AN ANALYSIS OF DEFECT PATTERNS FOR AUTOMATED WAFER DEFECT CLASSIFICATION AND SPATIAL DETECTION

Kaustubh Indane [1, 2, †]

1. Northwestern University

2. [k-indane/wafer-defects]

† Kaustubh.indane@gmail.com

Abstract

Semiconductor manufacturing requires rapid and accurate identification of defect patterns to reduce costs, improve yield, and accelerate decision-making. Traditional manual defect review is labor-intensive and prone to delays. This study proposes a deep learning pipeline to automate wafer defect classification and spatial localization. Wafer-level classification is performed using Random Forest, Convolutional Neural Network, and MobileNetv2, with the custom CNN achieving the highest accuracy (94%) and F1-score (94%). For spatial localization of defect patterns, a YOLOv10s model is developed, achieving strong detection performance (mAP@50 = 0.859). The compound pipeline classifies wafer-level defects and identifies multiple defect patterns, offering a scalable, high-performance solution for real-time defect detection in semiconductor fabrications.

Table of Contents

Introduction and Research Objectives

Semiconductor manufacturing relies on defect and variation reduction to improve performance, quality, and yield of products. However, defect identification and classification are traditionally labor-intensive processes that require significant engineering time for manual review and analysis. This results in increased costs, slower decision-making, and expensive delays in yield improvement. Strong automatic classification shortens time between problem identification and resolution, and can reduce human defect review efforts by 60-80%. This translates to an estimated annual savings between $500K and $2M (Averroes 2024). This study aims to develop a wafer defect classification deep learning model that automatically identifies and categorizes defect patterns from wafer map defectivity data.

Instead of classifying each wafer map as a certain defect pattern as given in the ground truth data, we will also develop a YOLOv10 model for spatial detection of spatially related defect patterns. This allows the model to identify multiple defect patterns on a single wafer with bounding boxes. Spatial identification is crucial for root cause sourcing in semiconductor manufacturing. Not only classifying the pattern of defects per wafer, but also identifying where multiple defect patterns are located, is fundamental for engineers to improve manufacturing processes.

We aim to build an automatic defect classification model by:

1. Extracting density, geometric, radial features from defectivity data
2. Creating Random Forest, Custom CNN, and MobileNet-v2 models for wafer level classification with high accuracy and low computation time

3. Building YOLOv10 model for spatial identification of spatially related defect classes (Scratch, Donut, Center, Edge-Loc, Loc)
4. Exploring performance of model results with speed, generalizability, and metrics, including confusion matrices, accuracy, and F1 score
5. Proposing a compound CNN and YOLOv10s pipeline

Keywords:

- Wafer: Circular silicon substrate used for fabricating chips
- Die: Individual rectangular chips that make up a wafer (1000s per wafer)
- Defect: A die which does not meet quality/customer standards

Literature Review

The automation of defect pattern recognition has been an active area of research for deep learning applications. Several studies have laid the groundwork for both wafer-level classification and spatial detection.

Batool et al. (2021) provide a comprehensive review of deep learning techniques applied to silicon wafer defect recognition. Their study emphasizes the effectiveness of convolutional neural networks in modeling spatial defect patterns, especially over traditional image processing or ML models like SVMs.

One widely adopted CNN model in resource-constrained environments is MobileNetv2, introduced by Sandler et al. (2018). MobileNetv2 is designed around the concepts of depth wise separable convolutions and inverted residual blocks, significantly reducing computational complexity without compromising on accuracy. These properties have made it a preferred model for real-time classification tasks, including those in defect inspection on edge devices.

Shinde et al. (2022) evaluated the performance of YOLOv3 and YOLOv4 architectures for wafer defect localization and classification. Their study demonstrated that these models could achieve over 94% classification accuracy in real-time on a dataset of 19,200 wafer maps.

Further advancements are seen in the work by Li et al. (2025), who developed YOLOSeg, an instance segmentation model based on YOLOv5s. YOLOSeg integrates a UNet-like structure to enhance segmentation capabilities, achieving an average precision of 0.821 and an intersection over union (IoU) of 0.732. This model excels in detecting small particle defects on wafer dies.

Ma et al. (2024) proposed CC-De-YOLO, a multiscale object detection method based on YOLOv7, tailored for wafer surface defect detection. By incorporating coordinate attention mechanisms and optimizing the sampling process, CC-De-YOLO achieved a mean average precision of 91.0% at IoU threshold 0.5.

Finally, Theodosiou et al. (2023) present a broader survey of ML-based methods for wafer defect recognition, categorizing efforts into rule-based, traditional ML, and deep learning frameworks. Their review underscores the trend toward hybrid models and compound pipelines—an approach echoed in this study through the combination of CNN for classification and YOLOv10s for spatial detection.

Data Introduction and Preparation

Column	Description	Type
Wafer_Map	List of lists of defects per row on wafer (2 = yes, 1 = no, 0 = background)	Object
Defect_Class	Wafer level classification of pattern	Object

Table 1. Data Dictionary – Each wafer has one row with Wafer_Map data and Defect_Class classification. Wafer defect data is given as a list of lists.

The dataset is from Multimedia Information Retrieval Lab at National Taiwan University who focus on audio/visual recognition, retrieval, and synthesis. 'MIR-WM811K' dataset is a collection of 30,000+ real-world wafer defect maps from an unspecified manufacturing fabrication in Taiwan. Examples of each defect pattern are given. More in ***Appendix A***:

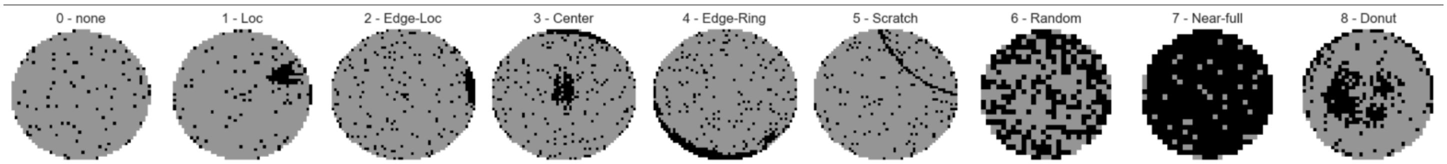


Figure 1. Example wafers sampled from each Defect_Class

All classes will be considered for wafer level classification. Spatially related classes were manually annotated with bounding boxes to establish ground truth. However, due to rectangular bounding boxes, it is difficult to boundary Edge-Ring defects, so they are excluded from spatial analysis. Near-full, Random, and None were also excluded as they have no benefit from spatial analysis. To reduce noise, isolated defects were removed. Wafer maps are in many dimensions, so all were transformed with nearest neighbor interpolation to be 45 x 45, the average wafer map dimension to reduce distortion (Figure 2). Class imbalance was handled by augmentation (with random rotation of 90, 180, and 270 degrees) and sampling:

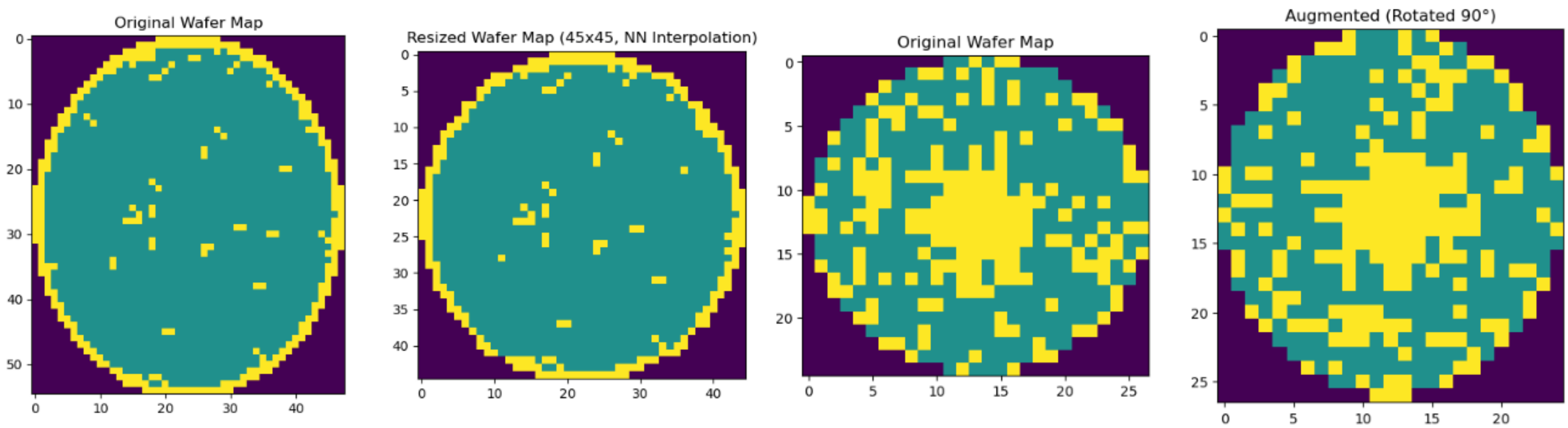


Figure 2. NN Interpolation + Augmentation. Left: Nearest Neighbor interpolation was used to standardize all wafer map dimensions to 45 x 45, producing good results and retaining pattern information. Before and after resizing of an Edge-Ring defect is shown. Right: To improve class imbalance, augmentation of 90, 180, 270 degrees retains pattern distinction while improving generalizability.

For wafer-level classification, a 15% test and 15% validation set were excluded from class balancing. Training distribution during balancing is shown:

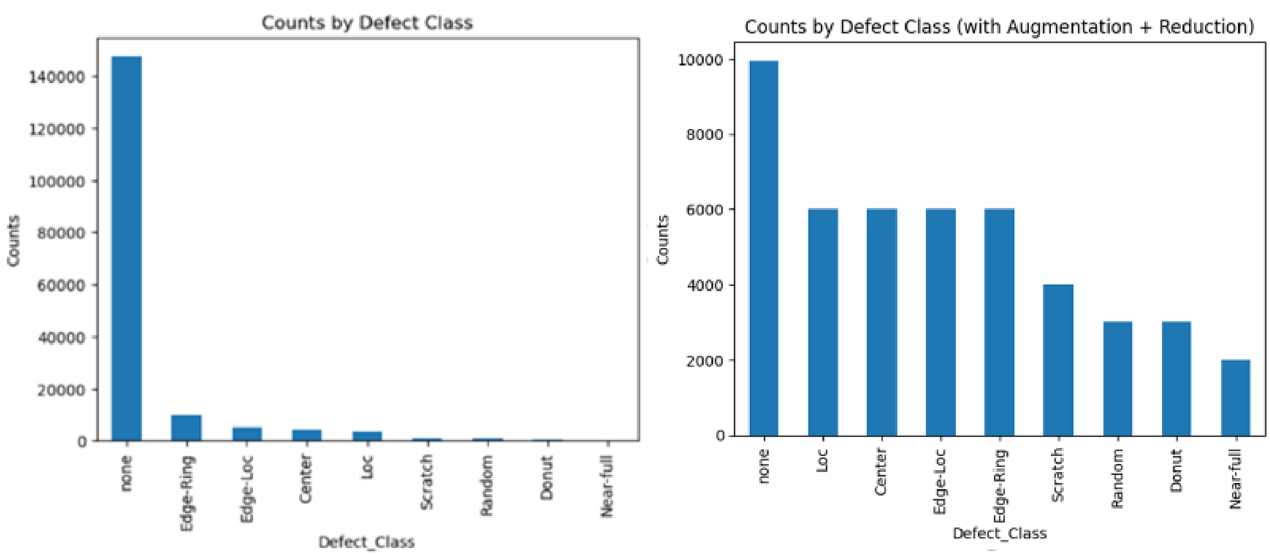


Figure 2. Training data before and after balancing with augmentation and reduction sampling to improve model classification power in underrepresented classes

For spatial classification with YOLOv10, a 15% test and 15% validation set was excluded from balancing.

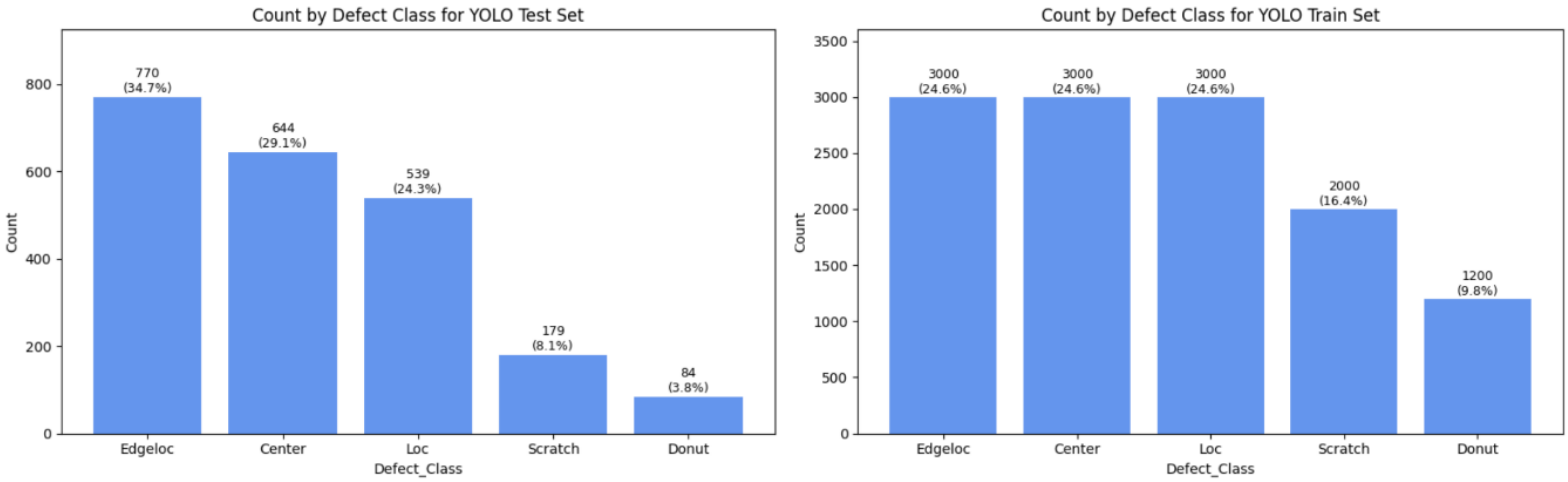


Figure 3. Train and Test data for YOLOv10s. Only classes that benefit from spatial localization will be considered. Left: A 15% test set is used for evaluation, and the validation set has an identical distribution. Right: The training set has 70% of data, which has been augmented and reduced to improve model classification power in underrepresented classes

Feature Engineering

Several defect density features for RF modelling were created. Total defect density count, regional and radial defect density counts based on Figure 5. Boundaries were chosen to best capture defect classifications.

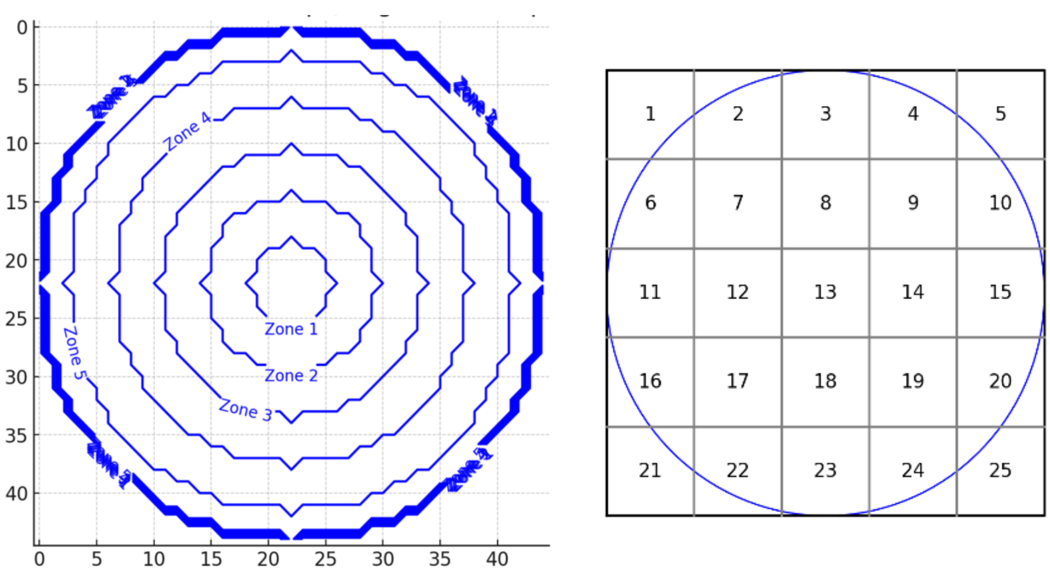


Figure 4. Radial and Regional Feature Boundaries. Left: Radial zoning of defect density features. These are chosen to best capture patterns. For example, Center and Donut can be distinguished with Zone 1/2, Edge-Ring defects with Zone 6. Right: Regional zoning allows localized patterns to be captured in features.

For YOLO data, *LabelImg* in Python was used to annotate bounding boxes to create training data for 14,000 images.

Methods

Wafer-level Classification

Three models are trained on analysis: RandomForest, CNN, and MobileNetv2:

1. **RandomForest:** A strong initial non-parametric model for non-linear decision boundaries with quick computation time. Defects that are in similar locations (Center, Donut, Near-full, Edge-Ring) will be distinguishable. Edge-Loc, Loc, Scratch, will struggle as they depend on spatial relationships. Allows intrepreability of feature importance.
2. **Custom CNN:** Best performance in image classification with strong spatial relationship learning.
3. **MobileNetv2:** A lightweight model optimized for mobile devices, offering strong accuracy and computational efficiency. It is a top model in wafer defect classification, enabling real-time decisions critical for timely resolution (Tsai 2025). This is the industry standard to judge Custom CNN's performance.

Modelling will be done in Tensorflow 2.10 and Keras libraries in Python using GeForce NVIDIA GPU. Evaluation and training is done with class imbalance in mind. Metrics include confusion matrices, accuracy, precision, recall, and F1 score. The aim is powerful classification, generalizability, and speed.

RandomForest

Structure
{'n_estimators': 1000, 'min_samples_split': 2, 'min_samples_leaf': 2, 'max_features': 'log2', 'max_depth': 70}

Table 2. RandomForest Model Hyperparameters - Extensive tuning with GridSearchCV was used to determine best performing model.

Custom CNN

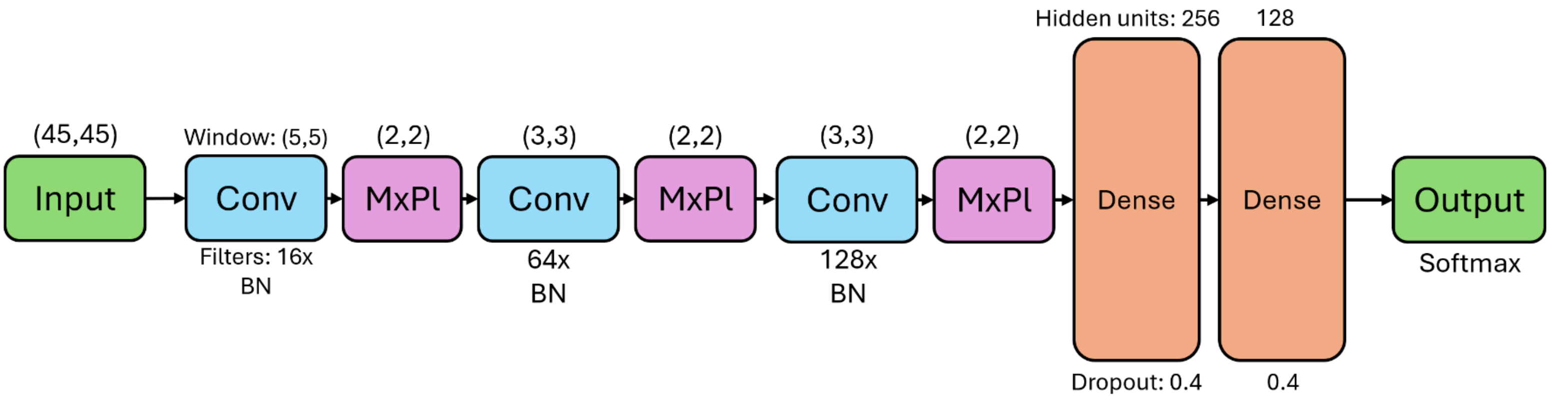


Figure 5. 3 Block CNN Design – The first convolutional layer has 16 filters, enough to capture edges sufficiently. It also uses a kernal size of 5x5 over 3x3 to improve scratch detection with larger windows. Max pooling was used to retain most important feature data in each convolution. Two dense layers provide the output for Softmax classification. Other hyperparameters were tuned for best performance.

MobileNetv2

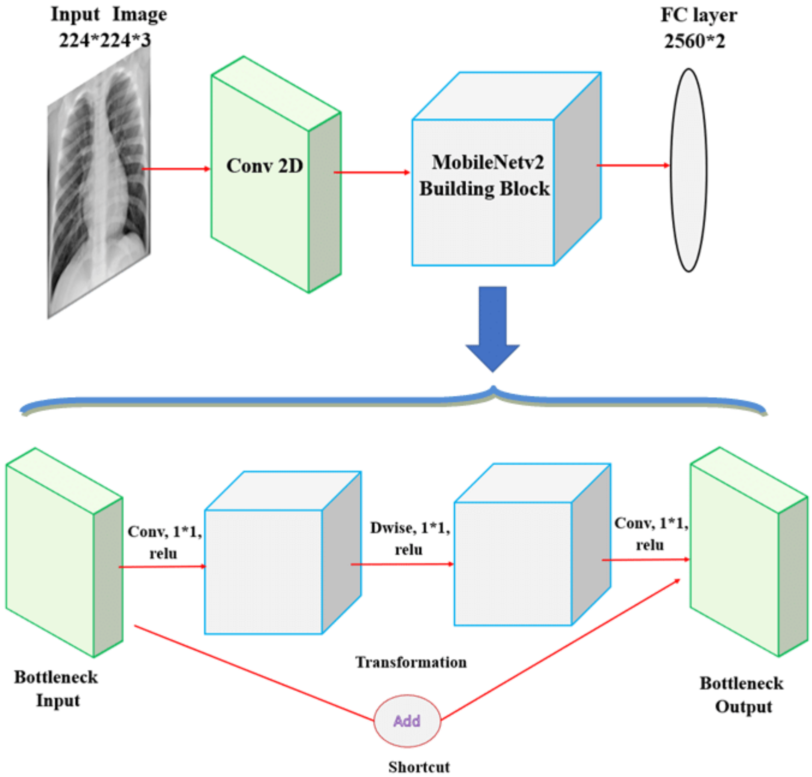


Figure 6. MobileNetv2 Design

Uses 1x1 convolution windows to reduce channel numbers before applying depthwise separable convolutions, a bottleneck structure. This expansion allows the model to capture more complex features and use inverted residual training. We used trainable weights with GlobalAveragePooling into two dense layers, then, a final sigmoid output classification.

Spatial Classification

YOLOv10s (You Only Look Once) combines highspeed inference with spatial understanding by performing classification and localization in a single forward pass. Predictions are made with moving bounding boxes, assigning a probability for each box to provide classifications. This allows multiple defect detection in a single wafer map while also drawing precise bounding boxes around each defect pattern.

Results

Wafer-Level Classification

Model	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-Score	Training Time (s)	Evaluation Time (s)
Random Forest	Test: 80% Training: 86%	87%	80%	83%	7.9	0.6
Custom CNN	Test: 94% Training: 93%	95%	94%	94%	95.0	2.6
MobileNetv2	Test: 88% Training: 92%	94%	88%	90%	407.2	5.9

Table 3. Modeling Results – Wafer level classification results for Random Forest, Custom CNN, and MobileNetv2 are given. Accuracy, along with Precision, Recall, F1-Score, and Evaluation time are key metrics to measure performance with the goal of powerful classification in all classes and real-time capability.

Due to class imbalances, an educated guess of 'none' would obtain 85% accuracy. Therefore, RandomForest is a weak overfit model, although with quickest evaluation time. CNNs are supreme performance in accuracy. The 3-Block CNN outperforms MobileNetv2 for F1 score, accuracy, and speed. MobileNetv2 aims to classify 1,000 classes, intake three channels of data, and take 95x95 input shape. Our adaptation into a smaller scope of 9 classes, with 1 channel of data, and with 45x45 input shape, may have given worse model performance.

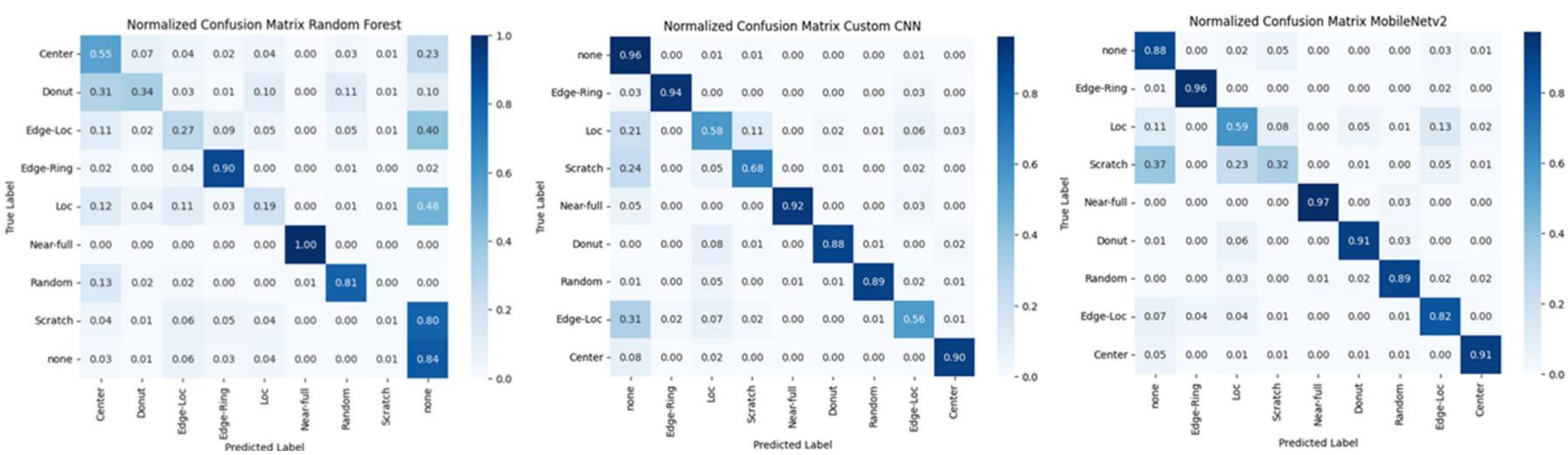


Figure 7. Normalized Confusion Matrices – Normalized accuracy for each model in each class is shown.

Confusion matrices reveal RF suffers identifying spatially related classes, Scratch for example (Figure 8). The custom CNN provides strong performance in every defect class bucket.

MobileNetv2 struggles at identifying scratches over the Custom CNN. This could be explained by the 1x1 convolutional window in MobileNetv2, versus the 5x5 window in the first CNN layer.

In production, the CNN allows for wafer level defect pattern recognition. YOLO would be fed spatially relevant classes from the CNN results (Scratch, Loc, Donut, Center, Edge-Loc). None, Near-full, Random, and Edge-Ring will not be fed, and are the highest accuracy classes of the CNN.

Spatial Classification

Class	N	Precision	Recall	mAP@50	mAP@50-95
All	2216	0.797	0.823	0.859	0.629
Center	644	0.880	0.910	0.948	0.731
Donut	84	0.800	0.976	0.946	0.806
Loc	539	0.787	0.794	0.845	0.604
EdgeLoc	770	0.796	0.723	0.811	0.522
Scratch	179	0.722	0.712	0.745	0.482

Table 4. YOLO Test Metrics by Class – 'N' illustrates wafer level class data to illustrate class imbalances. Precision, Recall, and mAP scores measure model power in each defect class.

YOLOv10s had excellent overall performance. ***Appendix B*** contains annotated result examples. The standard evaluation for object detection is Mean Average Precision (mAP) at an IoU. Intersection over Union is calculated between predicted and truth boxes. mAP@50 over 0.85 illustrates a very strong model. Scratch defects remain difficult to detect due to their thin, elongated shape, which is occasionally distorted by preprocessing interpolation. They also tend to appear in overlapping boxes, a scenario where YOLO typically struggles with accurate detection. It predicts a fixed number of bounding boxes per grid cell and relies on non-maximum suppression, which can suppress true positives with close objects.

Conclusions

This study successfully demonstrates a pipeline for wafer defect classification and spatial localization using deep learning. A custom 3-block CNN achieved the fastest classification performance (94% test accuracy, 94% F1-score). Even so, ~25% of Loc, Scratch, and Edge-Loc defects will be misclassified as ‘none.’ For spatial localization, YOLOv10s achieved excellent performance with an mAP@50 of 0.859 and mAP@50–95 of 0.629, enabling accurate identification of multiple defect types within a single wafer. The proposed compound pipeline of CNN classification, followed by YOLO detection, of spatially related classes only, is a powerful solution for automating defect analysis in semiconductor manufacturing.

Directions for Future Work

To further enhance classification accuracy, robustness, and scalability:

1. Explore better data interpolation methods to maintain variability, especially for Scratches, or adapt model to intake varying input dimensions
2. Incorporate high resolution three channel RGB data to capture richer defect morphologies, a crucial task in production settings. This enables not only classifying defect patterns, but allows morphologies of individual defects.
3. Design a production end-to-end pipeline with real-time access across many production devices

Acknowledgements

The author would like to thank Professor Aliana J. Maren from Northwestern University Master of Data Science Program for her guidance on academic report structure and feedback throughout this study.

Data Availability

The dataset is from Multimedia Information Retrieval Lab at National Taiwan University who focus on audio/visual recognition, retrieval, and synthesis. 'MIR-WM811K' dataset is a collection of 30,000+ real-world wafer defect maps from an unspecified manufacturing fabrication in Taiwan. The data is publicly available on their website:

https://mirlab.org/dataset/public/

Code Availability

The code supporting the findings of this study is available at the following GitHub repository: k-indane/wafer-defects. EDA, feature creations, and modelling code are all documented and shared.

References

Averroes. 2024. "Visual Inspection System Cost Breakdown & Calculating ROI." *Averroes AI Blog*, June 18, 2024. https://averroes.ai/blog/visual-inspection-system-cost-breakdown.

Batool, Uzma, Zool Hilmi Ismail, Mohd Ibrahim Shapiai, Muhammad Tahir, Noor Jannah Zakaria, and Ahmed Elfakharany. 2021. "A Systematic Review of Deep Learning for Silicon Wafer Defect Recognition." *IEEE Access* 9: 114390–114405. https://xplorestaging.ieee.org/ielx7/6287639/9312710/09517097.pdf?arnumber=9517097&utm_source=sciencedirect_contenthosting&getft_integrator=sciencedirect_contenthosting.

Data Science Process Alliance. n.d. “CRISP-DM: A Methodology for Data Mining.” *Data Science Process Alliance*. Accessed March 15, 2025. https://www.datascience-pm.com/crisp-dm-2/.

Gong, J., and L. Chen. n.d. "Wafer Map Failure Pattern Classification Using Deep Learning." *CS230 Deep Learning Projects, Stanford University*. Accessed March 15, 2025. https://cs230.stanford.edu/projects_fall_2019/reports/26259703.pdf.

Li, K. S.-M., Y. Zhou, J. Jiang, W. C. Wu, and S. M. Lin. 2021. "Hidden Wafer Scratch Defects Projection for Diagnosis and Quality Enhancement." *IEEE Transactions on Semiconductor Manufacturing* 34 (1): 9–16. https://ieeexplore.ieee.org/abstract/document/9273073.

Li, Yen-Ting, Yu-Cheng Chan, Chen-Che Huang, Yu-Chang Hsu, and Ssu-Han Chen. "YOLOSeg with Applications to Wafer Die Particle Defect Segmentation." *Scientific Reports* 15, no. 2311 (2025).

Ma, Jianhong, Tao Zhang, Xiaoyan Ma, and Hui Tian. "CC-De-YOLO: A Multiscale Object Detection Method for Wafer Surface Defect." *Frontiers of Computer Science* 18, no. 2 (2024): 261–285.

MIR Lab. n.d. "Public Data Set." *MIR Lab*. Accessed March 15, 2025. http://mirlab.org/dataSet/public/.

Patel, Ashish. 2019. "WM-811K Wafermap." *Kaggle*. Accessed March 15, 2025. https://www.kaggle.com/code/ashishpatel26/wm-811k-wafermap.

Rockatoo. 2020. "WM-811K Wafer Defect Classification Using CNN-WDI." *Kaggle*. Accessed March 15, 2025. https://www.kaggle.com/code/rockatoo/wm-811k-wafer-defect-classification-using-cnn-wdi.

Shinde, Prashant P., Priyadarshini P. Pai, and Shashishekhar P. Adiga. "Wafer Defect Localization and Classification Using Deep Learning Techniques." *IEEE Access* 10 (2022): 39969–39978.

Saqlain, M., B. Jargalsaikhan, and J. Y. Lee. 2019. "A Voting Ensemble Classifier for Wafer Map Defect Patterns Identification in Semiconductor Manufacturing." *IEEE*

Transactions on Semiconductor Manufacturing 32 (2): 171–182. https://ieeexplore.ieee.org/document/8664594.

Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474.

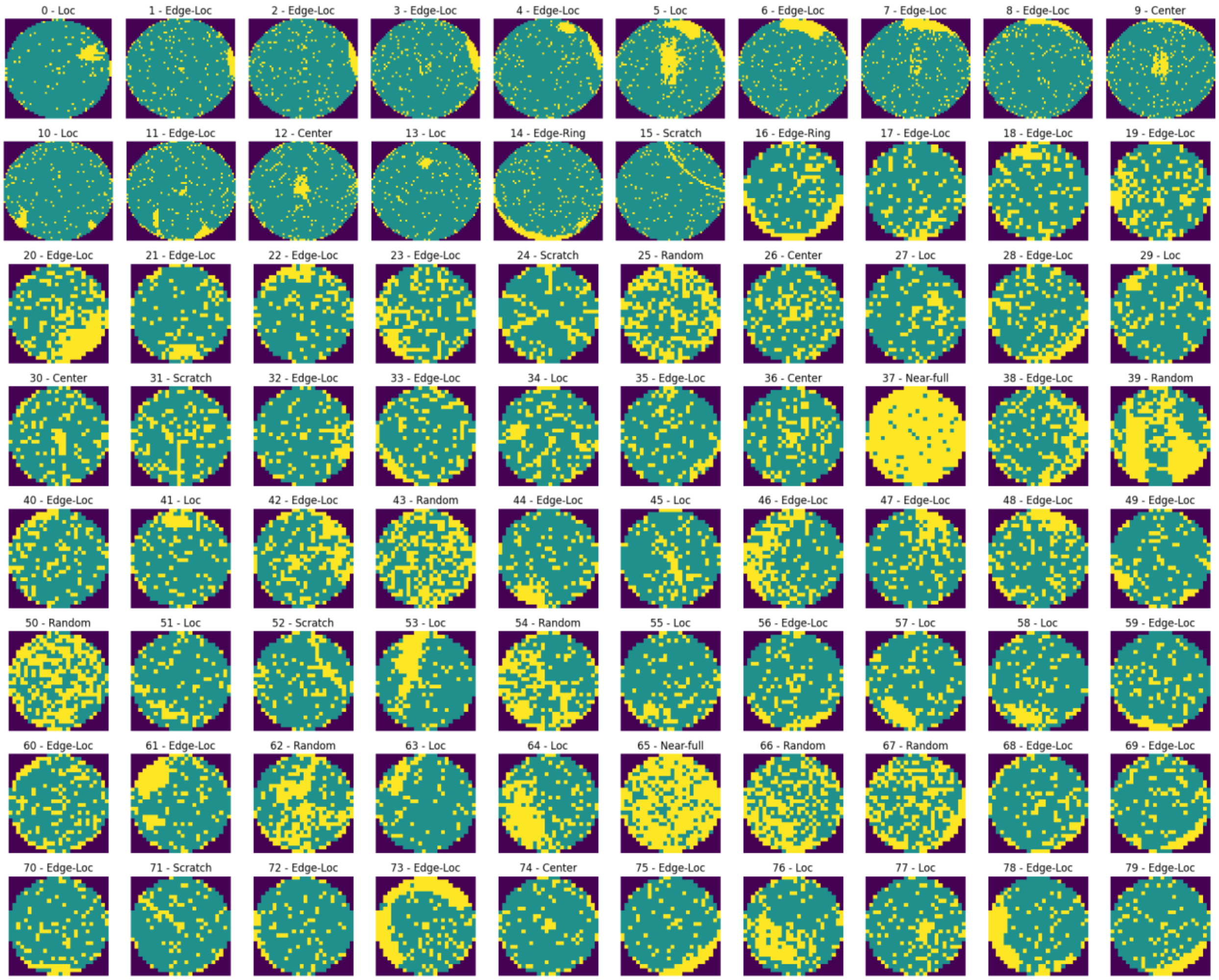
Theodosiou, T., A. Rapti, K. Papageorgiou, T. Tziolas, E. Papageorgiou, N. Dimitriou, G. Margetis, and D. Tzovaras. 2023. "A Review Study on ML-Based Methods for Defect-Pattern Recognition in Wafer Maps." *Procedia Computer Science* 217: 570–583. https://doi.org/10.1016/j.procs.2022.12.253.

Tsai, T. H., and C. Y. Wang. 2025. "Wafer Map Defect Classification Using Deep Learning Framework with Data Augmentation on Imbalance Datasets." *Journal of Image and Video Processing* 2025 (6). https://doi.org/10.1186/s13640-025-00666-3.

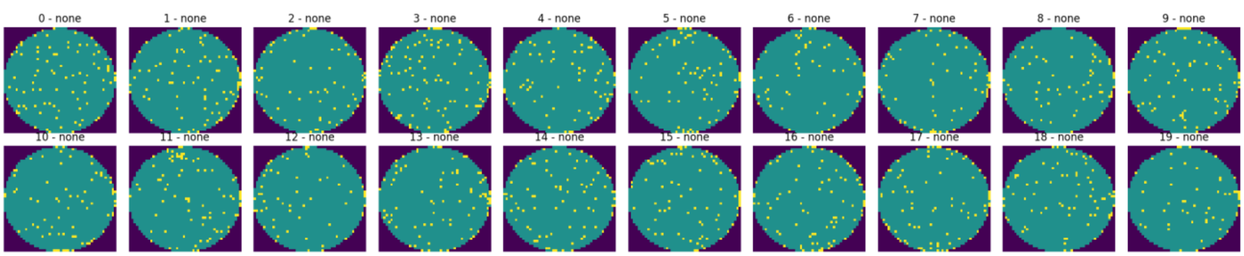
Wu, Meng-Sung. 2015. "Modeling Query-Document Dependencies with Topic Language Models for Information Retrieval." *Information Sciences* 312: 1–12. https://www.sciencedirect.com/science/article/abs/pii/S0020025515002212.

Github Copilot for code completions

Appendix A



0 - Loc
1 - Edge-Loc
2 - Edge-Loc
3 - Edge-Loc
4 - Edge-Loc
5 - Loc
6 - Edge-Loc
7 - Edge-Loc
8 - Edge-Loc
9 - Center
10 - Loc
11 - Edge-Loc
12 - Center
13 - Loc
14 - Edge-Ring
15 - Scratch
16 - Edge-Ring
17 - Edge-Loc
18 - Edge-Loc
19 - Edge-Loc
20 - Edge-Loc
21 - Edge-Loc
22 - Edge-Loc
23 - Edge-Loc
24 - Scratch
25 - Random
26 - Center
27 - Loc
28 - Edge-Loc
29 - Loc
30 - Center
31 - Scratch
32 - Edge-Loc
33 - Edge-Loc
34 - Loc
35 - Edge-Loc
36 - Center
37 - Near-full
38 - Edge-Loc
39 - Random
40 - Edge-Loc
41 - Loc
42 - Edge-Loc
43 - Random
44 - Edge-Loc
45 - Loc
46 - Edge-Loc
47 - Edge-Loc
48 - Edge-Loc
49 - Edge-Loc
50 - Random
51 - Loc
52 - Scratch
53 - Loc
54 - Random
55 - Loc
56 - Edge-Loc
57 - Loc
58 - Loc
59 - Edge-Loc
60 - Edge-Loc
61 - Edge-Loc
62 - Random
63 - Loc
64 - Loc
65 - Near-full
66 - Random
67 - Random
68 - Edge-Loc
69 - Edge-Loc
70 - Edge-Loc
71 - Scratch
72 - Edge-Loc
73 - Edge-Loc
74 - Center
75 - Edge-Loc
76 - Loc
77 - Loc
78 - Edge-Loc
79 - Edge-Loc



0 - none
1 - none
2 - none
3 - none
4 - none
5 - none
6 - none
7 - none
8 - none
9 - none
10 - none
11 - none
12 - none
13 - none
14 - none
15 - none
16 - none
17 - none
18 - none
19 - none

Appendix B