WaferSegClassNet - A Light-weight Network for Classification and Segmentation of Semiconductor Wafer Defects

Subhrajit Nag, Dhruv Makwana, Sai Chandra Teja R, Sparsh Mittal, C Krishna Mohan

arXiv:2207.00960v1 [cs.CV] 3 Jul 2022

Abstract

As the integration density and design intricacy of semiconductor wafers increase, the magnitude and complexity of defects in them are also on the rise. Since the manual inspection of wafer defects is costly, an automated artificial intelligence (AI) based computer-vision approach is highly desired. The previous works on defect analysis have several limitations, such as low accuracy and the need for separate models for classification and segmentation. For analyzing mixed-type defects, some previous works require separately training one model for each defect type, which is non-scalable.

In this paper, we present WaferSegClassNet (WSCN), a novel network based on encoder-decoder architecture. WSCN performs simultaneous classification and segmentation of both single and mixed-type wafer defects. WSCN uses a "shared encoder" for classification, and segmentation, which allows training WSCN end-to-end. We use N-pair contrastive loss to first pretrain the encoder and then use BCE-Dice loss for segmentation, and categorical cross-entropy loss for classification. Use of N-pair contrastive loss helps in better embedding representation in the latent dimension of wafer maps. WSCN has a model size of only 0.51MB and performs only 0.2M FLOPS. Thus, it is much lighter than other state-of-the-art models. Also, it requires only 150 epochs for convergence, compared to 4,000 epochs needed by a previous work. We evaluate our model on the MixedWM38 dataset, which has 38,015 images. WSCN achieves an average classification accuracy of 98.2% and a dice coefficient of 0.9999. We are the first to show segmentation results on the MixedWM38 dataset. The source code can be obtained from https://ckmvigil.github.io/wscn/

I. Introduction

Semiconductors are used in all modern electronic devices and technologies. Hence, semiconductor chips have become the key driver for industrial growth. Given the intense quality and low-cost demands placed on semiconductor manufacturing, achieving a high yield (fraction of fault-free devices out of total devices) is essential to generate higher revenues. As a result, yield-based quality is the primary criterion for a company to succeed in the market.

Wafer fabrication is a costly and complex process involving many process steps involving many process variables. Finally, the wafers are evaluated through end-of-line (EOL) tests for ensuring the products perform the desired functionality. One of these tests outputs a wafer map (WM) showing the count and locations of defective dies on every wafer. The increasing complexity of wafer design and increasing integration density have aggravated defects in semiconductor wafers. This has also led to mixed-type defects, where two or more defect patterns exist in a single wafer. Upon identifying the correct defect type, the engineers can localize the cause(s) of the defects in the process, correct them and increase the yield. Yet, classifying mixed-type defects is especially challenging since the location and angle of single defects can vary widely. Hence, they may combine in numerous ways to create mixed-type defects. Efficient detection of these defects is a crucial challenge faced by semiconductor manufacturing companies.

Traditionally, wafer defect detection has been done manually by experienced engineers [1]. However, this process is cumbersome and unscalable. Artificial intelligence (AI) has shown promising results in many real-world applications [2]. An AI-based computer-vision approach [3] allows defect detection in a contact-free manner. An AI model can extract edge features, surface textures, and pattern information to detect the defects accurately [4]. However, previous works on wafer defect detection have crucial limitations (refer Section II for more details). Some works have been evaluated on minuscule datasets [5], [6] and others have tested for only a small number of defects [7]. A few other works provide low classification accuracy (93.2%) [8]. All the previous works have only performed either classification or segmentation, but not both. Many previous works have mostly used synthetic WMs with simulated defects [9]. Thus, there is a need to design novel network for wafer defect classification and segmentation and evaluate them on large-scale realistic datasets.

Contributions: In this paper, we present WaferSegClassNet (WSCN), a deep-learning model for simultaneously performing both classification and segmentation of defects on WMs. To the best of our knowledge, WSCN is the first wafer defect analysis model that performs both segmentation and classification. WSCN follows a multi-task learning framework to segment and classify an image simultaneously.

WSCN follows encoder-decoder architecture. The decoder has two branches: one for segmentation and one for classification. WSCN is carefully designed to achieve high predictive performance yet remain a lightweight network. For example, the encoder uses separable convolutions to reduce the memory bandwidth and computational requirements. WSCN uses a "shared encoder"

Subhrajit Nag is at IIT Hyderabad. Dhruv Makwana and Sai Chandra Teja R are at CKMVigil Pvt Ltd. Sparsh is at IIT Roorkee. C Krishna Mohan is at IIT Hyderabad and CKMVigil Pvt Ltd. Email of the corresponding author: saichandrateja@ckmvigil.in

for segmentation and classification, which allows training WSCN end-to-end. WSCN has a model size of only 0.51MB and performs only 0.2M FLOPS. Thus, it is much lighter than other state-of-the-art models. By performing INT8 quantization, the model size of WSCN decreases further by three times, with negligible impact on predictive performance.

For supervised learning, the most widely used loss function is the "cross-entropy loss", however, it has crucial limitations (refer Section IV-D for more details). We use N-pair contrastive loss to first train the encoder for 100 epochs and then use BCE-DICE loss to train segmenation branch and Categorical Cross-entropy loss for classification branch for 50 epochs. Use of N-pair contrastive loss helps in better embedding representation in latent dimension of WMs having single defect and mixed defect types. This improves classification and segmentation results. Use of N-pair contrastive loss also reduces the time to train. Specifically, WSCN requires only 150 epochs for convergence, compared to 4,000 epochs needed by the DCNet [8] model.

We evaluate WSCN on a large dataset viz., the MixedWM38 dataset [10] (refer Section III for more details). It has 38,015 images, corresponding to 38 single and mixed type defect patterns, comprehensively covering all the defect patterns. For defect classification, we compare our model with ResNet50 [11], DenseNet121 [12], LeNet [13], AlexNet [14], EfficientNetB0 [15], MobileNetV2 [16], ResNet18 [11], and DCNet [8], a recently proposed CNN for wafer defect classification. WSCN achieves an average classification accuracy of 98.20%, which is much better than the 93.20% accuracy achieved by DCNet. The ResNet50 and DenseNet121 models achieve an accuracy of 97.19% and 98.34%, respectively. However, these models have a much larger model size. In fact, the model size of ResNet50 is 200× that of WSCN. LeNet, ResNet18, AlexNet, EfficientNetB0 and MobileNetV2 also achieve lower classification accuracy than WSCN.

A key challenge in analyzing mixed-type defects is that one defect may overlap with another defect. Since object detection only provides a bounding box around the defect, it is insufficient for separately detecting different defects. Hence, we perform defect *segmentation* and not defect *detection*. In MixedWM38 dataset, each WM has three regions: wafer boundary, background and defect. MixedWM38 dataset provides class labels, which are required for classification. However, MixedWM38 dataset does not provide annotations for bounding box detections and segmentation masks. To get around this limitation, we combine wafer boundary and background into a single class, and the defect into another class. We then perform binary segmentation to segment the defects (Section III-B).

As for segmentation results, WSCN achieves a dice coefficient of 0.9999 and an IoU value of 0.9999. These values indicate the effective segmentation capability of WSCN. Since no previous work has performed segmentation on this dataset, we compare WSCN with open-source segmentation models such as UNet [17], and DeepLabV3+ [18]. These models achieve a similar dice coefficient as WSCN. However, they require a much larger model size, e.g., the model size of UNet is 250× that of WSCN. The frame-rate of WSCN is 25.11 frames-per-second on the Tesla P100 GPU. This demonstrates that WSCN can work in real-time to perform defect detection with high-performance.

The remainder of this paper is organized as follows. Section II discusses related works on wafer defect analysis. Section III discusses the MixedWM38 dataset. Section IV discussed our proposed model. Section V presents the experimental platform, and comparative results. Finally, Section VI concludes this work.

II. RELATED WORK

Related works on defect classification: Nakazawa et al. [19] use CNNs for wafer defect classification. They train and validate their CNN models using synthetically generated WM. The testing is done on generated as well as on real WMs. Maksim et al. [20] evaluate the performance of well-known CNN models like MobileNetV2, VGG19, ResNet50, and ResNet34 on various WM defects. To address the problem of data scarcity, they make their own composite dataset consisting of synthetic data and experimental data. This composite dataset is used for training the models. Testing is done on the WM–811K dataset [21]. Wang et al. [8] propose a "deformable convolutional network (DCNet)" for defect classification. DCNet uses "deformable convolution" (DC) to extract the defects' feature representations by focusing on the sampling area on defective dies. The output layer is multi-labeled and is one-hot encoded. This transforms the mixed type of defects into individual single defects and ensures effective identification of each defect.

Kyeong et al. [9] train separate classification models for classifying single defects and then use all those models for finding the presence of mixed-type defects. However, such an approach is not scalable because it significantly increases storage and computation overhead. Also, it requires separately training for all those models. Further, they study only four single defect patterns, namely "circle", "ring", "scratch", and "zone". By contrast, we propose a single model for classifying single-type and mixed-type defects. Also, we test our model on 8 single-type and 29 mixed-type defect patterns.

Elliptical basis function neural networks use a hidden layer of elliptical units. They handle every input individually using separate weights. Sciuto et al. [4], [22] use elliptical basis neural networks for detecting defects in organic solar cells.

Related works on defect segmentation: Saad et al. [5] convert the wafer image from "RGB-space" to "L*a*b*" space, where all the color information is present in the a*b* spaces only. Then, they use k-means clustering to separate the pixels into two clusters, representing the defect and wafer, respectively. Nakazawa et al. [7] use encoder-decoder-based models for detection and segmentation of abnormal patterns of wafer defects. They generate abnormal wafer defects using "defect pattern generation model". They develop three separate encoder-decoder models based on SegNet [23], U-Net [17] and FCN [24]. Table I compares selected previous works on key parameters.

TABLE I: Comparison of related works

Work	Objective	Types of defects	Network	Dataset
Nakazawa et al. [19]	Classification	22 defects	CNN	28.6K for training, 1.19K for testing
Wang et al. [8]	Classification	8 single, 29 mixed	CNN based on deformable CONV	38015 images
Saad et al. [5]	Segmentation	1	k-means clustering	2 images
Nakazawa et al. [7]	Segmentation	11 defects	FCN, SegNet and U-Net	17K for training, 3.3K for testing
Han et al. [6]	Segmentation	1	RPN+U-net with dilated CONV	106 images
Ours	Classification+Segmentation	8 single, 29 mixed	Encoder-decoder CNN	38015 images

III. The MixedWM38 Dataset

A. Dataset description

The MixedWM38 dataset [10] was published by Wang et al. [8]. This dataset has 38 defect patterns, viz., one fault-free pattern, eight single defect patterns, thirteen 2 mixed-type patterns, twelve 3 mixed-type defect patterns, and four 4 mixed-type defects for a total of 38,015 WMs. The dataset contains images of 52×52 dimension and class information as one hot encoded array in numpy format. Table II describes these defects along with their possible causes. Table III shows 38 defect patterns.

TABLE II: Single-type defect classes and their possible causes (WM =wafer map, acr. = acronym)

Name	Acr.	Description	Cause of defect
Center	C	Defective die scattered in the centre of WM	Abnormality of RF (Radio Frequency) power, abnormality in liquid flow or abnormality in liquid pressure. Problem in the plasma area [1]. Thin film deposition [25] or abnormality in liquid pressure.
Donut	D	Defective die from centre of WM in a ring configuration	Redeposition of dissolved photoresist solids backing onto wafer during developing process, as the center rinsing step results less defective at the center of the wafer.
Edge-Loc	EL	Localized clusters on the edge	Uneven heating during diffusion process [26].
Edge- Ring	ER	Ring-shaped clusters around perimeter	Abnormal temperature control in the rapid thermal annealing process [27].
Local type	L	Localised clusters occuring regularly	Excess vibration in a given machine liberating enough particles [1]. Crystalline heterogeneity [28]. Due to silt valve leak, abnormality during robot handos or abnormality in the pump.
Nearful	NF	Unusual fault pattern	Reception failure.
Scratch	S	Distribution of faulty dies in a long, narrow region	Human error at the shipping and handling process [28]. Error at chemical mechanical polishing (CMP).
Random	R	Random defect without patterns	Contaminated pipes, abnormality in shower-head, or abnormality in control wafers.

TABLE III: 38 defect patterns in MixedWM38 dataset

No.	Single defect	No.	Two-mixed defect	No.	Three-mixed defect	No.	Four-mixed defect
1	Normal	10	C+EL	23	C+EL+L	35	C+L+EL+S
2	Center (C)	11	C+ER	24	C+EL+S	36	C+L+ER+S
3	Donut (D)	12	C+L	25	C+ER+L	37	D+L+EL+S
4	edgeLoc (EL)	13	C+S	26	C+ER+S	38	D+L+ER+S
5	edgeRing (ER)	14	D+EL	27	C+L+S		
6	Loc (L)	15	D+L	28	D+EL+L		
7	Nearful (NF)	16	ER+L	29	D+EL+S		
8	Scratch (S)	17	EL + S	30	D+L+S		
9	Random (R)	18	ER+S	31	D+ER+L		
		19	L+S	32	D+ER+S		
		20	D+ER	33	EL+L+S		
		21	D+S	34	ER+L+S		
		22	EL+L				

B. Data Preparation and Preprocessing

The dataset is divided into training and validation set containing 80 and 20 percent of the dataset, which is 30412 and 7603 images, respectively. In the MixedWM38 dataset, each WM has three regions: wafer boundary, background and defect. MixedWM38 dataset provides class labels, which are required for classification. However, MixedWM38 dataset does not provide annotations of segmentation masks. To get around this limitation, we combine wafer boundary and background into a single class and the defect into another class and then perform binary segmentation to segment the defects. We reshape the dataset images from 52×52 to 224×224 to learn features and patterns more effectively. To match the dimension of masks with the dimensions of the input image, we reshape masks to the shape of 224×224. We perform one-hot encoding on ground truth class labels to generate ground truth of shape $B \times 1 \times 38$, which is required for WSCN model, where B is the batch size.

IV. Proposed Approach

We present the overall architecture of our proposed WaferSegClassNet network (Section IV-A), the encoder and decoder stages (Section IV-B-IV-C). We also discuss the loss function used for pretraining the encoder (Section IV-D).

A. *WaferSegClassNet architecture*

The diversity of defect shapes and defect overlap for semiconductor wafer defect datasets has posed new challenges for identifying WM defects. To solve these challenges and enable CNN to identify different types of defect shapes and defects overlap, we propose a novel model, WaferSegClassNet (WSCN), that can perform both defect classification and segmentation. WSCN model follows a multi-task learning framework that allows it to learn to segment and classify an image simultaneously. A classification network works with global information, whereas a segmentation network incorporates more local information in the input image. Thus, we propose a single network that focuses both on global and local information. Some previous works (e.g., [29]) remove global random defects before applying classification or clustering. Our CNN-based approach does not require such ad-hoc steps, nor does it require clustering.

Figure 1(a) shows the overall architecture of WaferSegClassNet. Here, the spatial dimension of the input is 224×224, and the output dimension of the encoder branch is 14×14. WSCN follows encoder-decoder architecture, where the encoder performs downsampling and the decoder performs upsampling. In Figure 1(a), f denotes the number of filters. At each downsampling layer, f is used as 8, 16, 16, 32, and 64, respectively. At each up-sampling layer, f is used as 32, 16, 16 and 8, respectively. We now describe the encoder and decoder stages of our model.

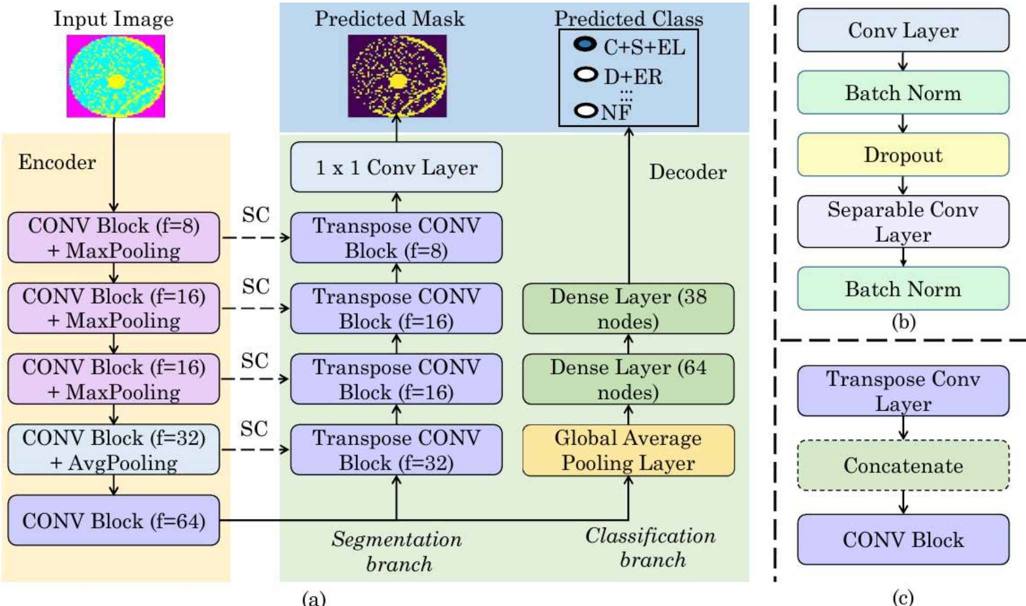


Fig. 1: a) WaferSegClassNet architecture ('f' shows the number of filters; SC = skip-connection), (b) CONV Block, (c) Transpose CONV block

B. *Encoder stage*

The encoder uses a series of convolution blocks with pooling layers to extract multi-scale local details. As depicted in Figure 1(b), the convolution block contains a 3×3 convolution layer with batch-normalization and dropout layers and a 3×3 separable convolution layer with batch-normalization. We use separable convolutions in the CONV block to reduce the memory bandwidth and computational requirements while also improving the representational efficiency. After the last convolution block, we use "average pooling" to linearly transform the vectorized feature maps. Addition of this average pooling layer acts as an effective regularizer and ensures no information loss. This is important for accurately classifying and segmenting the semiconductor wafer defects.

C. *Decoder stage*

In our network, the decoder block serves a dual purpose of performing classification and producing the semantic segmentation masks by recovering the spatial information.

Segmentation branch: The segmentation branch produces the binary segmentation mask. It has four transpose CONV blocks of 3×3 and a 1×1 convolution layer. The segmentation branch contains transpose CONV blocks for generating the output mask by upsampling the feature representation. As shown in Figure 1(c), transpose CONV blocks contain the transpose convolution layer. Transpose convolution upsamples the "input feature map" to a desired "output feature map" using learnable parameters. It interprets the coarse input data to fill-in the detail during upsampling. In the transpose convolution block, the

concatenation (“concatenate”) layer is used to enrich the semantic information of input image through skip connections. This reduces the error rate of mask generation.

Classification branch: The classification branch identifies the defect type based on the features extracted by the encoder. The classification branch contains “global average pooling” (GAP) layer and two dense layers with 64 and 38 nodes, respectively. We have used GAP layer because it is more native to the convolution structure and it enforces correspondence between feature maps and categories. GAP layer has no learnable parameters and hence, it avoids overfitting. GAP layer makes the network more robust towards spatial translations due to its nature of adding spatial information. The last dense layer outputs 38 class probabilities corresponding to 38 defect types.

D. N-pair contrastive loss function

The most widely used loss function for supervised learning is the “cross-entropy loss”. However, it has crucial limitations [30], such as noisy labels causing lack of robustness and poor performance with adversarial examples. In this paper, we use N-pair contrastive loss [31] for supervised learning where normalized embeddings from the same category are drawn together more closely than embeddings from different categories. This loss is used to train the encoder to generate vector embeddings of input images so that representations of images in the same class category are more similar than representations in different classes. Experimentally, it outperforms supervised training with “cross-entropy loss”. Training an encoder with N-pair contrastive loss helps better embedding representation in the latent dimension of semiconductor WM having single defect and mixed defect types. This improves the accuracy of the overall network. Pre-training an encoder with N-pair contrastive loss reduces the distance between similar embeddings, resulting in better feature learning in the encoder stage, which aids in both classification and segmentation branches of the decoder.

We pre-train the encoder for 100 epochs with N-pair contrastive loss to generate feature representation of input data. We use the learned feature representations from the encoder block in the segmentation branch, and classification branch by freezing weights learned from the encoder branch. Instead of a “dedicated network” for segmentation and classification, a “shared encoder” makes it possible to train WSCN end-to-end, thereby reducing training time and model size. Also, a “shared encoder” takes advantage of the shared information across multiple tasks and thus, helps in improvement of the model performance for all tasks.

V. Experimental Results

In this section, we describe the evaluation approach (Section V-A), and present the results of classification (Section V-B), including ROC-AUC curve (Section V-C). We also analyze the failure cases to gain more insights (Section V-D). We then present the results of segmentation (Section V-E). Finally, we report the latency value, and the model size reduction achieved by quantization (Section V-F) and the limitation of the proposed methodology (Section V-G).

A. Evaluation approach

Experimental platform: The experimental tests were conducted using TensorFlow-GPU 2.6, and CUDA 11.2. The initial “learning rate” is 0.001 and the “decay rate” is 0.1 whenever there is no convergence for 10 continuous epochs. The batch size is 64. We use Adam optimizer with N-pair contrastive loss for pre-training the encoder for 100 epochs. We then use Adam optimizer with BCE-Dice loss for training the segmentation network and categorical cross-entropy loss for training the classification for 50 epochs. BCE-DICE loss combines “binary cross-entropy loss”, which is “distribution-based loss” with “DICE loss” which is “region-based loss”.

Metrics: For evaluating classification performance, we use accuracy, MCC [32], ROC-AUC curve, precision and recall metrics. In the ROC curve, the “area under curve” (AUC) value ranges from 0 to 1. The higher the AUC, the better is the model’s performance at distinguishing between the positive and negative classes. Matthews correlation coefficient (MCC) is defined as follows:

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

A value of -1 and +1 indicate perfect misclassification and perfect classification, respectively. MCC considers true/false positive/negative and hence, is a better metric than ROC-AUC curve. MCC is the only binary classification rate that a high score only if the binary classifier could correctly predict a majority of positive instances and the majority of negative instances [32]. MCC is especially useful for mitigating the class imbalance issue [32].

For evaluating the performance of segmentation, we use the dice coefficient, also known as dice similarity coefficient and intersection over union (IoU) [33]. Although similar, these metrics have subtle differences [34] and hence, we have included both of them.

TABLE IV: Classwise accuracy results. Avg(1Defect) shows average of eight single defect patterns. Avg(2Defects) shows average of thirteen 2 mixed-type patterns. Average(3Defects) shows average of twelve 3 mixed-type defect patterns. Average(4Defects) shows the average of four 4 mixed-type defects.

Class	Accuracy				Precision				Recall			
	DenseNet	ResNet	DCNet	WSCN	DenseNet	ResNet	DCNet	WSCN	DenseNet	ResNet	DCNet	WSCN
1	100.00%	100.00%	99.70%	100.00%	100.00%	100.00%	94.00%	100.00%	100.00%	100.00%	91.00%	100.00%
2	100.00%	100.00%	97.80%	100.00%	99.00%	99.48%	93.00%	99.00%	100.00%	100.00%	97.00%	100.00%
3	99.00%	98%	96.50%	100.00%	98.00%	96.04%	95.00%	96.00%	99.00%	98.48%	93.00%	100.00%
4	96.00%	96.00%	94.40%	97.00%	99.00%	99.44%	96.00%	98.00%	97.00%	96.72%	91.00%	98.00%
5	100.00%	99.00%	99.80%	99.00%	97.00%	96.09%	93.00%	97.00%	100.00%	99.10%	97.00%	99.00%
6	100.00%	95.00%	93.80%	99.00%	100.00%	100.00%	99.00%	99.00%	100.00%	95.31%	100.00%	99.00%
7	97.00%	100.00%	95.80%	100.00%	92.00%	69.39%	90.00%	92.00%	97.00%	100.00%	94.00%	100.00%
8	99.00%	100.00%	93.40%	99.00%	99.00%	97.30%	60.00%	97.00%	100.00%	100.00%	88.00%	100.00%
9	98.00%	91.00%	100.00%	98.00%	99.00%	100.00%	97.00%	100.00%	98.00%	91.89%	93.00%	98.00%
10	99.00%	99.00%	99.20%	98.00%	98.00%	97.13%	94.00%	98.00%	99.00%	99.51%	94.00%	99.00%
11	100.00%	98.00%	97.90%	100.00%	98.00%	98.50%	92.00%	99.00%	100.00%	99.00%	99.00%	100.00%
12	99.00%	95.00%	98.50%	99.00%	100.00%	99.49%	92.00%	99.00%	100.00%	95.61%	96.00%	100.00%
13	100.00%	99.00%	96.70%	99.00%	98.00%	97.17%	97.00%	98.00%	100.00%	99.52%	89.00%	100.00%
14	98.00%	97.00%	99.30%	94.00%	100.00%	98.52%	96.00%	100.00%	98.00%	97.56%	92.00%	95.00%
15	100.00%	98.00%	96.10%	99.00%	98.00%	98.44%	91.00%	98.00%	100.00%	98.44%	98.00%	99.00%
16	98.00%	93.00%	98.30%	95.00%	99.00%	100.00%	94.00%	100.00%	98.00%	93.75%	97.00%	95.00%
17	99.00%	99.00%	92.80%	100.00%	97.00%	94.20%	96.00%	96.00%	99.00%	99.49%	94.00%	100.00%
18	99.00%	98.00%	93.90%	99.00%	98.00%	99.02%	98.00%	99.00%	100.00%	98.06%	89.00%	100.00%
19	99.00%	97.00%	92.30%	97.00%	100.00%	99.44%	94.00%	99.00%	99.00%	97.81%	91.00%	98.00%
20	97.00%	92.00%	94.60%	96.00%	99.00%	99.52%	95.00%	99.00%	97.00%	92.83%	91.00%	96.00%
21	99.00%	98.00%	90.70%	100.00%	99.00%	97.87%	96.00%	98.00%	99.00%	98.40%	92.00%	100.00%
22	98.00%	97.00%	90.30%	97.00%	99.00%	94.87%	98.00%	99.00%	98.00%	97.88%	88.00%	97.00%
23	97.00%	95.00%	88.90%	97.00%	98.00%	99.03%	99.00%	97.00%	97.00%	95.79%	96.00%	97.00%
24	98.00%	98.00%	89.40%	99.00%	99.00%	98.98%	92.00%	99.00%	99.00%	98.23%	100.00%	99.00%
25	97.00%	96.00%	91.40%	97.00%	98.00%	98.42%	93.00%	98.00%	98.00%	96.39%	91.00%	98.00%
26	99.00%	100.00%	92.50%	100.00%	99.00%	97.22%	97.00%	99.00%	99.00%	100.00%	97.00%	100.00%
27	98.00%	98.00%	90.50%	97.00%	99.00%	93.85%	97.00%	99.00%	98.00%	98.39%	93.00%	97.00%
28	98.00%	97.00%	88.30%	97.00%	100.00%	100.00%	95.00%	95.00%	98.00%	97.54%	91.00%	97.00%
29	98.00%	96.00%	90.50%	96.00%	99.00%	98.96%	98.00%	97.00%	98.00%	96.95%	97.00%	97.00%
30	99.00%	98.00%	92.30%	100.00%	98.00%	98.32%	89.00%	94.00%	99.00%	98.32%	100.00%	100.00%
31	99.00%	99.00%	91.50%	98.00%	98.00%	98.11%	90.00%	98.00%	99.00%	99.52%	94.00%	99.00%
32	97.00%	96.00%	88.30%	97.00%	99.00%	96.19%	99.00%	98.00%	97.00%	96.65%	88.00%	97.00%
33	97.00%	97.00%	86.20%	96.00%	100.00%	97.38%	97.00%	100.00%	97.00%	97.38%	93.00%	96.00%
34	98.00%	98.00%	89.00%	97.00%	97.00%	91.37%	98.00%	97.00%	98.00%	98.36%	94.00%	98.00%
35	95.00%	94.00%	87.00%	94.00%	100.00%	98.60%	96.00%	100.00%	96.00%	94.62%	99.00%	95.00%
36	97.00%	97.00%	90.60%	97.00%	98.00%	95.94%	99.00%	98.00%	97.00%	97.42%	96.00%	98.00%
37	98.00%	98.00%	86.40%	95.00%	99.00%	98.00%	95.00%	99.00%	99.00%	98.49%	89.00%	95.00%
38	98.00%	98.00%	88.20%	95.00%	100.00%	96.81%	92.00%	99.00%	98.00%	98.38%	92.00%	95.00%
Avg(1Defect)	98.63%	97.29%	96.44%	99.00%	97.88%	94.72%	90.38%	97.25%	98.88%	97.95%	93.78%	99.33%
Avg(2Defects)	98.85%	96.92%	95.43%	97.92%	98.69%	98.01%	94.85%	98.62%	99.00%	97.53%	93.08%	98.38%
Avg(3Defects)	97.92%	97.33%	89.90%	97.58%	98.67%	97.32%	95.33%	97.58%	98.08%	97.79%	94.50%	97.92%
Avg(4Defects)	97.00%	96.75%	88.05%	95.25%	99.25%	97.34%	95.50%	99.00%	97.50%	97.23%	94.00%	95.75%
Avg(All38)	98.34%	97.19%	93.20%	98.20%	98.61%	97.08%	94.00%	98.08%	98.55%	97.68%	95.00%	98.18%

B. Classification Results

We compare the results of classification for our model WSCN with DCNet [8], ResNet50 (shown as ResNet) [11] and DenseNet121 (shown as DenseNet) [12]. This is because DCNet was also trained on the same dataset and ResNet50 and DenseNet121 are well-known, open-source models that can be trained on any dataset. Table IV presents classwise accuracy, precision and recall results.

WSCN achieves an average classification accuracy of 98.20% on all 38 classes. For single, two mixed-type, three mixed-type, and four mixed-type defect patterns, the average classification accuracy of WSCN are 99.0%, 97.9%, 97.6% and 95.3%, respectively. Thus, with increasing mixed-type defects, the accuracy degrades gracefully. This is expected since three or four mixed-type defect patterns are difficult to classify, even for humans. For some of these images, the ground-truth (GT) label provided in the dataset itself appears to be confusing (refer Section V-D).

Table V summarizes the results on classification accuracy and also presents the Matthews correlation coefficient (MCC), the number of parameters, model size and number of computations (FLOPS) to allow comparison from different perspectives. We evaluate both small models such as LeNet, MobileNetV2, EfficientNetB0 and ResNet18 and large models such as AlexNet, ResNet50 and DenseNet121. This is to cover networks with high accuracy and networks with low model size.

The average classification accuracy of DCNet and ResNet50 are 93.20% and 97.19%, respectively. Thus, WSCN achieves superior results than DCNet and ResNet50. The classification accuracy of WSCN is marginally lower than that of DenseNet (98.34%). WSCN has only 0.09M parameters, a model size of 0.51MB, and a computation count of 0.2M. On analyzing

the MCC values, we note that WSCN (with contrastive loss) achieves higher MCC value than other networks, except for DenseNet121. A well-known limitation of DenseNet121 is that during inference, its memory consumption increases quadratically with network depth. Hence, it has a huge memory footprint (sum of model size and memory consumed by intermediate activations), leading to out-of-memory error. WSCN with contrastive loss has 0.09M parameters, but WSCN without contrastive loss has 0.08M parameters. The increase in the number of parameters is due to addition of a dense layer for generating required shape in latent dimension while using the contrastive loss. So considering the balance of model size and FLOPS, our proposed model WSCN can be regarded as the best model. In fact, the model size of WSCN is 200× lower than that of ResNet50.

Among the lighweight models, LeNet-5 has lower accuracy of 76.44%, while AlexNet and MobileNetV2 both shows accuracy of 97.91%. EfficientNetB0 has slightly lower accuracy of 96.87%. ResNet18 gives slightly better results than ResNet50, which is possible due to the simplicity of wafermap dataset.

TABLE V: Comparison of classification accuracy, MCC, parameters, model size and FLOPS of different models

Model	Accuracy	MCC	Params	Model Size (MB)	FLOPS
DenseNet121 [12]	98.34%	0.9869	8M	33.48	16.1M
ResNet18	97.96 %	0.9791	11.2M	44.98	22.39M
ResNet50 [11]	97.19%	0.9758	25.7M	103.34	51.3M
DCNet [8]	93.20%	-	2.3M	9.09	1.4M
LeNet [13]	76.44%	0.7581	5.9M	23.96	12.0M
AlexNet [14]	97.91%	0.9785	7.6M	30.47	15.2M
MobileNetV2 [16]	97.91%	0.9679	3.6M	14.83	7.1M
EfficientNetB0 [15]	96.87%	0.9620	5.4M	22.14	10.7M
WSCN (with BCE loss)	93.29%	0.9622	0.08M	0.51	0.2M
WSCN (with contrastive loss)	98.20%	0.9815	0.09M	0.51	0.2M

WSCN achieves a precision value of 98.08% and a recall value of 98.18%. As shown in Table IV, these precision and recall values are much superior to that of DCNet, better than that of ResNet50, and only slightly lower than that of DenseNet121. The values above 98% indicate that WSCN is highly effective. As we show in Section V-D, careful re-evaluation of GT labels on the misclassified data shows confusing GT labels. This has led to mispredictions which reflected in the recall/precision values.

Comparison with DCNet [8]: DCNet has used stochastic gradient descent (SGD), which is slow in convergence. In fact, DCNet requires 4,000 epochs to reach 93.2% accuracy. This indicates that the SGD optimizer is not optimal for WM classification. By contrast, we have used Adam optimizer with N-pair contrastive loss. This allows WSCN to converge in 150 epochs only, reducing the time to train. N-pair contrastive loss helps in better extraction of feature representations of the input image in the encoder. This also improves classification accuracy. As shown in Table V, on using a different loss such as the conventional BCE loss, the accuracy of WSCN reduces to 93.29%.

C. *ROC-AUC curve*

Figure 2 shows the ROC curve for performance evaluation of different classification models. WSCN has an AUC of 0.9907, which is marginally below that of DenseNet121 (0.9991) and EfficientNetB0 (0.9946). All other classifiers have lower AUC value than WSCN.

ROC-AUC curve depends on both the true positive rate and the false positive rate. However, the true negative and the false negative rate information is neglected in ROC-AUC curve. Thus, the ROC-AUC curve may provide a misleading impression that the model gives good performance even though there might be chances of misclassification. In Table V, we have presented the MCC metric, which gives a more complete picture of a classifier's performance than the ROC-AUC curve.

D. *Analysis of mispredicted wafer-maps*

We further investigate the 138 WMs that our model misclassifies. Figure 3 shows four such sample WMs. In Figure 3a, the GT annotation is Donut + EdgeLoc + Scratch. However, as evident from the figure and as predicted by our model, the GT should be Donut + Scratch only. Similarly, for Figures 3b and 3c, we find that the GT labels in the dataset are D + EL + L and ER + L, respectively, but our model predicts D + ER + L and ER, respectively. Our predictions appear to be more reasonable according to the visible defect pattern. For a few other mispredicted WMs also, we observe that the GT labels appear to be confusing, whereas our model predicts correctly.

For most misclassified 138 images, it is difficult for the model to learn the defects from the GT labels as there is no uniformity across the defect images. Especially, given the high similarity between EL and ER, distinguishing between them is challenging, and the correct GT label itself is debatable. Similarly, every "nearfull" defect is also a random defect. Hence, for four out of 138 images, the GT is "random", whereas the model predicts it as "nearfull" defect. Further, any localized defect, regardless of its size, is considered as L. Hence, it is difficult to distinguish an L defect and a regular (fault-free) pattern or other defect patterns. This discrepancy can be addressed if we have a dataset containing well-annotated bounding boxes or masks validated by semiconductor manufacturing experts.

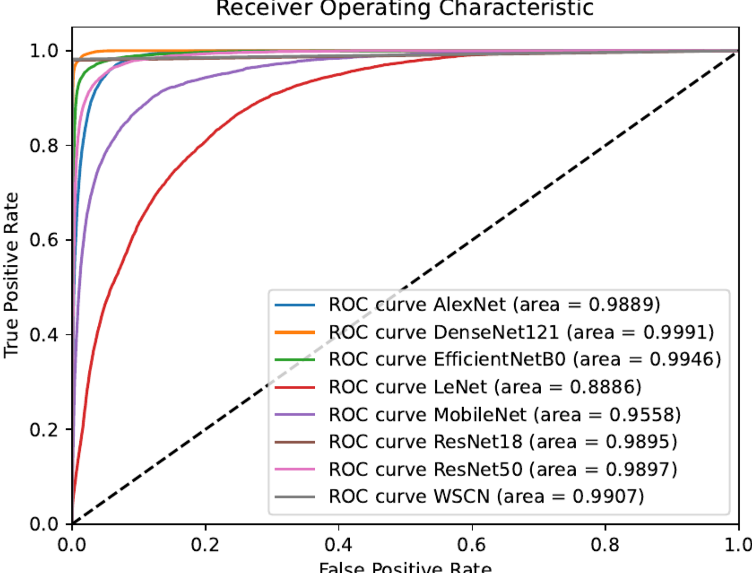
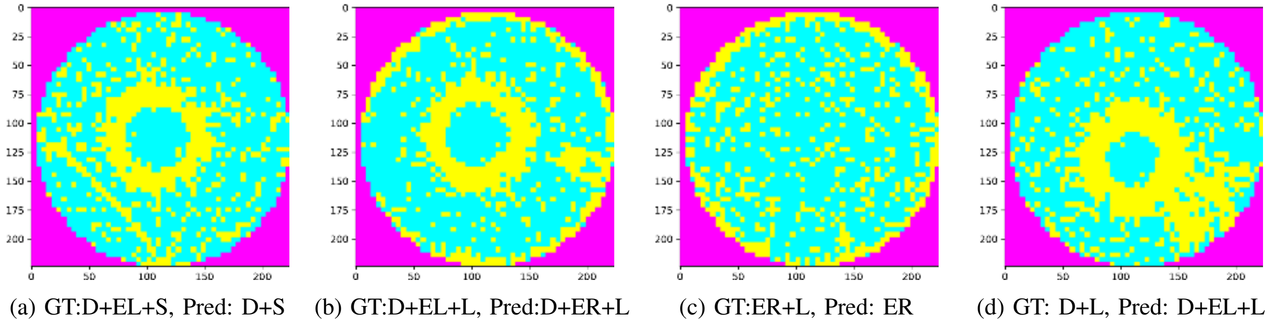


Fig. 2: ROC-AUC curve for different classifiers



(a) GT:D+EL+S, Pred: D+S (b) GT:D+EL+L, Pred:D+ER+L (c) GT:ER+L, Pred: ER (d) GT: D+L, Pred: D+EL+L

Fig. 3: Four sample failure cases (GT=ground-truth, Pred = prediction)

For some WMs, such as the one shown in Figure 3d, our model gives incorrect prediction. We believe that these WMs are difficult to predict even for humans.

E. Segmentation Results

We have used “IOU” and “dice coefficient” to measure the performance of our model. “Dice coefficient” demonstrates the similarity between the “ground truth” and “predicted mask”. As shown in Table VII in the “WSCN(FP32)” row, the value of dice-coefficient is 0.9999. This value indicates an extremely high similarity between ground truth and predicted mask. WSCN achieves an IoU of 0.9999, which confirms its high segmentation capability.

Comparative evaluation: Since ours is the first model to show segmentation on the MixedWM38 dataset, we compare WSCN against two well-known open-source segmentation models, namely UNet and DeepLabV3+ (with ResNet50 backbone). Table VII shows the results. While these models achieve comparable results as WSCN, their model size is much larger. In fact, WSCN’s model size is $250\times$ lower than that of UNet. Given this, WSCN can be considered the best among the segmentation models. Since the dice coefficient and IoU values of WSCN are 0.9999, the ROC-AUC curve remains close to 1. Hence, we omit it

Visualization of WSCN results: Table VI shows selected input images, corresponding ground-truth images, and the binary segmentation masks generated by WSCN. Evidently, the mask generated by WSCN is identical to the ground truth.

F. Latency and quantization results

Inference latency of WSCN: We measure the latency of WSCN execution as follows. We first resize the input image from 52×52 to 224×224 size. We compute results of each image in batch of 1000 images from test set by sending corresponding image to WSCN, our proposed model. We record the time taken by each image for completing this process. We repeat this

TABLE VI: Defect-class, corresponding input image, ground truth and predicted mask generated by WSCN

Defect Type	Input Image	Ground Truth	Generated Mask
Center			
Donut			
edgeLoc			
edgeRing			
Loc			
Nearful			
Scratch			
Random			

TABLE VII: Model size and segmentation results of open-source models, WSCN(FP32), i.e., without quantization and WSCN(INT8), i.e., with INT8 quantization

	Model Size	Dice Coefficient	IOU
UNet [17]	125.77MB	0.9998	0.9881
DeepLabV3+ (ResNet50 backbone) [18]	47.83 MB	0.9946	0.9924
WSCN(FP32)	0.51 MB	0.9999	0.9999
WSCN(INT8)	0.17MB	0.9999	0.9999

process ten times and take the average. We observe a frame-rate of 25.11 frames-per-second on the Tesla P100 GPU. Thus, WSCN can work in real-time to perform defect classification with high-performance.

Reducing model size using quantization: The reference WSCN model uses single-precision (FP32) weights. We further reduce the model size by performing full integer quantization. The representative dataset is used to determine the min-max values of the inputs in full-integer quantization. When the converter quantizes the model, these are required to appropriately calculate the quantization nodes. As shown in Table VII, quantization reduces the model size of WSCN to merely 0.17MB, with negligible impact on the dice-coefficient and IoU values.

G. Limitations of Proposed Methodology

We note the following areas for further improvement of our approach:

1. **Performance on unseen classes:** We have used predetermined defect classes and combinations of multiple defect classes. Due to the dynamic nature of the manufacturing process, all the different types of defects cannot be decided in the initial stage. We have used a CNN-based dynamic defect feature extraction approach from wafermap images. Since WSCN is trained on predetermined classes, it may show lower performance on real-world defect classes that are not introduced to the model in the training phase. To achieve high accuracy on those defects, we would need to retrain or fine-tune the model. Also, we plan to train WSCN using unsupervised learning to further improve its classification performance.

2. **Ability to distinguish *similar-looking* unseen classes:** We have trained our feature extraction encoder using N-pair contrastive loss. It helps in finding similarity between embeddings from the same class in the training phase. If our model comes across an unknown class which is similar to one of the known classes on which the model has been trained, our model would predict that unknown class to the similar known class. For example, if there is an unknown class which is similar to ER, then our model would predict the unknown class as ER defect. This is due to the fact that N-pair contrastive loss in the encoder maps the embeddings of similar patterns of defects close to each other in the latent representation.

3. **Improving training efficiency:** Currently, we have trained WSCN model from scratch. To increase the model's training efficiency and shorten the training time, we plan to use transfer learning.

VI. Conclusion

We propose WaferSegClassNet (WSCN), a lightweight deep learning architecture for performing the classification and segmentation of semiconductor wafer defects. We use N-pair contrastive loss to reduce the time to train and also boost classification and segmentation performance. The classification accuracy achieved by WSCN is higher than that of DCNet and ResNet50 and is in the same range as that of DenseNet121. The dice-score and IoU of WSCN are close to 1 and are competitive with open-source models such as DeepLabv3+ and UNet. WSCN's model size is 200$\times$ lower than that of ResNet50 and 250$\times$ lower than that of UNet. Thus, considering the balance of both predictive performance and model size, WSCN can be regarded as the best. We are the first to show segmentation performance on the MIXEDWM38 dataset. Our future work will extend our defect analysis technique to detect defects in other scenarios such as civil constructions.

Acknowledgement

The experimental results were obtained using the computing resources provided by IIT Roorkee under the grant FIG-100874.

References

[1] C. Hansen and P. Thyregod, "Use of wafer maps in integrated circuit manufacturing," *Microelectronics Reliability*, vol. 38, no. 6-8, pp. 1155–1164, 1998.

[2] G. Capizzi, G. Lo Sciuto, C. Napoli, R. Shikler, and M. Woźniak, "Optimizing the organic solar cell manufacturing process by means of afm measurements and neural networks," *Energies*, vol. 11, no. 5, p. 1221, 2018.

[3] S. Mittal, S. Srivastava, and J. P. Jayanth, "A survey of deep learning techniques for underwater image classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[4] G. L. Sciuto, C. Napoli, G. Capizzi, and R. Shikler, "Organic solar cells defects detection by means of an elliptical basis neural network and a new feature extraction technique," *Optik*, vol. 194, p. 163038, 2019.

[5] N. Saad, N. M. Sabri, A. Hasan, A. Ali, and H. M. Saleh, "Defect segmentation of semiconductor wafer image using k-means clustering," in *Applied Mechanics and Materials*, vol. 815. Trans Tech Publ, 2015, pp. 374–379.

[6] H. Han, C. Gao, Y. Zhao, S. Liao, L. Tang, and X. Li, "Polycrystalline silicon wafer defect segmentation based on deep convolutional neural networks," *Pattern Recognition Letters*, vol. 130, pp. 234–241, 2020.

[7] T. Nakazawa and D. V. Kulkarni, "Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 2, pp. 250–256, 2019.

[8] J. Wang, C. Xu, Z. Yang, J. Zhang, and X. Li, "Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, no. 4, pp. 587–596, 2020.

[9] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 3, pp. 395–402, 2018.

[10] https://github.com/Junliangwangdhu/WaferMap.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[15] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[19] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 309–314, 2018.

[20] K. Maksim, B. Kirill, Z. Eduard, G. Nikita, B. Aleksandr, L. Arina, S. Vladislav, M. Daniil, and K. Nikolay, "Classification of wafer maps defect based on deep learning methods with small amount of data," in *2019 International Conference on Engineering and Telecommunication (EnT)*. IEEE Dolgoprudny: Russia., 2019, pp. 1–5.

[21] 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 Transactions on Semiconductor Manufacturing*, vol. 28, no. 1, pp. 1–12, 2014.

[22] G. Lo Sciuto, G. Capizzi, R. Shikler, and C. Napoli, "Organic solar cells defects classification by using a new feature extraction algorithm and an ebnn with an innovative pruning algorithm," *International Journal of Intelligent Systems*, vol. 36, no. 6, pp. 2443–2464, 2021.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[25] M. Piao, C. H. Jin, J. Y. Lee, and J.-Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 250–257, 2018.

[26] M. H. Hansen, V. N. Nair, and D. J. Friedman, "Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects," *Technometrics*, vol. 39, no. 3, pp. 241–253, 1997.

[27] 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 Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 315–322, 2018.

[28] Y.-S. Jeong, S.-J. Kim, and M. K. Jeong, "Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping," *IEEE Transactions on Semiconductor manufacturing*, vol. 21, no. 4, pp. 625–637, 2008.

[29] C.-H. Wang, W. Kuo, and H. Bensmail, "Detection and classification of defect patterns on semiconductor wafers," *IIE transactions*, vol. 38, no. 12, pp. 1059–1068, 2006.

[30] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[31] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[32] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[33] https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2.

[34] "F1/dice-score vs iou," https://stats.stackexchange.com/q/276144.