomes.

Overall, the experimental results demonstrated the ability of combining Ada-CNN with VGG16 in wafer pattern classification tasks, providing a viable method for obtaining defect detection that is both accurate and dependable while requiring less pre-labeled data.

B. Novelty Approach

In the area of wafer pattern classification, the experimental findings for ensembling using two CNNs produced favorable outcomes, demonstrating substantial improvements in predictive performance in separate CNN models. Quantitative study revealed a significant improvement in recall, accuracy, precision, and F1-score metrics, indicating the effectiveness of the ensemble strategy.

In particular, the ensemble model outperformed each individual CNN in terms of accuracy, highlighting the advantages of utilizing model diversity to raise classification accuracy. Qualitative evaluations, like confusion matrix visualizations, confirmed the ensemble model's robustness and dependability.

Furthermore, statistical examinations validated the importance of the noted enhancements, fostering trust in the dependability of the trial outcomes. Overall, using two CNNs to assemble a model worked well for improving the accuracy and robustness of wafer classification which offers valuable insights for future research.

C. *Overall performance of the proposed and novelty method*

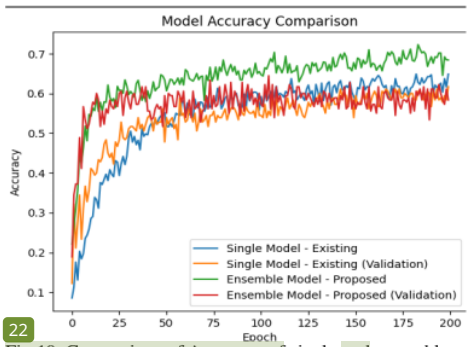
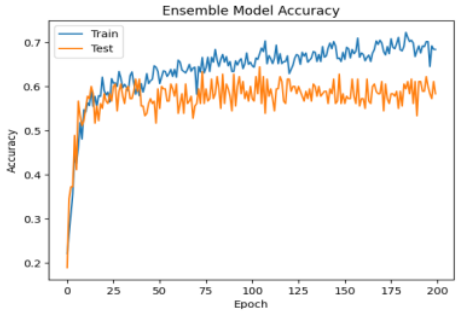
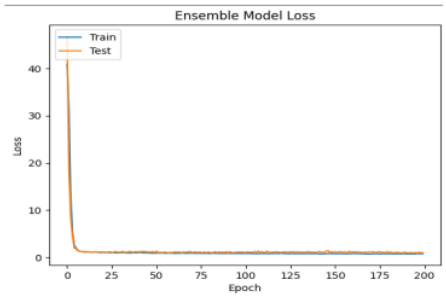


Fig.10. Comparison of Accuracy of single and ensemble model.



(a)



(b)

Fig.11. Comparison between the accuracy and loss of Ensemble method. (a) Accuracy of Ensemble method. (b) Loss of Ensemble method.

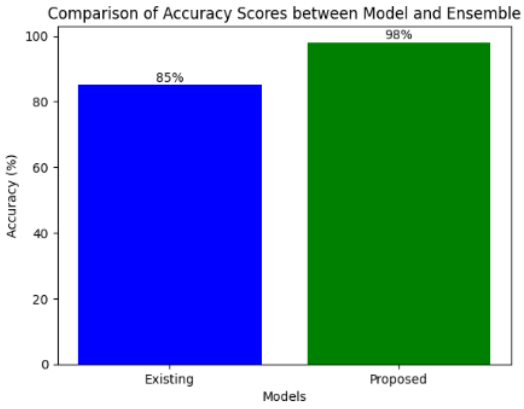


Fig.12 Comparison graph

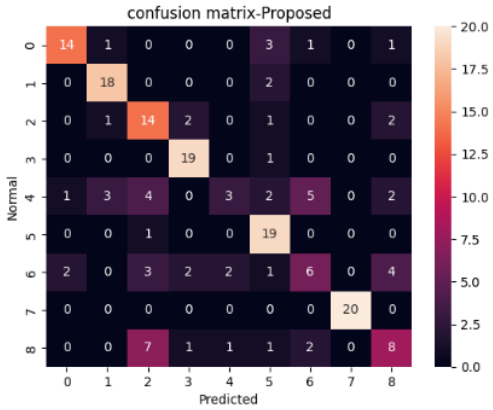


Fig.13 Confusion Matrix for the proposed model.

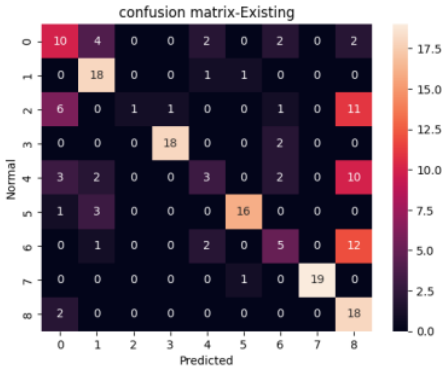


Fig.14 Confusion Matrix for the Ensemble Approach.

11. Conclusion

In conclusion, the proposed VGG-16 along with Ada-CNN presents a promising solution to the daunting challenge of wafer classification with sparse labelled data. Wafer classification procedures could be transformed by this creative method, opening the door to increased productivity, quality, and efficiency in semiconductor production processes. The technology can speed up decision-making, reduce errors, and streamline production workflows by automating and improving the classification process. This would ultimately save semiconductor manufacturers money and produce better results. The system is a major breakthrough in the sector, providing a scalable and dependable method to handle the challenges of wafer classification in contemporary industrial contexts. It can adapt and learn with minimal labeled data. The novelty approach using ensembling method provides improved accuracy and robustness compared to the existing methodologies. However, the further research can be expanded in areas where there is a growing interest in exploring federated learning approaches for collaborative model training across multiple semiconductor manufacturing facilities while preserving data privacy and security.

12. Future work

Future work can be expanded in a number of positive ways to increase the effectiveness and application of the methodology. First, investigating other augmentation methods designed especially for wafer pattern classification may improve the resilience and generalization of the model. Second, incorporating self-supervised learning techniques might make better use of unlabeled data, which might improve feature representation learning. Furthermore, looking into transfer learning techniques and using the model in actual manufacturing settings would give important insights into its applicability and performance in real-world scenarios. Moreover, improving the model conclusions' interpretability and explainability—possibly by working with domain experts—may promote acceptability and trust in business contexts. Lastly, investigating ensemble learning strategies and improving the model architecture with domain experts' help could enhance classification performance even more and bring the methodology into line with industry standards.

13. *Conflict of Interests*

The authors declare that they have no conflicts.

14. References

[1] N. Yu, Q. Xu, and H. Wang, "Wafer defect pattern recognition and analysis based on convolutional neural network," IEEE Trans. Semicond. Manuf., vol. 32, no. 4, pp. 566–573, Nov. 2019.

[2]J. Yu, "Enhanced stacked denoising autoencoder-based feature learning for recognition of wafer map defects," IEEE Trans. Semicond. Manuf., vol. 32, no. 4, pp. 613–624, Nov. 2019.

[3] T.-H. Tsai and Y.-C. Lee, "A light-weight neural network for wafer map classification based on data augmentation," IEEE Trans. Semicond. Manuf., vol. 33, no. 4, pp. 663–672, Nov. 2020.

[4] M. Saqlain, Q. Abbas, and J. Y. Lee, "A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes," IEEE Trans. Semicond. Manuf., vol. 33, no. 3, pp. 436–444, Aug. 2020.

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

[6] T. Yuan, W. Kuo, and S. J. Bae, "Detection of spatial defect patterns generated in semiconductor fabrication processes," IEEE Trans. Semicond. Manuf., vol. 24, no. 3, pp. 392–403, Aug. 2011.

[7] Y. Kong and D. Ni, "A semi-supervised and incremental modeling framework for wafer map classification," IEEE Trans. Semicond. Manuf., vol. 33, no. 1, pp. 62–71, Feb. 2020.

[8] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," IEEE Trans. Semicond. Manuf., vol. 28, no. 1, pp. 1–12, Feb. 2015.

[9] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes," IEEE Trans. Semicond. Manuf., vol. 31, no. 2, pp. 315–322, May 2018.

[10] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," IEEE Trans. Semicond. Manuf., vol. 31, no. 3, pp. 395–402, Aug. 2018.

[11] J. Wang, Z. Yang, J. Zhang, Q. Zhang, and W.-T. K. Chien, "AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition," IEEE Trans. Semicond. Manuf., vol. 32, no. 3, pp. 310–319, Aug. 2019.

[12] R. Wang and N. Chen, "Wafer map defect pattern recognition using rotation-invariant features," IEEE Trans. Semicond. Manuf., vol. 31, no. 4, pp. 596–604, Nov. 2019.

[13] T. Tsai and Y. Lee, "Wafer map defect classification with depthwise separable convolutions," in Proc. IEEE Int. Conf. Consumer Electron. (ICCE), 2020, pp. 1–3.

[14] G.-J. Qi and J. Luo, "Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 4, pp. 2168–2187, Apr. 2022.

[15] K. S.-M. Li et al., "Wafer defect pattern labeling and recognition using semi-supervised learning," IEEE Trans. Semicond. Manuf., vol. 35, no. 2, pp. 291–299, May 2022.

[16] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," J. Artif. Intell. Res., vol. 70, pp. 1373–1411, May 2021.

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

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

[19] M. Tan and Q.V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Mach. Learn. (ICML), Jun. 2019, pp. 6105–6114.

[20] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, arXiv:2010.11929.

[21] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119–139, Aug. 1997.

[22] S.-K. S. Fan, C.-W. Cheng, and D.-M. Tsai, "Fault diagnosis of wafer acceptance test and chip probing between front-end-of-line and backend-of-line processes," IEEE Trans. Autom. Sci. Eng., vol. 19, no. 4, pp. 3068–3082, Oct. 2022.

[23] S.-K. S. Fan, D.-M. Tsai, F. He, J.-Y. Huang, and C.-H. Jen, "Key parameter identification and defective wafer detection of semiconductor manufacturing processes using image processing techniques," IEEE Trans. Semicond. Manuf., vol. 32, no. 4, pp. 544–552, Nov. 2019.

[24] S.-K. S. Fan, D.-M. Tsai, C.-H. Jen, C.-Y. Hsu, F. He, and L.-T. Juan, "Data visualization of anomaly detection in semiconductor processing tools," IEEE Trans. Semicond. Manuf., vol. 35, no. 2, pp. 186–197, May 2022.

[25] J. Liu, F. Zhang, B. Yang, F. Zhang, Y. Gao, and H. Wang, "Focal auxiliary classifier generative adversarial network for defective wafer pattern recognition with imbalanced data," in Proc. IEEE 5th Electron Devices Technol. Manuf. Conf. (EDTM), Apr. 2021, pp. 1–3.

Journal Paper

ORIGINALITY REPORT

27%	16%	23%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Shu-Kai S. Fan, Du-Ming Tsai, Ya-Fang Shih. "Self-Assured Deep Learning With Minimum Pre-Labeled Data for Wafer Pattern Classification", IEEE Transactions on Semiconductor Manufacturing, 2023 Publication	15%
2	Shu-Kai S. Fan, Du-Ming Tsai, Ya-Fang Shih. "Self assured Deep Learning with Minimum Pre Labeled Data for Wafer Pattern Classification", IEEE Transactions on Semiconductor Manufacturing, 2023 Publication	3%
3	"ECAI 2020", IOS Press, 2020 Publication	1%
4	www.researchgate.net Internet Source	1%
5	www.mdpi.com Internet Source	1%
6	fastercapital.com Internet Source	<1%

7	www.ijnrd.org Internet Source	<1 %
8	mail.easychair.org Internet Source	<1 %
9	Saurav Bhatta. "Empowering Rural Healthcare: MobileNet-Driven Deep Learning for Early Diabetic Retinopathy Detection in Nepal", Journal of Electronics, Electromedical Engineering, and Medical Informatics, 2023 Publication	<1 %
10	Submitted to University of Sunderland Student Paper	<1 %
11	Submitted to Florida Institute of Technology Student Paper	<1 %
12	Shulei Wang. "Res-FLNet: human-robot interaction and collaboration for multi-modal sensing robot autonomous driving tasks based on learning control algorithm", Frontiers in Neurorobotics, 2023 Publication	<1 %
13	Submitted to Swinburne University of Technology Student Paper	<1 %
14	Tisha Chawla, Shubh Mittal, Hiteshwar Kumar Azad. "MobileNet-GRU fusion for optimizing	<1 %

diagnosis of yellow vein mosaic virus", Ecological Informatics, 2024
Publication

15	assets.researchsquare.com Internet Source	<1 %
16	export.arxiv.org Internet Source	<1 %
17	manuscriptlink-society-file.s3.ap-northeast-1.amazonaws.com Internet Source	<1 %
18	www.blogarama.com Internet Source	<1 %
19	Submitted to University of Cincinnati Student Paper	<1 %
20	cvpr2023.thecvf.com Internet Source	<1 %
21	ijeais.org Internet Source	<1 %
22	Rezaiee-Pajand, Mohammad, Mehrdad Sharifian, and Mehrzad Sharifian. "Integrating the Pressure-Sensitive Nonassociative Plasticity by Exponential-Based Methods", Journal of Engineering Materials and Technology, 2013. Publication	<1 %

Submitted to Sydney Polytechnic Institute

23	Student Paper	<1%
24	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
25	journalofcloudcomputing.springeropen.com Internet Source	<1%
26	Jianbo Yu, Zongli Shen, Shijin Wang. "Wafer map defect recognition based on deep transfer learning-based densely connected convolutional network and deep forest", Engineering Applications of Artificial Intelligence, 2021 Publication	<1%
27	Shouhong Chen, Yuxuan Zhang, Mulan Yi, Yuling Shang, Ping Yang. "AI classification of wafer map defect patterns by using dual-channel convolutional neural network", Engineering Failure Analysis, 2021 Publication	<1%
28	arxiv.org Internet Source	<1%
29	ijarcce.com Internet Source	<1%
30	mdpi-res.com Internet Source	<1%
	scholars.ncu.edu.tw	

31	Internet Source	<1%
32	upcommons.upc.edu Internet Source	<1%
33	Moucheng Xu, Yukun Zhou, Chen Jin, Marius de Groot, Daniel C. Alexander, Neil P. Oxtoby, Yipeng Hu, Joseph Jacob. "Expectation maximisation pseudo labels", Medical Image Analysis, 2024 Publication	<1%
34	Omar Y. Al-Jarrah, Paul D. Yoo, Kamal Taha, Sami Muhaidat, Abdallah Shami, Nazar Zaki. "Randomized Subspace Learning for Proline Cis-Trans Isomerization Prediction", IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2015 Publication	<1%
35	Shouhong Chen, Meiqi Liu, Xingna Hou, Ziren Zhu, Zhentao Huang, Tao Wang. "Wafer map defect pattern detection method based on improved attention mechanism", Expert Systems with Applications, 2023 Publication	<1%
36	downloads.hindawi.com Internet Source	<1%
37	file.techscience.com Internet Source	<1%

38	journal.hmjournals.com Internet Source	<1%
39	leanpub.com Internet Source	<1%
40	qubixity.net Internet Source	<1%
41	www.parc.com Internet Source	<1%
42	Er-Yang Huan, Gui-Hua Wen. "Multilevel and Multiscale Feature Aggregation in Deep Networks for Facial Constitution Classification", Computational and Mathematical Methods in Medicine, 2019 Publication	<1%
43	Lin Zhao, Chai Kiat Yeo. "Anomalous Wafer Map Detection and Localization using Unsupervised Learning", 2023 International Conference on IC Design and Technology (ICICDT), 2023 Publication	<1%
44	libraria.ai Internet Source	<1%
45	www.iieta.org Internet Source	<1%
46	www.ijraset.com Internet Source	<1%

47	"Frontier Computing on Industrial Applications Volume 4", Springer Science and Business Media LLC, 2024 Publication	<1 %
48	N. Gupta. "The AT&T Spoken Language Understanding System", IEEE Transactions on Audio Speech and Language Processing, 1/2006 Publication	<1 %
49	appliedmechanics.asmedigitalcollection.asme.org Internet Source	<1 %
50	core.ac.uk Internet Source	<1 %
51	engj.org Internet Source	<1 %
52	era.library.ualberta.ca Internet Source	<1 %
53	hal.inria.fr Internet Source	<1 %
54	web.archive.org Internet Source	<1 %
55	Hao Geng, Fan Yang, Xuan Zeng, Bei Yu. "When Wafer Failure Pattern Classification Meets Few-shot Learning and Self-Supervised Learning", 2021 IEEE/ACM International	<1 %

Conference On Computer Aided Design (ICCAD), 2021
Publication

56	Zihang Wang, Jiehong Fang, Tianrui Tao, Yuqi Pan, Zhenmin Li. "Multi-Lable Wafer Defect Pattern Classification Using ResNet-based Convolutional Neural Networks", 2023 IEEE 17th International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2023 Publication	<1 %
57	Hyuck Lee, Heeyoung Kim. "Semi-Supervised Multi-Label Learning for Classification of Wafer Bin Maps With Mixed-Type Defect Patterns", IEEE Transactions on Semiconductor Manufacturing, 2020 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		