Schlosser, Tobias; Friedrich, Michael; Beuth, Frederik; Kowerko, Danny

Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks

Journal of Intelligent Manufacturing

Provided in Cooperation with:
Springer Nature

Suggested Citation: Schlosser, Tobias; Friedrich, Michael; Beuth, Frederik; Kowerko, Danny (2022) : Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks, Journal of Intelligent Manufacturing, ISSN 1572-8145, Springer US, New York, NY, Vol. 33, Iss. 4, pp. 1099-1123, https://doi.org/10.1007/s10845-021-01906-9

Journal of Intelligent Manufacturing (2022) 33:1099–1123
https://doi.org/10.1007/s10845-021-01906-9

Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks

Tobias Schlosser[1] · Michael Friedrich[1] · Frederik Beuth[1] · Danny Kowerko[1]

Received: 20 April 2021 / Accepted: 24 December 2021 / Published online: 25 January 2022

Abstract
In the semiconductor industry, automated visual inspection aims to improve the detection and recognition of manufacturing defects by leveraging the power of artificial intelligence and computer vision systems, enabling manufacturers to profit from an increased yield and reduced manufacturing costs. Previous domain-specific contributions often utilized classical computer vision approaches, whereas more novel systems deploy deep learning based ones. However, a persistent problem in the domain stems from the recognition of very small defect patterns which are often in the size of only a few μm and pixels within vast amounts of high-resolution imagery. While these defect patterns occur on the significantly larger wafer surface, classical machine and deep learning solutions have problems in dealing with the complexity of this challenge. This contribution introduces a novel hybrid multistage system of stacked deep neural networks (SH-DNN) which allows the localization of the finest structures within pixel size via a classical computer vision pipeline, while the classification process is realized by deep neural networks. The proposed system draws the focus over the level of detail from its structures to more task-relevant areas of interest. As the created test environment shows, our SH-DNN-based multistage system surpasses current approaches of learning-based automated visual inspection. The system reaches a performance (F1-score) of up to 99.5%, corresponding to a relative improvement of the system's fault detection capabilities by 8.6-fold. Moreover, by specifically selecting models for the given manufacturing chain, runtime constraints are satisfied while improving the detection capabilities of currently deployed approaches.

Keywords Computer vision · Pattern and image recognition · Deep learning · Semiconductor manufacturing · Factory automation · Fault inspection

Introduction and motivation

Automated visual fault inspection processes involve the development and integration of systems for capturing and monitoring of manufacturing results. The related manufacturing processes include a multitude of complex processing steps, whereas one of these processing steps is concerned with the separation of the resulting components (Huang and Pan 2015). While incorporating the analysis of the used materials as well as the imaging of the investigated circuits, the utilized mechanical forces induced by the laser cutting process characterize the quality of the resulting components. The laser cutting process itself is defined by the prevailing temperature, pressure, and voltage values, whereas information about the amount of flawless chips is ultimately derived from the nature of the resulting components. The aim is therefore to identify potential manufacturing defects within the manufacturing chain, but also to determine influential factors while avoiding complications in subsequent processing steps. For this purpose, an important quality property is calculated based on the ratio of flawless to total chips, also called yield (Lee et al. 2017). Since a manual inspection can imply a considerable and in particular exhaustive time expenditure,

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

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

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

Frederik Beuth
frederik.beuth@cs.tu-chemnitz.de

[1] Junior Professorship of Media Computing, Chemnitz University of Technology, 09107 Chemnitz, Germany

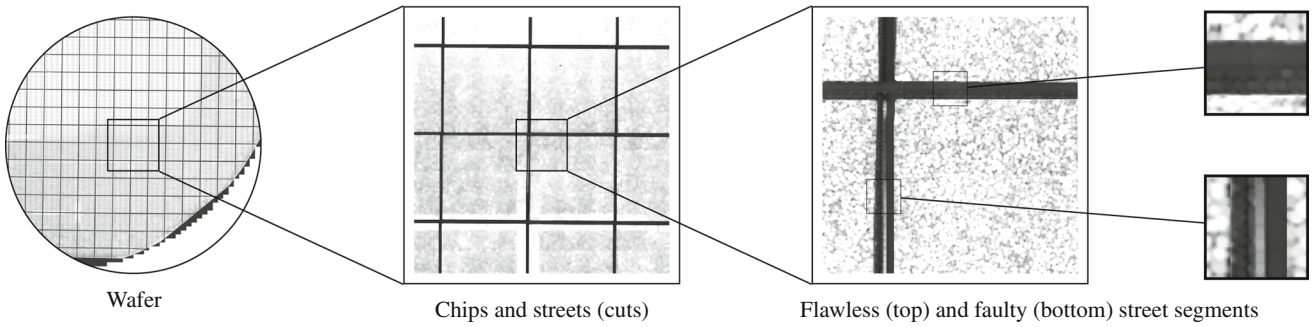


Fig. 1 Wafer overview (left) with chips and streets (middle) as well as flawless and faulty street segments (right) as a result of the wafer dicing process (reprinted from Schlosser et al. (2019)). In order to protect the intellectual property of the wafer imagery, the shown examples are alienated from the original imagery while retaining a close resemblance

the automation of quality control allows manufacturers to benefit from an increase in yield and a reduction of manufacturing costs.

We approach the problem of automated visual fault detection and recognition in the field of semiconductor manufacturing (Hooper et al. 2015; Rahim and Mian 2017). Silicon wafer dicing denotes the separation of silicon wafers into single components, whereas the dicing streets describe the scribed regions of interest on the wafer surface. One of the more commonly deployed separation approaches utilizes a dicing saw (Hooper et al. 2015). Another, alternative method is thermal laser separation, where a thermally induced mechanical force results in a cleave on the wafer surface (Rahim and Mian 2017). For this purpose, the cleave is guided along the scribe in this thermally induced separation. This separation process constitutes our quality criterion, as a cleave deviating from the scribe results in faulty chips and therefore a decrease in yield.

A particular challenge represents the detection and classification of complex defect patterns in pixel size, which often occur in this application area. Figure 1 illustrates this by means of a wafer segment (left), subdivided into chips (middle) and street cuttings (right) as they are produced through the cutting process along the scribes. While the data acquisition of semiconductor manufacturing processes often results in vast amounts of image data, defect patterns often occur in pixel size within image resolutions of up to $10^5 \times 10^5$ pixels.

The detection and classification of such small defect patterns is often a problem for deep learning based approaches. Deep neural networks are able to either process the given input in its native image resolution or utilize a downsampled version of the input imagery. However, both options lead to problems in the considered application area. The first case would therefore result in a larger network, possibly leading to slower processing performances, while a higher network connectivity would lead to additional problems such as data overfitting. In the second case, the input imagery would be downsampled for the network, whereas smaller structures within pixel size would be lost. Therefore, the deployed system's localization and classification capabilities have to be considered in order to allow a more efficient recognition of defect patterns.

Depending on the underlying manufacturing process, a variety of defect patterns can occur on the wafer surface, including "spur", "break out", "wrong size hole", "overetch", "missing hole", and "excessive short" as well as "cluster pattern", "edge ring", "linear scratch", and "semicircle" based defect patterns (Moganti and Ercal 1998; Cho and Park 2002), also described also by Chen and Liu (2000); Liu et al. (2002). However, the utilized wafer data are often synthesized or originates from only one manufacturer or one type of wafer, resulting in near-perfect classification accuracies which are in turn less informative regarding the system's real-world classification capabilities.

Another, related research area opposing to the inspection of visual material is concerned with the analysis of wafer maps which are based on a contact needle inspection. Wafer maps are typically generated by placing an uncut wafer into a wafer prober device (Cheng et al. 2021). This test device utilizes contact needles in order to establish a connection to individual circuits. After the wafer is cut, the defective circuits are sorted out, the results are color-coded as wafer maps, and their patterns are analyzed to optimize the manufacturing process. However, because of its intrusive nature, the electrical test bears the risk of potentially damaging the wafer (Cheng et al. 2021).

Our wafer data were captured from real-world dicing processes of different semiconductor wafers as they were provided to us by various third-party manufacturers. After each wafer was mounted on a taped frame, the dicing tape was expanded to visualize the extremely thin cuts under a wide-field light microscope following the cleaving process. Subsequently, the microscope scans each wafer, whereas the scanning stage allows, as in our application, a line-wise imag-

ing of up to 150 mm (6 inch) wide wafers. Finally, an option to stitch the resulting wafer imagery is provided.

In our process scenario, wafers are diced with 200 to 300 mm/s. Using laser-assisted dicing, up to 10 wafers per hour can be processed with a die size of $2 \times 2\,\text{mm}^2$ at a feed rate of 200 mm/s (Belgardt et al. 2017). The subsequent pipeline is process and customer dependent, but often consists of an imaging step (after expansion), following of a grab-and-place robotic system picking chips from the wafer. Here, the grabbing control system can benefit from information about the quality of the previous dicing step. In an efficient process pipeline, the given time for one wafer is in the order of minutes while several thousands of single chip and street images need to be processed by the image processing system. This corresponds to several tens of milliseconds per image, e.g., 6000 images to be processed within 5 min, resulting in 50 ms per image. This or even faster computation times were therefore defined to be among the requirements of our selected algorithms.

The conceptual theory behind our idea is inspired by findings from psychology and visual attention. We observed that human workers pay in fact more attention to task-relevant regions of interest (ROI, Itti et al. (1998)), i.e., streets or street-based segments, thus focusing them. Such a focus process is in terms of human research and neuroscience known as visual attention (Carrasco 2011; Hamker 2005; Beuth and Hamker 2015; Reynolds and Heeger 2009). Visual attention has many findings, whereas a common one is the focusing of human processing on an aspect of a scene (Hamker 2005). When it is a particular region, as in our case, attention is denoted as spatial visual attention (Carrasco 2011). There are certainly different approaches that allow the implementation of the localization process. One option is to choose a classical computer vision pipeline for its applicability. Certainly, more complex as well as visual attention inspired neuroscience-related models are also existing, inter alia, saliency models Itti et al. (1998) or system-level attention models Hamker (2005).

Related work

The following sections give an overview over related as well as currently deployed approaches to automated visual fault inspection within the domain of semiconductor manufacturing, separated into conventional baseline approaches (Sect. 1.1.1), machine learning (ML) based approaches (Sect. 1.1.2), as well as deep learning (DL) based ones (Sect. 1.1.3).

Conventional baseline approaches

Classically, the field of automated visual inspection utilized image processing approaches which are most commonly distinguished based on their functionality in projection-based, filter-based, and hybrid approaches (Huang and Pan 2015). Following Huang and Pan (2015), projection-based approaches often include principal component (PCA), linear discriminant (LDA), or independent component analysis (ICA), whereas filter-based approaches encompass spectral estimation and transformation-based approaches, including discrete cosine (DCT), Fourier (FT), and wavelet transforms. However, various conventional approaches were developed which utilize computer vision techniques and statistics, e.g., Sreenivasan et al. (1993), Zhang et al. (1999), Chen and Liu (2000), Tobin et al. (2001). For example, Zhang et al. (1999) deployed a post-sawing inspection system which extracts boundary features from reference images by utilizing computer vision techniques.

ML-based approaches

The domain of machine learning (ML) provides a range of powerful toolsets for image analysis. Hence, a broad variety of research utilizes machine learning based approaches to develop systems for automated visual fault inspection, often resulting in improved detection and classification rates while showing higher generalization and adaptation capabilities to novel defect patterns. As demonstrated by Chen and Liu (2000), Huang (2007), and Xie et al. (2014), the first learning-based approaches made use of multilayer perceptrons (MLP) and support vector machines (SVM), including supervised and unsupervised approaches. An adaptive resonance theory network (ART) was evaluated on real-world data by Chen and Liu (2000) and Liu et al. (2002). Chao-Ton et al. (2002) proposed a system for post-sawing inspection which applies various types of supervised artificial neural networks (ANN), combining backpropagation (Werbos 1990), a radial basis function network (RBF), and learning vector quantization (LVQ). Xie et al. (2014) applied SVMs on synthesized real-world data, while Xie et al. (2011), Song et al. (2013) utilized support vector regression (SVR) and SVMs for reliability analysis. Xie et al. (2013) systematically compared different multi-class SVMs and LVQ, while Mahadevan and Shah (2009) detected abnormal process behaviors in the semiconductor etching process. Nawaz et al. (2014) developed an inference system for the etching process based on a Bayesian network (Friedman et al. 1997) which evaluates the cause-effect relationships between root causes, equipment, and process parameters.

DL-based approaches

In the more recent years, deep learning and deep neural network (DNN) based approaches overtook conventional ML-based approaches for most use cases in terms of classification accuracy (LeCun et al. 2015). However, in the domain

of semiconductor manufacturing, current use cases are still limited and more research has to be conducted to further proof their applicability within the domain.

Chen et al. (2015) proposed a convolutional neural network (CNN) based approach for defect pattern evaluation of gearboxes where spectral information from time and frequency domains as well as other statistical measures were selected as input. Similarly to Lee et al. (2016), Zheng et al. (2016) investigated the use of multivariate and univariate time series data as CNN input. Following this approach, Lee et al. (2017) inspected the production process over time for an on-the-fly control and evaluated it on real-world data, while Lee and Kim (2018) deployed recurrent neuronal networks. Hsu and Liu (2021) designed a classification system that utilizes a large number of sensors while feeding their data with a temporal relation into their proposed multiple time series CNN. Another, related area of interest is concerned with DL-based approaches for the analysis of wafer map patterns generated by a wafer prober test (Hsu and Chien Hsu and Chien; Hyun and Kim 2020; Saqlain et al. 2020; Kim et al. 2021). Depending on the nature of the resulting patterns, inspectors can draw conclusions about the manufacturing process and optimize its parameterization in order to maximize yield.

The contribution of Nakazawa and Kulkarni (2018) represents a CNN-based classification approach trained on synthetic wafer maps in order to classify real wafer data more precisely, resulting in a classification accuracy of 98.2%. Following this approach, Cheon et al. (2019) developed a CNN for feature extraction and classification, reaching a classification accuracy of 96.2% without synthesizing any data for the training process itself. Finally, O'Leary et al. (2020) extended this approach by additionally applying energy-dispersive X-ray spectroscopy data, which in turn allowed them to reach a top-3 classification accuracy of 99.2%.

Following Nakazawa and Kulkarni (2018), Cheon et al. (2019), and O'Leary et al. (2020), even the latest research encompasses classical single CNN based architectures. As these CNN-based approaches do not allow a distinction depending on the level of detail, an approach has to be found which enables the recognition of fine-grained defect patterns in the automated visual inspection process. Furthermore, Li and Huang (2009) concluded that a high generalization performance has to be enforced. Therefore, supervised approaches are suitable for this task while being capable of achieving higher performances.

Advanced DL-based approaches including region-based DNNs (Bochkovskiy et al. 2020; Wen et al. 2020; Han et al. 2020) are, however, often limited by the resulting vast amounts of region proposals for the following classification process. Following the general approach and application of region-based DNNs, Wen et al. (2020) proposed a system for wafer surface defect inspection which is realized via a region proposal network (RPN). The resulting region proposals are utilized as input for their so-called deep multi-branch neural network, which results in a pixel-precise localization and segmentation of the occurring defects. A similar approach is conducted by Han et al. (2020) for wafer defect segmentation. However, even current state-of-the-art approaches to region proposal networks and region-based DNNs show a relatively low processing speed and thus frame rates when utilizing higher resolution imagery (Bochkovskiy et al. 2020). When it comes to runtime constraints in manufacturing processes, the visual fault inspection capabilities as well as the system's accuracy have to be maximized while minimizing the amount of wrongly classified samples.

Contribution of this work

This contribution is concerned with the design of a novel hybrid multistage system of stacked deep neural networks (SH-DNN) that combines the advantages of classical image processing approaches with artificial neural network based ones. Our hybrid approach draws the focus over the level of detail for each step of the inspection process to more task-relevant areas of interest. It performs the localization of the finest structures within pixel size via a classical computer vision pipeline, while the classification process is realized by deep neural networks. The deep neural networks are then able to process the given input in a much higher resolution. This allows a better processing and thus a more efficient detection of the finest structures which often range within a size of only a few µm and pixels within vast amounts of high-resolution imagery.

Furthermore, all included processing steps have to be suitable in regards to given runtime constraints in deployment, which we evaluate and account for. There are several different approaches that allow the implementation of the localization process. To automatically identify regions of interest and enhance their contribution for further analysis, we chose a classical computer vision pipeline for its applicability and to satisfy given runtime constraints. In the evaluation of this contribution, we give a broad overview over currently deployed models within each step of our proposed system. This evaluation includes the system's classification capabilities in each processing step as well as a visualization of the resulting defect patterns and their occurrences in order to facilitate a further assessment by an inspector.

Distinctions to our previous contributions

The idea of an SH-DNN-based approach for automated visual inspection is motivated by one of our previously presented work-in-progress contributions. Extending the in Schlosser et al. (2019) introduced system based on convolutional neural networks, the current contribution serves as a more gen-

eral approach. This covers a deeper and broader evaluation regarding the utilized machine and deep learning based approaches, whereas the system itself was extended to further interpret defect patterns within subregions. Additionally, an evaluation regarding the system's real-time capabilities is provided.

The data set differs from our previous contributions due to a reassessment of the data, leading as a notable difference to a revised defect pattern class specification. Furthermore, the data set was changed between this contribution and both Schlosser et al. (2019) and Beuth et al. (2020), to make our approach more applicable to the process.

Our other previous contribution of Beuth et al. (2020) represents an approach with the different data set and based on a biologically plausible model of visual attention. The previous work's system (Beuth et al. 2020) is a neuro-computational model deeply rooted within the neuroscience of the brain, including neuronal firing rates and expressed human behavior (Beuth 2019). The approach of Beuth et al. (2020) has allowed us to analyze the idea of attention deeply, which was one of the objectives of our previous contribution.

Section overview

The following sections introduce our proposed multistage system, its underlying addressing scheme for the steps of localization, classification, and data augmentation as well as an overview of the wafer manufacturing data (Sect. 2). Ensuing the implemented system for our application area in the domain of semiconductor manufacturing in Sect. 2.3, we give an overview over the processing steps for wafer, chip, street, and street segment processing following a set of typically selected approaches. For the test results, evaluation, and discussion (Sect. 3), we then give an overview over a multitude of different deployed and evaluated approaches from machine and deep learning. This includes not only an evaluation regarding training and test results, but also an evaluation regarding performance as well as a visualization of the resulting defect patterns and their occurrences.

Materials and methodology

Evoked by the parameterization of the underlying cutting process, machine errors, or human carelessness or exhaustion, highly diverse defect patterns can emerge. The existing defect patterns and error classes such as small holes, scratches, or bubbles on the outer layers of the wafer and street surfaces indicate the complexity of inspecting different flawless, anomaly, and faulty classes of defect patterns within varying circuit layouts. These classes of defect patterns are in turn often characterized by different structures and relationships between them. For this purpose, a distinction is made between the two error classes of flawless and faulty patterns. Some exemplary representative chip and street samples are shown in Tables 2 and 3. Flawless patterns do not possess any defect patterns that entail a negative influence on the following processing steps. In contrary to that, faulty patterns include all chips displaying a structural damage to the chip. Some patterns can be regarded as potentially defective, denoted as anomalies. Occurrences of such anomaly cases include for example artifacts from the imaging process or unidentifiable patterns. For the classification process, these samples are placed in the faulty pattern class, while anomaly cases can be later manually inspected by a process engineer via our visualization.

The following sections introduce our data set (Sect. 2.1), our realized addressing scheme to allow the localization of single elements such as chips and streets on the wafer (Sect. 2.2), as well as our hybrid multistage system of stacked deep neural networks (SH-DNN) (Sect. 2.3). Following the proposed multistage system, we introduce a baseline approach for chip classification (Sect. 2.4) as well as our training setup (Sect. 2.5).

Data set overview

Our data set consists of various types of wafers which all show different structures and image resolutions, ranging from 224×224 to up to 960×1024 pixels per chip. Most importantly, this also implies different characteristics and occurrences of subclasses for flawless and faulty chips, streets, and street segments. A summary is depicted in Table 1, resulting in six different wafer types as well as up to 5 000 chips and 13 500 streets for each wafer. For street segments, each street was separated into multiple squared street segment regions with minimal overlap.

Since the manufacturing process itself is already highly optimized in order to minimize possible occurring defects, image data sets for defect patterns within the domain often feature only a limited amount of truly faulty samples. To counteract the lower occurrence frequencies of individual error classes for chips, streets, and streets segments, each data set in every stage of the system is balanced for the training process beforehand by sampling and duplicating faulty samples. This sampling process is followed by a data augmentation to generate additional faulty samples. Despite several images being identical in the data set, the images appear differently while producing a good amount of image variations based on one source image (Beuth et al. 2020; Zhao and Kumar 2018).

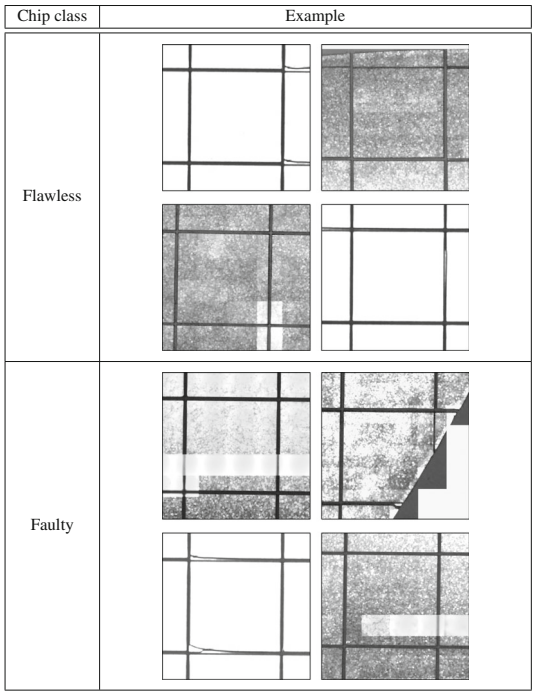
As it is often common practice to synthesize data for fault detection and classification, it is also noted at this point that all wafers where obtained from real-world dicing manufacturers, whereas each wafer was scanned after the cutting process.

Table 1 Data set overview with chips (including inside and border as well as flawless and faulty chips) and streets (including flawless and faulty streets) per wafer type

Wafer type	1	2	3	4	5	6	∑
# Chips	2 113	5 041	2 304	210	754	2 050	12 472
# Inside	1 113	3 378	1 771	115	612	1 336	8 325
# Border	1 000	1 663	533	95	142	714	4 147
# Inside-flawless	920	2 920	1 623	55	445	1 336	7 299
# Inside-faulty	193	458	148	60	167	0	1 026
# Streets	4 436	13 504	7 024	428	2 368	5 376	33 136
# Flawless	3 983	12 624	6 892	297	2 094	5 329	31 219
# Faulty	453	880	132	131	274	47	1 917

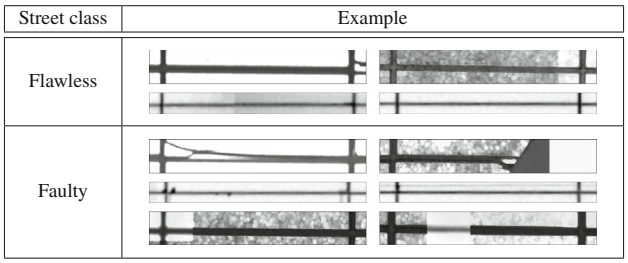
Flawless and faulty chips are denoted from the inside chips only. The streets are derived as well from the inside chips only

Table 2 Data set chip classes overview of eight representative examples



Flawless and faulty refers always to the quality of the center-displayed chip

Table 3 Data set street classes overview



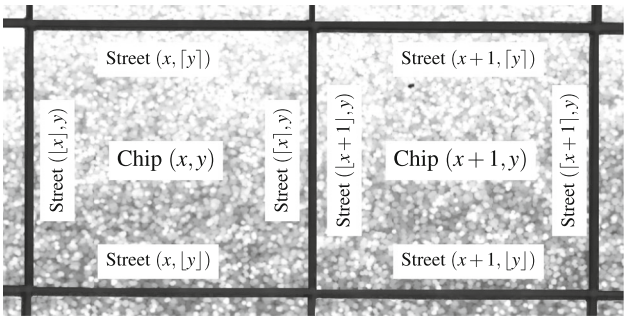


Fig. 2 Overview of our proposed addressing scheme for chips and streets. Every occurring chip is given a separate coordinate denoted by (x, y) with $x, y \in \{n+0.5, n \in \mathbb{Z}\}$. Street coordinates are derived from the chip coordinates by rounding up ($\lceil x \rceil$) or down ($\lfloor x \rfloor$). The point of origin is situated in the center of the wafer

Chip localization and addressing scheme

If necessary, as a first processing step, the wafer images are separated into their chips. In order to allow the localization of each processed wafer, an addressing scheme for its chips, streets, and street segments has to be determined. Since the focus of the localization and classification of defect patterns lies between the chips of a wafer, a coordinate is assigned to each street which corresponds to a unique position inside the wafer. The addressing scheme shown in Fig. 2 represents the basis of the chip, street, and street segment based addressing using the example of two adjacent chips whose point of origin is situated in the center of the wafer. Therefore, all chips are addressed by their coordinates (x, y) with $x, y \in \{n + 0.5, n \in \mathbb{Z}\}$, whereby the streets on the left $(\lfloor x \rfloor, y)$, on the right $(\lceil x \rceil, y)$, at the top $(x, \lceil y \rceil)$, and at the bottom $(x, \lfloor y \rfloor)$ are determined as dependent on their neighboring chips in the coordinate system. Streets are distinguished into left and right halves as well as upper and lower halves, hence the addressing scheme is extended by using an additional identifier $i \in \{l, r, t, b\}$ for left, right, top, and bottom in relation to the current parent chip, respectively.

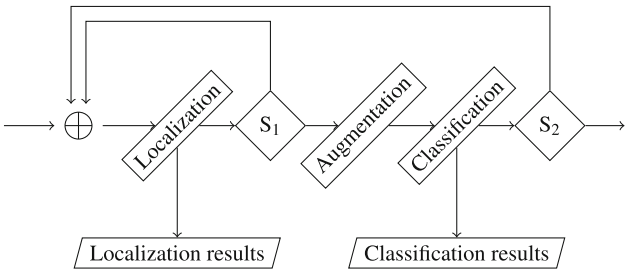


Fig. 3 Processing steps control flow graph of localization, augmentation, and classification with switches S_1 and S_2 (adapted and reprinted from Schlosser et al. (2019))

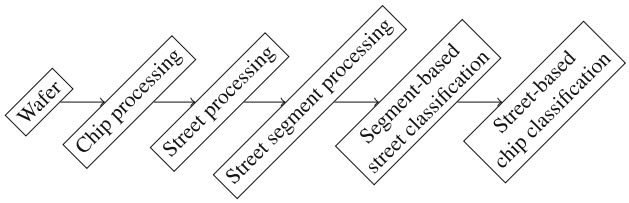


Fig. 4 Implemented visual fault inspection system. The system comprises the chip, street, and street segment localization, augmentation, and classification. Chip, street, and street segments each undergo one iteration of the in Fig. 3 shown control flow graph. For details, see Sect. 2.3

The proposed multistage system

Recognizing defect patterns with higher precision is possible due to a magnification of the underlying image material. While the localization of defect patterns in the finest structures in pixel size depends foremost on the provided image resolution, a distinction is therefore made by further differentiating depending on the level of detail, whereby chips, streets, and street segments are considered separately. Following this principle, we propose a hybrid multistage system of stacked deep neural networks (SH-DNN) based approach which allows the localization of a specifically provided region of interest (ROI, Itti et al. (1998)). This is advantageous for the following classification process, enabling the localized ROI to be processed in a much higher resolution.

Figure 3 depicts the control flow graph introducing the strategy of our approach. If necessary, the images of the provided chips are first separated according to their streets (localization). Then, the separated streets are classified in regards to their prevailing defect patterns in flawless and faulty streets (classification). In order to be able to counteract possible lower frequencies of occurrence of individual defect patterns, a data augmentation is applied to allow the creation of additional faulty samples (augmentation). The switches S_1 and S_2 are deployed as a measurement to control the system's processing steps in each iteration, i.e., allowing the later processing step to be skipped (S_1) or to determine if additional zoom levels are required (S_2). Finally, by backtracing defects, conclusions are drawn about the condition of the parental areas.

In addition, a recognition of defect patterns in the earlier stages of the automated visual inspection process eases the localization and classification of subsequent and more difficult to detect defect patterns in later stages of the system. At this point it is noted that erroneous samples within larger parental areas are returned and omitted from further assessment. Furthermore, the proposed multistage system's complexity could also be minimized. Following this principle, models with less complexity should be selected for the earlier stages to handle easier tasks, while deeper stages of the system will handle more complex classification problems.

Figure 4 represents our realized SH-DNN-based system and its processing steps. In the following sections we introduce each related processing step (Sects. 2.3.1 to 2.3.5). Our system processes a whole wafer to detect faulty and faultless chips. Before wafer images are processed, they were recorded by deploying different microscopes, which provide us with either unstitched subimages or, alternatively, preprocessed images which are stitched together by the microscope software into a single image. In our proposed system, firstly the chips are classified into chips that are situated inside the wafer area, and into chips on the wafer border or beyond it. Chips that are situated on/beyond the wafer border are also scanned by the microscope, but these chips are typically incomplete or broken (see also Fig. 1, wafer border). In our case, the manufacturers are not much interested in defects in the outer chips, thus only the inner chips are further processed. Next, our computer vision pipeline is evoked to locate the regions of interest for the inner chips. In our application, ROIs represent the chip borders, streets, as well as their surroundings. Finally, the determined ROIs are forwarded to the subsequent CNN in order to detect faults within these regions. For a further, fine-grained analysis, faulty streets are additionally investigated in detail by the street segment analysis for defective street regions. Finally, we calculate if a chip is faulty or not, depending on the classification of the four borders of each chip. To show the benefits of our level-of-detail approach, we will compare our full system to approaches without the level-of-detail as well as approaches for automated visual fault detection within semiconductor industry.

Separation of chips in inside and outside the wafer border

Firstly, a localization of the chips takes place according to their position on the wafer, i.e., the system separates chips into inside the wafer and chips outside or on the wafer border situated chips (see Fig. 4, step chip processing). Since the border chips are typically broken, they have to be excluded

Table 4 Custom CNN layer configuration for chip classification

Unit	Layer	Type	Output shape	Kernel size	Stride
conv1	conv1_1	conv	188 × 188 × 32	5 × 5	1
	conv1_2	conv	186 × 186 × 48	3 × 3	1
	pool1	max pool	62 × 62 × 48	3 × 3	3
	dropout1	dropout	62 × 62 × 48	/	/
conv2	conv2_1	conv	60 × 60 × 64	3 × 3	1
	conv2_2	conv	58 × 58 × 96	3 × 3	1
	pool2	max pool	29 × 29 × 96	2 × 2	2
	dropout2	dropout	29 × 29 × 96	/	/
conv3	conv3_1	conv	27 × 27 × 144	3 × 3	1
	conv3_2	conv	25 × 25 × 192	3 × 3	1
	pool3	max pool	12 × 12 × 192	2 × 2	2
	dropout3	dropout	12 × 12 × 192	/	/
fully conn	flatten1	flatten	27 648	/	/
	dense1	fully conn	192	/	/
	dropout4	dropout	192	/	/
	dense2	fully conn	2	/	/
Total trainable parameters: 5 781 986					

Table 5 Custom CNN layer configuration for street classification

Unit	Layer	Type	Output shape	Kernel size	Stride
conv1	conv1_1	conv	60 × 188 × 32	5 × 5	1
	conv1_2	conv	58 × 186 × 48	3 × 3	1
	pool1	max pool	19 × 62 × 48	3 × 3	3
	dropout1	dropout	19 × 62 × 48	/	/
conv2	conv2_1	conv	17 × 60 × 64	3 × 3	1
	conv2_2	conv	15 × 58 × 96	3 × 3	1
	pool2	max pool	7 × 29 × 96	2 × 2	2
	dropout2	dropout	7 × 29 × 96	/	/
conv3	conv3_1	conv	5 × 27 × 144	3 × 3	1
	conv3_2	conv	3 × 25 × 192	3 × 3	1
	pool3	max pool	3 × 8 × 192	1 × 3	1 × 3
	dropout3	dropout	3 × 8 × 192	/	/
fully conn	flatten1	flatten	4 608	/	/
	dense1	fully conn	192	/	/
	dropout4	dropout	192	/	/
	dense2	fully conn	2	/	/
Total trainable parameters: 1 358 306					

from further evaluation. The border chips encompass therefore chips which are cut by the wafer's border as well as all remaining areas beyond it. We found that the recognition of border chips is a relatively easy task as the occurring defect pattern on the wafer border is easier to detect. The chip is crossed by a large black shape constituting the wafer border (Fig. 1).

Our CNN for inside and border chip classification is inspired by *Simonyan and Zisserman's* VGG Network (Simonyan and Zisserman 2015) and consists as depicted in Table 4 of the first shown convolutional block and following fully connected one. As CNNs usually operate on mid-level image resolutions, the chip imagery is scaled to an initial image resolution of 192 × 192 pixels. This basic CNN is already able to achieve an accuracy of over 99%. As a consequence, chips that are already recognized as border chips are excluded from further processing.

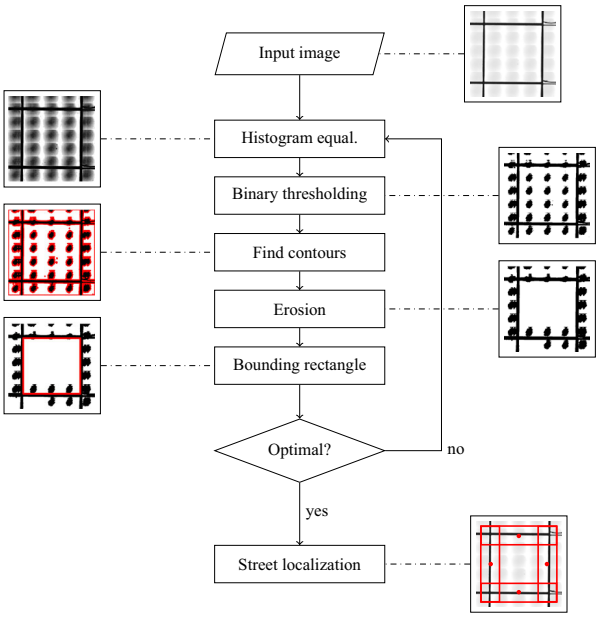


Fig. 5 Processing steps of localization with in red highlighted resulting contours and bounding rectangle. From the bounding rectangle, the street center coordinates (red dots) are determined. The resulting localized streets are outlined in red

Performant computer vision pipeline for fault localization

In order to extract the relevant regions of interests in a given image template of the integrated circuit layout, the following processing steps are applied (see Fig. 4, step street processing, part localization). The evoked functionality is shown from top to bottom in Fig. 5 as single processing steps, and they are provided by the computer vision library OpenCV.[1]

At first, a histogram equalization is applied to increase the contrast and to enhance the visibility of defect patterns under variations of brightness (OpenCV function "equalizeHist()"). The optimization of brightness is necessary due to the varying illumination of the underlying material as well as to compensate deviations arising from different materials and lighting conditions, therefore allowing a more robust feature extraction. To remove possible artifacts, the enhanced image is then binary thresholded ("threshold()"), and a contour-based edge detection and border following is applied ("findContours()") (Suzuki and Abe 1985). Subsequently, an erosion operation serves as an additional processing step ("erode()"), before a minimal bounding box enclosing the current chip is determined ("boundingRect()"). These processing steps are repeated for all threshold values of the binary thresholding before the final threshold value is selected from them by means of the largest connected subset of thresholds with bounding boxes as representative for segmentation. From this final bounding rectangle, the streets are determined via the center of the bounding box sides, which serve as street center coordinates.

While OpenCV provides all crucial functions of the computer vision pipeline, we use standard parameters for all functions, except for the parameters "cv2.RETR_TREE" and "cv2.CHAIN_APPROX_NONE" in the function "findContours()". The images are stored in grayscale format. Therefore, the binary thresholding is repeated for all grayscale values. While working with $2^8 = 256$ different gray levels due to 8-bit grayscale images, our tests showed that a range with a lower and an upper threshold value of 50 and 200 is often sufficient enough to find a set of promising thresholds. The binary thresholding process itself can be repeated or aborted as long as enough thresholds with ROIs have been determined. Regarding thresholding algorithms Roy et al. (2014), while approaches such as Otsu's method, Kapur's thresholding, and entropy-based thresholding yield promising results, they require more processing power than our simpler binary thresholding approach. However, binary thresholding typically comes with the underlying problem of finding a proper threshold. We solved this problem with the aforementioned approach to make it applicable for this use case, hence allowing the optimization of our pipeline.

Street classification

The street regions are localized by utilizing the pipeline of classical computer vision techniques (previous section). On each side, the center of the chip border is determined by the pipeline and returned as the center of the street region. We define a region of interest (ROI) of 120% the size of the chip and 6× the width of the street around this center, and follow the in Beuth et al. (2020) outlined procedure. The goal of our application is to detect dicing defects which occur in the space between the streets and the chips and possibly continue inside the chips. To allow an optimal coverage of these areas, the ROI is centered so that the currently investigated street with its parts of adjacent chips and street crossings is situated in the center of the image.

The street regions are then classified into the two classes of flawless and faulty streets via a CNN (see Fig. 4, step street processing, part classification). Our realized CNN architecture configuration for street classification is also influenced by *Simonyan and Zisserman's* VGG Network and consists of three convolutional blocks as shown in Table 5. Alternatively to our custom CNN network, we conducted in our proposed test environment (i) tests with standard deep learning networks and (ii) conventional machine learning methods, and investigated them for the classification.

In order to maximize the efficiency of our network, we emphasize the areas where defect patterns occur. Whereas for street-based classification each chip possesses four corresponding streets, every street is rotated so that the side

[1] https://opencv.org/.

with the chip border faces upwards. This results in an input layer where every image shows significantly more pixels in width than in height. The approach allows stable regions on the image where defect patterns might occur, e.g., the chip, while emphasizing their contribution. Therefore, we lowered the pooling in the direction of y for street-based classification in comparison to the chip-based one accordingly, while otherwise following the methodology of Beuth et al. (2020). Streets that are already recognized as defective are forwarded to the street-based chip classification, whereas only flawless streets are forwarded to the next processing step of street segment processing.

Street segment classification

To allow an even finer defect pattern localization and classification, we propose to add another processing step. Based on the scribed regions, segments of each street are separated (see Fig. 4, step street segment processing). Due to the vast amounts of street segments, the street segment recognition is performed without an initial labeling of the segments using unsupervised learning (Schmidhuber 2015). Unsupervised approaches are often deployed in automated and semi-automated visual inspection scenarios (Mei et al. 2018). Especially autoencoders (Baldi 2012) enable the detection of defect patterns without the need for labeled faulty samples. An autoencoder reconstructs its given input through various encoding and decoding steps within its network, whereas the relevant information content as well as a representation of the provided input is learned. Our selected model is inspired by Zhao et al. (2016) approach to Stacked What-Where Autoencoders (SWWAE) with residual learning (He et al. 2016a, b). The deployed SWWAE-based model with residual learning is a further development of the general autoencoder which reuses the encoded positions for the decoding process supported by additional skip connections via residual learning. Its implementation is described in Schlosser et al. (2020).

The deployed SWWAE is therefore utilized to encode and decode the provided street segments, whereas the encoder reduces the input to its relevant information content and the decoder reuses this information content in order to recover the input imagery. By minimizing the training loss, an objective function is optimized, which is utilized to differentiate novel street segment samples. Therefore, by training on flawless samples, it is possible to detect and discriminate faulty samples by evaluating the SWWAE's loss. Furthermore, a larger loss indicates a sample that deviates strongly from the set of trained flawless samples.

Chip classification via fault backtracing

In the final step, the result of the chip classification is returned based on the classified streets. The obtained error classes are mapped according to their classified defect patterns, ranging from flawless to faulty streets (see Fig. 4, step street-based chip classification). If a chip has at least one faulty side, it is defined as faulty. Otherwise, it is defined as flawless. Therefore, the street classification of the adjacent streets is adopted for occurring chips according to Table 3. The chip classification itself is then the final output of the system.

In parallel, the output of the street segment analysis is utilized via backtracing for the defect visualization (see Fig. 4, step segment-based street classification). Streets which consist of one or more faulty segments are considered faulty. Hence, a faulty street is derived from their faulty street segments. This does not only allow an easier assessment, but also allows the identification of trends regarding occurring defect patterns and their further progression.

Baseline approach for chip classification

To benchmark our system, we provide an additional model consisting of a single DNN as a baseline approach. The variant directly classifies chips for flawless and faulty samples, and no additional localization steps are deployed. The DNN model is again inspired by the VGG network, consisting of three convolutional blocks as illustrated in Table 4. For a straightforward comparison, we alternatively evaluated again (i) standard deep learning networks and (ii) conventional machine learning methods on chips in our proposed test environment.

Training setup

Following the previously outlined data set overview, the data set was balanced for the CNN classification in such a way that the classes roughly occur equally frequently (class-wise balancing) as described in Sect. 2.1. Depending on the wafer, strong illumination differences may occur, which are in turn suboptimal for the following processing stages. Therefore, we chose to apply a chip-wise histogram equalization as a wafer-wise equalization would not be sufficient enough, similar as in our previous work (Beuth et al. 2020). Furthermore, the data sets were sample-wise standardized by centering to a mean of zero and dividing through the standard deviation, both for the CNNs as well as the conventional machine learning approaches.

Regarding data augmentation, while the application of such methods often depends on the given characteristics of the existing image data as well as the properties of relevant features of the defect patterns that have to be recognized, a set of classical data augmentation approaches can be applied. The following randomized data augmentation methods are provided and implemented via the image

augmentation library imgaug.[2] Depending on the level l of the augmentation, the respective method is amplified l-fold (notation $l\times, l \in \mathbb{N}_{\geq 0}$). This includes a rotation by up to $l \times \pm 2°$, a translation in x and y by up to $l \times 5\%$ and $l \times 1\%$, a scaling by up to $l \times \pm 2\%$, as well as a mirroring on the x-axis (Schlosser et al. 2019; Beuth et al. 2020).

For training itself, the machine learning framework Keras[3] with its underling back end TensorFlow[4] was used, which also enables an accelerated processing by general-purpose graphics processing units (GPGPUs). To allow a comparison with current learning-based as well as conventional baseline approaches, the machine learning library scikit-learn[5] was utilized.

Standard parameters were deployed for all ML- and DL-based models that are tested in our evaluation. The parameterization of individual models was not adjusted or optimized by means of pre-trained weights, by utilizing transfer learning, or by deploying additional optimization approaches such as grid searches. To keep a fair comparison of the different models, we decided to utilize them as out-of-the-box models as provided by Keras, and run all of them with precisely the same hyperparameter configuration. An example for already implemented models are the ResNet models provided by Keras, ResNet-50, ResNet-101 and ResNet-152 (He et al. 2016a), which differ by their corresponding layers and depth as denoted by their identifier.

Our training setup includes: the Glorot initializer (Glorot and Bengio 2010) for weight initialization, the Adam optimizer (Kingma and Ba 2015) 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. To avoid interference between training and test data, we conducted our test runs with the data split randomly into training, validation, and test set with a ratio of 50/25/25%.

Test results, evaluation, and discussion

In order to quantify the detection and classification capabilities of defect patterns of the realized system over chips, streets, and street segments, an evaluation of the individual processing steps is performed. Following Fig. 4, this includes the chip, street, and street segment localization as well as the augmentation of the respectively given image data, but also an evaluation of the multiple models' complexity in order to ensure their applicability in real-time critical environments. The training and testing process itself should not only result in stable rates regarding training, validation, and test accuracies, but also satisfy given runtime constraints.

In the following sections, we evaluate our proposed system by utilizing different conventional baseline as well as DL-based approaches (Sect. 3.1). Finally, the system's performance is evaluated for every given processing step (Sect. 3.2). To meet current standards regarding runtime constraints as well as performance and visual fault detection and classification capabilities, a set of models is selected which in turn meet the requirements of an optimal balance of classification accuracy and performance.

Training and test results

Following the current state of art in visual inspection, mostly single DNN based approaches are deployed. While our evaluation gives an overview over possible influences of each processing step, the major outcome of the system is the classification based on streets.

Table 6 shows our test results of the selected approaches for conventional baseline-, semiconductor-, and DL-based approaches. These are divided into currently in the domain of semiconductor visual inspection deployed single DNN based solutions as well as in this field mostly uninvestigated approaches from DL. In addition, we evaluated conventional machine learning approaches as a baseline, which have been used for example in the field of semiconductor manufacturing in several works (Chen and Liu 2000; Chao-Ton et al. 2002; Huang 2007; Xie et al. 2011; Song et al. 2013; Xie et al. 2014). The current state of the art in the domain of automated visual inspection in the semiconductor industry is denoted in the result table via black outlines. The symbol * denotes the currently deployed single DNN based approaches in the domain that were reimplemented following their original publications.[6] In our test environment, each DNN was evaluated after 100 epochs of training, whereas the results were averaged over five runs. Over all tested approaches, the best system configuration reaches a performance of 99.5% in F1-score.

To highlight the benefits of our localization approach, we benchmarked a system where the localization stage was removed from the pipeline (Table 6, left column). This baseline system utilizes the chip images as input as explained in Sect. 2.4. For this purpose, a classification improvement factor (Table 6) is defined which represents the factor of decreasing the error rate between both systems. Based on the resulting F1-scores, the improvement factor is calculated by $^{(100\% - \text{<chip class. F1-score>})}/_{(100\% - \text{<street-based chip class. F1-score>})}$. The street-based classification as well as our simple baseline approach prove for the selected approaches in comparison

[2] https://github.com/aleju/imgaug.

[3] https://keras.io/, version 2.3.1.

[4] https://www.tensorflow.org/, version 2.1.

[5] https://scikit-learn.org/, version 0.23.1.

[6] https://github.com/TSchlosser13/Hexnet/blob/master/_ML/models/contrib/visual_inspection.py.

Table 6 Test results for the SH-DNN-based classification with conventional baseline approaches and DL-based approaches for chip and street classification

Test run	Baseline approach: classification based on chips		SH-DNN-based system: Classification based on streets		Improvement factor	
	F1-score [%]	Fault detection F1-score [%]	F1-score [%]	Fault detection F1-score [%]	F1-score	Fault detection F1-score
Conventional baseline approaches						
Linear discriminant analysis	62.8 ± 2.4	66.8 ± 2.7	90.5 ± 0.7	91.2 ± 0.6	3.92	3.77
Quadratic discriminant analysis	71.5 ± 1.4	64.0 ± 1.3	33.3 ± 0.0	0.0 ± 0.0	/	/
Extra trees classifier	**95.0 ± 0.9**	**94.8 ± 1.1**	**98.9 ± 0.2**	**98.9 ± 0.3**	**4.55**	**4.73**
Random forest classifier	94.0 ± 0.7	93.8 ± 0.6	**98.9 ± 0.2**	**98.9 ± 0.2**	**5.46**	**5.64**
Logistic regression	87.4 ± 0.6	87.0 ± 0.7	79.7 ± 1.5	79.5 ± 1.7	0.62	0.63
Ridge classifier	89.7 ± 0.6	89.3 ± 0.7	83.8 ± 1.9	83.6 ± 2.1	0.64	0.65
Gaussian Naïve Bayes	74.6 ± 2.5	72.3 ± 2.9	57.9 ± 4.1	62.9 ± 4.0	0.60	0.75
k-nearest neighbors classifier	93.3 ± 0.6	93.0 ± 0.7	98.8 ± 0.5	98.8 ± 0.5	5.58	5.83
Multilayer perceptron classifier	94.1 ± 0.5	93.9 ± 0.9	96.4 ± 1.9	96.4 ± 1.7	1.64	1.69
SVC with linear kernel	90.0 ± 0.5	89.4 ± 0.7	85.4 ± 0.7	85.3 ± 0.5	0.69	0.72
SVC with RBF kernel	91.9 ± 0.4	91.4 ± 0.4	94.4 ± 0.6	94.4 ± 0.8	1.45	1.54
Decision tree classifier	89.4 ± 1.3	89.3 ± 1.5	98.2 ± 0.9	98.2 ± 0.9	5.89	5.95
DL-based approaches: automated visual inspection approaches in semiconductor manufacturing						
CNN (2018)*	95.5 ± 1.3	95.4 ± 1.4	97.8 ± 0.9	97.8 ± 1.1	2.05	2.09
CNN (2019)*	94.8 ± 1.8	94.6 ± 1.7	98.0 ± 0.8	98.0 ± 0.8	2.60	2.70
CNN (2020)*	94.9 ± 1.4	94.7 ± 1.3	98.1 ± 0.8	98.1 ± 0.8	2.68	2.79
Custom CNN	94.7 ± 1.2	94.6 ± 1.3	97.4 ± 0.5	97.4 ± 0.4	2.04	2.08

Table 6 continued

Test run	Baseline approach: classification based on chips		SH-DNN-based system: Classification based on streets		Improvement factor	
	F1-score [%]	Fault detection F1-score [%]	F1-score [%]	Fault detection F1-score [%]	F1-score	Fault detection F1-score
DL-based approaches: general approaches						
DenseNet121 (2017)	95.6 ± 1.2	95.6 ± 1.2	99.2 ± 0.3	99.2 ± 0.3	5.50	5.50
DenseNet169 (2017)	**96.6 ± 1.3**	**96.5 ± 1.4**	99.4 ± 0.2	99.4 ± 0.2	5.67	5.83
DenseNet201 (2017)	96.1 ± 1.1	96.1 ± 0.9	99.3 ± 0.2	99.3 ± 0.2	5.57	5.57
InceptionResNetV2 (2017)	96.4 ± 0.7	96.3 ± 0.7	/	/	/	/
InceptionV3 (2016)	96.3 ± 0.6	96.3 ± 0.6	/	/	/	/
MobileNet (2017)	94.3 ± 0.5	94.2 ± 0.6	99.2 ± 0.1	99.2 ± 0.1	7.13	7.25
MobileNetV2 (2018)	94.9 ± 0.4	94.7 ± 0.4	99.3 ± 0.1	99.3 ± 0.1	7.29	7.57
NASNetLarge (2018)	86.8 ± 4.3	88.1 ± 4.8	99.1 ± 0.3	99.2 ± 0.3	14.67	14.88
NASNetMobile (2018)	92.5 ± 3.7	92.8 ± 3.5	99.2 ± 0.3	99.2 ± 0.3	9.38	9.00
ResNet50 (2016a)	95.7 ± 0.7	95.6 ± 0.6	99.3 ± 0.2	99.3 ± 0.2	6.14	6.29
ResNet50V2 (2016a)	96.1 ± 0.8	96.0 ± 0.7	98.7 ± 0.4	98.7 ± 0.4	3.00	3.08
ResNet101 (2016a)	96.0 ± 0.5	95.9 ± 0.5	99.2 ± 0.3	99.2 ± 0.3	5.00	5.13
ResNet101V2 (2016a)	96.3 ± 0.4	96.3 ± 0.4	99.4 ± 0.3	99.4 ± 0.3	6.17	6.17
ResNet152 (2016a)	95.9 ± 0.3	95.8 ± 0.3	99.3 ± 0.2	99.3 ± 0.2	5.86	6.00
ResNet152V2 (2016a)	95.7 ± 0.4	95.7 ± 0.4	**99.5 ± 0.1**	**99.5 ± 0.1**	**8.60**	**8.60**
VGG16 (2015)	33.3 ± 0.0	66.6 ± 0.0	33.3 ± 0.0	66.6 ± 0.0	/	/
VGG19 (2015)	33.3 ± 0.0	0.0 ± 0.0	33.3 ± 0.0	0.0 ± 0.0	/	/
Xception (2017)	95.8 ± 2.2	95.7 ± 2.4	/	/	/	/

Tests for inception module based DL models for street classification where omitted from the evaluation as they expect a minimal image resolution larger than 64 × 64 pixels. * denotes currently deployed single DNN based approaches within the domain of automated semiconductor visual inspection. In black outlined approaches constitute the current state of the art. The classification improvement is derived from the resulting F1-scores and is calculated by $^{(100 - \text{<chip class.>}}/_{(100 - \text{<street-based chip class.>})}$. For a visualization of the test results, see Fig. 11

Table 7 Performance (time per sample) and complexity (trainable parameters) for street classification

Test run	Trainable parameters	Performance in time per sample [ms]
Conventional baseline approaches		
All	/	< 0.1
DL-based approaches		
CNN (2018)*	2 192 098	0.13
CNN (2019)*	19 236 386	0.25
CNN (2020)*	73 455 426	0.66
Custom CNN	1 358 306	0.22
DenseNet121	7 039 554	0.78
DenseNet169	12 646 210	0.99
DenseNet201	18 325 826	1.00
MobileNet	3 230 914	0.28
MobileNetV2	2 260 546	0.32
NASNetLarge	84 924 884	2.00
NASNetMobile	4 271 830	0.84
ResNet50	23 591 810	0.82
ResNet50V2	23 568 898	0.81
ResNet101	42 662 274	1.00
ResNet101V2	42 630 658	1.00
ResNet152	58 375 042	1.00
ResNet152V2	58 335 746	2.00
VGG16	56 674 114	0.87
VGG19	61 983 810	0.94

* denotes currently deployed single DNN based approaches within the domain of automated semiconductor visual inspection. For a visualization of the test results, see appendix Table 11

Table 8 Performance (time per sample) and complexity (trainable parameters) for the respective steps of localization and classification with MobileNetV2 as based on the proposed multistage system in comparison

Test run	Trainable parameters	Performance in time per sample [ms]
Chip-based local.	/	< 0.1
Chip-based class.	2 260 546	1.00
Street-based local.	/	3.20
Street-based class.	2 260 546	0.32
Segment-based local.	/	< 0.1
Segment-based class.	251 931	0.41
Fault backtracing	/	< 0.1
Total	4 773 023	5.2

that our proposed SH-DNN-based system, which combines the advantages of classical image processing approaches and artificial neural networks, allows a more accurate detection of the finest structures. We observed an increase in F1-scores by comparing the results in Table 6. Especially ResNet-based approaches result in increased F1-scores for street-based classification. Chip classification and street-based chips classification differ for conventional baseline approaches by an absolute of up to 3.9% F1-score (i.a., Extra Tree Classifier), while the DL-based approaches differ by up to 3.8% (i.a., ResNet152V2). This enhancement corresponds to a decrease of the error rate measured by the improvement factor of up to 4.6-fold for the conventional approaches and even by 8.6-fold for the DL-based ones.

A more detailed analysis covers the single steps of our system. For our baseline approach of chip-based classification for flawless and faulty chips, we observe that the deployed models show F1-scores from 62.8% up to 95%, whereby we found that the lower scores belong to approaches such as LDA while the highest scores to approaches such as Extra Trees Classifier, Random Forest, or Multilayer Perceptron. The DL-based approaches exhibit scores of up to 96.6%, i.e., DenseNet169. These slightly reduced scores compared to our full system are explained by the complexity of clas-

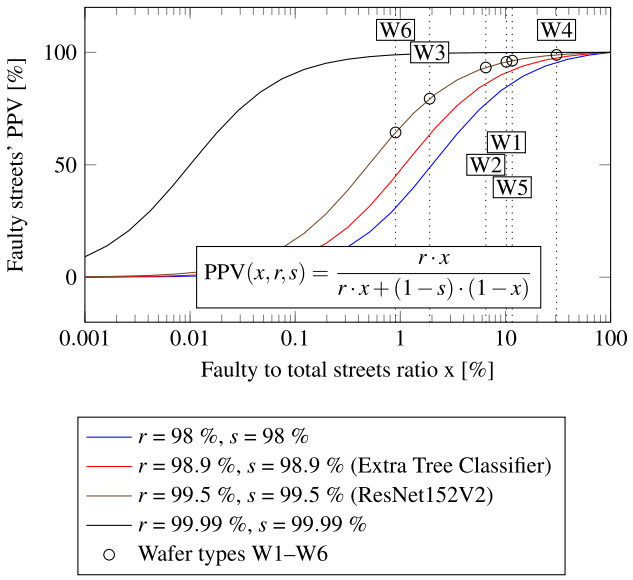


Fig. 6 Theoretical graphs of the positive predictive value (PPV) for classifiers with recall (r) and specificity (s) from 98 to 99.99% as a function of the faulty street ratio. Vertical dotted lines labelled with W1 to W6 mark our 6 wafer types as shown in Table 1 with their experimentally/manually determined fraction of faulty streets (x)

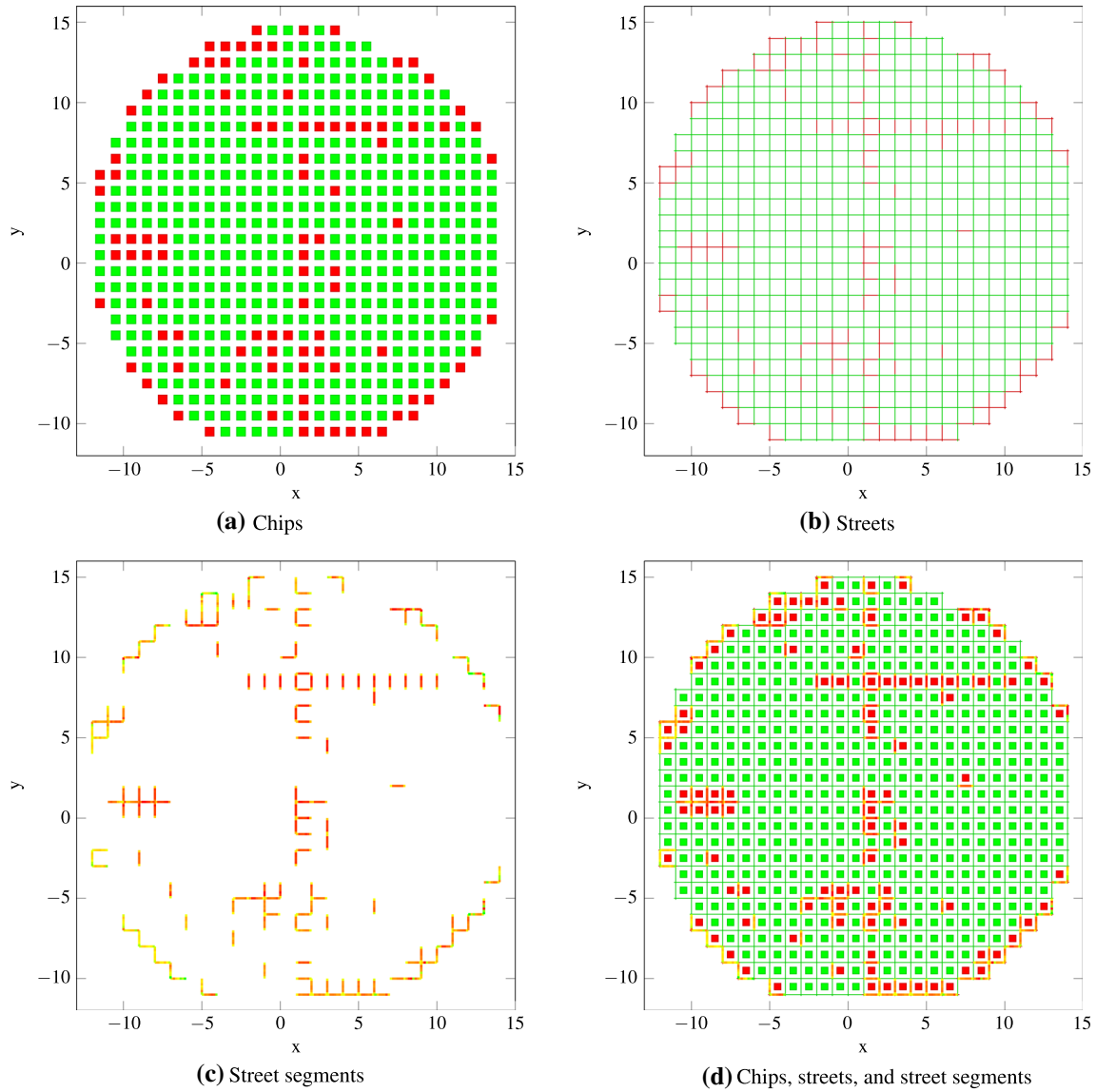
sifying street defect patterns on the significantly larger chip surface without any localization steps. This finding is especially observable for all DL-based approaches, which mostly result in score ranges of 95 to 96%, while a few networks classify less, i.e., VGG16 and VGG19.

For our initial step of chip-based classification into inside and border chips, a high performance of over 99% (F1-score) is observed even for our custom CNN. The obtained results can be explained by the fact that the task is relatively straightforward as the chips of the wafer border are crossed by a sizeable and visually distinguishable black shape constituting the wafer border (Fig. 1). Furthermore, we also evaluated related conventional baseline and DL-based approaches for the inside and border chip classification as illustrated in the appendix (Table 10). For such a simple task it is on the contrary observed that conventional baseline approaches already result in higher F1-scores. Models such as the Extra Trees Classifier as well as DenseNet169, InceptionResNetV2, and the NASNet models even reach scores of 99.3 (conventional baseline approaches) and 99.8% (DL-based approaches) (Fig. 9).

To allow a more in-depth evaluation of the system's classification capabilities, we additionally observed the resulting training and validation accuracies over time for the street classification of our custom CNN, NASNetLarge, MobileNetV2, and ResNet152V2 as shown in the appendix, Fig. 10.

The positive predictive value (PPV) provides us with the probability of correct test predictions (Altman and Bland 1994). It reports the probability that one as faulty reported chip indeed stems from an actually faulty chip. Therefore, it is an important metric in order to analyze our classification results in regards to the false negative rate, hence aiding in yield maximization during the design process of the proposed system. Figure 6 shows the theoretical graphs of the PPV as a function of the faulty street ratio given for classifiers with recalls (r) and specificities (s) from 98 to 99.99%. This illustrates that the PPV depends on the classifier's capabilities as well as the number of faulty chips in the data in relation to the total number of chips. Vertical dotted lines labelled with W1 to W6 mark the experimentally/manually determined fraction of faulty streets of our wafer types as shown in Table 1. Markers which are added on the brown graph belonging to r and s of 99.5% correspond to the best classifier in our evaluation, i.e., ResNet152V2 (Table 6). Accordingly, for 4 of 6 wafer types more than 90% of all faulty chip predictions would be actually faulty (Fig. 6). Even if the fault rate was smaller, i.e., roughly 1% faulty streets for wafer type 6, still more than 60% of all streets would be correctly identified as defective. However, the faulty fraction of new wafers is an unknown parameter and engineers have to make assumptions for their system to judge upon its reliability. We draw the following three conclusions regarding applicability: (i) Faulty street ratio $>$ 0.5%: Fault detections are reliable and less than 10% of them are actually flawless. In this context, the image analysis may be regarded as fully automatic. (ii) Faulty street ratio $>$ 0.2%: Fault detections are 10 to 50% flawless. If defects need to be quantified exactly, a semi-automatic approach could be applied here, whereas the algorithm rather acts as a recommender system and the detected faulty streets are controlled by an engineer manually. Compared to inspecting all streets manually, a considerable number of working hours can be saved. (iii) Faulty street ratio $<$ 0.2%: A distinctly higher recall/specificity than 99.5% is needed to (semi-)automatically judge the quality. This could be achieved if the classifier was adapted or trained for a specific wafer type and imaging system. We assume that values greater than 99.9% could be achieved by having more faulty street data from the same wafer type and imaging system.

Since data sets within the domain of semiconductor manufacturing often have restrictions in terms of privacy and redistribution, it is fairly difficult to make direct comparisons to the results of other works. Therefore, we implemented the proposed CNNs from related works within the domain, more namely Nakazawa and Kulkarni (2018); Cheon et al. (2019); O'Leary et al. (2020). As depicted in Table 6, the results of these models also improve as soon as we apply them to our system using our dataset.



(a) Chips **(b)** Streets

(c) Street segments **(d)** Chips, streets, and street segments

Fig. 7 Visualization of flawless (•), anomalous (•), and faulty (•) chips, streets, and street segments for our wafer 1 of type 1, whereas the respective defect patterns are visualized via color transitions from green over yellow and orange to red with their increasing degree of faultiness. The color transitions were generated via our SWWAE (Sect. 2.3.4)

Performance

The performances of the conventional baseline approaches, DL-based approaches, as well as the proposed SH-DNN-based multistage system were measured and evaluated. This includes for all approaches the time for a given chip image sample of size 192×192 pixels (Sect. 2.3.1) to be classified as well as the necessary processing time to allow the localization of a region of interest. Dependent on the current processing stage of the pipeline, a sample constitutes either a chip, a street, or a street segment image, respectively (Table 8).

To allow a comparison regarding applicability, we give an overview over a set of selected approaches from Table 6. Our chosen metrics include the performance in time per image sample to focus and classify a given image for the current processing step as well as the number of trainable parameters for training and testing as a value of the complexity of the system in use. The therefore in Table 8 and Fig. 11 depicted results show the performance and complexity results for the selected system with its model and layer configuration for localization and classification in comparison to conventional baseline as well as DL-based approaches.

For this purpose, all approaches were run in parallel, whereas the classical computer vision pipeline was parallelized for each iteration shown in Fig. 5. The classification process itself was realized to run in parallel using a GPGPU.

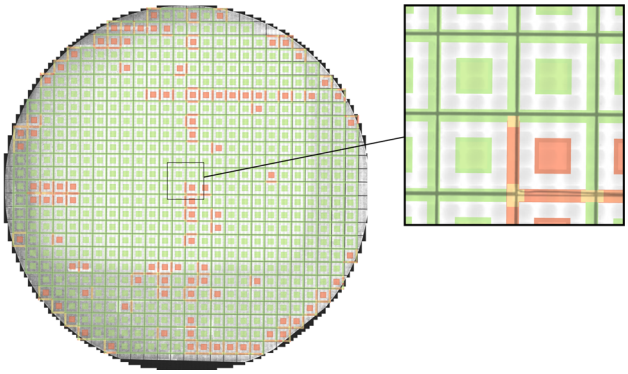


Fig. 8 Visualization of flawless (•), anomalous (•), and faulty (•) chips, streets, and street segments for our wafer 1 of type 1 as a wafer overlay. The respective defect patterns are visualized via color transitions from green over yellow and orange to red with their increasing degree of faultiness, whereas the color transitions were generated via our SWWAE (Sect. 2.3.4). Since streets usually directly neighbor two chips (see also Sect. 2.2 and Fig. 2), each obtained classification result can be further differentiated depending on the current chip of interest

Furthermore, our test environment is solely composed of current consumer grade hardware. This includes (i) our CPU, »Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz« with 7200 BogoMips and a maximum CPU load of 99%, (ii) our GPU, »TITAN RTX« with a maximum GPU load of 99%, (iii) our working memory with 128 GB of RAM, as well as (iv) our hard drive (SSD), »Samsung 970 EVO Plus SSD« with 500 GB. Therefore, future adaptations regarding further performance as well as runtime constraints can be satisfied with only relatively small hardware investments.

Ensuing the in Table 8 and Fig. 11 shown visual fault inspection and performance capabilities, Table 7 depicts the runtimes for the selected model MobileNetV2 for chip- and street-based classification as well as the selected SWWAE for street segment based classification. Additionally, the resulting runtimes for all localization steps as well as the fault backtracing are shown. We would choose MobileNetV2 as a model for a production system as a compromise between high performance and improved processing times. The resulting total for MobileNetV2 as well as all related processing steps shows an acceptable runtime with about 5.2 ms while resulting in a final F1-score of above 99%. Whereas related models for each step can reach comparable scores, they also result in worse accuracy and performance trade-offs.

It is therefore concluded that the selection of the best-fitting model for each processing step does not only results in accuracy- and performance-wise improvements, but also when considering single DNN based approaches, which even without considering greatly performance-depending approaches such as region-based DNNs often result in different performances depending on the task at hand. As an example, the region-based DNN proposed by Wen et al. (2020) utilizes an input image size of 256 × 256 pixels, reaching an average inspection time of 424 ms. Our approach, however, shows inspection times of 5 ms or less per inspected chip with a respective resolution of 192 × 192 pixels (Table 8). Hence, our conclusion to implement a computer vision pipeline instead of utilizing region-based DNNs. Yet, at this point it is also noted that such performance comparisons are only shallow as the differences in the image resolution, used hardware, and inspection requirements all differ. It is out of our performance analysis' scope to provide an in-depth comparison with algorithm performances Dinkelbach et al. (2012).

Visualization

The ground truth of the classified chips and streets as well as the resulting chip, street, and street segment error classes is visualized in Figs. 7 and 8. According to the implemented addressing scheme, this includes the classes of chips, streets, and street segments for flawless (•) as well as faulty (•) occurrences. In addition, anomaly occurrences (•) are highlighted by the visualization system representing suspicious pattern. The classes flawless and faulty originate from the classification of the previous subsystems, while the anomaly pattern is based on our SWWAE's capabilities. The respective defect patterns are visualized via color transitions from green over yellow and orange to red with their increasing degree of faultiness. The color transitions and therefore the degree of faultiness was generated via our SWWAE and is based on the SWWAE-models' resulting loss from evaluating a particular image sample (Sect. 2.3.4).

Whereas the visualized wafer overview with its respective processing steps and resulting chip (Fig. 7a), street (b), and street segment classification (c) is generated fully automatically, the combination of all processing steps is shown in Fig. 7d with the street segment results mapped on top of the street classification results. Additionally, the shown classification results can be further differentiated in terms of their specific defect patterns to furthermore allow an easier and more efficient assessment by an inspector (Fig. 8).

Conclusion and outlook

The designed and implemented automated visual fault inspection system combines the advantages of classical image processing approaches with deep learning based ones in the form of a hybrid multistage system of stacked deep neural networks (SH-DNN). The system draws its strengths from the ability to detect the finest structures which often range within a size of only a few µm and pixels within vast amounts of high-resolution imagery. For this purpose, the multistage system facilitates the focusing of the processing on a region of interest, therefore allowing a subsequent deep neural network

to classify with higher accuracy. A distinction depending on the level of detail enables the detection of defect patterns within larger parental areas while enabling them to be excluded from further assessment within the earlier stages of the automated visual inspection process. Following this principle, the results of every processing step are visualized as based on our proposed addressing scheme, furthermore allowing us to generate our own kind of wafer maps (Figs. 7 and 8). While these wafer maps are visually similar to the ones created by wafer prober tests, they are created risk-free since our inspection process is non-intrusive, which in turn helps to ease the process of further assessment by an inspector.

An evaluation was conducted with different machine and deep learning based approaches while comparing them to more commonly deployed approaches of automated visual inspection. As the results of our created test environment show, the implemented SH-DNN-based system surpasses the current state of the art of automated visual inspection. These improvements are highlighted by an increase of the system's performance (F1-score) from 95 to 98.9% (+3.9%) for conventional baseline approaches as well as from 95.7 to 99.5% (+3.8%) for single DNN based ones. This corresponds to an improvement of the system's fault detection capabilities by 4.6-fold and 8.6-fold, respectively (Table 6).

While our application area has given runtime constraints within the range of only a few milliseconds, the proposed system is able to perform under these constraints. Low complexity classification models are applied, which in turn benefit the system's real-time processing capabilities. Therefore, the system allows the localization and classification of defect patterns to be realized in a more precise and timely manner as compared to single learning-based models, but also enables a more efficient processing in terms of the system's complexity. These benefits are highlighted by the resulting runtimes of only 5.2 ms per chip image sample on current consumer grade hardware, whereas related approaches, including region-based DNNs, often result in much higher runtimes.

To allow a broader application within visual inspection related as well as other domains, future projects can be built on top of this system and enhanced with, e.g., sensor-based analysis such as audio or heat signatures. While our approach of a multistage system is certainly not limited to the inspection of semiconductor wafers, many related as well as mechanical engineering based problems might exist, e.g., Jia et al. (2020), where small defect patterns have to be recognized within larger amounts of data. Furthermore, the optics of the system, including camera and lightning settings, can be investigated and adapted for further application-specific fine-tuning.

Acknowledgements This research is partially funded by the European Social Fund for Germany as well as the German Federal Ministry of Education and Research within the project group localizeIT (funding code 03IPT608X).

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

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

Appendix

Binary classification definitions and terminology

Table 9 shows two definitions for binary classification, whereas Def. (1) is the one mainly used for discussion in this contribution. Binary classification metrics are presented in the following.

Table 9 True Positive (*TP*), False Positive (*FP*), True Negative (*TN*), and False Negative (*FN*) definitions

Def. (1)	Type	Def. (2)
Faulty chip/street classified as faulty chip/street	*TP*	Flawless chip/street classified as flawless chip/street
Faulty chip/street classified as flawless chip/street	*FP*	Flawless chip/street classified as faulty chip/street
Flawless chip/street classified as flawless chip/street	*TN*	Faulty chip/street classified as flawless chip/street
Flawless chip/street classified as faulty chip/street	*FN*	Faulty chip/street classified as faulty chip/street

– Recall / Sensitivity / True Positive Rate (r):

$$r = \frac{TP}{TP + FN} \tag{1}$$

– Specificity / Selectivity / True Negative Rate (s):

$$s = \frac{TN}{TN + FP} \tag{2}$$

– Precision (p) / Positive Predictive Value (PPV):

$$p = PPV = \frac{TP}{TP + FP} \tag{3}$$

– Negative Predictive Value (NPV):

$$NPV = \frac{TN}{TN + FN} \tag{4}$$

– F1-score ($F1$):

$$F1 = 2 \cdot \frac{r \cdot p}{r + p} \tag{5}$$

– Ratio of positives (P) to the total number of observations ($P + N$, with negatives N), also denoted as prevalence:

$$x = \frac{P}{P + N} \tag{6}$$

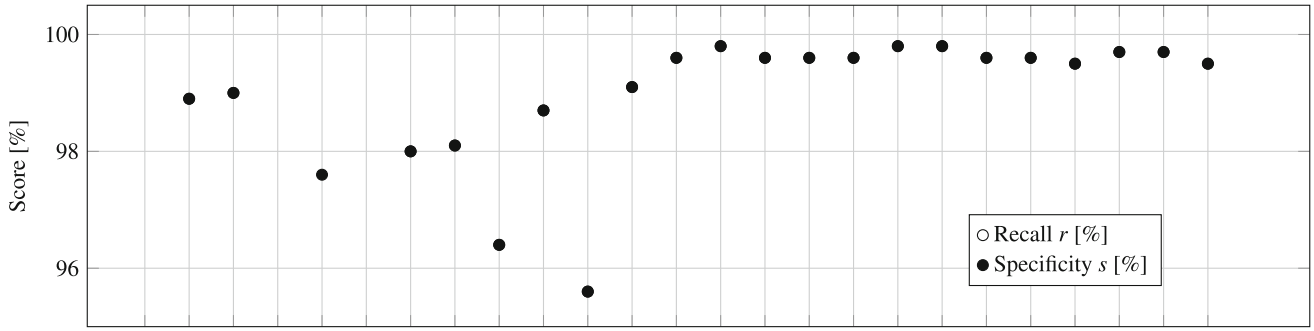
$$\text{Def. (1) in Table 9: } x = \frac{\#\,faulty\ chips/streets}{\sum chips/streets} \tag{7}$$

$$\text{Def. (2) in Table 9: } x = \frac{\#\,flawless\ chips/streets}{\sum chips/streets} \tag{8}$$

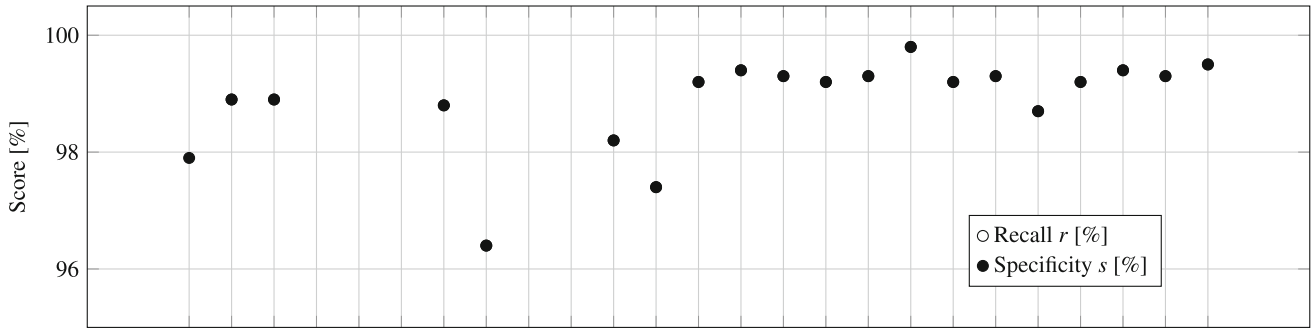
– PPV as a function of x:

$$PPV = \frac{r \cdot x}{r \cdot x + (1 - s) \cdot (1 - x)} \tag{9}$$

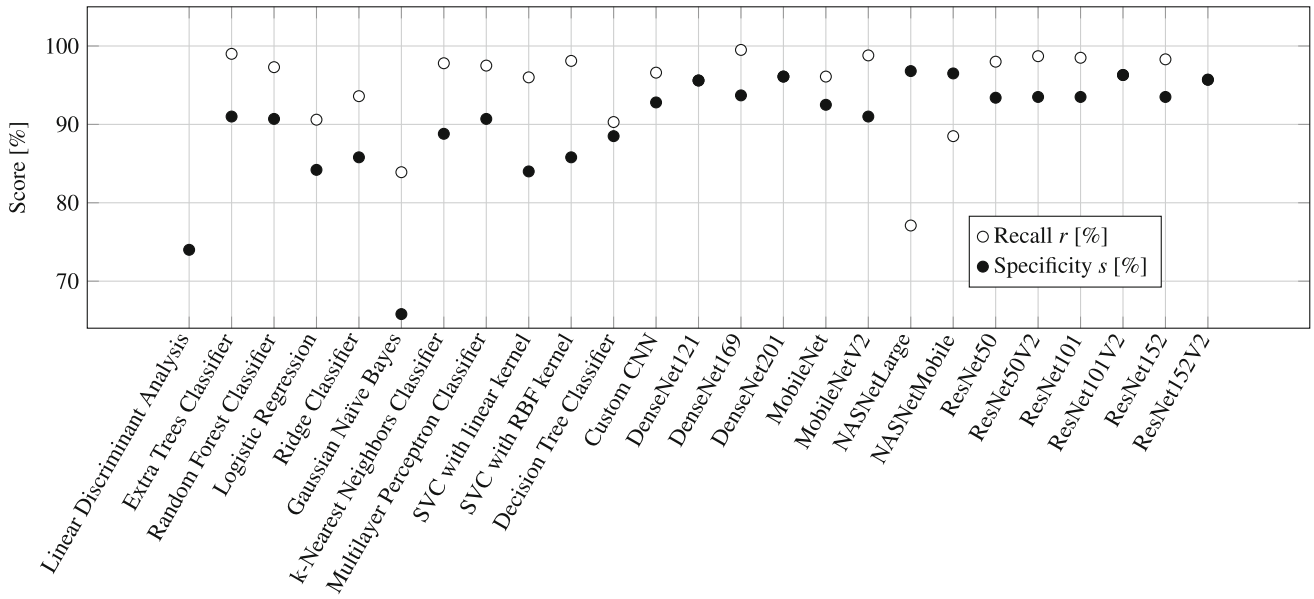
Note that PPV (and therefore $F1$) and NPV depend on the ratio of the (assumed or real) number of faulty streets/chips and the total number of streets/chips (x), while r and s do not. The Python library *scikit-learn* provides us with the two $F1$ metrics, one for each definition introduced in Table 9, denoted by Def. (1) and Def. (2), respectively. In Table 6, the first column ("F1-score") represents the arithmetic mean of $F1$ (Def. 1) and $F1$ (Def. 2), while the second column ("Fault detection F1-score") represents the F1-score following definition (1). Recall (r) and specificity (s) are shown in Fig. 11.



(a) Inside and border chips



(b) Flawless and faulty streets



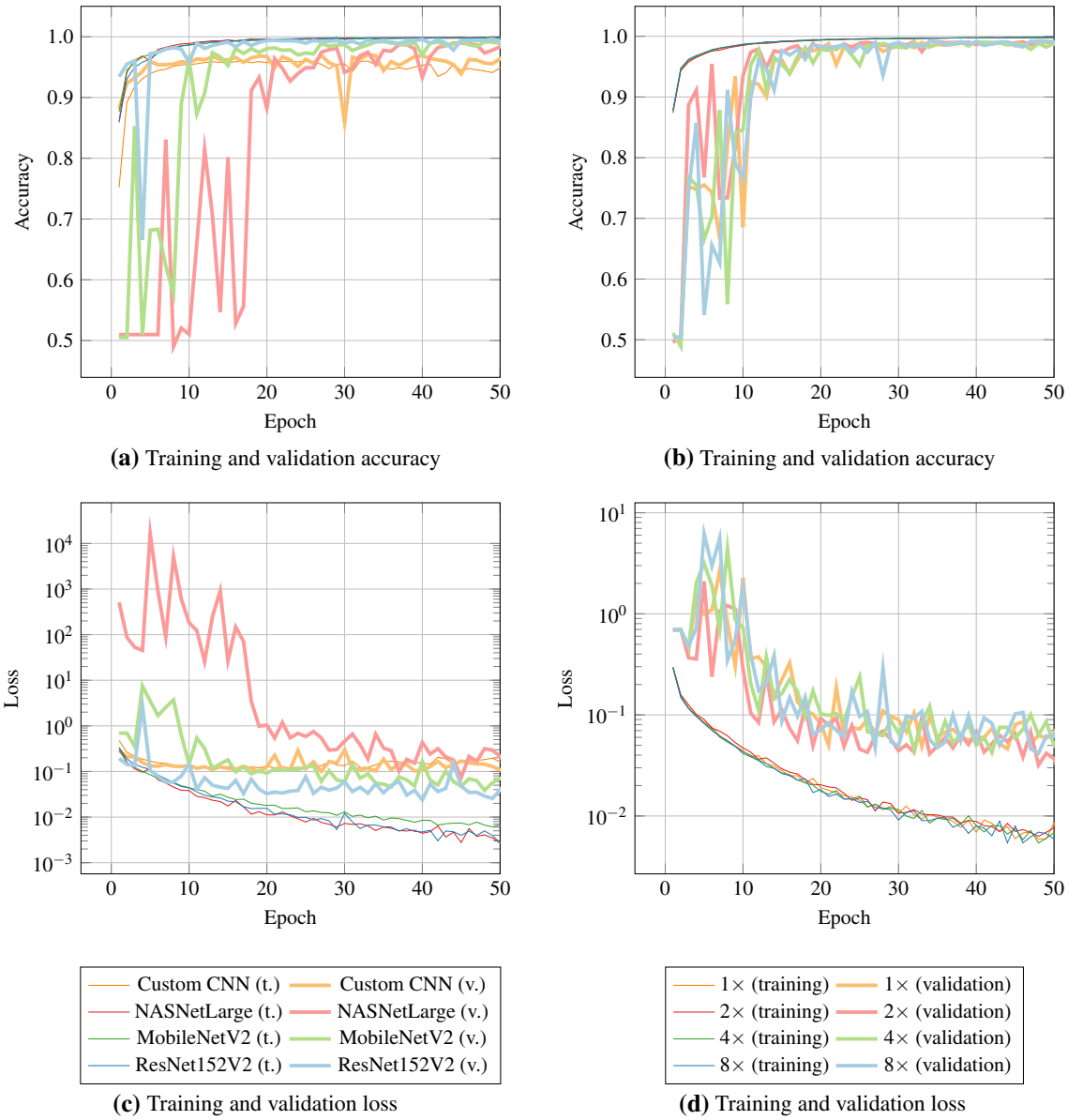
(c) Flawless and faulty chips

Fig. 9 Visualized test results for the SH-DNN-based classification with conventional baseline approaches and DL-based approaches for chip and street classification. Tests for inception module based DL models for street classification where omitted from the evaluation as they expect a minimal image resolution larger than 64×64 pixels

Table 10 Test results for the SH-DNN-based classification with conventional baseline approaches and DL-based approaches for inside and border chip classification

Test run	Chip-based classification: inside and border chips	
	F1-score [%]	Fault detection F1-score [%]
Conventional baseline approaches		
Linear discriminant analysis	69.5 ± 1.2	74.3 ± 1.1
Quadratic discriminant analysis	84.7 ± 1.1	82.5 ± 1.1
Extra trees classifier	**99.3 ± 0.1**	**99.4 ± 0.2**
Random forest classifier	99.0 ± 0.1	99.0 ± 0.1
Logistic regression	94.0 ± 0.5	94.0 ± 0.6
Ridge classifier	95.4 ± 0.6	95.5 ± 0.7
Gaussian Naïve Bayes	79.9 ± 2.2	78.1 ± 2.1
k-nearest neighbors classifier	98.0 ± 0.5	98.0 ± 0.4
Multilayer Perceptron classifier	98.1 ± 0.3	98.1 ± 0.3
SVC with linear kernel	96.4 ± 0.2	96.4 ± 0.2
SVC with RBF kernel	98.7 ± 0.1	98.7 ± 0.1
Decision tree classifier	95.6 ± 0.4	95.6 ± 0.5
DL-based approaches: automated visual inspection approaches		
CNN (2018)*	99.2 ± 0.3	99.2 ± 0.3
CNN (2019)*	99.4 ± 0.3	99.4 ± 0.3
CNN (2020)*	99.5 ± 0.3	99.5 ± 0.3
Custom CNN	99.1 ± 0.2	99.1 ± 0.2
DL-based approaches: general approaches		
DenseNet121 (2017)	99.6 ± 0.2	99.6 ± 0.2
DenseNet169 (2017)	**99.8 ± 0.1**	**99.8 ± 0.1**
DenseNet201 (2017)	99.6 ± 0.1	99.6 ± 0.1
InceptionResNetV2 (2017)	**99.8 ± 0.1**	**99.8 ± 0.1**
InceptionV3 (2016)	99.5 ± 0.2	99.5 ± 0.2
MobileNet (2017)	99.6 ± 0.2	99.6 ± 0.2
MobileNetV2 (2018)	99.6 ± 0.2	99.6 ± 0.2
NASNetLarge (2018)	**99.8 ± 0.1**	**99.8 ± 0.1**
NASNetMobile (2018)	**99.8 ± 0.1**	**99.8 ± 0.1**
ResNet50 (2016a)	99.6 ± 0.2	99.6 ± 0.2
ResNet50V2 (2016a)	99.6 ± 0.2	99.6 ± 0.2
ResNet101 (2016a)	99.5 ± 0.2	99.5 ± 0.2
ResNet101V2 (2016a)	99.7 ± 0.1	99.7 ± 0.1
ResNet152 (2016a)	99.7 ± 0.1	99.7 ± 0.1
ResNet152V2 (2016a)	99.6 ± 0.1	99.7 ± 0.1
VGG16 (2015)	33.3 ± 0.0	66.6 ± 0.0
VGG19 (2015)	33.3 ± 0.0	0.0 ± 0.0
Xception (2017)	99.7 ± 0.1	99.7 ± 0.1

* denotes currently deployed single DNN based approaches within the domain of automated and semi-automated visual inspection



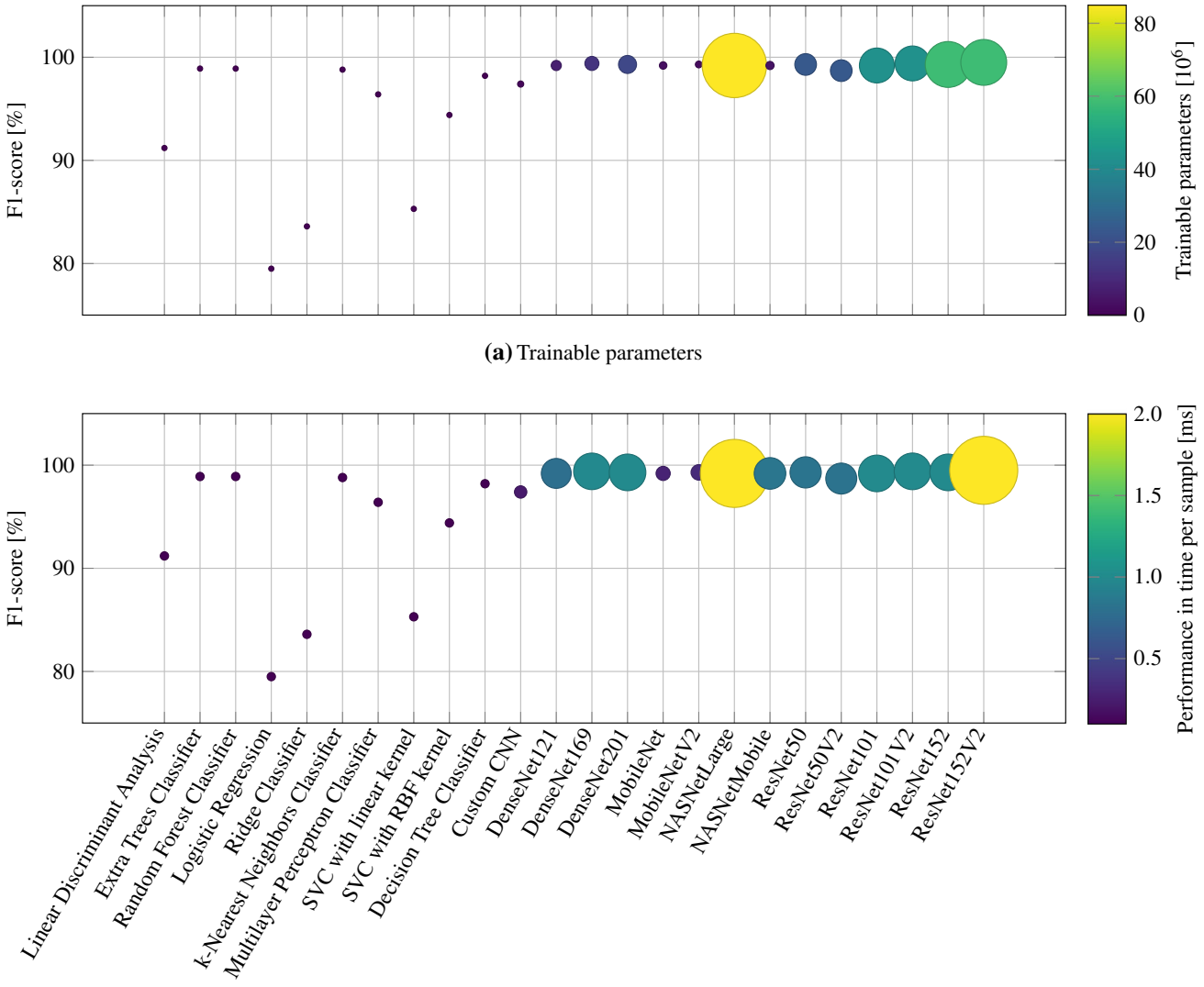
(a) Training and validation accuracy

(b) Training and validation accuracy

(c) Training and validation loss

(d) Training and validation loss

Fig. 10 Street classification training (t.) and validation (v.) results without data augmentation (left) for our custom CNN, NASNetLarge, MobileNetV2, and ResNet152V2, as well as with data augmentation (right) for MobileNetV2



(a) Trainable parameters

(b) Performance in time per sample

Fig. 11 Visualized fault detection F1-score for street classification as dependent on the models' **a** complexity (trainable parameters) and **b** performance in time per sample (color key). For the related numeric test results, see Tables 6 for the F1 scores and 7 for the trainable parameters

References

Altman, D. G., & Bland, jM. (1994). Statistics Notes: Diagnostic tests 2: Predictive values. *BMJ, 309*(6947), 102.

Baldi, P. (2012). Autoencoders, Unsupervised Learning, and Deep Architectures. *Proceedings of ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings, 27*, 37–50.

Belgardt, C., Kosuch, R., Lewke, D., Grimm, M., & Zühlke, U. (2017). Fast high yield cutting of 4 and 6 inch SiC-wafer using thermal laser separation (TLS). *Lasers in Manufacturing Conference, 2017*, 2–7.

Beuth, F. (2019). *Visual attention in primates and for machines - neuronal mechanisms*. PhD thesis, Technische Universität Chemnitz, Germany.

Beuth, F., & Hamker, F. H. (2015). A mechanistic cortical microcircuit of attention for amplification, normalization and suppression. *Vision Research, 116*(Part B), 241–257. https://doi.org/10.1016/j.visres.2015.04.004

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: *2020 46th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, https://doi.org/10.1109/IECON43393.2020.9255234.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

Carrasco, M. (2011). Visual attention: the past 25 years. *Vision Research, 51*(13), 1484–1525. https://doi.org/10.1016/j.visres.2011.04.012

Chao-Ton, Su., Yang, Taho, & Ke, Chir-Mour. (2002). A neural-network approach for semiconductor wafer post-sawing inspection. *IEEE Transactions on Semiconductor Manufacturing, 15*(2), 260–266. https://doi.org/10.1109/66.999602

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

Cheng, K. C. C., Chen, L. L. Y., Li, J. W., Li, K. S. M., Tsai, N. C. Y., Wang, S. J., Huang, A. Y. A., Chou, L., Lee, C. S., Chen, J. E., et al. (2021). Machine Learning-Based Detection Method for Wafer Test Induced Defects. *IEEE Transactions on Semiconductor Manufacturing, 34*(2), 161–167.

Cheon, S., Lee, H., Kim, C. O., & Lee, S. H. (2019). Convolutional Neural Network for Wafer Surface Defect Classification and the Detection of Unknown Defect Class. *IEEE Transactions on Semiconductor Manufacturing, 32*(2), 163–170. https://doi.org/10.1109/TSM.2019.2902657

Cho, H., & Park, W. S. (2002). Neural network applications in automated optical inspection: state of the art. *Algorithms and Systems for Optical Information Processing VI, SPIE, 4789*, 224–236. https://doi.org/10.1117/12.455971

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, IEEE, (pp. 1251–1258).

Dinkelbach, H. Ü., Vitay, J., Beuth, F., & Hamker, F. H. (2012). Comparison of GPU- and CPU-implementations of mean-firing rate neural networks on parallel hardware. *Network: Computation in Neural Systems, 23*(4), 212–236.

Chen, Fei-Long., & Liu, Shu-Fan. (2000). A neural-network approach to recognize defect spatial pattern in semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing, 13*(3), 366–373. https://doi.org/10.1109/66.857947

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning, 29*(2–3), 131–163.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, (pp. 249–56).

Hamker, F. H. (2005). The emergence of attention by population-based inference and its role in distributed processing and cognitive control of vision. *Computer Vision and Image Understanding, 100*(1), 64–106. https://doi.org/10.1016/j.cviu.2004.09.005

Han, H., Gao, C., Zhao, Y., Liao, S., Tang, L., & Li, X. (2020). Polycrystalline silicon wafer defect segmentation based on deep convolutional neural networks. *Pattern Recognition Letters, 130*, 234–241.

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep Residual Learning for Image Recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 770–8).

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Identity Mappings in Deep Residual Networks. In: *European Conference on Computer Vision*, (pp. 630–45), 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: *Proceeedings of the IEEE 65th Electronic Components and Technology Conference (ECTC), IEEE*, (pp. 1436–1446). https://doi.org/10.1109/ECTC.2015.7159786.

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. arXiv preprint arXiv:1704.04861.

Hsu, C. Y., & Chien, J. C. (2020). Ensemble convolutional neural networks with weighted majority for wafer bin map pattern classification. *Journal of Intelligent Manufacturing*, 1–14.

Hsu, C. Y., & Liu, W. C. (2021). Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing, 32*(3), 823–836.

Huang, C. J. (2007). Clustered defect detection of high quality chips using self-supervised multilayer perceptron. *Expert Systems with Applications, 33*(4), 996–1003. https://doi.org/10.1016/j.eswa.2006.07.011

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*, IEEE, (pp. 4700–4708).

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

Hyun, Y., & Kim, H. (2020). Memory-augmented convolutional neural networks with triplet loss for imbalanced wafer defect pattern classification. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 622–634.

Itti, L., Koch, C., & Niebur, E. (1998). A Model of Saliency-based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(11), 1254–1259. https://doi.org/10.1109/34.730558

Jia, X., Yang, X., Yu, X., & Gao, H. (2020). A Modified CenterNet for Crack Detection of Sanitary Ceramics. In: *Proc 46th Annual Conference of the IEEE Industrial Electronics Society - IECON 2020*, IEEE, (pp. 5311–5316). https://doi.org/10.1109/IECON43393.2020.9254351.

Kim, Y., Cho, D., & Lee, J. H. (2021). Wafer defect pattern classification with detecting out-of-distribution. *Microelectronics Reliability, 122*, 114157.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In: *International Conference on Learning Representations*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, D., Siu, V., Cruz, R., & Yetman, C. (2016). Convolutional neural net and bearing fault analysis. In: *Proceedings of the International Conference on Data Mining (DMIN)*.

Lee, K. B., & Kim, C. O. (2018). Recurrent feature-incorporated convolutional neural network for virtual metrology of the chemical mechanical planarization process. *Journal of Intelligent Manufacturing*, 1–14, https://doi.org/10.1007/s10845-018-1437-4.

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

Li, T. S., & Huang, C. L. (2009). Defect spatial pattern recognition using a hybrid SOM-SVM approach in semiconductor manufacturing. *Expert systems with Applications, 36*(1), 374–385. https://doi.org/10.1016/j.eswa.2007.09.023

Liu, S. F., Chen, F. L., & Lu, W. B. (2002). Wafer bin map recognition using a neural network approach. *International Journal of production research, 40*(10), 2207–2223. https://doi.org/10.1080/00207540210122275

Mahadevan, S., & Shah, S. L. (2009). Fault detection and diagnosis in process data using one-class support vector machines. *Journal of process control, 19*(10), 1627–1639. https://doi.org/10.1016/j.jprocont.2009.07.011

Mei, S., Yang, H., & Yin, Z. (2018). An unsupervised-learning-based approach for automated defect inspection on textured surfaces. *IEEE Transactions on Instrumentation and Measurement, 67*(6), 1266–1277. https://doi.org/10.1109/TIM.2018.2795178

Moganti, M., & Ercal, F. (1998). A subpattern level inspection system for printed circuit boards. *Computer Vision and Image Understanding, 70*(1), 51–62. https://doi.org/10.1006/cviu.1998.0600

Nakazawa, T., & Kulkarni, D. V. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network.

IEEE Transactions on Semiconductor Manufacturing, 31(2), 309–314. https://doi.org/10.1109/TSM.2018.2795466

Nawaz, J. M., Arshad, M. Z., & Hong, S. J. (2014). Fault diagnosis in semiconductor etch equipment using Bayesian networks. *Journal of Semiconductor Technology and Science, 14*(2), 252–261. https://doi.org/10.5573/JSTS.2014.14.2.252

O'Leary, J., Sawlani, K., & Mesbah, A. (2020). Deep Learning for Classification of the Chemical Composition of Particle Defects on Semiconductor Wafers. *IEEE Transactions on Semiconductor Manufacturing, 33*(1), 72–85. https://doi.org/10.1109/TSM.2019.2963656

Rahim, K., & Mian, A. (2017). A Review on Laser Processing in Electronic and MEMS Packaging. *Journal of Electronic Packaging*, *139*(3), https://doi.org/10.1115/1.4036239.

Reynolds, J. H., & Heeger, D. J. (2009). The normalization model of attention. *Neuron, 61*(2), 168–185. https://doi.org/10.1016/j.neuron.2009.01.002

Roy, P., Dutta, S., Dey, N., Dey, G., Chakraborty, S., & Ray, R. (2014). Adaptive thresholding: a comparative study. In: *2014 International conference on control, Instrumentation, communication and Computational Technologies (ICCICCT)*, IEEE, (pp. 1182–1186), https://doi.org/10.1109/ICCICCT.2014.6993140.

Sandler, M., Howard, A., Zhu, M, Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, IEEE, (pp. 4510–4520).

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.

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 IEEE 24th International Conference on Emerging Technologies and Factory Automation (ETFA)*, https://doi.org/10.1109/ETFA.2019.8869311.

Schlosser, T., Beuth, F., & Kowerko, D. (2020). Biologically Inspired Hexagonal Deep Learning for Hexagonal Image Generation. In: *2020 27th IEEE International Conference on Image Processing (ICIP)*, https://doi.org/10.1109/ICIP40778.2020.9190995.

Schmidhuber, J. (2015). *Deep Learning in neural networks: An overview.* https://doi.org/10.1016/j.neunet.2014.09.003

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representations*.

Song, H., Choi, K. K., Lee, I., Zhao, L., & Lamb, D. (2013). Adaptive virtual support vector machine for reliability analysis of high-dimensional problems. *Structural and Multidisciplinary Optimization, 47*(4), 479–491. https://doi.org/10.1007/s00158-012-0857-6

Sreenivasan, K. K., Srinath, M., & Khotanzad, A. (1993). Automated vision system for inspection of IC pads and bonds. *IEEE transactions on components, hybrids, and manufacturing technology, 16*(3), 333–338. https://doi.org/10.1109/33.232061

Suzuki, S., & Abe, K. (1985). Topological Structural Analysis of Digitized Binary Images by Border Following. *Computer Vision, Graphics, and Image Processing, 30*(1), 32–46.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, IEEE, (pp. 2818–2826).

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*.

Tobin, K. W., Jr., Karnowski, T. P., & Lakhani, F. (2001). Integrated applications of inspection data in the semiconductor manufacturing environment. *Metrology-based Control for Micro-Manufacturing, SPIE, 4275*, 31–40. https://doi.org/10.1117/12.429361

Wen, G., Gao, Z., Cai, Q., Wang, Y., & Mei, S. (2020). A novel method based on deep convolutional neural networks for wafer semiconductor surface defect inspection. *IEEE Transactions on Instrumentation and Measurement, 69*(12), 9668–9680.

Werbos, P. J. (1990). Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE, 78*(10), 1550–1560.

Xie, L., Li, D., & Simske, S. J. (2011). Feature dimensionality reduction for example-based image super-resolution. *Journal of Pattern Recognition Research, 6*(2), 130–139.

Xie, L., Gu, N., Li, D., Cao, Z., Tan, M., & Nahavandi, S. (2013). Concurrent control chart patterns recognition with singular spectrum analysis and support vector machine. *Computers & Industrial Engineering, 64*(1), 280–289.

Xie, L., Huang, R., Gu, N., & Cao, Z. (2014). A novel defect detection and identification method in optical inspection. *Neural Computing and Applications, 24*(7–8), 1953–1962. https://doi.org/10.1007/s00521-013-1442-7

Zhang, J. M., Lin, R. M., & Wang, M. J. J. (1999). The development of an automatic post-sawing inspection system using computer vision techniques. *Computers in Industry, 40*(1), 51–60. https://doi.org/10.1016/S0166-3615(99)00009-3

Zhao, J., Mathieu, M., Goroshin, R., & LeCun, Y. (2016). Stacked What-Where Auto-encoders. In: *4th International Conference on Learning Representations (ICLR)*.

Zhao, Z., & Kumar, A. (2018). Improving periocular recognition by explicit attention to critical regions in deep neural network. *IEEE Transactions on Information Forensics and Security, 13*(12), 2937–2952. https://doi.org/10.1109/TIFS.2018.2833018

Zheng, Y., Liu, Q., Chen, E., Ge, Y., & Zhao, J. L. (2016). Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. *Frontiers of Computer Science, 10*(1), 96–112. https://doi.org/10.1007/s11704-015-4478-2

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 8697–8710).