Multiscale Science and Engineering
https://doi.org/10.1007/s42493-024-00103-z

ORIGINAL RESEARCH

# Deep Learning-Based Detection of Defects in Wafer Buffer Zone During Semiconductor Packaging Process

Tae-Yeon Kim[1] · Sunjae Park[1] · Chang-Kue Lim[2] · Jae-Hyuk Choi[2] · Chang-Hyun An[2] · Lak-Hyun Song[2] · Do-Nyun Kim[1,3,4]

Received: 5 December 2023 / Revised: 5 December 2023 / Accepted: 5 January 2024
© The Author(s) 2024

**Abstract**

Effectively detecting defects in wafer buffer zones is crucial in semiconductor manufacturing processes. If defects are not detected in the middle of the semiconductor manufacturing process, the semiconductor eventually becomes a defective product after all processes are completed. To solve this problem, rule-based vision algorithms have been used to identify defects in the wafer buffer zone. After photographing the wafer buffer zone using a high-speed camera, defects are detected using the pixel values of the images. However, because of the resin bleed in the wafer, which is an epoxy compound, it is difficult to detect defects. Therefore, we introduced a deep learning method for semiconductor inspection and created a high-performance semiconductor inspection algorithm. The defects in the wafer buffer zone should be detected accurately and quickly. Furthermore, the approximate size of the defects must be extracted. We modified the Xception model to fit the wafer data characteristics considering both accuracy and speed. We proposed to extract the size of the defect using class activation mapping (CAM). We obtained an accuracy of 96.9% from the actual wafer dataset through this framework, and then managed to extract the size of the defect through CAM.

**Keywords** Wafer · Defect · Classification · Inspection · Deep learning

## Introduction

Semiconductor wafers undergo various packaging processes and unwanted defects must be accurately and quickly detected during each process. However, epoxy molding compounds (EMC) used to protect semiconductor devices from external impact, vibration, moisture, and radiation as well as the protection cover in the bottom of wafers make the visual and in-process detection of defects difficult (Fig. 1). While defects in wafers can be checked after all processes are completed, the manufacturing loss due to reduced yield becomes non-negligible. Fortunately, there is a narrow region not covered with EMC mold called the wafer buffer zone which can be observed during the process. Hence, real-time inspection of this zone for detecting wafer defects during the process is important.

Images obtained from the inspection of the wafer buffer zone can be classified into four types: Normal, Crack,

✉ Do-Nyun Kim
dnkim@snu.ac.kr

Tae-Yeon Kim
kimty1008@snu.ac.kr

Sunjae Park
huchi7669@gmail.com

Chang-Kue Lim
lcksms@ktsemi.co.kr

Jae-Hyuk Choi
jhchoi@ktsemi.co.kr

Chang-Hyun An
chan@ktsemi.co.kr

Lak-Hyun Song
nhsong@ktsemi.co.kr

[1] Department of Mechanical Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

[2] Korea Techno, 673-4, Opo-ro, Opo-eup, Gwangju-si, Gyeonggi-do 12790, Republic of Korea

[3] Institute of Advanced Machines and Design, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

[4] Institute of Engineering Research, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

Published online: 14 February 2024

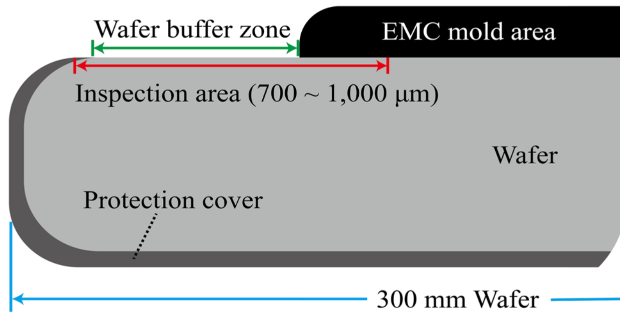



**Fig. 1** Schematic representation of inspection area

EMC defect, and Notch (Fig. 2). Notch (Fig. 2a) is the reference point of the wafer and hence is not a defect. We would like to detect cracks and EMC defects when inspecting the wafer buffer zone during the packaging processes. It is relatively easy to detect cracks when they exist on a clean surface (Fig. 2b). However, the detection becomes difficult when cracks appear on a surface with resin bleeding (Fig. 2c), especially if they are relatively small. EMC defects occur when they are not properly cover the mold area where the semiconductor die is located (Fig. 2d, e). If EMC excessively invades the buffer zone, the protecting cover can stick to it and cause damages to the wafer. If EMC in the mold area is insufficient, the semiconductor die cannot be fully protected. These EMC defects are not easy to be detected as they can be confused with notch or resin bleeding, which is normal.

For the detection of these wafer defect, rule-based computer vision algorithms can be used. For example, the Sobel operator can be used to detect edges of images and identify defects [1]. However, it is difficult to extract cracks using this operator if the wafer buffer zone is covered with resin bleeding because too many edges are extracted. Another popular approach is the Canny algorithm which extracts the meaningful edges by thresholding after denoising images using Gaussian filters [2]. While this algorithm generally provides better results than the Sobel operator, it suffers from the same difficulty in distinguishing cracks from resin bleeding.

Recently, various deep learning models that use high level features have been developed in computer vision. For example, object detection models recognize an object from an image by indicating its location in a bounding box [3–10]. Semantic segmentation models classify objects more densely than the detection model using the pixels within an image [11–15]. While we can know the location and size of defects with these models, they require bounding-box annotations or segmented labels for training. Also, these models are heavy in terms of the number of model parameters and require a long time for training and inference. Since the detection of wafer defects needs to be done quickly during the packaging processes, these models cannot be used directly in practice.

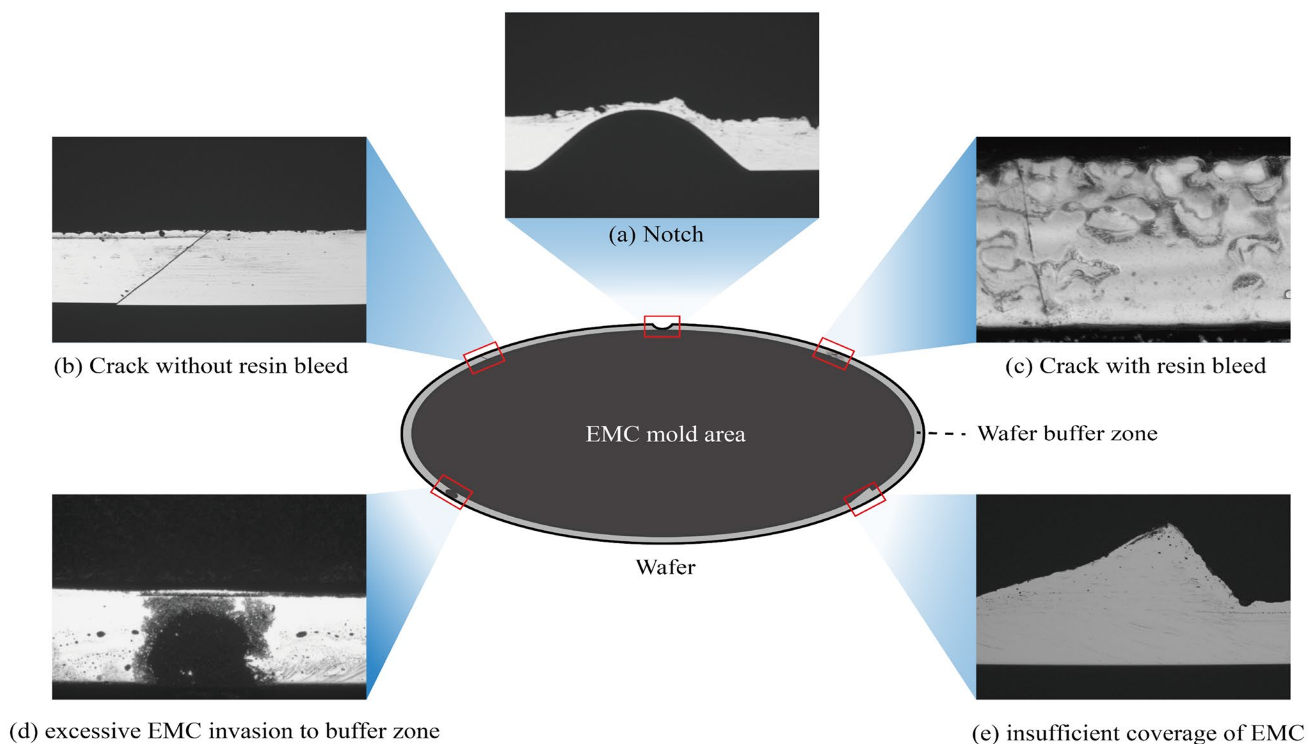



**Fig. 2** Types of images obtained during inspection

Alternatively, we can use the classification model which classifies images into the defined types [16–18]. It is lighter than detection and semantic segmentation models, and hence, is advantageous in terms of speed. In addition, there is no need to annotate labels in images when building the datasets for training. However, this model is unable to localize the defects within an image and measure their characteristic features such as the size and the area.

In this study, we develop a deep learning-based inspection method for detecting defects in the wafer buffer zone. The model was designed to find and localize a defect quickly, and infer its size as well (Fig. 3). We employed a classification model, Xception [19], and modified it to be suitable for inspecting the wafer buffer zone. To accelerate the inspection, we changed eight repetitions of middle flow of the original Xception model to one. In addition, the feature pyramid network (FPN) [20] was used to effectively handle various sizes of defects. We utilized the class activation map (CAM) [21] to generate a heat map of a specific class image, and hence, estimate the size of defects without any additional supervised learning. The length of cracks and the area of EMC defects can be approximately obtained while inspecting the wafer buffer zone. The proposed model showed higher detection accuracy and faster inference speed than baseline models.

## Dataset

We constructed a dataset by taking images of the wafer buffer zone for 300 mm wafers. Each wafer was placed on a YASKAWA pre-aligner that held it using the vacuum chuck method and rotated it once at a high speed. A high-speed camera (Basler acA1300-200 μm, Mono, 203 fps) synchronized with the rotational speed of wafers was used to photograph the wafer buffer zone. An area covering approximately 700–1000 μm from the wafer edge was inspected. We obtained 250–255 images per wafer for 51 wafers leading to 12,869 images in total. They were divided into 12,381 training and 488 test images (Table 1). The pixel size of each image is 1080 × 1440.

**Table 1** Composition of training and test dataset

| | Training | Test |
|---|---|---|
| Crack | 1047 | 105 |
| EMC defect | 905 | 107 |
| Normal | 10,322 | 262 |
| Notch | 107 | 16 |

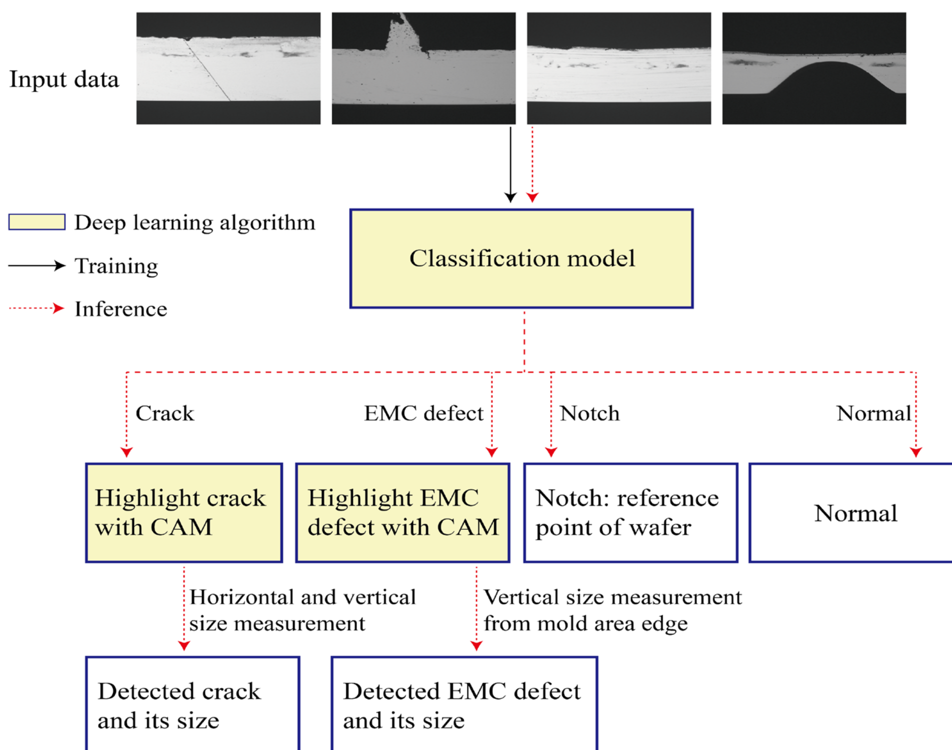
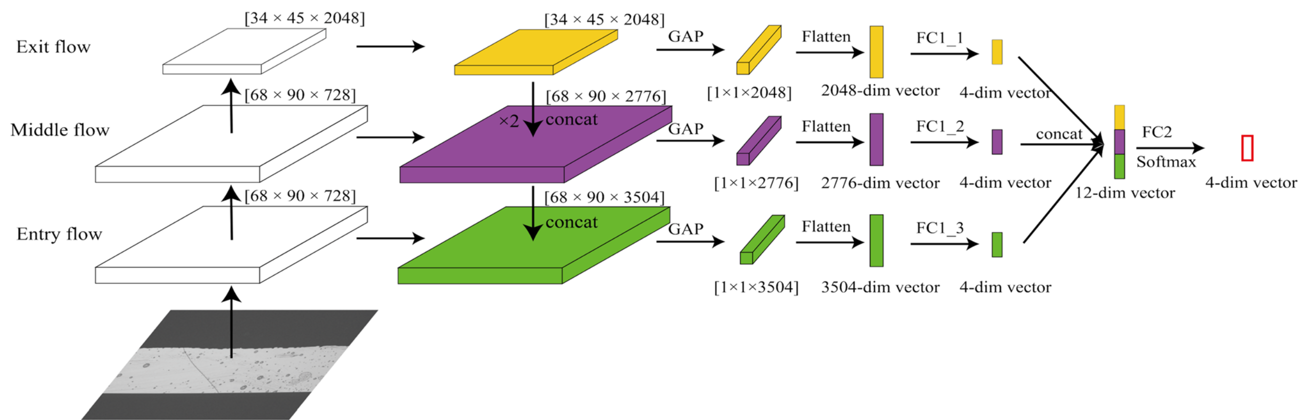



**Fig. 3** Overall workflow. The training flow and inference flow are represented with black solid lines and red dashed lines, respectively



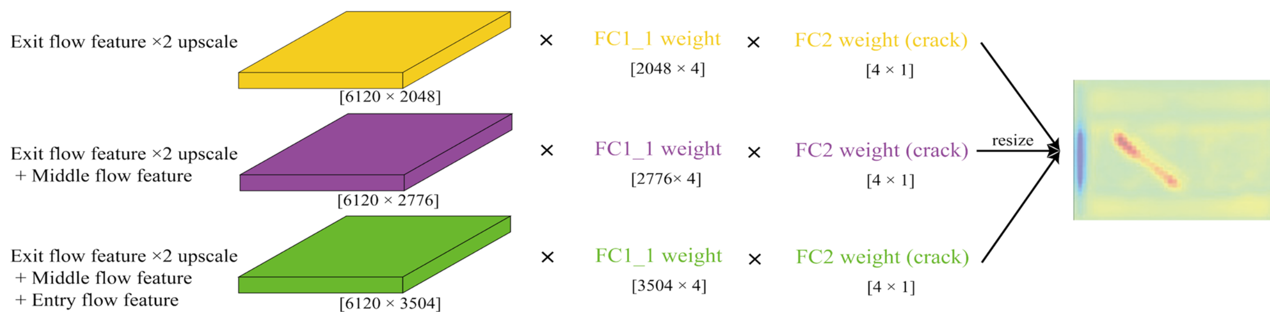




**Fig. 4** Our framework structure. Crack is determined in **a**. And then using feature maps in **a**, class activation map is obtained through **b**

## Method

Inspection model consists of the image classification for identifying defects and the estimation of their size (Fig. 4). The Xception deep learning model was employed for classification with modification to enhance the efficiency in detecting defects. The class imbalance problem was alleviated by adopting the focal loss [8]. The identified region of defects in the wafer buffer zone was inspected with CAM to estimate the size of cracks and EMC defects.

### Classification for Identifying Defects

Several classification models have been proposed by considering both accuracy and speed of inference [22–24]. We used Xception as the base structure for our classification model as it is effective to enhance the accuracy without increasing the model capacity. This model uses a depthwise separable convolution layer to completely separate cross-channel correlation and spatial correlation. Through decoupling channel from spatial, the computation time and parameter were reduced compared to the conventional convolution method. So Xception showed a better performance than Inception V3 [22] while it has a similar number of parameters. Furthermore, Xception has eight repetitions of the middle flow with depthwise separable convolution layer enabling it to learn richer features and its high performance has been demonstrated for several datasets such as ImageNet dataset with 1000 classes [25] and JFT dataset with 17,000 classes [26].

Unlike other public datasets, our wafer buffer zone dataset has four classes only. As a result, we do not need to

**Table 2** Results according to the number of iterations of the middle flow on test dataset

| Number of iteration | Accuracy | Parameter | Inference time |
|---|---|---|---|
| 1 | 94.3% | 9,495,564 | 24.73 s |
| 2 | 94.7% | 11,105,172 | 26.12 s |
| 3 | 94.9% | 12,714,780 | 26.82 s |
| 4 | 94.9% | 14,324,388 | 28.48 s |
| 5 | 96.3% | 15,933,996 | 31.77 s |
| 6 | 96.3% | 17,543,604 | 32.46 s |
| 7 | 96.3% | 19,153,212 | 34.07 s |
| 8 | 96.3% | 20,813,076 | 35.09 s |

**Table 3** Model comparison results on test dataset

| Model | Accuracy | Parameter | Inference time |
|---|---|---|---|
| ResNet-101 | 94.1% | 42,508,356 | 52.35 s |
| Xception | 96.3% | 20,813,076 | 35.09 s |
| EfficientNet-B4 | 96.9% | 17,555,788 | 40.98 s |
| Our model | **96.9%** | **9,520,744** | **24.96 s** |

Bold values indicate the best accuracy with the smallest number of model parameters and inference time among tested models

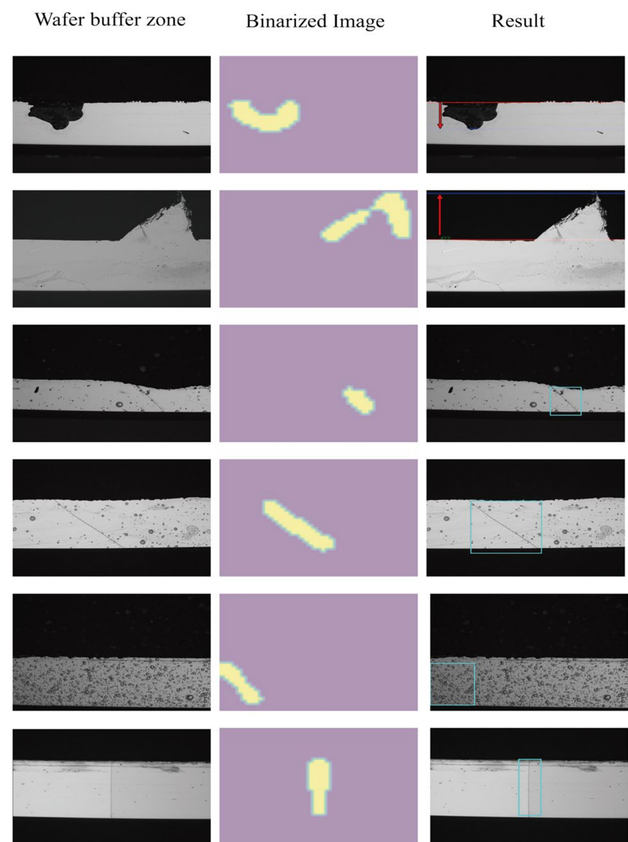



**Fig. 5** Schematic representation of inspection area

use an iterative structure of the middle flow in the original Xception model for classification. We tested the model by changing the number of iterations of the middle flow from eight to one. While the decrease in the model accuracy was acceptable (from 96.3 to 94.3%), a significant reduction in the number of parameters (from 20,813,076 to 9,520,744) and hence in the inference time (from 35.09 to 24.73 s) could be achieved without using the iterative structure (Table 2). Hence, the revised model for our dataset uses the middle flow only once without repetition.

The major difficulty in classifying defects in the wafer buffer zone is that the size of defects varies a lot. Cracks are usually thin and small while EMC defects are large to occupy most part of an image. Various methods have been proposed to improve the detection of objects with various sizes. For example, Lin et al. presented FPN to recognize objects of various sizes while consuming fewer computing resources [20]. FPN was able to extract features of multi sizes using image pyramid. But this structure is for object detection not for classification. This is particularly created for object detection task, where objects can appear at different scales and resolutions in the input image. We incorporated FPN into our modified Xception model to make it be able to respond to various defect sizes while using as small memory as possible for classification problem. Feature maps were extracted from the entry, middle, and exit flows (Fig. 4a). The exit flow feature map doubled in its size was merged into the middle flow feature map, which was then merged into the entry flow feature map again. Each of these combined feature maps went through global average pooling (GAP) and fully connected (FC) layers. Thereafter, the final classification results are presented by combining results of all FC layers.

**Table 5** Results on defect size estimation

| Defect | MAPE |
|---|---|
| Length of crack | 20.92% |
| Invasion depth of EMC defect | 18.77% |

Finally, we adopted the focal loss to address the problem of class imbalance. Actual industrial data often shows a severe class imbalance because the number of defective cases is much smaller than that of normal ones. Our dataset suffers from the same issue as it has 10,707 normal images (normal and notch cases) and 2164 defect images (crack and EMC defect cases). The focal loss relieves it by putting a higher weight to defect images more difficult to be classified

**Table 4** Recall and precision results on test dataset

| Model | Crack | | EMC defect | | Normal | | Notch | |
|---|---|---|---|---|---|---|---|---|
| | Recall | Precision | Recall | Precision | Recall | Precision | Recall | Precision |
| ResNet-101 | 85.7% | 98.9% | 88.8% | 97.9% | 99.6% | 91.6% | 93.8% | 100% |
| Xception | 85.7% | 98.9% | 98.1% | 100% | 99.6% | 94.2% | 100% | 94.1% |
| EfficientNet-B4 | 97.1% | 90.3% | 99.1% | 100% | 95.8% | 98.4% | 100% | 100% |
| Our Model | **98.1%** | **89.6%** | **99.1%** | **100%** | **95.4%** | **98.8%** | **100%** | **100%** |

Bold values indicate the model with the highest recall scores for detecting crack and EMC defects

**Table 6** Ablation study for FPN

| FPN | Accuracy | Parameter | Inference time |
|---|---|---|---|
| No | 94.3% | 9,495,564 | 24.73 s |
| Yes | **96.9%** | **9,520,744** | **24.96 s** |

Bold values indicate the effect of employing FPN on accuracy, model size, and inference time

correctly while giving a lower weight to normal images. As a result, the model can intensively learn the features of defect images with a relatively small number of data.

### Size Estimation of Defects

After identifying defects using the classification model, the size of defects should be estimated in order to check their severity. We need to measure the horizontal and vertical lengths of cracks and the invasion depth for EMC defects. We applied the CAM for this purpose, showing which region of the input image was mainly viewed by the model when it determined the class. In the classification model, the last feature map before GAP contains the most important information of each class. By multiplying the weight of FC layer and the feature maps of a specific class, regions mostly influencing the decision of that class appear in the heatmap. In our case, when a wafer buffer zone image was classified as defective by the classification part of the model, the feature maps from each scale were multiplied by the weights of corresponding FC layers. Then, their summation represented a heatmap revealing the most crucial parts for defect identification or defects.

We estimated the size of defects from binarized heatmaps. In the case of cracks, the largest grain was selected from the binarized image, and its horizontal and vertical lengths were measured. In the case of EMC defects, the vertical distance (or depth) between the grain and the EMC mold layer edge was measured.

## Experiments

Our classification model was trained using a real industrial wafer buffer-zone dataset. The 12,381 train set and 490 test set consisted of cracks, EMC defects, notches, and normal. Because there were significantly more normal images than other classes, many normal images were not included in the test set. The detailed data configuration is presented in [27]. Single RTX 3080 was used.

**Table 8** Ablation study for Focal loss

| Focal loss | Accuracy | Crack recall | EMC defect recall |
|---|---|---|---|
| No | 96.1% | 87.6% | 97.2% |
| Yes | **96.9%** | **98.1%** | **99.1%** |

Bold values indicate the effect of using focal loss on accuracy and recall

### Classification Performance

We used accuracy, recall, and precision, which are widely used for classification models, to measure the performance of our model. Accuracy is the value obtained by dividing the number of correct answers by the total number of test sets. Recall is the ratio of what the model predicts to be true among what is true. Precision is the ratio of what the model classifies as true to what is classifies as true. To compare the speed of the model, the number of parameters of the model and inference time for the test set were compared. The inference time was measured as the time taken from the start of the model to produce the final result of the test set.

We chose EfficientNet-B4, Xception, and ResNet-101 as the baseline models. Since the number of parameters increases from the model above EfficientNet-B5, EfficientNet-B4 is a practical model owing to its good memory efficiency and accuracy. ResNet-101 uses skip connections to create a deep neural network, and is used as a backbone for many deep learning models with high accuracy. We also proposed an efficient model for wafer buffer zone detection. Therefore, it performed better than the other models in terms of accuracy, model parameters, and inference time. First of all, ResNet-101 accuracy was 94.1%, Xception was 96.3%, and EfficientNet-B4 was 96.9%, so our model was better in accuracy than ResNet-101 and Xception, and we obtained the same result as EfficientNet-B4. But even with the same accuracy, recall score is more important. In particular, the defect recall score should be considered acceptable. This is because there is a large loss to be obtained after classifying the defect as normal. The performance of our model was also good for both the crack recall score and EMC defect recall score than other baseline models. Second, it is a lighter model when the parameter number is small, so the inference

**Table 7** Recall and precision results on test dataset

| FPN | Crack | | EMC defect | | Normal | | Notch | |
|---|---|---|---|---|---|---|---|---|
| | Recall | Precision | Recall | Precision | Recall | Precision | Recall | Precision |
| No | 85.7% | 90% | 98.1% | 97.2% | 95.8% | 94.4% | 100% | 100% |
| Yes | **98.1%** | **89.6%** | **99.1%** | **100%** | **95.4%** | **98.8%** | **100%** | **100%** |

Bold values indicate the effect of employing FPN on recall and precision for each class

time can be shorter. Our model was able to inspect more efficiently due to the decrease in the number of parameters and inference time than other models. Tables 3 and 4 present the performance of our model and the baseline models.

### Defect Size Inspection Performance

If our classification model predicted images as defects, the proposed framework obtained a heatmap using the CAM. The heatmap had a value between 0 and 1, and we binarized the images. Using binarized images, we can obtain the lengths of the cracks and EMC defects. Figure 5 shows the performance of defect size inspection. And Table 5 present the error results of the lengths of the cracks and the invasion depth of EMC defects. We used mean absolute percentage error (MAPE) to calculate performance of our model. In line with our purpose of detecting the approximate size of defects, the crack error was 20.92% and the EMC defect was 18.77% in the invasion depth. An error occurred because the thin and vertical crack was detected to have a thick horizontal length due to the CAM.

### Ablation Study

We introduced an FPN to distinguish cracks, EMC defects, normal, and notches of various sizes in the wafer buffer zone. In addition, we used focal loss to learn stable models for class imbalance. To determine the importance of each factor, we compared the differences in the accuracy of the model according to the use of the FPN and focal loss.

#### FPN

We tested the accuracy, parameter number, and inference time of our classification model with and without an FPN as shown in Table 6. We extracted three feature maps from the model and used them in pyramid format to detect defects. The model without an FPN is a modified Xception model that uses middle flow only once. The accuracy performance of our model was 2.6% better than that of the model without FPN. There was no significant difference between the parameter numbers and inference time. In particular, Table 7 shows the increase in recall scores for crack and EMC defect is more remarkable than the increase in accuracy.

#### Focal Loss

We tested whether the focal loss improved the accuracy of our model as shown in Table 8. The accuracy was 0.8% better with focal loss. Note that the defect recall score is significantly improved because the focal loss is strong against the class imbalance.

## Conclusion

In this study, we proposed a wafer buffer zone defect detection framework that can accurately and quickly detect defects and infer their size. In Table 3, it is demonstrated that our model achieves performance prity with EfficientNet; however, it offers advantages in terms of speed. And our model, particularly in terms of the recall score for defects, demenstrates higher performance compared to other models as can be seen in Table 4. We simplified Xception for speed and we improved the performance of our model in problem of detecting various sizes of defects using an FPN without increasing the model capacity for accuracy as presented in Tables 6 and 7. The recall score for defects was increased by applying focal loss to solve the problem of imbalance in actual industrial data in Table 8. In addition, we managed to extract the size of the defect using CAM without using a semantic segmentation or detection model. Although there is a limitation in not being able to extract accurate defect sizes, it has the advantage of being able to infer the approximate size without label because it is difficult to make a ground truth label of data. In conclusion, it is expected that our framework will be able to accurately and quickly inspect the existence of wafer defects inside semiconductor manufacturing equipment and contribute to the improvement of semiconductor manufacturing yield.

**Acknowledgements** This work was partly supported by Korea Institute for Advancement of Technology (KIAT) Grant of MOTIE (P0008697) and the Technology development Program of MSS (S2952758).

**Funding** Open Access funding enabled and organized by Seoul National University.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

## References

1. O.R. Vincent, O. Folorunso et al., A descriptive algorithm for Sobel image edge detection. in Proceedings of Informing Science & IT Education Conference (InSITE), vol. 40, pp. 97–107 (2009)
2. J. Canny, A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–698 (1986)
3. Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, H. Ling, M2Det: a single-shot object detector based on multi-level feature

pyramid network. in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 9259–9266 (2019)

4. B. Singh, M. Najibi, L.S. Davis, SNIPER: efficient multi-scale training. Adv. Neural Inf. Process. Syst. **31**, 9310–9320 (2018)
5. S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4203–4212 (2018)
6. J. Redmon, A. Farhadi, YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
7. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN. in Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
8. T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection. in Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
9. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector. in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Part I 14, pp. 21–37 (Springer, 2016)
10. T. Kong, A. Yao, Y. Chen, F. Sun, HyperNet: towards accurate region proposal generation and joint object detection. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 845–853 (2016)
11. O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation. in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Part III 18, pp. 234–241 (Springer, 2015)
12. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
13. H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation. in Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528 (2015)
14. H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)
15. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, J. Adam, Encoder–decoder with atrous separable convolution for semantic image segmentation. in Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018)
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
17. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
18. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. F. Chollet, Xception: deep learning with depthwise separable convolutions. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
20. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
21. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
22. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
23. E. Real, A. Aggarwal, Y. Huang, Q.V. Le, Regularized evolution for image classifier architecture search. in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4780–4789 (2019)
24. M. Tan, Q. Le, EfficientNet: rethinking model scaling for convolutional neural networks. in International Conference on Machine Learning, pp. 6105–6114 (PMLR, 2019)
25. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2015)
26. G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
27. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)


**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.