



Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes

Cheng Hao Jin¹ · Hyun-Jin Kim² · Yongjun Piao³ · Meijing Li⁴ · Minghao Piao⁵

Received: 9 July 2019 / Accepted: 22 January 2020 / Published online: 4 February 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Defect clusters on the wafer map can provide important clue to identify the process failures so that it is important to accurately classify the defect patterns into corresponding pattern types. In this research, we present an image-based wafer map defect pattern classification method. The presented method consists of two main steps: without any specific preprocessing, high-level features are extracted from convolutional neural network and then the extracted features are fed to combination of error-correcting output codes and support vector machines for wafer map defect pattern classification. To the best of our knowledge, no prior work has applied the presented method for wafer map defect pattern classification. Experimental results tested on 20,000 wafer maps show the superiority of presented method and the overall classification accuracy is up to 98.43%.

Keywords Wafer map · Defect pattern classification · Deep learning · Convolutional neural network · Error-correcting output codes · Support vector machine · Multi-class classification

Abbreviations

CNN	Convolutional neural networks
ECOC	Error-correcting output codes
SVM	Support vector machines
CNN-SVM	SVM classification based on CNN features
OPTICS	Ordering point to identify the cluster structure

CART	Classification and regression trees
NB	Naive Bayes
<i>k</i> NN	<i>k</i> -nearest neighbors
ReLU	Rectified linear unit
LDA	Linear discriminant analysis
LOGISTIC	Logistic regression
CNN-ECOC-X	Use CNN features for ECOC classification where X is used as binary classifiers
ANOVA	Analysis of variance
SVE	Soft voting ensemble

✉ Minghao Piao
myunghopark@gmail.com
Cheng Hao Jin
kimsunghoyust@gmail.com
Hyun-Jin Kim
tohappyhj@gmail.com
Yongjun Piao
ypiao@nankai.edu.cn
Meijing Li
mjli@shmtu.edu.cn

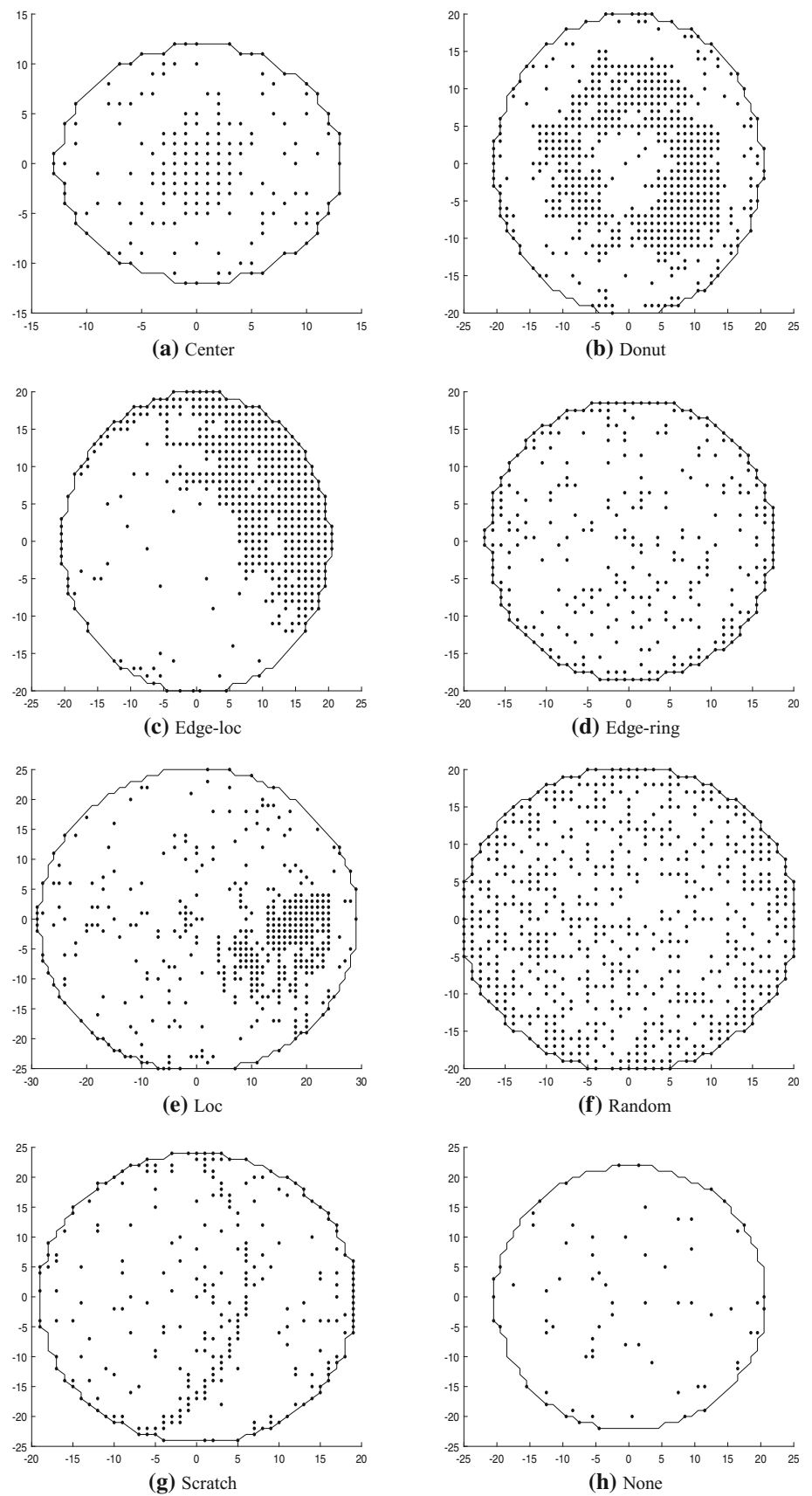
Introduction

The semiconductor wafer fabrication is a complex, long and costly process which involves hundreds of complicated chemical steps and requires monitoring a great number of process parameters (Chien et al. 2014). Due to such a complexity, it is nearly impossible to produce wafers without any defects even operated by well-trained process engineers with highly automated and precisely positioned equipments in a nearly particle-free environment (Wang et al. 2006).

Wafer map is a graphical representation of a silicon wafer at which all the good and defective die are contained. Wafer map defects are usually formed in clusters (Hansen et al.

¹ ENN Research Institute of Digital Technology, Beijing, China
² BISTel, Seoul 06754, South Korea
³ School of Medicine, Nankai University, Tianjin, China
⁴ College of Information Engineering, Shanghai Maritime University, Shanghai, China
⁵ Department of Computer Science, Chungbuk National University, Cheongju 28644, South Korea

Fig. 1 Typical wafer map pattern types



1997) and Fig. 1 illustrates typical defect pattern types. For example, in the center pattern type, most defective die are in the center of a wafer map while in the scratch pattern type, most defective die form a scratch and in the edge-ring pattern type, most defective die are in the edge-ring region and so on. These defect cluster patterns can provide clue to identify the process failures in the semiconductor manufacturing. For example, a uniformity problem during the chemical-mechanical planarization can cause center pattern, inappropriate wafer handling or poor shipment can cause the scratch pattern, a layer-to-layer misalignment during the storage-node process can cause edge-ring pattern and so on. Therefore, there is a strong need to accurately classify the defect patterns to quickly identify the root causes of failures.

In recent years, convolutional neural networks (CNN) (Krizhevsky et al. 2012) is one of the most popular deep learning methods which has shown excellent performance in a wide variety of areas including image classification (Krizhevsky et al. 2012; Rakhlin et al. 2018), defect pattern classification (Kim et al. 2019; Lin et al. 2019; Kyeong and Kim 2018; Nakazawa and Kulkarni 2018), recommender systems (Van den Oord et al. 2013), speech recognition (Xiong et al. 2018), natural language processing (Kim 2014), and face recognition (Ding and Tao 2018). CNN has several advantages that it relatively does not need any specific pre-processing, no prior knowledge and no human effort involved in feature extraction and it is also a good feature extractor.

Error-correcting output codes (ECOC) (Dietterich and Bakiri 1995) combining with multiple binary classifiers has shown high classification accuracy in multi-class classification problems. By combining the advantages of CNN and ECOC, in this research, we present an image-based wafer map defect pattern classification method. The presented method consists of two main steps: without any specific pre-processing, high-level abstraction features are extracted from CNN and then the extracted CNN features are fed to combination of ECOC and support vector machines (SVM) (Cortes and Vapnik 1995) for wafer map defect pattern classification.

The main contributions of this study can be summarized as follows:

1. CNN, ECOC, SVM as well as all combination of them are not new multi-class classification methods. However, to the best of our knowledge, this is the first time to apply the presented method for wafer map defect pattern classification.
2. For performance comparison, CNN and CNN feature-based SVM (CNN-SVM) classification methods are considered. Six different binary classifiers including SVM are also used in ECOC. Among them, the presented method shows the best performance.

The rest of the paper is organized as follows. “**Related work**” section discusses the related work of wafer defect pattern classification and “**Method**” section presents the framework of the presented method. The experimental results tested on 20,000 wafer maps are reported in “**Experimental results**” section. Finally, conclusion is given in “**Conclusion**” section.

Related work

Wafer map defect pattern classification is a multi-class classification task. Therefore, in this section, wafer map defect pattern classification and multi-class classification are discussed in more detail.

Wafer map defect pattern classification

There are a great number of methods have been proposed for wafer map defect pattern classification.

In (Fan et al. 2016), Ordering Point to Identify the Cluster Structure (OPTICS) (Ankerst et al. 1999) is first applied to remove outlier defects and then the extracted density-based and geometry-based features are used as input of SVM for classification. In (Piao et al. 2018), spatial filter (Gonzalez and Woods 2006) is applied to remove outlier defective die and the extracted random transform-based features are used as decision tree ensemble for classification. A set of novel rotation- and scale-invariant features is used as input of SVM for classification (Wu et al. 2015). In (Ooi et al. 2013), polar Fourier Transform and rotational moment invariants features are used as input of alternating decision tree classifier for classification. In (Chang et al. 2012), spatial filter is used to remove outlier defective die and then, linear hough transform is used to detect line spatial pattern, circular hough transform is used to identify bull’s-eye and blob spatial patterns, while zone ratio approach is used to pinpoint ring and edge spatial pattern. In (Yuan et al. 2010), support vector clustering (Ben-Hur et al. 2001) is used to remove outlier defective die and a Bayesian mixture model is proposed to model the defect cluster distributions where defect cluster patterns with amorphous/linear, curvilinear, and ring patterns are modeled by multivariate normal distribution, principal curve, and spherical shell, respectively. In (Wang 2008), spatial filter is used to remove outlier defective die and then a hybrid scheme combining entropy fuzzy c means with spectral clustering is used to extract the defect clusters. Finally, convexity and eigenvalue ratio are used to classify the defect pattern type. A DBSCANWBM framework (Jin et al. 2019) is proposed at which outliers are detected by applying DBSCAN with optimal parameter values and the detected outliers are removed differently according to the considered pattern types. Then

the detected patterns are classified based on discriminative features of the pattern types.

Recently, two image-based wafer map defect pattern classification methods are proposed (Kyeong and Kim 2018; Nakazawa and Kulkarni 2018). The method (Kyeong and Kim 2018) is a mixed-type pattern detection method at which each individual classification model is separately built for each defect pattern type of circle, ring, scratch, and zone. In (Nakazawa and Kulkarni 2018), CNN is directly applied to classify twenty-two classes of pattern types and the extracted image features from fully connected layer are used for wafer map image retrieval.

Multi-class classification

Multi-class classification is a task of classifying an unknown object into one of several pre-defined classes. Generally speaking, multi-class classification methods can be categorized into two groups: The first group is a direct multi-class classification method which includes methods such as classification and regression trees (CART) (Breiman et al. 1984), ID3 (Quinlan 1986), C4.5 (Quinlan 1993), naive Bayes (NB), k -nearest neighbors (k NN), multi-class SVM, neural network (Hagan et al. 1996), CNN (Krizhevsky et al. 2012) and so on. In contrast, the second group is an indirect method which decomposes the multi-class problem into a set of binary sub-problems. According to the commonly used decomposition strategies, the second group can be further divided into three broad categories: one-vs-all, one-vs-one and ECOC. In this research, we focus on ECOC model and interested readers can see (Lorena et al. 2008) for more review on the combination of binary classifiers in multi-class problems.

Summary of related work

A great number of publications have shown that ECOC can improve the classification accuracy (Dietterich and Bakiri 1995; Ali Bagheri et al. 2012; García-Pedrajas and Ortiz-Boyer 2011; Zheng et al. 2008; Liu 2006; Al-Shargie et al. 2018). However, among many binary classifiers, SVM is a much more common choice for ECOC (Zheng et al. 2008; Liu 2006; Al-Shargie et al. 2018; Othman and Rad 2019; Abd-Ellah et al. 2018; Dorj et al. 2018).

CNN itself can be used for multi-class classification. However, instead of directly being used for classification, CNN features are extracted first and then SVM is used for classification to improve classification accuracy (Tang 2013). Other combination of CNN and SVM can also be found in (Huang and LeCun 2006; Niu and Suen 2012; Xue et al. 2016).

As described above, since CNN can extract good features and ECOC with SVM used as binary classifiers can obtain high classification accuracy, we expect CNN feature-based ECOC with SVM used as binary classifiers can also

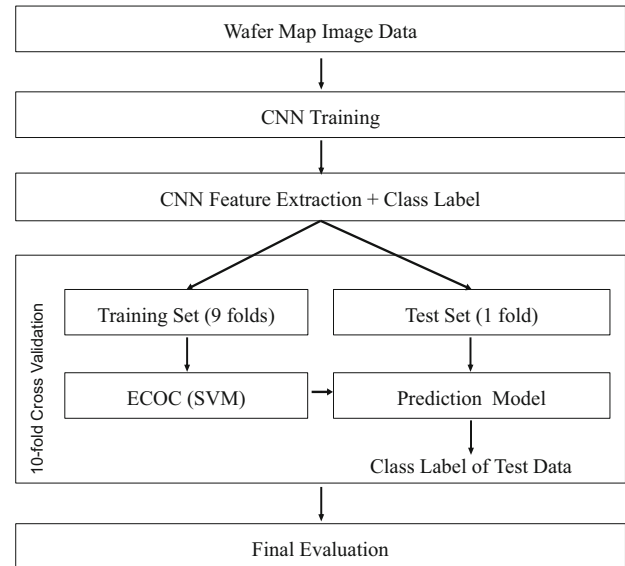


Fig. 2 Framework

obtain high classification accuracy. This is the right reason why combination of CNN, ECOC, and SVM are used in this research for wafer map defect pattern classification. Combination of CNN, ECOC, and SVM can also be found in some other domains (Othman and Rad 2019; Abd-Ellah et al. 2018; Dorj et al. 2018).

Method

The main framework of presented method is given in Fig. 2 where

1. wafer map image data is used to train CNN model,
2. CNN features are extracted from fully connected layer of the trained CNN,
3. extracted features and class labels are fed to ECOC (SVM used as binary classifiers) and perform 10-fold cross-validation,
4. final classification accuracy evaluation.

Convolutional neural network

CNN is a class of deep neural networks which has shown to be particularly effective in image classification, image and video recognition, object detection, recommender systems, medical image analysis, natural language processing and so on. CNN consists of an input layer, an output layer and many hidden layers between them. The hidden layers are a combination of convolution layers, normalization layers, pooling layers and fully connected layers.

The most important operation on CNN is in the convolution layers at which filters and input image carry out convolution operation and then the output of each convolved image is used as the input to the next layer. In this way, CNN combines the lower-level features of earlier layers to form higher level image features. These higher level image features are better suited for classification since they are in greater levels of abstraction (Donahue et al. 2014). In addition, there is no need to manually extract the useful features since the features are learned directly by the CNN. These advantages are the right reason why we use CNN in this research.

ECOC classification

ECOC (Dietterich and Bakiri 1995) is a powerful tool used for multi-class classification which can not only improve the classification accuracy but also reduce both variance and bias errors (Kong and Dietterich 1995). This is the reason why we choose ECOC as our classification model.

The main idea of ECOC is combining multiple binary classifiers for multi-class classification. In this research, SVM binary classifiers are utilized as ECOC basic classifiers. In the following, ECOC and SVM are briefly described.

ECOC

ECOC consists of coding and decoding steps.

In the coding step, the code matrix M is defined from data. M is defined as:

$$M \in \{1, -1\}^{C \times L} \tag{1}$$

where C is the number of classes, L is the length of codewords (the number of binary classifiers), each row represents a codeword of the corresponding class, each column represents the corresponding binary classifier, the value $M_{cl} = 1$ (or -1) means the samples associated with class c will be treated as positive (or negative) class for l th binary classifier. Then each of L binary classifiers is used to train according to the partition of the classes in the corresponding column of M .

In the decoding step, class labels of the test data are predicted. For a given test sample, each classifier generates a value 1 or -1 so that a L length output vector can be obtained. The obtained output vector is compared to each codeword in the code matrix and the class whose codeword has the closest distance to the output vector is chosen as the predicted class.

In ECOC, the number of classifiers L depends on what kind of coding design is used. A coding design can direct which classes are trained by each binary classifier. There are many coding designs, however, one-versus-all (Nilsson 1965) and one-versus-one (Hastie and Tibshirani 1998) are the most widely used coding designs. In one-versus-all, the

number of n classifiers are needed since n -class problem is converted into n two-class problems and for the i th two-class problem, class i is separated from the remaining classes. While in one-versus-one, the number of classifiers $n(n-1)/2$ are needed since n -class problem is converted into $n(n-1)/2$ two-class problems, which accounts for all pairs of classes. In this research, one-versus-one coding design is used.

Support vector machines

SVM (Cortes and Vapnik 1995) was originally introduced for binary classification. SVM searches for the maximum marginal hyperplane.

Training data of instance-label pairs can be expressed as:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), x_i \in R^d, y_i \in \{+1, -1\} \tag{2}$$

where y_i is the class label which can take one of two values, either +1 or -1, n is the number of training samples and d is the number of dimensions.

If the training dataset can be linearly separable, any separating hyperplane can be written as:

$$w^T x_i + b = 0, i = 1, 2, \dots, n \tag{3}$$

where w is d -dimensional weight vector ($w = (w_1, w_2, \dots, w_d)$) and b is a bias term.

Hyperplane considering the side of margin can be written as:

$$\begin{aligned} w^T x_i + b &\geq 1 \quad \forall y_i \in 1 \\ w^T x_i + b &\leq -1 \quad \forall y_i \in -1 \end{aligned} \tag{4}$$

Combining the two inequalities in Eq. 4 is equivalent to

$$y_i(w^T x_i + b) \geq 1, i = 1, 2, \dots, n \tag{5}$$

The weight vector w and bias term b need to be learned from training data to find the maximum marginal hyperplane. This can be obtained by solving the following minimization problem for w and b , subject to Eq. 5:

$$\text{minimize} \quad \frac{1}{2} \|w\|^2 \tag{6}$$

where $\|w\|$ is the Euclidean norm of w , that is $\sqrt{w \cdot w} = \sqrt{w_1^2 + w_2^2 + \dots + w_d^2}$

Such an optimization problem can be solved by the following Lagrange formulation since it is much easier to handle and also can extend to the nonlinear case:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1) \tag{7}$$

where $\alpha = (\alpha_1, \dots, \alpha_n)^T$ and α_i are the nonnegative Lagrange multipliers.

Equation 7 needs to be minimized with respect to w and b . The optimal solution is given by the saddle point and the solution satisfies the following Karush–Kuhn–Tucker conditions:

$$\frac{\partial \mathcal{L}(w, b, \alpha)}{\partial w} = 0 \quad (8)$$

$$\frac{\partial \mathcal{L}(w, b, \alpha)}{\partial b} = 0 \quad (9)$$

Equations 8 and 9 lead to:

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (10)$$

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (11)$$

Substituting Eqs. 8 and 11 into Eq. 7, we obtain the following dual problem:

$$\text{maximize } \mathcal{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (12)$$

$$\text{subject to } \sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad i = 1, 2, \dots, n \quad (13)$$

In the solution, training samples with $\alpha_i > 0$ are support vectors and all the other training samples have $\alpha_i = 0$. Once the support vectors are found, the class label of a given test sample can be predicted with the following decision function:

$$D(x) = \sum_{i \in S} \alpha_i y_i x_i^T x + b \quad (14)$$

where S is the set of support vector indices, b is a automatically determined numeric parameters by the optimization and x is test sample.

However, if the training data is not linearly separable, then the constraints in Eq. 4 is too strict so that there no hyperplane exists. In this case, a slack variable can be used to solve the problems. If the readers interested in nonlinear separable cases, please see (Cortes and Vapnik 1995) for more details.

Experimental results

Data

Table 1 shows a labeled WM-811K (Wu et al. 2015) dataset and the used dataset for experiments. The WM-811K dataset consists of 811457 real wafer maps and among them, domain

experts are recruited to label the pattern types of 172950 wafer maps. In this labeled WM-811K dataset, there are 9 types of defects (center, donut, edge-loc, edge-ring, loc, near-full, random, scratch, and none) and its pattern type distribution is shown in Table 1a. However, near-full pattern type is excluded from experiments since it can be simply identified by the defect cover ratio whose computation cost of classification is much less than that in the presented method. Therefore, the remaining eight pattern types are considered in this research. As can be seen from Table 1a, the number of wafer maps in each pattern type is highly imbalanced. The number of wafer maps in each pattern type is fixed as 2500 to make the pattern types balance in our experiments. In a labeled WM-811K dataset, since the number of wafer maps in each pattern type of center, edge-loc, edge-ring, loc, and none pattern is greater than 2500, this quantity of wafer maps is randomly selected from each of these pattern types. In contrast, the number of wafer maps in each pattern type of donut, random, and scratch pattern is less than 2500 so that synthetic wafer maps are generated for these pattern types. Each synthetic wafer map is generated from a selected real wafer map by removing two defective die where this pair of defective die is randomly selected from all pair combinations of two different defective die. In this way, four, two, and two synthetic wafer maps are respectively generated from each wafer map of 555 donut, 866 random, and 1193 scratch patterns. Therefore, total 2220 (555 * 4), 1732 (866 * 2), and 2386 (1193 * 2) synthetic wafer maps are respectively generated for donut, random and scratch patterns. Then the lack number of synthetic 1945 (2500 - 555) donut, 1634 (2500 - 866) random, and 1307 (2500 - 1193) scratch patterns are randomly selected from each of the generated 2220 donut, 1732 random, and 2386 scratch patterns to fulfill the quantity requirements. Table 1b shows the pattern type distribution of used dataset for experiments. Therefore, total 20000 wafer map images are used for experiments. The WM-811K dataset is in numeric format so that the each numeric wafer map data need to be converted into grayscale image data. Grayscale image of size [256 256] is used as CNN input.

CNN configurations

Fig. 3 shows the CNN architecture used in this research. CNN architecture is composed of input layer, 7 convolution related layers, fully connected layer and softmax layer, and output layer. In each of 7 convolution related layers, there are convolutional layer, batch normalization layer, ReLU (Rectified Linear Unit) layer and max pooling layer.

In each convolutional layer, a filter of size [3 3] and zero padding of size 1 is applied. The filter size [3 3] is applied since it is the smallest filter size which can capture the space of left/right, up/down, and center. Defects in edge-ring and edge-loc patterns are located in the wafer map edges. Zero

Table 1 Labeled WM-811K dataset and used dataset for experiments

Type	Count
<i>(a) Labeled WM-811K</i>	
Center	4294
Donut	555
Edge-loc	5189
Edge-ring	9680
Loc	3593
Near-full	149
Random	866
Scratch	1193
None	147431
Total	172950
<i>(b) Used dataset</i>	
Center	2500
Donut	2500
Edge-loc	2500
Edge-ring	2500
Loc	2500
Random	2500
Scratch	2500
None	2500
Total	20000

padding is applied since it can prevent the convolution operation from losing such edge information. Batch normalization layer is used to normalize the activations and gradients propagating through a network, making network training an easier optimization problem. Batch normalization layers between convolutional layers and ReLU layers can be used to speed up network training and reduce the sensitivity to network initialization. After each batch normalization layer, the most widely used activation function ReLU is applied. After each ReLU layer, the max pooling layer is used to reduce the feature map size and remove redundant spatial information. Such kind of reduction can make it possible to increase the number of filters in deeper convolutional layers without requiring large amount of computation per layer. Therefore, as can be seen from Fig. 3, the only difference between 7 convolution related layers is the number of filters. In each max pooling layer, pool size [2 2] and stride size [2 2] are applied not to make the pooling regions overlap. A fully connected layer is a layer at which each neuron has full connections to all the neurons in the preceding layer and it combines the features to classify the images. The number of outputs in fully connected layer is equal to the number of classes. Softmax layer normalizes the output of the fully connected layer. The stochastic gradient descent with momentum is used to applied for CNN training. Deeper layers contain higher-level features which are constructed based on the lower-level features of earlier layers. Therefore, fully connected layer right before the classifica-

tion layer is used to extract the CNN features for CNN-feature based classification.

Classification accuracy

Classifications methods of CNN and CNN-SVM are used for comparison. ECOC can be combined with any binary classifiers. For comparison, six different binary classifiers of NB, linear discriminant analysis (LDA), CART, k NN, logistic regression (LOGISTIC), and SVM are also applied in CNN feature-based ECOC (CNN-ECOC) classification. Therefore, in this subsection, experiment results of total eight classification methods are presented. The 10-fold cross-validation is applied to all eight classification methods. Due to random weight initialization in CNN and random partition of samples in 10-fold cross-validation, each of eight classification methods is run ten times to see the accuracy variances. Classification results of CNN, CNN-SVM and CNN-ECOC are respectively shown in Tables 2, 3 and 4. For convenience, the used six classification methods in Table 4 are shortened to CNN-ECOC-NB, CNN-ECOC-LDA, CNN-ECOC-CART, CNN-ECOC- k NN, CNN-ECOC-LOGISTIC, CNN-ECOC-SVM and the corresponding classification results are respectively shown in Table 4a–f. Among them, the presented method in this research is CNN-ECOC-SVM.

LIBSVM (Chang and Lin 2011) is used for CNN-SVM, while all the other programs are implemented with MATLAB 2018b (Matlab 2018). Except in CNN classification, the extracted CNN features are first standardized and the one-vs-one decomposition strategy is applied in all remaining seven classification methods. Linear SVM is applied both in CNN-SVM and CNN-ECOC-SVM. The default template function of 'templateNaiveBayes', 'templateDiscriminant', 'templateTree', 'templateKNN' and 'templateSVM' are respectively used for binary classifier of NB, LDA, CART, k NN and SVM. The template function 'templateLinear' is used for LOGISTIC binary classifier where the learner is specified with 'logistic'. In these template functions, all input parameters are used with the default values during training except the learner option 'logistic' in the 'templateLinear' function. The default values used in each of these template functions can be found in MATLAB 2018b (Matlab 2018).

In Tables 2, 3, and each table in Table 4, the first 'Iter' column shows the iteration number, the rightmost 'Avg' column shows the average classification accuracy of each iteration, and the bottom two rows respectively show the average classification accuracy and standard deviation of each pattern type over 10 iterations.

From Tables 2 and 3, it can be seen that CNN-SVM significantly outperforms CNN since

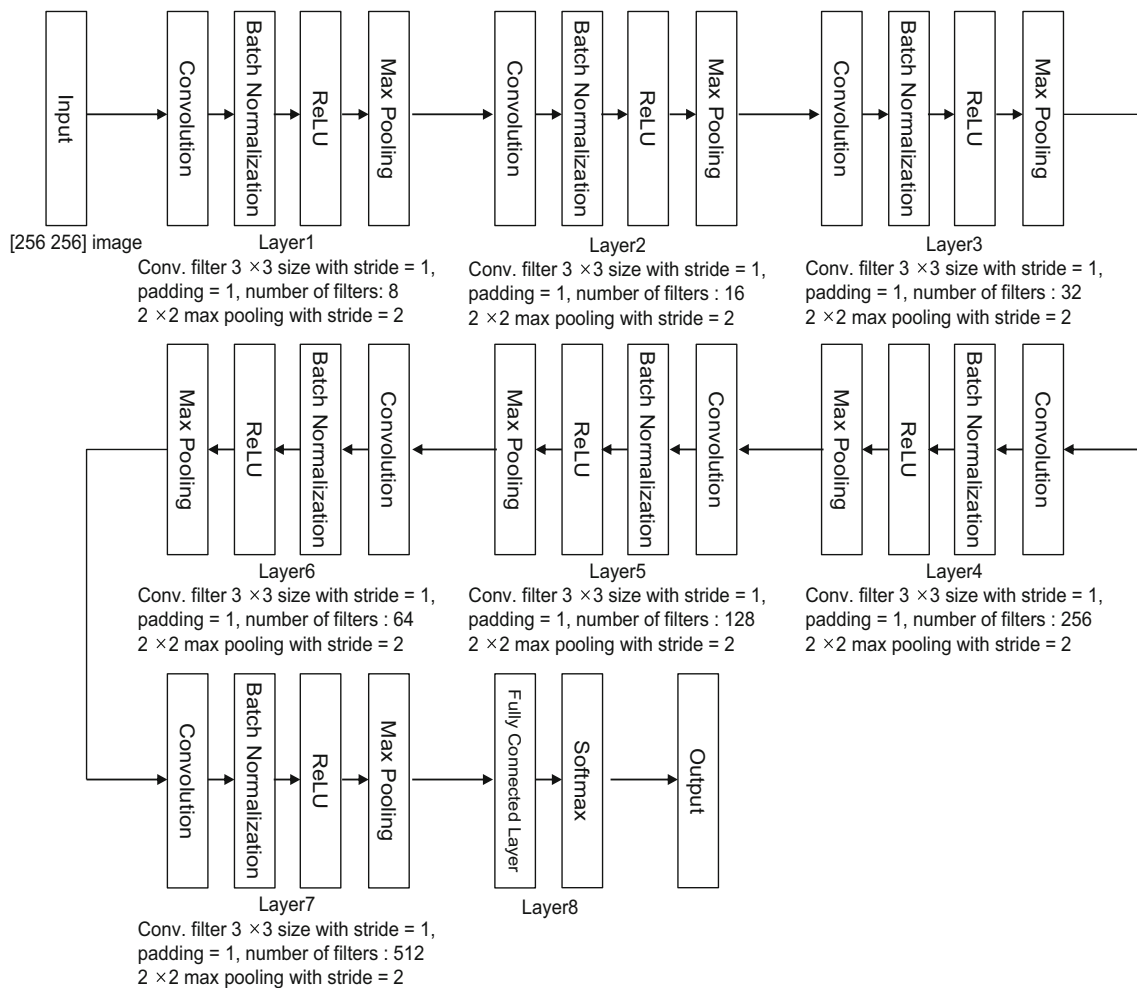


Fig. 3 CNN architecture

1. even the lowest average classification accuracy of 10 iterations 97.40% in CNN-SVM is much higher than the highest that 90.90% in CNN classification,
2. average classification accuracy of each pattern type in CNN-SVM is higher than that of each corresponding pattern type in CNN classification, and
3. the standard deviation of each pattern type in CNN-SVM is much lower than that of each corresponding pattern type in CNN classification. Although there is a slight increase in overall standard deviation, the overall average classification accuracy is improved 7.42 (97.92–90.50)%.

The only difference between these two methods is in their objectives. Softmax layer used in CNN is used to minimize cross-entropy or maximizes the log-likelihood, while SVM is used to simply find the maximum margin between data samples of different classes. Such a significant improvement is mainly due to better generalization ability of SVM than that of softmax.

In terms of classification accuracy, among all eight classification methods, CNN-ECOC-SVM with 98.43% in Table 4f obtains the highest overall classification accuracy and CNN-SVM with 97.92% in Table 3 obtains the 2nd highest overall classification accuracy:

1. In terms of average classification accuracy in each iteration: Even the lowest average classification accuracy in CNN-ECOC-SVM, 98.26% is higher than the highest that in each of all compared classification methods except CNN-SVM. Compared with CNN-SVM in Table 3, 98.26% is lower only in two iterations, 5th iteration of 98.29% and 6th iteration of 98.35%.
2. In terms of average classification accuracy in each pattern type over 10 iterations: CNN-ECOC-SVM could not win all compared methods in all eight pattern types. However, CNN-ECOC-SVM obtains the highest average classification accuracy in each of four center, edge-loc, edge-ring, and loc pattern types; in donut pattern type, only CNN-SVM (99.98%) and CNN-ECOC-kNN (100%) are higher

Table 2 CNN classification accuracy (%) (CNN)

Iter	Center	Donut	Edge-loc	Edge-ring	Loc	Random	Scratch	None	Avg
1	95.00	99.30	80.90	93.60	78.00	98.90	87.40	92.00	90.64
2	95.50	99.10	78.80	94.80	76.60	98.30	90.00	92.80	90.74
3	95.30	99.30	79.60	95.30	76.20	98.60	87.60	92.10	90.50
4	94.60	99.10	79.00	94.50	76.40	98.60	87.20	93.10	90.31
5	94.30	99.10	79.60	94.00	77.40	98.40	88.30	93.70	90.60
6	95.20	99.40	80.90	94.70	77.60	98.50	87.20	93.70	90.90
7	94.60	99.40	80.00	94.50	77.50	98.40	86.90	93.40	90.59
8	94.40	98.30	78.70	94.60	76.60	98.80	88.50	93.10	90.38
9	94.20	98.90	77.70	93.60	75.60	98.00	88.00	93.30	89.91
10	94.70	99.30	78.70	94.00	78.60	98.20	87.80	92.20	90.44
Avg	94.78	99.12	79.39	94.36	77.05	98.47	87.89	92.94	90.50
Std	0.45	0.33	1.02	0.55	0.92	0.27	0.90	0.64	0.27

Table 3 CNN feature-based SVM classification accuracy (%) (CNN-SVM)

Iter	Center	Donut	Edge-loc	Edge-ring	Loc	Random	Scratch	None	Avg
1	99.10	100.00	93.80	98.10	94.40	99.80	98.20	97.70	97.64
2	99.00	100.00	95.30	98.60	94.90	99.80	98.40	97.70	97.96
3	99.20	99.80	93.00	98.30	93.30	99.80	98.40	97.40	97.40
4	99.10	100.00	94.60	98.80	93.80	99.60	98.20	97.70	97.73
5	99.20	100.00	95.70	99.10	95.00	100.00	99.10	98.20	98.29
6	99.20	100.00	96.30	99.20	95.40	99.80	98.60	98.30	98.35
7	99.10	100.00	95.40	98.70	94.10	99.80	97.00	97.90	97.75
8	98.90	100.00	95.30	99.30	94.30	100.00	98.20	97.80	97.98
9	99.00	100.00	95.10	98.50	95.10	99.90	98.00	97.70	97.91
10	99.10	100.00	95.00	99.20	95.40	99.70	98.70	98.60	98.21
Avg	99.09	99.98	94.95	98.78	94.57	99.82	98.28	97.90	97.92
Std	0.10	0.06	0.95	0.41	0.71	0.12	0.55	0.36	0.30

than that in CNN-ECOC-SVM (99.95%); in random pattern type, only CNN-ECOC-kNN (99.96%) is higher than that in CNN-ECOC-SVM (99.87%); in scratch pattern type, only CNN-ECOC-kNN (99.09%) is higher than that in CNN-ECOC-SVM (98.69%); in none pattern type, only CNN-ECOC-LDA (99.51%) is higher than that in CNN-ECOC-SVM (98.39%). Compared with CNN-SVM, CNN-ECOC-SVM wins in all pattern types except in donut pattern type. In donut pattern type, CNN-SVM obtains average classification of 99.98% while CNN-ECOC-SVM obtains 99.95%.

- In terms of standard deviation in each pattern type over 10 iterations: Compared with CNN-SVM, in CNN-ECOC-SVM only the standard deviation in center and donut pattern types are respectively higher than that in each corresponding pattern type. However, in center pattern type, CNN-ECOC-SVM with 99.28% obtains higher average classification accuracy than that 99.09% in CNN-SVM. Although as described above, in donut pattern type, CNN-ECOC-SVM obtains lower average classification

99.95% than that 99.98% in CNN-SVM, the overall standard deviation in CNN-ECOC-SVM, 0.19 is lower than that 0.30 in CNN-SVM.

From Tables 2, 3, and each table in Table 4, a summary of accuracy comparison is presented in Table 5 where each row is taken from the row of average classification accuracy and overall standard deviation in each corresponding method. In Table 5, the best value in each column is marked in bold.

Among all eight classification methods, the statistical analysis of ANOVA is applied to determine whether 10 iterations of each method has a common average classification accuracy. The mean difference is significant at the 0.05 level. The corresponding result is given in Table 6 where the columns of 'Source', 'SS', 'df', 'MS', 'F', 'Prob>F' respectively indicate the source of the variability, the sum of squares due to each source, the degrees of freedom associated with each source, the mean squares for each source, F-statistic and p value. In Table 6, the small p value $2.8599e-54$ indicates

Table 4 CNN feature-based ECOC classification accuracy (%)

Iter	Center	Donut	Edge-loc	Edge-ring	Loc	Random	Scratch	None	Avg
<i>(a) NB as binary classifiers (CNN-ECOC-NB)</i>									
1	93.50	98.90	88.40	93.60	68.30	97.40	94.00	95.30	91.18
2	92.30	97.80	88.90	90.10	63.00	97.70	92.70	95.60	89.76
3	94.50	99.20	90.50	93.30	72.70	98.80	94.90	96.40	92.54
4	95.50	99.40	89.60	92.50	78.90	98.70	95.10	97.00	93.34
5	92.90	99.30	88.20	93.60	68.90	97.20	93.50	96.20	91.23
6	94.40	98.60	90.00	93.20	71.10	98.50	95.40	95.90	92.14
7	97.10	98.60	90.40	94.50	76.40	98.60	94.80	95.60	93.25
8	94.00	97.00	88.10	95.40	65.40	97.70	94.80	96.60	91.13
9	93.60	98.60	87.20	92.10	71.90	94.60	95.10	96.90	91.25
10	93.60	98.70	88.60	93.30	65.00	98.80	96.40	97.40	91.48
Avg	94.14	98.61	88.99	93.16	70.16	97.80	94.67	96.29	91.73
Std	1.36	0.73	1.09	1.42	5.07	1.28	1.04	0.69	1.10
<i>(b) LDA as binary classifiers (CNN-ECOC-LDA)</i>									
1	95.30	98.20	84.40	94.20	84.80	100.00	94.10	99.20	93.78
2	94.20	97.60	84.00	92.20	83.00	99.70	90.40	99.50	92.58
3	95.20	98.10	83.10	93.50	84.00	99.90	94.80	99.70	93.54
4	94.80	98.30	85.80	92.40	86.60	99.30	91.50	99.40	93.51
5	94.60	96.40	84.90	93.10	85.20	99.70	90.00	99.60	92.94
6	95.70	97.10	85.60	93.40	86.90	99.60	92.70	99.60	93.83
7	94.40	97.70	85.20	95.60	82.20	99.40	95.90	99.40	93.73
8	93.60	97.00	81.90	92.60	85.00	100.00	92.60	99.70	92.80
9	95.70	98.00	88.10	91.80	86.50	99.40	91.90	99.90	93.91
10	94.20	98.50	85.20	95.40	83.40	99.90	92.80	99.10	93.56
Avg	94.77	97.69	84.82	93.42	84.76	99.69	92.67	99.51	93.42
Std	0.70	0.67	1.67	1.30	1.61	0.26	1.86	0.24	0.47
<i>(c) CART as binary classifiers (CNN-ECOC-CART)</i>									
1	98.40	99.80	91.90	97.80	92.80	98.90	97.80	96.30	96.71
2	98.70	99.80	94.10	98.90	94.00	99.60	98.40	96.50	97.50
3	98.50	99.60	93.60	98.30	93.20	99.20	97.20	96.60	97.03
4	98.20	99.50	91.70	98.00	90.90	99.20	97.60	97.40	96.56
5	98.60	99.80	93.10	98.20	93.30	99.60	97.40	97.00	97.13
6	98.80	99.90	93.10	98.80	94.60	99.20	98.10	96.90	97.43
7	97.90	99.60	91.00	97.80	91.50	99.40	97.40	95.90	96.31
8	97.90	99.20	90.00	97.80	90.10	99.20	97.00	96.60	95.98
9	98.80	99.70	93.80	98.20	94.10	99.60	97.40	96.60	97.28
10	98.70	99.60	92.80	98.30	92.70	99.60	97.30	96.30	96.91
Avg	98.45	99.65	92.51	98.21	92.72	99.35	97.56	96.61	96.88
Std	0.34	0.20	1.32	0.39	1.46	0.25	0.43	0.42	0.49

that average classification accuracy of 10 iterations in eight methods are not the same.

In addition, the result of pairwise comparisons for one-way ANOVA also is presented in Table 7. In Table 7, the first two column show the pair of compared methods, 'Lower confidence' column contains the lower confidence interval, 'Upper confidence' column contains the upper confidence interval and the 'p value' column contains the p value for

the hypothesis test that the corresponding mean difference is not equal to 0. The two compared methods in each row of 1-10th, 14-22th, 24th, 25th and 27th row are significantly different since the confidence intervals does not include zero and p value is smaller than 0.05. In contrast, the two methods in each row of 11-13th, 23th, 26th and 28th row are not significantly different. From 13th, and 28th rows, it can be seen that

Table 4 continued

Iter	Center	Donut	Edge-loc	Edge-ring	Loc	Random	Scratch	None	Avg
<i>(d) kNN as binary classifiers (CNN-ECOC-kNN)</i>									
1	98.30	100.00	92.90	98.30	93.60	100.00	99.00	96.80	97.36
2	98.60	100.00	93.90	98.60	94.40	100.00	99.00	97.80	97.79
3	98.60	100.00	94.10	98.50	93.70	100.00	99.30	97.60	97.73
4	98.70	100.00	93.90	97.70	94.00	99.90	99.10	96.80	97.51
5	98.70	100.00	92.60	98.00	93.60	100.00	98.80	96.30	97.25
6	98.70	100.00	92.70	97.30	93.60	100.00	99.20	96.70	97.28
7	98.10	100.00	92.30	97.80	90.50	99.80	99.10	97.00	96.83
8	98.80	100.00	93.20	97.80	94.20	99.90	99.30	97.10	97.54
9	97.90	100.00	93.30	98.20	92.00	100.00	99.10	97.40	97.24
10	98.40	100.00	92.20	97.90	90.60	100.00	99.00	97.40	96.94
Avg	98.48	100.00	93.11	98.01	93.02	99.96	99.09	97.09	97.35
Std	0.30	0.00	0.69	0.40	1.45	0.07	0.15	0.46	0.31
<i>(e) LOGISTIC as binary classifiers (CNN-ECOC-LOGISTIC)</i>									
1	99.30	100.00	94.80	98.30	95.00	99.90	98.50	98.40	98.03
2	99.00	99.80	96.20	98.80	95.60	99.80	98.20	98.10	98.19
3	99.20	100.00	94.70	98.80	94.70	99.80	98.70	97.80	97.96
4	98.90	100.00	95.40	98.30	94.80	99.70	98.50	98.40	98.00
5	99.00	100.00	94.90	98.50	93.60	99.80	98.30	98.40	97.81
6	98.80	100.00	93.40	97.90	93.00	99.60	97.60	97.60	97.24
7	99.10	100.00	96.00	98.70	95.00	99.60	98.40	98.60	98.18
8	99.00	100.00	95.10	97.70	93.90	99.00	98.50	98.30	97.69
9	99.20	99.80	94.50	98.40	94.30	99.80	98.70	98.30	97.88
10	99.10	99.80	94.30	98.40	93.70	99.30	97.10	98.00	97.46
Avg	99.06	99.94	94.93	98.38	94.36	99.63	98.25	98.19	97.84
Std	0.15	0.10	0.82	0.36	0.80	0.28	0.51	0.31	0.31
<i>(f) SVM as binary classifiers (CNN-ECOC-SVM)</i>									
1	99.40	100.00	97.10	99.20	97.30	99.80	99.10	98.40	98.79
2	99.00	99.90	95.40	98.80	96.20	100.00	99.00	98.10	98.30
3	99.20	100.00	96.50	98.80	96.70	99.80	98.70	97.80	98.44
4	99.40	100.00	95.20	99.00	95.50	99.90	98.50	98.60	98.26
5	99.50	99.80	96.10	98.90	95.80	99.80	98.40	98.80	98.39
6	99.40	100.00	95.90	99.30	96.00	99.90	98.40	98.40	98.41
7	99.20	99.90	95.40	98.90	96.40	99.90	98.20	98.50	98.30
8	99.10	99.90	96.60	99.10	95.40	99.80	98.70	98.70	98.41
9	99.20	100.00	95.70	98.50	95.90	99.80	98.80	98.20	98.26
10	99.40	100.00	97.20	98.30	97.50	100.00	99.10	98.40	98.74
Avg	99.28	99.95	96.11	98.88	96.27	99.87	98.69	98.39	98.43
Std	0.16	0.07	0.72	0.30	0.71	0.08	0.31	0.30	0.19

the presented method CNN-ECOC-SVM is not significantly different from CNN-SVM and CNN-ECOC-LOGISTIC.

Except the classification accuracy, other classification performance measures of precision, recall, specificity, and F-measures are also considered, and the comparison results are shown in Table 5. For each classification method in Table 8, the iteration with the highest average classification accuracy over 10 iterations is selected for comparison. The selected

iteration number for each corresponding method is shown in 'Iteration no.' column. In Table 8, the best value in each performance measure is marked in bold. It can be seen from Table 8 that CNN-ECOC-SVM respectively shows the highest accuracy, precision, recall, and F-measure of 98.79%, 98.79%, 98.79%, 99.83%, and 98.79%.

Extensive experimental results have proved our original expectation and the main reasons why CNN-ECOC-SVM

Table 5 A summary comparison of classification accuracy

Method	Pattern type (%)								Overall	
	Center	Donut	Edge-loc	Edge-ring	Loc	Random	Scratch	None	Accuracy (%)	Std
CNN	94.78	99.12	79.39	94.36	77.05	98.47	87.89	92.94	90.50	0.27
CNN-SVM	99.09	99.98	94.95	98.78	94.57	99.82	98.28	97.90	97.92	0.30
CNN-ECOC-NB	94.14	98.61	88.99	93.16	70.16	97.80	94.67	96.29	91.73	1.10
CNN-ECOC-LDA	94.77	97.69	84.82	93.42	84.76	99.69	92.67	99.51	93.42	0.47
CNN-ECOC-CART	98.45	99.65	92.51	98.21	92.72	99.35	97.56	96.61	96.88	0.49
CNN-ECOC- <i>k</i> NN	98.48	100.00	93.11	98.01	93.02	99.96	99.09	97.09	97.35	0.31
CNN-ECOC-LOGISTIC	99.06	99.94	94.93	98.38	94.36	99.63	98.25	98.19	97.84	0.31
CNN-ECOC-SVM	99.28	99.95	96.11	98.88	96.27	99.87	98.69	98.39	98.43	0.19

Table 6 ANOVA result

Source	SS	df	MS	F	Prob>F
Between groups	688.091	7	98.2987	382.17	2.8599e-54
Within groups	18.519	72	0.2572	–	–
Total	706.610	79	–	–	–

Table 7 Pairwise comparison test

Index	Method 1	Method 2	Lower confidence	Upper confidence	<i>p</i> value
1	CNN	CNN-SVM	– 8.1291	– 6.7129	0.0000
2	CNN	CNN-ECOC-NB	– 1.9371	– 0.5209	0.0000
3	CNN	CNN-ECOC-LDA	– 3.6251	– 2.2089	0.0000
4	CNN	CNN-ECOC-CART	– 7.0911	– 5.6749	0.0000
5	CNN	CNN-ECOC- <i>k</i> NN	– 7.5541	– 6.1379	0.0000
6	CNN	CNN-ECOC-LOGISTIC	– 8.0511	– 6.6349	0.0000
7	CNN	CNN-ECOC-SVM	– 8.6371	– 7.2209	0.0000
8	CNN-SVM	CNN-ECOC-NB	5.4839	6.9001	0.0000
9	CNN-SVM	CNN-ECOC-LDA	3.7959	5.2121	0.0000
10	CNN-SVM	CNN-ECOC-CART	0.3299	1.7461	0.0005
11	CNN-SVM	CNN-ECOC- <i>k</i> NN	– 0.1331	1.2831	0.1978
12	CNN-SVM	CNN-ECOC-LOGISTIC	– 0.6301	0.7861	1.0000
13	CNN-SVM	CNN-ECOC-SVM	– 1.2161	0.2001	0.3412
14	CNN-ECOC-NB	CNN-ECOC-LDA	– 2.3961	– 0.9799	0.0000
15	CNN-ECOC-NB	CNN-ECOC-CART	– 5.8621	– 4.4459	0.0000
16	CNN-ECOC-NB	CNN-ECOC- <i>k</i> NN	– 6.3251	– 4.9089	0.0000
17	CNN-ECOC-NB	CNN-ECOC-LOGISTIC	– 6.8221	– 5.4059	0.0000
18	CNN-ECOC-NB	CNN-ECOC-SVM	– 7.4081	– 5.9919	0.0000
19	CNN-ECOC-LDA	CNN-ECOC-CART	– 4.1741	– 2.7579	0.0000
20	CNN-ECOC-LDA	CNN-ECOC- <i>k</i> NN	– 4.6371	– 3.2209	0.0000
21	CNN-ECOC-LDA	CNN-ECOC-LOGISTIC	– 5.1341	– 3.7179	0.0000
22	CNN-ECOC-LDA	CNN-ECOC-SVM	– 5.7201	– 4.3039	0.0000
23	CNN-ECOC-CART	CNN-ECOC- <i>k</i> NN	– 1.1711	0.2451	0.4622
24	CNN-ECOC-CART	CNN-ECOC-LOGISTIC	– 1.6681	– 0.2519	0.0017
25	CNN-ECOC-CART	CNN-ECOC-SVM	– 2.2541	– 0.8379	0.0000
26	CNN-ECOC- <i>k</i> NN	CNN-ECOC-LOGISTIC	– 1.2051	0.2111	0.3693
27	CNN-ECOC- <i>k</i> NN	CNN-ECOC-SVM	– 1.7911	– 0.3749	0.0002
28	CNN-ECOC-LOGISTIC	CNN-ECOC-SVM	– 1.2941	0.1221	0.1790

Table 8 A comparison of performance measures

Method	Iteration number	Accuracy	Precision	Recall	Specificity	F-measure
CNN	6th in Table 2	90.90	90.90	90.90	98.70	90.84
CNN-SVM	6th in Table 3	98.35	98.35	98.35	99.77	98.36
CNN-ECOC-NB	4th in Table 4a	93.34	93.45	93.34	99.05	93.28
CNN-ECOC-LDA	9th in Table 4b	93.91	94.61	93.91	99.13	94.01
CNN-ECOC-CART	2nd in Table 4c	97.50	97.50	97.50	99.64	97.50
CNN-ECOC-kNN	2nd in Table 4d	97.79	97.79	97.79	99.69	97.80
CNN-ECOC-LOGISTIC	2nd in Table 4e	98.19	98.19	98.19	99.74	98.19
CNN-ECOC-SVM	1st in Table 4f	98.79	98.79	98.79	99.83	98.79

could obtain such a good performance are that CNN can extract the most discriminative features and combination of ECOC and SVM can improve the classification accuracy.

An interesting observation can be found that each of all eight classification method shows the highest two standard deviation values in edge-loc and loc pattern types. We carefully guess that this phenomenon may come from lack of significant differences between these two defect pattern types. However, we leave the deep research on explanations for our future work.

The WM-811K (Wu et al. 2015) dataset is widely applied in SVM-based method (Wu et al. 2015), OPTICS-SVM (Fan et al. 2016), JLNDA-FD (Yu and Lu 2016), decision tree ensemble learning-based method (Piao et al. 2018), and soft voting ensemble (SVE) (Saqlain et al. 2019) for wafer map defect pattern classification. Since the used dataset in each of them are not completely identical and also different from that used in this research, it is impossible to perform a direct comparison. However, for rough comparison with CNN-ECOC-SVM, only the overall classification accuracies are provided: SVM-based method 94.63%, OPTICS-SVM 94.30%, JLNDA -FD 90.50%, decision tree ensemble learning-based method 90.50%, and SVE 95.87%. Obviously, the overall classification accuracy of CNN-ECOC-SVM is much higher than that of the state-of-the-art methods.

Conclusions

In this research, we present an image-based wafer map defect pattern classification method. The presented method consists of mainly two steps: feature extraction and classification. In the feature extraction step, high-level features are extracted from CNN and in the classification step, the extracted CNN features are used to train ECOC model for wafer map defect pattern classification. In the ECOC model, SVM binary classifiers are employed.

Experimental results conducted on 20000 wafer maps show that the presented method achieves the best classifica-

tion accuracy up to 98.43% in comparison to other wafer map defect pattern classification methods.

There are a lot of work need to be done for future work. In this research, only the one-vs-one decomposition strategy is applied. However, to improve the classification accuracy, other decomposition strategies and other binary classifiers also need to be considered.

Acknowledgements Most of this work was done when the first author was at BISTel. This work was supported by the World Class 300 Project (R&D) (S2641209, “Development of next generation intelligent smart manufacturing solution based on AI & Big data to improve manufacturing yield and productivity”) of the MOTIE, MSS(Korea) and supported by the National Natural Science Foundation of China (Grant Nos. 61702324 and 61911540482) in People’s Republic of China.

References

- Abd-Ellah, M. K., Awad, A. I., Khalaf, A. A., & Hamed, H. F. (2018). Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks. *EURASIP Journal on Image and Video Processing*, 2018(1), 97. <https://doi.org/10.1186/s13640-018-0332-4>.
- Al-Shargie, F., Tang, T. B., Badruddin, N., & Kiguchi, M. (2018). Towards multilevel mental stress assessment using SVM with ECOC: An EEG approach. *Medical & Biological Engineering & Computing*, 56(1), 125–136. <https://doi.org/10.1007/s11517-017-1733-8>.
- Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). Optics: ordering points to identify the clustering structure. *ACM Sigmod Record, ACM*, 28, 49–60. <https://doi.org/10.1145/304182.304187>.
- Ali Bagheri, M., Montazer, G.A., & Escalera, S. (2012). Error correcting output codes for multiclass classification: application to two image vision problems. In *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)* (pp. 508–513). IEEE. <https://doi.org/10.1109/AISP.2012.6313800>.
- Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *Journal of Machine Learning Research*, 2(Dec), 125–137.
- Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. *Wadsworth International Group.*, <https://doi.org/10.1201/9781315139470>.
- Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2, 1–27.

- Chang, C. W., Chao, T. M., Horng, J. T., Lu, C. F., & Yeh, R. H. (2012). 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. <https://doi.org/10.1109/TCPMT.2012.2215327>.
- Chien, C. F., Chang, K. H., & Wang, W. C. (2014). An empirical study of design-of-experiment data mining for yield-loss diagnosis for semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 25(5), 961–972.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>.
- Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2, 263–286.
- Ding, C., & Tao, D. (2018). Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 1002–1014. <https://doi.org/10.1109/TPAMI.2017.2700390>.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning* (pp. 647–655).
- Dorj, U. O., Lee, K. K., Choi, J. Y., & Lee, M. (2018). The skin cancer classification using deep convolutional neural network. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-018-5714-1>.
- Fan, M., Wang, Q., & van der Waal, B. (2016). Wafer defect patterns recognition based on optics and multi-label classification. In *IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (pp. 912–915). IEEE. <https://doi.org/10.1109/IMCEC.2016.7867343>.
- García-Pedrajas, N., & Ortiz-Boyer, D. (2011). An empirical study of binary classifier fusion methods for multiclass classification. *Information Fusion*, 12(2), 111–130. <https://doi.org/10.1016/j.inffus.2010.06.010>.
- Gonzalez, R. C., & Woods, R. E. (2006). *Digital Image Processing* (3rd ed.). Upper Saddle River, NJ: Prentice-Hall Inc.
- Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (1996). *Neural network design* (Vol. 20). Boston: PWS Pub.
- Hansen, M. H., Nair, V. N., & Friedman, D. J. (1997). Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects. *Technometrics*, 39(3), 241–253. <https://doi.org/10.2307/1271129>.
- Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. In *Advances in Neural Information Processing Systems* (pp. 507–513). <https://doi.org/10.1214/aos/1028144844>
- Huang, F.J., & LeCun, Y. (2006). Large-scale learning with svm and convolutional nets for generic object categorization. In *Proceeding of Computer Vision and Pattern Recognition Conference (CVPR '06)*. <https://doi.org/10.1109/CVPR.2006.164>
- Jin, C. H., Na, H. J., Piao, M., Gouchol, P., & Ho, R. K. (2019). A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map. In *IEEE Transactions on Semiconductor Manufacturing*.
- Kim, M., Lee, M., An, M., & Lee, H. (2019). Effective automatic defect classification process based on cnn with stacking ensemble model for tft-lcd panel. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-019-01502-y>
- Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882). <https://doi.org/10.3115/v1/D14-1181>.
- Kong, E.B., & Dietterich, T.G. (1995). Error-correcting output coding corrects bias and variance. In *Machine Learning Proceedings* (pp. 313–321). Elsevier. <https://doi.org/10.1016/B978-1-55860-377-6.50046-3>.
- Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105). <https://doi.org/10.1145/3065386>.
- Kyeong, K., & Kim, H. (2018). Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. In *IEEE Transactions on Semiconductor Manufacturing*. <https://doi.org/10.1109/TSM.2018.2841416>.
- Lin, H., Li, B., Wang, X., Shu, Y., & Niu, S. (2019). Automated defect inspection of LED chip using deep convolutional neural network. *Journal of Intelligent Manufacturing*, 30(6), 2525–2534.
- Liu, Y. (2006). Using svm and error-correcting codes for multiclass dialog act classification in meeting corpus. In *Ninth International Conference on Spoken Language Processing*.
- Lorena, A. C., De Carvalho, A. C., & Gama, J. M. (2008). A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, 30(1–4), 19. <https://doi.org/10.1007/s10462-009-9114-9>.
- Matlab. (2018). <https://www.mathworks.com/products/matlab.html>.
- Nakazawa, T., & Kulkarni, D. V. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing*, 31(2), 309–314. <https://doi.org/10.1109/TSM.2018.2795466>.
- Nilsson, N. J. (1965). *Learning machines*. New York: McGrawHill.
- Niu, X. X., & Suen, C. Y. (2012). A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4), 1318–1325. <https://doi.org/10.1016/j.patcog.2011.09.021>.
- Ooi, M. P. L., Sok, H. K., Kuang, Y. C., Demidenko, S., & Chan, C. (2013). Defect cluster recognition system for fabricated semiconductor wafers. *Engineering Applications of Artificial Intelligence*, 26(3), 1029–1043. <https://doi.org/10.1016/j.engappai.2012.03.016>.
- Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in Neural Information Processing Systems* (pp. 2643–2651).
- Othman, K. M., & Rad, A. B. (2019). An indoor room classification system for social robots via integration of CNN and ECOC. *Applied Sciences*, 9(3), 470. <https://doi.org/10.3390/app9030470>.
- Piao, M., Jin, C. H., Lee, J. Y., & Byun, J. Y. (2018). Decision tree ensemble based wafer map failure pattern recognition based on radon transform based features. *IEEE Transactions on Semiconductor Manufacturing*, 31(2), 250–257. <https://doi.org/10.1109/TSM.2018.2806931>.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. <https://doi.org/10.1007/BF00116251>.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Amsterdam: Elsevier.
- Rakhlin, A., Shvets, A., Iglovikov, V., & Kalinin, A. A. (2018). Deep convolutional neural networks for breast cancer histology image analysis. In *International Conference Image Analysis and Recognition* (pp. 737–744). Springer. https://doi.org/10.1007/978-3-319-93000-8_83.
- Saqlain, M., Jargalsaikhan, B., & LEE, J. Y. (2019). A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. In *IEEE Transactions on Semiconductor Manufacturing*. <https://doi.org/10.1109/TSM.2019.2904306>.
- Tang, Y. (2013). Deep learning using linear support vector machines. arXiv preprint [arXiv:1306.0239](https://arxiv.org/abs/1306.0239)
- Wang, C. H. (2008). Recognition of semiconductor defect patterns using spatial filtering and spectral clustering. *Expert Systems with Applications*, 34(3), 1914–1923. <https://doi.org/10.1016/j.eswa.2007.02.014>.
- Wang, C. H., Kuo, W., & Bensmail, H. (2006). Detection and classification of defect patterns on semiconductor wafers. *IIE Transactions*, 38(12), 1059–1068. <https://doi.org/10.1080/07408170600733236>.

- Wu, M. J., Jang, J. S. R., & Chen, J. L. (2015). Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing*, 28(1), 1–12. <https://doi.org/10.1109/TSM.2014.2364237>.
- Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., & Stolcke, A. (2018). The microsoft 2017 conversational speech recognition system. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5934–5938). IEEE. <https://doi.org/10.1109/ICASSP.2018.8461870>.
- Xue, D. X., Zhang, R., Feng, H., & Wang, Y. L. (2016). Cnn-svm for microvascular morphological type recognition with data augmentation. *Journal of Medical and Biological Engineering*, 36(6), 755–764. <https://doi.org/10.1007/s40846-016-0182-4>.
- Yu, J., & Lu, X. (2016). Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. *IEEE Transactions on Semiconductor Manufacturing*, 29(1), 33–43. <https://doi.org/10.1109/TSM.2015.2497264>.
- Yuan, T., Bae, S. J., & Park, J. I. (2010). Bayesian spatial defect pattern recognition in semiconductor fabrication using support vector clustering. *The International Journal of Advanced Manufacturing Technology*, 51(5), 671–683. <https://doi.org/10.1007/s00170-010-2647-x>.
- Zheng, G., Qian, Z., Yang, Q., Wei, C., Xie, L., Zhu, Y., et al. (2008). The combination approach of SVM and ECOC for powerful identification and classification of transcription factor. *BMC Bioinformatics*, 9(1), 282. <https://doi.org/10.1186/1471-2105-9-282>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.