IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 August 2022, accepted 29 August 2022, date of publication 5 September 2022,
date of current version 26 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204278

APPLIED RESEARCH

Explainable Deep Learning System for Advanced Silicon and Silicon Carbide Electrical Wafer Defect Map Assessment

RICCARDO EMANUELE SARPIETRO[1], CARMELO PINO[1], SALVATORE COFFA[1], ANGELO MESSINA[1], SIMONE PALAZZO[2], SEBASTIANO BATTIATO[3], (Senior Member, IEEE), CONCETTO SPAMPINATO[2], AND FRANCESCO RUNDO[1]

[1]STMicroelectronics, ADG Research and Development Power and Discretes, 95121 Catania, Italy
[2]Pattern Recognition and Computer Vision Laboratory (PeRCeiVe Lab), Department of Electrical, Electronics and Computer Engineering, University of Catania, 95124 Catania, Italy
[3]Image Processing Laboratory (IP-LAB) Group, Dipartimento di Matematica e Informatica (DMI), University of Catania, 95124 Catania, Italy

Corresponding author: Francesco Rundo (francesco.rundo@st.com)

This work was supported in part by the REACTION "first and euRopEAn siC eigTh Inches pilOt liNe"-Horizon 2020 Program under Grant 783158, and in part by the PIACERI "Programma ricerca di ateneo UNICT 2020–2022 linea 2," Università di Catania through the Project: Safe and Smart Farming With Artificial Intelligence and Robotics.

ABSTRACT The recent increasing demand of Silicon-on-Chip devices has triggered a significant impact on the industrial processes of leading semiconductor companies. The semiconductor industry is redesigning internal technology processes trying to optimize costs and production yield. To achieve this target a key role is played by the intelligent early wafer defects identification task. The Electrical Wafer Sorting (EWS) stage allows an efficient wafer defects analysis by processing the visual map associated to the wafer. The goal of this contribution is to provide an effective solution to perform automatic evaluation of the EWS defect maps. The proposed solution leverages recent approaches of deep learning both supervised and unsupervised to perform a robust EWS defect patterns classification in different device technologies including Silicon and Silicon Carbide. This method embeds an end-to-end pipeline for supervised EWS defect patterns classification including a hierarchical unsupervised system to assess novel defects in the production line. The implemented "Unsupervised Learning Block" embeds ad-hoc designed Dimensionality Reduction combined with Clustering and a Metrics-driven Classification Sub-Systems. The proposed "Supervised Learning Block" includes a Convolutional Neural Network trained to perform a supervised classification of the Wafer Defect Maps (WDMs). The proposed system has been evaluated on several datasets, showing effective performance in the classification of the defect patterns (average accuracy about 97%).

INDEX TERMS Artificial intelligence, convolutional neural network, explainable architectures, hierarchical clustering.

I. INTRODUCTION

The semiconductor-based technological development process has led to a revolution which has impacted all innovation fields including communications, computing, artificial intelligence, medical devices, and so on.

The wide application of semiconductor has enabled the emergence of new markets. The global chip shortage led by automotive industry for vehicle electrification or a supply-chain related issues caused by geopolitical crises in the East countries for rare-earth elements management has seriously questioned the value of semiconductor as a strategic asset.

Framing the chip shortage as business problem brought semiconductor industry to increase fab investments, but this

The associate editor coordinating the review of this manuscript and approving it for publication was Okyay Kaynak.

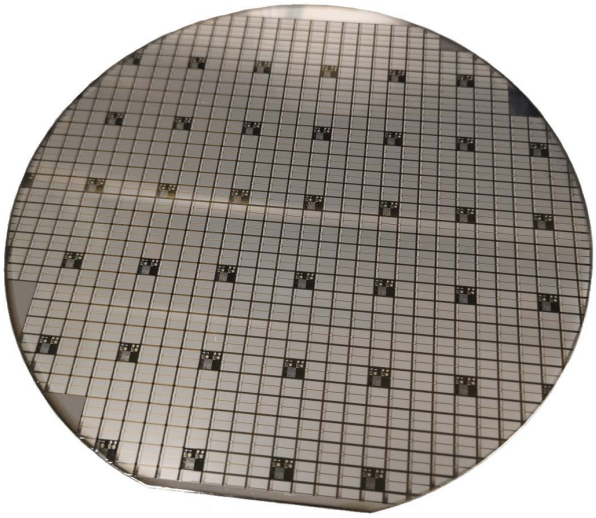


FIGURE 1. 6″ Silicon carbide wafer.

short term solution needs time frame to build new factories. Therefore, industry needs to develop effective and efficient solutions to properly satisfy this growing demand without new investments and costs.

Furthermore, the optimization of Ultra Very Large Scale Integration (UVLSI) process and the introduction of new technologies, such as Silicon Carbide (in Fig. 1 a Silicon Carbide Wafer is reported) that will replace current Silicon technology in the high-power and high-temperature applications (thanks to its efficiency and switching properties given by the physical behaviour [1]), generates new production issues that makes defect patterns analysis more difficult.

The manufacturing of an integrated circuit goes through two main steps, the Front-End and Back-End phases [2].

The first one is related to the manufacturing process of the die (device), while the second one includes the remaining part of the production process including the packaging [2], [3], [4].

Wafers fabrication process requires several chip-probing (CP) such as the front-side metallization, backside grinding and metallization and so on [2], [3], [4].

Significant production-process drift can be generated in all device manufacturing steps which consequently produces defects in the wafers.

The wafer defect patterns identification (during the production phase) is one of the key mode for improving production performance of a semiconductor company [5].

To automate the wafer defects detection, a robust characterization of the patterns embedded in the wafer surface is needed. In particular, the analysis of the geometric morphology of these defect patterns provides a sort of fingerprint that can be efficiently used to retrieve the cause that generated the manufacturing process-drift and consequently correlates it with the production yield [6].

One of the most used approaches to characterize production defects in semiconductor wafers is based on the visual

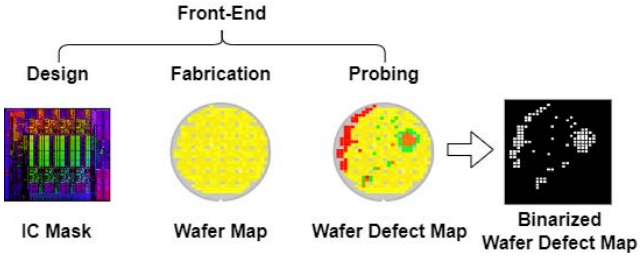


FIGURE 2. Front-end pipeline description for binarized WDM generation.

analysis of the defect maps at Electrical Wafer Sorting (EWS) stage in which a series of electrical conformance-tests will be performed (short-circuit tests, leakage, parasitic capacitance, and so on) [7].

Specifically, the EWS binarized Wafer Defects Maps (WDMs) are considered as excellent tool for identifying predictive markers of production yield or issues in the upstream manufacturing lines.

More in detail, the binarized WDMs are obtained at the end of the Front-End manufacturing process (Fig. 2) where designed devices are emebedded in disc-shaped wafers and tested by a probing machine. The probing machine verifies the device functionality through electrical tests, assigning a test-outcome color to each device and by distinguish them in fully, partially or not working devices thereby creating a defect map. The binarization of WDMs consists in assigning the white color (value ''1'') to partially or not working devices while the black color (value ''0'') is assigned to full working devices and background.

The pipeline herein proposed is based on the wafer defects analysis at the end of the Front-End manufacturing, i.e., when the binarized WDM has been generated. Therefore, from a careful monitoring of the so generated WDM, semiconductor manufacturers will be able to build correlation models with the issues upstream the production lines or to predict the impact on the production yield of a specific defect pattern, defining properly policies of recovery.

The main contribution of this work is the development of a deep pipeline for a robust and intelligent classification of defect patterns both in Silicon (Si) technology and in the production of Silicon Carbide (SiC) devices. In subsequent development (currently being designed) we will deal with the correlation between the classified wafer defect patterns and the issues upstream the production process and therefore with the related yield.

This work is arranged into three main sections: related works where several approaches to assess defect pattern recognition problem are briefly described, materials and methods where the proposed approach is discussed from mathematical and computational perspectives, experiments and results section in which the performance and benchmark comparisons of the designed approach will be outlined. The final section will also include a description of the delivered tool named STAI-EWS. This tool embeds the pipeline described in this contribution and it is currently in use in Silicon and Silicon Carbide technology production lines.

II. RELATED WORKS

Deep Learning solutions for addressing semiconductor application issues related to pattern recognition, semantic segmentation and classification have grown significantly in the last few years. Several researchers investigated different approaches of WDMs recognition by using deep learning based on supervised, unsupervised and hybrid approaches. Most of these works have been evaluated on public datasets or by using internal data or synthetic ones.

A. SUPERVISED-LEARNING-BASED APPROACH

Several solutions based on deep architectures with a supervised-learning paradigm have been proposed in scientific literature. In [8] a basic 3-layers Convolutional Neural Network (CNN) has been designed in order to classify 22 different simulated WDM classes by using a Poisson Point Process [9] approach with an overall accuracy of 98.20% in test set.

In [10] leverage a novel Information Gain (IG)-based splitter with a spatial filtering to remove random noise over the WDMs was proposed. The authors proposed a general regression network (RGRN) model to identify and classify both single-defect and mixed-defect patterns. The latter method showed very promising results with 99.51% of accuracy for single defect patterns and 86.00% for mixed ones.

In [11] authors delivered a modified VGG-19 architecture with ad-hoc drop-out system to classify out-of-distribution data with WDMs rotated by 5 degree to match the pattern distribution and maximize the correlation with reference image. This method seems very effective with an accuracy on test set (1, 311 Wafer Maps) of 97.71% while with out-of-distribution accuracy of 97.18%.

An interesting approach has been proposed by [12] where the authors combined four classification models: Each classifier involves a 3-layers CNN with a downstream stack of two fully connected layers to classify a synthetic dataset of WDMs virtually generated using real distribution based approach [13]. The performance of the showed method confirmed the usefulness of using synthetic datasets to improve the performance of the deep classifier, reaching 91.00% in accuracy over a severely noisy dataset and 97.40% over a moderately noisy one.

Other authors designed and evaluated pre-trained (on classical ImageNet or COCO or KITTI dataset) deep models such as DenseNet-169 [14] or R-CNN [15] to leverage transfer learning approach in order to improve the performance of the underlying deep classifier in defect patterns assessment. The performance of the analyzed methods showed that DenseNet-169 reachs 87.70% in test set while R-CNN an overall accuracy of 97.73%.

An interesting approach has been presented in [16] by proposing an Ensemble Convolutional Neural Network based on LeNet, AlexNet and GoogleNet with a weighted majority function based on models' output weights. By multiple experiments varying learning rates and optimizers they achieved an overall accuracy of 98.57% on WM-811K dataset.

B. UNSUPERVISED-LEARNING-BASED APPROACH

In [17] researchers proposed a Gaussian Mixture of Variational Autoencoder (GMVAE) where extracted visual features from the source WDM and by means of an ad-hoc Dirichlet process they were able to provide a robust WDMs clustering. This approach has been benchmarked against traditional Bayesian non-parametric models using the adjusted rand index (ARI) and adjusted mutual information (AMI) as measure of similarity between clusters, obtaining 0.76 as highest values in both ARI and AMI.

In [18] authors proposed a pre-processing statistical technique on a custom dataset containing 6 wafer lots, consisting in a binarization of wafer maps, filling the inner testing wafer points on the wafer using the around median value and reducing the noise using a median filter. At the end of the pre-processing stage, variational autoencoders are used as feature extractors to decompose high-dimensional wafer maps to a low-dimensional latent representation. Finally, a traditional K-means or hiearchical clustering were involved and similarity evaluated by Silhouette Score. Unfortunately, the mentioned authors provided only the 2D-latent plot representation of their method without any performance metric.

In [19] authors proposed a Siamese CNN which learned an embedding space based on similarities of WDM images. A G-means clustering as hierarchical clustering pipeline has been used as downstream block to find the optimal clusters distribution. The authors applied and evaluated their solution on classical public WM-811K dataset. The hybrid approach confirmed a promising effectiveness of 91.20% in accuracy with a corruption ratio of only 10% down to 64.20% with a corruption of 40%.

C. HYBRID-LEARNING-BASED APPROACH

The authors of [20] proposed a combination of three techniques based on distributed K-Means++ for clustering as well as a statistical mining patterns by FPGrowth [21] and finally a deep classifier based on a 5-layers CNN backbone for making a robust defect maps classification of a custom input wafer dataset. The method seems very promising as they collected 95.00% in F1-score.

The authors of [22] proposed a Stacked Convolutional Sparse Denoising Auto-Encoder (SCSDAE) in which the designed convolutional layers were used to extract wafer visual features. The so collected features will be processed by the auto-encoder part of the architecture in order to retrieve an internal unsupervised latent representation of those features suitable to perform a robust features-related defects clustering. The method showed 95.13% of accuracy using a 5-fold cross validation.

A promising approach has been showed in [23] in which the authors proposed an approach based on dimensionality reduction of the input defect maps distribution followed by an autoencoder based processing. Specifically, the input defect maps were fed to Principal Component Analysis (PCA) that extracts features. The so collected features will be processed

by the downstream auto-encoder which tried to reconstruct the input embedded patterns from internal latent representation. This approach has been evaluated on WM-811K dataset with an accuracy of 97.27% in a 5-fold cross validation setting.

The authors of [24] proposed an Adaptive Balanced GAN based approach to preliminary improve the source WDMs class balancing. The authors trained the GAN embedded discriminator to assess the differences between the synthetic wafer maps with respect to the real ones. Further, the discriminator will be used as deep classifier of both (synthetic and real) WDMs, reaching an accuracy of 96.00% in all the 9 classes of WM-811K dataset.

A novel approach has been proposed in [25] in which Hybrid Quantum Deep Learning is applied by transforming WDMs of WM-811K dataset in feature maps using self-proliferation and self-attention (SP&A) blocks and compared the proposed approach against traditional Deep Learning approaches (i.e., CNN) reaching an overall accuracy of 98.10%.

Authors in [26] proposed a self-supervised learning approach based on Convolutional Auto-Encoders and Dirichlet process. Convolutional Auto-encoders is used to extract meaningful features from WDMs input based on WM-811K dataset, these features are then clustered by the Dirichlet process mixture model (DPMM) imposing pseudo-labels and the previous CAE is fine-tuned in a self-supervised learning fashion. With this approach they achieved a weighted-macro accuracy of 96.10%.

Our proposed deep learning pipeline can be configured as a hybrid deep learning approach for classification of defect patterns related not only for devices Silicon (Si) technology based but also for Silicon Carbide (SiC) ones. The aforementioned methods proposed in the last years by industry and academy, are mainly based on WM-811K public dataset with performance tricked by its class imbalance (a more detailed description can be found in IV-B1). Our proposed approach has been trained on multiple datasets. The related robustness has been evaluated and compared by using Explainable Artificial Intelligence (XAI) methods between the designed CNN and State-Of-The-Art architectures implemented for similar applications. The implementation of the proposed pipeline in the form of a Web-application represents a valid tool for semiconductor manufacturing allowing a robust production failures assessment.

III. MATERIALS AND METHODS

The authors propose in this contribution a hybrid deep learning approach for WDMs assessment where the overall scheme of the proposed full hybrid pipeline is reported in Fig. 3.

The first part of the designed pipeline is the ''Unsupervised Learning Block'' which embeds four sub-systems: The Resize and Filter, Dimensionality Reduction followed by Clustering and Metrics-driven Classification.

The second part of the proposed pipeline is composed by the ''Supervised Learning Block'' structured with ad-hoc designed Convolutional Neural Network trained to perform a supervised classification of the resized input WDMs.

As introduced, in the common semiconductor production lines there is a concrete need to correctly identify and classify defect patterns as predictive markers of manufacturing issues and production yield. Furthermore, it becomes necessary to characterize novel and unknown defect patterns related to a new issues in the upstream production process which needs to be properly investigated.

The classical manufacturing issues which produces wafer defects mainly concern to failures, impurities or degradation of the production lines [27], [28]. For the work herein described, it is worth mentioning the case of Silicon Carbide (SiC). The SiC-based manufacturing pipelines show defect patterns which are usually significantly different with respect to the silicon-based ones (a more detailed description about datasets can be found in IV-A1 and IV-B1). For this reason, an advanced ''unsupervised'' pipeline suitable to identify new defect patterns is investigated. In this way, the herein proposed pipeline will be able to catch new issues in the upstream production lines, through a hybrid approach (unsupervised / supervised) that will be able to correctly characterize defect patterns.

Each of the designed parts of the proposed full pipeline will be described in the following sub-sections.

A. UNSUPERVISED LEARNING BLOCK

The designed Unsupervised Learning Block is composed by four sub-systems: Resize and Filter, Dimensionality Reduction, Hierarchical Clustering and Metrics-driven Classification. Each of the mentioned sub-systems will be described in detail.

1) RESIZE AND FILTER SUB-SYSTEM

The input of this sub-system is the high-resolution binarized WDMs (usually at classical wafer dimension, i.e., $20,000 \times 20,000$ spatial resolution) resized (using bicubic algorithm [29]) to ad-hoc reduced spatial dimension by the resize block. From our internal investigation, an optimal resolution for the herein analyzed application is 61×61. However, the spatial resolution resizing does not have any significant impact on the overall performance of the proposed unsupervised pipeline to the extent that the defect patterns information are preserved. Due to the adopted Resize Block, each pixel of the processed WDM image no longer represents a single die (device) but may represent a set of dies (devices) according to the adopted photo-lithography process [30].

Before applying the dimensionality reduction and hierarchical clustering techniques as reported in Fig. 3, a Filter Block is preliminary applied to the resized input WDM. This filter discards defect maps whose patterns show a low-impact in the upstream production issues. Specifically, a wafer map showing few defective dies (i.e., the so called ''Spot Wafer Map'' as in Fig. 4a or a defect map with no defective dies

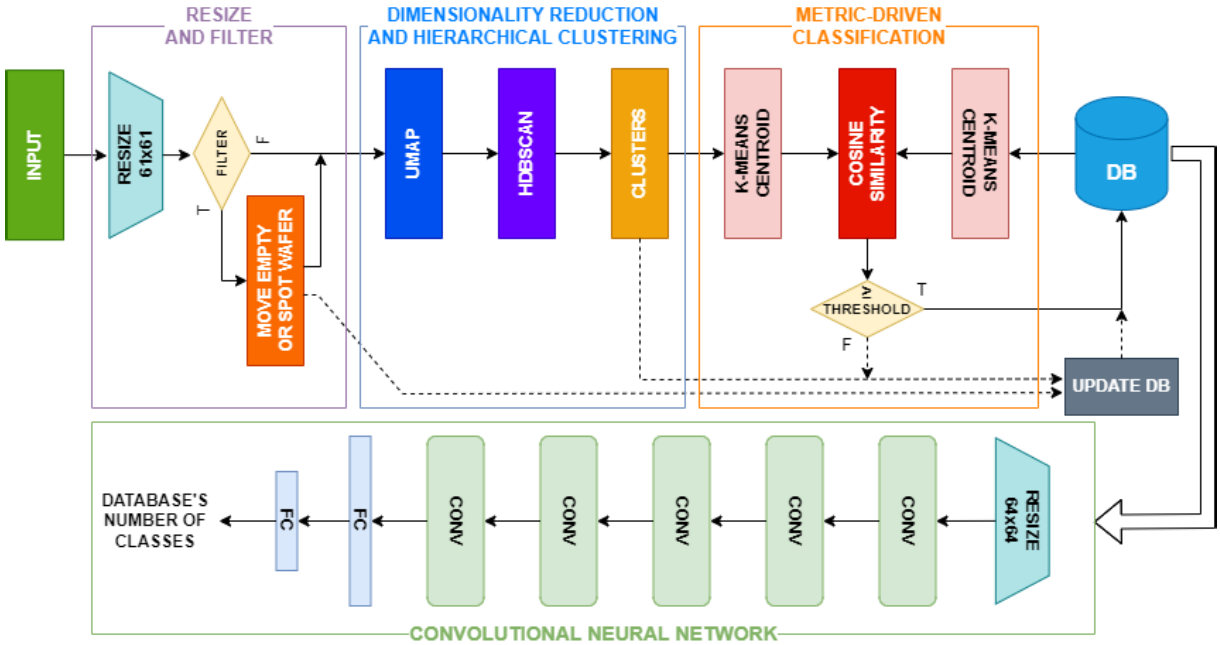


FIGURE 3. The overall scheme of the proposed pipeline.

(i.e., the so called "Empty Wafer Map" as in Fig. 4b) will be discarded as they do not produce any significant impact in the production lines but only computational cost of the pipeline. In order to characterize the defect maps as "Spot" or "Empty" ad-hoc thresholds have been defined.

As reported in Fig. 3 the introduced Resize and Filter sub-system will enable dimensionality reduction and clustering to speed up computation. The Resize and Filter sub-system will update the WDMs to the internal database accordingly.

2) DIMENSIONALITY REDUCTION SUB-SYSTEM

The target of this sub-system is to perform ad-hoc dimentionality reduction of the visual features distribution extracted from the WDM inputs.

After the Resize and Filter sub-system, defect map images are fed as input in the "Dimensionality Reduction Sub-System" which has the target to reduce the dimensional-complexity of the pre-processed input wafer maps. To perform the aforementioned dimensionality reduction, the proposed pipeline embeds an approach based on Uniform Manifold Approximation and Projection (UMAP) algorithm [31].

UMAP algorithm has the target to reduce an input high-dimensional connected graph into a projected low-dimensionality space. The goal of UMAP is to keep the high-dimensionality space-features into the projected low-dimensionality space by using the so called Riemannian manifold [32].

More in detail, the pre-processed input WDM images at 61×61 resolution will be flattened and reshaped to $3,721$ dimensions. In order to fed this reshaped input vector into UMAP, the authors have assumed that each sample (of the so reshaped vector) will represent a specific dimension in the related high-dimensional space as total of $3,721$ dimensions. These samples are known as data-points. This dimensional reduction approach is a key-process of the WDMs unsupervised clustering. For this reason, the unsupervised sub-system was designed to process batch of WDMs, specifically, the whole set of wafers produced at each production cycle (from our tests 350 wafer maps based on Silicon Carbide technology were processed - on average - per week cycle).

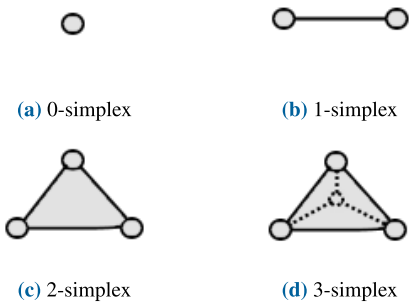
(a) Spot (b) Empty

FIGURE 4. Spot and empty wafer maps.

The UMAP algorithm is based on the following main parts: High-Dimensionality-to-Graph Block and Graph projection Block.

B. HIGH-DIMENSIONALITY-TO-GRAPH BLOCK

The target of this block is to build a weighted graph associated to the input set of WDMs (high-dimensional space). Let introduce such mathematical assumptions needed to reduce the dimensionality of the high-dimensional space associated to wafer maps. Specifically, the authors have assumed that data-points are uniformly distributed over the input high-dimensional space. Considering that this assumption is not always satisfied in a real application, we have applied a Riemann's metric (G_r) that allows to consider input data-points as uniformly distributed in the input space, thus making



(a) 0-simplex (b) 1-simplex

(c) 2-simplex (d) 3-simplex

FIGURE 5. Low dimensional simplices.

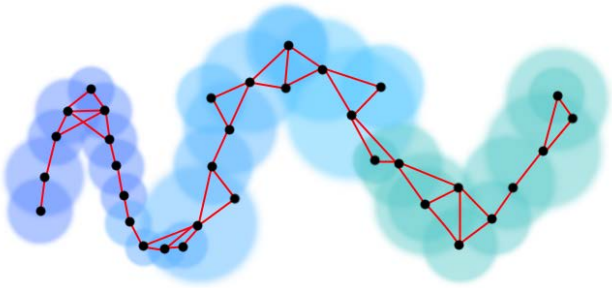


FIGURE 6. An instance of 2D KNN generated graph.

mathematical assumptions robust [32]. Further mathematical assumptions are given below:

Assumption A_1: Given a set of data uniformly distributed on the manifold M (respect to the related Riemann's metric G_r), for each point in M there exists a correlated point G_r^p on the tangent space T_pM of the manifold M.

Assumption A_2: The aforementioned Riemann's metric is locally constant, i.e., given a ball of fixed volume, it contains the same number of points regardless the position on the manifold M.

Assumption A_3: If the assumption A_1 and A_2 are satisfied then Riemann geometry theory confirmed that the manifold M is locally connected.

In order to build the high-dimensional weighted graph from the mentioned uniformly distributed data-points, we have to introduce the concept of simplicial complex [33] and K-Nearest Neighbour [34].

More in detail, the cited manifold M will be created by using such elements of the simplicial complex such as: points (Fig. 5a), line segments (Fig. 5b), triangles (Fig. 5c). Each of the mentioned elements can be each combined and connected together (like tetrahedron Fig. 5d) in order to create a n-dimensional object.

In the input high-dimensional space the data-points are connected along edges (through simplicial complex elements). An edge can be defined as topological structure after data-points connection. The concept of "edge" is correlated to the concept of "weight", i.e., a measure of distance (in the Riemann geometry meaning) between edges [31].

As introduced, K-Nearest Neighbour (KNN) [34] approach was used to graph construction in the input high-dimensional space. Specifically, by ad-hoc changing of "k" parameter of KNN algorithm we are able to build a k-complexity dependent and weighted graph by connecting the edges. A more detailed graph-structure can be obtained by using small "k" value where data-points are inside a dense region in the manifold M. Otherwise, for large "k" value, a sparse-graph structure will be generated.

More in detail, the graph is generated as follow: Assume the parameter "k" as a circle radius around each data-point in the input high-dimensional space. This circle radius can be extended or shrinked in order to connect each data-point with others. By changing the circle radius (k parameter) we are able to connect more or less data-points to the graph. An instance of so generated 2D simple graph is reported in Fig. 6.

After that, we normalize the measure of distance between the edges in the graph (i.e., the weights) by associating a fuzzy topology representation of the graph in which distance values may change between zero and one [31].

The use of UMAP allows to obtain considerable advantages (compared to classical dimensionality reduction techniques such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD), t-Distributed Stochastic Neighbor Embedding (t-SNE)) as it allows to preserve the global and local features of the high-dimensional input space into the projected low-dimensional space by optimizing the degree of dimensionality for feature representation.[1]

C. GRAPH PROJECTION BLOCK

The second step of the UMAP algorithm is the input high-dimensional graph projection into low-dimensional ones. Basically, with this step the authors want to build a new low-dimensional weighted graph by optimizing a cross-entropy-based function that embeds the weights associated to the edges of both graphs (the input high-dimensional and the projected ones to be defined by the optimization process). In Eq. 1 the adopted cross-entropy $\Theta(\mu, \upsilon, A)$ function is reported:

$$\Theta(\mu,\upsilon,A) = \sum_{\alpha\in A} \overbrace{\mu(\alpha)\log\left(\frac{\mu(\alpha)}{\upsilon(\alpha)}\right)}^{\text{right group}} + \underbrace{+\,(1-\mu(\alpha))\log\left(\frac{1-\mu(\alpha)}{1-\upsilon(\alpha)}\right)}_{\text{right gap}} \tag{1}$$

where A is a reference set, i.e., the set of the input high-dimensional wafer defect data-points, μ and υ are the related weights defined in $\alpha \to [0, 1]$ due to mentioned fuzzy representation. In the so created low-dimensional space (due to the previously optimization) a connected graph is associated. The output of UMAP processing is then a set of features

[1]This particular phenomenon is called Curse of dimensionality, the number of dimensions needed to represent features grows exponentially with the increasing amount of data.

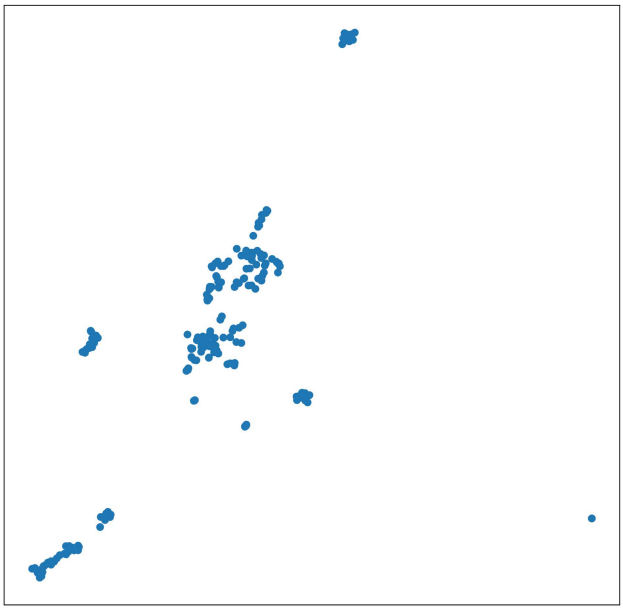


FIGURE 7. An instance of 2D projected low-dimensional space of input silicon carbide WDMs.

in low-dimensional space (with associated graph) retrieved from the input WDMs (data-points). We have defined as Φ this set of UMAP output data-points features.

An instance of 2D UMAP-projected space of an input Silicon Carbide WDMs is reported in Fig. 7.

The so obtained low-dimensional data-points will be fed as input to the ''Hierarchical Clustering Sub-System''.

1) HIERARCHICAL CLUSTERING SUB-SYSTEM

Lowered dimension input WDMs arranged as data-points Φ are fed as input to the designed Hierarchical Clustering Sub-System based on the usage of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [35]. The use of HDBSCAN allows to obtain considerable advantages compared to classical parametric clustering techniques such as K-Means and Gaussian Mixture Models (GMM) or non-parametric clustering such as DBSCAN [36] and Mean shift. HDBSCAN allows to create a hierarchy of clusters based on density and not on heuristically predefined parameters improving both cluster separation and cohesion.

HDBSCAN is a clustering algorithm extending old DBSCAN approach, where the key-part is the ''core-object'', defined as follow.

The result of the previous Dimensionality Reduction sub-system is a $m \times n$ matrix containing distances of m data-points in the lowered n dimensional space. An object a_p is defined ''core-object'' with respect to a radius r and a smoothing factor m_p, if drawing a circumference with radius r and centered in a_p, it is possible to identify a minimal set of data-points [37]) within the circumference. The circumference is usually indicated with the term ''r-neighborhood'' while the data-points outside the ''r-neighborhood'' are defined as ''noise''. After that, we leveraged the following definitions related to core-object:

Definition D_1*:* Two core-objects are considered r-reachable if data-points in the related core objects are nested all together;

Definition D_2*:* N core-objects are density-connected if they are directly or transitively r-reachable;

Definition D_3*:* A cluster (C) can be defined with respect to its radius (r) and smoothing factor (m_p), as non-empty subset of density-connected core-objects;

We can also define other properties related to distance between core-objects:

Definition D_4*:* The core distance d_{core} of a core-object a_p (with reference to its radius r and smoothing factor m_p) is the distance between a_p to its nearest neighbor in m_p;

Definition D_5*:* A core-object is considered r-core-object if the correlated radius r is greater than or equal to the core distance of a_p.

After the core objects definition, HDBSCAN provides an internal graph reconstruction starting from input low-dimensional data-points and core-objects previously defined.

This graph is usually named as Mutual Reachability Graph and it is defined as:

Definition D_6*:* Mutual Reachability Graph is a weighted graph with the data-points configured as graph-vertices while for each edge (data-points connection) ad-hoc weights are defined as measure of the mutual reachability distance of related data-points.

Definition D_7*:* Mutual reachability distance d_{mr} is defined as the maximum distance between core distance a_p, core distance a_q and the distance between the two core-objects a_p and a_q. In Eq. 2 the mathematical representation of the d_{mr}.

$$d_{mr}(a_p, a_q) = \max\{(d_{core}(a_p), d_{core}(a_q), d(a_p, a_q)\} \quad (2)$$

At this point, HDBSCAN provides a mutual reachability graph by connecting core-objects and by weighting the connection through the mutual reachability distance d_{mr}.

Through ad-hoc thresholding applied to the overlapping edges of the mutual reachability graph, the mutual reachability graph connection scheme can be re-configured by optimizing the number connection-complexity. To do that, HDBSCAN embeds the usage of Minimum Spanning Tree (MST) approach [35], [38]. MST re-configures and reduces in complexity the input densely connected graph by a classical graph-theory approach which provides a new graph with a minimal set of edges that connects all the components.[2] An instance of MST optimized graph associated to an input Silicon Carbide WDMs is reported in Fig. 8.

The target of the unsupervised pipeline which embeds UMAP and HDBSCAN is to provide a final hierarchical structure which highlights the key group of clusters associated to the input set of similar wafer defect patterns. Based on the performed analysis, we have obtained a non-hierarchical MST optimized and densely connected weighted graph.

[2]A more detailed explanation about HDBSCAN can be found at [35].

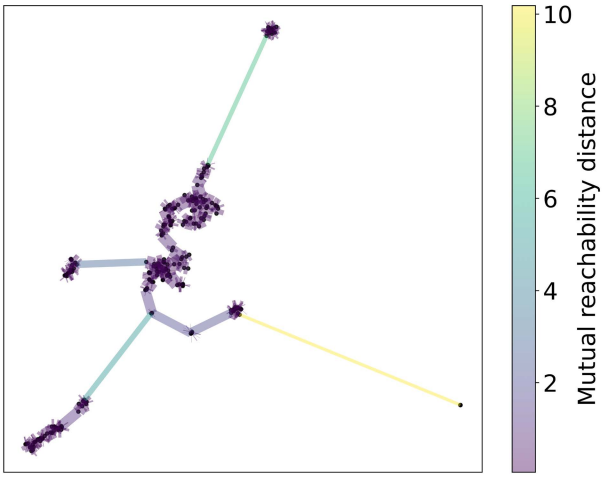


FIGURE 8. An instance of minimum spanning tree optimized graph.

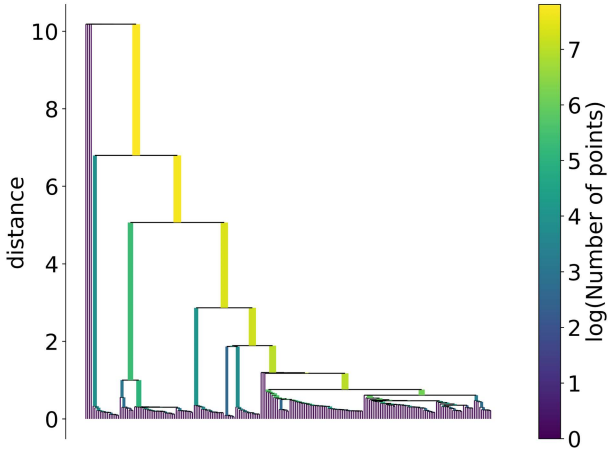


FIGURE 9. An instance of dendrogram of input silicon carbide WDMs.

Therefore it is necessary to construct from this graph a hierarchical structure which specifically highlights the groups of clusters associated with the patterns of the input wafer set. For this reason graph edges sorting by their tree distance in increasing order have been applied.

More in detail, hierarchical re-configuration of the graph can be described with a traditional dendrogram representation [39] which highlights the number of clusters and the distance between those clusters. An instance of dendrogram is reported in Fig. 9. Anyway, dendrogram is not a suitable visualization for the final assessment of the optimal number of clusters and needs some heuristic assumptions to generate the clusters distribution (for instance the cluster size parameter has to be defined heuristically) [39].

To overcome this issue, HDBSCAN proposed the usage of ''excess of mass'' as a method to extract an optimal number of clusters.

Given the probability density function, components of the dendrogram can be disconnected or connected accordingly to the density function $f(x)$.

Basically, by ad-hoc increasing or decreasing density λ parameter related to cluster's density, computed as $\frac{1}{d_{core}}$,

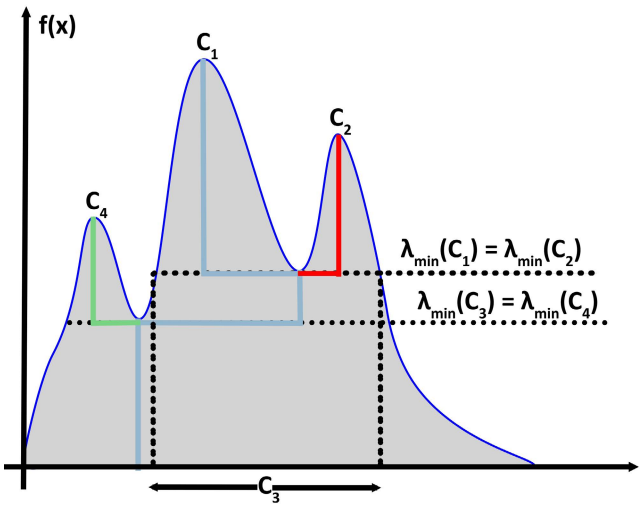


FIGURE 10. Diagram of the "excess of mass" approach applied to clustering.

clusters (C_i) are split or merged according to the density value λ_i, where eligible clusters are the one that will survive at the λ density changes.

The Eq. 3 reports the mathematical integral equation related to the ''excess of mass''. In Fig. 10 we reported an instance of excess of mass approach on the probability density function of clusters.

$$E_{dm}(C_i) = \int_{x \in C_i} (f(x) - \lambda_{min}(C_i))dx \tag{3}$$

Through the approach described by Eq. 3 and in Fig. 10 the authors were able to retrieve the optimized number of clusters C_i through density λ_i. For instance, the two meaningful clusters C_1 and C_2 are merged at the corresponding minimum density level λ_{min} related to C_1 and C_2 and create cluster C_3, then merged cluster C_3 will be merged with another cluster C_4 according to the minimum density level λ_{i+1}, and so on.

Finally, the output of HDBSCAN sub-system is a set of core-objects representing the final set of optimized clusters. These defined clusters will be re-mapped back to the UMAP block in order to associate them to source data-points. In Fig.11 an instance of the mentioned UMAP re-mapping is reported.

In Fig. 11 clusters related to input defect maps (in Silicon Carbide technology) with significant similar features have been highlighted and grouped by color.

At the end, the set of optimized clusters will be processed by the following Metrics-driven Classification Sub-System.

2) METRICS-DRIVEN CLASSIFICATION SUB-SYSTEM

In details, the target of this sub-system is to assess the matching between the identified defect map clusters (from UMAP and HDBSCAN) with the well-classified defects classes stored in the database available in the pipeline.

To do that, we have integrated the K-Means approach [40] with the target to retrieve only one centroid for each cluster (basically K=1). K-means is also applied to the well classified

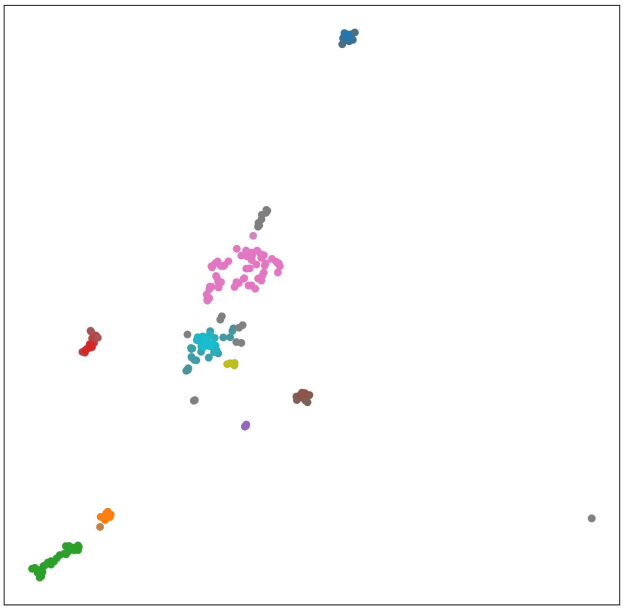


FIGURE 11. UMAP cluster-to-data-point plot related to silicon carbide WDMs input.

wafer defect patterns stored in the internal database embedded in the proposed pipeline (see Fig. 3). K-means centroids represent the mean value of the WDMs in the clusters and in well-classified WDMs stored in the database. In order to compare them, a Cosine Similarity metric has been defined. The following Eq. 4 showed the applied metric comparison:

$$cos(\theta) = \frac{WDM_{new} \times WDM_{DB}}{\|WDM_{new}\| \times \|WDM_{DB}\|} \tag{4}$$

where WDM_{new} is the computed K-means cluster centroid related to the WDM clusters while WDM_{DB} is the same related to the well-classified WDMs stored in the Database. The cosine similarity score ranges from -1 to 1, as -1 represents high dissimilarity of the centroids while, conversely, 1 represents high similarity of the input data.

We have defined ad-hoc threshold of 0.90 as good trade-off to define the full similarity on the WDMs. Basically, in the case of high similarity (similarity beyond the pre-determined threshold) of the cluster-centroid with ones of the reference well-classified centroids, the input defect maps will be considered similar to the compared class already stored in the internal database and therefore they will be added to the database and stored as belonging to that specific class. Otherwise, in the case of dissimilarity with all the classes already stored in the database, a new class will be created associated with the defect maps from which the clusters were generated.

After the so described unsupervised processing of the source defect maps (and the following update of the database on the basis of the similarity check previously described) the proposed pipeline enables the supervised classification of the defect maps as described in the next section.

D. SUPERVISED LEARNING BLOCK

The designed Supervised Learning Block takes as input the defect patterns stored in the internal database possibly updated by the Unsupervised Learning Block.

To perform the mentioned supervised classification, ad-hoc designed Deep Convolutional Neural Network has been implemented. It is composed by 5 convolutional layers having a kernel size 3×3, padding and striding set to 1. For each convolutional layer a ReLU activation function followed by a Batch Normalization are applied. The number of kernels is doubled at each layer, starting from 64 till to 512. Starting from the second layer a Max-pooling of size 2×2 and striding set to 2 is applied. The so designed Convolutional Neural Network backbone is described in details in Table 1. Specifically, we have designed two type of deep convolutional network backbones (differentiating the input layer and the final layers that embed the fully connected) in order to validate the best of these in performance and to facilitate the benchmark comparison phase. More details about the two implemented backbones are now given.

The **Big CNN**. This first backbone embeds an input layers at $224 \times 224 \times 3$ as data resolution/channels while shows a final stack of two fully connected layers which embeds 100, 352 and 1, 024 neurons respectively.

The **Small CNN**. This second backbone embeds a single-channel input layer at 64×64 and a final set of two fully connected layers is composed by 8, 192 and 1, 024 neurons respectively. As introduced, the need to have two deep architectures is mainly for performance validation as well as in reference to a more robust benchmarking of the proposed solution as some scientific literature solutions with which our method has been compared have inputs of $224 \times 224 \times 3$ or single-channel. In Table 1, the details of the implemented deep backbones.

As reported in Table 1, the final number of well defined WDM classes has been defined to 45. Although this number can vary significantly according to the new classes that may emerge from the unsupervised clustering block. More details about this defect map classes are reported in the next sections.

The experimental results we have collected were related to this setup although similar considerations can be extended to any number of defect map classes. However, an attempt is made to minimize the number of defect pattern classes in order to efficiently characterize production. Furthermore, as new defect classes are identified, they are analyzed and resolved in the upstream production line, thus contributing to the maintenance of a minimum number of defect classes.

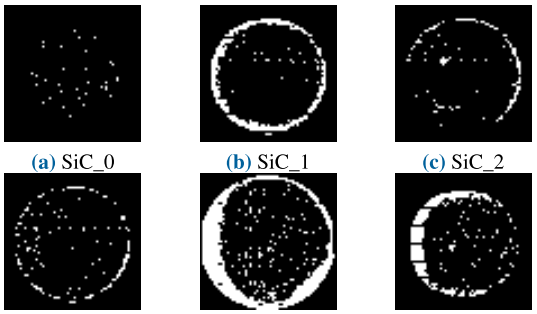
IV. EXPERIMENTS AND RESULTS

This section reports experimental results for Unsupervised Learning and Supervised Learning approach and some details about the STAI-EWS application we have developed to perform the test through an user-friendly tool.

TABLE 1. Description of convolutional neural network.

Block	Layer(s) Description	Kernel(s) Number	Output Size Big CNN	Output Size Small CNN
Input			224x224x3	64x64x1
Convolution	k=3 s=1 p=1 ReLU BatchNorm	64	224x224x64	64x64x64
Convolution	k=3 s=1 p=1 ReLU BatchNorm MaxPool (k=2, s=2)	128	112x112x128	32x32x128
Convolution	[...]	256	56x56x256	16x16x256
Convolution	[...]	512	28x28x512	8x8x512
Convolution	[...]	512	14x14x512	4x4x512
Fully Connected	Linear, ReLU	1	100,352 to 1,024	8,192 to 1,024
Fully Connected	Linear	1	1,024 to 45	1,024 to 45

where k = kernel, s = striding and p = padding.



(a) SiC_0 (b) SiC_1 (c) SiC_2

(d) SiC_3 (e) SiC_4 (f) SiC_5

FIGURE 12. A subset of silicon carbide defect map patterns.

A. UNSUPERVISED LEARNING APPROACH

In this sub-section the details about the training procedures, dataset and experimental results related to unsupervised learning approach, are reported.

1) UNSUPERVISED LEARNING BLOCK: SILICON CARBIDE DATASET

STMicroelectronics Silicon Carbide is an internal dataset recently created by failure engineers of STMicroelectronics related to new production line issues in Silicon Carbide (SiC) devices. The dataset contains 2, 238 WDMs related to Silicon Carbide devices and stored as RGB images at 61×61 resolution, grouped in such imbalanced 54 new patterns and named by a progressive number.

The Figs. 12, 13, 14 report such instances of the SiC defect maps. Specifically: Fig. 12b reported a ring-like pattern with multiple contiguous good dies at the center; Fig. 12f reported an half-moon-like pattern with multiple contiguous good dies at the center on the edge of the wafer; Fig. 13a reported a checker-board like pattern with multiple contiguous defective dies at the center; Fig. 13b reported a wafer full of defective dies with several straight lines of good dies; Fig. 13e reported an half-moon-like pattern but with longer and contiguous defective dies along the edge; Fig. 13g reported a ring-like pattern but with a scratch on the upper side of the wafer; Fig. 13h reported defective dies at the bottom center part, good dies in straight horizontal lines; Fig. 13s reported defective dies at the center and straight vertical lines of good dies; Fig. 13t reported an instance of split ring-like pattern on the right side of the Wafer; Fig. 13u reported a straight horizontal lines of good dies and defective dies arranged on the left side; Fig. 14b reported such defective dies at the center with straight vertical lines and spots of good dies; Fig. 14d reported an amplified version of ring-like pattern; Fig. 14f reported an amplified and inner half-moon-like pattern on the left side of the wafer; Fig. 14g reported an amplified version of ring-like pattern with good dies arranged vertically at the center of the wafer similar to SiC_6 in Fig. 13a; Fig. 14v reported a wafer full of good dies with a horizontal centered line and spots arranged like a checkerboard of good dies; Fig. 14w reported a right half side of the wafer with defective dies and a straight horizontal line of good dies.

From this internal dataset, a subset of 225 unlabelled mixed WDMs is randomly chosen and arranged in a 3D Surface Plot as reported in Fig. 15. This plot allows to spot predominant patterns by visual inspection, due to the binarization of WDMs where good dies have value "0" and defective dies have value "1". By stacking up binarized WDMs along pixel coordinates (x and y axes) the sum of "1" values (z axis) will represent the spatial distribution of defective dies on the wafer. A high value of z at a specific coordinate point will represent a predominant defective pattern.

This dataset has been used to evaluate the unsupervised block of the proposed pipeline. The defect maps embedded in this dataset have been previously analyzed by engineers of STMicroelectronics, in this way, we were able to better and more accurately evaluate the outcomes of the unsupervised analysis.

2) UNSUPERVISED LEARNING BLOCK: TRAINING PROCEDURE

As introduced in the unsupervised learning block description, dimensionality reduction and hierarchical clustering have been applied. The adopted configuration of parameter-values

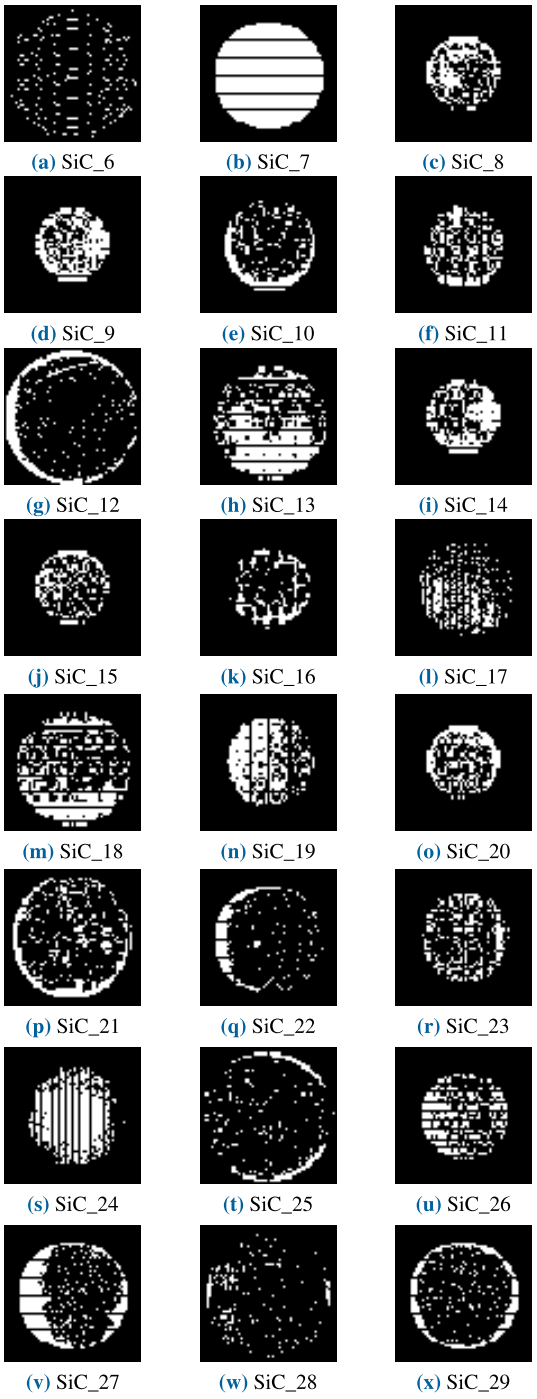


FIGURE 13. A subset of silicon carbide defect map patterns (part 2).

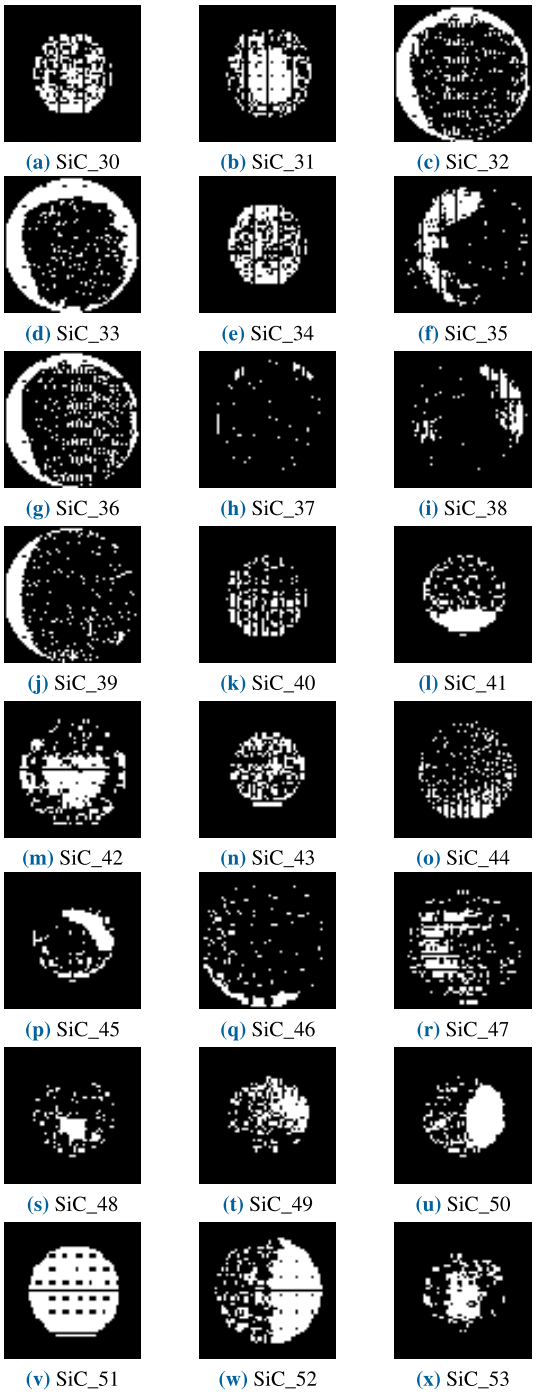


FIGURE 14. A subset of silicon carbide defect map patterns (part 3).

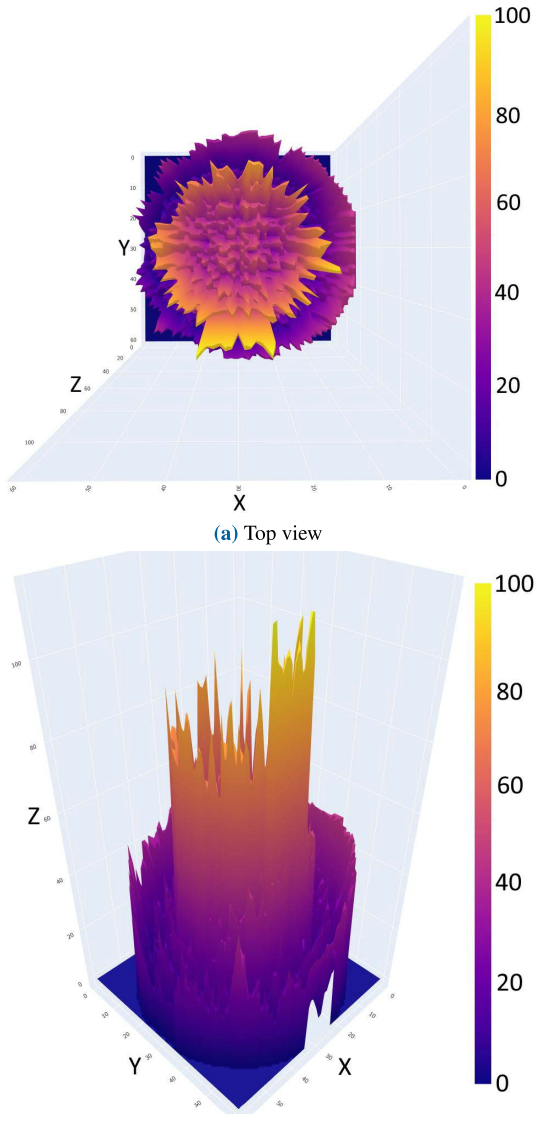
are reported in Tables 2 and 3 for UMAP and HDBSCAN respectively.

More in detail, referred to Table 2 related to UMAP:

- ''Number of components'' is the number of dimensions of the embedded space;
- ''Number of neighbors'' is the number of neighboring data-points suitable to preserve local density;
- ''Minimum distance'' is the minimum distance between embedded data-points;
- ''Distance metric'' is the metric used to compute distances in high dimensional space.

While, referred to Table 3 related to HDBSCAN:

- ''Minimum cluster'' is a parameter suitable to determine the minimum number of clusters;
- ''Minimum number of samples'' is the minimum number of data-points in a point's neighborhood to be considered as core point;
- ''Cluster selection (ϵ)'' is the threshold applied to separate or merge clusters;



(a) Top view

(b) Side view

FIGURE 15. Silicon carbide stacked wafer bin maps plot.

- "Distance metric" is the metric used to compute distance inside HDBSCAN processing;

In both UMAP and HDBSCAN, the parameter-value "Distance metric" has been evaluated by using Euclidean (Eq. 5) or Manhattan (Eq. 6) distances.

$$d_{Euclidean} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

$$d_{Manhattan} = \sum_{i=1}^{n} |x_i - y_i| \tag{6}$$

where n is the number of dimensions.

The Distance metric applied affects how the embeddings in UMAP and clusters in HDBSCAN are generated. Euclidean distance computes the shortest distance between two data-points x_i and y_i while Manhattan distance computes the absolute distance between two data-points x_i and y_i in a grid-like environment. According to [41] Manhattan distance should be used in high-dimensional space scenario as it is more robust to outliers but at the same time it is affected by the curse of dimensionality drawbacks. While, Euclidean distance affects embeddings generation by squaring the distance of far-way data-points x_i and y_i. In practical application, the authors opted for a metric rather than another due to the statistical distribution of the data.

TABLE 2. UMAP parameters.

Parameter	Value
Number of components	2 to 20
Number of neighbors	5, 10, 15
Minimum distance	0.0, 0.5, 1
Distance metric	Euclidean or Manhattan

TABLE 3. HDBSCAN parameters.

Parameter	Value
Minimum cluster size	5, 10, 15
Minimum number of samples	1, 2, 3, 5, 10
Cluster selection (ϵ)	0.0, 0.5, 1
Distance metric	Euclidean or Manhattan

The combination of the indicated parameters in Tables 2 and 3 enabled the evaluation of 30, 780 models for clusters generation starting from input WDMs. For this reason, we have used ad-hoc performance indexes to evaluate the performance of each model. Specifically, we have adopted the Silhouette Score, Calinski-Harabasz Index and Davies-Bouldin Index defined as follow.

Silhouette Score (SS) [42], [43] provides a generic characterization of the cluster. The score is computed over each cluster as in Eq. 7

$$SS = \frac{\bar{b} - \bar{a}}{max(\bar{a}, \bar{b})} \tag{7}$$

where:

- $\bar{a}$ is the average distance between samples in the same class;
- $\bar{b}$ is the average distance between samples in the nearest clusters.

SS ranges from -1 to $+1$ where -1 represents incorrect clustering while $+1$ represents highly dense clustering. SS around zero indicates overlapping clusters.

Calinski-Harabasz Index (CHI) [43], [44] is the ratio of the between-clusters dispersion and the within-cluster dispersion. The index is computed as in Eq. 8:

$$CHI = \frac{tr(B_k)}{tr(W_k)} \times \frac{s_E - k}{k - 1} \tag{8}$$

where:

$$W_k = \sum_{q=1}^{k} \sum_{x \in R_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (r_q - r_E)(r_q - r_E)^T \tag{9}$$

- $tr(B_k)$ is the trace of the between cluster dispersion matrix;
- $tr(W_k)$ is the trace of the within-cluster dispersion matrix;
- E is the dataset of clusters;
- s_E is the size of the dataset;
- k is the number of clusters;
- R_q is the set of samples in cluster q;
- r_q is the center of cluster q;
- R_E is the center of E;
- n_q is the number of samples in cluster q.

A higher value of the CHI means that clusters are dense and well separated.

Davies-Bouldin Index (DBI) [43], [45] measures the average 'similarity' between clusters, i.e., it measures the distance between clusters with the size of the clusters themselves. The index is computed over each cluster as reported in Eq. 10

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\frac{c_i + c_j}{d_{ij}} \tag{10}$$

where:

- d_{ij} is the distance between cluster centroids i and j;
- c_i or c_j are the average distances between each sample in the cluster i or$j$ and the centroid's cluster.

A value of DBI close to zero indicates a better cluster partition.

Now, the experimental results of the unsupervised learning block are reported.

3) UNSUPERVISED LEARNING BLOCK: EXPERIMENTAL RESULTS

Table 4 summarizes the best model according to the three metrics involved. Specifically, according to:

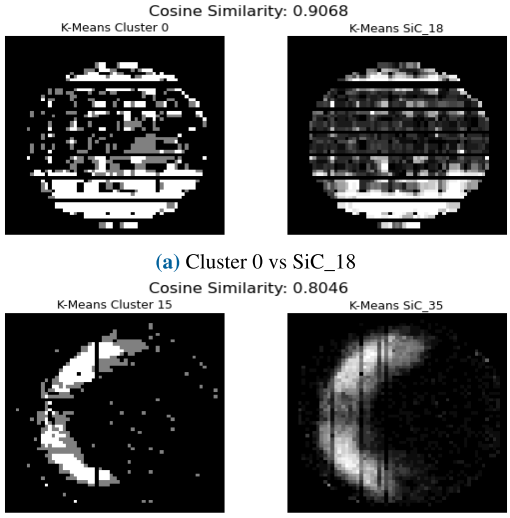
Silhouette score: The best model is the 14,620th where only 8 clusters are found and the fifth cluster is wrongly grouped. Furthermore 4 WDMs were mistakenly excluded from clustering.

Calinski-Harabasz Index: The best model is the 24,625th where only 9 clusters are found and the ninth cluster is wrongly grouped. Furthermore 7 WDMs were mistakenly excluded from clustering.

Davies-Bouldin Index: The best model is the 1,643th where only 2 clusters are found and they are wrongly grouped. Furthermore 4 WDMs were mistakenly excluded from clustering.

The summary of the performed tests and outcomes are reported in Table 4.

Taking into account this preliminary and unsatisfactory results, the authors tried to further optimize the UMAP and HDBSCAN parameters selection (Tables 2 and 3). More in detail, after careful tests the authors discovered that by only using Euclidean distance in both UMAP and HDBSCAN and a combination of: a lower number of components (i.e., 4), a lower number of neighbors (i.e., 2), a lower minimum cluster size (i.e., 2) and a minimum number of samples



(a) Cluster 0 vs SiC_18

(b) Cluster 15 vs SiC_35

FIGURE 16. Cosine similarity between clusters.

(i.e., 1), allowed to preserve local density by obtaining a better clustering performance despite the increasing number of clusters. This allowed to find 41 clusters with a range of cluster sample size from 2 to 8, with some bigger cluster containing 15 or 25 WDMs. After the clustering was performed as in the previous paragraph, we proceeded by computing K-means centroids. Moreover, we have compared the related centroids with the well-classified SiC dataset of STMicroelectronics by applying a threshold of 90% with a Cosine Similarity validation. Some of the collected outcomes have been reported in Figs. 16, 17.

Although K-Means is a classical approach which often shows limits in the determination of the centroids of clusters, in the application herein proposed we noticed that the combination with Cosine Similarity allowed a robust match between unlabelled clusters. As introduced, the SiC dataset have been previously annotated by engineers of STMicroelectronics so that we have checked the similarity between the clusters identified by the unsupervised pipeline with the classes already identified. In our experiments only 7 clusters have a lower value of Cosine Similarity between 70 and 80% and only 2 clusters have been misclassified with classes from SiC dataset (further inspection revealed a bad clustering and a wrong centroid generation). Such instances are reported in Figs. 16, 17.

B. SUPERVISED LEARNING APPROACH

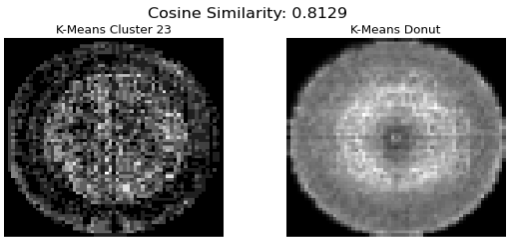
In this section is reported details about training procedures, dataset and experimental results related to supervised learning approach.

1) SUPERVISED LEARNING BLOCK: SILICON DATASET

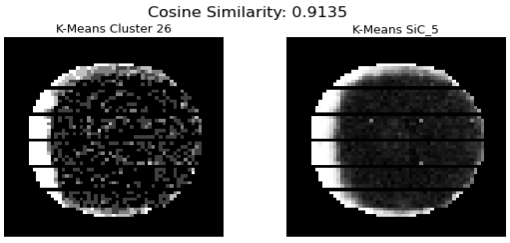
For training and validation of the supervised block, three datasets have been used: two public dataset WM-811K [4] and MixedWM38 [46] as well as an internal dataset (based on

TABLE 4. Unsupervised clustering benchmark performance.

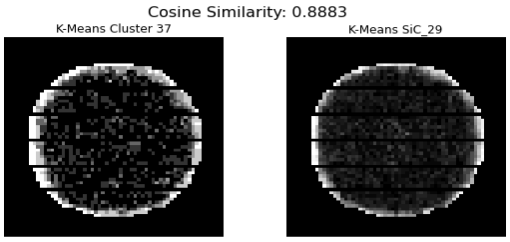
Model	UMAP				HDBSCAN				Silhouette Score	Davies Bouldin Index	Calinski Harabasz Index
	Number of components	Number of neighbors	Minimum distance	Distance	Minimum sample size	Minimum cluster size	Cluster selection Epsilon	Distance			
14,620	11	5	0.0	Euclidean	3	5	1.0	Euclidean	0.7704	0.2436	1704.05
24,625	17	5	0.5	Manhattan	5	5	0.0	Manhattan	0.6768	0.1998	539.64
1,643	3	5	0.0	Euclidean	2	5	1.0	Manhattan	0.2784	0.2784	2160.27



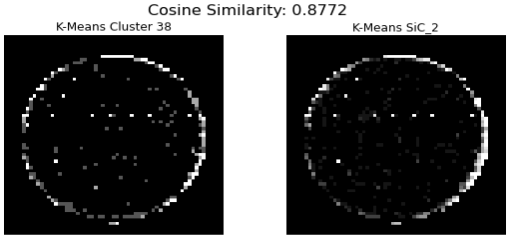
(a) Cluster 23 vs Donut



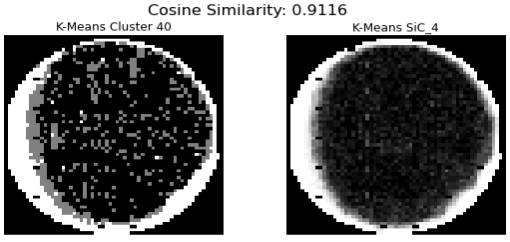
(b) Cluster 26 vs SiC_5



(c) Cluster 37 vs SiC_29



(d) Cluster 38 vs SiC_2



(e) Cluster 40 vs SiC_4

FIGURE 17. Cosine similarity between clusters (cont.).

Silicon technology) collected by STMicroelectronics production groups. In the following we describe the datasets used.

WM-811K [4] is a dataset created by TSMC (Taiwan Semiconductor Manufacturing Company). The source

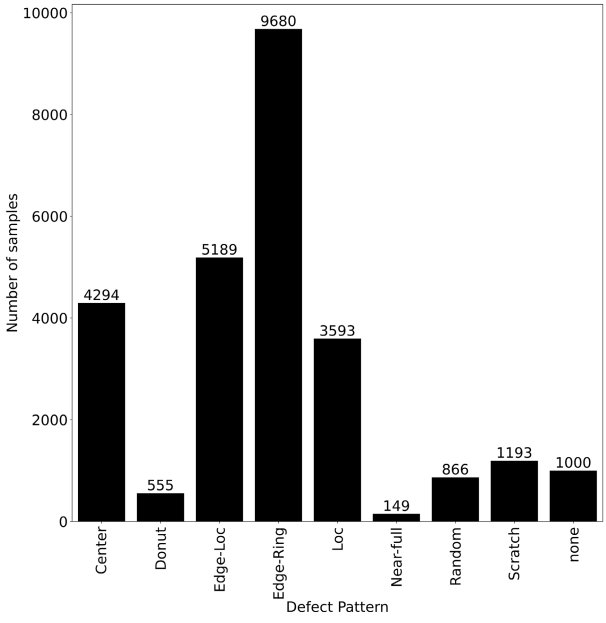
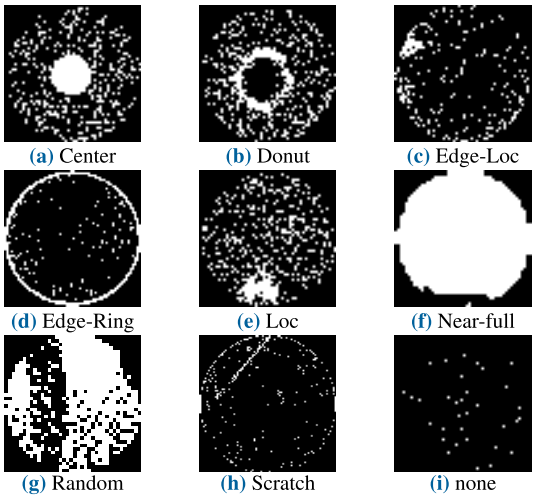


FIGURE 18. WM-811K WDMs distribution.



(a) Center (b) Donut (c) Edge-Loc
(d) Edge-Ring (e) Loc (f) Near-full
(g) Random (h) Scratch (i) none

FIGURE 19. Binarized WDMs of WM-811K dataset.

dataset is composed by 811, 457 samples, but only 172, 948 are properly labelled wafer maps, as RGB images at different resolution (from minimum resolution of 15 × 3 to 212 × 84). The dataset contains a total of 9 different wafer defect patterns with one classified as ''none''. The dataset is imbalanced as the ''none'' class contains more samples with normal production yield (about 150, 000 samples) with respect the other ones. The following Fig. 18 shows the defect pattern distribution in this subset for a total of 26, 519 samples.

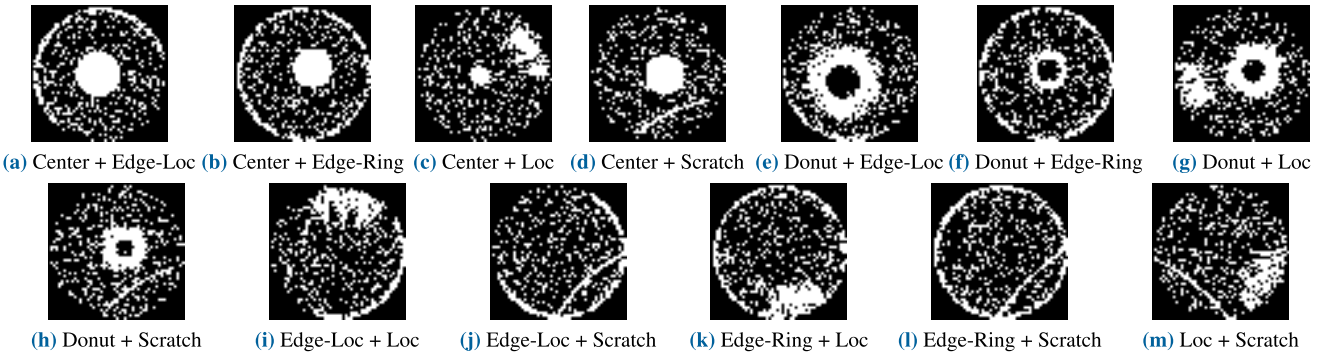


FIGURE 20. 2 mixed type WDMs of mixedWM38 dataset.

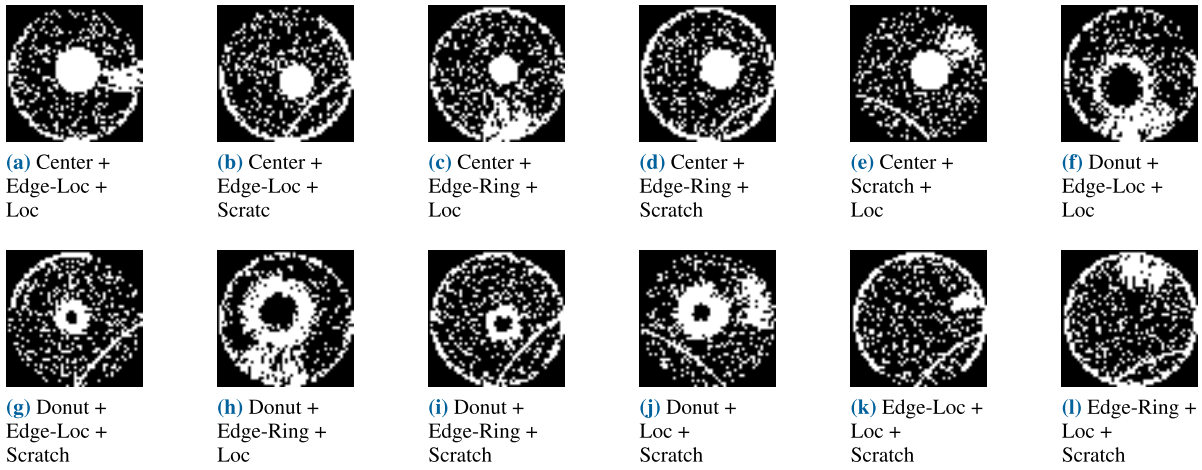


FIGURE 21. 3 mixed type WDMs of mixedWM38 dataset.

As reported in Fig. 18 the dataset includes 9 different patterns (including "none" with different morphology). As shown in Fig. 19, where: 1) defective dies arranged at the center; 2) defective dies arranged as a Donut-like shape; 3) a group of defective dies located on the edge; 4) defective dies along the edge; 5) a group of defective dies located anywhere; 6) a wafer full of defective dies; 7) random defective dies; 8) single circular scratch; 9) none (normal) Wafer Maps with few defective dies.

To cover the mentioned imbalance issue, we significantly reduced the "none" class only to 1,000 random samples.

MixedWM38 [46] dataset contains 38, 015 well-classified wafer maps, including the so called "normal" pattern (the "none" class in the WM-811K dataset) as well as 8 single native defect patterns and 29 defect patterns grouped in 2 mixed type (Fig. 20), 3 mixed type (Fig. 21) and four mixed type (Fig. 22), for a total amount of 38 defect patterns at 52×52 fixed resolution.

STMicroelectronics Silicon Dataset contains WDMs of Silicon devices classified in 7 different patterns named by a progressive number as shown in Fig. 23, where: (a) defective dies at the bottom of the wafer with two straight lines of good dies; (b) defective dies arranged like checker-board; (c) double straight scratches; (d) full wafer of defective dies with straight horizontal lines of good dies; (e) defective dies grouped and arranged as circles along the edge; (f) defective dies located at the bottom; (g) multiple circular scratches. That dataset is imbalanced and composed by 6, 732 samples at 61×61 fixed resolution.

As introduced, the final used dataset is a combination of the previous ones containing a total of 71, 266 WDMs with 45 different classes re-arranged and grouped in a more balanced way. This full dataset has been split into training, validation and test sets according to a 80-10-10 hold-out methodology.

2) SUPERVISED LEARNING BLOCK: TRAINING PROCEDURE

The designed supervised deep system has been trained by using single datasets as well as combined ones. Preliminary, ad-hoc data augmentation method has been employed including random rotation, horizontal and vertical flip.

As introduced, we have implemented two deep network backbones (Big CNN and Small CNN). Both models have been trained for 100 epochs in PyTorch framework vers. 1.10 [47] with CUDA 11.4 running on a workstation based on Intel Core i9-12900K with 64GB DDR4-3600MHz of RAM coupled with NVIDIA RTX 3060 with 12GB of VRAM. We have tested for benchmark comparison both pre-trained (on ImageNet) State-Of-The-Art (SOTA) backbones as well as the same trained from scratch [48]. The Adam algorithm [49] has been used as optimizer with an initial learning rate of $1e-4$ and, Cross Entropy function has been used as

(a) Center + Loc + Edge-Loc + Scratch (b) Center + Loc + Edge-Ring + Scratch (c) Donut + Loc + Edge-Loc + Scratch (d) Donut + Loc + Edge-Ring + Scratch

FIGURE 22. 4 mixed type WDMs of mixedWM38 dataset.

(a) Si_11 (b) Si_23 (c) Si_27 (d) Si_38 (e) Si_40 (f) Si_44 (g) Si_70

FIGURE 23. Binarized silicon WDMs provided by STMicroelectronics.

TABLE 5. Benchmark models training configuration.

Parameter	Pre-trained SOTA	SOTA trained from scratch	CNN and ViT at 64x64
Input Wafer Defect Map Image Size	224x224		64x64
Color Space	RGB		Grayscale
Normalization	mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]		None
Data augmentation	Random: rotation, horizontal and vertical flip		
Batch size	128	16	128
Loss Type	Cross Entropy Loss (with class weighting)		
Optimizer	Adam (with a learning rate of 1e-4)		

performance Loss function with class weighting (Eq. 11)

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^T$$
$$l_n = -\sum_{c=1}^{C} w_c log \frac{exp(x_{n,c})}{\sum_{i=1}^{C} exp(x_{n,i})} y_{n,c} \qquad (11)$$

where x is the input tensor, y is the target class label, w is the weight (rescaled by weight given to each class), C is the number of classes and N spans the minibatch.

In Table 5 the designed deep networks training configuration is reported.

3) SUPERVISED LEARNING BLOCK: BASELINE BENCHMARK

Before evaluating the combination of the three previously described datasets, we have validated our proposed solutions against architectures proposed by authors of public datasets, specifically WM-811K [4] and MixedWM38 [46].

More in detail, authors of WM-811K dataset produced a confusion matrix ([4], Fig. 13a) of the performance obtained from their Dual-stage WMFPR method applied to the WM-811K dataset. In [4] the mentioned confusion matrix was reported while we reported in Table 6 the related confusion matrix converted to overall accuracy, precision, recall, F1-score indexes for benchmark comparison between the proposed Dual-stage WMFPR [4] and our proposed solutions.

From Table 6 the overall accuracy of the Dual-stage WMFPR seems slightly higher than ours. The related reason is connected to the over-sampled class "none" which actually allows the method proposed in [4] (which embeds about 110, 000 wafer samples of "none") to apparently outperform our method. In fact, from the details of the performances for single classes reported in Table 6, it is evident how our architectures significantly outperform the method proposed in [4] which it recovers with the only "none" class which is strongly imbalanced and in any case not very significant for the analyzed WDMs assessment. Therefore, the robustness of the proposed method is evident in relation to the defect classes that are most valid in the analysis of the defect patterns. More details about Dual-stage WMFPR and our proposed architectures can be found in Table 6.

About the MixedWM38 dataset the authors of [46] described a split of their dataset (training and validation set) as 80% and 20% providing a performance assessment of their method based on the usage of precision and recall metrics. In Table 7 we have reported the benchmarks comparison between the method reported in [46] named DC-Net against our proposed ones. As showed in Table 7 our proposed solutions (in both the designed configurations) outperformed the DC-Net approach designed in [46] by an average of 4% in overall accuracy. The performances related to the classification of single defect-classes (both native and mixed) showed that our method outperforms the DC-Net approach [46] on average, confirming the effectiveness of the proposed approach. More details about DC-Net and our proposed architectures can be found in Table 7.

4) SUPERVISED LEARNING BLOCK: FULL DATASET BENCHMARK

As applied for single datasets, the proposed deep network has been validated with the full dataset embedding the previous mentioned ones and split as follow: 57, 012 wafer defect samples as Training Set, 7, 127 samples for Validation Set and 7, 127 samples for Testing Set. As benchmark indexes

TABLE 6. Dual-stage WMFPR [4] versus proposed deep networks-WM-811K dataset.

	Dual-stage WMFPR				Proposed CNN at 64x64			Proposed CNN at 224x224			
Failure Type	**Precision**	**Recall**	**F1-Score**	**Support**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Support**
Center	0.5928	0.8486	0.6980	832	**0.9813**	**0.9545**	**0.9677**	**0.9750**	**0.9750**	**0.9750**	440
Donut	0.5902	0.7397	0.6565	146	**0.7564**	**0.9672**	**0.8489**	**0.8871**	**0.9016**	**0.8943**	61
Edge-Loc	0.4761	0.8506	0.6105	2,772	**0.9312**	**0.9189**	**0.9250**	**0.9123**	**0.9226**	**0.9174**	530
Edge-Ring	0.9502	0.7966	0.8667	1,126	**0.9853**	**0.9915**	**0.9884**	**0.9791**	**0.9936**	**0.9863**	944
Loc	0.4201	0.6847	0.5207	1,973	**0.9145**	**0.8470**	**0.8794**	**0.9012**	**0.8470**	**0.8732**	366
Near-full	0.8455	0.9789	0.9073	95	**0.9375**	**1.0000**	**0.9677**	**1.0000**	0.6667	0.8000	15
Random	0.7045	0.7977	0.7482	257	**0.9512**	**0.9398**	**0.9455**	**0.9487**	**0.8916**	**0.9193**	83
Scratch	0.3943	0.8240	0.5334	693	**0.8182**	**0.8926**	**0.8538**	**0.8833**	**0.8760**	**0.8797**	121
none	**0.9970**	**0.9570**	**0.9765**	110,701	0.8077	0.9130	0.8571	0.8269	0.9348	0.8776	92
accuracy	**0.9463**	**0.9463**	**0.9463**	-	0.9416	0.9416	0.9416	0.9416	0.9416	0.9416	-
macro avg	-	-	-	118,595	0.8981	0.9361	0.9148	0.9237	0.8899	0.9025	2,652
weighted avg	-	-	-	118,595	0.9436	0.9416	0.9419	0.9417	0.9416	0.9412	2,652

TABLE 7. DC-Net [46] versus proposed deep networks-mixedWM38 dataset.

	DC-Net			Proposed CNN at 64x64			Proposed CNN at 224x224			
Failure Type	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Support**
Center	0.9300	0.9700	0.9496	**0.9778**	**1.0000**	**0.9888**	**0.9778**	**1.0000**	**0.9888**	88
Center+Edge-Loc	0.9400	0.9400	0.9400	**1.0000**	**0.9712**	**0.9854**	**1.0000**	**0.9712**	**0.9854**	104
Center+Edge-Loc+Loc	**0.9900**	**0.9600**	0.9748	0.9895	0.9400	0.9641	0.9898	**0.9700**	**0.9798**	100
Center+Edge-Loc+Scratch	0.9200	1.0000	0.9583	**0.9952**	**0.9764**	**0.9857**	**0.9951**	**0.9623**	**0.9784**	212
Center+Edge-Ring	0.9200	0.9900	0.9537	**0.9720**	**1.0000**	**0.9858**	**0.9720**	**1.0000**	**0.9858**	104
Center+Edge-Ring+Loc	0.9300	0.9100	0.9199	**0.9709**	**0.9709**	**0.9709**	**0.9626**	**1.0000**	**0.9810**	103
Center+Edge-Ring+Scratch	0.9700	0.9700	0.9700	**0.9882**	**1.0000**	**0.9941**	**0.9545**	**1.0000**	**0.9767**	84
Center+Loc	0.9200	0.9600	0.9396	**0.9813**	**0.9906**	**0.9859**	**0.9813**	**0.9906**	**0.9859**	106
Center+Loc+Edge-Loc+S	**0.9600**	**0.9900**	**0.9748**	0.9278	0.9574	0.9424	**0.9889**	**0.9468**	**0.9674**	94
Center+Loc+Edge-Ring+S	**0.9900**	**0.9600**	**0.9748**	0.9612	0.9706	0.9659	0.9706	0.9706	0.9706	102
Center+Loc+Scratch	0.9700	0.9300	0.9496	**0.9890**	**0.9783**	**0.9836**	**1.0000**	**0.9783**	**0.9890**	92
Center+Scratch	0.9700	0.8900	0.9283	**0.9794**	**1.0000**	**0.9896**	**0.9596**	**1.0000**	**0.9794**	95
Donut	0.9500	0.9300	0.9399	**0.9905**	**0.9905**	**0.9905**	**1.0000**	**1.0000**	**1.0000**	105
Donut+Edge-Loc	0.9600	0.9200	0.9396	**0.9714**	**0.9714**	**0.9714**	**0.9903**	**0.9714**	**0.9808**	105
Donut+Edge-Loc+Loc	0.9500	0.9100	0.9296	**1.0000**	**0.9892**	**0.9946**	**1.0000**	**1.0000**	**1.0000**	93
Donut+Edge-Loc+Scratch	0.9800	0.9700	0.9750	**0.9519**	**0.9340**	**0.9429**	**0.9528**	**0.9528**	**0.9528**	106
Donut+Edge-Ring	0.9100	0.9800	0.9437	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9898**	**0.9949**	98
Donut+Edge-Ring+Loc	0.8900	1.0000	0.9418	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	94
Donut+Edge-Ring+Scratch	0.9000	0.9400	0.9196	**0.9720**	**0.9905**	**0.9811**	**0.9722**	**1.0000**	**0.9859**	105
Donut+Loc	0.9400	0.9700	0.9548	**0.9783**	**1.0000**	**0.9890**	**1.0000**	**1.0000**	**1.0000**	90
Donut+Loc+Edge-Loc+S	0.9500	0.8900	0.9190	**0.9891**	**0.9579**	**0.9733**	**0.9890**	**0.9474**	**0.9677**	95
Donut+Loc+Edge-Ring+S	0.9200	0.9200	0.9200	**0.9817**	**1.0000**	**0.9907**	**0.9640**	**1.0000**	**0.9817**	107
Donut+Loc+Scratch	**0.9900**	0.8800	0.9318	0.9894	**0.9588**	**0.9738**	0.9789	**0.9588**	**0.9688**	97
Donut+Scratch	**0.9600**	0.9400	0.9499	**0.9652**	**0.9911**	**0.9780**	0.9561	**0.9732**	**0.9646**	112
Edge-Loc	0.9600	0.9100	0.9343	**0.9902**	**1.0000**	**0.9951**	**0.9899**	**0.9703**	**0.9800**	101
Edge-Loc+Loc	0.9800	0.8900	0.9328	**1.0000**	**0.9898**	**0.9949**	**1.0000**	**0.9388**	**0.9684**	98
Edge-Loc+Loc+Scratch	0.9700	0.9300	0.9496	**0.9899**	**1,0000**	**0.9949**	**0.9500**	**0.9694**	**0.9596**	98
Edge-Loc+Scratch	0.9400	0.9100	0.9248	**1.0000**	**0.9900**	**0.9950**	**0.9900**	**0.9900**	**0.9900**	100
Edge-Ring	0.9300	0.9700	0.9496	**0.9914**	**1.0000**	**0.9957**	**0.9661**	**0.9913**	**0.9785**	115
Edge-Ring+Loc	0.9500	0.9100	0.9296	**0.9717**	**0.9810**	**0.9763**	**0.9714**	**0.9714**	**0.9714**	105
Edge-Ring+Loc+Scratch	0.9800	0.9400	0.9596	**0.9889**	**0.9468**	**0.9674**	**0.9677**	**0.9574**	**0.9626**	94
Edge-Ring+Scratch	0.9600	0.9200	0.9396	**0.9811**	**1.0000**	**0.9905**	**0.9720**	**1.0000**	**0.9858**	104
Loc	0.9900	1.0000	0.9950	**0.9907**	**1.0000**	**0.9953**	**0.9904**	**0.9717**	**0.9810**	106
Loc+Scratch	0.9800	0.8800	0.9273	**0.9857**	**0.9718**	**0.9787**	**0.9459**	**0.9859**	**0.9655**	71
Near-full	0.9000	0.9400	0.9196	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9231**	**0.9600**	13
Random	0.9700	0.9300	0.9496	**1.0000**	**1.0000**	**1.0000**	**0.9897**	**1.0000**	**0.9948**	96
Scratch	0.6000	0.8800	0.7135	**0.9907**	**0.9907**	**0.9907**	**1.0000**	**1.0000**	**1.0000**	107
Normal (none)	0.9400	0.9100	0.9248	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	103
accuracy	0.9400	0.9500	0.9450	**0.9840**	**0.9840**	**0.9840**	**0.9811**	**0.9811**	**0.9811**	-
macro avg	-	-	-	**0.9843**	**0.9844**	**0.9843**	**0.9813**	**0.9803**	**0.9806**	3,802
weighted avg	-	-	-	**0.9841**	**0.9840**	**0.9839**	**0.9814**	**0.9811**	**0.9810**	3,802

we have used the accuracy in training, validation and testing phase. In Table 8 the collected performances for all the tested deep backbones and related configurations (both pre-trained on ImageNet and trained from scratch) are reported.

As expected, pre-trained models showed worse performance than trained from scratch architectures as the features related to defect maps are scarcely overlapped to those correlated to the ImageNet database and therefore the feature maps of a pre-trained network is more difficult to converge to feature maps associated with WDMs. A network that builds its own feature maps from scratch is able to learn better and therefore perform better.

We also tested architecture based on Vision Transformer [50] (ViT RGB at 224 × 224 × 3 and ViT at 64 × 64 × 1 spatial resolutions) which however underperformed compared to ours. We believe that this result is to be further

TABLE 8. Full dataset(s) benchmark comparison.

Model	Epoch	Training	Validation	Test	Runtime
resnet152_rgb_224_scratch	96	0.9727	0.9681	0.9672	1d 11h 17m
densenet161_rgb_224_scratch	95	0.9772	0.9680	0.9655	1d 16h 26m
cnn_l_64_scratch	**75**	**0.9761**	**0.9649**	**0.9630**	**2h 11m**
vgg19_rgb_224_scratch	76	0.9496	0.9560	0.9549	1d 2h 17m
cnn_rgb_224_scratch	**36**	**0.9541**	**0.9502**	**0.9518**	**1d 34m**
vit_rgb_224_scratch	83	0.9382	0.9430	0.9382	22h 1m
vit_l_64_scratch	100	0.9434	0.9473	0.9345	1h 29m
densenet_161_rgb_pretrained	81	0.7014	0.7110	0.7021	11h 7m
resnet152_rgb_pretrained	93	0.6616	0.6692	0.6709	9h 44m
vgg19_rgb_pretrained	71	0.4665	0.5203	0.5088	8h 40m

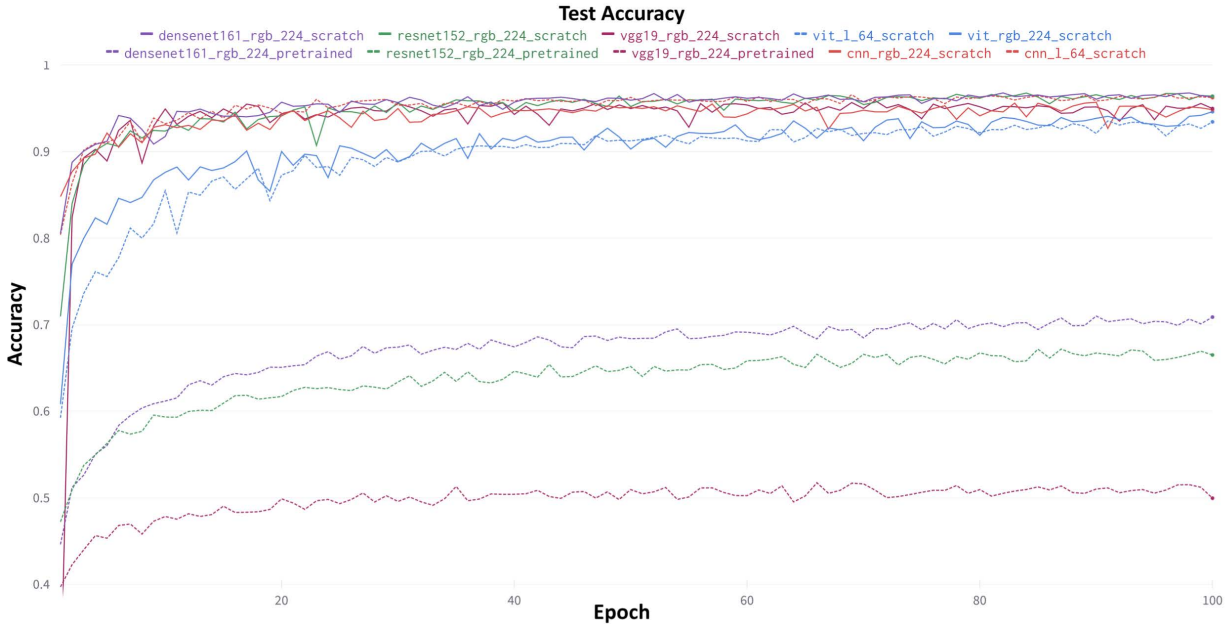


FIGURE 24. Test set accuracy curves.

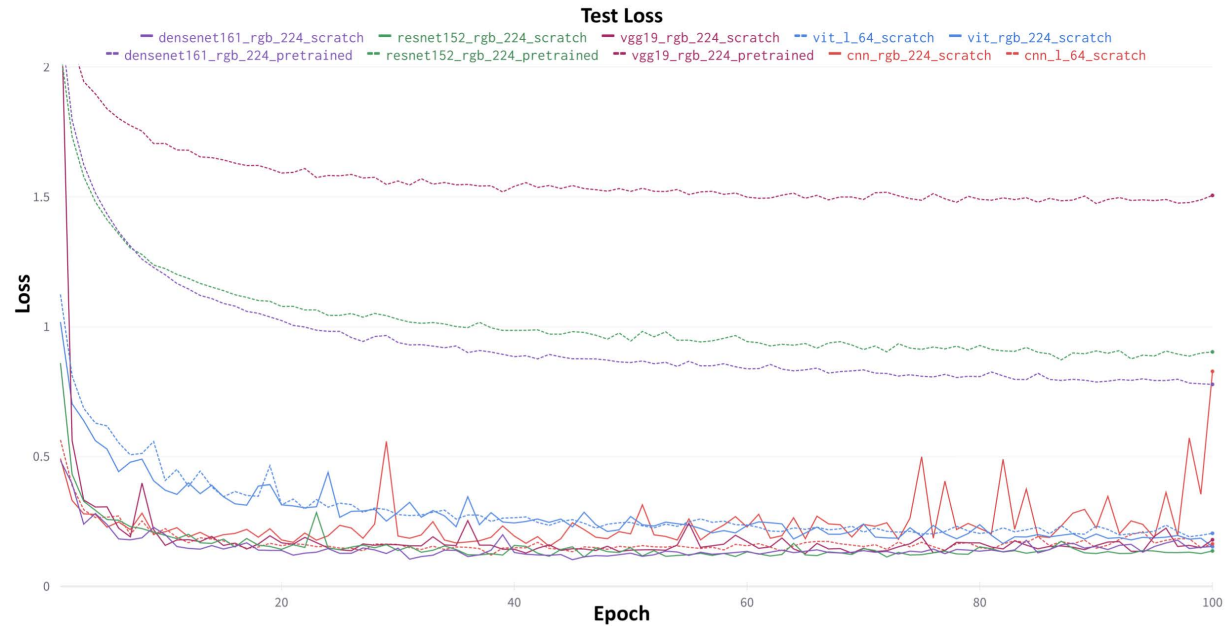


FIGURE 25. Test set loss curves.

FIGURE 26. Loc WDM from WM-811K.

validated in a much larger dataset than the current one taking into consideration that transformer-based architectures require a considerable amount of input data to build the attention-based internal representation of the input data. From Table 8 we noticed that the proposed solution based on CNN Grayscale at 64 × 64 performed better than the CNN RGB at 224 × 224 with a training accuracy of 97.61% as best validation model and 96.30% as test-set accuracy.

Fig. 24 and 25 reported benchmarks related to accuracy and the related loss curves in test set of all the tested architectures.

5) SUPERVISED LEARNING BLOCK: EXPLAINABILITY

Increasing the architecture's complexity by adding residual blocks (as ResNet and DenseNet architectures) and attention blocks (as in Vision Transformer architectures), we contribute to increase model's predictive power and robustness by giving to architectures the ability to generalize on new examples [51], [52]. A more complex architecture should be able to predict with a higher degree of confidence and to learn the main features that characterize the inputs faster. Although, model's performance is still evaluated using traditional metrics like accuracy, precision and recall we do not have any about what the model learned during training and what the model is going to predict when new examples are given. With these assumptions we need methods to explain what is really happening inside the model instead of considering it as acting like a black box.

The goal of Explainable Artificial Intelligence (XAI) is to explain the internal layer activations on the basis of which the deep model provides the desired solution. The methods of XAI most used in scientific literature are based on Integrated Gradients [53], Grad-CAM [54], [55] and Attention Maps [56] methods. More details about these mentioned approaches are now given.

Integrated Gradients [53] is a method to solve attribution of the prediction in deep network. It is based on two axioms: Sensitivity and Implementation Invariance.[3]

Formally, our deep network is represented by the function $F : R^n \rightarrow [0, 1]$, if we consider the straight-line path from the baseline a' to the input a, and compute gradients at all points along the path. Integrated gradients are obtained by cumulating those gradients. Specifically, integrated gradients are defined as the integral-path of the gradients along the straight-line path from the baseline a' to the input a. The integrated gradient along the i^{th} dimension for input a and baseline a' (with m, the number of steps in the Riemann approximation of the integral) is defined by the following Eq. 12:

$$IG(a) = (a_i - a_i') \times \sum_{k=1}^{m} \frac{\delta F(a' + \frac{k}{m} \times (a_i - a_i'))}{\delta a_i} \times \frac{1}{m} \quad (12)$$

Grad-CAM [54], [55] is a method to produce visual explanation of underlying Convolutional Neural Network models making them more explainable. Grad-CAM uses gradient information flowing into the last convolutional layer of the network to assign values to each neuron for a particular outcome. Given a localization map related to the class C, the Grad-CAM computes the gradient of the score of class C (before the Softmax) with respect to the feature map of the previous activated convolutional layer. This so computed gradient is global-average pooled over the width (i) and height (j) dimensions to obtain the neuron weighting.

Attention Maps As mentioned in [50], attention roll-out mechanism [56] applied in Computer Vision problems is defined as soft shading approach to focus learning on the region of interest of the input image. From a mathematical point of view, it is a recursive approach across all the weights and layers of the deep network and where for each layer, the corresponding attention map is multiplied by the previous ones as per Eq. 13:

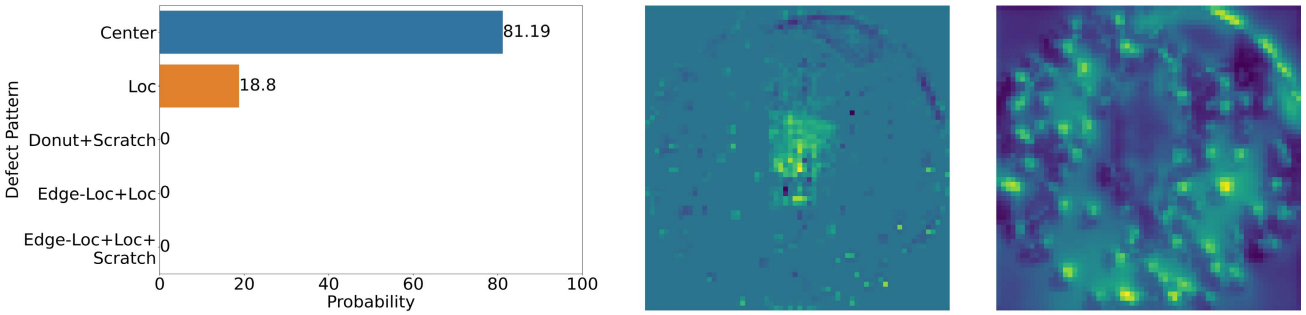
$$\bar{A}(l_i) = \begin{cases} A(l_i)\bar{A}(l_{i-1}) & \text{if i > j} \\ A(l_i) & \text{if i = j} \end{cases} \quad (13)$$

where $A(l_i)$ is the corresponding attention weight-map at layer i^{th} (for i to j, so from the first layer to the latest ones).

In order to show the behaviour of models, XAI methods aforementioned are now applied to an instance of "Loc" wafer defect pattern (Fig.26). As reported in Figs. 27-36 for each the tested deep backbones, we have computed explainability methods in order to reconstruct the internal representation used by the network for performing the related wafer patterns classification. The first aspect that is highlighted is related to the fact that although the defect pattern is single, the networks internally activate more similar classes such as Center, Edge-Loc and Loc (as highlighted in the Prediction plot showed in Fig. 27a, 28a). Anyway, the output of the network is represented by the most representative class of this internal map.

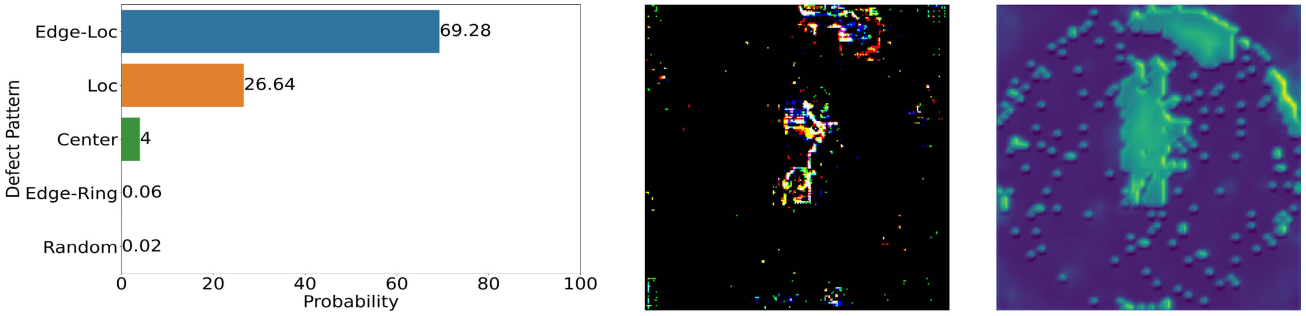
Starting from Fig. 27, our proposed deep network at 64 × 64 resolution (i.e., CNN at 64 × 64) predicted Center and Loc and they are quite visible in the Integrated Gradients but not in Grad-CAM. Instead, the proposed deep network at 224 × 224 (i.e., CNN at 224 × 224) (Fig. 28) predicted both Loc and Edge-Loc patterns with a higher confidence and both Integrated Gradients and Grad-CAM confirmed the corresponding patterns were activated. All the other models,

[3] A more detailed explanation can be found at [53].



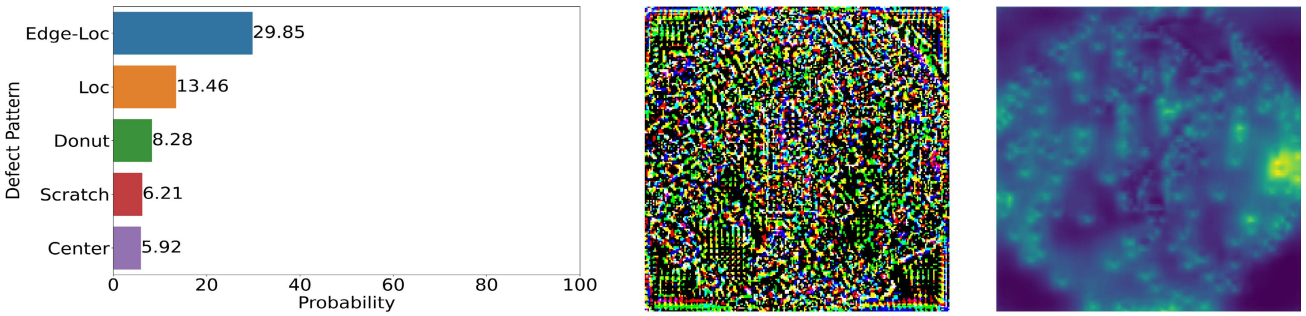
(a) Prediction (b) Integrated Gradients (c) Grad-CAM

FIGURE 27. Explainability analysis: CNN at 64 × 64 trained from scratch.



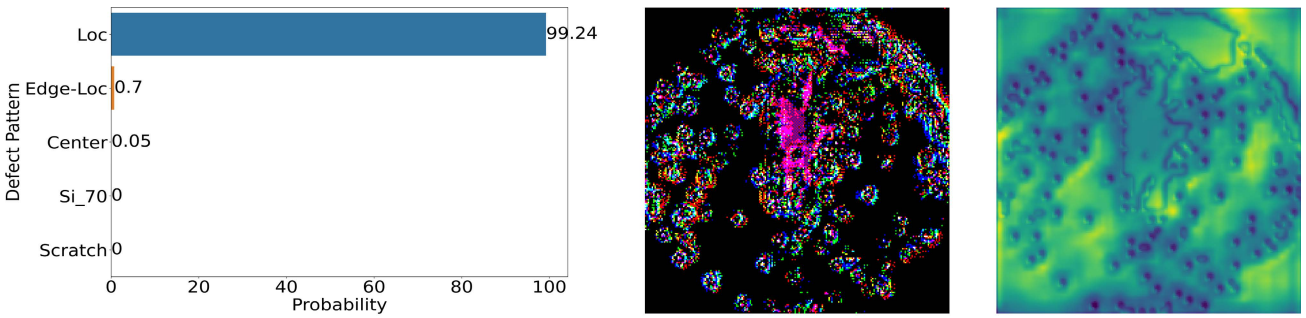
(a) Prediction (b) Integrated Gradients (c) Grad-CAM

FIGURE 28. Explainability analysis: CNN at 224 × 224 trained from scratch.



(a) Prediction (b) Integrated Gradients (c) Grad-CAM

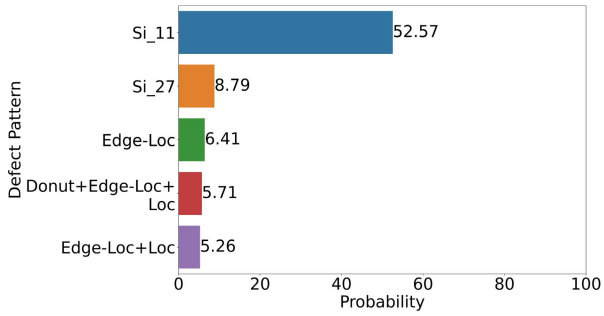
FIGURE 29. Explainability analysis: VGG-19 pre-trained.



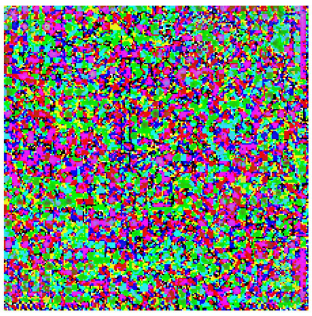
(a) Prediction (b) Integrated Gradients (c) Grad-CAM

FIGURE 30. Explainability analysis: VGG-19 trained from scratch.

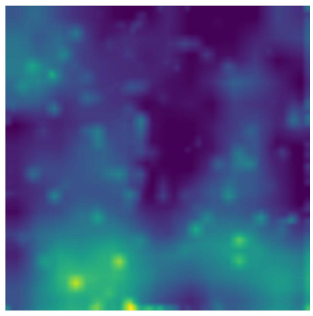
both pre-trained (Fig. 29, 31, 33) and trained from scratch (Fig. 30, 32, 34), showed the same behaviour, i.e., they were not able to make right predictions as confirmed by XAI based on Grad-CAM and integrated Gradients which not enabled any significant activation maps. It is interesting to highlight that the tested deep models trained from scratch were able to make a better prediction referred to high significant activation maps such as VGG19, ResNet-152 and DenseNet-161



(a) Prediction

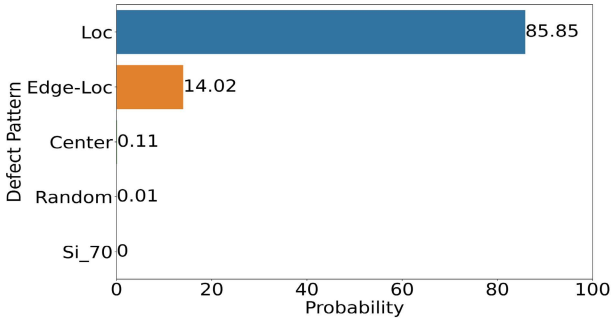


(b) Integrated Gradients



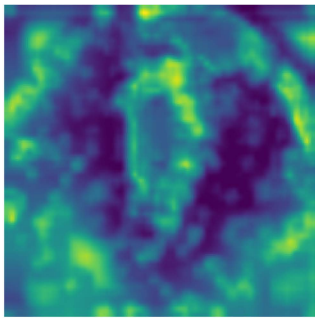
(c) Grad-CAM

FIGURE 31. Explainability analysis: ResNet-152 pre-trained.



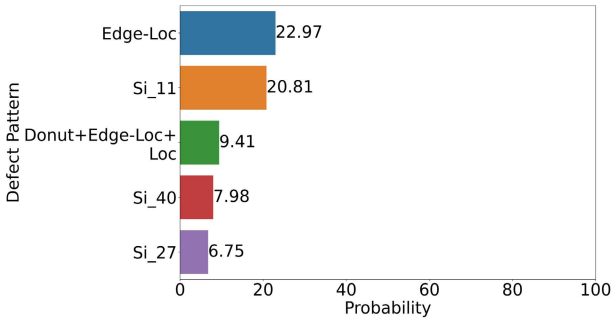
(a) Prediction

(b) Integrated Gradients

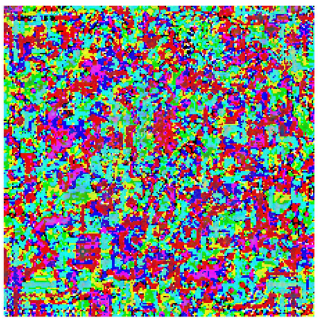


(c) Grad-CAM

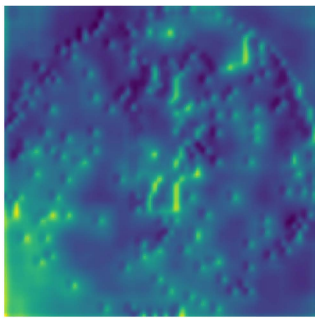
FIGURE 32. Explainability analysis: ResNet-152 trained from scratch.



(a) Prediction

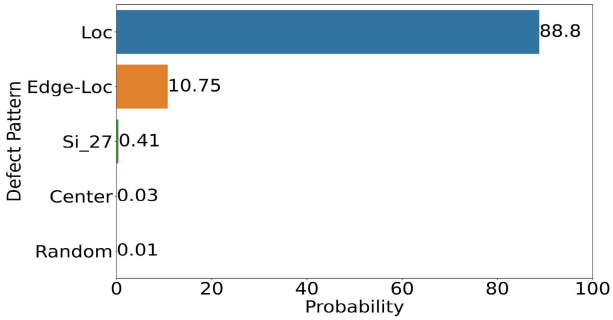


(b) Integrated Gradients

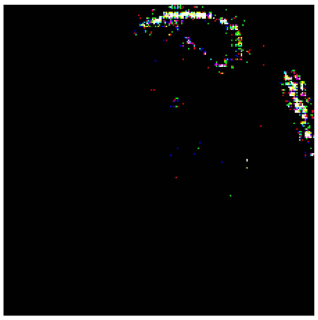


(c) Grad-CAM

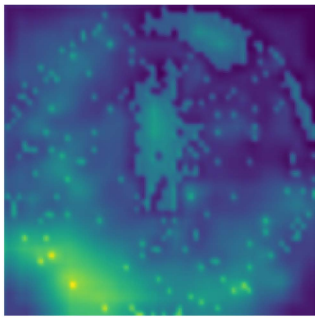
FIGURE 33. Explainability analysis: DenseNet-161 pre-trained.



(a) Prediction



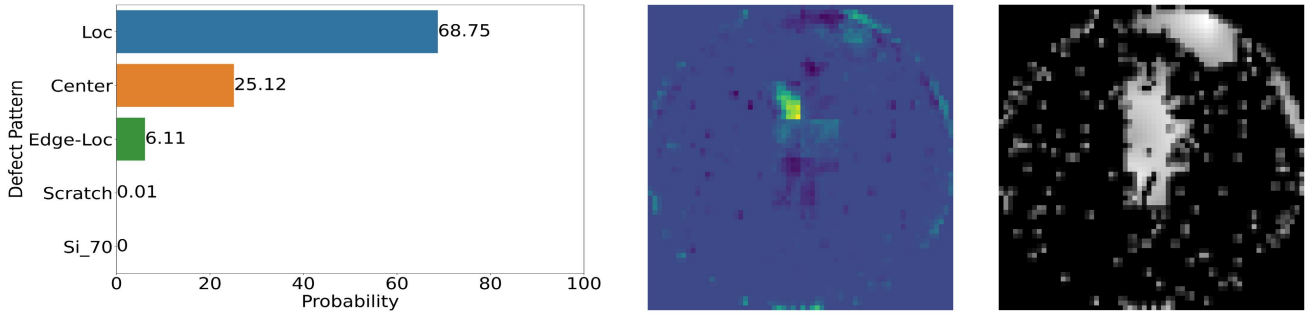
(b) Integrated Gradients



(c) Grad-CAM

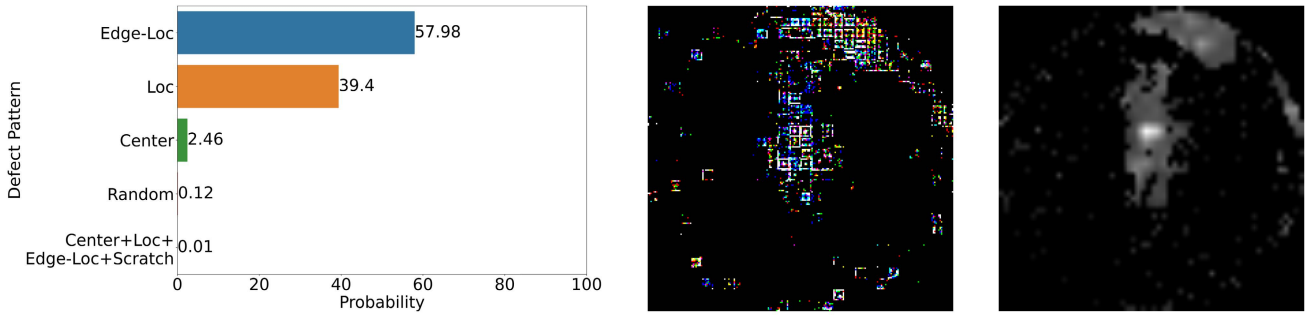
FIGURE 34. Explainability analysis: DenseNet-161 trained from scratch.

which were able to predict Loc and Edge-Loc patterns. Vision Transformers (ViT) both at 64 × 64 (Fig. 35) and 224 × 224 (Fig. 36) performed quite well as confirmed by the Activation Maps (Grad-CAM can not be applied to ViT architecture) and Integrated Gradients outcomes fairly consistent with the input defect pattern Loc in Fig. 26.



(a) Prediction (b) Integrated Gradients (c) Attention Map

FIGURE 35. Explainability analysis: ViT at 64 × 64 trained from scratch.



(a) Prediction (b) Integrated Gradients (c) Attention Map

FIGURE 36. Explainability analysis: ViT at 224 × 224 trained from scratch.

C. THE STAI-EWS TOOL

The target of this sub-section is to introduce the developed AI-boosted tool currently used in the STMicroelectronics laboratories. We have developed a user-friendly solution which covers both the implemented Unsupervised and Supervised sub-systems. Specifically, we have released a deep learning web-driven software application by using Python boosted by the open source framework Streamlit [57] in combination with Plotly [58] and PyTorch.

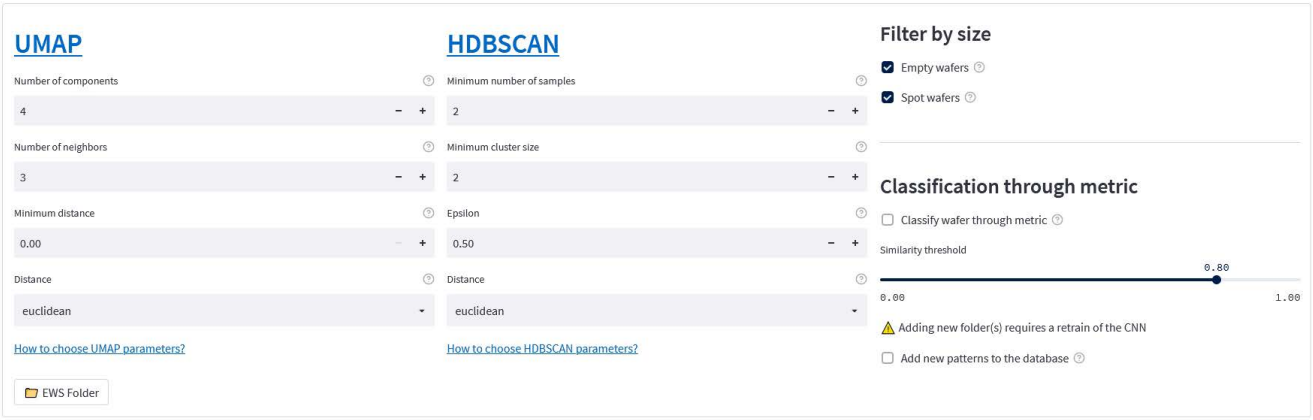
The released application was named "STAI-EWS" which means "STMicroelectronics Artificial Intelligence-based Electrical Wafer Sorting assessment" (a video demonstration can be found as supplementary material).

The STAI-EWS tool has been designed with an intuitive user interface. More in details, the tool is composed by the following parts: the sidebar and the main page.

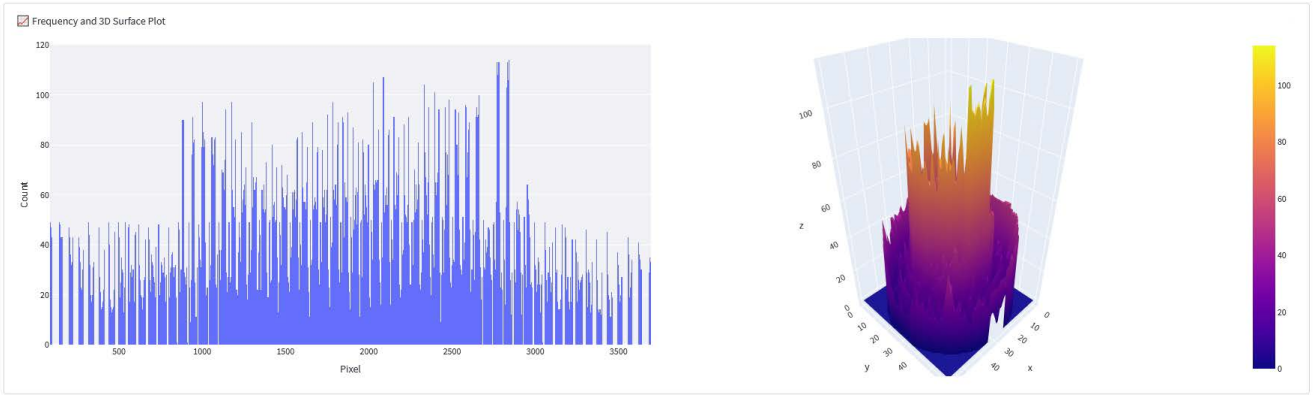
- **The sidebar** shows the current version of the STAI-EWS application and the navigation menu which allows the user to select: Supervised Wafer Defect Pattern Recognition, Unsupervised WDMs Clustering, Manage Database and Manage Convolutional Neural Network (CNN);
- **The main page** shows the current section configuration of the STAI-EWS tool.

As introduced, the whole proposed pipeline has been embedded in the STAI-EWS tool. Just more details about the working-flow of the implemented options:

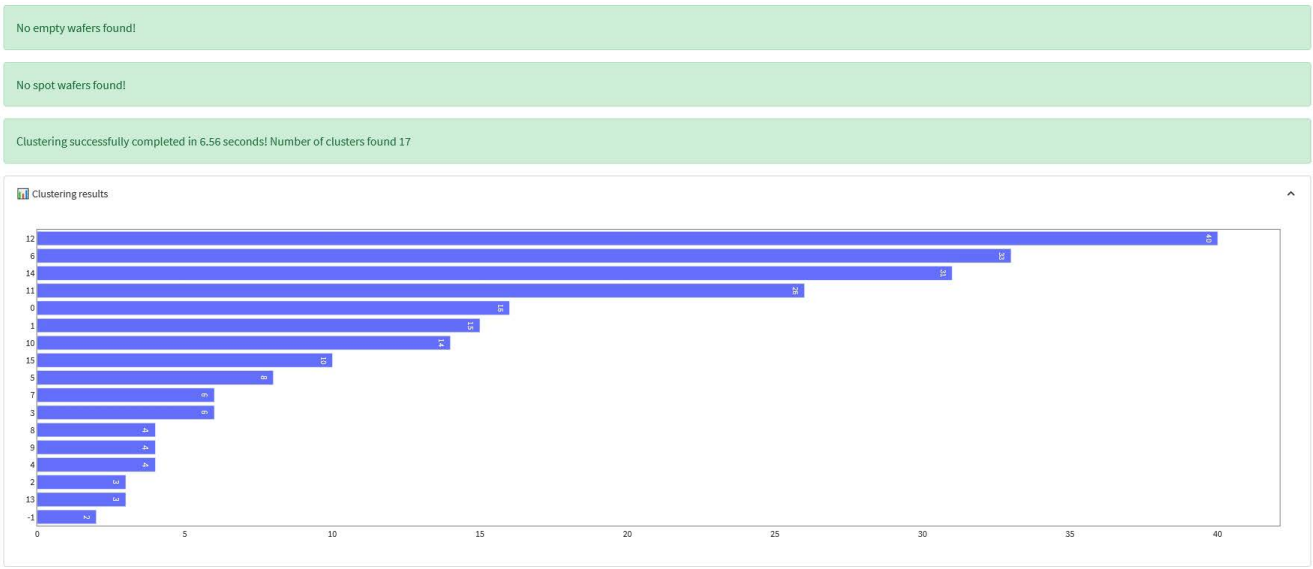
- **The Supervised wafer defect pattern recognition option.** As described in IV-B with this option, the user will be able to infer such input WDMs through the well-trained Convolutional Neural Network. Feed-forward inference can be done either as a single WDMs as well as by group of defect maps. The related classification of the input WDMs will be done with associated reports.
- **Unsupervised WDMs clustering option.** As described in IV-A this option allows the user to perform unsupervised clustering of the input WDMs followed by a downstream comparison with internal database looking for new wafer defect pattern classes. The GUI of the STAI-EWS tool shows the capability to change multiple parameters for UMAP and HDBSCAN such as the dimensionality reduction factors, filtering parameters, thresholds configuration, and so on. A related 3D Surface plot will be created to have an overview of the input WDMs against the adopted dimensionality reduction and hierarchical clustering configuration. The STAI-EWS tool allows the user to enable the K-Means centroids computation and related Cosine similarity. As introduced, in case of novel defect patterns the internal database will be automatically updated and the related CNN re-trained accordingly (this option can be disabled by the user). In Fig. 37 an instance of the unsupervised sub-system embedded in the STAI-EWS tool.
- **The Configuration-Management of the Database.** This section allows the user to configure the STAI-EWS tool internal defect maps database including the ability



(a) Dimensionality Reduction and Clustering options.



(b) Frequency and 3D Surface Plot of stacked WDMs.

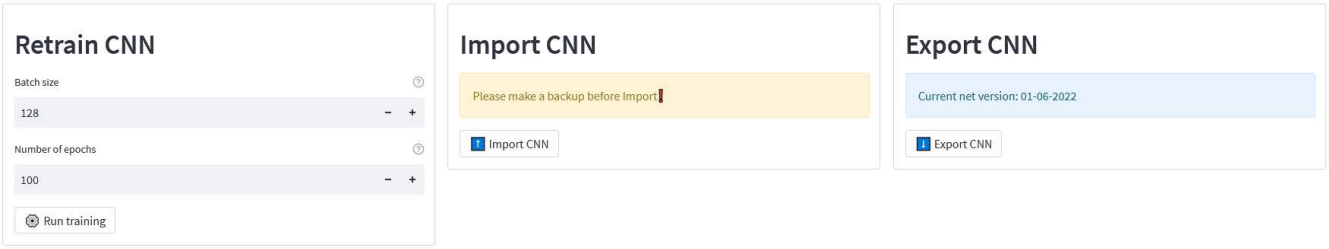


(c) Clustering results.

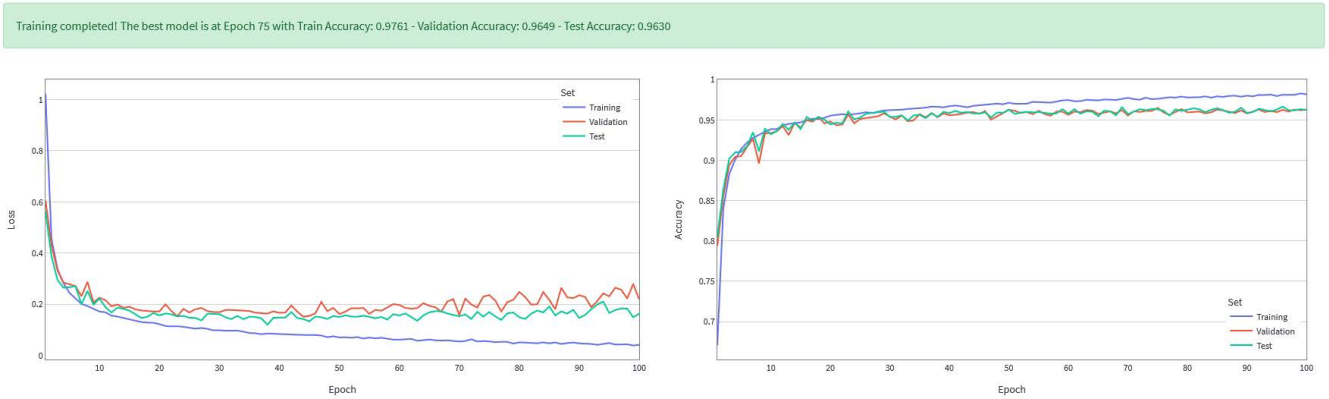
FIGURE 37. STAI-EWS tool: instance of the clustering report.

to backup of the current database. Moreover, the user can restore or update the database;

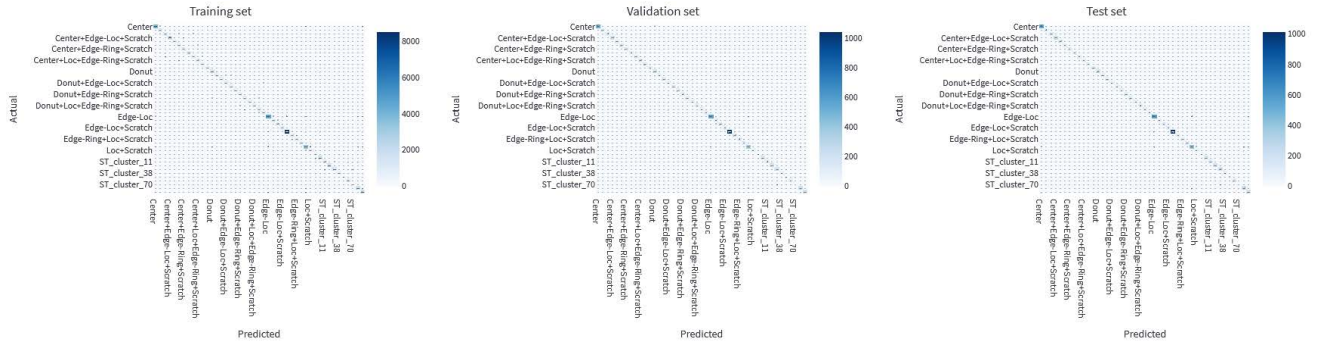
- **The Configuration-Management of the Deep Network (CNN).** In this section the user can validate or re-train the underlying deep CNN. A related benchmark report of the current CNN performance (including curves, confusion matrix, and so on) is generated at the end of the usage of this option. In Fig. 38 an instance of this option embedded in the STAI-EWS tool.



(a) Retrain /Import / Export CNN options.



(b) Loss and Accuracy curves for Training, Validation and Test sets.



(c) Confusion matrices for Training, Validation and Test sets.

Failure Type	Precision	Recall	F1-Score	Support
Center	0.9902	0.9848	0.9875	4198
Center+Edge-Loc	0.9863	0.9863	0.9863	802
Center+Edge-Loc+L	0.9812	0.9812	0.9812	796
Center+Edge-Loc+S	0.9961	0.9866	0.9914	1572
Center+Edge-Ring	0.985	0.9949	0.9899	791
Center+Edge-Ring+	0.976	0.9898	0.9828	781
Center+Edge-Ring+	0.9814	0.99	0.9857	798
Center+Loc	0.9878	0.9939	0.9908	816
Center+Loc+Edge-L	0.9848	0.9676	0.9761	803
Center+Loc+Edge-R	0.9788	0.9752	0.977	805
Center+Loc+Scratc	0.9847	0.9785	0.9816	791
Center+Scratch	0.9803	0.9938	0.987	802
Donut	0.9732	0.988	0.9806	1251
Donut+Edge-Loc	0.9838	0.9826	0.9832	805
Donut+Edge-Loc+L	0.9987	0.9974	0.9981	782
Donut+Edge-Loc+S	0.9829	0.9757	0.9793	823
Donut+Edge-Ring	0.9841	0.9901	0.9871	812
Donut+Edge-Ring+I	0.9975	0.9988	0.9981	801

Failure Type	Precision	Recall	F1-Score	Support
Center	0.9792	0.9829	0.9811	527
Center+Edge-Loc	0.9901	1	0.995	100
Center+Edge-Loc+L	0.9794	0.95	0.9645	100
Center+Edge-Loc+S	1	0.9898	0.9949	196
Center+Edge-Ring	0.981	1	0.9904	103
Center+Edge-Ring+	0.9833	0.9833	0.9833	120
Center+Edge-Ring+	0.9725	1	0.986	106
Center+Loc	0.9802	1	0.99	99
Center+Loc+Edge-L	0.9626	0.945	0.9537	109
Center+Loc+Edge-R	0.9813	0.9722	0.9767	108
Center+Loc+Scratc	0.9912	0.9825	0.9868	114
Center+Scratch	0.9888	1	0.9944	88
Donut	0.9062	0.9732	0.9385	149
Donut+Edge-Loc	0.9778	0.9888	0.9832	89
Donut+Edge-Loc+L	1	0.9744	0.987	117
Donut+Edge-Loc+S	0.9663	0.9663	0.9663	89
Donut+Edge-Ring	0.9714	0.9903	0.9808	103
Donut+Edge-Ring+I	0.9808	1	0.9903	102

Failure Type	Precision	Recall	F1-Score	Support
Center	0.977	0.9719	0.9744	569
Center+Edge-Loc	1	0.9694	0.9845	98
Center+Edge-Loc+L	0.9712	0.9712	0.9712	104
Center+Edge-Loc+S	0.9957	0.9914	0.9935	232
Center+Edge-Ring	0.9906	0.9906	0.9906	106
Center+Edge-Ring+	0.9897	0.9697	0.9796	99
Center+Edge-Ring+	1	1	1	96
Center+Loc	0.9882	0.9882	0.9882	85
Center+Loc+Edge-L	0.9773	0.9773	0.9773	88
Center+Loc+Edge-R	0.9663	0.9885	0.9773	87
Center+Loc+Scratc	1	0.9684	0.984	95
Center+Scratch	0.9565	1	0.9778	110
Donut	0.9615	0.9677	0.9646	155
Donut+Edge-Loc	0.9905	0.9811	0.9858	106
Donut+Edge-Loc+L	0.9806	1	0.9902	101
Donut+Edge-Loc+S	0.9663	0.9773	0.9718	88
Donut+Edge-Ring	1	0.9882	0.9941	85
Donut+Edge-Ring+I	1	0.9897	0.9948	97

(d) Precision, Recall and F1-Score for Training, Validation and Test sets.

FIGURE 38. **STAI-EWS tool: the configuration-management of the CNN.**

V. CONCLUSION

This work proposes an interesting hybrid approach to address one of the key-issue of semiconductor industries, i.e., the robust and effective defects assessment of the production lines. Through the combination of unsupervised and supervised deep pipelines we are able to early identify the defects in the production lines by means of a downstream analysis at EWS stage. By means of the investigated analysis of the associated binarized WDMs, we showed the ability of our proposed solution to provide a robust classification of the

defect patterns as well as an effective ability to identify new defect patterns which worth to be inspected in the upstream production lines. This hybrid solution enabled an end-to-end pipeline to be applied in the production lines of semiconductor company embedding different technologies. In fact, we have validated our solution in different environment both with public dataset and by using internal ones provided by STMicroelectronics. We have also validated our solution both in Silicon technology as well as in Silicon Carbide confirming the effectiveness of the proposed system both in unsupervised analysis (for identifying novel defect patterns) as well as in the supervised classification of the input well-known defect patterns. Through the usage of innovative dimensionality reduction and clustering features analysis (UMAP and HDBSCAN) we are able to build an internal robust representation of the features associated to the input wafer defect maps. Finally, by means of an XAI methods, we validated our solution by analyzing the activation maps of the designed deep network in order to check the internal representation of the used deep networks.

Finally, the released STAI-EWS tool allows a simple usage of the proposed pipeline by means of ad-hoc user-friendly interface currently used in the STMicroelectronics labs. Future works aim to extend the proposed architecture embedding the sub-systems which allow to automatically identify the upstream production issues associated to each of the classified (and novel) defect maps as well as to retrieve a robust assessment of the related production yield impact.

ACKNOWLEDGMENT

(Concetto Spampinato and Francesco Rundo contributed equally to this work.)

REFERENCES

[1] T. Kimoto and J. A. Cooper, *Fundamentals of Silicon Carbide Technology: Growth, Characterization, Devices and Applications*. Singapore: Wiley, 2014.

[2] *Introduction to Semiconductor Technology—An900 Application Note*, STMicroelectronics, Geneva, Switzerland, 2000.

[3] Y. Wei and H. Wang, "Mixed-type wafer defect recognition with multi-scale information fusion transformer," *IEEE Trans. Semicond. Manuf.*, vol. 35, no. 2, pp. 341–352, Mar. 2022.

[4] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015.

[5] U. Batool, M. I. Shapiai, M. Tahir, Z. H. Ismail, N. J. Zakaria, and A. Elfakharany, "A systematic review of deep learning for silicon wafer defect recognition," *IEEE Access*, vol. 9, pp. 116572–116593, 2021.

[6] N. Yu, Q. Xu, and H. Wang, "Wafer defect pattern recognition and analysis based on convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 566–573, Aug. 2019.

[7] H. Das, S. Sunkari, J. Justice, H. Pham, G. Park, and Y. H. Seo, "Statistical analysis of killer and non-killer defects in sic and the impacts to device performance," in *Materials Science Forum*, vol. 1004. Zürich, Switzerland: Trans Tech Publ, 2020, pp. 458–463.

[8] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018.

[9] R. Pasupathy, *Generating Homogeneous Poisson Processes*. Hoboken, NJ, USA: Wiley, Jan. 2011, Art. no. eorms0355.

[10] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 315–322, May 2018.

[11] Y. Kim, D. Cho, and J.-H. Lee, "Wafer map classifier using deep learning for detecting out-of-distribution failure patterns," in *Proc. IEEE Int. Symp. Phys. Failure Anal. Integr. Circuits (IPFA)*, Jul. 2020, pp. 1–5.

[12] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 3, pp. 395–402, Aug. 2018.

[13] J. Koo and S. Hwang, "A unified defect pattern analysis of wafer maps using density-based clustering," *IEEE Access*, vol. 9, pp. 78873–78882, 2021.

[14] Z. Shen and J. Yu, "Wafer map defect recognition based on deep transfer learning," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2019, pp. 1568–1572.

[15] J.-C. Chien, M.-T. Wu, and J.-D. Lee, "Inspection and classification of semiconductor wafer surface defects using CNN deep learning networks," *Appl. Sci.*, vol. 10, no. 15, p. 5340, 2020.

[16] C.-Y. Hsu and J.-C. Chien, "Ensemble convolutional neural networks with weighted majority for wafer bin map pattern classification," *J. Intell. Manuf.*, vol. 33, no. 3, pp. 831–844, Mar. 2022.

[17] J. Hwang and H. Kim, "Variational deep clustering of wafer map patterns," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 3, pp. 466–475, Aug. 2020.

[18] P. Tulala, H. Mahyar, E. Ghalebi, and R. Grosu, "Unsupervised wafermap patterns clustering via variational autoencoders," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[19] S. Park, J. Jang, and C. O. Kim, "Discriminative feature learning and cluster-based defect label reconstruction for reducing uncertainty in wafer bin map labels," *J. Intell. Manuf.*, vol. 32, no. 1, pp. 251–263, Jan. 2021.

[20] K. Nakata, R. Orihara, Y. Mizuoka, and K. Takagi, "A comprehensive 'big-data-based' monitoring system for yield enhancement in semiconductor man ufacturing," in *Proc. Int. Symp. Semiconductor Manuf. (ISSM)*, Dec. 2016, pp. 1–3.

[21] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining Knowl. Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[22] J. Yu, X. Zheng, and J. Liu, "Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map," *Comput. Ind.*, vol. 109, no. C, pp. 121–133, Aug. 2019.

[23] J. Yu and J. Liu, "Two-dimensional principal component analysis-based convolutional autoencoder for wafer map defect detection," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8789–8797, Sep. 2021.

[24] J. Wang, Z. Yang, J. Zhang, Q. Zhang, and W.-T. K. Chien, "AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 310–319, Aug. 2019.

[25] Y.-F. Yang and M. Sun, "Semiconductor defect detection by hybrid classical-quantum deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2323–2332.

[26] D. Kim and P. Kang, "Dynamic clustering for wafer map patterns using self-supervised learning on convolutional autoencoders," *IEEE Trans. Semicond. Manuf.*, vol. 34, no. 4, pp. 444–454, Aug. 2021.

[27] P.-C. Chen, "Defect inspection techniques in SiC," *Nanosc. Res. Lett.*, vol. 17, no. 1, p. 30, Dec. 2022.

[28] K. Takenaka, T. Tawara, and T. Kato, *Crystal Defect and Dislocation Analysis of SiC Wafers by Transmission Polarization Microscopy*, vol. 65. Tokyo, Japan: Fuji Electric Review, Dec. 2019.

[29] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

[30] I. Tiginyanu, V. Ursaki, and V. Popa, *Nanoimprint Lithography (NIL) and Related Techniques for Electronics Applications*. Amsterdam, The Netherlands: Elsevier, 2011, pp. 280–329.

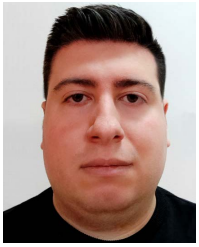
[31] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.

[32] J. M. Lee, *Introduction to Riemannian Manifolds*. New York, NY, USA: Springer Berlin Heidelberg, 2018.

[33] E. H. Spanier, *Algebraic Topology*, 1st ed. New York, NY, USA: Springer, 1995.

[34] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: Nn Pattern Classification Techniques*. Los Alamitos, CA, USA: IEEE Computer Society Press IEEE Computer Society Press Tutorial, 1991.

[35] L. McInnes, J. Healy, and S. Astels, "Hdbscan: Hierarchical density based clustering," *J. Open Source Softw.*, vol. 2, no. 11, p. 205, Mar. 2017.
[36] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
[37] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD. Rec.*, vol. 28, no. 2, pp. 49–60, Jun. 1999.
[38] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, 1957.
[39] D. M. Hillis, C. Moritz, and B. K. Mable, *Molecular Systematics*, 2nd ed. Sunderland, MA, USA: Sinauer Associates, 1996.
[40] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[41] J. Bussche and V. Vianu, *Database Theory ICDT 2001: 8th International Conference London, U.K., January 4-6, 2001 Proceedings* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2001.
[42] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.
[43] F. Pedregosa, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2012.
[44] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat., Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[45] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[46] J. Wang, C. Xu, Z. Yang, J. Zhang, and X. Li, "Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 4, pp. 587–596, Nov. 2020.
[47] A. Paszke, "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.
[48] *Pytorch Transfer Learning for Computer Vision Tutorial*. Accessed: Sep. 1, 2022. [Online]. Available: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[50] A. Dosovitskiy, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[51] Y. Bengio and O. Delalleau, "On the expressive power of deep architectures," in *Proc. 22nd Int. Conf. Algorithmic Learn. Theory*. Berlin, Germany: Springer-Verlag, 2011, pp. 18–36.
[52] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2924–2932.
[53] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," 2017, *arXiv:1703.01365*.
[54] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," 2016, *arXiv:1610.02391*.
[55] J. Gildenblat and Contributors. (2021). *Pytorch Library for Cam Methods*. [Online]. Available: https://github.com/jacobgil/pytorch-grad-cam/
[56] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," 2020, *arXiv:2005.00928*.
[57] *Streamlit*. Accessed: Sep. 1, 2022. [Online]. Available: https://streamlit.io/
[58] *Plotly*. Accessed: Sep. 1, 2022. [Online]. Available: https://plot.ly/



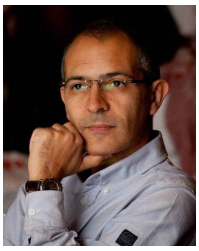
RICCARDO EMANUELE SARPIETRO received the master's degree in data science for management from the University of Catania, Italy, in 2021. His research interests include pattern recognition, object detection using deep learning algorithms, machine learning, and computer vision.

CARMELO PINO received the master's and Ph.D. degrees from the University of Catania, Italy. He worked as a Research Assistant with the University of Catania. He is currently a member of the Artificial Intelligence for Modeling and Predictive Reliability Team, STMicroelectronics, ADG Research and Development Power and Discretes, where he works as an Advanced Research Senior Engineer. Since 2014, he has been a member of the Pattern Recognition and Computer Vision Laboratory, University of Catania; and from 2020 to 2022, he was a Research Assistant with the National Institute for Astrophysics (INAF), Catania, Italy. His research interests include the areas of artificial intelligence applied to medical data processing, detection and segmentation in endoscopic video imaging systems, visual-knowledge ontology modeling, processing of radio-astronomical images, and temporal series analysis.

SALVATORE COFFA was born in Carlentini, Italy, in 1962. He received the Basic and Ph.D. degrees in physics from the University of Catania, Catania, Italy, in 1985 and 1991, respectively. For more than 30 years of research activity, he has achieved several important results in various fields, and more specifically, a large expertise in the field of technology transfer from basic research ideas to prototypes and then to products and applications. This expertise has been build up combining advanced research work (within or in cooperation with university, research labs, and small/medium enterprises) and application to technologies and products within STMicroelectronics. He has innovated front-end and back-end technologies in the field of power devices introducing new Si power structures (using thrench and thin wafers) and power structures in semiconductors, like SiC and GaN. SiC power devices are now in full mass production within STMicroelectronics. He has authored more than 250 publications in international refereed journals and holds more than 50 patents.

ANGELO MESSINA received the M.Sc. degree in electronics engineering from the University of Catania, in 1997, the Ph.D. degree in advanced technologies for the photonics and the opto-electronics and electromagnetic modeling, and the M.Sc. degree in physics from the University of Messina, where he worked four years in the former research team of the Physics of Matter and Electronic Engineering Department. He is appointed as an Independent Scientific Evaluator for Regional, Italian, and EU Research and Development and Innovation funded programs; a scientific conferences reviewer; or the chair. He is the coauthor of tens of publications. He was appointed associated for research to CNR-IMM in 2017. Since 1999, he has been working with STMicroelectronics, being now the Portfolio Manager of several EU Research and Development and Innovation Funded Project, working in the Public Affairs for Italy Department.

SIMONE PALAZZO received the Ph.D. degree from the University of Catania, Italy, in 2017, with a thesis on human–machine interaction modalities for object segmentation and categorization in images and videos. He is currently an Assistant Professor with the University of Catania. His current research interests lie in medical image analysis, continual learning, video object segmentation, and scene understanding.



SEBASTIANO BATTIATO (Senior Member, IEEE) received the degree *(summa cum laude)* in computer science from the University of Catania, in 1995, and the Ph.D. degree in computer science and applied mathematics from the University of Naples, in 1999. From 1999 to 2003, he was the Leader of the ''Imaging'' Team, STMicroelectronics, Catania. He joined the Department of Mathematics and Computer Science, University of Catania (as an Assistant Professor in 2004, an Associate Professor in 2011, and a Full Professor in 2016). He was the Chairperson of the Undergraduate Program in Computer Science (from 2012 to 2017) and the Rector's Delegate for Education: postgraduates and Ph.D. students (from 2013 to 2016). He is currently a Full Professor of computer science with the University of Catania. He is also the Scientific Coordinator of the Ph.D. Program in Computer Science (XXXIII-XXXVI cycles) and a Deputy Rector for strategic planning and information systems with the University of Catania. He is involved in research and directorship of the IPLab research laboratory (http://iplab.dmi.unict.it). He coordinates IPLab's participation on large scale projects funded by national and international funding bodies, and private companies. He has participated as a principal investigator in many international and national research projects He has supervised about 15 Ph.D. students and three postdoctoral researchers. He was the Director and the Founder of the International Computer Vision Summer School (ICVSS). He has edited six books and coauthored about 300 papers in international journals, conference proceedings, and book chapters, and has also been involved as a guest editor of several special issues published in international journals. He is also the co-inventor of about 22 international patents, a reviewer of several international journals, and has been a regular member of numerous international conference committees. His research interests include computer vision, imaging technology, and multimedia forensics. He was a recipient of the 2017 PAMI Mark Everingham Prize for the series of annual ICVSS Schools. He was the Chair of several international events, such as MMFORwild in 2020; IMPROVE in 2021; INTELLYSIS from 2020 to 2021; SIGMAP from 2019 to 2020; ICIAP in 2017; VINEPA in 2016; ACIVS in 2015; VAAM in 2014, 2015, and 2016; VISAPP from 2012 to 2015; IWCV in 2012; ECCV in 2012; ICIAP in 2011; ACM MiFor from 2010 to 2011; and SPIE EI Digital Photography in 2011, 2012, and 2013. He was an Invited Speaker/a Lecturer at several international conferences/meetings, such as SIGMAP in 2019, DFRWS EU in 2018, CompSysTech in 2016, ACIVS in 2016, EU IAI in 2015, IS&T Electronic Imaging from 2012 to 2015, and WIFS in 2015. He has been involved in the evaluation process of research projects on behalf of national and international institution, such as FTI-EU, MISE, MIUR, ANVUR, Invitalia, FinCalabra, FinPiemonte, Regione Friuli-Venezia Giulia, Regione Puglia, Research Promotion Foundation (RPF) of Cyprus, and Netherlands Organization for Scientific Research (NWO).

CONCETTO SPAMPINATO is currently an Associate Professor with the University of Catania, Italy. He is also a courtesy Faculty Member with the Center for Research in Computer Vision, University of Central Florida. In 2014, he created and currently leads the Pattern Recognition and Computer Vision Laboratory (PeRCeiVe Lab). His research interests include machine learning and its application in multiple domains from medical image analysis to autonomous robot navigation. He is also an Associate Editor of *Computer Vision and Image Understanding*, IEEE TRANSACTIONS ON MULTIMEDIA, and *Machine Vision and Applications* journals, and the Area Chair for multiple top-tier conferences, including CVPR 2022.

FRANCESCO RUNDO received the degree in computer science engineering and the Ph.D. degree in applied mathematics for technology from the University of Catania. He is currently a Senior Technical Staff Team Leader with STMicroelectronics, Catania. He is also a member of the Automotive Research and Development Power and Discretes Division, STMicroelectronics. He is also the Team Leader and the Project Leader regarding to the development of artificial intelligence-based solutions (hardware and software) for automotive, industrial, and medical applications. He is a member of the Computer Science Ph.D. Scientific Board, Department of Mathematics and Computer Science, University of Catania. He is also a member of the Computer Science Ph.D. Scientific Board, National Ph.D. Program of Artificial Intelligence. He has coauthored more than 100 contributions in international journals, conference proceedings contributions, SI series, posters, abstracts, and lectures. He is also the co-inventor of several international patents. His main research interests include advanced bio-inspired models, advanced and perceptual deep learning, embedded systems for deep learning algorithms, advanced deep learning, and mathematical modeling for automotive, industrial, and healthcare applications. He is a member of several international conference program committees. He serves as a reviewer and a guest editor for several special issues organized by such key-editors in the field of computer science. He serves as an Associate Editor for *IET Networks* and *Applied Computational Intelligence and Soft Computing* (Hindawi), a Research-Topic Editor for *Frontiers in Computer Science* and *Frontiers in Neuroinformatics*, and a Topic Editor for *Electronics* and *Drones* journals.