Journal of Intelligent Manufacturing (2022) 33:831–844
https://doi.org/10.1007/s10845-020-01687-7

Ensemble convolutional neural networks with weighted majority for wafer bin map pattern classification

Chia-Yu Hsu[1] · **Ju-Chien Chien**[2,3]

Received: 9 February 2020 / Accepted: 1 October 2020 / Published online: 17 October 2020

Abstract
Wafer bin maps (WBM) provides crucial information regarding process abnormalities and facilitate the diagnosis of low-yield problems in semiconductor manufacturing. Most studies of WBM classification and analysis apply a statistical-based method or machine learning method operating on raw wafer data and extracted features. With increasing WBM pattern diversity and complexity, the useful features for effective WBM recognition are highly dependent on domain knowledge. This study proposes an ensemble convolutional neural network (ECNN) framework for WBM pattern classification, in which a weighted majority function is adopted to select higher weights for the base classifiers that have higher predictive performance. An industrial WBM dataset (namely, WM-811K) from a wafer fabrication process was used to demonstrate the effectiveness of the proposed ECNN framework. The proposed ECNN has superior performance in terms of precision, recall, F1 and other conventional machine learning classifiers such as linear regression, random forest, gradient boosting machine, and artificial neural network. The experimental results show that the proposed ECNN framework is able to identify common WBM defect patterns effectively.

Keywords Wafer bin map · Deep learning · Convolutional neural network · Ensemble classification · Weighted majority · Semiconductor manufacturing

Introduction

With the rapid development of semiconductor manufacturing technology, controlling the production process effectively is critical for minimizing process variation to enhance yield (Chien et al. 2013; Hsu 2014). Circuit probe (CP) testing is used to evaluate each die on the wafer after the wafer fabrication processes. Wafer bin maps (WBMs) represent the results of a CP test and provide crucial information regarding process abnormalities, facilitating the diagnosis of low-yield problems in semiconductor manufacturing (Hsu and Chien 2007; Chien et al. 2013; Hsu 2015). A WBM is a two-dimensional defect pattern which is transformed into binary values and used to select the testing bin code. The dies that pass the functional test are denoted as 0 and the defective dies are denoted as 1. Depending on the various sources of variation, the WBM consists of random, systematic, or mixed defects generated during semiconductor fabrication (Hsu and Chien 2007; Hsu et al. 2020). Random defect patterns are caused by random particles or noises during the manufacturing process. Systematic defect patterns show spatial correlation across wafers such as Center, Donut, Edge-Local, Edge-Ring, Local, Near-full, Random, Scratch, and None as shown in Fig. 1. Based on the systematic patterns, domain engineers can rapidly determine the causes of defects (Hsu and Chien 2007). Mixed failure patterns combine the random and systematic defects on a wafer as shown in Fig. 1. The mixed pattern can be identified if the extent of the random defects is slight.

One of the most effective ways to ensure that the causes of process variation can be assigned is to analyze the spatial defect patterns on the wafers. WBMs provide important information for engineers to identify the potential root cause of errors rapidly by recognizing patterns correctly. As the driving force for semiconductor manufacturing technology,

✉ Chia-Yu Hsu
chiayuhsu@mail.ntut.edu.tw

[1] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan

[2] Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan

[3] Artificial Intelligence for Intelligent Manufacturing Systems (AIMS) Research Center, Ministry of Science & Technology, Hsinchu 30013, Taiwan

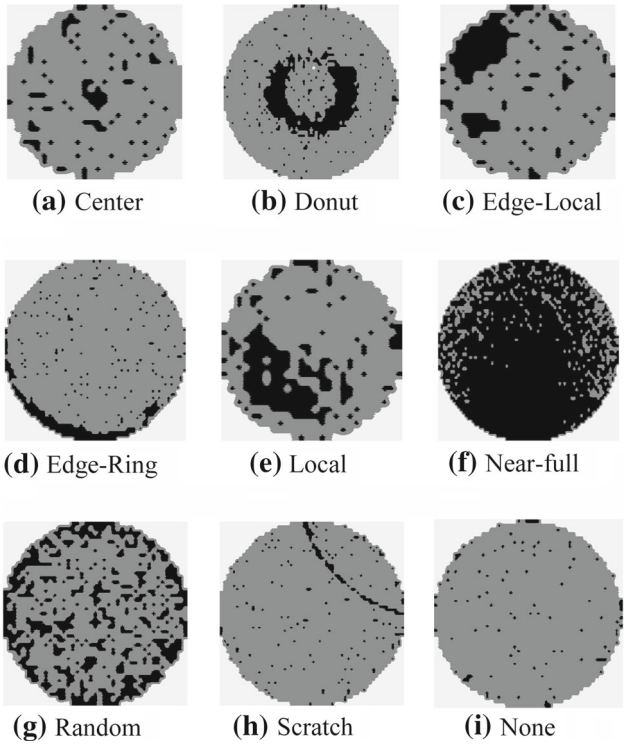

Fig. 1 Examples of wafer bin maps

correct classification of WBM patterns becomes more difficult, because patterns may vary in size, density, rotation angle and noise level. Nowadays, most companies still rely on engineers' experiences of visual inspections and personal judgment to classify the map patterns. This manual approach is not only subjective, and inconsistent, but is also very time consuming and inefficient.

According to the input to the classification model, the method of WBM pattern classification can be separated into three approaches: raw wafer data, extracted features, and WBM image. Using raw wafer data, ART neural network has been used to construct clusters of WBM and then domain experts could recognize the type of these clusters quickly, rather than identifying each WBM (Hsu and Chien 2007; Chien et al. 2013; Liu and Chien 2013). In order to enhance the signal and remove the noise (ESRN), morphology methods, statistical tests, and moment invariant techniques were used to reduce noise and improve the clearness of the pattern. Using ART-based neural networks for WBM clustering is advantageous if the new WBM defect pattern is unknown. Moreover, several shape-specific probability density functions (pdf), such as principal curve, bivariate normal distribution, and spherical shell, have been proposed to detect the regions of defect patterns (Hwang and Kuo 2007; Yuan and Kuo 2008a, b; Yuan et al. 2011). Using these model-based approaches is better for building a detection model where multiple defect patterns occur on a wafer. Jeong et al. (2008) proposed a spatial correlogram to represent the spatial autocorrelation across the wafer, and then transformed the raw wafer data into a one dimension series. Different types of WBM defect pattern have particular trends in a spatial correlogram and this dynamic time wrapping method is used to calculate the similarity between two series. The main shortcomings of using raw wafer data are the heavy computation cost for a large-scale WBM dataset, and low accuracy due to the amount of noise on a wafer and the consequent need for data pre-processing for signal enhancement and noise reduction (Hsu and Chien 2007).

In the second approach, feature generation from raw wafer data has been used to build a classification model. In particular, density-based features (Fan et al. 2016), geometry-based features (Wu et al. 2015), radon-based features (Piao et al. 2018), and rotation-invariant features (Wang and Chen 2019) were used for feature extraction, and the features extracted used as input for various classifiers such as SVM (Baly and Hajj 2012; Wu et al. 2015) and decision tree (Piao et al. 2018). For example, Wu et al. (2015) selected geometry-based and radon-based features, and SVM classifier to identify WBM defect patterns. A large-scale WBM dataset including eight systematic defect patterns and one normal pattern, called WM-811K, was used for performance evaluation. Yu and Lu (2016) presented a joint local and non-local linear discriminant analysis (JLNDA) with four kinds of features to detect the WBM failure patterns. Because no individual machine learning classifier is best for all kinds of dataset, an ensemble method which combines all individual classifiers, can be used to improve the final prediction accuracy (Galar et al. 2011). The ensemble results are better than any individual classifier (Saha and Ekbal 2013). Piao et al. (2018) proposed a decision tree ensemble learning-based WBM defect pattern recognition method based on radon transform-based features. However, relying features that are generated in advance is not enough to cover all kinds of WBM failure patterns. Saqlain et al. (2019) extracted 66 features including density-based, geometry-based, and radon-based features from raw wafer images and applied a voting ensemble classifier incorporating logistic regression (LR), random forests (RF), gradient boosting machine (GBM), and artificial neural network (ANN) with three kinds of features. These were used for WBM defect classification. It is essential to capture useful features to improve the performance of machine learning classifiers, but the effective features were extracted manually and relied on specific domain judgements for various WBM defect patterns (Yu 2019). This approach can be improved by using feature learning from the WBM image directly, to generate the effective features or kernels for different types of WBM defect patterns without making significant modification (Yu et al. 2019a).

Recently, convolutional neural network (CNN) has become a standard image classification method (Krizhevsky

et al. 2012), which learns the critical features for image classification from an image automatically, without manual feature extraction in advance. Unlike manual feature extraction, CNN builds the classification model and extracts the effective features at the same time. CNN models have been applied to defect inspection in battery electrode (Badmos et al. 2020), solar cell surface (Chen et al. 2020), laser manufacturing (Gonzalez-Val et al. 2020), light-emitting diode (Lin et al. 2019), and panel display (Liu et al. 2020). Additionally, CNN-based approaches are also receiving growing attention for WBM defect pattern classification and outperform other machine learning-based methods with high accuracy (Kyeong and Kim 2018; Nakazawa and Kulkarni 2018, 2019; Yu 2019; Yu et al. 2019a, b). For example, Nakazawa and Kulkarni (2018) used CNN for WBM defect pattern classification. Similarly, Kyeong and Kim (2018) applied CNN to recognize failure patterns, where each type of WBM pattern needed an individual CNN model. To build a classifier with several defect patterns and a non-defect pattern, Yu et al. (2019) used two CNN models with 8-layers and 13-layers for WBM inspection and WBM pattern classification. An enhanced stacked denoising autoencoder (ESDAE) with manifold regularization was proposed for inspecting defective WBMs and WBM pattern classification (Yu 2019). The CNN-based approaches can capture effective features without manual intervention and are easy to apply without specific domain knowledge. However, the computation effort is large and many WBM images are essential for CNN implementation. In addition, the class imbalance problem must be taken into account, because defects, and WBM defect patterns, are relatively rare in semiconductor fabrication. To solve the class imbalance issue, the undersampling technique for patterns with no defect is applied first to produce a binary classification model to identify normal or abnormal patterns. However, it is difficult to determine a suitable threshold for selection of normal wafers with high accuracy.

To bridge the gap between the existing studies, this study proposes an ensemble CNN (ECNN) framework with weighted majority for WBM defect pattern classification. State-of-the-art CNN models, such as LeNet, AlexNet, and GoogleNet, are used as base classifiers. To incorporate the advantages of different base classifiers for identifying WBM defect patterns, a weighted majority is used, in which the weights assigned to the base classifiers depend on its recognition rate for each WBM defect pattern. The proposed ensemble CNN framework is evaluated for its performance on the WM-811K dataset (Wu et al. 2015). The performance of ECNN was compared with several individual classifiers with extracted features. The CNN-based model has better classification accuracy than the existing methods.

The remainder of this paper is organized as follows. Next section describes the details of the proposed ECNN framework. Then, performance comparisons for various WBM classification models are examined in relation to the WM-811K dataset. Finally, this study concludes with a discussion of our contributions and further research directions.

Proposed ECNN framework

The proposed ECNN framework with weighted majority (ECNN) includes three individual CNN classifiers. WBMs are typically accompanied by random noise. Most existing studies about WBM classification used the morphology method to enhance the signal and remove the noise (ESRN). However, this study develops an end-to-end model for WBM classification without performing ESRN and the critical features of WBM classification are extracted automatically. Figure 2 illustrates the proposed ECNN framework for WBM pattern classification, in which the WBM dataset is split into a training dataset, a validation dataset, and a testing dataset. The training dataset is used to build the classification model and the validation dataset is used to examine the model performance and tuning of hyperparameter setting. The testing dataset is used to evaluate the final classification results. A weighted majority function for each base CNN model was adopted, using weights for the CNN classifiers based on their recognition performance of each WBM defect pattern in the validation dataset. Before further CNN model training, each raw wafer data is transformed into a WBM image with 300×300 pixels and the WBMs are subtracted by mean image per channel.

Base classifier training and weighted majority are the two main steps of the proposed ensemble model. In this study, we examined the performance of potential classifiers and selected state-of-the-art CNN models, LeNet (LeCun et al. 1998), AlexNet (Krizhevsky et al. 2012), and GoogleNet (Inception-v1) (Szegedy et al. 2015), which have different numbers of convolution and pooling layers. That means the decision boundary of each base classifier should be as different as possible. In order to extract the features from WBM data rather than predefined features, the CNN-based classifiers are LeNet (5 layers), AlexNet (8 layers), and GoogleNet (22 layers).

CNN is a neural network that is effective for analyzing image data. Convolution is used to extract the critical information from the original image. Figure 3 illustrates the CNN structure in WBM classification, in which an input layer, convolution layers, pooling layers, fully connected layers and an output layer are selected. The input layer is used to receive two-dimensional WBM images as input. The convolutional layer is used to compute a dot product of a small data region and a filter. For example, the filter with 2×2 size is moved across the whole WBM and then the images after convolution are called feature maps. Typically, convolution decreases the size of feature map, but we can maintain the size by adding

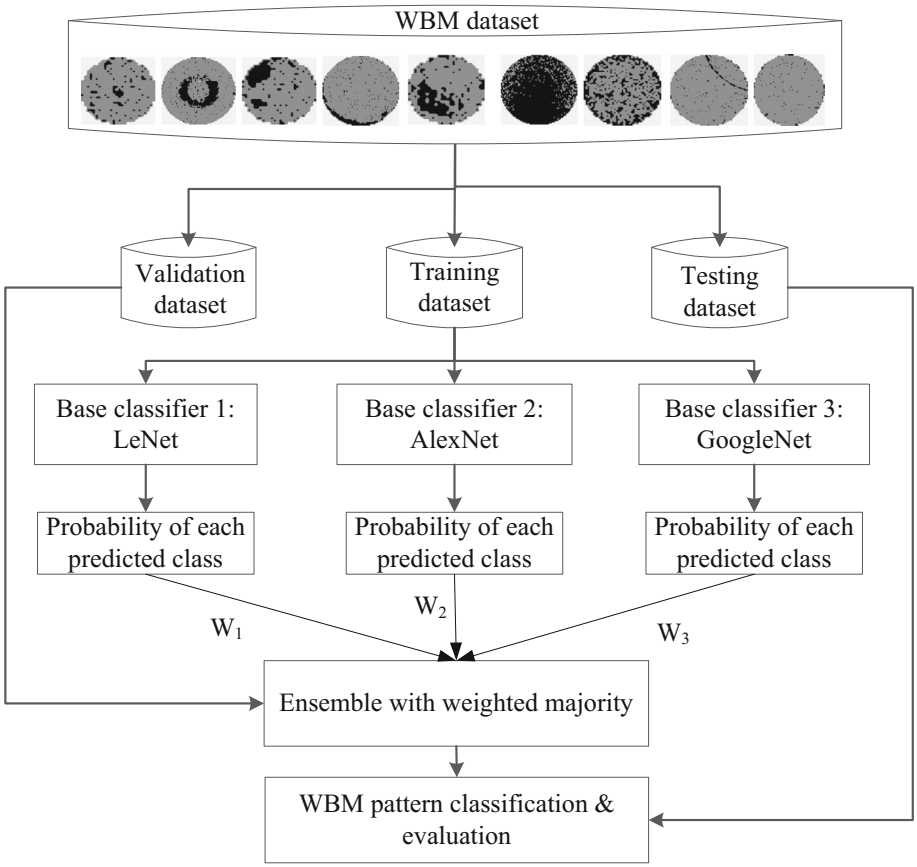

Fig. 2 Proposed ECNN framework

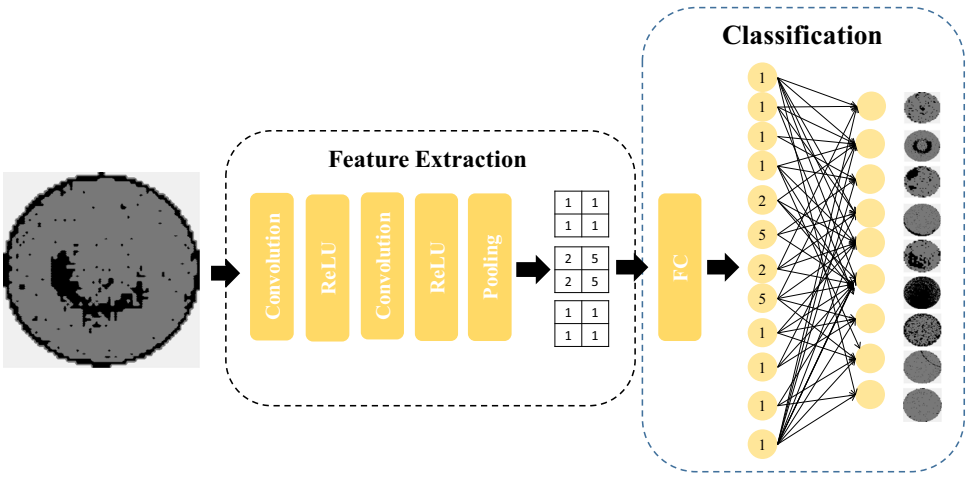

Fig. 3 Illustration of CNN structure in WBM classification

padding of zeros at the edge of the original image data of WBM. Different kinds of filters can result in different feature maps representing different features and the number of filters must be determined in advance. In order to keep positive information in the feature map, a rectified linear unit (ReLU) activation function is usually stacked with the convolutional layer. The feature extraction consists of the convolutional layer, followed by a pooling layer, which reduce the size of the feature map by extracting a local feature such as local maximum or local average. After the convolutional and pooling layers, a fully connected (FC) layer is used for WBM classification. The FC layer is a multilayer perception neural network. Finally, the output layer generates the probability value using the softmax function and determines the class of WBM by the maximum probability value.

The proposed ensemble classifier with weighted majority is unlike the bagging ensemble approach, which uses an averaging model that combines the prediction from each base classifier equally. The diversity of ensemble classifiers ensures that each selected base classifier has a unique perfor-

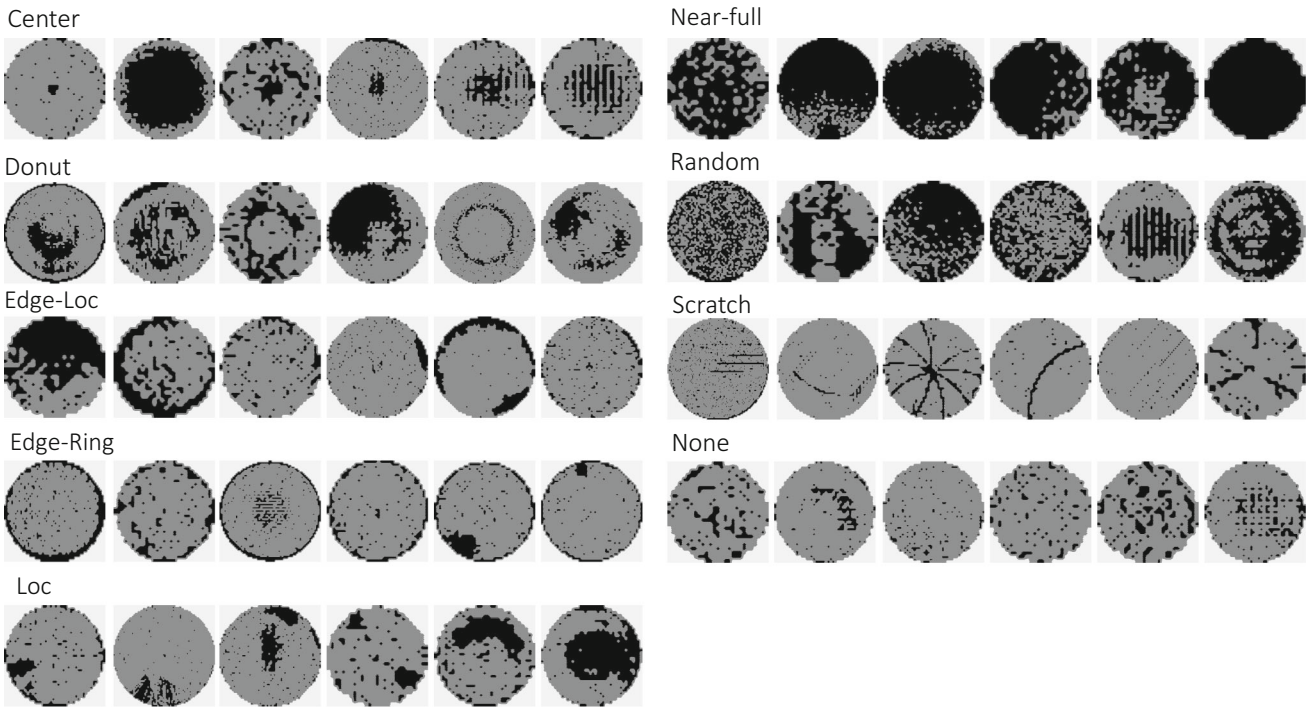

Fig. 4 Example of WBMs in WM-811K

mance when classifying the WBM defect patterns. However, there may be base classifiers which could be useful for classifying certain WBM defect patterns and should be incorporated to extend the diversity and improve performance for certain WBM defect patterns. Similarly, some base classifiers may have less power to identify some WBM defect patterns and their influence should be reduced in WBM defect pattern classification. In order to merge the various results of the three base classifiers, a weighted majority function that enables multiple classifiers to contribute to WBM defect pattern classification in proportion to their estimated performance is used as follows:

$$C_i = \arg\max P(\mathbf{y}|\mathbf{X}_i) \tag{1}$$

and

$$P(\mathbf{y}|\mathbf{X}_i) = \sum_{k=1}^{M} \mathbf{w}_k P_k(\mathbf{y}|\mathbf{X}_i) \tag{2}$$

where $\mathbf{X}_i$ denotes the ith input WBM image, $\mathbf{y}$ is the vector of classified label. For example, assuming there are five WBM defect classes, the first class is denoted as (1, 0, 0, 0, 0). The parameter M is the number of base classifiers that are considered in the ensemble model. The probability of $P_k(\mathbf{y}|\mathbf{X}_i)$ denotes the output value of kth base classifier which is calculated from the softmax function in the output layer of kth base classifier. The weight $\mathbf{w}_k$ is a vector of weight for each WBM defect class which is determined according to the fraction of the total amount of relevant WBMs that were actually retrieved. The weights from the validation dataset for the ensemble classifier during model training are more robust and avoid overfitting.

Evaluation and discussion

Data description

The performance of proposed ensemble CNN was evaluated using the WBM dataset, WM-811K (Wu et al. 2015), which consists of 811,459 WBMs collected from a real-world fabrication. 172,950 of the WBMs (21.3%) have been labeled by domain engineers. There are nine types of WBM used in model evaluation including *Center* (4294), *Donut* (555), *Edge-Loc* (5189), *Edge-Ring* (9680), *Loc* (3593), *Near-full* (149), *Random* (866), *Scratch* (1193) and *None* (147,431) as shown in Fig. 4. The pattern *Center* is a block of defect near the central area of a wafer. The pattern *Donut* is a hollow and block defects located within the wafer. The patterns *Edge-Loc* and *Edge-Ring* are systematic defects with cluster and moon shape at the wafer edge. The pattern *Loc* is a cluster defect within the wafer. The pattern *Near-full* means that the defects cover most of the wafer. The pattern *Random* indicates that a small number of defective areas are located on a wafer randomly. The pattern *Scratch* is a defect in a straight

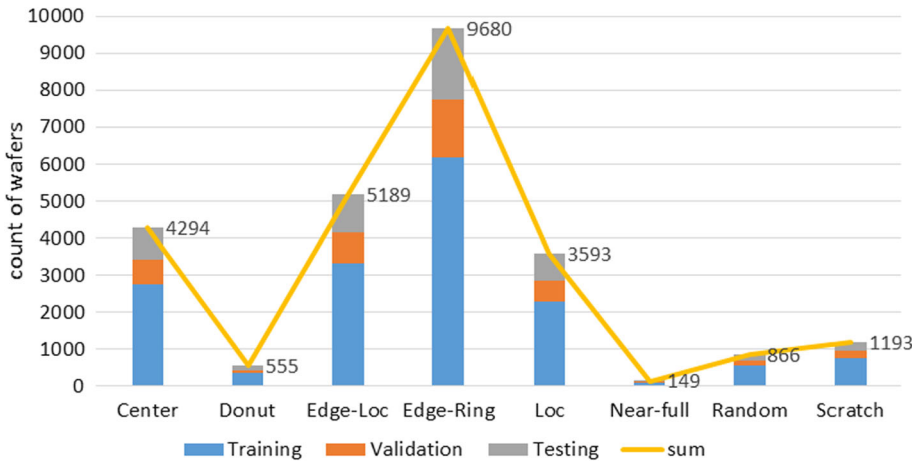

Fig. 5 Count of WBMs for experiments (Training: 64%, Validation: 16%, Testing: 20%)

line or curve. The pattern *None* indicates that there is no systematic pattern, and the resulting pattern was caused by random particles falling on a wafer and results in randomly distributed defects. The model performance was evaluated by fivefold cross-validation, and then the 172,950 WBMs was divided into training (64%), validation (16%), and testing (20%) datasets for each type of defect pattern. Figure 5 shows the number of each failure pattern, with the exception of the *None* pattern. The distribution of each type is imbalanced. In order to take into account the different die size, each record of raw data of a WBM is transformed into an image with 300×300 pixels.

Hyperparameter setting of CNN models

The performance of CNN model training is influenced by the hyperparameter setting. The Adam optimizer was initially used with the following setting of hyper-parameters: the epoch is 10, the batch size is 64, and the learning rate is 0.0001. The convergence of loss and accuracy in the validation dataset were used to evaluate whether the model is adequate or not. Figure 6 shows the loss and accuracy in the validation dataset for LeNet, AlexNet, and GoogLeNet. It shows the good convergence of each CNN model. Both loss and accuracy vary only slightly with the increase of training epoch. Therefore, the epoch for further analysis is fixed as 10.

Hyperparameter setting and network architectures are critical in neural network models. The network architectures are selected from three base CNN models: LeNet, AlexNet, and GoogLeNet. These CNN models are used as base classifiers for the proposed ECNN model. The hyperparameter settings such as batch size, learning rate, and optimizer are compared in fivefold cross-validation. The initial setting of batch size is 64, learning rate is 0.0001, and Adam optimizer is used for weight optimization.

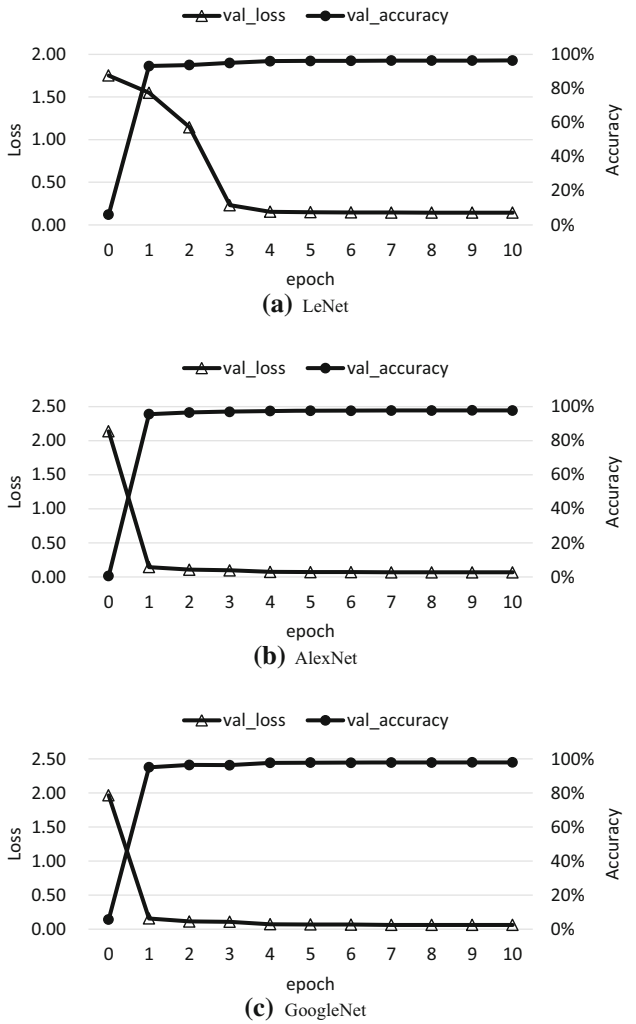

Fig. 6 Illustration of validation performance in CNN model training

Batch size indicates the frequency of weight updates. To compare the value of batch size, the learning rate is set as 0.0001 and the Adam optimizer is used. The recall value

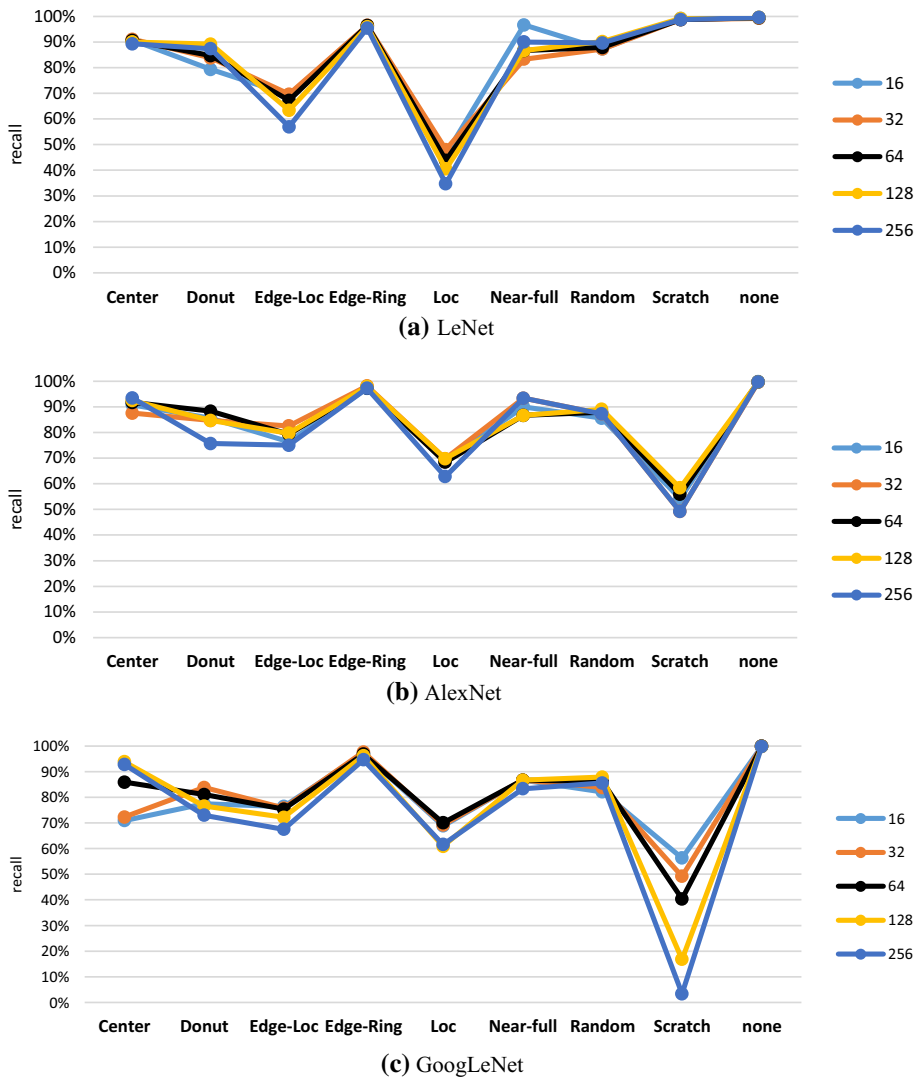

Fig. 7 Recall of different batch sizes in three CNN models

of different batch size in three CNN models are shown in Fig. 7. The patterns *Center*, *Donut*, *Edge-Ring*, *Near-full*, and *None* have higher recall value than patterns *Edge-Loc*, *Loc*, and *Scratch*. Figure 7a shows that the patterns *Edge-Loc* and *Loc* are identified less well by LeNet than other types of defect. Figure 7b, c show that the pattern *Scratch* produces worse performance in both AlexNet and GoogLeNet than other types of defect.

Learning rate is evaluated, based on a batch size of 64 and the Adam optimizer is used. Figure 8 shows the recall for different learning rates the in three CNN models. Figure 8a shows the similar performance of nine WBM defect patterns using various learning rates in LeNet. There are large difference among various learning rates in AlexNet as shown in Fig. 8b. In particular, the learning rate of 0.0001 has higher recall than the others. Figure 8c also shows that the low learning rate works better for GoogleNet, except for the patterns *Random* and *Scratch*.

The optimizer is used to update weights in CNN model training. Five types of optimizers were used: mini-batch gradient descent (SGD), adaptive gradient algorithm (AdaGrad) (Duchi et al. 2011), AdaDelta (Zeiler 2012), Root Mean Square Propagation (RMSprop) (Tieleman and Hinton 2012), and adaptive moment estimation (Adam) (Kingma and Ba 2014). They are compared in terms of recall in the nine WBM defect patterns. The batch size was set to 64 and the learning rate to 0.0001. Figure 9 shows the recall of different optimizers in the three CNN models. There are large differences of recall for each WBM defect pattern. The choice of optimizer has a large impact on CNN models. For these three CNN models, AdaDelta optimizer has the worst performance of those tested. For patterns *Center*, *Donut*, *Edge-Loc*, *Edge-Ring*, *Loc*, *Random*, and *Scratch*, Adam and RMSProp perform better than SGD and AdaGrad.

To summarize the performance for various hyperparameter settings, Table 1 shows the average recall for each WBM

Fig. 8 Recall of different learning rates in three CNN models

defect pattern by each of the three CNN models. The average recall denotes the mean of recall for the nine WBM patterns. The learning rate of 0.0001 is better than 0.0005 or 0.0010. The learning made little difference to recall when using the LeNet model. In terms of optimizer, Adam optimizer is the best of those tested. To determine the batch size, the best number is not the same for the three CNN models. A large batch size means that weights are updated less often than for small batch sizes. Table 2 shows the time for model training decreases as batch size increases. Considering the trade-off between model performance and speed of model training, a batch size of 64 is used for the further ensemble CNN model.

After determining the hyperparameter settings of the base classifiers, the classification performance of LeNet, AlexNet, and GoogleNet is examined in a fivefold cross-validation. Figure 10 shows the average recall and the standard deviation for model training. The patterns *Edge-Ring*, *Near-full*, and *None* have high recall (over 90%). The results in patterns *Center*, *Donut*, and *Random, and Scratch* have at least one CNN model with high recall. The average recall of patterns *Edge-Loc* and *Loc* are lower than 80%. Examining the standard deviation of each CNN for these nine WBM patterns, LeNet, with few network layers, has the smallest deviation.

Performance evaluation with ensemble CNN

In this section, the accuracy of the proposed ensemble CNN model using weighted majority, which incorporates the performance of the various CNN classifiers for each WBM defect pattern is evaluated. Table 3 presents an accruacy comparison among eight individual classifiers and three ensemble classifiers. First, we examine the performance of the three CNNs (LeNet, AlexNet, and GoogleNet) with the performance of the other six individual classifiers, WMFPR (Wu et al. 2014), LR, RF, GBM, ANN (Saqlain et al. 2019), and CNN with 3 stacked convolution-pooling structures (Kyeong and Kim 2018). In total, 116 predefined features were used as input feature for support vector machine (SVM) classi-

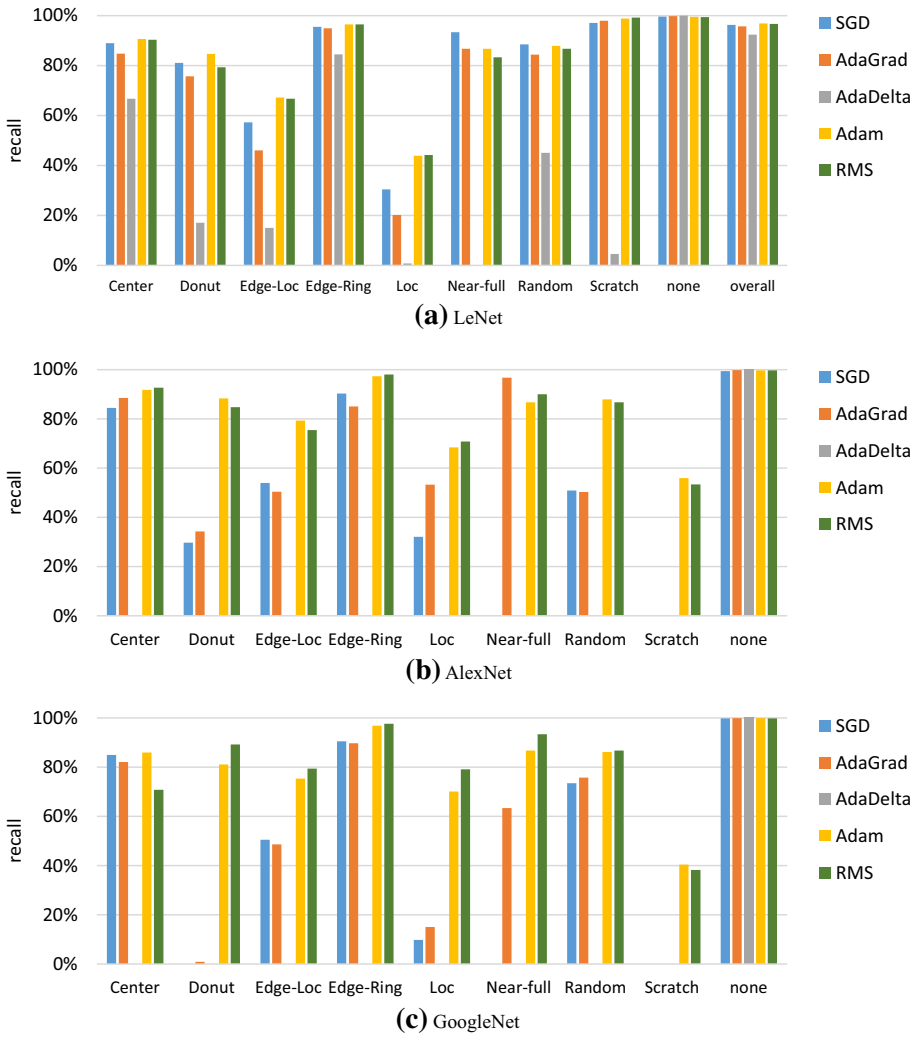

Fig. 9 Recall of different optimizers in three CNN models

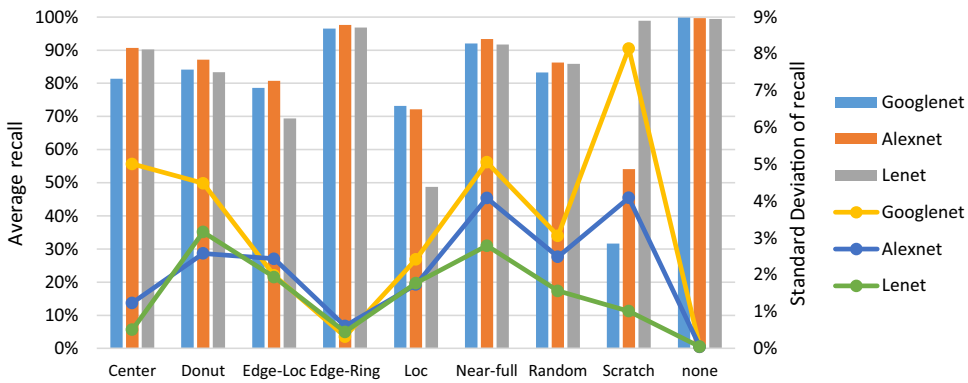

Fig. 10 CNN model performance in fivefold cross-validation

fier in WMFPR, including 36 geometry-based features (with and without noise reduction) and 80 Radon-based features (with and without noise reduction). Four individual classifiers, LR, RF, GBM, and ANN, were selected, with 66 features including density-based (20), Radon-based (40), and geometry-based (6) features and their input features. In addition to input and output layers, the CNN consists of three convolutional and pooling layers, in which the ReLU activation function is added after each convolutional layer. The CNN with 3 stacked convolution-pooling layers (Kyeong and Kim 2018) adopted a ReLU activation function after each convolutional layer. This approach may reduce the diversity of feature extraction because of repeated transformation by the ReLU activation function. The input of LeNet, AlexNet,

Table 1 Average recall of various parameter settings

Hyperparameter setting		LeNet (%)	AlexNet (%)	GoogleNet (%)
Batch size	16	**84.92**	83.16	79.51
	32	84.18	83.56	79.81
	64	83.94	83.90	**82.69**
	128	83.84	**84.25**	76.75
	256	82.41	81.53	73.50
Learning rate	0.0010	83.21	66.45	79.89
	0.0005	82.94	71.45	80.69
	0.0001	**83.40**	**83.44**	**82.26**
Optimizer	SGD	81.30	48.96	58.56
	AdaGrad	76.70	62.01	63.83
	AdaDelta	37.10	11.11	19.77
	RMSProp	82.83	83.47	82.62
	Adam	**83.94**	**83.90**	**82.69**

Table 2 Computation time (second) of various batch size

Batch size	LeNet	AlexNet	Googlent
16	7920	5040	5460
32	4080	2524	2928
64	2235	1359	1704
128	1290	793	1080
256	799	529	812

and GoogleNet are raw WBM images rather than predefined features. The accuracy is a weighted average based on the accuracy for each type of WBM pattern and their percentage recall in the training sample. For example, the weight of pattern *None* is 0.852 (147431/172950). The selected conventional CNN models, namely LeNet (96.94%), AlexNet (97.75%), and GoogleNet (97.35%) outperform the WMFPR (94.63%), LR (95.06%), RF (94.42%), GBM (95.35%), ANN (95.25%), and CNN (89.80%) models in terms of accuracy.

To further compare the performance of ensemble classifier, two existing ensemble classifiers for WBM classification were selected for comparison. These were the majority-voting ensemble (MVE) and the soft-voting ensemble (SVE). Both MVE and SVE were weighted by the results from LR, RF, GBM, and ANN. The proposed ECNN with weighted majority has higher accuracy (98.57%) than MVE (95.74%) and SVE (95.87%) which are ensembles of LR, RF, GBM, and ANN. The three base CNN models are not only superior to WMFPR, LR, RF, GBM, ANN but also have higher accuracy than MVE and SVE.

As the number of WBM in each class is unbalanced in WM-811K, we also examine the classification performance of various base classifiers for each WBM defect pattern in terms of precision, recall, and F_1. The selected LeNet, AlexNet, and GoogleNet models outperform the LR, RF, GBM, and ANN models and have higher precision and recall as shown in Figs. 11 and 12. The performance for patterns *Loc* and *Scratch* are better in terms of precision than other WBM patterns. Comparing these base classifiers, LR is the worst, and LR could be replaced by any of the other base classifiers in ensemble classification. The proposed ECNN not only has 0.82% improvement in terms of accuracy as shown in Table 3, but also is superior to individual LeNet, AlexNet, and GoogleNet for various WBM pattern in terms of precision and recall. Figure 13 shows the F_1 value,

Table 3 Accuracy comparison of different classifiers

Model	Classifier type	Accuracy (%)
WMFPR (SVM)	Individual	94.63
LR	Individual	95.06
RF	Individual	94.42
GBM	Individual	95.35
ANN	Individual	95.25
CNN	Individual	89.80
LeNet	Individual	96.94
AlexNet	Individual	97.75
GoogleNet	Individual	97.35
MVE	Ensemble	95.74
SVE	Ensemble	95.87
ECNN	Ensemble	98.57

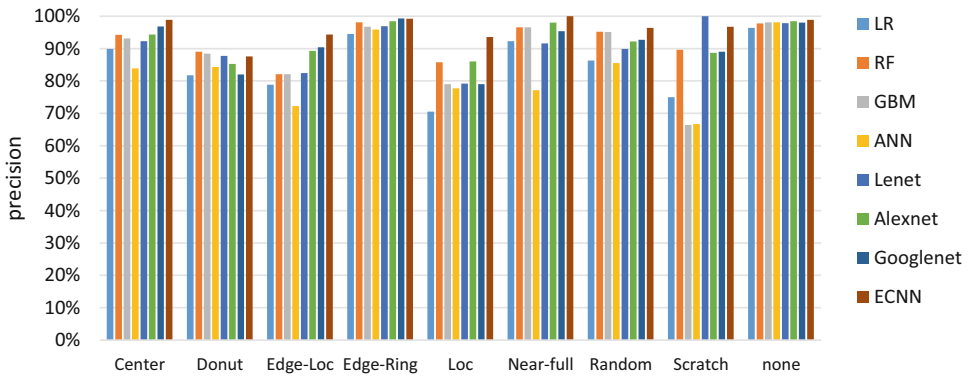

Fig. 11 Performance comparison of precision for LR, RF, GBM, ANN, LeNet, AlexNet, GoogleNet

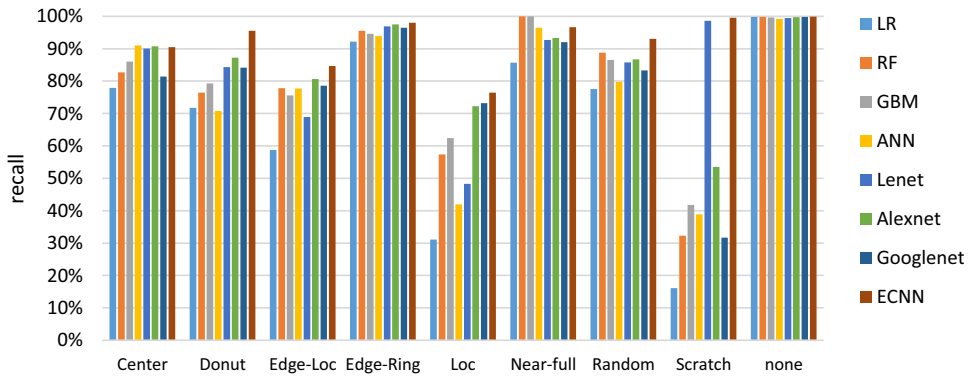

Fig. 12 Performance comparison of recall for LR, RF, GBM, ANN, LeNet, AlexNet, GoogleNet

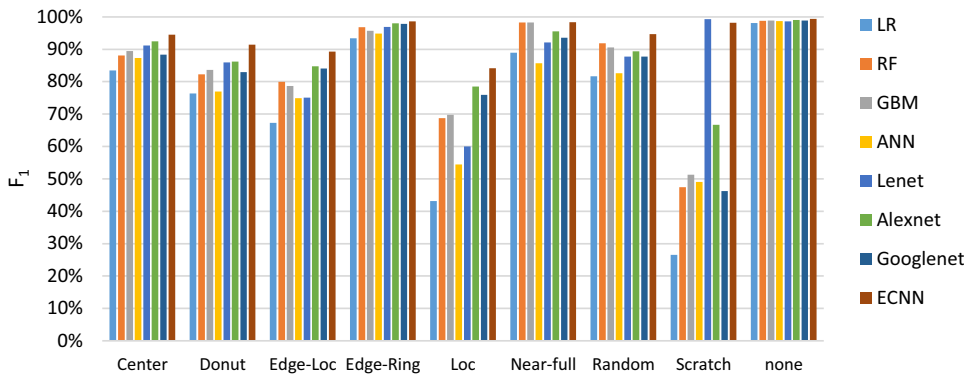

Fig. 13 Performance comparison of F_1 for LR, RF, GBM, ANN, LeNet, AlexNet, GoogleNet

which is an overall measure, in which the selected LeNet, AlexNet, and GoogleNet models have better performance than the LR, RF, GBM, and ANN models in patterns *Center*, *Donut*, *Edge-Loc*, *Edige-Ring*, *Loc*, *Scratch*, and *None*. RF and GBM models are slightly better than LeNet, AlexNet, and GoogleNet. According to the results of analysis, there is no one base classifier that completely outperforms other classifiers on all WBM patterns. Improvement can be achieved by the proposed weighted majority function which considers the contribution to classification performance of each base classifier in different WBM patterns.

The performance of the ECNN is also compared with the performance of two ensemble models, namely MVE and SVE. Table 4 presents a performance measure of precision, recall, F_1 of these ensemble classifiers for the nine WBM defect classes. The best results for each WBM defect type are shown in bold. The proposed ECNN is superior to both MVE and SVE in classifying all WBM defect types including patterns *Center*, *Donut*, *Edge-Loc*, *Edge-Ring*, *Loc*, *Near-full*, *Random*, *Scratch*, *None* with F_1 value of 94.47%, 91.38%, 89.24%, 98.60%, 84.14%, 98.31%, 94.71%, 98.14%, and 99.37%, respectively. The proposed ECNN has also highest value in terms of precision except for pattern *Donut* and the highest value in terms of recall except for pattern *Near-full*. The reason the ECNN performs well is that the weights are assigned to the base CNN models which have high recall in validation dataset. This can decrease the impact of misclassification by majority voting. The classification performance of MVE and SVE are poor for patterns *Loc* and *Scratch,* where they have low recall values as a result of the even

Table 4 Performance comparison of different ensemble classification models

Method	Defect type	Precision (%)	Recall (%)	F_1 (%)
MVE	Center	89.79	87.20	88.47
	Donut	87.37	78.30	82.59
	Edge-Loc	78.75	79.72	79.23
	Edge-Ring	97.56	94.29	95.90
	Loc	85.65	51.73	64.50
	Near-full	96.55	100.00	98.25
	Random	96.15	84.27	89.82
	Scratch	79.41	33.47	47.09
	None	97.95	99.68	98.81
SVE	Center	92.54	87.31	89.85
	Donut	**91.49**	81.13	86.00
	Edge-Loc	81.80	78.02	79.86
	Edge-Ring	97.94	94.71	96.30
	Loc	83.91	55.78	67.01
	Near-full	93.33	**100.00**	96.55
	Random	95.78	89.33	92.44
	Scratch	81.36	39.67	53.33
	None	97.93	99.72	98.82
ECNN	Center	**98.85**	**90.45**	**94.47**
	Donut	87.60	**95.50**	**91.38**
	Edge-Loc	**94.31**	**84.68**	**89.24**
	Edge-Ring	**99.22**	**97.99**	**98.60**
	Loc	**93.53**	**76.46**	**84.14**
	Near-full	**100.00**	96.67	**98.31**
	Random	**96.41**	**93.06**	**94.71**
	Scratch	**96.73**	**99.58**	**98.14**
	None	**98.83**	**99.91**	**99.37**

distribution of noise in the WBM (Saqlain et al. 2019). However, the recall for pattern *Loc* by ECNN is 76.46%, which is 47.8% and 37.1% and better than that of MVE (51.73%) and SVE (55.78%), respectively. The recall for pattern *Scratch* by ECNN is 99.58%, which is 197.5% and 151% better than that of MVE (33.47%) and SVE (39.67%), respectively. The diversity of base classifiers is important for ensemble models if they are to capture good classification performance. For example, the classification performances for pattern *Scratch* by AlexNet and GoogleNet are both poor. According to the weighted majority function, the main weight to classify pattern *Scratch* depends on LeNet.

To investigate the misclassifications of WBM pattern in testing dataset, several WBMs were selected for illustration as shown in Fig. 14. Domain experts in wafer fabrication were consulted, and they stated that a major difficulty arose when a pattern was located close to the boundary between two WBM defect patterns. For example, the ECNN predicts the class of #01, #02, and #03 WBMs in Fig. 14 are *Center* because amounts of defect occurring in the central area of the wafer. Similarly, WBMs #04-#09, #11-#13, #15-#16, #18-#19, and #22 in Fig. 14 are ambiguous in terms of their defect classes. For example, WBMs # 16 and # 22 are labelled *None* because of a slight random noise on the wafer. The ECNN identifies these two WBMs as patterns *Loc* and *Edge-Loc* because of the bulk defect on the wafer, and the domain experts accept these results as reasonable. Moreover, the #14, #15, and # 17 WBMs in Fig. 14 seem consist of two defect types. For example, #17 WBM has both the patterns *Center* and *Edge-Loc* together. Some of the original labels of the WBMs should be corrected, such as #10, #20-#24 WBMs in Fig. 14. For example, WBM #10 is a pattern *Scratch* rather than a pattern *Loc* and WBM #23 is a pattern *Edge-Loc* rather than a pattern *Loc*.

Conclusion

The study proposes an ECNN framework for WBM defect classification based on a weighted majority for three base CNN models. The ECNN is a practical and effective method for WBM defect pattern classification. It provides an end-to-end model to extract the effective features from WBM images automatically, without predefined features or a manually set threshold, and as such it represents a practical and theoretical improvement on other models reported in the literature. In particular, a weighted majority function for each base CNN model was designed on the basis of the recognition performance for each WBM defect pattern. The experimental results based on an industrial WBM case (WM-811K dataset) demonstrates that the proposed ECNN is not only effective in recognizing WBM defect patterns with high accuracy (98.57%), but that is is also robust in the face of class imbalance. The proposed ECNN also has superior performance in terms of precision, recall, F_1 when compared with other conventional machine learning classifiers such as LR, RF, GBM, ANN and ensemble classifiers such as MVE and SVE. As the diversity of WBM failure patterns is increasing in real settings, the merits of the ECNN over other methods is even more important.

Future research in the area of WBM classification should investigate the trade-off between model performance and the cost of labeling different patterns. Data-driven models are sensitive to the label of the WBM image. According to the empirical results from the WM-811K dataset, label uncertainty decreases the WBM classification accuracy. The correctness of the annotated label is essential for high accuracy when using CNN-based models (Jin et al. 2020; Park et al. 2020; Shim et al. 2020). In order to enhance the performance of the ECNN classification model, the annotation should be as correct and consistent as possible for patterns *Edge-Loc* and *Loc*. In addition to WBM pattern classification,

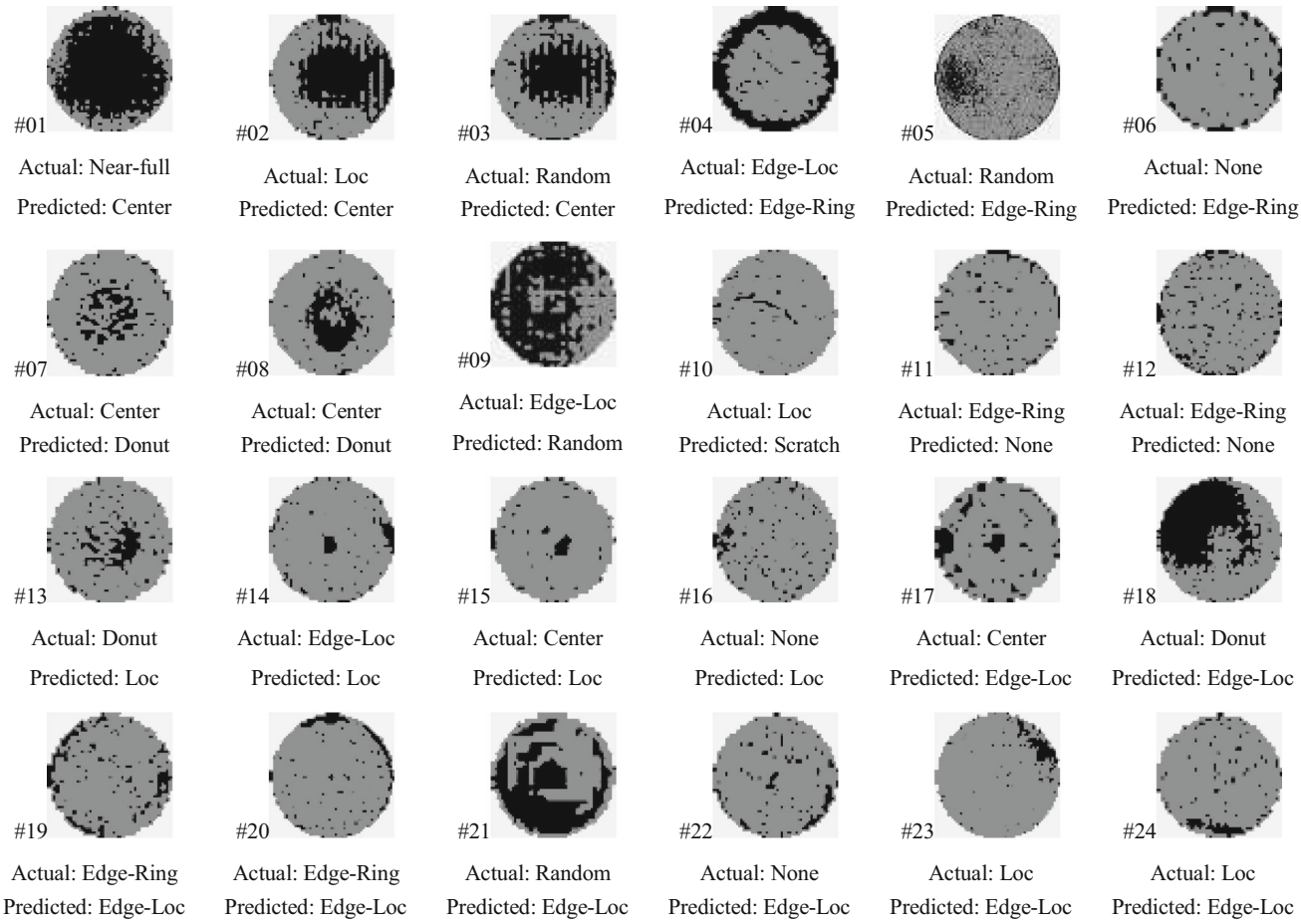

Fig. 14 WBMs classification results with actual and predicted labels

the causes of different WBM patterns should be analyzed and built into a correlated model for quickly removing abnormality and failure. Future research could further examine the robustness of the proposed ECNN in various classification systems, such as fault detection and classification in equipment monitoring.

Acknowledgement This research was supported by the Ministry of Science and Technology, Taiwan (MOST106-2628-E-027-002-MY3; MOST108-2813-C-027-017-E; MOST 108-2745-8-027-003).

References

Badmos, O., Kopp, A., Bernthaler, T., & Schneider, G. (2020). Image-based defect detection in lithium-ion battery electrode using convolutional neural networks. *Journal of Intelligent Manufacturing, 31*(4), 885–897.

Baly, R., & Hajj, H. (2012). Wafer classification using support vector machines. *IEEE Transactions on Semiconductor Manufacturing, 25*(3), 373–383.

Chen, H., Pang, Y., Hu, Q., & Liu, K. (2020). Solar cell surface defect inspection based on multispectral convolutional neural network. *Journal of Intelligent Manufacturing, 31*(2), 453–468.

Chien, C. F., Hsu, C. Y., & Chang, K. H. (2013a). Overall wafer effectiveness (OWE): A novel industry standard for semiconductor ecosystem as a whole. *Computers & Industrial Engineering, 65*(1), 117–127.

Chien, C. F., Hsu, S. C., & Chen, Y. J. (2013b). A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence. *International Journal of Production Research, 51*(8), 2324–2338.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research, 12,* 2121–2159.

Fan, M., Wang, Q., & van der Waal, B. (2016). Wafer defect patterns recognition based on OPTICS and multi-label classification. In *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (pp. 912–915).

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(4), 463–484.

Gonzalez-Val, C., Pallas, A., Panadeiro, V., & Rodriguez, A. (2020). A convolutional approach to quality monitoring for laser manufacturing. *Journal of Intelligent Manufacturing, 31*(3), 789–795.

Hsu, C. Y. (2014). Integrated data envelopment analysis and neural network model for forecasting performance of wafer fabrication operations. *Journal of Intelligent Manufacturing, 25*(5), 945–960.

Hsu, C. Y. (2015). Clustering ensemble for identifying defective wafer bin map in semiconductor manufacturing. *Mathematical Problems in Engineering*, Article no. 707358.

Hsu, C.-Y., Chen, W. J., & Chien, J. C. (2020). Similarity matching of wafer bin maps for manufacturing intelligence to empower industry 3.5 for semiconductor manufacturing. *Computers & Industrial Engineering, 142,* 106358.

Hsu, S. C., & Chien, C. F. (2007). Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. *International Journal of Production Economics, 107*(1), 88–103.

Hwang, J. Y., & Kuo, W. (2007). Model-based clustering for integrated circuit yield enhancement. *European Journal of Operational Research, 178*(1), 143–153.

Jeong, Y. S., Kim, S. J., & Jeong, M. K. (2008). Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping. *IEEE Transactions on Semiconductor Manufacturing, 21*(4), 625–637.

Jin, C. H., Kim, H.-J., Piao, Y., Li, M., & Piao, M. (2020). Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes. *Journal of Intelligent Manufacturing*, 1–15. https://doi.org/10.1007/s10845-020-01540-x.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS) Conference* (pp. 1097–1105).

Kyeong, K., & Kim, H. (2018). Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. *IEEE Transactions on Semiconductor Manufacturing, 31*(3), 395–402.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

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, E., Chen, K., Xiang, Z., & Zhang, J. (2020). Conductive particle detection via deep learning for ACF bonding in TFT-LCD manufacturing. *Journal of Intelligent Manufacturing, 31*(4), 1037–1049.

Liu, C. W., & Chien, C. F. (2013). An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing. *Engineering Applications of Artificial Intelligence, 26*(5–6), 1479–1486.

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.

Nakazawa, T., & Kulkarni, D. V. (2019). Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 32*(2), 250–256.

Park, S., Jang, J., & Kim, C. O. (2020). Discriminative feature learning and cluster-based defect label reconstruction for reducing uncertainty in wafer bin map labels. *Journal of Intelligent Manufacturing*, 1–13. https://doi.org/10.1007/s10845-020-01571-4.

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.

Saha, S., & Ekbal, A. (2013). Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering, 85,* 15–39.

Saqlain, M., Jargalsaikhan, B., & Lee, J. Y. (2019). A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 32*(2), 171–182.

Shim, J., Kang, S., & Cho, S. (2020). Active learning of convolutional neural network for cost-effective wafer map pattern classification. *IEEE Transactions on Semiconductor Manufacturing, 33*(2), 258–266.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015), Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-RMSProp, COURSERA: Neural networks for machine learning. Technical report.

Wang, R., & Chen, N. (2019). Wafer map defect pattern recognition using rotation-invariant features. *IEEE Transactions on Semiconductor Manufacturing, 32*(4), 596–604.

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.

Yu, J. (2019). Enhanced stacked denoising autoencoder-based feature learning for recognition of wafer map defects. *IEEE Transactions on Semiconductor Manufacturing, 32*(4), 613–624.

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.

Yu, N., Xu, Q., & Wang, H. (2019a). Wafer defect pattern recognition and analysis based on convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing, 32*(4), 566–573.

Yu, J., Zheng, X., & Liu, J. (2019b). Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map. *Computers in Industry, 109,* 121–133.

Yuan, T., & Kuo, W. (2008a). A model-based clustering approach to the recognition of the spatial defect patterns produced during semiconductor fabrication. *IIE Transactions, 40*(2), 93–101.

Yuan, T., & Kuo, W. (2008b). Spatial defect pattern recognition on semiconductor wafers using model-based clustering and Bayesian inference. *European Journal of Operational Research, 190*(1), 228–240.

Yuan, T., Kuo, W., & Bae, S. J. (2011). Detection of spatial defect patterns generated in semiconductor fabrication processes. *IEEE Transactions on Semiconductor Manufacturing, 24*(3), 392–403.

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. arXiv:1212.5701.