GenAI or not GenAI ?
Comparing AI methodologies to solve Defect Wafer Map Classification Problem

Carla Battiato[1,*,†], Francesco Lanza[1,†], Filippo L.M. Milotta[1,†], Alessandro Orofino[1,†] and Rosetta Rizzo[1,†]

[1]*STMicroelectronics, Stradale Primosole, 50, Catania, Sicily, Italy*

Abstract

The progress and advances in the field of Artificial Intelligence have significantly contributed to solving complex challenges. The AI community is currently focusing on the adoption of generative models to accomplish tasks such as text and code generation, as well as developing chatbots and virtual assistants for customer service. In this paper, we aim to determine whether a generative approach can effectively serve its purpose in the semiconductor manufacturing domain, specifically in the classification of defect wafer maps. Recognizing the correct signature during this step allows engineers to react promptly and take appropriate countermeasures to prevent the diffusion of defects in subsequent manufacturing steps, thereby increasing the yield and quality of the final product. To this end, we compared classical ML techniques, DL approaches, and VLM applications to classify defects in wafer maps. The results are discussed, and relevant highlights are presented to identify use cases where a generative approach can drive the digital transformation of the manufacturing process. Finally, conclusions are drawn, and potential future scenarios are outlined.

Keywords

Wafer Map Defect Classification, Visual Language Model, Machine Learning, Deep Learning, Generative AI, Defect Recognition

1. Introduction

The manufacture of silicon chips is the cornerstone of this century's digital transformation, enabling new frontiers in fields such as *automotive*, *robotics* and *artificial intelligence*. In this sector, it has become essential to maximize productivity to satisfy growing market demand and meet specific quality standards for the devices produced. The intricacy of semiconductor manufacturing is thoroughly described in the second chapter of this book [1] for modeling and analysis of semiconductor wafer fabrication facilities. *Defectivity* is identified as one of the core processes in chips production. It involves a series of complex steps for identifying any significant physical defect that can impact the yield of a silicon *Wafer*, and consequentially of the Wafer batch, technically defined as *Lot*. Defectivity control originated as a process in which the main actors were the inspection equipment and the operator, known as the defectivity engineer. During the manufacturing flow traversed by the lot, the production line includes several defectivity inspection steps where all the wafers within a lot, are scanned to verify if the previous equipment and processes introduced flaws critical for any die on the wafer. Defectivity engineers are responsible for analyzing the inspection results. Until a few years ago, their job involved manually reviewing each inspected wafer on a digitally plotted map, classifying the most significant defects found on the wafer map, and identifying any specific patterns.

To elaborate further, a wafer could be classified as *Random* if defects are randomly distributed across its surface, reflecting the state of a regular production chain where, for example, dust can be the cause

23rd International Conference of the Italian Association for Artificial Intelligence, November 25–28, 2024, Bolzano, Italy

[*] Corresponding author.

[†] These authors contributed equally.

carla.battiato@st.com (C. Battiato); francesco.lanza@st.com (F. Lanza); filippo.milotta@st.com (F. L.M. Milotta); alessandro.orofino@st.com (A. Orofino); rosetta.rizzo@st.com (R. Rizzo)

0009-0000-9491-1814 (C. Battiato); 0000-0003-4382-6366 (F. Lanza); 0000-0002-9459-9530 (F. L.M. Milotta); 0000-0002-5781-000X (R. Rizzo)

of sparse defects. As another example, during polishing or photolithography phases the wafer can be affected by the concentration of defects in a particular region of the surface, in addition to the noise generated by random defects. When the resulting pattern can be assigned to a specific signature class (properly performing *Wafer Map Classification*), engineers can take immediate actions, such as holding the lot, stopping a machine, and generally reducing production waste as much as possible. Manually performing Wafer Map classification is expensive in terms of both time and resources, and it is highly dependent on the human experience. These are the reasons that supported the introduction in the industrial ecosystem of the *Automatic Wafer Classification* (AWC) process, where machines can substitute, or at least reduce, the human intervention.

To enhance AWC, several image processing techniques, machine learning or deep learning algorithms have been analyzed and then applied in the industrial field [2, 3]. Indeed, Batool et al. [4] conducted a systematic literature review collecting approaches and methods applied to classify and recognize defect patterns, including standard machine learning approaches [5, 6], autoencoders (AE) [7, 8], and generative methods such as the Generative Adversarial Networks (GAN) [9, 10]. While these methods are now widely used in the semiconductor industries to make defectivity recognition, research-and-development to enhance the process remains an open-ended challenge. On the grounds, and following the wave of the Generative AI era, the aim of this paper is to show and determine whether a generative approach based on the adoption of large language models (LLMs), or as in this case, visual language models (VLMs), can make a significant difference in the semiconductor scenario. To avoid misleading contexts where a generative approach is proposed as a *panacea* for all needs (one-size-fits-all problem), it is important to identify *when* and *where* it should be used. The advantages of generative models based on LLMs or VLMs are evident to all. However, it is crucial to understand the appropriate contexts and situations for their recommended usage. In this work, given a specific context, we show how machine learning approaches, deep learning methods, and visual language models strive. To make a comparative study, we selected as VLM the open model named PaliGemma [11], that is based on the models of the Gemma family (outstanding models for a large variety of open-world tasks) [12], and is designed following the PaLI architecture (state-of-the-art architecture for challenges like Visual Question Answering (VQA) [13], Language Understading, and Image Captioning) [14]. It is important to notice that the purpose of this paper is not to identify the top-performing visual language model compared to other VLMs or LLMs. Rather, the goal is to compare artificial intelligence techniques using state-of-the-art models to demonstrate which strategy is most effective depending on the given manufacturing context and business requirements.

In summary, this paper investigates the following research question: "Generative AI is a hot topic, but what is the actual potential of Visual Language Models when applied to the Defect Wafer Map Classification task?".

The remainder of the paper is structured as follows: the description of the leveraged datasets, together with the definition of our proposed methods for Defect Wafer Map Classification, are reported in Section 2. Outcomes of the experimental phase is reported in Section 3. Finally, in Section 4, we draw our conclusions and we suggest some potential future improvements.

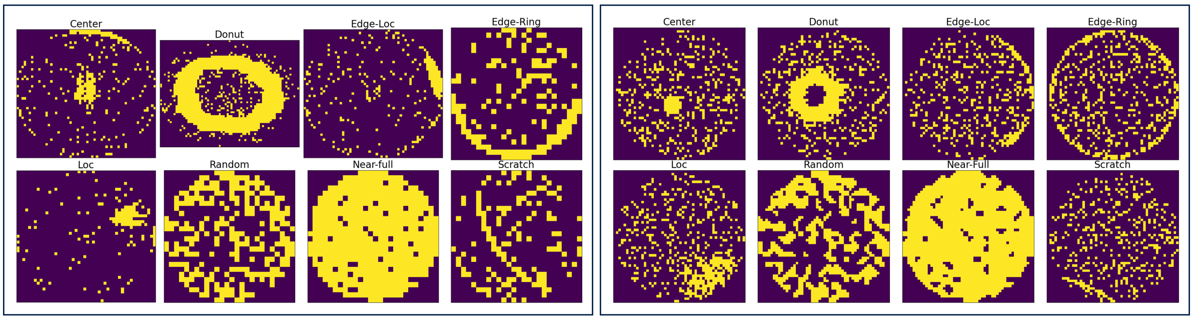
2. Materials and Methods

2.1. Datasets Overview

In this work, we used two different datasets concerning semiconductors industry. Both are publicly available: WM811K [15, 16] and WMPR [17]. Datasets description is reported below.

2.1.1. Dataset 1: WM811K

WM811K is one of the largest known wafer map dataset available to the public [15]. It is used primarily for research in the field of semiconductor manufacturing, for enabling the exploration of machine learning methods for defect pattern recognition and classification. As its name is suggesting, it contains



(a) WM811K

(b) WMPR

Figure 1: Typical wafer maps for any class in datasets (a) WM811K, and (b) WMPR.

811,457 wafer maps, defined as a 2D representation of a semiconductor wafer. In other words, wafer maps are represented as a single channel image with 3 possible values: 0 is the background (area external to the wafer), 1 is identifying good dies on the wafer, and 2 the defective ones, instead. WM811K provides high variability in wafer maps size, as there are 632 different sizes ranging from 6×21 to 300×202 pixels.

WM811K is also including labels for 8 defect types. The list of the defects, together with their count percentage, is the following: Edge-Ring (37.9%), Edge-Loc (20.3%), Center (16.8%), Loc (14.1%), Scratch (4.7%), Random (3.4%), Donut (2.1%), Near-full (0.6%). As can be seen, distribution of the defect classes is skewed and high unbalanced. However, even if the dataset counts 811,457 wafer maps, only the 3% of them is actually labeled with one of the 8 before mentioned classes. In our benchmark, for validation purpose, we will leverage this reduced labeled portion of WM811K, that is counting 25,519 labeled wafer maps. An example of typical wafer maps for this dataset is shown in Figure 1(a).

2.1.2. Dataset 2: WMPR

WMPR is another wafer map dataset available to the public [17], similar to WM811K. Its name stands for *Wafer Map for Pattern Recognition*[1]. As in WM811K, wafer maps are represented as a single channel image with 3 possible values: 0 is the background, 1 is identifying good dies on the wafer, and 2 the defective ones, instead. Wafer maps size is the same for all the images in the dataset, set to 51×51 pixels.

Originally, WMPR was tought for addressing Mixed-Type Defect Patterns in wafer maps. Indeed, authors included up to 29 classes with multiple defects. However, for the scope of our work, and for enabling a coherent comparison with WM811K, we focused only of the single-type defect classes, which are the same defined previously in WM811K. The list of the defects, together with their count percentage, is the following: Edge-Ring (14.3%), Edge-Loc (14.3%), Center (14.3%), Loc (14.3%), Scratch (14.3%), Donut (14.3%), Random (12.3%), Near-full (1.9%). Distribution of the defect classes is more balanced than WM811K, but there are two classes (Random and Near-full) with lesser samples than the other ones. The dataset counts 7,015 wafer maps with a single defect. An example of typical wafer maps for this dataset is shown in Figure 1(b).

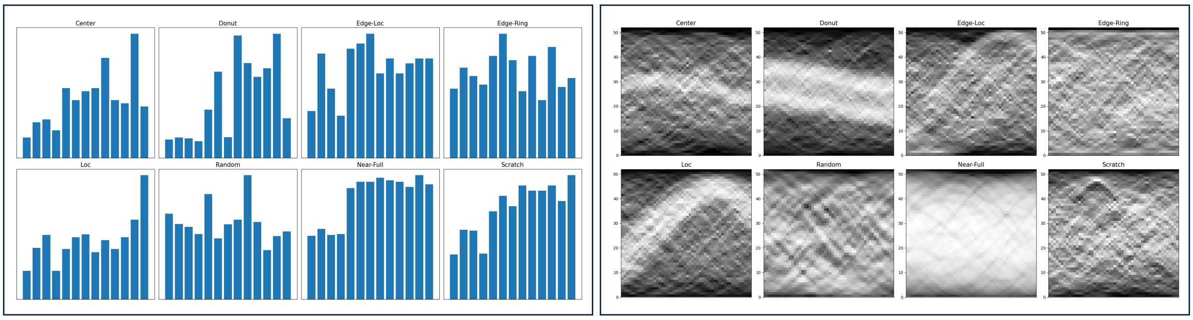
2.2. Methods Overview

In this work, we compared several classification approaches with the main purpose of assessing the potential impact of GenAI on the Defect Wafer Map Classification task. We designed a benchmark for this task, where we compared methods based on: pure Machine Learning (**ML**), Deep Learning (**DL**), and GenAI-Visual Language (**VLM**) models. The comprehensive list of all the benchmarked models is reported in Table 1.

[1] Note that Wang et al. [17] did not specifically named the dataset as WMPR, but for simplicity we just called it in this way.

Table 1
Benchmarked models for Defect Wafer Map Classification.

ML *Machine Learning*	**DL** *Deep Learning*	**VLM** *GenAI – Visual Language Models*
XGBoost [18]		
LightGBM [19]	SimpleNet [20]	PaliGemma-3b-pt-224 [11]
Logistic Regression	VGG16 [21]	PaliGemma-3b-pt-448 [11]
Random Forest	ResNet [22]	PaliGemma-3b-pt-896 [11]
Decision Tree		



(a) Region Density

(b) Radon

Figure 2: Examples of features based on (a) Region Density and (b) Radon, in WMPR.

2.3. Data Preprocessing

Accordingly to the different classification methods described in Section 2.2, we defined proper preprocessing steps, as described below.

2.3.1. ML Preprocessing

For Machine Learning based classification, we followed the features extraction preprocessing initially thought for WM811K [15, 16]. Specifically, it consists of three main feature categories:

- 13 Region Density Based Features, defined dividing the wafer map into 13 not-overlapped regions and then computing fail density of each region.
- 40 Radon Based Features, defined generating a 2D representation of the wafer map applying the Radon Transform, and computing the so called sinogram. For handling possible different wafer map size, Radon Transform values are further processed through a cubic interpolation to obtain fixed dimension feature values for row mean and row standard deviation from sinograms. For both row mean and row standard deviation, the dimension of this resampling is fixed to 20 (giving a total of 40 Radon Based Features);
- 6 Geometry Based Features, defined identifying the *most salient area* (i.e., the area in the wafer map with the highest amount of adjacent defects), and then computing geometrical and statistical features like region area, perimeter, length of major axes, length of minor axes, solidity and eccentricity.

Thus, each wafer map will be processed extracting a total of 59 visual features, that can be used for training ML based classification models. Examples of these preprocessed features for WM811K can be retrieved directly from the dataset references [15, 16], while, for WMPR wafer maps previously shown in Figure 1(b), some examples are given in Figure 2.

2.3.2. DL Preprocessing

For Deep Learning based classification, in order to make homogeneous wafer maps in both WM811K and WMPR, we applied a rescaling of all the wafer maps to the fixed size of 448×448 pixels (interpolating with *Nearest* modality, for avoiding to introduce non-binary values in the images). We choose this resolution as it is higher than any image in both datasets. We removed from the datasets the concept of background level (i.e., pixel values equal to 0), remapping good dies from 1 to 0, and defective dies from 2 to 1. Through this preprocessing for Deep Learning based methods, we ensured that same resolution is adopted for both the benchmarked datasets, and that input layers of all the trained Convolutional Neural Networks could be configured similarly in terms of height, width and number of channels for input images.

2.3.3. VLM Preprocessing

For VLM based classification, we leveraged three pretrained (*pt*) versions of PaliGemma-3b, namely: *pt-224*, *pt-448* and *pt-896*, where the version number indicates which was the image size used for pretraining a specific model [11]. Thus, we prepared rescaled wafer maps of WM811K and WMPR to 224×224, 448×448, and 896×896 pixels, for each PaliGemma model, respectively. As done in Section 2.3.2, we interpolated with *Nearest* modality. Moreover, since PaliGemma is leveraging a dynamic range within $[-1{,}1]$, we remapped binary values of our wafer maps accordingly.

Finally, we preprocessed each dataset for having the expected structure for fine-tuning PaliGemma: *[text, image, suffix]*. In our work, *text* is always equal to "*Which defect do you see in this wafermap?*", and it represents the textual query made by the user when prompting the VLM for the task of VQA [13]. For what is concerning the other two fields, *image* is the binary representation of the rescaled wafer map, and *suffix* is the label defining the defect class of the wafer map.

3. Experimental Results

Our experiments were executed on a Standard_NC24ads_A100_v4 Azure virtual machine, equipped with 24 CPU cores, and one A100 80GB PCIe GPU card. Experiment Settings and Experiment Outcomes are reported below.

3.1. Experiment Settings

In our experiments, we splitted our datasets with a ratio of 60/20/20 percentage for Training, Validation and Test Sets, respectively. Then, a K-Folding Cross Validation was applied with K=4. Any split and fold are ensured to be stratified, in order to have always the same proportion of defect classes. For what is concerning the benchmarking of ML-based models, we leveraged the AutoML feature available on Databricks, a cloud data analysis platform[2]. A Grid-Search for Fine Tuning the Hyper-parameters was conducted for all the methods. We set meaningful ranges for DL and VLM methods, while for ML methods we leveraged automatic Grid-Search through AutoML. Finally, PaliGemma models were fine-tuned with Parameter Efficient Fine Tuning (PEFT) and Quantized Low Rank Adapters (QLoRA), configured with *quantization* at 4 bits and *rank* set to 8 [23].

3.2. Experiment Outcomes and Discussion

Outcomes for Defect Classification Benchmark are shown in Table 2. The performances of the trained models are compared accordingly to the F1-Score computed over the Test Set (that is always the same for each fold). Since we applied K-Fold Cross Validation, we reported Mean ± Standard Deviation of F1-Score over the K defined folds, together with the best (the maximum) F1-Score. In Table 2, Training Time and the most important Hyper-parameters for the best models are also shown.

[2] https://docs.databricks.com/en/machine-learning/automl/index.html (Last Visited on September 2024)

Table 2

Experimental Results for Defect Classification Benchmark. Best performances for Model Type is highlighted in bold, while overall best performance per dataset is also underlined. GPU was leveraged for VLM and DL models, while ML were executed on CPU. Training Time is reported accordingly to GPU or CPU usage.

			F1 Score (%)		Train	
Dataset	**Type**	**Model**	**Mean**	**Best**	**Time**	**Hyper Params for Best Model**
WM811K	**VLM**	**PaliGemma-3b-pt-224**	$94.21(\pm 0.19)$	**94.42**	$\sim 25h$	**LR:** 1e-4, **Epochs:** 15 **Batch:** 1
		PaliGemma-3b-pt-448	$94.15(\pm 0.20)$	94.34	$\sim 35h$	**LR:** 1e-4, **Epochs:** 15, **Batch:** 1
		PaliGemma-3b-pt-896	$94.17(\pm 0.22)$	94.36	$\sim 120h$	**LR:** 1e-4, **Epochs:** 15, **Batch:** 1
	DL	SimpleNet	$77.48(\pm 0.75)$	78.44	$\sim 5m$	**LR:** 1e-5, **Epochs:** 50, **Batch:** 10
		VGG16	$91.16(\pm 0.50)$	**91.85**	$\sim 41m$	**LR:** 1e-5, **Epochs:** 20, **Batch:** 10
		ResNet	$90.75(\pm 0.52)$	91.53	$\sim 22m$	**LR:** 1e-4, **Epochs:** 40, **Batch:** 10
	ML	**XGBoost**	$88.52(\pm 4.38)$	**91.91**	$\sim 13m$	**LR:** 5e-1, **N_Trees:** 497, **MaxDepth:** 10
		LightGBM	$75.34(\pm 20.23)$	91.36	$\sim 20m$	**LR:** 1e-1, **N_Leaves:** 113, **N_Trees:** 23
		Logistic Regression	$87.08(\pm 2.31)$	88.55	$\sim 1m$	**C:** 71, **Mode:** multinomial
		Random Forest	$45.99(\pm 15.95)$	71.24	$\sim 4m$	**N_Trees:** 426, **MaxDepth:** 6, **MinSplit:** 8e-2
		Decision Tree	$60.73(\pm 5.25)$	68.93	$\sim 2m$	**MaxDepth:** 5, **MinSplit:** 1e-1, **MinLeaf:** 5e-2
WMPR	**VLM**	PaliGemma-3b-pt-224	$98.96(\pm 0.24)$	99.24	$\sim 7h$	**LR:** 1e-4, **Epochs:** 15, **Batch:** 1
		PaliGemma-3b-pt-448	$98.91(\pm 0.19)$	99.07	$\sim 10h$	**LR:** 1e-4, **Epochs:** 15, **Batch:** 1
		PaliGemma-3b-pt-896	$99.12(\pm 0.26)$	**99.46**	$\sim 46h$	**LR:** 1e-4, **Epochs:** 15, **Batch:** 1
	DL	SimpleNet	$81.66(\pm 2.66)$	85.57	$\sim 1m$	**LR:** 1e-4, **Epochs:** 5, **Batch:** 10
		VGG16	$98.33(\pm 0.55)$	98.96	$\sim 21m$	**LR:** 1e-5, **Epochs:** 10, **Batch:** 10
		ResNet	$98.62(\pm 0.42)$	**99.05**	$\sim 17m$	**LR:** 1e-4, **Epochs:** 20, **Batch:** 10
	ML	**XGBoost**	$98.35(\pm 1.21)$	**99.28**	$\sim 4m$	**LR:** 2e-1, **N_Trees:** 463, **MaxDepth:** 2
		LightGBM	$84.82(\pm 25.58)$	99.00	$\sim 6m$	**LR:** 1e-1, **N_Leaves:** 644, **N_Trees:** 643
		Logistic Regression	$94.52(\pm 15.66)$	98.58	$\sim 2m$	**C:** 5e-2, **Mode:** multinomial
		Random Forest	$44.33(\pm 28.22)$	94.73	$\sim 2m$	**N_Trees:** 272, **MaxDepth:** 4, **MinSplit:** 4e-2
		Decision Tree	$55.38(\pm 25.44)$	85.16	$\sim 2m$	**MaxDepth:** 5, **MinSplit:** 7e-2, **MinLeaf:** 6e-2

Starting our discussion from outcomes on WM811K, we can see that the best proposed performing models per model type (VLM, DL and ML) are PaliGemma-3b-pt-224, VGG16 and XGBoost, respectively. They are tight each other in terms of performances, reaching F1-Score values higher than 91% in the best k fold, improving from 81% of Wu et al. [15]. In particular, PaliGemma models reach the highest F1-Scores (94.42% in the best case). The three available resolution settings for PaliGemma have low impact on final performances, but heavy impact in terms of Training Time, particularly if compared with the other methods: fine-tuning a VLM on GPU requires hours, while training a Convolutional Neural Network (still on GPU) or even a ML model from scratch (in CPU) with WM811K requires just a bunch of minutes (and sometimes seconds!). Indeed, this is a key-finding: PaliGemma can be really powerful if fine-tuned, and looks capable of addressing the Defect Wafer Map Classification task better than DL and ML models. Finally, among the Cross-Validation performances, VLM and DL models are more stable around the best performances (that is, low F1-Score Standard Deviation values), while ML models tend to higher variability.

When assessing how well PaliGemma performed on WMPR, we can draw similar conclusions. Considering that wafer maps in WMPR always have a native size of 51×51 pixels, the three available resolution settings for PaliGemma were already expected to have low impact on final performances. Indeed, they are similar each others. Similarly to WM811K, PaliGemma performed slightly better than the best DL and ML models. Again, there is a considerable overhead in terms of required Training Time for fine-tuning PaliGemma. Based on the specific manufacturing business need, the trade-off between required Training Time and performances may justify costs related to higher computational time and resources necessary for enabling VLM fine-tuning.

The drill-down with respect to each defect class in both WM811K and WMPR is shown in Figure 3, accordingly to the best models discussed before and reported in Table 2. Generally, the most complex class to be recognized is *Near-full*, often confused with *Random*. For WM811K, *Donut* and *Scratch* are also challenging classes, while for WMPR, *Edge-Ring* and *Edge-Loc* are often confused between each other (indeed they look similar, as visible in Figure 1(b)).

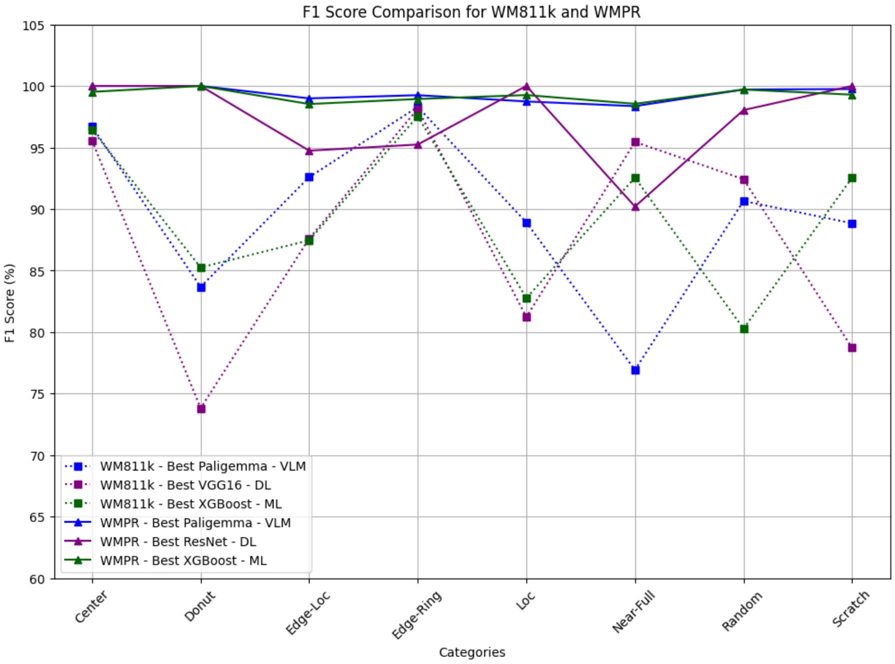


Figure 3: F1-Score for each defect class in WM811K and WMPR, accordingly to the best identified model for VLM, DL and ML Models (refer to Table 2).

4. Conclusions

Defect identification is one of the core processes in semiconductor manufacturing. It involves a series of complex steps to identify every significant physical defect that can affect the yield of a silicon wafer. Manual inspection on the wafer map is expensive in terms of both time and resources, and is highly dependent on human experience. These are the reasons behind the introduction of Automatic Wafer Classification (AWC) into the industrial ecosystem, where machines can replace or at least reduce human intervention.

Given the context, in this work we made a comparative study among machine learning approaches (ML), deep learning methods (DL), and visual language models (VLMs). The main purpose of this paper was not to identify the top-performing VLM model compared to other VLMs. Rather, the goal was to compare artificial intelligence techniques using state-of-the-art models to demonstrate which strategy is most effective depending on the given manufacturing context and business requirements. We benchmarked several models on two publicly available datasets: WM811K [15] and WMPR [17]. For both datasets, VLM, ML and DL models got similar performances. In particular, the selected VLM, PaliGemma, demonstrated to be really powerful if fine-tuned, and capable of addressing the Defect Wafer Map Classification task better than DL and ML models ($+2.5\%$ and $+0.2\%$ F1-Scores for WM811K and WMPR, respectively). However, fine-tuning PaliGemma on GPU required hours, a Training Time significantly higher if compared to DL and ML, that instead requires minutes or even few seconds. Driven by the manufacturing business, balancing between the required training time and performances may justify the costs associated with the increased computational time and resources needed for enabling VLM fine-tuning and reaching a slightly higher F1-Score. In some industrial scenarios, even a $+0.1\%$ of improvement can be crucial.

As next steps, we are planning to extend the presented benchmark adding other VLMs, like Florence-2 [24]. We also planned to address the open-set issue in defect classification. Indeed, it cannot be ensured that all the classes identified in the dataset used for training the classification models will remain the same. The root causes that brought to a defect can be many (i.e., mechanical issues, human factor, chemical or electrical reason, and even catastrophic events). Thus, new defects may appear over time when performing inference on the deployed classification models. We will investigate how to endows our AI system to be able to perform continuous model retraining when an unknown defect is discovered to promptly react if the unknown model is impacting the quality and the yield.

Acknowledgments

We would like to express our gratitude to *Salvo Ciccia*, *Mario Marroccia*, *Giuseppe Ursino*, and *Hugues Duverneuil*, for their support and guidance throughout the preparation of this paper. Their suggestions, review and corrections have significantly improved the quality of this research work.

References

[1] L. Mönch, J. W. Fowler, S. J. Mason, Production planning and control for semiconductor wafer fabrication facilities: modeling, analysis, and systems, volume 52, Springer Science & Business Media, 2012.

[2] R. di Bella, D. Carrera, B. Rossi, P. Fragneto, G. Boracchi, Wafer defect map classification using sparse convolutional networks, in: Image Analysis and Processing–ICIAP 2019: 20th International Conference, Trento, Italy, September 9–13, 2019, Proceedings, Part II 20, Springer, 2019, pp. 125–136.

[3] L. C. Viagrande, F. L. Milotta, P. Giuffrè, G. Bruno, D. Vinciguerra, G. Gallo, Semisupervised classification of anomalies signatures in electrical wafer sorting (ews) maps., in: VISIGRAPP (5: VISAPP), 2020, pp. 278–285.

[4] U. Batool, M. I. Shapiai, M. Tahir, Z. H. Ismail, N. J. Zakaria, A. Elfakharany, A systematic review of deep learning for silicon wafer defect recognition, IEEE Access 9 (2021) 116572–116593.

[5] J. Lin, J. Kung, P. Cheng, A. Hwu, C. Wang, Y. Hsu, Wafer pattern classification and auto disposition by machine learning, in: Proc. Joint Int. Symp. e-Manuf. Design Collaboration (eMDC) Semiconductor Manuf.(ISSM), 2017, pp. 3–5.

[6] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, U. Lee, Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes, IEEE Transactions on Semiconductor Manufacturing 31 (2018) 315–322.

[7] Y. Kong, D. Ni, Recognition and location of mixed-type patterns in wafer bin maps, in: 2019 IEEE international conference on smart manufacturing, industrial & logistics engineering (SMILE), IEEE, 2019, pp. 4–8.

[8] J. Yu, J. Liu, Two-dimensional principal component analysis-based convolutional autoencoder for wafer map defect detection, IEEE Transactions on Industrial Electronics 68 (2020) 8789–8797.

[9] Y. Ji, J.-H. Lee, Using gan to improve cnn performance of wafer map defect type classification: Yield enhancement, in: 2020 31st annual SEMI advanced semiconductor manufacturing conference (ASMC), IEEE, 2020, pp. 1–6.

[10] J. Wang, Z. Yang, J. Zhang, Q. Zhang, W.-T. K. Chien, Adabalgan: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition, IEEE Transactions on Semiconductor Manufacturing 32 (2019) 310–319.

[11] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, et al., Paligemma: A versatile 3b vlm for transfer, arXiv preprint arXiv:2407.07726 (2024).

[12] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al., Gemma: Open models based on gemini research and technology, arXiv preprint arXiv:2403.08295 (2024).

[13] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, D. Parikh, Vqa: Visual question answering, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2425–2433.

[14] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner,

B. Mustafa, L. Beyer, et al., Pali: A jointly-scaled multilingual language-image model, arXiv preprint arXiv:2209.06794 (2022).

[15] M.-J. Wu, J.-S. R. Jang, J.-L. Chen, Wafer map failure pattern recognition and similarity ranking for large-scale data sets, IEEE Transactions on Semiconductor Manufacturing 28 (2015) 1–12.

[16] M. Fan, Q. Wang, B. van der Waal, Wafer defect patterns recognition based on optics and multi-label classification, in: 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), IEEE, 2016, pp. 912–915.

[17] J. Wang, C. Xu, Z. Yang, J. Zhang, X. Li, Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition, IEEE Transactions on Semiconductor Manufacturing 33 (2020) 587–596.

[18] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[19] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, Advances in neural information processing systems 30 (2017).

[20] Z. Liu, Y. Zhou, Y. Xu, Z. Wang, Simplenet: A simple network for image anomaly detection and localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 20402–20411.

[21] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[22] 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, 2016, pp. 770–778.

[23] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, arXiv preprint arXiv:2305.14314 (2023).

[24] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, L. Yuan, Florence-2: Advancing a unified representation for a variety of vision tasks, arXiv preprint arXiv:2311.06242 (2023).