Enhancing Generalization of Wafer Defect Detection by Data Discrepancy-aware Preprocessing and Contrast-varied Augmentation

Chaofei Yang, Hai Li, Yiran Chen

Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708
{chaofei.yang, hai.li, yiran.chen}@duke.edu

Jiang Hu

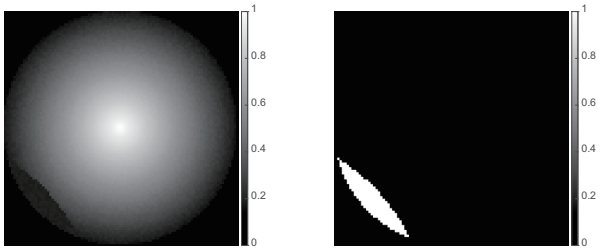
Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX 77843
jianghu@tamu.edu

Abstract— Wafer inspection locates defects at early fabrication stages and traditionally focuses on pixel-level defects. However, there are very few solutions that can effectively detect large-scale defects. In this work, we leverage Convolutional Neural Networks (CNNs) to automate the wafer inspection process and propose several techniques to preprocess and augment wafer images for enhancing our model's generalization on unseen wafers (e.g., from other fabs). Cross-fab experimental results of both wafer-level and pixel-level detections show that the F1 score increases from 0.09 to 0.77 and the Precision-Recall area under curve (PR AUC) increases from 0.03 to 0.62 using our proposed method.

I. Introduction

Wafer inspection is an important step in semiconductor chip fabrication. Defects of interest (DOIs) of wafers are detected to ensure the yield of clean wafers. This operation is executed multiple times at different stages throughout the whole fabrication process. Wafer inspection is split into two major stages, namely, unpatterned wafer inspection and patterned wafer inspection [1]. Unpatterned wafers, or bare wafers, are inspected by wafer manufacturers. These wafers are visually uniform across their surfaces. In this stage, there are no circuit patterns or dies yet on the wafers. The roughness of the wafers is examined by projecting a laser beam. Patterned wafer inspection is performed by chipmakers. In this stage, there is already some circuitry on the wafers. Pattern images between adjacent dies are compared and the difference is used to detect defects [2]. In this work, we focus on a specific DOI named *entry transition signature*, which happens when the wafer is entering the equipment. Figure 1 illustrates this DOI. Note that the wafer data used in our experiments come from a leading semiconductor manufacturing company (hereinafter called *the Company*) and is subject to the constraints of Non-Disclosure Agreement. Hence, we can only conceptually show a manually generated pseudo wafer image here. Wafer-level defect detection is a challenging task because of the vast amount of defects, which vary from layer to layer and fab to fab. As a result, handcrafting the required detection recipes is impractical.

In recent years, Convolutional Neural Networks (CNNs) have demonstrated impressive performance in computer vision tasks [3]. This observation motivated us to apply CNNs to automate the wafer defect detection process and improve its efficiency. However, the data obtained from different fabs are so diverse that the generalization of a CNN model trained by standard normalization and augmentation methods (hereinafter called the *standard method*) is generally poor. That is, the model trained from the fab1 data can not successfully predict the defects of the fab2 data. The size of our wafer dataset is also small, which further limits the training quality of the model.



(a) (b)

Fig. 1: Illustration of the entry transition signature: (a) pseudo wafer image; (b) mask of the DOI.

In order to address these issues, we propose several methods to preprocess and augment the wafer images used for model training. For the preprocessing, we propose an outlier-excluded adaptive clipping method to iteratively calculate the real mean and variance and clip the pixels beyond a threshold. We also combine the masked normalization with clipping to avoid the interference of background pixels. For the augmentation, we propose to leverage gamma correction [4] to alter the contrast of the wafer images. We also apply the random rotation by considering the geometric shape of the wafer. We perform our experiments on the wafer images obtained from two fabs. We achieve a 97.78% accuracy on within-fab wafer defect detection using *the Company*'s CNN-based in-house model. As a variation of Fully Convolutional Network (FCN), this model was originally developed by *the Company* for the study of the patterned wafer defect detection. We also increase the F1 score from 0.09 to 0.77 and the PR AUC from 0.03 to 0.62 on cross-fab wafer defect detection by replacing the standard method with our proposed method. Our experiments also confirm the extensibility of the proposed method to other segmentation models such as FCN [5] and Mask R-CNN [6].

978-1-7281-4123-7/20/$31.00

II. Preliminary

A. Wafer Defect Detection

Wafer images can be generated by three methods: electron-beam, brightfield inspection, and darkfield inspection [7], [8]. After obtaining the wafer images, we can apply customized algorithms to identify defective areas.

In this work, we use the darkfield inspection to generate the wafer haze maps, i.e., the aforementioned wafer images. However, our developed model and method can be easily extended to other inspection methods with necessary customization. The defects to be detected are generated from a Copper Electroplating (CuECP) process, which causes phenomenal yield issues in both IC and memory fabs such as Micron, Global Foundries, Samsung, and TSMC. There are several types of signatures reported from the field in the CuECP layer. Out of the remaining haze map signatures, we focus on the *entry transition signature* for the following two reasons: First, this signature commonly exists in both IC and memory fabs; second, among all the signatures, the entry transition signature appears first when the process quality degrades. Thus, an accurate detection of this DOI allows customers to fix process drift at the earliest possible time.

B. Convolutional Neural Networks and Semantic Segmentation

CNNs are capable to effectively extract spatial features of images and achieve state-of-the-art performance in computer vision applications. Among these applications, semantic segmentation segments an image into different areas each of which represents a different class. As a variation of CNN, FCN [5] is an end-to-end, pixel-to-pixel structure that is widely used in semantic segmentation. FCN adds a skip architecture to the base CNN model to combine information from various layers. This information is then sent to the deconvolution layers to perform semantic segmentation. Mask R-CNN (Region-based Convolutional Neural Network) [6] adds a set of mask predictions in parallel to a set of bounding box predictions and obtains state-of-the-art results on instance segmentation. In our experiments, we evaluate the above models since wafer defect detection can be also considered as a semantic segmentation problem, where the wafer haze map is segmented into the background and defective regions.

C. Related Works

There has been a growing body of research on wafer defect detection over the recent decade [10], [11], [14]. However, these works mostly focused on the detection of pixel-level defects and used traditional image analyzing tools to achieve this goal. Machine learning models such as CNNs were also adopted in some other works [12], [13]. Despite of the success in these works, we haven't seen any techniques for detecting larger defects in fine-grained wafer images were reported. Furthermore, there are yet no implementations of state-of-the-art deep learning models.

III. Methodology

A. Motivation

In this work, we obtained wafer haze maps from two fabs and partition them into training, validation, and testing datasets. As we will show in Section IV-B, *the Company*'s model trained with the standard method has limited generalization to handle cross-fab data: when the training and testing datasets are from the same fab, i.e., within-fab, the results of the accuracy, F1 score, and PR AUC are reasonably good. However, when the training and the testing datasets are from different fabs, i.e., cross-fab, all these metrics are far from unsatisfactory. In addition, the amount of training data is usually limited, which further constrains the training quality of the defect detection model. We carefully examined the obtained wafer haze maps from the fabs and made the following observations:

- The wafer area is circular. The surrounding background area does not contribute to the training quality of the model.
- The DOI is contrast-sensitive: When the contrast between the DOI area and the normal area is low, i.e., the average intensity of the pixels in the DOI area is close to the one of the normal area, it is very hard to detect such a DOI.
- The intensity range of the wafer pixels is wide. Additionally, the distribution of wafer pixels in the low (high) intensity region is dense (sparse).
- The distribution parameters vary between the fabs, including the intensity range and the sparsity.

The goal of this work is to develop an approach to effectively detect wafer defects based on CNN models by leveraging the above observations. In Section III-B, two data preprocessing techniques are proposed to address the discrepancy between the data obtained from different fabs. In Section III-C, two data augmentation techniques are developed to deal with the limited size of the training dataset. As we shall show in our experiments, the combination of both data preprocessing and augmentation methods will greatly enhance the generalization of the CNN models for wafer defect detection.

B. Data Preprocessing

Because the raw data is usually biased, noisy, and has large variance, data preprocessing is commonly adopted in machine learning applications. Normalization is a standard procedure of the data preprocessing which subtracts the mean from the data and divide the data by the standard variance to generate the normalized data with zero mean and unit variance. Such a standard normalization process, however, cannot be directly applied to wafer defect detection. Figure 2 (a) and (b) compare the wafer haze map histograms before and after applying the standard normalization. Here the y-axis is displayed in logarithmic scale. The normalized wafer haze maps from fab1 and fab2 are still very different: The pixels of fab2's wafers have a more sparse distribution in the high-intensity region than the pixels of fab1's wafers and the pixels of fab2 have a wider intensity range. This comparison indicates that the data discrepancy between the wafer haze maps from different fabs cannot be eliminated by the standard normalization. Thus, we propose the so-called data discrepancy-aware preprocessing to address this issue.

1) Masked normalization: As shown in Figure 1, even though the haze map is square, the effective area of the wafer, i.e., the wafer area, is circular. The rest of the area is called the background. This fact is also reflected in Figure 2 (a) where

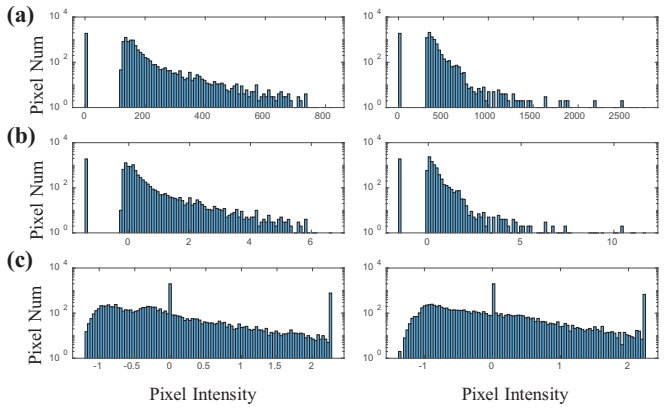


Fig. 2: Histogram comparison (left: fab1 wafers, right: fab2 wafers): (a) original wafer haze maps; (b) haze maps after the standard normalization; (c) haze maps after the proposed preprocessing method.

the pixels at zero intensity indeed represent the background area. Since the background pixels do not contribute to the training of the model, we want to keep the intensity of these pixels as zeros as the inputs of the CNN-based model. The standard normalization, however, shifts the intensity of the background pixels to negative values. We propose the following procedure to manage the background pixels in particular:

(i) Detect the background pixels and obtain a mask for them;
(ii) Exclude the masked pixels from the pending wafer haze map (say, valid data);
(iii) Perform the normalization (standard or proposed) on the valid data only;
(iv) Attach the pixels in the background mask and set their intensities to zero.

As a result, only the pixels of the wafer area will be processed in the feature extraction phase using CNNs.

2) Outlier-excluded adaptive wafer haze map clipping: Figure 2 (a) shows that although the intensity range of the pixels is wide, there exist some pixels whose intensities are significantly higher than that of most of the pixels. This observation is particularly obvious in fab2. We call such pixels "outliers", which need to be properly handled in data preprocessing. The presence of these outliers may due to various reasons such as fabrication discrepancy and process variation. Note that the distributions of these outliers are different between fab1 to fab2. In addition, the intensity range of fab2 pixels is significantly larger than the one of fab1 ($\sim 3.1\times$).

Our proposed solution of the above issues is summarized in Algorithm 1. First, we obtain the mask for the whole wafer area and shrink the mask by $mask\_th = 2$ pixels. This is because we find that the boundary of the wafer area is a mix of high- and low-intensity pixels, which affect the detection of the defect boundary. Note that this operation will not interfere the detection since the defective area is usually much wider. Excluding these pixels also helps the calculation of the mean and the variance. Second, within the valid data (masked area), we iteratively calculate the current mean and variance, and

Algorithm 1 Outlier-excluded Adaptive Clipping with Masked Normalization.

Input: Wafer haze map $\boldsymbol{x}_i \in \mathbb{R}^{dim}$ (dim is the dimension of the data, in our case is 98×98, $i \in [1, training\_set\_size]$). Iteration number $iter_{num} = 10$. Clipping threshold $clip\_\sigma = 3$. Outlier exclusion threshold $outlier\_\sigma = 4$. Mask shrink threshold $mask\_th = 2$.
Do:
1. Calculate the mask for wafer area $mask$.
2. Shrink $mask$ by $mask\_th$ pixels.
3. Calculate the real mean μ_{x_i} and the standard variance σ_{x_i} by excluding outliers:
Set $data\_valid$ as pixels in $mask$ of $\boldsymbol{x}_i$.
for $iter = 0$; $iter < iter\_num$; $iter++$ **do**
  a. Calculate μ_{x_i} and σ_{x_i} of $data\_valid$.
  b. Remove pixels outside $(\mu_{x_i} - outlier\_\sigma \cdot \sigma_{x_i}, \mu_{x_i} + outlier\_\sigma \cdot \sigma_{x_i})$ from $data\_valid$.
end for
4. $\boldsymbol{x}_i[\boldsymbol{x}_i > \mu_{x_i} + clip\_\sigma \cdot \sigma_{x_i}] = \mu_{x_i} + clip\_\sigma \cdot \sigma_{x_i}$
  $\boldsymbol{x}_i[\boldsymbol{x}_i < \mu_{x_i} - clip\_\sigma \cdot \sigma_{x_i}] = \mu_{x_i} - clip\_\sigma \cdot \sigma_{x_i}$
6. Apply the standard normalization on $\boldsymbol{x}_i$.
7. Set pixels in the background area as zeros.
Output: The preprocessed wafer haze map $\hat{\boldsymbol{x}}_i$.

then remove the pixels we considered as the outliers, that is, their intensities are larger or smaller than certain thresholds. After several iterations, we exclude all outliers and obtain the real mean and variance. At the end, we clip the wafer area based on these statistics, apply the standard normalization, and attach the background zeros.

After the clipping, the processed wafer haze maps are distributed over about the same intensity range and maintain the similar features, as shown in Figure 2 (c).

C. Data Augmentation

Data augmentation is often used when training samples are very limited either in *quantity* or *variety*, which is exactly our case. In this work, we propose two augmentation techniques that are customized to wafer haze maps.

1) Random rotation: To address the limitation in *quantity*, we propose random rotation. Since the valid wafer area is almost a perfect circle, we can rotate the wafer area in random degrees to enrich the spatial information of the wafer data. Thus the segmentation result will not overfit specific spatial features of wafer data in the training dataset, e.g., defects are only present at specific angles. The rotation operation can be performed by applying the following formulas:

$$\begin{aligned} x_2 &= cos(\theta)(x_1 - x_0) - sin(\theta)(y_1 - y_0) + x_0 \\ y_2 &= sin(\theta)(x_1 - x_0) + cos(\theta)(y_1 - y_0) + y_0, \end{aligned} \tag{1}$$

where (x_0, y_0) are the coordinates of the center of the rotation, θ is the angle of the rotation, (x_1, y_1) is the source pixel (the pixel before the rotation) and (x_2, y_2) is the destination pixel (the pixel after the rotation). In this way, every destination pixel is assigned to a source pixel. Note that if the destination pixel is not assigned to any source pixel (due to the sin and cos functions), it will create a "hole" such as $x_2 = y_2 = 0$. We propose to use bilinear interpolation [9] to solve this issue, i.e., assigning an interpolant to the destination pixel when the above scenario occurs. Bilinear interpolation can be expressed as the following equation:

$$f(x,y) = \frac{\begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(x_1,y_1) & f(x_1,y_2) \\ f(x_2,y_1) & f(x_2,y_2) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix}}{(x_2 - x_1)(y_2 - y_1)}, \tag{2}$$

where $f(x,y)$ is the pixel value at point (x,y) and (x_i, y_j) is one of the four points used for interpolation.

Other interpolation algorithms such as bicubic interpolation and spline interpolation require information from more surrounding pixels thus become more computationally intensive. The results, however are similar to that of bilinear interpolation in this case.

2) Random gamma correction: To address the limitation in *variety*, we propose to use random gamma correction to augment the wafer data. Gamma correction is widely used to encode and decode luminance values in videos or images to compensate the difference between the perceptions of human eyes and camera sensors [4]. Gamma correction, or gamma encoding, is a nonlinear operation defined by the following expression:

$$V_{out} = \alpha V_{in}^{\gamma}, \tag{3}$$

where V_{in} and V_{out} are pixel values of the images before and after gamma correction. α is a constant and γ is the gamma index. Usually $\alpha = 1$ and $V_{in}, V_{out} \in (0,1)$.

We perform experiments on pseudo wafer haze maps that are similar to the ones in Figure 1, to illustrate how gamma correction works. Figure 3 shows the experiment results adopting $\gamma = 1, 0.5, 1.5$, separately. The range of the chosen gamma is larger than the one we use in the experiment, i.e., $(0.5, 1.5)$ vs. $(0.75, 1.25)$, for a better illustration. Both haze map and histogram comparisons are included. The haze map becomes brighter or darker when $\gamma < 1$ or $\gamma > 1$, respectively. The consequence is also illustrated as the shift of pixels' intensities in the histograms. This result indicates that gamma correction can effectively alter the contrast of wafer haze maps by applying different γs without changing spatial features. On the contrary, other pixel-level operations such as adding Gaussian noise either decreases the quality of the image or incurs distortion. Since entry transition signatures are contrast-sensitive, augmenting data with gamma corrections can greatly improve the variety of the data and thus enhance the model's robustness. We also observe in the experiment results that gamma correction can substantially increase PR AUC, which means the pixel-level segmentation is greatly improved.

By combining random rotation and random gamma correction (i.e., randomly choose γ from $(0.75, 1.25)$), we increase the size of training dataset by $30 \sim 40\times$. Note that we implement online data augmentation, i.e., randomly augmenting the data at the start of each training epoch, to further improve the variety. More details can be found in Section IV.

IV. Experiments

A. Dataset and Evaluation Metrics

The diameter of the wafers we possess is 30cm. We choose to use 3mm pixel haze map, i.e., one pixel represents the average values of a 3×3 mm wafer area, since the entry transition signature is at the macro level. This hypothesis is validated by the comparison study on using 3mm and 600μm pixel haze maps: our results show that these two configurations obtain similar performance. The dimension of the data is 98×98 (not 100×100 because of the special handling at the edge). Our wafer dataset includes the data obtained from two fabs. Note that it took our collaborated company one year to collect these data from customers for this particular task. This fact indicates that the wafer defect detection task typically has a very small amount of training data.

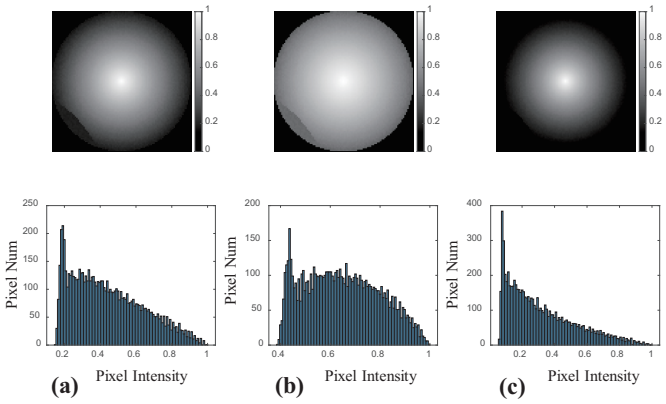


Fig. 3: Comparison of gamma corrected wafer haze maps and corresponding histograms: (a) $\gamma = 1$ (no gamma correction); (b) $\gamma = 0.5$; (c) $\gamma = 1.5$.

A detailed description can be found in Table I. The numbers of the wafers with and without DOIs can be found in the rows named "DOI" and "non-DOI", respectively. Note that data from fab1 are used for both training and testing while the data from fab2 wafers are used only for testing. For fab1 data, we choose to use 40% for training, 20% for validation, and 40% for testing. All the DOI regions of the data are manually annotated by professionals for both training and testing.

In the experiments, we use multiple metrics including recall, precision, F1 score, and accuracy to evaluate the mode's wafer-level performance (classification). We also use PR AUC to evaluate the model's pixel-level performance (segmentation). The PR curve is obtained by applying different thresholds of the segmentation model and indicates the tradeoff between the pixel-level precision and recall.

B. Standard Method - Baseline

We train *the Company*'s model on fab1's data and test it on the data from both fab1 and fab2. We implement the baseline method, i.e., training with the standard method, to demonstrate the effectiveness of applying the CNN model on the wafer defect detection task and its limitation. We include the results on accuracy, F1 score, and PR AUC for a complete analysis.

TABLE I: Wafer dataset description.

	Fab1	Fab2
Total	228	343
DOI	26	59
non-DOI	202	284
Dimension	98×98	98×98
Purpose	training and testing	testing

	No DOI	Has DOI
No DOI detected	79	0
Has DOI detected	2	9

	No DOI	Has DOI
No DOI detected	282	56
Has DOI detected	2	3

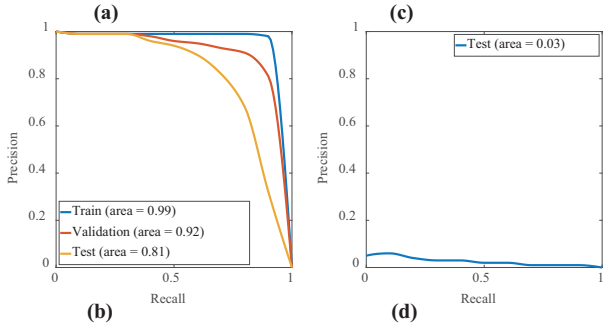


Fig. 4: Result of baseline method. Train on fab1, test on fab 1: (a) confusion matrix; (b) PR curve; train on fab 1, test on fab2: (c) confusion matrix; (d) PR curve.

	No DOI	Has DOI
No DOI detected	80	0
Has DOI detected	1	9

	No DOI	Has DOI
No DOI detected	268	12
Has DOI detected	16	47

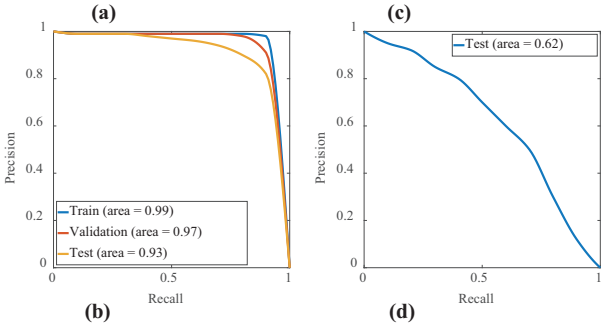


Fig. 5: Result of the proposed method. Train on fab1, test on fab 1: (a) confusion matrix; (b) PR curve; train on fab1, test on fab2: (c) confusion matrix; (d) PR curve.

The confusion matrix and PR curve are summarized in Figure 4, where (a) and (b) are the results tested on fab1's data, (c) and (d) are the results tested on fab2's data. The corresponding metrics can be summarized as the follows:

- For testing on fab1: recall = 100%, precision = 81.82%, accuracy = 97.78%, F1 score = 0.900, PR AUC = 0.81.
- For testing on fab2: recall = 5.09%, precision = 60%, accuracy = 83.09%, F1 score = 0.094, PR AUC = 0.03.

The above results indicate that: 1) *the Company*'s model and the standard method work well in the analysis of the within-fab wafers (e.g., train and test in the same fab), despite of the limited amount of the training data; 2) the model trained with fab1's data using the standard method performs badly on fab2's data (cross-fab data). Even though the accuracy (83.09%) looks reasonable, it is indeed because most of the wafer data are non-DOI data (i.e., 284 out of 343 for fab2's data) and the model by default classifies the wafers as non-DOI (near-zero recall). This also indicates that the accuracy cannot solely provide a useful evaluation when there is a large skew in the class distribution. The F1 score, on the other hand, is a better choice here, taking both the recall and precision into consideration. The low recall (only 5.09%) indicates that the model cannot detect any DOIs on the wafer haze maps in most cases. In addition, the PR AUC is nearly zero. It indicates that pixel-wise segmentation performance is even worse, that is, almost all the pixels are assigned as background with extremely high confidence scores. In summary, the standard data preprocessing and augmentation method work well on within-fab data but not on cross-fab data.

C. Proposed Data Preprocessing and Augmentation Methods

In this section, we summarize our proposed method for the training of *the Company*'s model on both within-fab and cross-fab data. The corresponding confusion matrix and PR curve are shown in Figure 5. By comparing the results in Figure 4 and Figure 5, we can make the following observations:

- For the within-fab data, both the standard method and proposed method achieve similar results. Compared to the standard method, the proposed method has one less false detection and the PR AUC increases from 0.81 to 0.93. The limited improvement is because the original result of the standard method is already very good on this small dataset.
- For the cross-fab data, the proposed method detects many more DOIs than the standard method does. Note that the number of false detection also increases, which slightly affects the precision. The improvement in the defect detection is also reflected by the recall and the F1 score: The recall increases from 5.01% to 79.66% and the F1 score increases from 0.09 to 0.77. The PR AUC also increases from 0.03 to 0.62, indicating a substantial improvement in the pixel-level detection too.

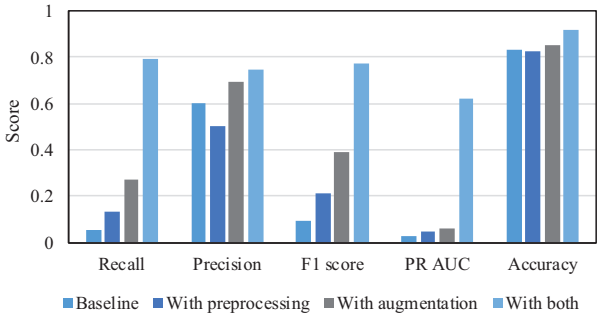


Fig. 6: Comparisons of recall, precision, F1 score, PR AUC, and accuracy under different combinations of the proposed techniques.

Next, in order to individually evaluate the benefit that was introduced by each technique included in the proposed method, we apply different combinations of data preprocessing and data augmentation techniques. The results are summarized in Figure 6. Four experiments are performed with the standard method (*baseline*), the data preprocessing only (*with preprocessing*), the data augmentation only (*with augmentation*), and the combination of these two techniques (*with both*), respectively. We include the same metrics from Section IV-B

TABLE II: Comparison of all metrics for *the Company*'s model, FCN, and Mask R-CNN.

Metrics	*the Company*'s model			FCN [5]			Mask R-CNN [6]		
	baseline	proposed	improvement	baseline	proposed	improvement	baseline	proposed	improvement
Recall	5.09%	79.66%	15.65×	3.39%	76.27%	22.12×	8.48%	87.72%	10.35×
Precision	60.00%	74.60%	1.24×	40.00%	73.77%	1.84×	55.56%	74.63%	1.34×
F1 score	0.09	0.77	8.56×	0.06	0.75	11.81×	0.15	0.81	5.48×
PR AUC	0.03	0.62	20.67×	0.03	0.55	18.33×	0.03	0.66	22.00×
Accuracy	83.09%	91.84%	1.11×	83.09%	92.96%	1.12×	82.75%	91.25%	1.10×

in the results.

From the results, we can see that the data augmentation obtains slightly better results than the data preprocessing. It implies that the quantity/variety of the training data plays a more important role in the training of the model under our setup. Moreover, a reasonable PR AUC is obtained only when all the techniques are applied. Our proposed method also achieves a slightly better accuracy than the baseline. These results convincingly validate the enhanced generalization of the CNN model when our proposed method is applied.

D. Extensibility of the Proposed Method

We also apply other CNN models to perform wafer defect detection with our proposed method to validate its extensibility. The first one is the FCN [5] based on VGG-16 and the second one is the Mask R-CNN [6] based on ResNet101. The same metrics aforementioned in Section IV are used in our experiments here. We also include the improvement of the proposed method over the standard method. Table II summarizes all the results, including the ones of *the Company*'s model for comparison purpose. Note that *the Company*'s model is a variation of the FCN.

As a state-of-the-art CNN model, Mask R-CNN achieves a very good performance in some segmentation competitions. In our wafer defect detection task, however, the performance of Mask R-CNN is only slightly better than that of FCN and *the Company*'s model. It is probably because of the limited size and dimension of the training dataset. Nonetheless, our proposed method still greatly improve the generalization of both Mask R-CNN and FCN. On average, we obtain an $8.50\times$ improvement in the F1 score and $20.33\times$ improvement in the PR AUC across all the models. We believe our proposed method can be extended to other CNN models with necessary customization. Note that even though only entry transition signatures are examined, the proposed method should be easily adapted to other defects with similar spatial features (e.g., area, type of wafer) with no or minor modifications. This is due to the similarity in texture between raw wafer haze maps whose characteristics are captured by our method.

V. Conclusion

In this paper, we demonstrate the feasibility to use state-of-the-art segmentation models for wafer defect detection. Particularly, we study the entry transition signature and achieve an accuracy of 97.78% on within-fab data. We also propose customized preprocessing and augmentation techniques in order to effectively perform defect detection on cross-fab data. Our results show that the proposed method can improve the F1 score and PR AUC by up to $11.81\times$ and $22.00\times$, respectively, indicating a substantially enhanced generalization of the adopted segmentation model. We plan to test our method on more DOIs and continue to improve the performance of the model once more training data are collected from different fabs.

Acknowledgments

This work is partially supported by Semiconductor Research Corporation Tasks 2810.022 through UT Dallas' Texas Analog Center of Excellence (TxACE) and NSF-1822085 through NSF IUCRC for Alternative Sustainable and Intelligent Computing (ASIC).

References

[1] Christopher F Bevis *et al.*, "Systems for inspection of patterned or unpatterned wafers and other specimen," *Google Patents*, US Patent 7,068,363, 2006.

[2] Haruo Yoda *et al.*, "An automatic wafer inspection system using pipelined image processing techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no.1, pp. 4–16, 1998.

[3] Alex krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[4] Shih-Chia Huang *et al.*, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1032–1041, 2013.

[5] Jonathan Long *et al.*, "Fully convolutional networks for semantic segmentation," *Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[6] Kaiming He *et al.*, "Mask r-cnn," *International Conference on Computer Vision*, pp. 2980–2988, 2017.

[7] Kenneth W Tobin *et al.*, "Inspection in semiconductor manufacturing," *Webster's Encyclopedia of Electrical and Electronic Engineering*, vol. 10, pp. 242–262, 1999.

[8] Christopher R Fairley*et al.*, "High throughput brightfield/darkfield wafer inspection system using advanced optical techniques," *US Patent* 6,288,78, 2001.

[9] Shengkui Gao *et al.*, "Bilinear and bicubic interpolation methods for division of focal plane polarimeters," *Optics Express*, vol. 19, no. 27, pp. 26161–26173, 2011.

[10] Wangyang Zhang *et al.*, "Multi-wafer virtual probe: minimum-cost variation characterization by exploring wafer-to-wafer correlation," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 47–54, 2010.

[11] Kaoru Sakai *et al.*, "Defect detection method using statistical image processing of scanning acoustic tomography," *International Symposium on the Physical and Failure Analysis of Integrated Circuits*, pp. 293–296, 2016.

[12] Je-Kang Park *et al.*, "Machine learning-based imaging system for surface defect inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303–310, 2016.

[13] Takeshi Nakazawa *et al.*, "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.

[14] Jianbo Yu *et al.*, "Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis," *IEEE Transactions on Semiconductor Manufacturing*, vol. 29, no. 1, pp. 33–43, 2016.