

Review

A Review on Machine and Deep Learning for Semiconductor Defect Classification in Scanning Electron Microscope Images

Francisco López de la Rosa ¹, Roberto Sánchez-Reolid ^{1,2}, José L. Gómez-Sirvent ¹, Rafael Morales ^{1,3}
and Antonio Fernández-Caballero ^{1,2,*}

¹ Instituto de Investigación en Informática de Albacete, Universidad de Castilla-La Mancha, 02071 Albacete, Spain; Francisco.LopezRosa@uclm.es (F.L.d.l.R.); roberto.sanchez@uclm.es (R.S.-R.); jose.gomez@uclm.es (J.L.G.-S.); rafael.morales@uclm.es (R.M.)

² Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha, 02071 Albacete, Spain

³ Departamento de Eléctrica, Electrónica, Automática y Comunicaciones, Universidad de Castilla-La Mancha, 02071 Albacete, Spain

* Correspondence: antonio.fdez@uclm.es

Abstract: Continued advances in machine learning (ML) and deep learning (DL) present new opportunities for use in a wide range of applications. One prominent application of these technologies is defect detection and classification in the manufacturing industry in order to minimise costs and ensure customer satisfaction. Specifically, this scoping review focuses on inspection operations in the semiconductor manufacturing industry where different ML and DL techniques and configurations have been used for defect detection and classification. Inspection operations have traditionally been carried out by specialised personnel in charge of visually judging the images obtained with a scanning electron microscope (SEM). This scoping review focuses on inspection operations in the semiconductor manufacturing industry where different ML and DL methods have been used to detect and classify defects in SEM images. We also include the performance results of the different techniques and configurations described in the articles found. A thorough comparison of these results will help us to find the best solutions for future research related to the subject.

Keywords: computer vision; deep learning; machine learning; defect classification; scanning electron microscope; semiconductor; review



Citation: López de la Rosa, F.; Sánchez-Reolid, R.; Gómez-Sirvent, J.L.; Morales, R.; Fernández-Caballero, A. A Review on Machine and Deep Learning for Semiconductor Defect Classification in Scanning Electron Microscope Images. *Appl. Sci.* **2021**, *11*, 9508. <https://doi.org/10.3390/app11209508>

Academic Editor: Byung-Gyu Kim

Received: 30 August 2021

Accepted: 11 October 2021

Published: 13 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Semiconductors are present in almost all devices used in our daily lives. Therefore, the semiconductor manufacturing industry is continuously growing and gaining importance. According to the STATISTA website, the global semiconductor industry's revenues were about \$429 billion in 2019 [1]. Defect detection and classification is crucial in any manufacturing industry and, obviously, it is also crucial in the semiconductor manufacturing industry, where accurate and cost-effective inspection systems that help discover and classify semiconductor defects early in the manufacturing process are essential to improving revenues in this market sector.

So far, different algorithms have been developed to perform defect detection and classification tasks. In the beginning, when the emerging semiconductor manufacturing industry was less automated, human operators would be able to perform the inspection operations manually with the help of optical microscopes due to the larger size of the defects appearing on the wafers. Then, as the size of the defects decreased, the need for some sort of alternative to help human operators perform the inspection task grew. That growing need was met with the help of computer vision and new microscopy techniques [2]. The typical steps that are commonly followed in order to manufacture a semiconductor wafer are depicted in Figure 1. The manufacturing process is not linear. It is composed of hundreds of repetitions of the operations that collect the figure. A single error in one

of those repetitions could jeopardize the whole manufacturing process [3]. As a result, today it is common for an expert to identify defects by visual judgement using a scanning electron microscope (SEM) after each of these steps along the manufacturing chain. Manual surface inspection methods conducted by quality inspectors suffer from the disadvantages of low efficiency, high labour intensity, low accuracy and low real-time efficiency [4]. These weaknesses can be addressed by using advanced computer vision techniques.

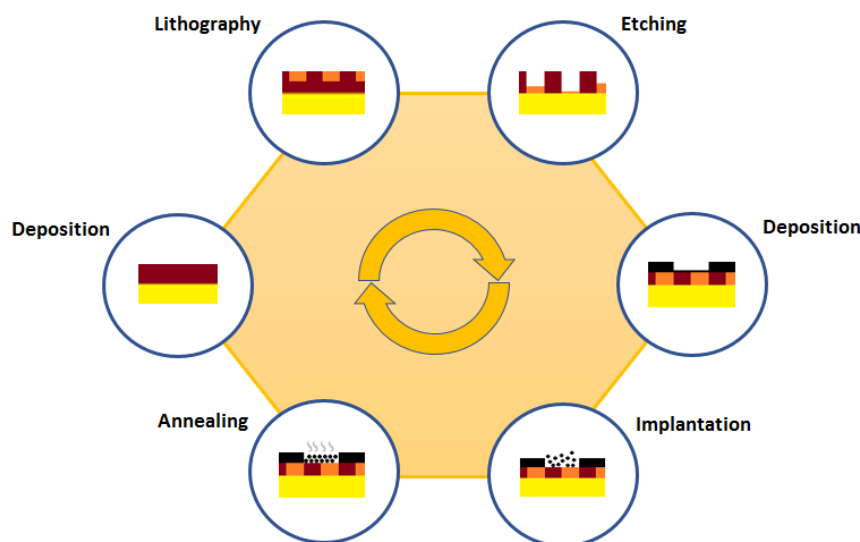


Figure 1. Semiconductor wafer manufacturing steps.

With the advent of innovative machine learning (ML) and deep learning (DL) methods [5], which typically outperform traditional computer vision algorithms when fed with high-volume datasets [6], a revolution is taking place in inspection systems. Novel and traditional methods constitute the state-of-the-art of the inspection systems [7]. In many occasions, traditional methods are used to preprocess the images in order to improve them, extracting regions of interest (ROI) with the purposes of not wasting computational time focusing on redundant data and increasing the detectability of the defects. Then these ROIs are directly fed into the classifiers [8,9]. These approaches in which both traditional and novel methods are combined are commonly referred to as hybrid approaches [10]. Although many ML and DL methods have been used [11], it is now, as a result of the enormous increase in graphics processing unit (GPU) computing power, that they can be applied efficiently, thus achieving high accuracy and high performance ultimately.

This scoping review is intended to collect all ML and DL methods used up to now for the detection and classification of defects in semiconductor wafers from SEM images. The main challenge faced by this review is to shed light on these novel methods that will for sure replace or at least complement and assist human operators in the task of defect detection and classification in the semiconductor industry, moving from a present situation where this task is mostly performed manually to a new reality where detection and classification will be performed automatically. Therefore, this review will allow future authors to have prior knowledge on the best performing models. This will contribute to the growing trend of applying these techniques to the semiconductor manufacturing industry in defect detection and classification, improving the results obtained and increasing both revenue and efficiency in the industry.

The paper is structured as follows. First, the methodology for searching the associated literature will be presented in Section 2. Then, we will focus on the fundamentals of the SEM in Section 3. Next, different ML methods will be analysed in Section 4. After that, DL methods and, fundamentally, the most typical CNN models and the general components of these networks will be introduced in Section 5. Later, the results and discussion will be conducted in Section 6. Finally, we will end this review with the conclusions in Section 7.

2. Search Methodology

In this section, the search methodology will be described in detail. As mentioned previously, the aim of the scoping review is to collect all the possible information regarding ML and DL methods for semiconductor defect detection and classification using SEM images.

2.1. Search Strategy

In order to collect the necessary bibliography for this review, three different databases (Scopus, IEEE Xplore and ACM Digital Library) were selected. The search terms used to obtain the desired articles are shown in Table 1. Instead of directly searching in title, abstract and keywords, the terms have been searched in the full text of the papers in order to perform a more exhaustive search.

Table 1. Search terms.

Search Term	Description
defect OR flaw OR imperfection OR fault OR crack OR bug OR deficiency	Synonyms for defect
detection OR detecting OR recognition OR recognising OR identification OR identifying	Synonyms for detection
classification OR classifying OR categorising OR categorisation	Synonyms for classification
vision OR visual OR image	Screening the articles which work with visual detection
wafer OR semiconductor	Seeking articles in which the defects appear in semiconductor wafers
SEM OR “scanning electron microscope” OR “scanning electron microscopy”	Articles with SEM as inspection device
“deep learning” OR “machine learning”	Articles in which defects are classified using these techniques

2.2. Inclusion and Exclusion Criteria

With the purpose of refining the results of the previously mentioned search, we have introduced some inclusion and exclusion criteria, which are listed below.

2.2.1. Inclusion Criterion

- Every publication, from inception to year 2020, that faces the semiconductor defect detection and classification task by means of a deep learning or a machine learning approach starting from a dataset composed by SEM images must be included.

2.2.2. Exclusion Criteria

- We will include just one copy per publication, removing duplicates.
- Publications that do not exclusively use SEM images in the dataset will be excluded.
- Articles that do not use any deep learning or machine learning technique will be excluded.
- Articles that do not perform the defect detection and classification task on semiconductor images will be excluded.

2.3. Refined Results Acquisition Procedure

Figure 2 schematically shows the application of the inclusion and exclusion criteria. The process of refinement was carried out in three stages: initial search, selection and inclusion. Throughout the initial search stage, the raw results of the databases were collected. Up to 224 articles were found in total, 219 in Scopus, 5 in IEEE Xplore and none in the ACM Digital Library. In the selection stage, the duplicated documents were

eliminated. As many as 3 documents were duplicated. Then, another 143 documents were removed after reading the abstract, title and keywords. As the terms were searched throughout the full document, those 143 articles clearly did not address the objective of the review. Finally, at the inclusion stage, 70 more papers were removed from the list after careful reading. Another 29 of them were eliminated for not using SEM images. In fact, SEM was mentioned throughout the article, but other devices were used to construct the data sets. Another 19 articles were removed because they did not perform defect detection on semiconductor materials. Again, semiconductor materials were mentioned in the full text, but defect detection and classification was devoted to other types of materials and defects. Finally, 22 articles that did not employ machine or deep learning were removed. These techniques were mentioned in the articles as future work or an alternative. Therefore, solely 9 of the 224 initial documents remained on the list. The fact is that this may seem like a really low number of articles. However, to the authors' knowledge, all papers related to the subject of the review have been collected. These 9 documents will be used for the detailed study.

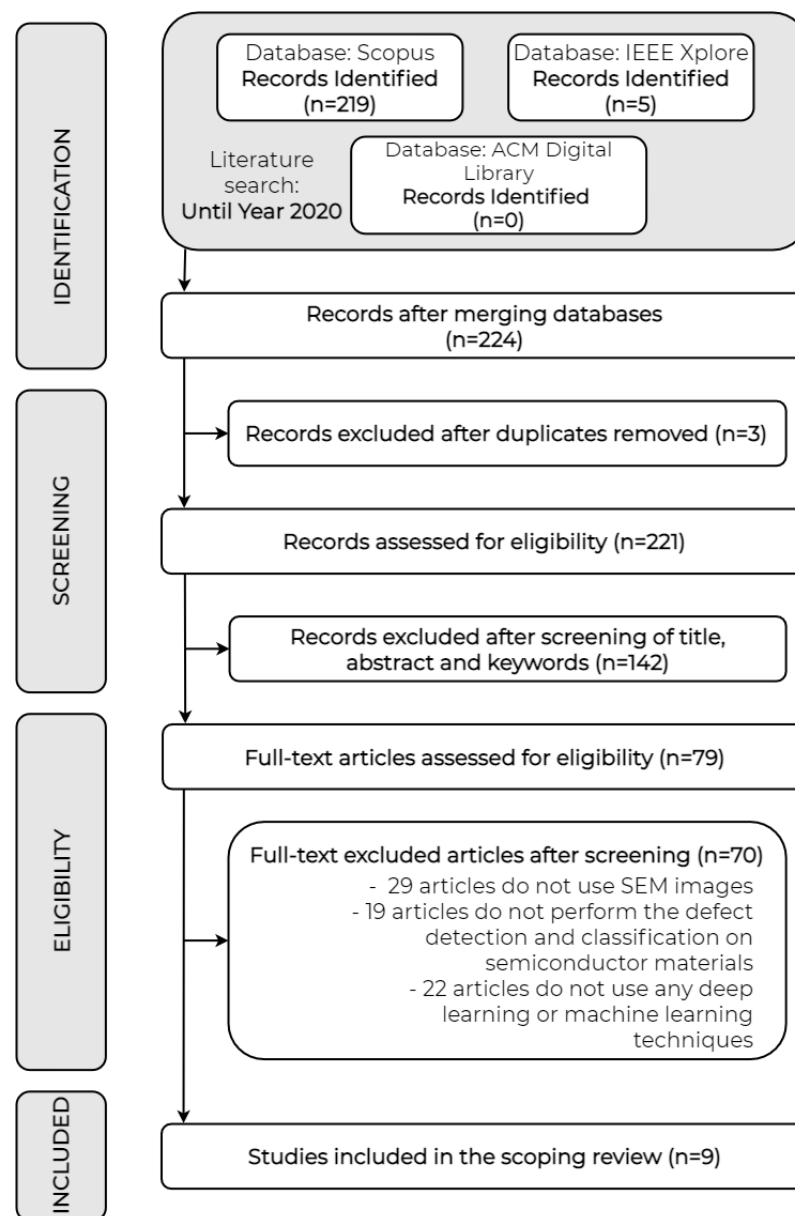


Figure 2. Search methodology.

2.4. Research Questions

The documents found during the search should help us to answer the following research questions:

- Which ML methods achieve the best performance in the detection and classification of semiconductor defects from SEM images?
- Which DL methods achieve the best performance in the detection and classification of semiconductor defects from SEM images?

The performance of the different approaches will be evaluated by means of the accuracy metric in Section 6.

3. Scanning Electron Microscopy

As mentioned above, scanning electron microscopy (SEM) plays an important role in the inspection operations of most semiconductor industries [12–16]. Moreover, the authors of this review are willing to work with images captured with this microscopy in the near future. For these reasons, the focus will be on this type of microscopy throughout the scoping review. However, there are other types of microscopy employed during inspection operations, such as optical microscopy, scanning transmission electron microscopy (STEM), and acoustic microscopy, among others. It should be noted that the ML and DL techniques highlighted in this paper when applied to SEM images can also be extrapolated to images containing other kinds of defects or obtained by other types of microscopes such as those mentioned above. Although less used in the semiconductor manufacturing industry, they can offer good results in certain applications that do not require such high performance as provided by SEM. In this section, we will briefly introduce the basics of SEM to obtain an overview of its operation, mentioning its technical characteristics and its main components.

Fundamentals of SEM

The resolution limit is defined as the minimum distance at which two different structures can be separated and distinguished as independent objects. For example, the resolution of the human eye is about 0.1 mm, while the resolution limit of light microscopy is about 2000 Å. The SEM, with the right settings, may well reach the range of 1 to 10 nm. This improvement in resolution allows much more information to be acquired from the images, which will lead to better performance in the process of detecting and classifying defects in general [17].

SEM technology uses a focused electron beam to scan along the surface of the sample, generating a wide range of signals which are fused and converted into a visual signal with the help of a cathode ray tube. Two categories of electron-sample interactions are distinguished: elastic interactions that result from the deflection of the incident electron by the sample's atomic nucleus or by electrons in the outer shell of similar energy without significant loss of energy, and inelastic interactions that occur through various interactions between the incident electrons and the sample's electrons and atoms, resulting in the transfer of substantial energy to that atom by the electron in the primary beam. The elastic interaction generates back-scattered electrons (BSE) and the inelastic interaction generates secondary electrons (SE).

Those electrons, as well as other signals such as X-rays, Auger electrons and cathodoluminescence, are used to form and analyse the image of the sample. For example, BSE provides compositional and topographic information in SEM, SE provides mainly topographic information, X-rays are used to obtain chemical information of the sample, Auger electrons, which have very low energies, are only used in surface analysis, and cathodoluminescence is a mechanism for energy stabilisation. Some key parameters, such as the beam electron energy and the atomic number of the specimen, determine the characteristics of what is called the region of primary excitation. Higher energies lead to deeper penetration lengths, but surface resolution decreases. Higher atomic numbers lead to a lower depth, since the higher the number of particles, the easier it is to stop the penetration of the electrons. Therefore, the goal is to find a balance and, depending on the requirements

of the scenario, determine the optimal beam energy. A schematic figure of an SEM device, including all its components, can be appreciated in Figure 3.

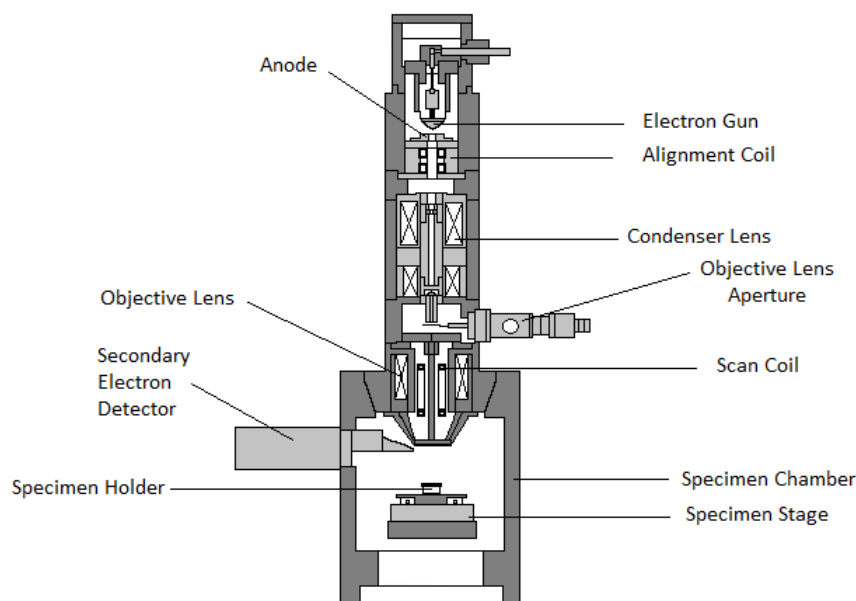


Figure 3. Scanning electron microscope.

The basic components of an SEM device are the electron gun, the lenses, the scan coil and the secondary electron detector. Those elements are going to be briefly described along the following lines. The electron gun task is to produce a stable electron beam with high current and directed to a small spot. There are several types of electron guns such as tungsten electron guns, lanthanum hexaboride guns (LaB6) and field emission guns. Other key components are the lenses. Two kinds of lenses are distinguished: the condenser lenses where the electron beam is converged in a parallel stream, and the objective lenses, which are used to focus the beam into a probe point. In order to form an image, the scan coil deflects the beam throughout the x and y axis to cover all the sample. Finally, the secondary electron detector, another crucial component, collects the secondary electron which, as we mentioned before, has the important role of providing fine topographic information.

4. Machine Learning

With the shrinkage of the products of the semiconductor manufacturing industry, the killer defects that completely ruin the product are also becoming smaller [18]. This fact has encouraged industries to implement high-resolution microscopy techniques, such as SEM explained above, and to develop ML methods that detect and classify these tiny killer defects with great precision. In addition, ML approaches include the important advantage of providing reliable models in noisy real-world industrial environments [19]. ML has the ability to figure out the relationships in large datasets. If the right method is selected and its settings are optimal, defect detection and classification can be achieved easily. The dataset must be divided into different sets. One of them, usually the largest, is used to train the algorithm, another to validate it and the last to evaluate it. ML techniques are mainly classified into three major groups attending their learning strategy: supervised learning, unsupervised learning, and semi-supervised learning [20].

4.1. Supervised Learning

In supervised learning, the strategy is to train a classifier with a set of labelled data. Therefore, the number of categories to classify the dataset is known. Once the system has learned to identify the different patterns, the classifier is able to assign each piece of data to its corresponding category [21]. There are several methods of supervised learning. The

methods that have been applied in the works selected in the search that make use of these supervised methods are introduced. Some of the methods described below have not been used in the selected papers, but we present them as background for future work.

4.1.1. Support Vector Machines (SVM)

SVMs rely on the hyperplane concept to separate a set of data into different classes [22]. The key to performing a correct classification is to obtain the optimal hyperplane, which will be the one that maximises the space between the classes of the dataset. It was initially designed to perform a binary classification. Although it is more complex, nowadays it can be used in multidimensional problems. The most commonly used kernels employed in SVM are linear kernels, quadratic kernels, polynomial kernels, Gaussian kernels and radial kernels. In our search results, we have found a paper using SVM [23]. Notice that SVM is not the main method for defect detection and classification in the paper, but the same dataset is applied to compare the accuracy of several methods, which we consider worth mentioning here.

4.1.2. Decision Trees (DT)

A DT is a hierarchical method where an entry is divided into several branches [24]. The amount of information is augmented with every set of new branches. The goal is to obtain the highest separation between the data of the dataset. In order to achieve this goal, the initial dataset is split by means of binary divisions into branches along several iterations where the entropy is reduced [21,25]. The process ends up when the maximum tree depth is reached or a run-time cut-off is met [26]. There are several DT algorithms such as regression tree, tree medium, and some ensemble methods such as random forest and bagged tree. As for SVMs, we have found one paper that uses a DT algorithm, concretely a random forest algorithm, to compare its accuracy with the principal method employed [27].

4.1.3. K-Nearest Neighbours (K-NN)

K-NN is based on the idea that a prediction regarding a data-point can be made from its neighbours [28]. The letter K makes reference to the number of neighbours that will be used. In order to obtain proper results, we have to set an optimal K value, taking into account that a low value of K will lead to a bad prediction and a large value will lead to poor performance [26]. Throughout our refined search, we found a paper that implements this K-NN algorithm to detect unknown class defects [23].

4.1.4. Naive Bayes

Bayesian classifiers assign the most likely class to a given example described by its feature vector. The learning process in these classifiers can be significantly simplified assuming that the features have a statistical independence [29]. The fact is that, although this assumption is not very realistic, their performance is good enough to compete with more complex existing methods. The probabilities can be adjusted as new input data enter the classifier.

4.1.5. Discriminant Analysis (DA)

DA is a group of methods that assign the probability of a given dataset belonging to a certain class based on the Bayes' theorem [30]. The class each input image will belong to will be the one with the highest probability. There are several DA classifiers. The most important ones are linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) [31].

4.2. Unsupervised Learning

This type of method is based on learning by using an unlabelled dataset. The model obtained is automatically adapted to the observations. The model is mainly created through the use of clustering methods. There are some methods worth mentioning, which

are described below. We are also mentioning the papers from the search that make use of unsupervised methods. Some of the methods described below have not been used in the papers, but we also introduce them as a background for future works.

4.2.1. K-Means

k-means is a clustering method that aims to separate the initial unlabelled dataset into k clusters [32]. The value of k is predefined. A sample belongs to the cluster with the closest mean value to the cluster's mean value. In order to draw a map in our mind, a cluster is represented by a point with a value equal to the mean value of the cluster. The different samples of the cluster will be surrounding that point. The idea is to establish a number of clusters that cover the maximum number of samples while minimising the inertia (the distance between the cluster point and the different samples that belong to that cluster) [21]. During our search, one paper implementing the k-means algorithm to analyse SEM images was found [33].

4.2.2. K-Medoids

This clustering method relies on the medoid concept [34]. A medoid, in this case a cluster medoid, is the point of the cluster whose value is the closest to the mean value of the whole cluster, or, in other words, the most central point of the cluster. k-medoids is a very robust method with low sensitivity to noise and disturbance.

4.2.3. Self-Organising Maps (SOM)

A SOM or self-organising neural network is an unsupervised clustering method that allows to map the high-dimensional input data to a low-dimensional field (usually bi-dimensional) without losing the topological structure [35]. The maintenance of the topological structure is performed by applying a neighbourhood function [21]. Throughout our search, we also found one paper that implements an SOM method to perform the defect detection and classification task [36].

4.3. Semi-Supervised Learning

The types of methods that use the semi-supervised learning strategy are trained both with labelled and unlabelled data. The most common approach is to first train the model with labelled data and then continue training with unlabelled data. Usually the size of the labelled batch is much smaller than the size of the unlabelled data. Semi-supervised learning also receives the name of reinforcement learning.

5. Deep Learning

Deep learning (DL) is generally handled as a particular branch of ML [37]. Deep learning has come to substitute the “manual” feature extraction process by an autonomous and efficient feature extraction one [38]. Several DL techniques such as recurrent neural networks (RNN), restricted Boltzmann machines (RBM), autoencoders (AE), variational autoencoders (VAR) and convolutional neural networks (CNN) have already been applied in smart manufacturing [39,40]. However, as revealed by our search, the most used techniques for inspection are CNN with doubt [41–43]. CNN are deep structured neural networks that are mainly employed in image processing tasks [44]. The different elements that compose these networks will be highlighted and the various CNN models that can be used for defect detection and classification will be listed. CNN are inspired by the human visual perception system, in particular by the visual cortex, which is the element of the brain in charge of detecting and identifying objects.

When we discussed ML techniques, it was explained that large datasets with hundreds of images of each class were needed to obtain accurate results. With DL techniques, not large but huge datasets with thousands of images are needed to obtain accurate results. Fortunately, the initial dataset can be increased by applying data augmentation techniques based on image transformations (which include operations such as flipping,

noising, padding, scaling, and cropping [45]) or even performing the so-called One-Channel Augmentation (which consists in adding random pixel noise, brightness adjustment, blur, edge extraction, etc.) [46] in order to obtain a rich and sufficient dataset. Even some CNNs such as generative adversarial networks (GAN) [47] and conditional convolutional variational autoencoder (CCVAE) [48] can be used to perform this task. As for ML, in DL, the data set is divided into three datasets for the learning phase: a training dataset (usually the largest one), a test dataset and a validation dataset. Early stopping conditions must be included with the aim of avoiding overfitting and saving computing time [49]. The main approach is to monitor one of the metrics involved in training (such as loss or accuracy) and stop training if there is no improvement after a given number of epochs.

After this brief introduction, the structure of this section is explained. First, the general components that usually appear in CNN will be presented. Next, the different CNN typically used for defect detection and classification will be introduced. Finally, the two existing approaches for carrying out defect detection and classification will be discussed.

5.1. Elements of a CNN

5.1.1. Neurons

The neuron is the smallest element on a CNN. Each neuron has a weight that is modified during learning. This weight, in combination with different biases, is used during the training of the different network layers. If the input image is a grayscale image, there are as many neurons as pixels in the image. If the input image is an RGB image, there will be one neuron per pixel and channel, i.e., the number of neurons will be three times the number of pixels in the RGB image.

5.1.2. Layers

Convolutional Layers

A set of interconnected neurons builds the so-called convolutional layer, another key component of these networks. CNN usually incorporate several convolutional layers, which are the most relevant elements for feature extraction. The first layers are used to extract low-level features and, as one goes deeper into the convolutional layers, high level features are obtained [50]. The trend is to move towards deeper solutions to improve the accuracy of the CNN. The features are extracted by performing the convolution operation, where a group of input pixels (the initial image or the result of the previous convolution) are multiplied (scalar product) by a small matrix called kernel. The kernel, which slides through all the input pixels, contains weight factors to store extracted features [51]. Those weights, which are at first random, are usually adjusted by applying the backpropagation algorithm, which minimises the cost function through several iterations. The resulting matrix obtained from all convolution operations is known as a feature map.

There are different parameters that must be set up for each convolution. The three that are explained next are the most important ones.

1. **Kernel size:** is the first parameter that needs to be established in a convolutional layer. There is a wide range of options but, commonly, the most used sizes are 1×1 , 3×3 and 5×5 .
2. **Stride:** is a parameter that defines the step size of the kernel. For example, if the stride has a value of 3, the kernel will move 3 pixels horizontally after each convolution operation. Typical values for the stride are 1, 2 and 3.
3. **Depth:** indicates the number of kernels that are used in each convolution. Each kernel generates a feature map, and the totality of the feature maps receives the name of feature mapping. The most used approach is to start with a few kernels along the first layers and continue increasing this number until the last convolutions.

Activation Layer

The activation layers, also known as non-linearity layers, are crucial for enhancing the performance of CNN. As real data are non-linear, it is necessary to introduce non-linearity

in the dataset. The activation layers are usually introduced after each convolution. There are several functions that are used to introduce this non-linearity, but maybe the most important ones are ReLU where the negative pixel values are replaced by zero, and sigmoid that usually returns a value between 0 and 1.

Pooling Layer

A pooling layer, used for sub-sampling, usually comes after the activation layer. Without sub-sampling, the volume of data would grow up continuously convolution after convolution, requiring a huge computational effort and, consequently, much more processing time. Sub-sampling aims to perform a filtering operation throughout the input data (the data coming from the corresponding activation layer) in order to eliminate a large part while maintaining the most relevant information. Therefore, only the relevant information will feed the next convolutional layer. This operation also normalises the data, which means that the variance of the data is reduced. In these layers, as in the convolutional ones, the stride and the size of the kernel or filter must be determined. A typical value of two is usually adopted for both parameters. There are different pooling or sub-sampling methods. The most relevant ones are briefly exposed below:

1. **Average pooling:** offers the mean value of the sub-sampled pixels as the output value.
2. **Max pooling:** offers the highest value of the sub-sampled pixels as the output value.
3. **Other methods:** are not as popular as the previous ones. The reason is that they are more specific methods that offer a great performance under certain particular scenarios. Some examples are mixed pooling, stochastic pooling, spatial pyramid pooling (SPP) or region of interest pooling (ROIP).

Figure 4 shows a graphical explanation of the operation of the max and average pooling algorithms. In this case, a 2×2 kernel and a stride with a value equal to 2 is used. It can be appreciated that max pooling provides the highest value of the kernel window, while average pooling yields the average value of the kernel window.

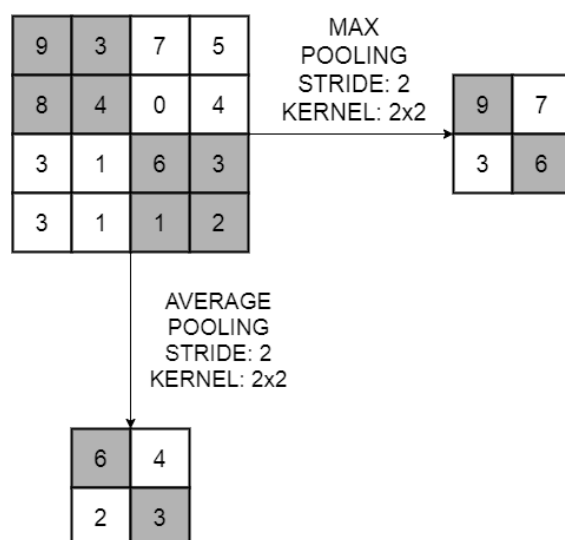


Figure 4. Max pooling and average pooling algorithms.

Fully Connected Layer

The fully connected (FC) layers are connected to each of the neurons of the previous layer. These layers perform a high-reasoning operation, reducing the multidimensional data of the previous layers to a $(1 \times 1 \times N)$ vector, where N is the number of output classes. Neurons in these layers also have an associated weight, which, as with the convolutional layer neurons, is adjusted after several iterations by the backpropagation algorithm. To avoid overfitting and to boost network performance, the dropout algorithm is included in these layers. The purpose of the algorithm is to eliminate the connections of the FC layer

with neurons whose weight is equal to zero, so that neurons having a zero contribution to the determination of a certain output class do not participate in the calculation. This reduces training time and contributes to a lighter system.

Classification Layer

The 1D vector obtained from the FC layers feeds the classification layer. The classification layer transforms the vector elements into probabilities of belonging to each class. The most common function in the classification layer is softmax. The probabilities obtained can be used directly to perform the classification, approximating the highest probability to 1 (the element belongs to that class) and the rest of the probabilities to 0 (the element does not belong to the rest of the classes) [52]. Another approach is to use an ML classifier such as the ones explained above to make the final classification. These classifiers may substitute the softmax function and also be complementary to it. Several classifiers can even be applied in parallel to compare results and receive feedback to help improve the reliability of the results.

5.1.3. Convolutional Neural Network Models

The different elements that build convolutional neural networks (CNN) have to be combined to obtain good solutions. Undoubtedly, the number of configurations is almost infinite. However, there is a good number of CNN configurations that have demonstrated excellent results. These CNNs are called backbone networks, since they are the basis of the different architectures that are implemented to solve, among others, the problem of defect detection and classification.

In this section, an overview of some widely used CNNs will be provided, mentioning the works from our search that make use of such architectures. Again, some backbone networks that are not used in any of the works we have found during the search will be added to provide background for future projects.

AlexNet

AlexNet is a pioneer within CNN [53]. It is very simple in comparison to the latest models. AlexNet is composed of five convolutional layers, a pooling layer (max pooling), three FC layers and an activation layer using a ReLU function. It is not very common to find applications that use AlexNet today, but it is still very important because it was precisely that CNN, which inspired the rest of the models [52].

VGGNet

VGGNet won the localisation and classification tracks of the ILSVRC competition in 2014 [54]. Therefore, we are talking about a powerful network. It has two versions that are widely used, VGG-16 and VGG-19. VGG-16 is composed of 13 convolutional layers using 3×3 kernels, 5 pooling layers and 3 FC layers. VGG-19 includes three more convolutional layers. As the network structure is somewhat simple and its performance is great, it has been used as a backbone network to develop many different applications [50,52]. In the search carried out in our scoping review, we found a couple of documents in which VGGNet was implemented for the task of defect detection and classification [55,56].

GoogleNet

GoogleNet or Inception V1 [57] is an example of an inception network. It presents two main advantages. The first is a significant reduction in the parameters that the network must manage (about twelve times less than AlexNet). The second is the implementation of inception modules, which allow several convolution operations to be performed using different kernels on the same layer. It is necessary to consider that although the depth of the network is twenty-two convolutional layers, taking into account the layers within the inception modules, there would be more than fifty convolutional layers. It also replaces the max pooling algorithm with an average pooling algorithm in the last layers.

A figure has been prepared to visualise how the inception modules operate. Figure 5 shows the parallel convolution operations that can be performed in the same layer with the help of the inception modules [52]. Moreover, there have been several advances from the initial Inception V1. Each new model is a step beyond the previous one in terms of precision and processing time, exceeding the previous model. Some of the new models are Inception V2, Inception V3, Inception V4, Inception-ResNet V1 and Inception-ResNet V2. An article has been found in our search implementing Inception V2 [58], another one with Inception V3 [59] and another one with Inception-ResNet V2 [55].

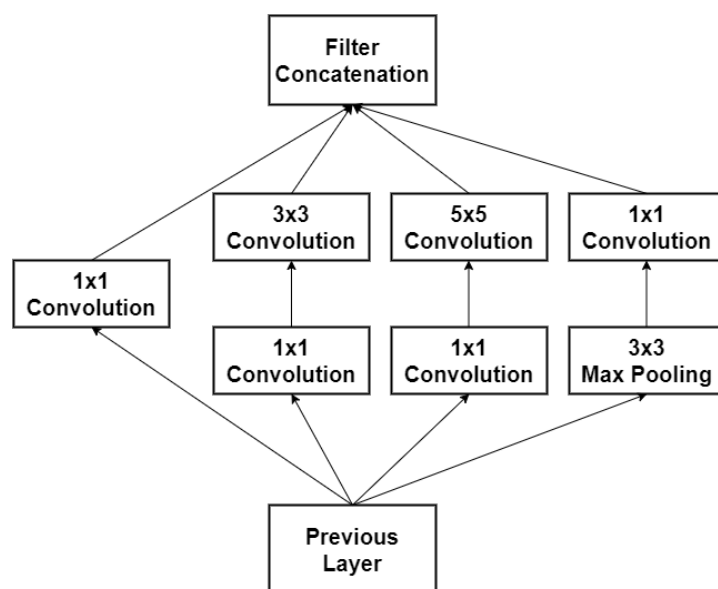


Figure 5. Inception module structure: parallel convolutions.

ResNet

The advent of ResNet revolutionised the world of CNN [60]. ResNet won the Best Paper Award at the Computer Vision and Pattern Recognition Conference in 2016. It has a depth of 152 layers, which is achieved through a novel idea. Each layer does not have to adjust every single weight of its neurons but only learns a residual correction of the previous layer. This depth allows for huge accuracy. In fact, ResNet is considered one of the most accurate models on CNN. Therefore, it is widely used for tasks that require detecting and classifying very small objects or, as in our case, defects [50]. Throughout our search, we have found one paper that implements the ResNet model for detecting and classifying defects [55,59].

MobileNet

The main difference between MobileNet and the models mentioned above is the replacement of the traditional convolution operation [61]. The novel convolution can be divided into two: a point-wise convolution (carried out by means of 1×1 kernels) and a depth-wise convolution. As a result, the size of the model is smaller and the complexity is much lower. As accuracy is not affected, MobileNet outperforms two contrasted networks such as VGGNet and GoogleNet [52].

EfficientNet

Originally developed back in 2019 [62], EfficientNet was born to solve the scaling problems of the state-of-the-art convolutional neural networks. Up to that time, there were three different ways to perform the scaling of a network with the aim of gaining precision while classifying: going deeper (which involves more layers), going wider (more channels) or working with the image resolution. Those scaling techniques were carried out individually. EfficientNet performs what is known as a compound scaling, where up to

three constants are determined by means of a grid search, giving the user the possibility of determining a fourth to adapt the network to the equipment's capacities. Starting from a base model (EfficientNetB0) that includes an input convolutional layer, seven mobile inverted bottleneck blocks (MB Convolution) [63] and a fully connected layer, the different models (ranging from 1 to 7) are obtained through this compound scaling operation.

Other Models

As mentioned above, there are many suitable CNN configurations with acceptable results. Of the documents analysed in the scoping review, some do not use any of the above models but built their own ones. Some authors even compared the performance of their models with the previous ones. The configuration of these models will be briefly discussed in the results and discussion section. Moreover, several articles implementing CNN models designed by the authors have been selected in the scoping review [23,27,64].

5.1.4. Other Configurable Parameters

In this subsection, two different configurable parameters of convolutional neural networks will be explained: loss functions and optimisers.

Loss Functions

The loss in a CNN is, basically, the error in which the CNN incurs while predicting. That error can be computed by means of different loss functions. Within all the existing loss functions, there are five that should be highlighted due to their common use in this CNN field: mean absolute error (MAE) (mainly used in regression models), mean square error (MSE) (as MAE, used in regression models), binary cross-entropy (BCE) (which is designated for binary classification issues), categorical cross-entropy (CCE) (multi-class classification tasks) and sparse categorical cross-entropy (SCCE) (also for multi-class classification issues).

Optimisers

The optimisers or optimisation algorithms are crucial parameters of a CNN. They determine the training speed and, definitely, the final performance of the CNN [65]. The optimiser is commonly chosen by using a grid search, in which different algorithms are tested. The most employed optimisers are stochastic gradient descent (SGD) [66] Momentum [67], Nesterov [68], RMSProp [69] and Adam [70].

5.2. One-Stage and Two-Stage Approaches

The models presented may be implemented to solve different problems. As for the detection and classification of defects, two approaches are distinguished:

- One-stage approaches or classification-based methods. Detection and classification (for example, of defects) is carried out simultaneously in a single stage. The main objective of the approach and its greatest advantage is the detection and classification in real-time. The disadvantage of this approach, compared to the two-stage approach, is that its accuracy is significantly lower. Therefore, it is focused on tasks that must be agile or fast and that do not require high accuracy. An example is the YOLO (You Only Look Once) architecture. YOLO can use different CNN models as a backbone such as VGGNet and GoogleNet.
- Two-stage approaches or region proposal-based methods. The detection and classification tasks are carried out separately. First, a network generates proposals of regions for object detection, and then a different network is fed with those region proposals to definitively locate and classify the object (in our particular case, the defect). Since the detection and classification task is executed in two stages, the time required to perform them is greater than in the single-stage approach. Despite this increase in time, the approach is very popular for tasks that require high accuracy. This approach can be considered for future works if, for instance, the location of the defect is not

sufficiently clear. Along the first stage, the defects would be located, while in the second one they would be classified. Further information can be found in [71]. An example of this approach is region-based CNN (R-CNN) and its variants, considered one of the best architectures in terms of accuracy.

6. Results and Discussion

In this section, we will present and discuss the results obtained when applying ML and DL methods to the defect detection and classification task in the papers selected during our search. This will allow us to obtain an overview of the performance of each method implemented and to select those methods that provide a better performance for the development of future projects. Certainly, the various papers have used a different dataset, and the technology available at the time of the research was not the same. Therefore, although the results obtained through the different methods can be compared, they must be judged according to the available technology and the quality of the dataset in each publication. The results can be observed in Table 2.

Table 2. Methods and their metrics.

Reference	Method	Accuracy
[23]	CNN (self design)	0.962
	SVM (radial basis function)	0.925
	K-NN	0.933
[27]	CNN (self design)	0.953
	Random forest	0.942
[33]	K-means	—
[36]	SOM	*
[55]	Inception V2	0.900
	ResNet 50	0.875
	VGGNet16	0.844
	R-Inception V2	0.974
	R-ResNet 50	0.968
	R-VGGNet16	0.960
[56]	CNN (self design)	0.821
[58]	Inception V1	0.873
	Commercial ADC	0.772
[59]	Inception V2	0.600
	ResNet 50	0.700
[64]	CNN Back-propagation	1
	CNN Linear Vector Quantisation	1
	CNN Radial Basis Function	0.900

Note: * The results are presented in terms of sensitivity and specificity, not accuracy.

From Table 2, it can be observed that the accuracy metric has been chosen to compare the papers. The main reason is that almost all papers from the reviewed literature use this metric to evaluate their models. However, there are two exceptions. In the first one [33], there is no numerical result because the paper focuses on defect detection and quantification. Despite this, the paper has been included due to the feature extraction and classification that is performed with the K-means clustering method. For the second exception [36], different metrics have been used to evaluate the SOM model. Those metrics are sensitivity and the specificity.

In order to properly comment and discuss the results and obtain both a particular comparison of the methods and models used in each article and an overview of all the models and methods discovered, this section will be divided into two parts. In the first

one, we will focus on the comparison and discussion of all the methods used in each article. In the second, we will look at the whole set of articles as an overview.

6.1. Article by Article Discussion

As shown in Table 2, three different methods have been used to carry out the task of detecting and classifying defects in one of the articles [23]. The article has two main objectives. The first is to implement a CNN to perform the detection and classification of defects on the surface of the wafers, and the second is to detect defects of unknown class. Two different datasets are used. One of them, consisting of 2123 160×160 images, after being augmented to 6369 images by means of flipping and rotating the original images, is used for training and testing. The other is composed of 30 unknown defect images. To achieve the first objective of the paper, a CNN designed by the authors themselves is implemented. It contains four convolutional layers, two max pooling layers, one FC layer and ReLU as its activation function. This CNN, which achieves a test accuracy of 0.962, is compared with a classical ML method. Specifically, it is an SVM with the radial base function, which achieves a test accuracy of 0.925. Thus, we can clearly see that the designed CNN exceeds the SVM in this case and for this particular data set. Finally, to achieve the second objective, a K-NN is fed with the CNN output. K-NN only fails in 2 of the 30 images, and only in one of the five categories that are analysed, so that its accuracy is 0.933.

In the following article [27], the authors implemented a CNN designed by themselves to perform the task of defect detection and classification. This CNN was composed of three convolutional layers, one max pooling layer and two FC layers. The available dataset for this experiment, once augmented, contained 12,000 images. The test accuracy achieved was 0.953. The authors also applied the same dataset to a random forest method, obtaining a test accuracy of 0.942. Thus, it can be stated that, although the results are adequate and very similar, the designed CNN outperforms the random forest method in this case. Regarding the average training time for this experiment, it depends on the number of clients used when computing. The average training lasts for about 160 s when there is only one client, while when there are from six to ten clients, it decreases to 80 s.

A k-means approach is then adopted in the article [33]. The objective of the study is to analyse an SEM image to detect, quantify and classify the image features using clusters in which the defects are included. As can be seen in Table 2, the authors do not provide quantitative results. They present a novel tool that allows to quantify and group the features, defects in this case, in different clusters according to their features and to manage them together.

The next article that appears in Table 2 is [36]. Again, there is no result on accuracy in the table but for another reason. The authors designed an SOM-based method to perform automatic defect detection and classification on wafers. The results have not been included in the table because different combinations of mask sizes and number of clusters were attempted to determine the best one. The results were expressed in terms of specificity (number of negative or non-defective true ROIs (regions of interest) detected divided by total negative or non-defective ROIs) and sensitivity (number of positive or defective true ROIs detected divided by total positive or defective ROIs). The best combination of results was reached by using 8×8 masks with three, four or five groups, achieving a value of 0.967 for sensitivity and 1 for specificity.

Then, in Reference [55], the authors used three different CNN models in order to classify different wafer maps: Inception V2, ResNet-50 and VGGNet-16. The achieved accuracy was, 0.9, 0.875 and 0.844, respectively. It seems clear that Inception V2 and ResNet-50, which have almost identical results, outperform VGGNet-16. Then, the Radon transform was merged with the previously mentioned CNN models, creating new models named R-Inception V2, R-ResNet-50 and R-VGGNet-16. These new models had an accuracy of 0.974, 0.968 and 0.96. As we can appreciate, every model of this new approach outperformed its corresponding

simpler model. Among the new models, the same tendency was maintained. R-Inception V2 and R-ResNet-50 offered similar results and both slightly outperformed R-VGGNet-16.

Moreover, in [56], the authors classified the chemical composition of particle defects on semiconductor wafers. A VGGNet-based model was assigned for achieving that goal. The idea of the article was to merge the features extracted along the convolution operations with data extracted from an energy-dispersive X-ray (EDX) microscope. In order to do that, the authors started from a dataset of 5761 SEM images belonging to 8 classes, along with 98 images with no defect. The average training time was of 0.264 s/defect/epoch/GPU. The test accuracy reached was 0.992 for the Top-3 accuracy and 0.821 for the Top-1 accuracy. Then, two different transfer learning approaches were followed, obtaining worse results without saving computing time significantly.

Continuing with article [58], the authors developed a CNN model based on Inception V1 that surpassed the commercial state-of-the-art automatic defect classification (ADC) system in the field of semiconductor defect detection and classification. Up to 5388 images of eight different classes constitute the dataset for this experiment. In this case, the CNN selected achieved a test accuracy of 0.873, whereas the ADC system achieved a test accuracy of 0.772. Therefore, the CNN model clearly outperforms the ADC system. Regarding the run-time, it varies from 40 h when no transfer learning is included to 4 h once transfer learning is included.

In [59], the authors compared two different CNN models, Inception V3 and ResNet-50, in semiconductor defect detection and classification. As shown in Table 2, the accuracy of the models was 0.6 and 0.7, respectively. These values are poor, probably due to the initial dataset's distinctiveness and/or quality, as it contains only 736 images, which belong to seven classes. Nevertheless, the ResNet-50 model outperformed the Inception V3 one. Regarding the training time, it takes for about 4.5 h to train Inception V3 and 9 h to train ResNet-50.

Finally, in [64], the authors proposed three different CNN models to carry out the post-sawing inspection task. The three CNNs were designed by the proper authors. The first CNN used a back propagation algorithm that reached an accuracy of 1. The second one used a linear vector quantisation (LVQ) algorithm and also achieved an accuracy of 1. Finally, the third one implemented a radial basis algorithm and obtained an accuracy of 0.9. For this dataset, backpropagation and LVQ offered the same result, outperforming the radial basis one. Although these results are included, it seems clear that they are not really comparable with the ones of the previous papers, as obtaining accuracy values of 100% reveals that the classification task was not that complex.

6.2. General Overview

Once the results of the different articles have been discussed one by one, the challenge is to construct a comprehensive overview of the results. It can be said that CNN surpasses ML methods, at least in all the papers we have collected. This can be explained by the ability of CNN to extract meaningful features from large datasets. If the datasets were smaller, ML methods could probably outperform CNN, the performance of which declines when the volume of the datasets is reduced.

Among all CNN models, special attention should be paid to those designed by the authors themselves. As the authors are well aware of the requirements of each particular application, they can configure the CNN to achieve decent performances and even be lighter than existing predesigned models.

When talking about predesigned models, the Res-Net and Inception V3 models, which obtain similar and satisfactory results, should be highlighted. As for ML-based methods, SVM and random forest are probably the ones that perform best in the works studied in this scoping review.

6.3. Limitations of This Work

This work is subject to some limitations. In addition to the proper limitations that each article included has itself, our review presents its own limitations. These stem from the difficulty of comparing the several articles that comprise it. Despite using similar images and models, each work uses its own dataset and proposes its own task. Each dataset is composed of a different number of images and classes with different sizes and distributions. As for the tasks, each has its own complexity. All these aspects make it difficult to draw fully concluding remarks.

7. Conclusions

This work has presented a scoping review in relation to the detection and classification of defects in semiconductors from scanning electron microscope (SEM) images through the use of machine learning (ML) and deep learning (DL) approaches. Throughout the paper, we have addressed several issues.

First, our search strategy was determined, obtaining nine final articles from a total of 224 found initially. Next, we focused on the fundamentals of SEM, explaining the objective of this microscopy and the main components of an SEM device. Later, we described the most typical ML methods, classifying them into supervised, unsupervised and semi-supervised methods. Then, we presented the different components of a CNN and, later, the most typical models as well as the two main approaches that can be followed for defect detection: one-stage and two-stage. Finally, the results obtained in the different articles were presented and discussed.

The final conclusion should be that the main purpose of the scoping review has been fulfilled. To the best of our understanding, all the knowledge about the detection and classification of defects in semiconductors from SEM images using ML and DL has been gathered.

From the authors' point of view, as only nine papers that address the challenge of detecting and classifying defects in semiconductor materials from SEM images have been found, there is a huge opportunity with a multitude of exploitable approaches to achieve even better results than those obtained so far in the literature reviewed. All the information that has been gathered will help in future works to directly rule out some methods that offer poor performance and to focus directly on the higher performance methods. Therefore, the main contribution of this work is to guide future authors towards the best performing methods to further improve the results in defect detection and classification and thus contribute to the development and prosperity of the industry.

Author Contributions: Conceptualisation, A.F.-C.; formal analysis, F.L.d.I.R. and R.S.-R. funding acquisition: A.F.-C. investigation, F.L.d.I.R., R.S.-R., J.L.G.-S. and R.M. methodology, F.L.d.I.R. supervision, R.M. and A.F.-C. writing—original draft, F.L.d.I.R. and R.S.-R. Writing—review and editing, R.M. and A.F.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by iRel40, a European co-funded innovation project that has been granted by the ECSEL Joint Undertaking (JU) (grant number 876659). The funding of the project comes from the Horizon 2020 research programme and participating countries. National funding is provided by Germany, including the Free States of Saxony and Thuringia, Austria, Belgium, Finland, France, Italy, the Netherlands, Slovakia, Spain, Sweden, and Turkey. The publication is part of the project PCI2020-112001, funded by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU"/PRTR.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funding sources had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AE	Autoencoder
BCE	Binary cross-entropy
BSE	Back-scattered electron
CCE	Categorical cross-entropy
CCVAE	Conditional convolutional autoencoder
CNN	Convolutional neural network
DA	Discriminant analysis
DL	Deep learning
DT	Decision Tree
FC	Fully-connected
GAN	Generative adversarial network
GPU	Graphics processing unit
K-NN	K-nearest neighbours
LDA	Linear discriminant analysis
MAE	Mean absolute error
ML	Machine learning
MSE	Mean square error
QDA	Quadratic discriminant analysis
RBM	Restricted Boltzmann machine
ReLU	Rectifier linear unit
RNN	Recurrent neural network
ROI	Region of interest
SCCE	Sparse categorical cross entropy
SE	Secondary electron
SEM	Scanning electron microscope
SGD	Stochastic gradient descent
SOM	Self-organising maps
STEM	Scanning transmission electron microscopy
SVM	Support vector machine
VAR	Variational autoencoder
YOLO	You only look once

References

1. STATISTA. *Monthly Semiconductor Sales Worldwide from 2012 to 2020 (in Billion U.S. Dollars)*; Statista GmbH: Hamburg, Germany, 2019.
2. Huang, S.H.; Pan, Y.C. Automated visual inspection in the semiconductor industry: A survey. *Comput. Ind.* **2015**, *66*, 1–10. [[CrossRef](#)]
3. Park, H.; Choi, J.E.; Kim, D.; Hong, S.J. Artificial Immune System for Fault Detection and Classification of Semiconductor Equipment. *Electronics* **2021**, *10*, 944. [[CrossRef](#)]
4. Zheng, X.; Zheng, S.; Kong, Y.; Chen, J. Recent advances in surface defect inspection of industrial products using deep learning techniques. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 35–38. [[CrossRef](#)]
5. Górriz, J.M.; Ramírez, J.; Ortiz, A.; Martínez-Murcia, F.J.; Segovia, F.; Suckling, J.; Ferrández, J.M.; Leming, M.; Zhang, Y.; Álvarez-Sánchez, J.R.; et al. Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications. *Neurocomputing* **2020**, *410*, 237–270. [[CrossRef](#)]
6. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep learning vs. traditional computer vision. In *Science and Information Conference*; Springer Nature: Cham, Switzerland, 2019; pp. 128–144.
7. Abd Al Rahman, M.; Mousavi, A. A Review and Analysis of Automatic Optical Inspection and Quality Monitoring Methods in Electronics Industry. *IEEE Access* **2020**, *8*, 183192–183271. [[CrossRef](#)]
8. Xiao, M.; Wang, W.; Shen, X.; Zhu, Y.; Bartos, P.; Yiliyasi, Y. Research on defect detection method of powder metallurgy gear based on machine vision. *Mach. Vis. Appl.* **2021**, *32*, 1–13. [[CrossRef](#)]

9. Li, G.; Shi, J.; Luo, H.; Tang, M. A computational model of vision attention for inspection of surface quality in production line. *Mach. Vis. Appl.* **2013**, *24*, 835–844. [\[CrossRef\]](#)
10. Schlosser, T.; Beuth, F.; Friedrich, M.; Kowerko, D. A novel visual fault detection and classification system for semiconductor manufacturing using stacked hybrid convolutional neural networks. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1511–1514.
11. Sharp, M.; Ak, R.; Hedberg, T., Jr. A survey of the advancing use and development of machine learning in smart manufacturing. *J. Manuf. Syst.* **2018**, *48*, 170–179. [\[CrossRef\]](#)
12. Tomlinson, W.; Halliday, B.; Farrington, D.; Skumanich, A. In-line SEM based ADC for advanced process control. In Proceedings of the 2000 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop, Boston, MA, USA, 12–14 September 2000; pp. 131–137. [\[CrossRef\]](#)
13. Avinun-Kalish, M.; Sagi, O.; Im, S.M.; Lee, C.; Oh, J.; Lim, J.; Yoo, H.; Kim, C. Novel SEM based imaging using secondary electron spectrometer for enhanced voltage contrast and bottom layer defect review. In Proceedings of the 2009 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, Berlin, Germany, 10–12 May 2009; pp. 223–227. [\[CrossRef\]](#)
14. Becker, B.; Porat, R.; Eschwege, H. Identification of yield loss sources in the outer dies using SEM based wafer bevel review. In Proceedings of the 2010 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, San Francisco, CA, USA, 11–13 July 2010; pp. 119–122. [\[CrossRef\]](#)
15. Newell, T.; Tillotson, B.; Pearl, H.; Miller, A. Detection of electrical defects with SEMVision in semiconductor production mode manufacturing. In Proceedings of the 2016 27th Annual SEMI Advanced Semiconductor Manufacturing Conference, Saratoga Springs, NY, USA, 16–19 May 2016; pp. 151–156. [\[CrossRef\]](#)
16. Jain, A.; Sheridan, J.G.; Levitov, F.; Aristov, V.; Yasharzade, S.; Nguyen, H. Inline SEM imaging of buried defects using novel electron detection system. In Proceedings of the 2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA, 30 April–3 May 2018; pp. 259–263. [\[CrossRef\]](#)
17. Zhou, W.; Apkarian, R.; Wang, Z.L.; Joy, D. Fundamentals of scanning electron microscopy (SEM). In *Scanning Microscopy for Nanotechnology*; Springer Nature: Cham, Switzerland, 2006; pp. 1–40. [\[CrossRef\]](#)
18. Jain, A.; Sheridan, J.G.; Xing, R.; Levitov, F.; Yasharzade, S.; Nguyen, H. SEM imaging and automated defect analysis at advanced technology nodes. In Proceedings of the 2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), Saratoga Springs, NY, USA, 15–18 May 2017; pp. 240–248. doi: 10.23919/mipro.2017.7966585. [\[CrossRef\]](#)
19. Bustillo, A.; Pimenov, D.Y.; Mia, M.; Kapłonek, W. Machine-learning for automatic prediction of flatness deviation considering the wear of the face mill teeth. *J. Intell. Manuf.* **2021**, *32*, 895–912. [\[CrossRef\]](#)
20. Dey, A. Machine learning algorithms: A review. *Int. J. Comput. Sci. Inf. Technol.* **2016**, *7*, 1174–1179.
21. Sánchez-Reolid, R.; López, M.T.; Fernández-Caballero, A. Machine Learning for Stress Detection from Electrodermal Activity: A Scoping Review. *Preprints* **2020**. doi: 10.20944/preprints202011.0043.v1. [\[CrossRef\]](#)
22. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process Syst.* **1996**, *9*, 155–161.
23. Cheon, S.; Lee, H.; Kim, C.O.; Lee, S.H. Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 163–170. [\[CrossRef\]](#)
24. Freund, Y.; Mason, L. The alternating decision tree learning algorithm. In Proceedings of the Sixteenth International Conference on Machine Learning, Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 124–133.
25. García-Moreno, A.I.; Alvarado-Orozco, J.M.; Ibarra-Medina, J.; Martínez-Franco, E. Ex-situ porosity classification in metallic components by laser metal deposition: A machine learning-based approach. *J. Manuf. Process.* **2021**, *62*, 523–534. [\[CrossRef\]](#)
26. Blevins, J.; Yang, G. Machine learning enabled advanced manufacturing in nuclear engineering applications. *Nucl. Eng. Des.* **2020**, *367*, 110817. [\[CrossRef\]](#)
27. Lei, H.; Teh, C.; Li, H.; Lee, P.H.; Fang, W. Automated Wafer Defect Classification using a Convolutional Neural Network Augmented with Distributed Computing. In Proceedings of the 2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference, Saratoga Springs, NY, USA, 24–26 August 2020; pp. 1–5. doi: 10.1109/ASMC49169.2020.9185253. [\[CrossRef\]](#)
28. Dudani, S.A. The distance-weighted k-nearest-neighbor rule. *IEEE Trans. Syst. Man. Cybern.* **1976**, *6*, 325–7. [\[CrossRef\]](#)
29. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, DC, USA, 4–10 August 2001; Volume 3, pp. 41–46.
30. Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K.R. Fisher discriminant analysis with kernels. In Proceedings of the Neural Networks for Signal Processing IX, Madison, WI, USA, 25–25 August 1999; pp. 41–48. [\[CrossRef\]](#)
31. Singh, A.; Thakur, N.; Sharma, A. A review of supervised machine learning algorithms. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development, New Delhi, India, 16–18 March 2016; pp. 1310–1315.
32. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc. C* **1979**, *28*, 100–108. [\[CrossRef\]](#)
33. Halder, S.; Cerbu, D.; Saib, M.; Leray, P. SEM image analysis with K-means algorithm. In Proceedings of the 2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference, Saratoga Springs, NY, USA, 30 April–3 May 2018; pp. 255–258. [\[CrossRef\]](#)
34. Kaufman, L.; Rousseeuw, P.J. Partitioning Around Medoids (Program PAM). In *Finding Groups in Data: An Introduction to Cluster Analysis*; Wiley: New York, NY, USA, 1990; pp. 68–125. [\[CrossRef\]](#)
35. Kohonen, T. The self-organizing map. *Proc. IEEE* **1990**, *78*, 1464–1480. [\[CrossRef\]](#)

36. Chang, C.Y.; Li, C.; Chang, J.W.; Jeng, M. An unsupervised neural network approach for automatic semiconductor wafer defect inspection. *Expert Syst. Appl.* **2009**, *36*, 950–958. [\[CrossRef\]](#)
37. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [\[CrossRef\]](#)
38. Mery, D. Aluminum Casting Inspection using Deep Object Detection Methods and Simulated Ellipsoidal Defects. *Mach. Vis. Appl.* **2021**, *32*, 1–16. [\[CrossRef\]](#)
39. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing: Methods and applications. *J. Manuf. Syst.* **2018**, *48*, 144–156. [\[CrossRef\]](#)
40. Lei, C.W.; Zhang, L.; Tai, T.M.; Tsai, C.C.; Hwang, W.J.; Jhang, Y.J. Automated Surface Defect Inspection Based on Autoencoders and Fully Convolutional Neural Networks. *Appl. Sci.* **2021**, *11*, 7838. [\[CrossRef\]](#)
41. Fukushima, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Netw.* **1988**, *1*, 119–130. [\[CrossRef\]](#)
42. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology, Antalya, Turkey, 21–23 August 2017; pp. 1–6. [\[CrossRef\]](#)
43. Zhang, Z.; Wen, G.; Chen, S. Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding. *J. Manuf. Process.* **2019**, *45*, 208–216. [\[CrossRef\]](#)
44. Khodja, A.Y.; Guersi, N.; Saadi, M.N.; Boutasseta, N. Rolling element bearing fault diagnosis for rotating machinery using vibration spectrum imaging and convolutional neural networks. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 1737–1751. [\[CrossRef\]](#)
45. Xia, C.; Pan, Z.; Fei, Z.; Zhang, S.; Li, H. Vision based defects detection for Keyhole TIG welding using deep learning with visual explanation. *J. Manuf. Process.* **2020**, *56*, 845–855. [\[CrossRef\]](#)
46. Wang, J.; Lee, S. Data Augmentation Methods Applying Grayscale Images for Convolutional Neural Networks in Machine Vision. *Appl. Sci.* **2021**, *11*, 6721. [\[CrossRef\]](#)
47. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Bengio, Y.; Courville, A. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
48. Yun, J.P.; Shin, W.C.; Koo, G.; Kim, M.S.; Lee, C.; Lee, S.J. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *J. Manuf. Syst.* **2020**, *55*, 317–324. [\[CrossRef\]](#)
49. Tran, M.Q.; Liu, M.K.; Tran, Q.V. Milling chatter detection using scalogram and deep convolutional neural network. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 1505–1516. [\[CrossRef\]](#)
50. Han, J.; Zhang, D.; Cheng, G.; Liu, N.; Xu, D. Advanced deep-learning techniques for salient and category-specific object detection: A survey. *IEEE Signal Process. Mag.* **2018**, *35*, 84–100. [\[CrossRef\]](#)
51. Nguyen, T.P.; Choi, S.; Park, S.J.; Park, S.H.; Yoon, J. Inspecting method for defective casting products with convolutional neural network (CNN). *Int. J. Precis. Eng.-Manuf.-Green Technol.* **2021**, *8*, 583–594. [\[CrossRef\]](#)
52. Dhillon, A.; Verma, G.K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* **2020**, *9*, 85–112. doi: 10.1007/s13748-019-00203-0. [\[CrossRef\]](#)
53. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
54. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
55. Yuan-Fu, Y.; Min, S. Double Feature Extraction Method for Wafer Map Classification Based on Convolution Neural Network. In Proceedings of the 2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference, Saratoga Springs, NY, USA, 24–26 August 2020; pp. 1–6. [\[CrossRef\]](#)
56. O’Leary, J.; Sawlani, K.; Mesbah, A. Deep Learning for Classification of the Chemical Composition of Particle Defects on Semiconductor Wafers. *IEEE Trans. Semicond. Manuf.* **2020**, *33*, 72–85. [\[CrossRef\]](#)
57. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9. [\[CrossRef\]](#)
58. Imoto, K.; Nakai, T.; Ike, T.; Haruki, K.; Sato, Y. A CNN-based transfer learning method for defect classification in semiconductor manufacturing. In Proceedings of the 2018 International Symposium on Semiconductor Manufacturing, Tokyo, Japan, 10–11 December 2018; pp. 1–3. [\[CrossRef\]](#)
59. Monno, S.; Kamada, Y.; Miwa, H.; Ashida, K.; Kaneko, T. Detection of Defects on SiC Substrate by SEM and Classification Using Deep Learning. In *International Conference on Intelligent Networking and Collaborative Systems*; Springer Nature Switzerland AG: Cham, Switzerland, 2018; pp. 47–58. [\[CrossRef\]](#)
60. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [\[CrossRef\]](#)
61. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
62. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.

-
63. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
 64. Su, C.T.; Yang, T.; Ke, C.M. A neural-network approach for semiconductor wafer post-sawing inspection. *IEEE Trans. Semicond. Manuf.* **2002**, *15*, 260–266. [[CrossRef](#)]
 65. Choi, D.; Shallue, C.J.; Nado, Z.; Lee, J.; Maddison, C.J.; Dahl, G.E. On empirical comparisons of optimizers for deep learning. *arXiv* **2019**, arXiv:1910.05446.
 66. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [[CrossRef](#)]
 67. Polyak, B.T. Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.* **1964**, *4*, 1–17. [[CrossRef](#)]
 68. Nesterov, Y.E. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR* **1983**, *269*, 543–547.
 69. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
 70. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
 71. Miao, R.; Jiang, Z.; Zhou, Q.; Wu, Y.; Gao, Y.; Zhang, J.; Jiang, Z. Online inspection of narrow overlap weld quality using two-stage convolution neural network image recognition. *Mach. Vis. Appl.* **2021**, *32*, 27. [[CrossRef](#)]