Friedrich, Michael; Schlosser, Tobias; Kowerko, Danny

Simulation of semiconductor wafer dicing induced faults on chips and their application as augmentation method for a deep learning based visual inspection system

Journal of Intelligent Manufacturing

Journal of Intelligent Manufacturing (2026) 37:573–596
https://doi.org/10.1007/s10845-024-02559-0

Simulation of semiconductor wafer dicing induced faults on chips and their application as augmentation method for a deep learning based visual inspection system

Michael Friedrich[1] · **Tobias Schlosser**[1] · **Danny Kowerko**[1]

Received: 31 March 2024 / Accepted: 23 December 2024 / Published online: 13 January 2025

Abstract
In semiconductor wafer dicing, one particular area of interest is the process of visual inspection to detect manufacturing defects that occur throughout the manufacturing process. Emerging defect patterns are typically in the micrometer range, translating to barely visible defects in pixel size on high-resolution imagery. The availability of rarely occurring defects is generally limited due to the labor-intensive nature of the related, highly specialized annotation task. Therefore, this contribution proposes a hybrid system for wafer and chip image data synthesis utilizing wafer and dicing path templates for the synthetic generation of large quantities of labeled flawless and, rarely, faulty chip and dicing street imagery. These are utilized for subsequent deep learning based defect detection and classification by employing a residual neural network as the core classifier for our visual inspection system. Our results show promising prospects when the original image data are supplemented with synthesized images by creating so-called composite data sets. Compared to the system's baseline on the original data set, an F1-score-based relative improvement of up to 3.98 times was achieved. Furthermore, a novel synthetic-composite leave-one-out cross-validation (SC-LOOCV) method is proposed as a means to analyze the quality of our synthesized data for each specific wafer type. Based on these experiments, we scored a relative improvement of up to 5.99. For all our wafer types, overall relative improvement factors of 1.99 (composite) and 2.83 (SC-LOOCV) highlight the benefits of our realized system.

Keywords Computer vision · Pattern recognition · Visual inspection · Data synthesis · Deep learning · Convolutional neural networks

Introduction and motivation

The need for visual inspection based yield optimization within semiconductor manufacturing arises primarily due to the industry's high quality standards for the manufacturing process. Consequently, the associated costs for a semiconductor wafer are derived from both the required materials as well as the individual processing steps within the complex manufacturing process. Here, the so-called yield describes a quality criterion based on the ratio of flawless chips to total chips after the process of wafer separation (Lee et al., 2017). However, as manual inspection of the resulting semiconductor products is very labor intensive and requires expert knowledge, DL-based classification algorithms such as deep neural networks (DNN) are a suitable approach to solve the problem of (semi-)automated visual inspection (Schlosser et al., 2022). The availability of data, their annotations in the form of labels, as well as their characteristics regarding defect patterns and their occurrences are of central importance. However, the acquisition of labeled ground truth data is a costly and error prone process performed by human experts within the field. Therefore, data synthesis approaches may alleviate common issues associated with otherwise sparsely available ground truth data.

The underlying data often consists of high-resolution real-world recordings of separated semiconductor wafers (Fig. 1, left). A commonly used separation approach involves the

✉ Danny Kowerko
danny.kowerko@cs.tu-chemnitz.de

Michael Friedrich
michael.friedrich@cs.tu-chemnitz.de

Tobias Schlosser
tobias.schlosser@cs.tu-chemnitz.de

[1] Chemnitz University of Technology, 09107 Chemnitz, Germany

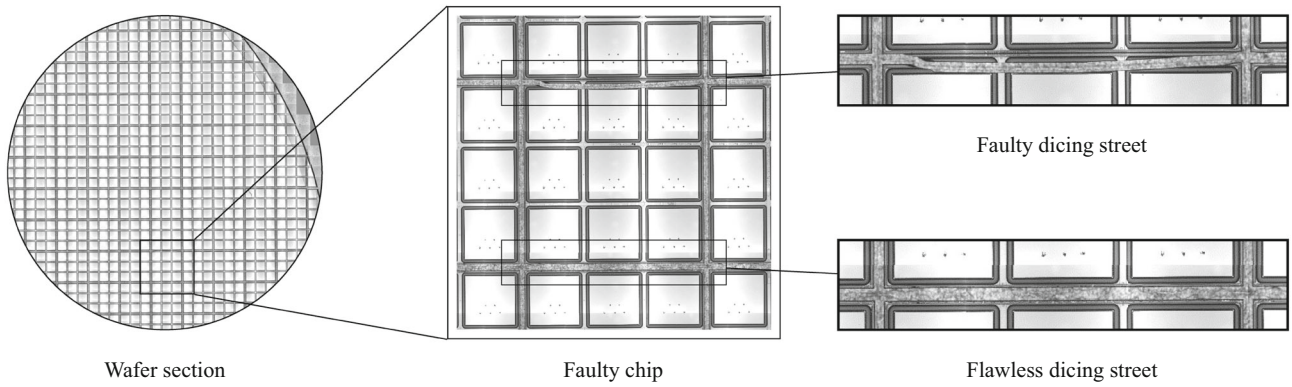


Fig. 1 Overview of the areas of interest within our underlying original wafer data. The quality of a chip depends on the characteristics of its four surrounding dicing streets. When no dicing street deviates towards the inner structures of the currently observed chip, the chip is considered flawless. Otherwise, it is considered faulty

use of a dicing saw (Hooper et al., 2015). An alternative method, on which our data set is based on, is thermal laser separation, where a thermally induced mechanical force creates a cleave on the wafer surface (Rahim & Mian, 2017). Typically, the separation process follows a scribed area on the wafer surface. Alongside this marking, the material is cleaved. The resulting cleave is barely visible due to the very small size of the resulting components. A typical wafer is up to 300 mm in diameter and can contain several thousand chips (Rahim & Mian, 2017). The wafer itself is situated on a dicing tape, an elastic material that is expanded after the separation process to visualize the course of the cleave, whereas the resulting gap between chips is called a dicing street (Fig. 1, middle). Analyzing the possible deviation of the cleave from the scribed area is the core task of visual inspection. An example of a flawless and a faulty dicing street is shown in Fig. 1 (right). The resulting data set suffers heavily from a class imbalance between flawless and faulty chips, signifying a general difficulty for learning-based algorithms. For this purpose, our proposed solution involves the design of a data synthesis approach that alleviates the underlying issue of data imbalance as well as rarely occurring or missing samples.

Furthermore, the assessment of the impact of data synthesis in relation to the described inspection problem not only aims to alleviate this issue but also raises the question of the general classification performance given the underlying data set and its improvement by means of domain randomization (DR) (Tobin et al., 2017). DR refers to the transfer quality of classification capabilities based on synthetic data and its comparison to real-world data. Therefore, whether the classification capabilities of our system can be improved without changing the nature or complexity of the underlying DL algorithm is investigated.

Related work

The following sections provide an overview of related work relevant to the simulation of semiconductor wafer dicing induced defects on chips and their application as augmentation method for deep learning based visual inspection systems. For this purpose, we differentiate between visual inspection based approaches in general in "Visual inspection" section while also giving an overview of issues and potential solutions related to data scarcity in "Data scarcity" section. Subsequently, the related, necessary steps for data synthesis are discussed in the context of learning-based visual inspection in "Data synthesis" section.

Visual inspection

While a wide range of conventional visual inspection methods exist within the domain of semiconductor manufacturing (Huang & Pan, 2015), encompassing approaches such as projection, filter-based, and baseline ML approaches, conventional approaches often require the implementation of two stages: (i) feature extraction and (ii) defect identification. In order to design such a system, extensive domain knowledge is often required. However, the recent advancements in DL have provided tools that enable a more generic classification within one stage (Zheng et al., 2021). Additionally, DL-based approaches have now outperformed other (learning-based) methods in many fields.

The following contributions are related to the tasks of (semi-)automated visual inspection by utilizing DL algorithms based on wafer data sets that are either related to wafer maps or the direct inspection of defects on the wafer surface. In Chien et al. (2020), four visible surface defects on semiconductor wafers were classified using a convolu-

tional neural network (CNN) as well as a pre-trained faster region-based CNN (Faster R-CNN). Both approaches reach similar results, with the highest accuracies ranging from 98 to 99%. Another approach presented by Saqlain et al. (2020) analyzed wafer maps based on their proposed CNN for automatic wafer defect identification (CNN-WDI), reaching an average classification accuracy of 96.2% on nine different wafer map defects. Imoto et al. (2019) proposed a CNN with a transfer learning (Weiss et al., 2010) based approach, analyzing defects on a wafer surface with a data set consisting of 5386 samples, which significantly outperformed traditional automatic defect classification systems. Schlosser et al. (2019) introduced a multistage system for the classification of defects occurring on wafer surfaces. The proposed stacked hybrid CNN (SH-CNN) combines the benefits of classical image processing algorithms with artificial neural networks. The resulting chip classifications are evaluated based on their underlying street classifications, achieving a mean accuracy of 92%. Beuth et al. (2020) proposed a similar system by combining a biologically plausible model of visual attention with DL, where the process of street region cropping is realized by the proposed visual attention model, reaching an average classification accuracy of 91.81% on the wafer defect data set. Following, Schlosser et al. (2022) refined the introduced system by Schlosser et al. (2019), utilizing, among other learning-based models, the residual neural network (ResNet) *ResNet152V2* (He et al., 2016a, b) as backbone model while increasing the level of detail even further with the introduction of a segment-based street classification, reaching an overall classification accuracy of 99.5% in F1-score.

Several other topics similar to our wafer problem address the visual inspection process of small defects within high-resolution imagery based on DL algorithms (Lin et al., 2023). Chen et al. (2015) evaluated the performance of a CNN-based approach for classifying 12 different defect patterns occurring on gearboxes, reaching a global percentage of true positives over all defect patterns of 98.4% with a corresponding total error of 1.6%. Cha et al. (2018) analyzed various structural defect types within high-resolution images of steel and concrete elements. They proposed a Faster R-CNN, reaching a mean average precision of 87.8% over all defect types. Fotouhi et al. (2021) employed a pre-trained version of AlexNet for the visual inspection of laminated composite structures such as wind turbine blades and aircraft, achieving accuracies between 87 and 96% for identifying defect types and their severity.

Data scarcity

According to Bansal et al. (2022), data scarcity leads to four major challenges in the context of deep learning: (i) lack of relevant data, (ii) limited training data, (iii) model overfitting, and (iv) data imbalance. Overcoming those challenges is crucial for successfully deploying learning-based system in the real-world. For image data, a wide range of augmentation methods exist that can help to alleviate those issues (Shorten & Khoshgoftaar, 2019). Such augmentation methods can be grouped into three categories: (1) basic image manipulations, (2) deep learning approaches, and (3) meta-learning. (1) Basic image manipulations include operations such as geometric or color space transformations, kernel filters, mixing images (Inoue, 2018), and random erasing (Zhong et al., 2020). They are applied within the data space and are fast and straightforward to implement with the ability to incorporate domain-specific knowledge. (2) Deep learning approaches on the other hand encompass techniques such as adversarial training (Goodfellow et al., 2014), neural style transfer (Gatys, 2015), and generative adversarial network (GAN) based data augmentation (Tanaka & Aranha, 2019). They typically take place within the feature space and are therefore more difficult to fine-tune and interpret. (3) Last but not least, meta-learning strategies such as neural augmentation (Perez & Wang, 2017), AutoAugment (Cubuk et al., 2018), and smart augmentation (Lemley et al., 2017) are multi-layered learning strategies, where artificial neural networks are utilized to optimize augmentation strategies instead of manually designing augmentation strategies. Choosing the optimal augmentation strategy for a real-world use case is generally challenging. However, Wong et al. (2016) states that if a viable strategy for image augmentation within the data space can be found, it provides benefits over other augmentation approaches, such as feature space augmentations, in terms of reducing overfitting and increasing model performance.

Data synthesis

In the following, different approaches involving data synthesis for the purpose of alleviating issues related to data scarcity for problems such as sparse and imbalanced data as well as the insufficient availability of high-quality annotations are introduced (Nikolenko, 2021). In particular, annotated defect data in the domain of visual inspection are difficult to obtain since the labeling process itself is time consuming and requires expert knowledge, which in turn may result in costly data preparation investments (Peres et al., 2021).

Within Maksim et al. (2019), the publicly available *WM-811K* wafer map data set (Wu et al., 2014) was investigated by creating composite data sets consisting of a fraction of the original data from *WM-811K*, for which the training set was supplemented with synthetically generated data based on parametric models. By utilizing residual neural networks, they achieved a classification accuracy of up to 87.8%.

In comparison, Gupta et al. (2016) proposed a fully convolutional regression network (FCRN) for the recognition

of text in natural scenes. To achieve this goal, they trained their network on a synthetic data set generated by mapping text snippets on objects in real-world scenes without any manual annotations. Their proposed approach achieved an F-score of 84.2%, outperforming current methods for detecting text within natural images. Another approach that utilizes the synthetic enrichment of real-world scenery was proposed by Ekbatani et al. (2017), where the underlying problem was concerned with the counting of pedestrians. They created a data set of one million highly realistic images consisting of real-world scenes that were synthetically enhanced with up to 29 pedestrians. The resulting data set was fed into a CNN for classification, managing to predict the number of pedestrians within a scene to a satisfactory extent. Tremblay et al. (2018) analyzed the effects of DR for the recognition of car bounding boxes within real-world scenes by utilizing various state-of-the-art ANNs. The training was conducted on synthetic data with a wide range of generation parameters, even purposely creating unrealistic data of cars within random scenes to emphasize the effects of DR, forcing the underlying networks only to learn essential features. The achieved results are promising and comparable to real-world data sets without the need for labor-intensive annotation work, showing that DR is a suitable method for bridging the gap between synthetic and original data. The contribution of Tripathi et al. (2019) evaluated the viability of generating synthetic data via image composition with the goal of augmenting the training data. They proposed a framework with a trainable synthesizer as well as a target network where training is performed in an adversarial manner. Their results show that fewer samples are required to reach certain accuracies compared to conventional, random data augmentations, outperforming state-of-the-art person detection algorithms, such as the Single Shot MultiBox Detector (SSD), by 2.7%.

Contribution of this work

The contribution of this work is the exploration of potential data augmentation approaches in combination with different data set structuring methods and their combined impact on the classification capabilities of various deep neural network models. The given data baseline consists of images from a visual inspection process within the semiconductor manufacturing domain, constituting a difficult imaging challenge due to small defect sizes in comparison to the overall image resolution. Therefore, the core contributions are summarized as follows:

1. **Data synthesis system.** The design and implementation of a data synthesis system for the simulation of semiconductor wafer dicing induced defects on chips. The proposed data synthesis system (see also Fig. 3) is based on the assumption that content from the original wafer data can be reused within parts of the simulation process.
2. **Data set structuring.** We generated various data subsets consisting of original and synthetically generated chip data to gain a better understanding of the individual benefits and limitations of synthetically generated data in a wide range of experiments.
3. **Comprehensive model performance analysis.** In order to assess the simulation quality of our generated synthetic data sets, we employed them as training data for, among other learning-based models, our classification stage's backbone model, *ResNet152V2*. *ResNet152V2* was selected as a baseline for our conducted experiments to obtain results comparable to those of Schlosser et al. (2022). However, we also provide results on further, well-established DL models. The process of generating synthetic data while subsequently evaluating the respective classification results was repeated multiple times until the functionality and parameterization of our synthesis system achieved satisfying results. Two groups of experiments regarding the quality of the generated synthetic data sets were conducted. Finally, we confirmed the progressiveness our approach in comparison to commonly deployed image augmentation methods.

Fundamentals and implementation

The following sections provide an overview of the key concepts and the implementation of our system. We begin by introducing the fundamentals of generating wafer dicing induced defects on chips using splines for defect approximation in "Splines as an approach for defect approximation" section. Next, we present our data augmentation synthesis pipeline in "Synthesis pipeline" section, followed by a description of our classification system and learning-based approaches in "Classification system" section. Additionally, we provide an overview of the wafer chip data set in "Data set overview" section and outline the structure of our experiments in "Experiment structure" section.

Splines as an approach for defect approximation

Splines are often utilized for smooth and flexible interpolation or approximation of data points. Splines do have a set of different key advantages (Wold, 1974). (i) They are mathematically founded, which allows their description as mathematical functions. (ii) Given these functions, they allow the approximation of physical processes, including our application area, semiconductor wafer dicing, where a cutting laser is guided along a scribed area of interest, resulting in a set of chips and their dicing streets. (iii) Therefore, they are, in terms of their underlying complexity, more compre-

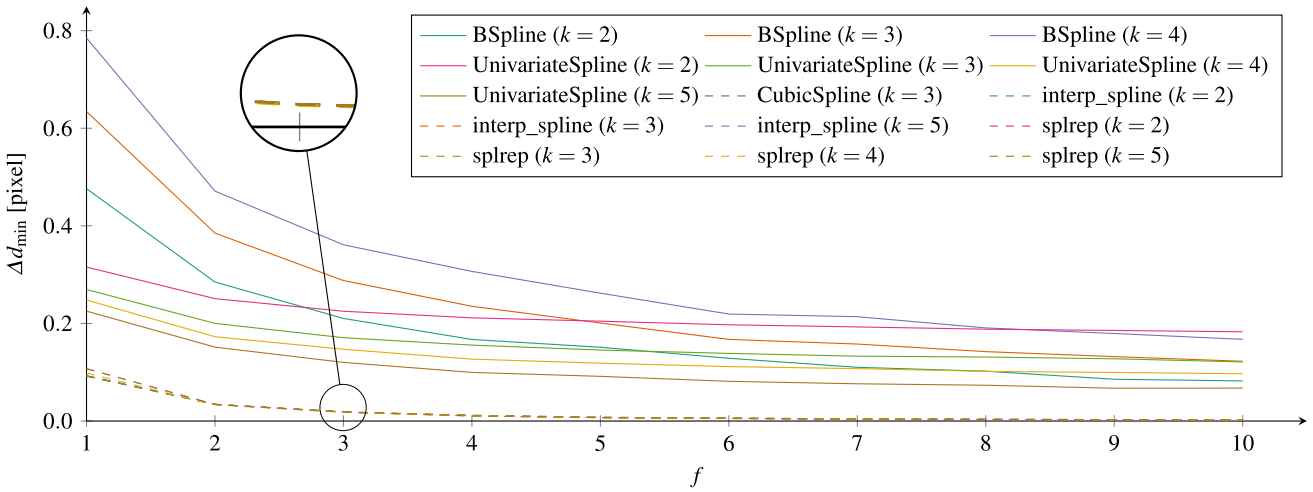


Fig. 2 Approximation accuracies of various spline interpolation methods provided by the Python library SciPy, where k represents the degree of the underlying piecewise polynomial function. The calculation for $\Delta d_{\min}$ is shown in Eq. 1. f is the sampling factor, which controls the rate at which the resulting spline functions are sampled. All dashed lines represent results starting at a mean error of $\Delta d_{\min} < 0.1$

hensible in their usage as simulation parameters. However, their heterogeneity depends on the underlying functions that are used for simulation (problem of homogeneity (Ruppert & Carroll, 2000)).

In order to determine the most suitable spline interpolation method for the simulation of faulty dicing streets, a detailed analysis was conducted on how well different common interpolation options approximate the underlying defect measurements (see also Fig. 2). To achieve comparable evaluation results, the averaged shortest distance $\Delta d_{\min}$ from each measured ground truth defect, i.e., error point, to the resulting sample points of the spline interpolation was calculated. This process was repeated for all defects within the ground truth, also shown in Eq. 1.

Assume we have a set of n defects $\mathcal{P} = \{P_i | i = 1 \dots n, n \in \mathbb{N}\}$, where each defect P_i corresponds to a subset of m measured points $P_i = \{p_{ij} | j = 1 \dots m, m \in \mathbb{N}\}$. Each point p_{ij} is represented by a 2D coordinate $p_{ij} = \{(x, y) | (x, y) \in \mathbb{R}^2\}$. For each P_i, the spline S_i is approximated as a sequence of o spline segments $S_i = \{s_{il} | l = 1 \dots o, o \in \mathbb{N}\}$, where the number of spline segments o is determined by the number of measured points within the current defect, which is scaled by a sampling factor f. The distance d_{ij} from a point p_{ij} to the spline S_i is the minimum distance from p_{ij} to the closest segment $s_{il} \in S_i$. The relevant spline segments s_{il} are selected within an ϵ-area around the measurement points p_{ij}. It is calculated as follows:

$$d_{ij} = \min_{l \in \{1,\dots,o\}} \left(\frac{\| s_{il} \times (p_{ij} - a_{il}) \|}{\| s_{il} \|} \right) \tag{1}$$

where p_{ij} is the measured point within P_i, a_{il} is the starting point of a spline segment s_{il}, s_{il} is the spline segment of spline S_i.

To incorporate the iteration over all defects and their respective subsets of measured points, the average error for each subset P_i is calculated as follows (Schlosser et al., 2024a):

$$\text{Error}(P_i) = \frac{1}{m} \sum_{j=1}^{m} d_{ij} \tag{2}$$

The averaged shortest distance $\Delta d_{\min}$ is defined as:

$$\Delta d_{\min} = \frac{1}{n} \sum_{i=1}^{n} \text{Error}(P_i) \tag{3}$$

This reflects the calculation over subsets of measured points and the computation of distances from these points to the respective spline. The results for various spline interpolation methods are highlighted in Fig. 2. Their related implementations can be found with the Python library SciPy, for which their class or function names are provided. For all following considerations, we selected the spline functionality of "interp_spline" with B-spline degree $k = 3$ as it shows one of the best overall approximation results.

Synthesis pipeline

Our proposed data synthesis system in Fig. 3 consists of 8 processing steps ranging from the retrieval of input data and

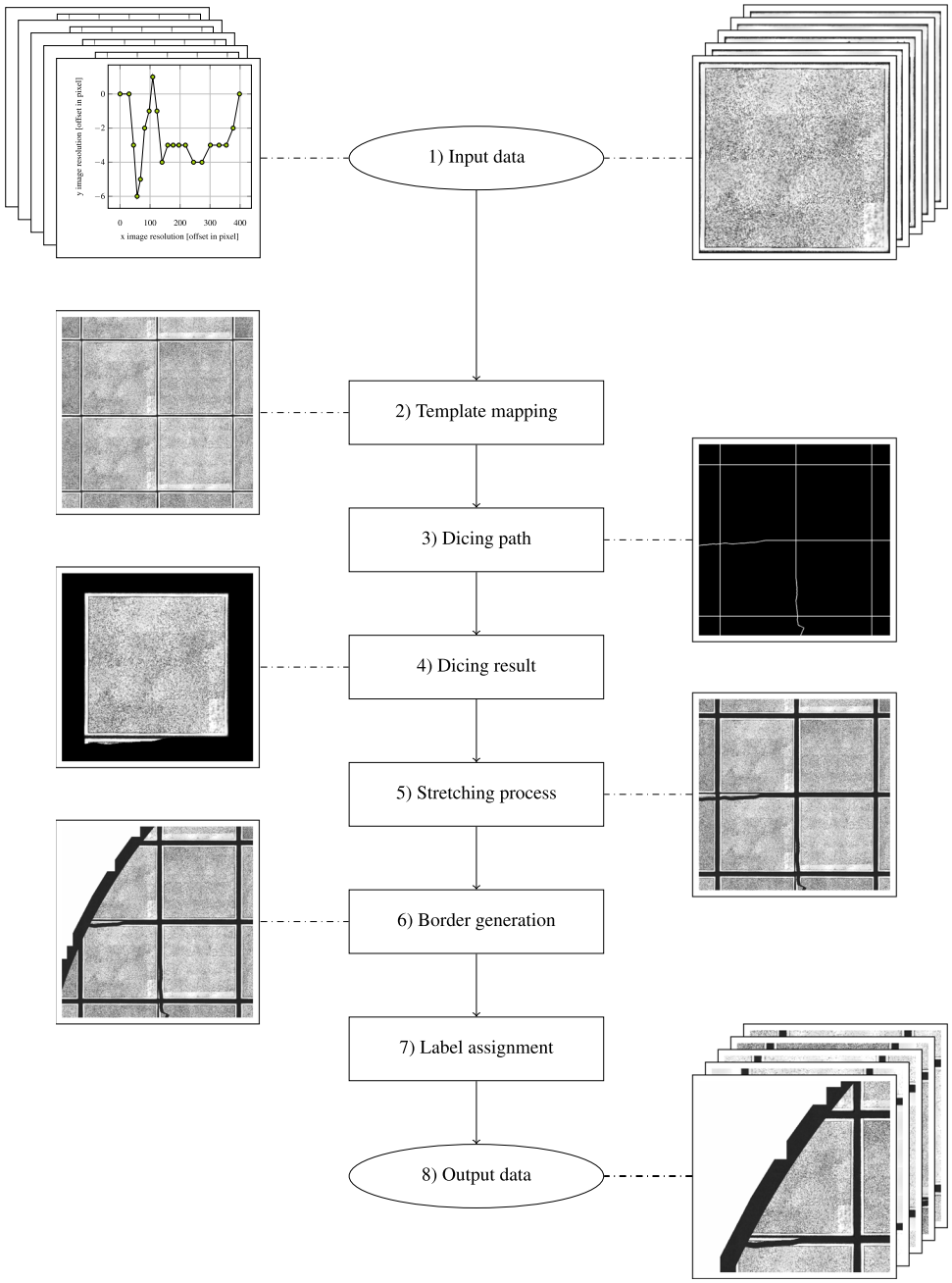


Fig. 3 Control flow chart of our proposed data synthesis system and its related processing steps. For a detailed description of the related processing steps, see "Synthesis pipeline" section

our template mapping in "Input data" and "Template mapping" section to the label assignment and the output of data and synthesis results in "Label assignment" and Output data" section. In the following, we will explain every step of our pipeline within the subsequent section. The synthesis system itself is implemented in the Python programming language by utilizing publicly available computational libraries such as NumPy,[1] OpenCV[2] and SciPy.[3]

[1] NumPy project page, https://numpy.org/.

[2] OpenCV project page, https://opencv.org/.

[3] SciPy project page, https://scipy.org/.

Input data

To ensure a wide range of possible synthesized wafer solutions, the option to map textures onto all areas of the resulting wafer, chip, and street surfaces is provided. As most of the texture mappings are optional, only the provision of digitized defects (Fig. 3, 1, left) and chip templates (Fig. 3, 1, right) is required. The chip templates represent a set of images that constitute the core of the wafer synthesis process where all the provided chip templates are required to have the same image dimensions for the following mapping process. To simulate the deviation of the dicing laser from the scribe lines, faulty

sections of real wafers are sampled and stored as (x, y) point sets. These are subsequently used as variable inputs for our spline interpolation, which enables the approximation of real defect paths.

Template mapping

Initially, the chip templates are mapped on a grid of a given input size representing the dimensions of the desired wafer to be generated. Since the chips themselves show different structures and levels of illumination, it is essential that templates are provided in a sufficient variety to properly represent the underlying data set. The visible lines between the mapped chips in Fig. 3 (2) represent the scribe lines for the separation process.

Dicing path

In Fig. 3 (3), a map of the generated dicing paths is shown. Straight lines represent a dicing street that follows the scribe properly, whereas a dicing street that deviates from the scribe represents a defect. These defects are generated via spline interpolation based on real-world defects measurements within the original data set. In order to enable a unified fitting of the resulting spline functions, all measured defects are prepared as orientationally aligned defect templates. The resulting dicing path map (Fig. 3, 3) generated in this manner corresponds to the previously created template mapping (Fig. 3, 2).

Dicing result

To simulate the dicing process (Fig. 3, 4), a combination of morphological and masking operations is deployed. Initially, a cropping of the template map (Fig. 3, 2) based on a mask corresponding to segments of the dicing path map is obtained (Fig. 3, 3). However, to prevent data loss in the original chip templates, the morphological operation dilation is applied to the resulting mask throughout the separation process. A more detailed visualization of this process is depicted in Fig. 4, which shows the individual processing steps ranging from the generation of the related mask to the synthesis of the final dicing result.

Stretching process

To visualize the dicing streets, the stretching process of the wafer's underlying dicing tape is simulated by applying the respective translation vectors to the positions of the diced chips, therefore taking the expansion of the dicing tape into consideration. Subsequently, the resulting gaps between the chips are made visible as shown in Fig. 3 (5).

Border generation

At this point, all chips and their related dicing streets are fully realized, leaving only the simulation of the wafer border to complete the typical circular look of a wafer as visualized in Fig. 1 (left). Therefore, the generated grid is cut based on the given wafer size, for which different simulation options for the wafer border itself exist. One such example is shown in Fig. 3 (6). Here, the border consists of the wafer's underlying dicing tape, which is overlapping past its border.

Label assignment

As all generative steps of the synthesis process are now complete, an assignment of labels for the resulting chip and street segments is mandatory for them to be usable for out-of-the-box training purposes (Fig. 3, 7). This is achieved by considering the relative position of the chip to the wafer border. In addition, a more detailed assessment is carried out based on the dicing streets that surround these chips. The labels flawless and faulty are assigned based on the current dicing path's course on the underlying dicing path map for each street (Fig. 3, 3). Each chip that is adjacent to at least one faulty dicing street is labeled as faulty.

Output data

Following the label assignment of chips and streets on the synthesized wafer, the resulting areas of interest, i.e., chips and streets, are automatically cropped from the synthesized wafer image and prepared as usable data sets (Fig. 3, 8). The resulting sections of synthesized chips and streets are stored as labeled data sets given the addressing scheme proposed by Schlosser et al. (2022). With the class labels of flawless and faulty chips and streets, the resulting class assignments are encoded within our obtained data organization, whereby, among others, the locations of chips and streets are encoded with their respective file names.

Classification system

Following the implementation of our hybrid multistage system of stacked deep neural networks (SH-DNN) in Schlosser et al. (2022), the processing steps of localization and classification as well as data augmentation are utilized on wafer and chip data. No additional changes were introduced for the localization steps of chips and streets within our system. For the subsequent classification stage, The *Hexagonal Image Processing Framework Hexnet*[4] was utilized to enable the application of general learning-based classification approaches (Schlosser et al., 2024b). In Hexnet, the

[4] Hexnet project page, https://github.com/TSchlosser13/Hexnet.

(a) Original chip image (b) Snipped from the defect map (c) Generated mask image (d) Resulting faulty chip

Fig. 4 Visualization of chip imagery for the required processing steps in Fig. 3 (4). Shown are **a** the original chip currently to be processed, **b** a sample from the selected defect map, **c** the subsequently generated mask for the dicing process, and **d** the resulting faulty chip

ML framework Keras[5] as well as its underlying back end TensorFlow[6] are employed, which in turn enables an accelerated processing via general-purpose graphics processing units (GPGPU).

Similar to Schlosser et al. (2022), our proposed training setup includes: the Glorot initializer for weight initialization, the Adam optimizer with a standard learning rate of 0.001 and exponential decay rates of 0.9 and 0.999, as well as a batch size of 32. Each investigated DNN is evaluated after 50 epochs of training, for which the results of five training runs are assessed in terms of the mean and standard deviation in classification performance. To avoid interference between training and test data, we conducted our test runs with a randomized data split ratio of 80/20% for our training and test sets. For all learning-based approaches and DNNs, no changes to their architectures or parameterizations were introduced. No optimizations by means of pre-trained weights, by means of transfer learning (Weiss et al., 2010), or by deploying additional optimization approaches such as grid searches were introduced. As emphasized in Schlosser et al. (2022), different learning-based approaches may yield different results depending on the current classification stage of the system.

Data set overview

The utilized original data set comprises various wafer types from the contributions of Beuth et al. (2020); Schlosser et al. (2022), which all contain different structures in terms of varying integrated circuit designs, illumination, noise, texture, and image resolutions, which range from 224×224 to 912×908 pixels per chip. The varying structures on the chip and wafer surfaces can be summarized as five different wafer types as shown in Table 1. However, it is noted that a sixth wafer type contained in Beuth et al. (2020) and Schlosser et al. (2022) was discarded as it does not contain any suitable defects to assess the quality of our proposed synthesis system. Additionally, each chip is categorized as either flawless or faulty depending on the quality of its surrounding dicing streets. When one of the dicing streets surrounding a chip deviates from the scribed area towards the inner chip area, the chip is considered faulty. Otherwise, it is considered flawless. In total, we differentiate between five different defect types that occur within the data set. A visual comparison of original and synthetically generated samples is given in Table 2. Within those error classes, the error length varies between 4 pixel for the smallest and 3621 pixel for the largest error. On the shorter end, the spectrum represents defect types such as nose and chipping, which are typically confined to the area of a single chip. Conversely, the longer end of the spectrum represents more significant deviations, such as oversize and undersize, which can span multiple chips on the wafer surface due to substantial misalignment of the dicing street from the intended scribe. Given that our faulty class is defined to include only chips whose geometric size was reduced by deviations of the dicing street, we classified the defect types chipping, undersize, and border as faulty, while nose and oversize were included in the flawless class.

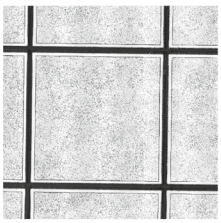
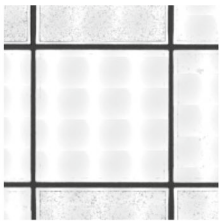
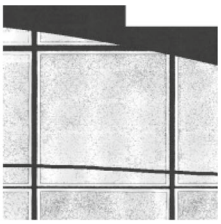
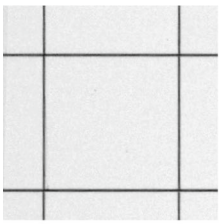
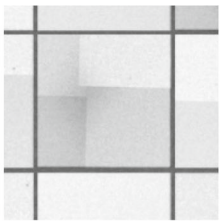
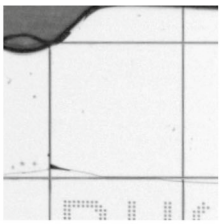
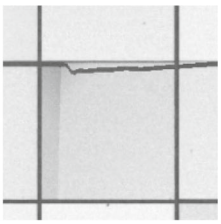
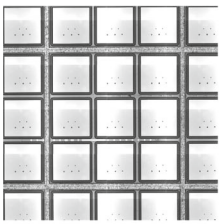
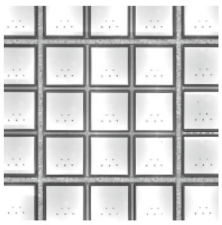
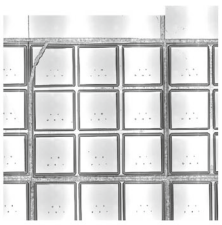
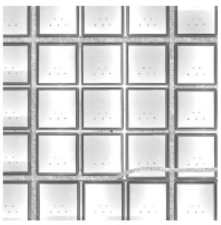
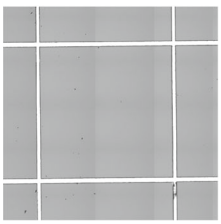
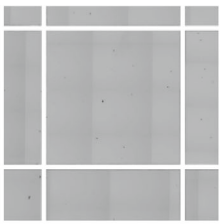
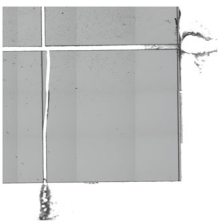
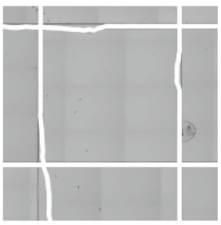
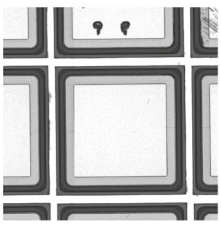
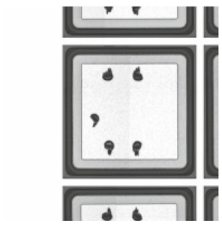
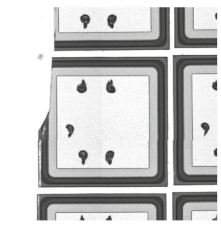
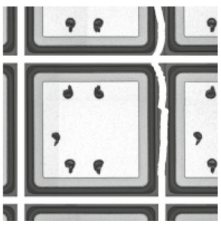
Since this contribution is primarily concerned with the impact of data synthesis on DNN classification capabilities, we employ a one-shot, or single-stage, classification strategy. Consequently, every image derived from a rasterized wafer image undergoes processing by a singular DL classifier as opposed to the multistage classification systems of our previous contributions Schlosser et al. (2019, 2022). While this approach increases the difficulty of the classification problem by considering images located on the wafer border and beyond as faulty, it also enables the emphasis of increased classification scores that would otherwise easily reach accuracies of over 99% (Schlosser et al., 2022).

All data sets suffer from class imbalances due to the limited number of faulty samples, for which the data sets were augmented by utilizing oversampling for their faulty data

[5] Keras project page, https://keras.io/, version 2.3.1.

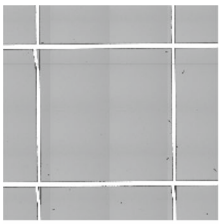
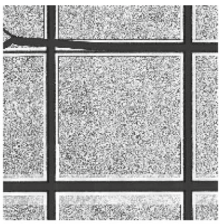
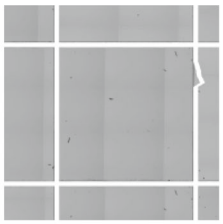
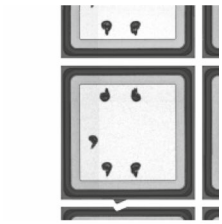
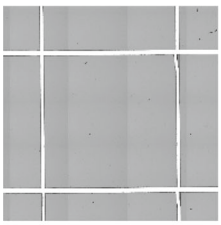
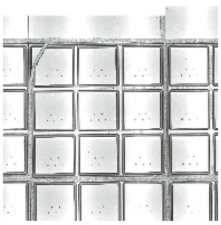
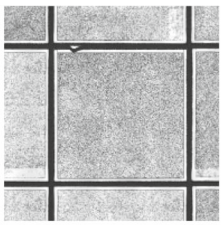
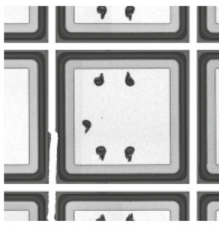
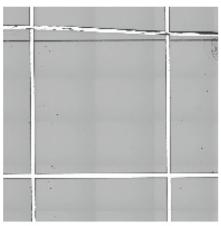
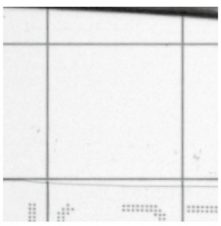
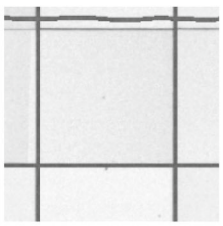
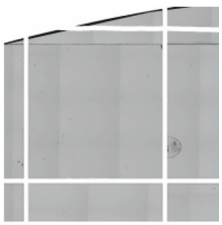
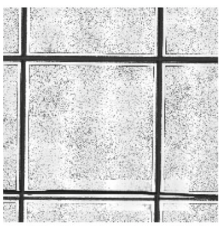
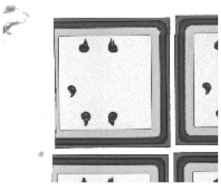
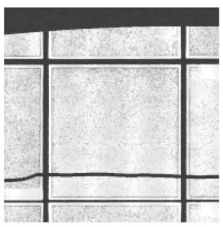
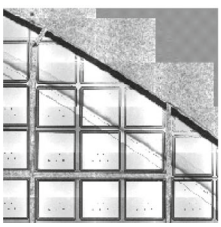
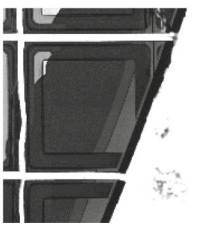
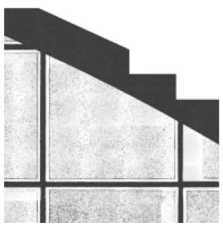
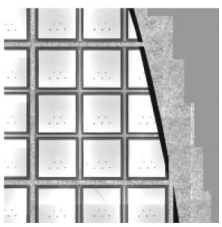
[6] TensorFlow project page, https://www.tensorflow.org/, version 2.1.

Table 1 Exemplary overview of flawless and faulty chip classes and our different wafer types within the underlying original and synthetic data sets

Chip class	Inside-flawless		Inside-faulty	
	Original	Synthetic	Original	Synthetic
Wafer type 1				
Wafer type 2				
Wafer type 3				
Wafer type 4				
Wafer type 5				

subsets. This simple preprocessing technique can help to alleviate some of the possible issues that ML and DL algorithms have with imbalanced data sets (Mohammed et al., 2020). Overall, the number of flawless chips was increased from 5963 to 6438, while the number of faulty chips was increased from 1026 to 6438 given the original data set. Moreover, the data set exhibits a notable disparity in its sample distribution across the different wafer types as detailed in Table 3. This disparity arises since certain wafer types, specifically types 1 and 3, are represented by multiple wafers within the data set. Wafer type 2, on the other hand, has a significantly higher chip count and therefore a lower original image resolution per chip sample. It is also noteworthy that wafer types 4 and 5 only consist of a quarter of a wafer instead of a full wafer. This characteristic further complicates the classification tasks associated with these particular wafer types.

Table 2 Exemplary overview of the five defect types within the underlying original and synthetic data sets

Defect type	Original data		Synthetic data	
Nose				
Chipping				
Oversize				
Undersize				
Border				

Subsequently, synthetic wafers were created based on the nature of the original data using our proposed data synthesis system. The resulting synthetic wafers were in turn used to create composite data sets containing original and generated data based on the ratios displayed in Table 3. In addition, various subsets of these data were created as shown in Table 5.

Experiment structure

To validate the effectiveness of the proposed data synthesis method, we conducted a series of experiments using various deep learning classifiers and data set variations. In the following, we provide an outline of these experiments:

Table 3 Our original and composite wafer data sets, for which the number of augmented samples is indicated in brackets, are shown

Wafer type	1	2	3	4	5	All
			Original wafer data			
Wafer resolution [pixel]	10,000 × 9868	10,240 × 10,316	12,945 × 12,999	7500 × 7500	14,000 × 12,532	–
# Wafers	2	1	4	1	1	9
Chips training	1691 (249)	4033 (639)	1843 (877)	168 (80)	604 (108)	8339 (1953)
Flawless	722 (248)	2336 (0)	1359 (1)	44 (80)	356 (0)	4817 (329)
Faulty	969 (1)	1697 (639)	484 (876)	124 (0)	248 (108)	3522 (1624)
Chips testing	422 (66)	1008 (160)	461 (223)	42 (22)	150 (30)	2083 (501)
Flawless	180 (64)	584 (0)	339 (3)	11 (21)	89 (1)	1203 (89)
Faulty	242 (2)	424 (160)	122 (220)	31 (1)	61 (29)	880 (412)
Chips total	2428	5840	3404	312	892	12,876
			Composite wafer data			
Chips training	6395 (907)	18,733 (3947)	6451 (2947)	600 (170)	3304 (606)	35,483 (8577)
Flawless	3403 (248)	11,340 (0)	4698 (1)	215 (170)	1955 (0)	21,611 (419)
Faulty	2992 (659)	7393 (3947)	1753 (2946)	385 (0)	1349 (606)	13,872 (8158)
Chips testing	422 (66)	1008 (160)	461 (223)	42 (22)	150 (30)	2083 (501)
Flawless	180 (64)	584 (0)	339 (3)	11 (21)	89 (1)	1203 (89)
Faulty	242 (2)	424 (160)	122 (220)	31 (1)	61 (29)	880 (412)
Chips total	7790	23,848	10,082	834	4,090	46,644

The composite data set includes all the original wafer data along with the synthetically generated samples. The synthetically generated data maintains the same flawless-to-faulty error ratios as the original wafers to enable the comparison of both data sets

1. **Baseline experiments.** Training on original data sets as well as composite data sets. In both cases, testing is exclusively performed on the original data. For data set balancing purposes, all baseline classification experiments are performed with balanced data sets via class-wise oversampling.
   (a) **Comprehensive model performance analysis.** For the purpose of gaining initial insights into suitable model types, we deployed 18 different deep learning models and evaluated their performance on both the original and composite data sets. The results are presented in Table 4 and Fig. 5.
   (b) **Investigation of individual wafer types.** To analyze the difficulties within our data set and evaluate the potential benefits of the proposed data synthesis method, we split the data set based on wafer type and assessed each of the five resulting subsets using *ResNet152V2* as classifier. The results are detailed in Table 5, Figs. 6, and 7.
   (c) **Comparison of augmentation methods.** In order to evaluate the progressiveness of our chosen approach to data synthesis, two different augmenters that utilize classical data augmentation methods are proposed, (i) a `simple` and (ii) a `complex` augmenter. (i) Our simple augmenter encompasses randomized affine transformations, including rotations, translations, and scaling. Additionally, image flipping is implemented. (ii) Our complex augmenter additionally encompasses image cropping and padding, further affine transformations such as image shearing, blurring, and sharpening filters, as well as hue and saturation filters. These augmentation operations are applied with different probabilities in randomized orders with the help of the image augmentation library imgaug (Jung, 2020a, b). The results are showcased in Table 6 and Fig. 9.
2. **LOOCV experiments.** Leave-one-out cross-validation (LOOCV) is deployed as a suitable method for the evaluation of our models' generalization capabilities (Wong, 2015). We differentiate between two different approaches to LOOCV as highlighted in Table 5.
   (a) **O-LOOCV experiments.** LOOCV with the original wafer data. Training is performed on different subsets of the original data, while testing is performed on the original data.
   (b) **SC-LOOCV experiments.** LOOCV with composite wafer data. We propose a novel synthetic-composite leave-one-out cross-validation method (SC-LOOCV) to analyze the quality of our synthesized data for each specific wafer type. LOOCV is extended by training on different subsets of original and synthetic data, while testing is performed on the original data.

Test results and evaluation

In the following sections, the findings of this contribution are presented starting with an overview of our test setup in "Test setup" section. Subsequently, our synthesized data set is introduced in "Synthesized data sets" section with a visual overview of defects in comparison to our related original data set. Following, our core findings are discussed in "Baseline classification experiments" and "LOOCV classification experiments" sections, including the evaluation of the classification results of various deep neural networks. Our proposed experiments are conducted by creating different data subsets encompassing: (i) a composite data set that is composed of the original data set that has been supplemented with synthetic data, and (ii) a leave-one-out cross-validation data set with subsets of the original data set and our composite data set.

Test setup

The following sections provide a brief overview of our test setup in terms of utilized software, hardware, and performance and evaluation metrics.

Software

To facilitate the deployment of Hexnet, the package management and deployment distribution Anaconda has been utilized, which in turn allows the distribution and setup of Hexnet on different operating systems. The created virtual environments are powered by Python version 3.7. All otherwise utilized dependencies are addressed in "Synthesis pipeline" and "Classification system" sections.

Hardware

Our hardware for evaluation as well as all related training and testing purposes includes (i) our CPU, "Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz" with 7200 BogoMips, (ii) our GPU, "TITAN RTX" with 24 GB VRAM, as well as (iii) our working memory with 128 GB of RAM at 2666 MHz.

$$I_{O,C} = \frac{1 - F_{1_O}}{1 - F_{1_C}} \tag{4}$$

where $I_{O,C}$ is the improvement factor, F_{1_O} is the F1-score of original data sets, F_{1_C} is the F1-score of composite data sets.

Performance metrics

Throughout the presented experiments, the F1-score was the selected main performance fitness and error metric of choice. While the F1-score is not the most suitable metric for evaluating imbalanced data sets (Bhowan et al., 2011), this issue was partially alleviated by oversampling and therefore creating balanced data sets between the flawless and faulty classes for each of the respective wafer types for all conducted experiments. The improvement factor $I_{O,C}$ (Schlosser et al., 2022, 2024a) for the assessment of the resulting classification capabilities is defined in Eq. 4. It calculates the relative improvement of a model's defect detection capabilities based on its F1-score, for which the results of two different classification experiments are compared with each other.

Synthesized data sets

To realize synthesized data sets in close alignment with the characteristics of their original data, our data synthesis system relies heavily on the usage of existing content. The underlying annotation task can be broken down to the cropping of chip templates and measurements of faulty dicing paths in the form of measurement points. In total, 245 chip templates and 78 faulty dicing paths consisting of 2274 measurement points were utilized from the original data set. Overall, the generated synthetic data set used in the presented experiments includes 23 generated wafers and 27,144 generated chips, where 16,794 chips are considered flawless and 10,350 faulty. Since a higher ratio of simulated faulty chips would have resulted in a class imbalance between flawless and faulty chips, we applied oversampling to the synthetic data set to reach a balance similar to that of the original data set, obtaining a total of 33,768 generated chips. Samples of the generated synthetic flawless and faulty chip classes are shown in Table 1.

Baseline classification experiments

Although the provided original data set was balanced by oversampling, the underlying issues were only partially alleviated. By increasing the total number of chips, we increased the total number of training samples from 10,292 chips within the original data set to 46,644 chips by combining the generated synthetic data set with the original data. An overview of the different restructured data sets is given in Table 5. For better comparability, the distributions of the respective test sets remain unchanged as in our previous experiments on the original data (Schlosser et al., 2022). Overall, the supplementation of the original data with synthesized data resulted in general improvements, whereas wafer types 2 and 5 resulted in minor changes in F1-score-based classification accuracy with relative improvement factors of 1.12 and 0.78, respectively. However, the classification capability on the data sets with wafer types 1, 3 and 4 increased significantly, with improvement factors ranging from 2.26 to 3.98. While all the individual wafer type data sets achieved improvement factors of $\geq$ 0.78, the combined data set encompassing all the wafer types also resulted in an overall improvement of 1.28 in terms of the relative improvement factor. As real-world application cases often show an imbalance between different types of wafers, our overall improvement incorporates this imbalance since a sizeable fraction of our data set originates from wafer type 2.

Additionally, Fig. 7a displays the relative improvement factors based on the true positive rate (TPR, also known as recall or sensitivity) and the true negative rate (TNR, also known as the specificity) (Schlosser et al., 2024a) for the 30 individual experiments conducted throughout our baseline experiments. These metrics are of particular interest for distinguishing the individual classification improvements for flawless and faulty samples of the realized data synthesis system. Overall, the mean TPR increased from $77.30 \pm 34.59\%$ for the original data set to $87.37 \pm 16.56\%$ for the composite data set, resulting in a relative improvement of 1.80. Consequently, the TNR increased slightly from 89.57 ± 10.52 to $90.43 \pm 6.09\%$, translating to a relative improvement of 1.28. The overall results in Table 5 show an increased mean F1-score of $81.04 \pm 14.87\%$ for the original data set to $88.95 \pm 7.96\%$ for the composite data set. Therefore, our baseline experiments confirm our observations regarding the viability of the synthesized data sets generated by the proposed system as a means of data augmentation for sparse and imbalanced data sets.

These results are furthermore qualitatively confirmed in Fig. 8, presenting gradient-weighted class activation mappings (Grad-CAM) (Selvaraju et al., 2016, 2017). Provided are the obtained class activation maps for our baseline classification experiments. These show improved activations for our composite wafer data when considering their general nature. A noteworthy observation can be seen when comparing both classes for both data sets: Overall, faulty chip CAMs focus on the dicing streets, whereas flawless chip CAMs tend to focus the chips' surfaces as well. Therefore, CAMs for our composite wafer data are more focused on the relevant image contents, which in turn can explain the improved model performance with our synthesized wafer data.

LOOCV classification experiments

Leave-one-out cross-validation can be seen as one of the more challenging tasks for learning-based approaches. Typically, leave-one-out cross-validation is performed by splitting a data set into n subsets. Each of the resulting subsets is used as a test set once, while the remaining $n - 1$ subsets are utilized for training as described in (Refaeilzadeh

Table 4 Model performance comparison for different DL-based approaches with our original data set and our composite data set

Model	Original wafer data	Composite wafer data	$I_{O,C}$
DenseNet121 (2017)	94.65 ± 0.39	**96.05 ± 0.45**	**1.35**
DenseNet169 (2017)	94.66 ± 0.21	**96.28 ± 0.44**	**1.44**
DenseNet201 (2017)	94.55 ± 0.66	**96.06 ± 0.32**	**1.38**
InceptionResNetV2 (2017)	94.90 ± 0.43	**95.53 ± 0.49**	**1.14**
InceptionV3 (2016)	95.07 ± 0.44	**95.70 ± 0.19**	**1.15**
MobileNet (2017)	92.22 ± 0.75	**94.73 ± 0.53**	**1.48**
MobileNetV2 (2018)	91.29 ± 0.34	**93.60 ± 0.85**	**1.36**
NASNetLarge (2018)	33.31 ± 0.05	**91.29 ± 1.06**	**7.66**
NASNetMobile (2018)	51.62 ± 22.48	**92.40 ± 1.00**	**6.37**
ResNet50 (2016a)	94.19 ± 0.33	**94.71 ± 0.56**	**1.10**
ResNet50V2 (2016b)	94.01 ± 0.31	**95.76 ± 0.20**	**1.41**
ResNet101 (2016a)	93.79 ± 0.35	**95.65 ± 0.40**	**1.43**
ResNet101V2 (2016b)	94.52 ± 0.51	**95.55 ± 0.49**	**1.23**
ResNet152 (2016a)	94.01 ± 0.45	**95.51 ± 0.24**	**1.33**
ResNet152V2 (2016b)	93.80 ± 0.36	**95.14 ± 0.47**	**1.28**
VGG16 (2015)	33.33 ± 0.00	33.33 ± 0.00	1.00
VGG19 (2015)	33.33 ± 0.00	33.33 ± 0.00	1.00
Xception (2017)	94.24 ± 0.51	**95.47 ± 0.43**	**1.27**

All results were measured in macro average F1-score [%] and averaged over 5 runs. Increased results are highlighted in bold. For a graphical visualization, see Fig. 5

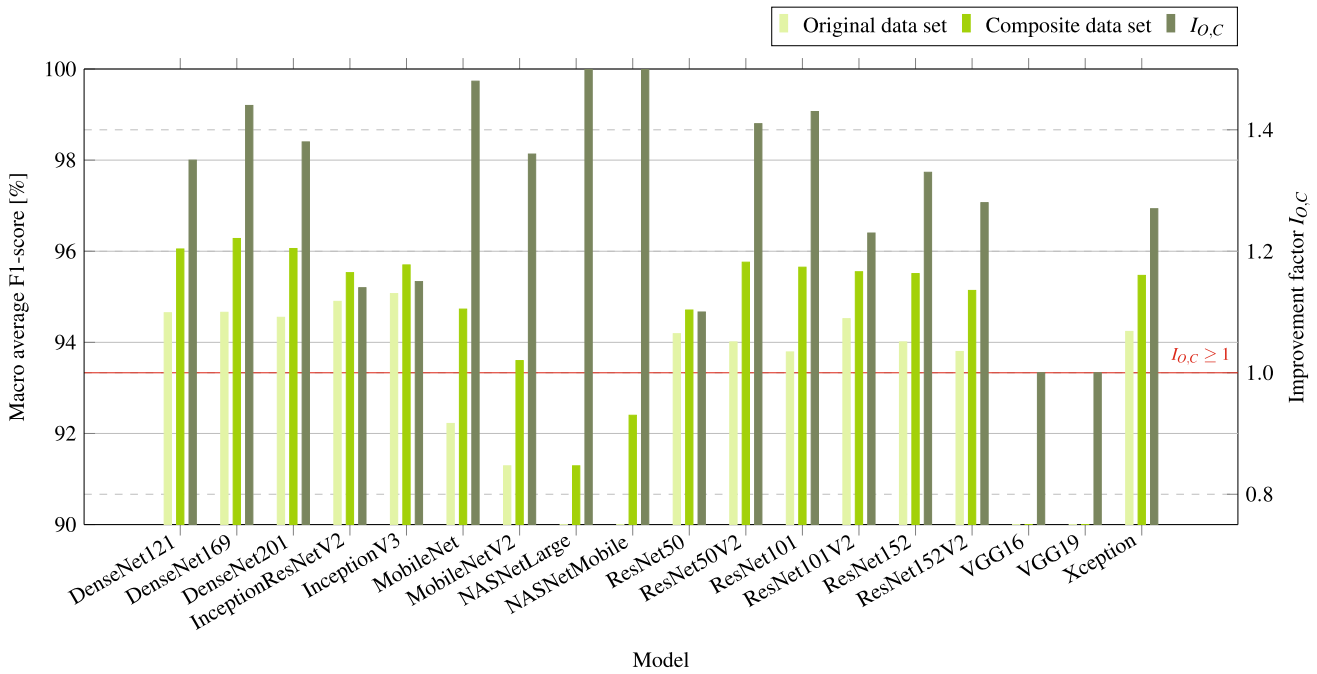


Fig. 5 Model performance visualization of Table 4. NASNetLarge and NASNetMobile show improvement factors of 7.66 and 6.37, respectively, which result from their strongly reduced classification capabilities on the original data set. In comparison, VGG16 and VGG19 performed very poorly on both data sets. The obtained improvement factors are limited to a range of $0.75 \leq I_{O,C} \leq 1.5$

et al., 2009). The general goal of LOOCV is to evaluate a model's classification capabilities on unknown data, which is a suitable application for our use case within the domain of semiconductor manufacturing as the internal chip structures change significantly between wafer types. However, as noted in "Contribution of this work" section, we perform LOOCV with a slight modification. While the principles of LOOCV remain the same by selecting one of our 5 available

Table 5 Classification results of our baseline and LOOCV classification experiments

Wafer type	Baseline		LOOCV		$I_{O,C}$	
	Original	Composite	O-LOOCV	SC-LOOCV	Baseline	LOOCV
1	84.18 ± 5.92	**92.99 ± 0.65**	50.28 ± 6.74	**88.68 ± 3.15**	**2.26**	**4.39**
2	92.68 ± 0.70	**93.49 ± 0.74**	**88.83 ± 0.86**	86.97 ± 1.62	**1.12**	0.86
3	84.94 ± 25.80	**96.22 ± 1.81**	37.69 ± 6.45	**89.60 ± 1.94**	**3.98**	**5.99**
4	49.38 ± 22.43	**79.98 ± 6.06**	37.94 ± 7.19	**67.77 ± 4.48**	**2.53**	**1.93**
5	**81.24 ± 2.87**	75.87 ± 11.64	**43.75 ± 9.36**	42.04 ± 4.64	0.78	0.97
All	93.80 ± 0.36	**95.14 ± 0.47**	–	–	**1.28**	–
Mean	81.04 ± 14.87	**88.95 ± 7.96**	51.70 ± 19.13	**75.01 ± 18.34**	**1.99 ± 1.08**	**2.83 ± 2.03**

Wafer type	Original			Composite		
	Train	Test	$\sum$	Train	Test	$\sum$
	Contained chips within the data subsets of the original baseline experiments					
1	1940	488	2428	7302	488	7790
2	4672	1168	5840	22,680	1168	23,848
3	2720	684	3404	9398	684	10,082
4	248	64	312	770	64	834
5	712	180	892	3910	180	4090
All	10,292	2584	12,876	44,060	2584	46,644

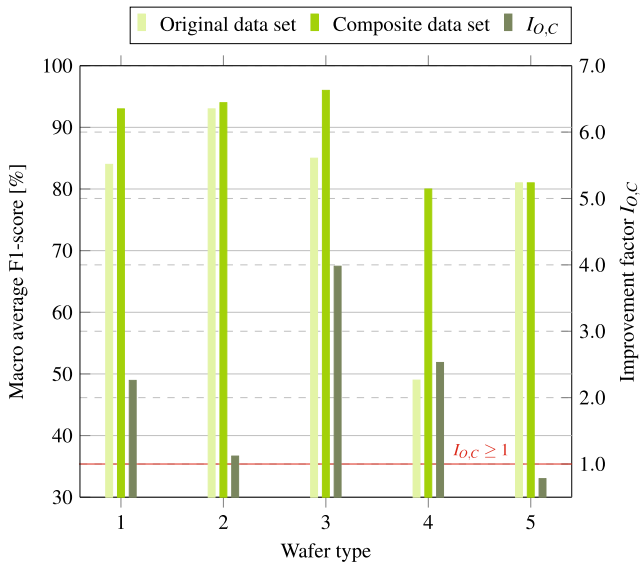
Wafer type	O-LOOCV			SC-LOOCV		
	Train	Test	$\sum$	Train	Test	$\sum$
	Contained chips within the data subsets of the LOOCV experiments					
1	10,448	2428	12,876	15,810	2428	7790
2	7036	5840	12,876	25,044	5840	30,884
3	9472	3404	12,876	16,150	3404	19,554
4	12,564	312	12,876	13,086	312	13,398
5	11,984	892	12,876	15,182	892	16,074

The original data are used as a baseline and compared to our composite data set, which consists of original and synthetic training data. In the last set of columns, the resulting improvement factor $I_{O,C}$ is derived. For all displayed experiments, the test sets consisted of original wafer data. All results were measured in macro average F1-score [%] and averaged over 5 runs. Increased results are highlighted in bold

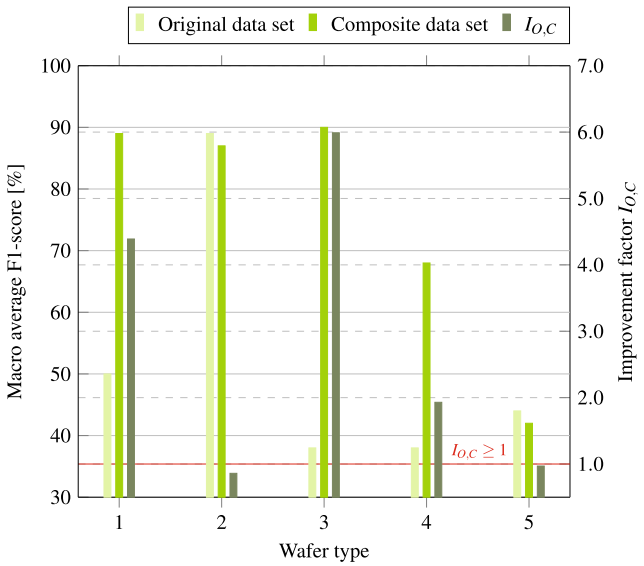
wafer types as our test set with the remaining 4 wafer types forming our training set (also called LOOCV with original wafer data, O-LOOCV), the training set is additionally supplemented with synthetic samples of the wafer type to be evaluated within the test set (also called synthetic-composite LOOCV, SC-LOOCV).

Our classification results are presented in Table 5. Initially, O-LOOCV was performed on the original wafer data. Overall, the obtained results show reduced scores on the original data in comparison to our baseline experiments, barely reaching an F1-score of 50%. The subset where the wafer types 1, 3, 4 and 5 were used for training and tested against wafer type 2 yielded improved results with an overall F1-score of 88.83±0.86%. This could be related to the fact that the internal chip structures of wafer type 2 appear to have a reduced complexity compared to those of other wafer types, therefore benefiting from the effects of domain randomization due to the diversity between the other wafers present in the training set (see also Table 1). In comparison, the data sets that were supplemented with synthetic wafers for the left-out original wafers resulted in an overall increase in performance. The wafer types 1 and 3 benefited the most from the supplementation with synthetic data, significantly increasing their respective overall improvement factors with 4.39 and 5.99. Wafer type 4 resulted in an increase of the overall performance by 1.93, whereas the wafer types 2 and 5 obtained a slight decrease in terms of classification capabilities.

The relative improvement factors presented based on the TPR and TNR for the 25 individual experiments conducted throughout our LOOCV experiments in Fig. 7b further illustrate the difficulties between the individual wafer types. Overall, the mean TPR increased from 27.76 ± 36.50% for



(a) Baseline (see also Table 5)



(b) LOOCV (see also Table 5)

Fig. 6 Model performance visualization of Table 5. Our results for our baseline and LOOCV experiments are shown. The obtained improvement factors are limited to a range of $0.5 \leq I_{O,C} \leq 7.0$. To provide an example for the calculation of the improvement factor given wafer type 1 in (**a**): According to Eq. 4, the F1-scores of the related original and composite data sets are used to calculate the respective relative improvement factor via $I_{O,C} = (1 - 0.8418)/(1 - 0.9299) \approx 2.26$

O-LOOCV to 64.24 ± 31.16% for SC-LOOCV, achieving a relative improvement of 2.02. However, the TNR slightly decreased from 94.08 ± 7.01 to 89.84 ± 2.69%. This can be explained by the classification system creating more false negatives in the initial assessment. Yet, the total number of false negatives could be decreased from 16,182 to 6341 samples with SC-LOOCV.

Finally, Table 5 shows a mean F1-score over all O-LOOCV experiments of 51.70 ± 19.13%, underlining the general difficulty of this experiment type with the original data set. In comparison, our proposed synthesis approach achieves a mean F1-score of 75.01 ± 18.34% with a relative improvement of 2.83 ± 2.03 by utilizing the resulting composite data set (SC-LOOCV). Therefore, a significant improvement in the overall classification capabilities of our system could be achieved. Future evaluations regarding the generation of synthetic data, however, should also assess the parameterization of our synthesis system regarding – yet unknown – characteristics within our original data set, for which further investigations will have to be conducted.

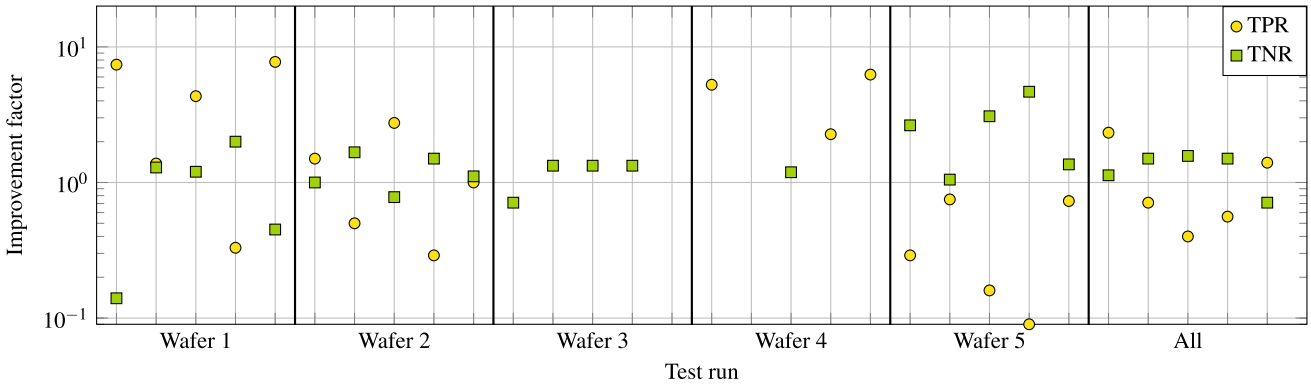
Discussion

In this section, we provide further insights regarding the progressiveness of our chosen approach as data augmentation method in "Progressiveness as data augmentation method" section, our models' runtime performance, benefits with synthetic data, and decision making regarding model selection in "Runtime performance", "Additionally observed benefits of synthetic data", and "Decisions regarding model selection" section, as well as our finally obtained wafer defect map visualizations in "Wafer defect map visualization" section.

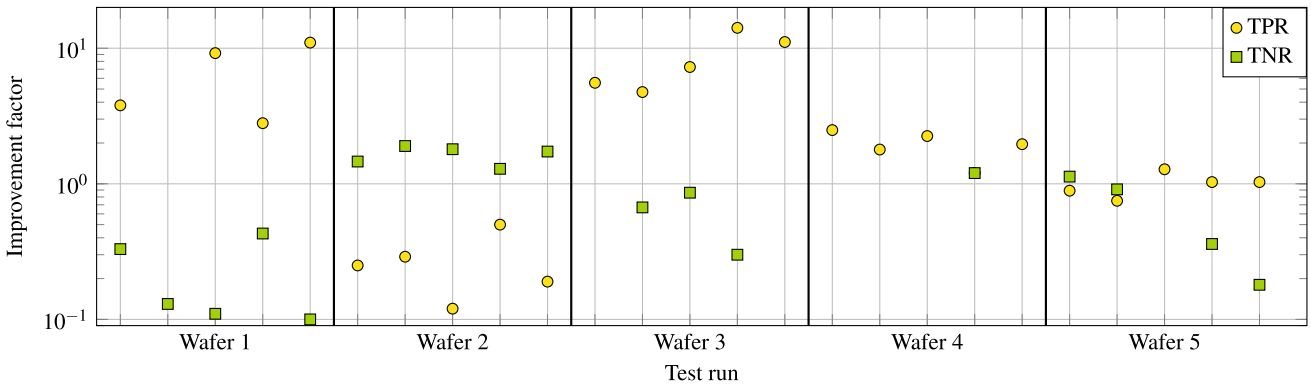
Progressiveness as data augmentation method

In order to evaluate the progressiveness of our proposed approach to the simulation of semiconductor wafer dicing induced fault on chips and their application as augmentation method for a deep learning based visual inspection system, further, conventional approaches to data augmentation are assessed. For this purpose, Table 6 and Fig. 9 give an overview of different classification results with our chosen classification backbone model, *ResNet152V2*. These results are separated into experiments with our original wafer data, conventional data augmentation methods that are commonly applied in image processing, and our composite wafer data.

For our augmenters, (i) `simple` and (ii) `complex`, the parameter `asi` $\in \{1, 2, 3, 4, 8\}$ denotes the augmentation size, whereby an `asi` of 2, for instance, denotes an augmentation of the original train set's size by factor 2. For (i), the parameter `ast` $\in \{1, 2, 3, 4, 8\}$ denotes the strength of the augmentation, which magnifies the application of the respective augmentation operations by the given factor. For

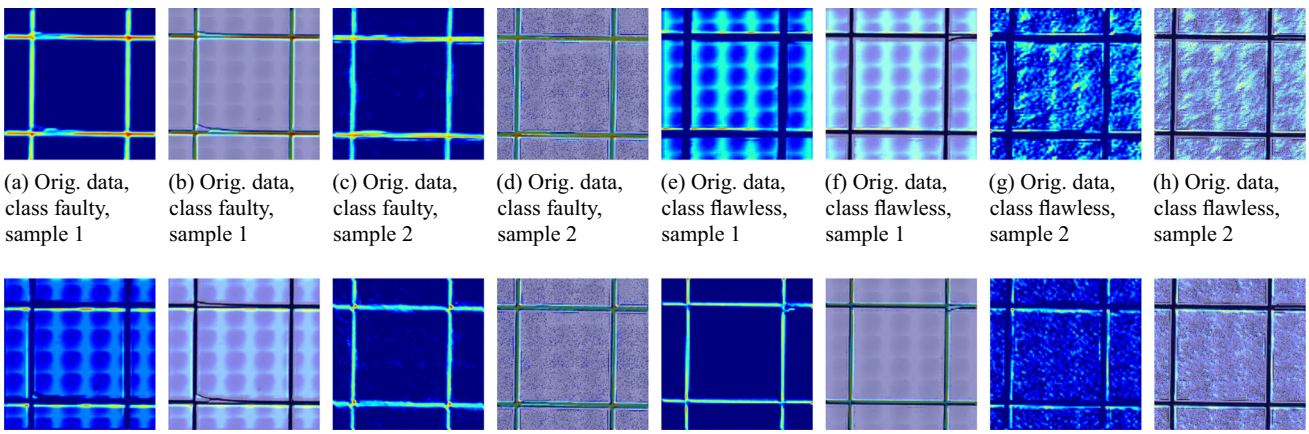


(a) Baseline test runs



(b) LOOCV test runs

Fig. 7 The improvement factors of our **a** baseline and **b** LOOCV classification experiments based on the true positive rate (TPR) and the true negative rate (TNR) are shown. For each of the five test runs, the order at which the samples were trained was randomized, whereas the training and the test sets remained otherwise unchanged



(a) Orig. data, class faulty, sample 1
(b) Orig. data, class faulty, sample 1
(c) Orig. data, class faulty, sample 2
(d) Orig. data, class faulty, sample 2
(e) Orig. data, class flawless, sample 1
(f) Orig. data, class flawless, sample 1
(g) Orig. data, class flawless, sample 2
(h) Orig. data, class flawless, sample 2
(i) Comp. data, class faulty, sample 1
(j) Comp. data, class faulty, sample 1
(k) Comp. data, class faulty, sample 2
(l) Comp. data, class faulty, sample 2
(m) Comp. data, class flawless, sample 1
(n) Comp. data, class flawless, sample 1
(o) Comp. data, class flawless, sample 2
(p) Comp. data, class flawless, sample 2

Fig. 8 Class activation map comparison with *ResNet152V2* after 50 epochs of training for wafer type 1. Row-wise, results for our original wafer data (top) and composite wafer data (bottom) are shown. Column-wise, class activation heatmaps and overlapped heatmaps (left and right, respectively) are depicted for the classes faulty and flawless with two different chip samples each

further information on the utilized augmentation operations, see Hexnet.[7]

By comparing the results obtained in Table 6 it is evident that our approach achieves the best result in comparison to our baseline data and their augmented versions, for which conventional augmentation operations have been employed. For (i,ii), increasing `asi` result in increased classification scores. Yet, increased train set sizes also mean `asi`-fold increased training times. For (i), increasing `ast` seem to have only a minor influence on model performance. Overall, a classification scores of 94.11-−94.22% (original wafer data), 90.38-−92.43% (conventional data augmentation), and 95.64% (composite wafer data) are obtained. These observations confirm the progressiveness of our chosen approach but also emphasize the domain-specific challenges present within our data set. Therefore, it can be concluded that the basic image manipulations employed by our simple and complex augmenters diverge from the real-world data distribution and do not adequately represent the test set.

Runtime performance

In addition to our model classification analysis with different DL-based approaches in Table 4, we provide a comprehensive runtime overview in Table 7 based on three critical metrics: the models' trainable parameters [m], their training time per sample, and their testing time per sample. Here, a key observation is the correlation between the number of trainable parameters and the models' training time per sample. With the exception of VGG16 (84.73 million trainable parameters) and VGG19 (112.32), an increased model complexity, i.e., the number of trainable parameters, also increases the training time required per sample for a given model. However, for our testing times, this trend is less noticeable. Only NASNetLarge (84.73 million trainable parameters) requires a significant increase in testing time per sample of 4.70 ms. For a potential deployment, the testing time per sample, also known as the inference time, is the most relevant metric following its classification accuracy. In our previous work (Schlosser et al., 2022), we determined a runtime requirement of 50 ms per image sample for a deployment in application. Based on Table 7, we therefore conclude that all our analyzed models meet this requirement, with none of them exceeding a testing time of 5 ms per sample, which would even enable the utilization of multiple models within one ensemble of models. Considering an inference time of 2.83 ms per chip for *ResNet152V2*, which was the primarily utilized model for most of our experiments, we can summarize that the classification time needed to process an entire wafer lies between 0.88 s for wafer type 4, representing the least amount of chips per wafer, and 16.53 s for wafer type 5, representing the most amount of chips per wafer, as detailed in Table 3.

Additionally observed benefits of synthetic data

Given the increased standard deviations of our original wafer data classification results vs. our composite wafer data classification results in Table 5 for both our baseline and our LOOCV experiments, it is evident that the problems of high variance and data overfitting can be alleviated. At the same time, this also enables the utilization of larger DNNs that do show problems when training on our original wafer data, such as NASNetLarge and NASNetMobile (Table 4). Our experiments highlight that our classification pipeline can be further adapted to previously unknown data with our data synthesis system, which depends on the characteristics of the underlying original data (see also differences in classification accuracies per wafer type in Table 5). Finally, with the provided Tables 5 (classification results) and 7 (model runtime comparison) an informed decision regarding model selection can be made.

Decisions regarding model selection

To assess the quality of our synthetic data, *ResNet152V2* was deployed as our main model for evaluation. It was used as a baseline for all conducted experiments to generate results comparable to the contribution of Schlosser et al. (2022). While *DenseNet201* and *InceptionResNetV2* show slightly increased mean F1-scores, *ResNet152V2* shows improved standard deviations, which justifies its use for our LOOCV classification experiments following our baseline classification experiments.

Wafer defect map visualization

In Fig. 10, a qualitative evaluation of our LOOCV experiments is provided, which shows the resulting visualized wafer maps with ground truth visualizations (left column), O-LOOCV classification result visualizations (middle column), and SC-LOOCV classification result visualizations (right column). For the ground truth, flawless (•) and faulty chips (•) are visualized. For the classification results, correct (•) and incorrect chip classifications (•) are visualized for our O-LOOCV and SC-LOOCV experiments. Subsequently, the finally obtained wafer defect map visualizations can enable easier and more efficient (semi-)automated assessments by an inspector.

[7] Hexnet project page: augmenters.py: Dataset Augmenters, https://github.com/TSchlosser13/Hexnet/blob/master/_ML/misc/augmenters.py.

Table 6 Model performance comparison for *ResNet152V2* (He et al., 2016b) with different data sets to evaluate the progressiveness of different approaches to data augmentation (`asi` = augmentation size, `ast` = augmentation strength)

Data set	Macro average F1-score [%]
Original wafer data	
No augmentation	94.22 ± 0.45
Oversampling	94.11 ± 0.40
Conventional data augmentation	
Simple augmenter (`asi` = 1, `ast` = 1)	91.09 ± 1.77
Simple augmenter (`asi` = 1, `ast` = 2)	90.58 ± 0.99
Simple augmenter (`asi` = 1, `ast` = 3)	90.38 ± 1.22
Simple augmenter (`asi` = 2, `ast` = 1)	91.37 ± 1.25
Simple augmenter (`asi` = 2, `ast` = 2)	92.39 ± 0.35
Simple augmenter (`asi` = 2, `ast` = 3)	91.98 ± 0.57
Simple augmenter (`asi` = 3, `ast` = 1)	91.93 ± 0.88
Simple augmenter (`asi` = 3, `ast` = 2)	92.09 ± 0.50
Simple augmenter (`asi` = 3, `ast` = 3)	92.01 ± 0.36
Simple augmenter (`asi` = 4, `ast` = 4)	92.43 ± 0.43
Simple augmenter (`asi` = 8, `ast` = 8)	91.99 ± 0.54
Complex augmenter (`asi` = 1)	89.54 ± 0.87
Complex augmenter (`asi` = 2)	91.20 ± 0.55
Complex augmenter (`asi` = 3)	91.37 ± 0.94
Complex augmenter (`asi` = 4)	91.99 ± 0.43
Complex augmenter (`asi` = 8)	93.65 ± 0.13
Composite wafer data	
Our approach	$\mathbf{95.64 \pm 0.23}$

All results were measured in macro average F1-score [%] and averaged over 5 runs. The best result is highlighted in bold. For a graphical visualization, see Fig. 9

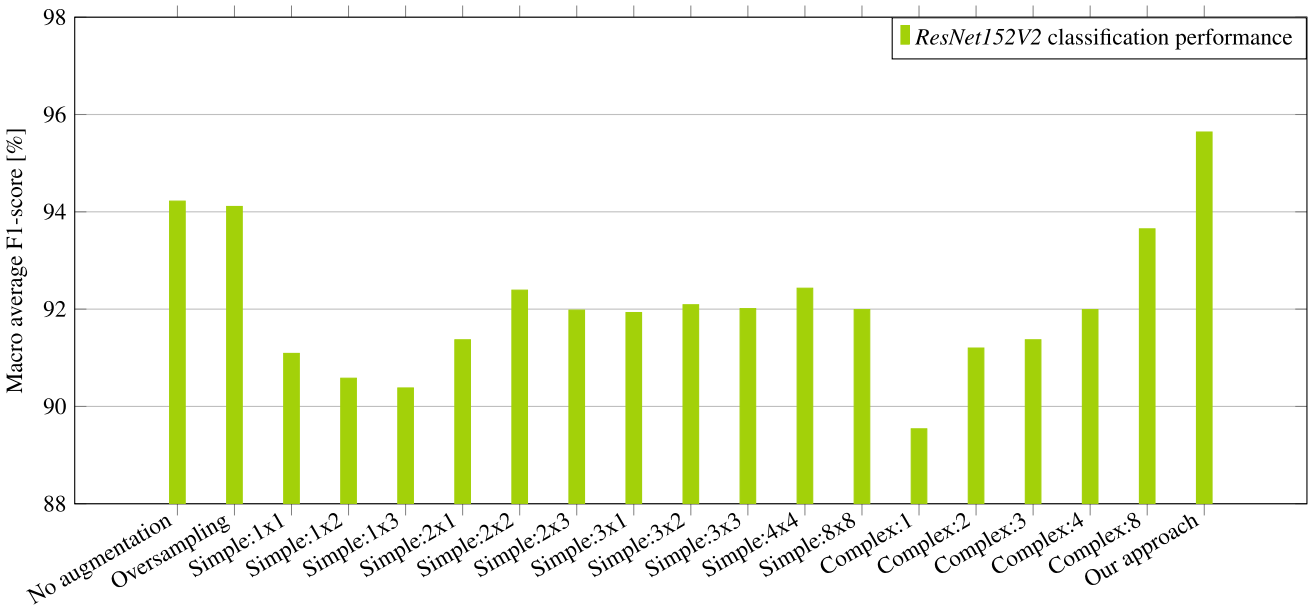
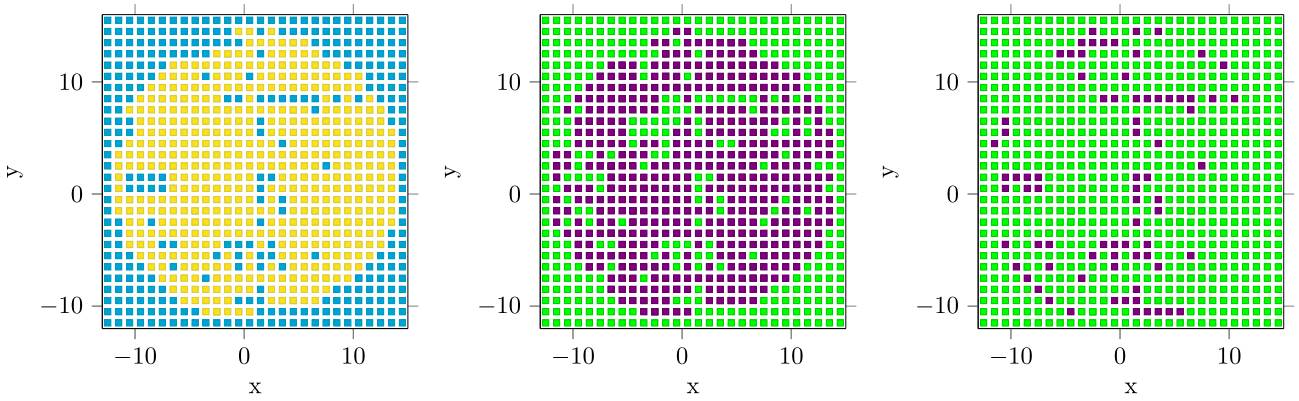
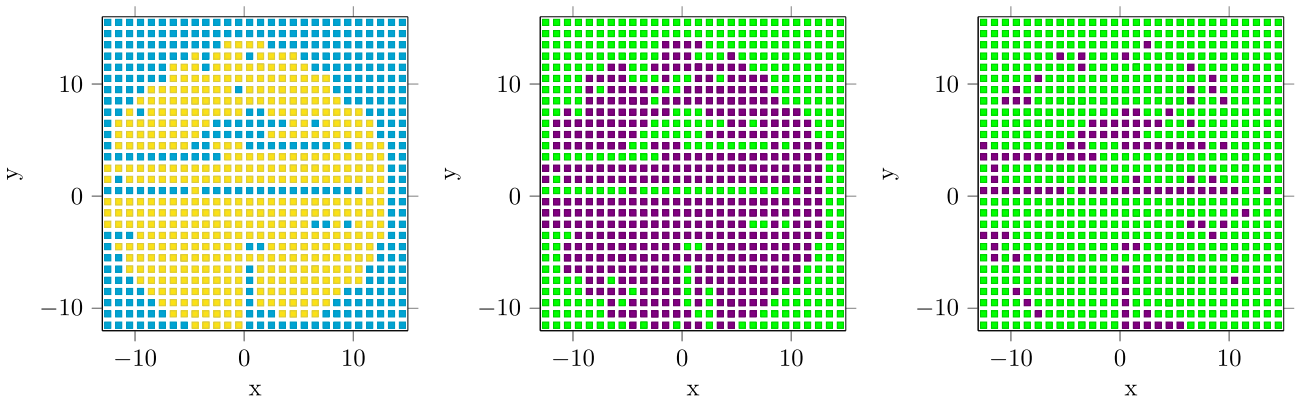


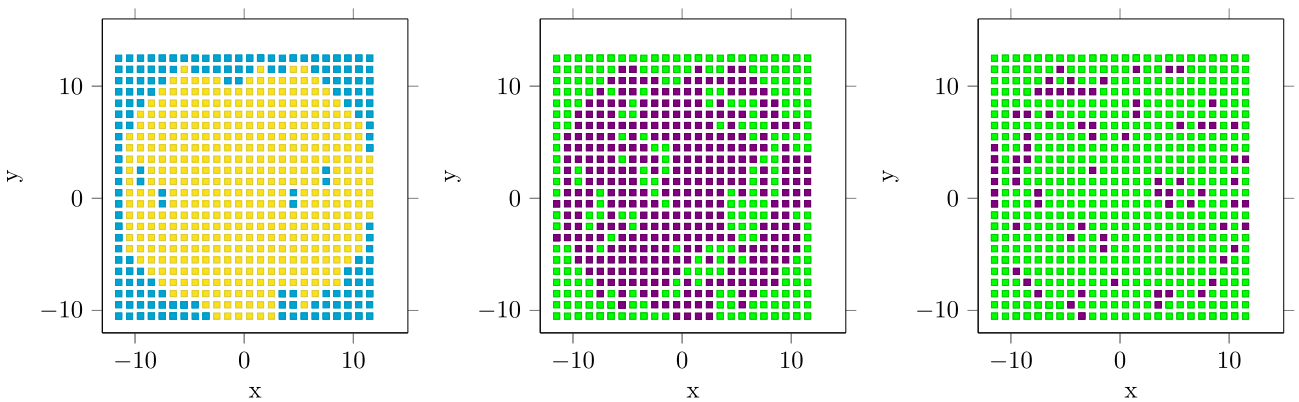
Fig. 9 Model performance comparison for *ResNet152V2* (He et al., 2016b) with different data sets to evaluate the progressiveness of different approaches to data augmentation. For our augmenters, (i) `simple` and (ii) `complex`, the augmentation size `asi` and the augmentation strength `ast` are denoted as follows: (i) "Simple:`asixast`" and (ii) "Complex:`asi`". All results were measured in macro average F1-score [%] and averaged over 5 runs



(a) Wafer 1-1: ground truth (b) Wafer 1-1: O-LOOCV classification (c) Wafer 1-1: SC-LOOCV classification



(d) Wafer 1-2: ground truth (e) Wafer 1-2: O-LOOCV classification (f) Wafer 1-2: SC-LOOCV classification



(g) Wafer 3-1: ground truth (h) Wafer 3-1: O-LOOCV classification (i) Wafer 3-1: SC-LOOCV classification

Fig. 10 Visualized wafer maps for our qualitative evaluation with exemplary LOOCV experiment results for wafer types 1 and 3 with *ResNet152V2* (denotation: <wafer type>-<wafer number>). The resulting visualized wafer maps with their ground truth visualizations (left column), O-LOOCV classification result visualizations (middle column), and SC-LOOCV classification result visualizations (right column) are shown. For the ground truth, flawless (•) and faulty chips (•) are visualized. For the classification results, correct (•) and incorrect chip classifications (•) are visualized (Color figure online)

Table 7 Model runtime comparison with trainable parameters as well as training and testing times per sample

Model	Trainable parameters [m]	Training time/sample [ms]	Testing time/sample [ms]
DenseNet121 (2017)	6.96	3.15	1.64
DenseNet169 (2017)	12.49	3.92	2.13
DenseNet201 (2017)	18.10	5.00	2.64
InceptionResNetV2 (2017)	54.28	5.12	2.80
InceptionV3 (2016)	21.77	2.15	1.31
MobileNet (2017)	3.21	2.19	0.64
MobileNetV2 (2018)	2.23	2.40	0.69
NASNetLarge (2018)	84.73	15.45	4.70
NASNetMobile (2018)	4.24	4.85	1.73
ResNet50 (2016a)	23.54	2.80	1.26
ResNet50V2 (2016b)	23.52	2.43	1.19
ResNet101 (2016a)	42.56	4.54	1.94
ResNet101V2 (2016b)	42.53	4.20	1.99
ResNet152 (2016a)	58.22	6.45	2.74
ResNet152V2 (2016b)	58.19	6.06	2.83
VGG16 (2015)	107.01	3.89	1.58
VGG19 (2015)	112.32	4.48	1.77
Xception (2017)	20.81	5.73	1.34

Conclusion and outlook

In this contribution, our proposed and developed data synthesis system was evaluated for its ability to generate wafer imagery with close to real-world resemblance, encompassing high-resolution wafer, chip, and street images and annotations. The synthesis system can be deployed to create labeled synthetic data sets of unknown or underrepresented wafer (and chip) types and defect patterns.

A data set consisting of 33,768 synthetic chips was created and evaluated based on its classification capabilities given our visual inspection system, which is based on different learning-based approaches from machine learning, such as deep neural networks. Two experiments were conducted on our synthetic data set. For our initial baseline experiments, our synthetic data were utilized to supplement the training data, otherwise consisting of original chip imagery. The relative classification capabilities were significantly increased by up to 3.98 times on the individual wafer type data subsets, whereas an F1-score of 95.14% was obtained on our data set encompassing all wafer types. For the second experiment, a variation of leave-one-out cross-validation (LOOCV) was introduced for the purpose of validating the quality of our generated synthetic data sets. The underlying idea was to evaluate the classification capabilities of an existing visual inspection setup on a previously unknown type of wafer where no labeled data with only a few faulty samples are present. The left-out wafer type of the LOOCV experiments was supplemented with synthetic data for the training set. In this scenario, the relative classification capabilities of the underlying model were significantly increased by up to 5.99 times, showing an overall improvement of 2.83 times over all wafer types.

Consequently, our results show that the classification capabilities of the utilized DL algorithms can be consistently improved by supplementing the training data sets with synthetic data without changing the nature or complexity of the classifiers used. Furthermore, the experiments detailed in Table 4 imply that the augmentation of the training set with synthetic data enables the application of more sophisticated DL algorithms. Additionally, Table 4, in conjunction with Table 7, serves as a guideline for selecting a suitable trade-off between runtime and classification accuracy.

Subsequently, this work will serve as a baseline for future contributions with application-specific fine-tuning, for which novel learning-based approaches to image generation and classification have yet to be investigated within the context of (semi-)automated visual inspection. Therefore, future contributions should also investigate approaches to improve our data synthesis and classification process, for which class activations could be further utilized to substantiate the explainability of our realized system. For this purpose, current approaches to image classification, such as vision transformers (Dosovitskiy et al., 2021), including, i.a., data-efficient image transformers (DeiT) and hybrid transformers (Han et al., 2021; Touvron et al., 2021a, b), as well as image generation, such as generative adversarial networks (Gui et

al., 2021) and diffusion models (Cao et al., 2024), could be leveraged.

While the proposed data synthesis system is currently limited to being a domain-specific solution, it offers the significant advantage of synthesizing entire wafers. Achieving this with deep learning based generative methods (Gui et al., 2021) alone would be challenging due to the high resolution of the underlying images and low sample count (Table 3). Additionally, the synthesis system supports a broad range of simulation parameters, facilitating the exploration of solutions in the domain of meta-learning (Shorten & Khoshgoftaar, 2019). However, the importance of preserving plausible neighbor relationships between chips and streets has not yet been fully explored and warrants further investigation, particularly in comparison with generative approaches. Subsequently, the system's optical components, including parameters such as camera and lighting configurations, will require thorough examination to enable a precise application-specific fine-tuning (Schlosser et al., 2022).

Acknowledgements The European Union, through the European Social Fund for Germany, partially funded this research under grant number 100670286.

Author contributions Michael Friedrich and Tobias Schlosser conducted this contribution's writing process and evaluation with the help of Danny Kowerko in realizing this manuscript. The experiments performed and the associated implementations were carried out by Michael Friedrich under the supervision of Tobias Schlosser and Danny Kowerko.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability Not applicable.

Code availability *The Hexagonal Image Processing Framework Hexnet* is available via its project page and code repository under URL: https://github.com/TSchlosser13/Hexnet.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

Bansal, M. A., Sharma, D. R., & Kathuria, D. M. (2022). A systematic review on data scarcity problem in deep learning: Solution and applications. *ACM Computing Surveys (CSUR), 54*(10s), 1–29. https://doi.org/10.1145/350228787

Beuth, F., Schlosser, T., Friedrich, M., & Kowerko, D. (2020). Improving automated visual fault detection by combining a biologically plausible model of visual attention with deep learning. In *IECON 2020 The 46th annual conference of the IEEE Industrial Electronics Society* (pp. 5323–5330). IEEE. https://doi.org/10.1109/IECON43393.2020.9255234, https://ieeexplore.ieee.org/document/9255234.

Bhowan, U., Johnston, M., & Zhang, M. (2011). Developing new fitness functions in genetic programming for classification with unbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics), 42*(2), 406–421. https://doi.org/10.1109/TSMCB.2011.2167144

Cao, H., Tan, C., Gao, Z., Xu, Y., Chen, G., Heng, P. A., & Li, S. Z. (2024). A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering, 36*(7), 2814–2830. https://doi.org/10.1109/TKDE.2024.3361474

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering, 33*(9), 731–747. https://doi.org/10.1111/mice.12334

Chen, Z., Li, C., & Sanchez, R. V. (2015). Gearbox fault identification and classification with convolutional neural networks. *Shock and Vibration, 1*, 390134. https://doi.org/10.1155/2015/390134

Chien, J. C., Wu, M. T., & Lee, J. D. (2020). Inspection and classification of semiconductor wafer surface defects using CNN deep learning networks. *Applied Sciences, 10*(15), 5340. https://doi.org/10.3390/app10155340

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1800–1807). IEEE. https://doi.org/10.1109/CVPR.2017.195. https://ieeexplore.ieee.org/document/8099678

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2018). AutoAugment: Learning augmentation policies from data. *Computing Research Repository (CoRR)*. https://doi.org/10.48550/arXiv.1805.09501, https://arxiv.org/abs/1805.09501

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations (ICLR)*. https://doi.org/10.48550/arXiv.2010.11929

Ekbatani, H. K., Pujol, O., & Segui, S. (2017). Synthetic data generation for deep learning in counting pedestrians. In *Proceedings of the 6th international conference on pattern recognition applications and methods—ICPRAM* (pp. 318–323). SciTePress. https://doi.org/10.5220/0006119203180323. https://www.scitepress.org/Link.aspx?doi=10.5220/0006119203180323

Fotouhi, S., Pashmforoush, F., Bodaghi, M., & Fotouhi, M. (2021). Autonomous damage recognition in visual inspection of laminated composite structures using deep learning. *Composite Structures, 268*, 113960. https://doi.org/10.1016/j.compstruct.2021.113960

Gatys, L. A. (2015). A neural algorithm of artistic style. *Computing Research Repository (CoRR)*. https://doi.org/10.48550/arXiv.1508.06576

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural informa-*

tion processing systems 27 (NIPS 2014) (Vol. 27, pp. 1–9). https://proceedings.neurips.cc/paper_files/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering, 35*(4), 3313–3332. https://doi.org/10.1109/TKDE.2021.3130191

Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2315–2324). IEEE. https://doi.org/10.1109/CVPR.2016.254. https://ieeexplore.ieee.org/document/7780623

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. In *35th International conference on neural information processing systems (NIPS'21)* (Vol. 34, pp. 15908–15919). https://doi.org/10.48550/arXiv.2103.00112. https://dl.acm.org/doi/abs/10.5555/3540261.3541478

He, K., Zhang, X., Ren, S., & Sun, J.(2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90. https://ieeexplore.ieee.org/document/7780459

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision* (pp. 630–645). Springer. https://doi.org/10.1007/978-3-319-46493-0_38

Hooper, A., Ehorn, J., Brand, M., & Bassett, C. (2015). Review of wafer dicing techniques for via-middle process 3DI/TSV ultrathin silicon device wafers. In *2015 IEEE 65th electronic components and technology conference (ECTC)* (pp. 1436–1446). IEEE. https://doi.org/10.1109/ECTC.2015.7159786. https://ieeexplore.ieee.org/document/7159786

Howard, A., G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *Computing Research Repository (CoRR)*. https://doi.org/10.48550/arXiv.1704.04861

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2261–2269). IEEE. https://doi.org/10.1109/CVPR.2017.243. https://ieeexplore.ieee.org/document/8099726

Huang, S. H., & Pan, Y. C. (2015). Automated visual inspection in the semiconductor industry: A survey. *Computers in Industry, 66*, 1–10. https://doi.org/10.1016/j.compind.2014.10.006

Imoto, K., Nakai, T., Ike, T., Haruki, K., & Sato, Y. (2019). A CNN-based transfer learning method for defect classification in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 32*(4), 455–459. https://doi.org/10.1109/TSM.2019.2941752

Inoue, H. (2018). Data augmentation by pairing samples for images classification. *Computing Research Repository (CoRR)*. https://doi.org/10.48550/arXiv.1801.02929

Jung, A. (2020a). imgaug Documentation. Tech. Rep. https://readthedocs.org/projects/imgaug/downloads/pdf/stable/

Jung, A. (2020b). imgaug Repository. https://github.com/aleju/imgaug

Lee, K. B., Cheon, S., & Kim, C. O. (2017). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing, 30*(2), 135–142. https://doi.org/10.1109/TSM.2017.2676245

Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Smart augmentation learning an optimal data augmentation strategy. *IEEE Access, 5*, 5858–5869. https://doi.org/10.1109/ACCESS.2017.2696121

Lin, S., He, Z., & Sun, L. (2023). A novel micro-defect classification system based on attention enhancement. *Journal of Intelligent Manufacturing, 35*(2), 703–726. https://doi.org/10.1007/s10845-022-02064-2

Maksim, K., Kirill, B., Eduard, Z., Nikita, G., Aleksandr, B., Arina, L., Vladislav, S., Daniil, M., & Nikolay, K. (2019). Classification of wafer maps defect based on deep learning methods with small amount of data. In *2019 International conference on engineering and telecommunication (EnT)* (pp. 1–5). IEEE. https://doi.org/10.1109/EnT47717.2019.9030550, https://ieeexplore.ieee.org/document/9030550.

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)* (pp. 243–248). IEEE. https://doi.org/10.1109/ICICS49469.2020.239556, https://ieeexplore.ieee.org/document/9078901.

Nikolenko, S. I. (2021). *Synthetic data for deep learning* (Vol. 174). Springer. https://doi.org/10.1007/978-3-030-75178-4

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *Computing Research Repository (CoRR), 11*(2017), 1–8. https://doi.org/10.48550/arXiv.1712.04621

Peres, R. S., Guedes, M., Miranda, F., & Barata, J. (2021). Simulation-based data augmentation for the quality inspection of structural adhesive with deep learning. *IEEE Access, 9*, 76532–76541. https://doi.org/10.1109/ACCESS.2021.3082690

Rahim, K., & Mian, A. (2017). A review on laser processing in electronic and MEMS packaging. *Journal of Electronic Packaging, 139*(3), 030801. https://doi.org/10.1115/1.4036239

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems* (pp. 532–538). https://doi.org/10.1007/978-0-387-39940-9_565

Ruppert, D., & Carroll, R. J. (2000). Theory & methods: Spatially-adaptive penalties for spline fitting. *Australian & New Zealand Journal of Statistics, 42*(2), 205–223. https://doi.org/10.1111/1467-842X.00119

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2018). MobileNetV2: inverted residuals and linear bottlenecks. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 4510–4520). IEEE. https://doi.org/10.1109/CVPR.2018.00474. https://ieeexplore.ieee.org/document/8578572

Saqlain, M., Abbas, Q., & Lee, J. Y. (2020). A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing, 33*(3), 436–444. https://doi.org/10.1109/TSM.2020.2994357

Schlosser, T., Beuth, F., Friedrich, M., & Kowerko, D. (2019). A novel visual fault detection and classification system for semiconductor manufacturing using stacked hybrid convolutional neural networks. In *2019 24th IEEE international conference on emerging technologies and factory automation (ETFA)* (pp. 1511–1514). IEEE. https://doi.org/10.1109/ETFA.2019.8869311, https://ieeexplore.ieee.org/document/8869311

Schlosser, T., Friedrich, M., Beuth, F., & Kowerko, D. (2022). Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks. *Journal of Intelligent Manufacturing, 33*, 1099–1123. https://doi.org/10.1007/s10845-021-01906-9

Schlosser, T., Friedrich, M., Meyer, T., & Kowerko, D. (2024a). A consolidated overview of evaluation and performance metrics for machine learning and computer vision. https://doi.org/10.13140/RG.2.2.14331.69928. https://www.researchgate.net/publication/374558675_A_Consolidated_Overview_of_Evaluation_and_Performance_Metrics_for_Machine_Learning_and_Computer_Vision.

Schlosser, T., Friedrich, M., Meyer, T., Kowerko, D., & Eibl, M. (2024b). Biologically inspired hexagonal deep learning for

hexagonal image processing with the hexagonal image processing framework Hexnet. https://www.researchgate.net/publication/363491206_Biologically_Inspired_Hexagonal_Deep_Learning_for_Hexagonal_Image_Processing_With_The_Hexagonal_Image_Processing_Framework_Hexnet

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2016). Grad-CAM: Why did you say that? arXiv preprint (pp. 1–4). arXiv:1611.07450

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: visual explanations from deep networks via gradient-based localization. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 618–626). IEEE. https://doi.org/10.1109/ICCV.2017.74. https://ieeexplore.ieee.org/document/8237336

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1), 1–48. https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations* (pp. 1–14). https://doi.org/10.48550/arXiv.1409.1556

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2818–2826). IEEE. https://doi.org/10.1109/CVPR.2016.308. https://ieeexplore.ieee.org/document/7780677

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Thirty-First AAAI conference on artificial intelligence*. https://doi.org/10.1609/aaai.v31i1.11231. https://ojs.aaai.org/index.php/AAAI/article/view/11231

Tanaka, F. H. K. D. S., & Aranha, C. (2019). Data augmentation using GANs. *Computing Research Repository (CoRR)*. https://doi.org/10.48550/arXiv.1904.09135. https://arxiv.org/abs/1904.09135

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 23–30). IEEE. https://doi.org/10.1109/IROS.2017.8202133. https://ieeexplore.ieee.org/document/8202133

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021a). Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th international conference on machine learning, PMLR* (pp. 10347–10357). https://doi.org/10.48550/arXiv.2012.12877. https://proceedings.mlr.press/v139/touvron21a.html

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., & Jégou, H. (2021b). Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision (pp. 32–42)*. IEEE. https://doi.org/10.48550/arXiv.2103.17239. https://openaccess.thecvf.com/content/ICCV2021/html/Touvron_Going_Deeper_With_Image_Transformers_ICCV_2021_paper.html

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., & Birchfield, S. (2018). Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) workshops* (pp. 969–977). IEEE. https://doi.org/10.1109/CVPRW.2018.00143. https://www.computer.org/csdl/proceedings-article/cvprw/2018/610000b082/17D45WUj90D

Tripathi, S., Chandra, S., Agrawal, A., Tyagi, A., Rehg, J. M., & Chari, V. (2019). Learning to generate synthetic data via compositing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 461–470). IEEE. https://doi.org/10.1109/CVPR.2019.00055. https://ieeexplore.ieee.org/document/8953554

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Wold, S. (1974). Spline functions in data analysis. *Technometrics, 16*(1), 1–11. https://doi.org/10.2307/1267485

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding data augmentation for classification: When to warp? In *2016 International conference on digital image computing: Techniques and applications (DICTA)* (pp. 1–6). IEEE. https://doi.org/10.1109/DICTA.2016.7797091. https://ieeexplore.ieee.org/document/7797091

Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition, 48*(9), 2839–2846. https://doi.org/10.1016/j.patcog.2015.03.009

Wu, M. J., Jang, J. S. R., & Chen, J. L. (2014). Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing, 28*(1), 1–12. https://doi.org/10.1109/TSM.2014.2364237

Zheng, X., Zheng, S., Kong, Y., & Chen, J. (2021). Recent advances in surface defect inspection of industrial products using deep learning techniques. *The International Journal of Advanced Manufacturing Technology, 113*, 35–58. https://doi.org/10.1007/s00170-021-06592-8

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, 34*, 13001–13008. https://doi.org/10.1609/aaai.v34i07.7000

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 8697–8710). IEEE. https://doi.org/10.1109/CVPR.2018.00907. https://ieeexplore.ieee.org/document/8579005