Wafer Defect Map Classification Using Sparse Convolutional Networks

Roberto di Bella* Diego Carrera† Beatrice Rossi† Pasqualina Fragneto†
Giacomo Boracchi*

* Politecnico di Milano, Milan, Italy
† STMicroelectronics, Agrate Brianza, Italy

Abstract. Chips in semiconductor manufacturing are produced in circular wafers that are constantly monitored by inspection machines. These machines produce a wafer defect map, namely a list of defect locations which corresponds to a very large, sparse and binary image. While in these production processes it is normal to see defects that are randomly spread through the wafer, specific defect patterns might indicate problems in the production that have to be promptly identified.
We cast wafer monitoring in a challenging image classification problem where traditional convolutional neural networks, that represent state-of-the-art solutions, cannot be straightforwardly employed due to the very large image size (say 20,000 x 20,000 pixels) and the extreme class imbalance. We successfully address these challenges by means of Submanifold Sparse Convolutional Networks, deep architectures that are specifically designed to handle sparse data, and through an ad-hoc data augmentation procedure designed for wafer defect maps. Our experiments show that the proposed solution is very successful over a dataset of almost 30,000 maps acquired and annotated by our industrial partner. In particular, the proposed solution achieves significantly high recall on normal wafer defect maps, that represent the large majority of the production. Moreover, our data augmentation procedure turns out to be beneficial also in smaller images, as it allows to outperform the state-of-the-art classifier on a public datasets of wafer defect maps.

Keywords: Sparse Convolutional Networks, Wafer Defect Map, Industrial Monitoring, Pattern Classification, Quality Control

1 Introduction

Semiconductor manufacturing is a long and expensive process, which involves many specialized steps to yield wafers containing hundreds of chips, see Figure 1(a). Multiple sophisticated *inspection tools* are employed along the production line, which locate defective regions inside each chip and assemble a Wafer Defect Map (WDM), namely a list of coordinates where all defects within a wafer lie. In normal production conditions, defects appear randomly distributed over WDMs, without any specific spatial arrangement. In contrast, WDMs portraying *patterns* like those in Figure 1 might indicate problems occurred during the

production. In fact, some of these patterns can be traced to a particular problem in a manufacturing step and their prompt detection allows timely alerts to prevent huge production waste, thus substantially improving production efficiency. Therefore, automatic tools that identify patterns in WDMs are of paramount importance for semiconductor industries, which have to guarantee high-quality standards and high-throughput production to satisfy the growing demand of chips, lately further pushed by automotive/mobile/wearable/IoT sectors.

Algorithms for identifying defect patterns on wafers have been quite extensively investigated in the literature [1, 7], and the most effective solutions rely on classifiers [3, 4, 15, 14]. However, none of these classifiers handles WDMs directly, since each WDM corresponds to a very large binary image, whose size is limited only by the resolution of the inspection tool (in our case each WDM has resolution $20,000 \times 20,000$). This representation is impossible to handle by standard classifiers as they would require many computational resources (both in term of operations and memory) to process images of this size. Therefore, most of the existing solutions [3, 4, 10, 15] reduce the image size by a preliminary preprocessing step, which corresponds to a lossy conversion of the information contained in the original WDM that might prevent capturing the full diversity of defect patterns.

Our intuition is that, to successfully employ a classifier, and in particular a convolutional neural network (CNN) [8], two major challenges have to be addressed. First, the network should be designed to efficiently process very large inputs, to provide the classifier with the entirety of the WDM information content. Second, the training procedure should cope with severe class imbalance, since in real-world monitoring conditions some defect classes occur very rarely.

Here we adopt Submanifold Sparse Convolutional Networks (SSCN) [5] to handle our WDMs that are very large and at the same time sparse. Convolutional layers in a SSCN implement a convolution operator that modifies only the nonzero values in the feature maps. As a result, very deep SSCNs preserve the sparsity of the input data (which is reduced only by max pooling operations) and this property allows the network to better capture those peculiar patterns in the input. Moreover, the submanifold convolution operator is computationally more efficient than its traditional counterpart, both in terms of number of operations performed and memory required, since only the nonzero coefficients in each feature maps are stored and processed.

Our contribution is twofold: we are the first to adopt SSCN to classify images that are as large as WDMs by means of a very deep architecture. Moreover, to cope with class imbalance, we design an ad-hoc data-augmentation procedure to map each WDM in a set of realistic WDMs. In particular, we extend the set of standard geometric transformations performed in data augmentation routines, and introduce cropping and mixing with synthetically generated WDMs. To this purpose, we learn from our training set a statistical model of the distribution of random defects in normal WDMs, and perform data augmentation in a class-specific procedure that provides very rich information to the network and at the same time mitigates class imbalance.

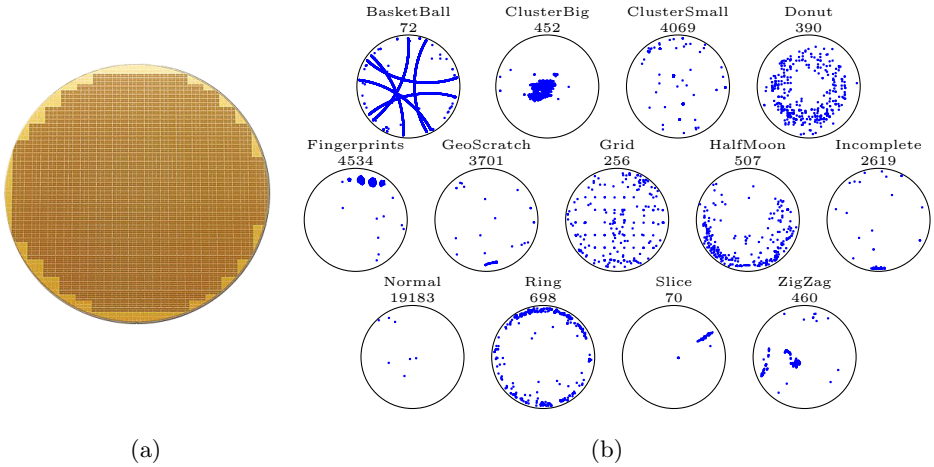

Fig. 1. (a) An example of wafer containing few hundreds of chips (the small cells). (b) An example of WDM for each of the class in ST Dataset. We also report the number of instances of each class in our dataset to emphasize the severe class imbalance.

Our experiments, performed on a large dataset of WDMs collected and annotated by our industrial partner, show that the proposed approach is a better option than a state-of-the-art pre-trained network (VGG-16 [13]) fine-tuned on low-resolution images obtained by preprocessing the WDMs. In particular, our SSCN achieves a significantly higher recall on normal WDMs, which represent the vast majority of the industrial production. Moreover, we show that, on the large WM811K dataset [14], CNNs trained by employing our class-specific image-augmentation procedure can outperform the solution in [14] based on hand-crafted features.

Related Works. All the solutions in the literature preprocess WDMs to reduce their size. Most often, WDMs are pre-processed to create a wafer bin map, namely a binary image where each pixel corresponds to a chip and indicates whether that chip contains defects or not. Since the number of chips in the wafer is relatively small (typically a few hundreds), wafer bin maps are much easier to handle. Wafer bin maps have been analyzed using either unsupervised or supervised methods. Unsupervised methods ([7, 1]) create clusters of similar wafers by clustering algorithms such as Adaptive Resonance Theory, k-means and Particle Swarm Optimization. The solution in [1] combines the clustering with a statistical test based on Log Odd Ratio, that preliminary screens the wafer bin maps to determine which ones do not present any specific pattern (such as the WDMs in *Normal* class, see Figure 1). Although unsupervised methods have the great advantage of not requiring annotated datasets, they are meant to group together similar wafers rather than associating each wafer to a class of a predefined set, which is instead the problem we address here.

Most supervised methods employ hand-crafted features to monitor the wafer production. Geometric features are often very intuitive and include regional features (e.g. area, perimeter, eccentricity of a defects cluster) and density-based features (e.g. location of the most defect-dense area). Transformed domain features analyze the wafer image in a different domain through transforms like Radon or Hough, that make specific patterns clearly noticeable. A mixture of these features [3, 4, 15] are assembled in a vector and fed to a classifier, usually a Support Vector Machine (SVM) or a decision tree. On the one hand, hand-crafted features are not always able to grasp meaningful patterns in whatever conditions might appear, e.g. when these are rotated, shifted or affect only parts of the wafer surface, to name common issues in WDM monitoring. On the other hand, the impressive achievements of CNN in many visual recognition tasks suggest that approaches based on learned features have a great potential in wafer classification.

To the best of our knowledge, only [9, 10] use Deep Learning models to classify specific patterns in WDMs. However, as opposed to the approach we propose here, [9] operates on wafer bin maps and address a simpler classification problem that consists in distinguishing radial map patterns from non radial ones. The solution presented in [10] adopts a different preprocessing yielding low-resolution grayscale images (instead of wafer bin maps that are binary) where each pixel corresponds to a chip and its intensity value indicates the number of defects found in that chip. This solution has been trained exclusively on a synthetically generated dataset and tested on a small batch of real data, which does not cover all the classes considered during training. In [14] it is shown that performing transfer learning of a pretrained model (the Alexnet [8]) does not outperform the proposed solution based on hand-crafted features.

2 Problem Formulation

A WDM is a list of 2D coordinates indicating the locations of the defects inside the wafer. Obviously, a WDM can be represented as a binary image $w \in \{0,1\}^{K \times K}$ where each pixel (i,j) corresponds to a location on the wafer checked by the inspection machine and $w(i,j) = 1$ when a defect is found at (i,j). Each WDM w corresponds to a label $\ell \in \mathcal{L}$, depending on the spatial arrangements of defects in w. Our goal is to define a classifier $\mathcal{K}$ that associates to each WDM w a label $\widehat{\ell} = \mathcal{K}(w)$. To this purpose, we assume a training set of n labeled WDMs $\mathcal{W} = \{(w_1, \ell_1), \ldots, (w_n, \ell_n)\}$ is provided.

While these are rather customary settings, WDMs classification requires to address two major challenges. At first, the resolution of a WDM w is huge – in our case $K = 20,000$ – and a grayscale image of such resolution would require almost 3 GB to be loaded in memory in single precision. The second challenge is the severe class imbalance: while it is very easy to collect WDMs from *Normal* class, some patterns, such as *BasketBall*, occur very rarely during the production, thus are also very under-represented in the training set. Figure 1(b) illustrates the 13 classes of WDM patterns, as identified in our dataset by domain experts, and show that a few classes are heavily under represented in out training set.

3 Proposed Solution

In this section we present our solution to WDM classification. First, we introduce the network architecture we design to handle very large WDMs, then we describe the specific data-augmentation procedures we use both during training and test phases.

Network Architecture. As described in Section 2, our problem can be easily cast in the image classification framework, but traditional convolutional neural networks cannot be straightforwardly used for WDMs classification since they handle images at relatively low resolution (e.g., the VGG16 in [13] takes as input 224×224 RGB images). In fact, input of such dimension would require huge training and testing time and memory, to store all the feature maps of the CNN. To overcome this issue, we built a very deep network stacking Submanifold Sparse Convolutional (SSC) layers [5]. A SSC layer implements a modified convolution operator that is designed to process sparse data. The main advantage of the SSC w.r.t. its traditional counterpart is that it efficiently handles sparse data as a list of the coordinates of nonzero locations. Moreover, this layer preserves the sparsity of the input, since it does not increase the number of nonzero values in the feature maps. This property better preserves defect patterns through the layers of the network.

Our SSCN recalls for the VGG16 architecture, and the basic building block is composed by a SSC layer with ReLu activations followed by a max pooling layer with stride 2, thus the resolution of the feature maps is reduced by a factor 4 after each block. We stack 13 of these building blocks, followed by a convolutional layer and finally a fully connected one. The output of the last layer is a vector of $\#\mathcal{L}$ scores, whose maximum value determines the class of the processed WDM. To the best of our knowledge, this is the first architecture trained to process very large binary images as the WDMs we consider.

We remark that our very deep architecture replaces preliminary binning that is typically employed to reduce the WDMs dimension [10]. As we will show in Section 4, our entirely data-driven solution outperforms CNN trained over low-resolution images of the wafer as this preprocessing is a-priori defined and not optimized over training data.

Data Augmentation. As shown in Figure 1, our dataset is highly imbalanced and contains a relatively small number of WDMs compared to the datasets typically used in image classification. To increase the dataset size and avoid overfitting during training, we design a data augmentation procedure that implements a set $\mathcal{T}^\ell$ of label-preserving transformations on our WDMs:

$$\mathcal{T}^\ell = \left\{T^\ell_{\boldsymbol{\theta}} \colon \{0,1\}^{K\times K} \to \{0,1\}^{K\times K},\ \boldsymbol{\theta} \in \Theta_\ell\right\}, \tag{1}$$

where $\boldsymbol{\theta}$ denotes the parameters defining each transformations, and Θ_ℓ is the set of transformations parameter which also depends on the label ℓ. In practice, each $T^\ell_{\boldsymbol{\theta}}$ is a composition of transformations commonly used for data augmentation, such as rotations around the center of the wafer, horizontal flip, and small translations of the defective coordinates. Moreover, we perform two transformations that were specifically designed for WDMs, namely noise injection and random

mixing. Noise injection adds a small number of defects to each WDM to increase network robustness and reduce the risk of overfitting. In particular, WDMs in the *Normal* class can be seen as pure noise, since the defects in the wafer are not due to any specific problem during the production. Therefore, we estimate the distribution of the number of defects in *Normal* WDMs from the training set and use this distribution to draw the number N of defects that has to be added to each WDM. Our study and production engineers confirm that there is no particular arrangement of defects in normal WDMs, thus we uniformly sample defect coordinates within the WDM. This part of data augmentation is conveniently performed in polar coordinates since WDMs are circular. Adding noise does not change the class a WDM belongs to, because a few defects randomly spread in the WDM are present in every wafer. Random mixing consists in cropping portions of WDMs from samples of those classes that are very peculiar and are less represented in the training set, such as the *BasketBall* and *Donut*, and superimposing them to obatin novel WDMs that are used as additional training examples. In these cases, production engineers were not able to distinguish these mixed WDMs from the real ones. This data augmentation procedure is constantly invoked during training, generating new batches by transforming the original WDMs using $T^{\ell}_{\boldsymbol{\theta}}$ where the parameters $\boldsymbol{\theta}$ are randomly sampled by Θ_{ℓ}.

In principle the network trained on augmented WDMs should extract a high-level representations [2] of a WDM that are invariant to the transformations in $\mathcal{T}^{\ell}$. However, invariance is hardly achieved in practice, thus we enforce data augmentation also when classifying WDMs by our SSCN. To this purpose, we define the set of transformations

$$\mathcal{T} = \{T_{\boldsymbol{\theta}} \,:\, \boldsymbol{\theta} \in \Theta\}, \tag{2}$$

where $\Theta = \cap_{\ell \in \mathcal{L}} \Theta_{\ell}$ is the set of transformations common to all classes, thus preserving all the labels in $\mathcal{L}$. For each WDM w to be tested, we compute a set $\mathcal{A}(w)$ of N augmented WDMs:

$$\mathcal{A}(w) = \{T_{\boldsymbol{\theta}_i}(w),\ i = 1, \ldots, N\}, \tag{3}$$

where each $\boldsymbol{\theta}_i$ is randomly sampled from Θ. Then, the whole set $A(w)$ is fed to the network and the classifier output $\widehat{\ell} = \mathcal{K}(w)$ is the class achieving the maximum average score over $\mathcal{A}(w)$.

4 Experiments

Our experiments aim at showing that *i*) our solution based on SSCN can successfully handle WDMs and outperforms traditional CNNs trained to classify images having lower resolution than WDMs, and *ii*) a properly designed and trained deep architecture can outperform traditional classifiers based on hand crafted features also in WDMs classification.

Datasets. We test our architectures on two different datasets. The first one is the ST dataset, which comprises 29,746 WDMs acquired in the production site

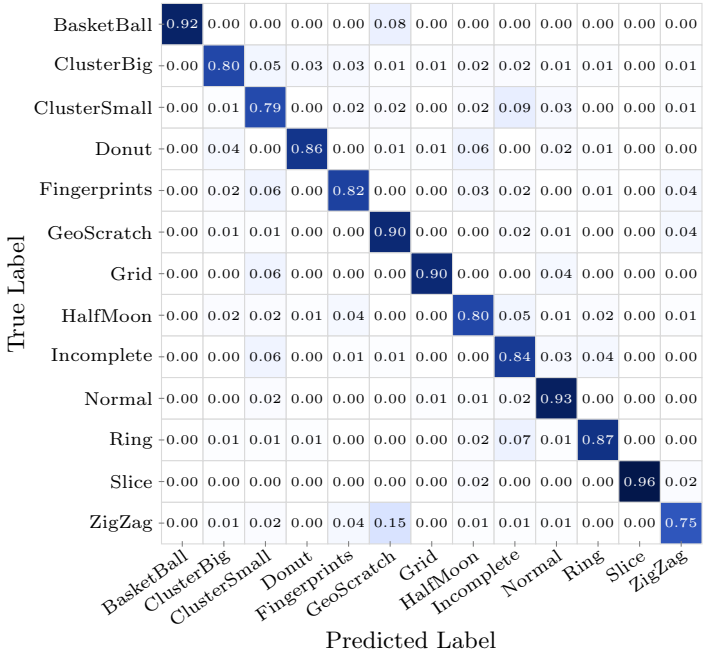

Fig. 2. Confusion matrix obtained by our SCNN on the ST dataset. Our network achieves very high accuracy on all the classes, and most misclassified samples belong to classes that are very similar, such as Incomplete and Cluster Small.

of STMicroelectronics in Agrate Brianza, Italy. Beside the *Normal* class (which does not present any defect pattern), engineers have identified 12 classes illustrated in Figure 1. The second one is the WM-811K, which is a public dataset [14] composed of 172,951 images provided by Taiwan Semiconductor Manufacturing Company. The dataset includes nine classes and is highly unbalanced: as in the ST dataset, the *Normal* class (called *None* in [14]) covers almost the 70% of the dataset. Differently from ST dataset, the WM-811K dataset contains wafer bin maps, which are very small resolution images ranging from 15×15 pixels to 200×200 pixels. We resize all this images to 64×64 using nearest neighbor interpolation, as this is by far the most common resolution in the dataset.

Figures of Merit. The traditional figure of merit used in multiclass classification problems is the *confusion matrix*, which assesses the classifier performance on each class separately, but it does not provide an overall measure of classification performance. The Area Under the ROC Curve (AUC) is probably the best option in binary classification, as it is independent from the class proportions and from the threshold employed, but it does not admit a straightforward extension to multiclass problems, thus we consider two versions of multiclass AUC. The first one is the *1vsRest-AUC* [11] and corresponds to the average of all the binary AUCs computed in a one-versus-rest fashion. In particular, for each class, a binary classification problem is tested, where the selected class is the positive one and the remaining ones are merged in the negative class. The 1vsRest-AUC

Table 1. Results on the STdataset.

Class	SSCN w/0 Aug.	SSCN	VGG16
Normal	94%	93%	89%
BasketBall	69%	92%	100%
ClusterBig	66%	80%	79%
ClusterSmall	71%	80%	82%
Donut	70 %	89%	91%
Fingerprints	55 %	85%	86%
GeoScratch	49 %	87%	89%
Grid	69%	91%	88%
HalfMoon	41%	77%	80%
Incomplete	75%	86%	92%
Ring	77 %	87%	86%
Slice	49%	96%	92%
ZigZag	44%	77%	75%
1vsRest-AUC	0.9824	**0.9902**	0.9887
1vs1-AUC	0.9430	**0.9860**	0.9858

is then obtained as a weighted average of the AUC values computed for each of these binary classification problems. In contrast, the *1vs1-AUC* in [6] employs a one-versus-one scheme and average the AUC from all the possibile pairs. The main difference between the two measures is that the *1vs1*-AUC is independent from the class proportions, while *1vsRest*-AUC is not. Since our model are relatively fast to train on a GPU, in all our experiments we assess classification performance by means of 10-fold cross validation, and average our results to reduce the variance in the figures of merit.

Considered Methods. To show the effectiveness of our SSCN on the ST dataset we consider the following alternatives in the ST dataset we consider as alternative solution a CNN obtained by training over the *VGG16* [13], a state-of-the-art convolutional neural network that won the localization task in ILSVRC 2014 competition [12]. Since the VGG16 takes as input 224×224 images, we perform a preliminary binning operation to obtain a low-resolution representation of the original WDM. In particular, each WDM is transformed in a 224×224 grayscale image where each pixel indicates the number of defects in the corresponding bin. Then, we perform a fine tuning on the ST dataset, which is a customary procedure in transfer learning. The VGG16 is trained and tested in the same conditions as our SSCN, i.e. we perform the same data augmentation on WDMs before binning. To assess the importance of data augmentation, we test the proposed SSCN both with data-augmentation and without (*SSCN w/o Aug*).

On the WM-811K dataset we adopt a traditional CNN rather than a sparse CNN, since the input images are rather small and not very sparse. Therefore, we consider a comparable architecture, though less deep, obtained by stacking traditional convolutional and max pooling layers. As alternative method, we consider the solution in [14] (denoted here as *SVM*), that extracts hand-crafted features and classify the feature vectors using a Support Vector Machine.

Advantages of directly handling WDMs instead of images We train our SSCN network over the ST dataset using the Adadelta optimizer [16] with parameters $\rho = 0.9$ and $\epsilon = 10^{-6}$. Training requires about 8 hours (for 100 epochs) and was performed using two GPUs (a Titan Xp and a Titan V), while the averaged time required to classify a WDM is 0.061 ± 0.055 seconds (we compute $N = 250$ augmented WDMs in (3)). The high variance is due to the fact that the number of operations performed by SSCNs highly depends on WDM sparsity, which varies a lot in our dataset.

Figure 2 shows the confusion matrix of the classification performance over the ST dataset. The accuracy over different classes indicate that our SSCN achieves very good classification performance and that most of classification errors are among very similar classes (e.g., *Incomplete* and *ClusterSmall*, see Figure 1(b)). Due to space limitation, Table 1 reports only a comparison in terms of class accuracy for the proposed SSCN against *VGG16* and *SSCN w/o Aug.* These values correspond to the diagonals of the confusion matrices, and show that data augmentation is key to improve classification performance. When augmentation is omitted during training and testing, the classification accuracy drops below 50% in many classes.

The performance of SSCN and VGG16 are very similar in a few classes in terms of accuracy, and also the AUC values, shown in the last rows of Table 1, are rather close. However, when the AUC is close to 1, small improvements can be very significant. In fact, the first column indicates that the SSCN w/o Aug. is significantly worst than both SSCN and VGG16, although it achieves only slightly smaller 1vsRest-AUC values. Most importantly, our SSCN achieves 93% accuracy on the *Normal* class, while VGG does not exceed 89% accuracy. High accuracy on normal data is certainly important in an industrial monitoring scenario, since the vast majority of manufactured wafers belongs to the *Normal* class. Low accuracy on the *Normal* class results in a large number of false alarms. Therefore, directly handling the huge and sparse WDM (using our SSCN) greatly reduces the false alarms w.r.t. to a traditional CNN that operated on low-resolution images.

Finally, the difference between the two AUC measures indicates the effect of class imbalance: the 1vsRest-AUC is always higher than the 1vs1-AUC, since the latter is independent from the class proportion. This effect is more evident for *SSCN w/o Aug*, which achieves lower accuracy on the other classes.

Comparison with Classification over Hand-Crafted Features Table 2 reports the diagonals of the confusion matrices for both our CNN and the SVM in [14] over the WM-811K dataset. As in the experiments on the ST dataset, the proposed solution is evaluated using 10-fold cross validation, while the performance of [14] are reported from the paper, and have been computed on a specific training and test split.[1] Our CNN significantly outperforms SVM [14], achieving an accuracy gap ranging from a minimum of 0.1% for the *Edge-Loc*,

[1] Unfortunately, the implementation of [14] has not been provided for a comparison over a 10-fold cross validation.

Table 2. Class accuracy on the WM-811K dataset. The values of the 1vs1-AUC and 1vsRest-AUC of our CNN are 0.9989 and 0.9955, respectively. We cannot compute them for the classifier in [14], since no posterior probabilities were provided.

Class	SMV	CNN
Normal	95.7%	97.9%
Center	84.9%	94.0%
Donut	74.0%	97.1%
Edge-Loc	85.1%	85.2%
Edge-Ring	79.7%	96.8%
Loc	68.5%	72.7%
Near-Full	97.9%	99.3%
Random	79.8%	94.9%
Scratch	82.4%	87.6%

up to a maximum of 23.1% for the *Donut*. Moreover, both the multiclass AUC values are above 0.99, indicating very good classification performance.

5 Conclusions

Accurate and automatic monitoring solutions are crucial for improving efficiency in semiconductor manufacturing. Here, we address the problem of classifying defect patterns on Wafer Defect Maps generated by inspection machines during the production. Our solution employs Submanifold Convolutional Neural Networks, which are perfectly suited for WDMs as they appear as huge and sparse binary images. As a result, our SCNN efficiently handles WDMs without any pre-processing procedure that alternative solutions typically require. Moreover, we propose a specific data-augmentation procedure for WDMs that turns out to be crucial to effectively train both SSCN and CNN. Our experiments, performed on a dataset of WDMs acquired in the production sites of our industrial partner, show that our SCNN achieves high accuracy on all the classes, and that outperforms all the alternatives on the *Normal*. Since *Normal* WDMs represent the vast majority of the production, this performance gap is very relevant as it yields few false alarms during monitoring. Future works address the problem of detecting unknown patterns appearing on WDMs, as this would enable to promptly react to problems that have never been observed before or that are too rare to collect enough training samples.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp and Titan V GPUs that researchers from Politecnico di Milano have used in this research.

References

1. An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing. Engineering Applications of Artificial Intelligence **26**(5), 1479 – 1486 (2013)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (2013)
3. Chang, C.W., Chao, T.M., Horng, J.T., Lu, C.F., Yeh, R.H.: Development pattern recognition model for the classification of circuit probe wafer maps on semiconductors. IEEE Transactions on Components, Packaging and Manufacturing Technology **2**(12), 2089–2097 (2012)
4. Fan, M., Wang, Q., van der Waal, B.: Wafer defect patterns recognition based on optics and multi-label classification. In: Proceedings of the IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). pp. 912–915 (2016)
5. Graham, B., Engelcke, M., van der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9224–9232 (2018)
6. Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Machine learning **45**(2), 171–186 (2001)
7. Hsu, C.Y.: Clustering ensemble for identifying defective wafer bin map in semiconductor manufacturing. Mathematical Problems in Engineering **2015** (2015)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS). pp. 1097–1105 (2012)
9. Nakata, K., Orihara, R., Mizuoka, Y., Takagi, K.: A comprehensive big-data-based monitoring system for yield enhancement in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing **30**(4), 339–344 (2017)
10. Nakazawa, T., Kulkarni, D.V.: Wafer map defect pattern classification and image retrieval using convolutional neural network. IEEE Transactions on Semiconductor Manufacturing **31**(2), 309–314 (2018)
11. Provost, F., Domingos, P.: Tree induction for probability-based ranking. Machine learning **52**(3), 199–215 (2003)
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
14. Wu, M.J., Jang, J.S.R., Chen, J.L.: Wafer map failure pattern recognition and similarity ranking for large-scale data sets. IEEE Transactions on Semiconductor Manufacturing **28**(1), 1–12 (2015)
15. Yu, J., Lu, X.: Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. IEEE Transactions on Semiconductor Manufacturing **29**(1), 33–43 (2016)
16. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)