Received: 27 July 2021 | Revised: 3 August 2021 | Accepted: 9 August 2021

DOI: 10.1002/qre.2983

RESEARCH ARTICLE

A Deep convolutional neural network with residual blocks for wafer map defect pattern recognition

Fu-Kwun Wang | **Jia-Hong Chou** | **Zemenu Endalamaw Amogne**

Department of Industrial Management, National Taiwan University of Science and Technology, Taipei City, Taiwan

Correspondence
Fu-Kwun Wang, Department of Industrial Management, National Taiwan University of Science and Technology, Taipei City 106335, Taiwan.
Email: fukwun@mail.ntust.edu.tw

Funding information
Ministry of Science and Technology in Taiwan, Grant/Award Number: MOST-107-2221-E011-100-MY3

Abstract

Different deep convolution neural network (DCNN) models have been proposed for wafer map pattern identification and classification tasks in previous studies. However, factors such as input image resolution effect on the classification performance of the proposed models and class imbalance in the training set after splitting the data into training and test sets have not been considered in the previous studies. We propose a DCNN model with residual blocks, called the Opt-ResDCNN model, for wafer map defect pattern classification by considering 26 × 26, 64 × 64, 96 × 96, and 256 × 256 input images and class imbalance issues. The model with a balance function can improve the performance. We compare the proposed model with the published defect pattern classification models in terms of accuracy, precision, recall, and F1 value. Using a publicly available wafer map data set (WM-811K), the proposed method on the four different resolutions can obtain an excellent average accuracy, precision, recall, and F1 score. Regarding accuracy, the proposed model results are 99.90%, 99.86%, 90.28%, 98.88%, respectively. These results are better than the published papers.

KEYWORDS

class imbalance, deep convolutional neural network, defect pattern recognition, residual blocks, wafer map

1 | INTRODUCTION

Currently, the demand for electrical products is increasing, and the quality, reliability, and capability of wafer production becomes a significant issue. For example, Taiwan Semiconductor Manufacturing Company (TSMC) mass-produced 5 nm wafers in 2020 and plans to produce 3 nm wafers in 2022.[1] Integrated circuits (IC) manufacturing requires high investment, precise technology, and a complex manufacturing process. It is essential to detect defects in the wafer pattern to improve the yield, quality, and reliability of the IC manufacturing process. The wafer manufacturing process contains hundreds of workstations, which makes it challenging to solve defect detection problems. Even though high-precision automated equipment and sensors are installed and detailed analysis methods are used, we cannot completely solve defective wafer production.[2] Wafer defects can be divided into two categories: global defects and local defects.[3] Global defects are caused by unexpected causes, such as molecules in a clean production room. These types of defects are expensive and difficult to solve. Local defects are caused by human error, equipment failure, and chemical problems.[3] Generally, local patterns' spatial patterns include linear, non-linear curves, and random densities, such as scratch, ring, edge-location, location, and center.[4] These spatial patterns provide very useful information that can help automate the defect detection process.

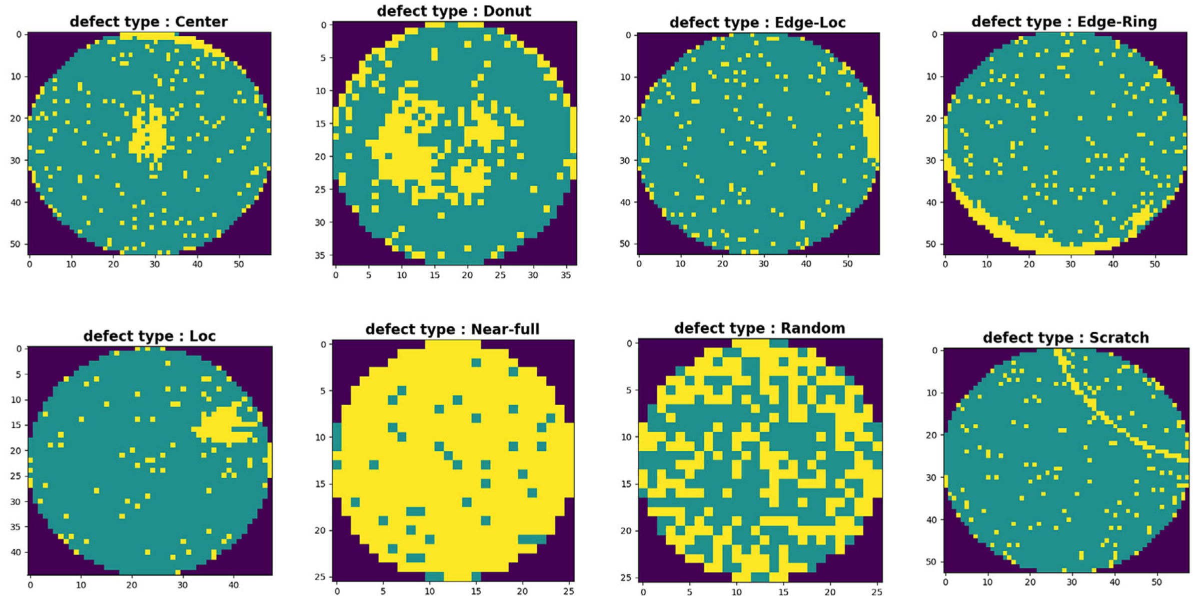

FIGURE 1 The eight wafer defect patterns

The wafer map plays a vital role in the semiconductor manufacturing industry. After the engineer inspects the unique spatial curve or feature through the visual inspection tool, the defect can be recognized and determined during the manufacturing process.[5] Therefore; it is essential to have an engineer with excellent defect pattern recognition ability. In order to establish an analysis system framework to identify the root cause of failures, manufacturing engineers classified the failure modes and divided them into groups of similar modes.[6] Therefore, effective wafer analysis tools are essential for semiconductor manufacturing companies to improve their competitiveness.[7] Local defects can be identified by converting digital image data and generating feature analysis methods. Some studies have used different feature extraction methods to identify local defects.[7–11]

In recent years, to increase the level of automation engineering, sensors have been installed to collect and analyze data through various machine learning (ML) and deep learning (DL) algorithms. The algorithm has been applied to identify defect patterns in wafer maps. Although many studies have researched wafer map defect pattern identification (WMDPI), most of the research focuses on the feature extraction of input data rather than recognizing and analyzing image-based wafer maps as input.

A useful and powerful model structure called convolutional neural network (CNN) has been proven successful in several studies for image classification problems. The goal of the convolutional layer is to learn and extract features from the input image data. Each convolutional layer contains several kernel filters, which are used to generate new feature maps. The most critical component of CNN is the pooling layer, which can reduce the feature maps' resolution to achieve shift-invariance.[12] As the convolutional layer's depth increases, the vanish/exploding gradient problem will become a problem that needs to be solved. In 2015, two learning structures called residual block,[13] and dense block[14] are used to solve the vanishing gradient problem without complex computation. This study proposes a deep CNN (DCNN) model with residual blocks called Opt-ResDCNN model. The DNN model is improved by stacking more convolutional layers, residual blocks, and optimal dropout layers. The proposed WMDPI and classification model has never been used before to the best of our knowledge. The model was compared with previously published research, and its performance was verified with image data of different resolutions using a real wafer map data set called WM-811K.

The WM-811K data set is the largest wafer map data set containing 811,457 real-world wafer maps collected from actual assembling lines. The data set involves eight different types of defect patterns (center, donut, edge-local, edge-ring, local, random, scratch, and near-full) and non-failure patterns. Domain experts label these defect types.[7] Figure 1 depicts the original WM-811K data set with failure pattern types, and the distribution of failure pattern types is shown in Figure 2. As shown in Figure 2, the WM-811K data set has an imbalance between these eight defect patterns. For example, the highest and lowest percentage distributions of these eight defect types are local and donut failure types and the percentage difference is about 0.37%. Data imbalance is a crucial problem, and it needs to be trained to train a good DL model for

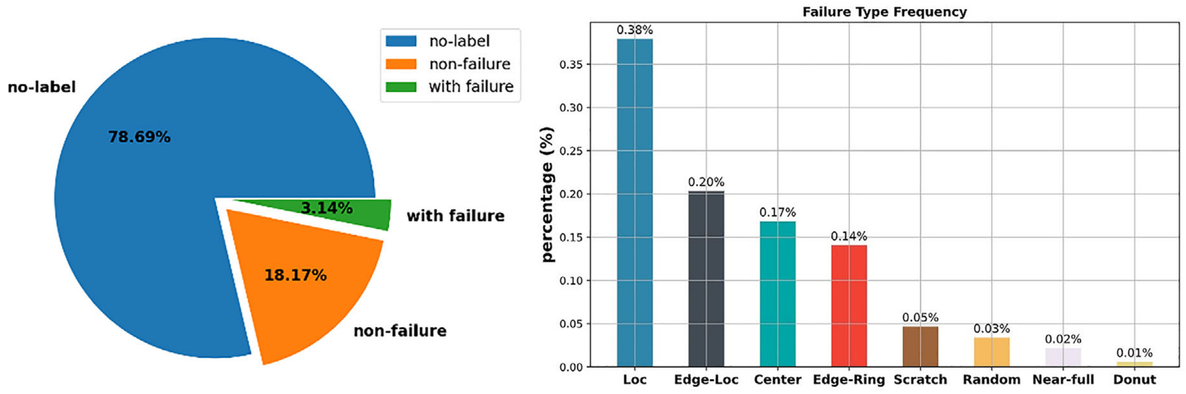

FIGURE 2 Wafer map distribution in the WM-811K dataset

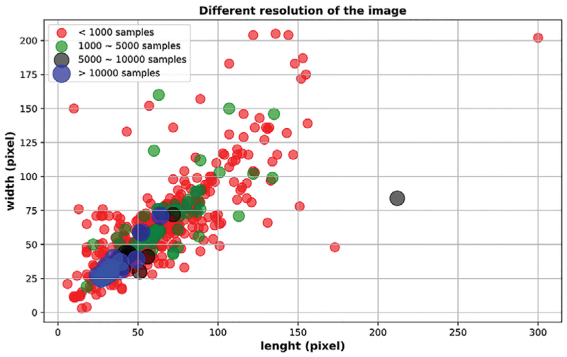

FIGURE 3 Image resolution distribution in the WM-811K dataset

pattern recognition of wafer maps. Therefore, proper data balancing techniques are required to achieve good classification performance.

The effect of different input image resolutions on wafer map defect pattern recognition and classification performance is discussed in this study. Figure 3 shows the image resolution distribution in the WM-811K data set. Large samples are concentrated inside a resolution length of 25–64 pixels on the x-axis and a resolution width of 25–71 pixels on the y-axis.

The proposed method has the following three contributions:

1. Using the same analysis process, the proposed Opt-ResDCNN model can fit and obtain excellent results for different resolutions.
2. The impact of different input image resolutions on the classification performance of a model is investigated.
3. A class balance function that impacts the model's classification performance is introduced for the train-test set split.

The rest of the article is organized as follows. Section II introduced related work on the WM-811K data set. The construction process of the proposed model is discussed in Section III. The experimental results of different resolutions and the comparison results of other methods are presented in Section IV. Finally, Section V provides conclusions.

2 | RELATED WORK

The first CNN structure was created by Lecun et al.[15] in 1980 for image classification purposes. With the advancement in computation device technologies, image classification problems have become easy to solve these days. Wu et al.[7] created the WM-811K data set, including 811,467 real-world wafer maps. Later many studies have proposed many defect identification methods for this data set.

TABLE 1 Related works on the WM-811K dataset

Reference	Training samples	Test samples	Resolution size	Channel	Model	Overall accuracy
Shawon et al.[16]	12730	705	[26, 26, 3]	RBG	DCNN	99.29%
Yu and Liu.[33]	13451	4483	[96, 96, 1]	Grayscale	PCACAE	97.27%
Ji and Lee[34]	8160	1000	[64, 64, 1]	Grayscale	GAN-CNN	98.30%
Jin et al.[35]	18000	2000	[256, 256, 1]	Grayscale	CNN-ECOC-SVM	98.43%
Tsai and Lee[36]	103770	51885	[64, 64, 3]	RBG	DMC1	97.01%
Maksim et al.[37]	11760	9471	[96, 96, 1]	Grayscale	VGG-19	84.81%
	11760	9471	[96, 96, 1]	Grayscale	ResNet-34	81.91%
	11760	9471	[96, 96, 1]	Grayscale	ResNet-50	87.84%
	11760	9471	[96, 96, 1]	Grayscale	MobileNetV2	85.39%
Shen et al.[40]	7112	2000	[224, 224, 1]	Grayscale	T-DenseNet	87.70%

Yu et al.[8] built a manifold learning-based wafer map defect detection system that uses an unsupervised learning method called joint local and nonlocal linear discriminant analysis (JLNDA). Fan et al.[9] proposed a combination method based on an ordering point to identify the cluster structure (OPTICS) clustering with a support vector machine (SVM) classifier that recognizes more than one failure on a single wafer map with an accuracy of 94.3%. Piao et al.[10] presented the decision tree ensemble-based method. Saqlain et al.[11] used a soft voting ensemble (SVE) method that combines the best results of four ML classifiers, logistic regression (LR), random forests (RF), gradient boosting machine (GBM), and artificial neural network (ANN).

The CNN model has recently been used to identify defect patterns in wafer maps. The CNNs have an excellent ability for learning image data. CNNs use lower parameters and are less time-consuming than the conventional fully connecting network compared with traditional neural networks. CNNs can overcome previously mentioned problems and can be used in automated assembly lines to identify wafer map defects quickly. As CNN depth becomes increasing, the problems of vanishing and exploding gradient become challenging to solve. He et al.[13] proposed the residual neural network (ResNet) that combines with underlying mapping and identity mapping to improve the accuracy from significantly increased depth without any new computation algorithms. Huang et al.[14] proposed the dense convolutional network (DensNet) in which the dense block was modified from the residual block, and the computation was changed from summation to concatenation.

Shawon et al.[16] proposed a DCNN architecture with a data augmentation technique to solve the data balance problem and achieve high accuracy. Yu et al.[17] presented a hybrid DL model called stacked convolutional sparse denoising autoencoder (SCSDAE).

Regarding the defect class imbalance in the wafer map, different data balancing techniques can generate additional images for the minority failure types to balance the data set and reduce the image in majority failure types. There are three methods for data balancing,[18] which are called data-level methods,[19–22] algorithm-level methods,[23–28] and hybrid approaches.[29–33] Ji and Lee[34] proposed generative adversarial networks (GAN) to improve the CNN-based classifier performance.

Jin et al.[35] proposed a framework to extract the features from high-resolution wafer maps pattern by CNN and fed the extracted features to a combination of error-correcting output codes (ECOC) and SVM. Tsai and Lee[36] proposed an efficient light-weight deep convolution model called defect map classification (DMC) network to reduce computation time and use fewer parameters. Maksim et al.[37] tried to improve pattern recognition performance in a few amounts of experimental data conditions. They got the pre-trained weight from only synthesized data. They implemented it on four previously proposed models, VGG-19,[38] MobileNetV2,[39] ResNet-34, and ResNet-50.[13] A model based on the integration of transfer learning and DensNet called T-DensNet is used for wafer map defect-recognition.[40]

Since deep CNNs stacked more convolutional layers, their performance becomes better as the layers' number becomes deeper. However, the classification performance of the proposed models based on those ResNet and DensNet on the WM-811K data set is somehow poor since the authors used the whole model structure directly without changing its construction for this specific dataset. Some of the previously published wafer-map defect pattern recognition models on the WM-811K data set are summarized in Table 1.

In this study, a CNN structure model with residual blocks, called Opt-ResDCNN, is created to increase the model's depth, and the dropout layer is also added to solve the overfitting problem. Convolutional autoencoder (CAE) is used for data augmentation purposes, while over-sampling and under-sampling data-level methods are used to deal with the class imbalance issue. The proposed model's performance is verified using different input image resolutions by following the same analysis procedure for different input image resolutions. The proposed model's efficiency is compared with the previously published papers.

3 | PROPOSED MODEL

We propose a model based on a deep CNN with residual blocks, called Opt-RseDCNN, for defect recognition and classification of the unbalanced WM-811K data set. The original WM-811K data set is preprocessed to identify labeled and unlabeled patterns, and then the class distribution of the labeled patterns is checked. The image size of the original data set is adjusted to the required resolution to see the impact of different image resolutions on the model's classification performance. The one-hot encoding method is used to change one-channel (grayscale) original image data into a three-channel (RGB) format. CAE is used to balance the labeled patterns of the original data set. The balanced dataset is split into train-test set with the same training and test samples as previous studies to make a fair comparison.

After the data set is divided into training and test sets, the training set is balanced by a simple balancing function to improve the model's classification performance. To ensure the consistent performance of the proposed model, the model will run ten iterations. The average of the performance index results of these 10 iterations is used to measure the model's performance. The flow chart of the proposed model is shown in Figure 4.

In Figure 2 above, we can see that 78.69% (638507 samples) of the raw WM-811K data set is an unlabeled one. 21.31% (172950 samples) of the data set is used to train the proposed DL model by removing the unlabeled data set. After that, the one-hot-encoding approach is utilized to generate an RGB image channel. The benefit of increasing the channel number is to extract more features from the RBG image by using a 3D filter in the convolutional layer.

The demonstration of the image one-hot encoding approach is shown in Equation (1).

$$Ch_i = \begin{cases} if\ Grayscale\ = i-1,\ ouput = 1 \\ \quad else\ output\ = 0 \end{cases},\ i = 1,\ 2, 3\ 3 \tag{1}$$

where Ch_i is the value of the RGB channels: red, green, and blue are represented as the 1st, 2nd, and 3rd channels, respectively. If the value of a grayscale image is 0, the output of the first channel will be 1.

3.1 | Data balancing with CAE

The dataset description in the introduction section shows that the WM-811K dataset has a data imbalance problem. Many models have been proposed to deal with data imbalance problems.[16,17,31–34] The CAE model is used for data generation purposes to solve the data imbalance issue in this study. The CAE model is an unsupervised DL model.[31] The CAE model consists of two parts: encoder and decoder, and the objective of the CAE model is to reconstruct the input data during the forward and backpropagation calculations. The encoder structure tries to extract the useful features from the input data. The decoder structure tries to reconstruct the input image by valuable feature extraction from the encoder output. Figure 5 shows the general CAE model structure with five convolutional layers.

In Figure 5, each convolutional layer in the encoder part is connected to a pooling layer. The pooling layer is utilized to extract features from the convolutional layer's output feature maps.[41] In the decoder part, the transposed convolutional layer is used for up-sampling, and the transposed convolution process can learn the optimum up-sampling during model training. The CAE model is used for images of different resolutions considered in this study, resulting in a high reconstruction rate. The structure of the proposed CAE model is shown in Table 2.

The encoder extracts the feature from the input x to the next hidden layer. The output feature maps of the *l-th* hidden layer in the encoder (H^l) is given by (2).

$$H^l = \sigma\left(x * W^l + b^l\right),\ l\ =\ 1,\ 2,\ \dots\dots,\ n \tag{2}$$

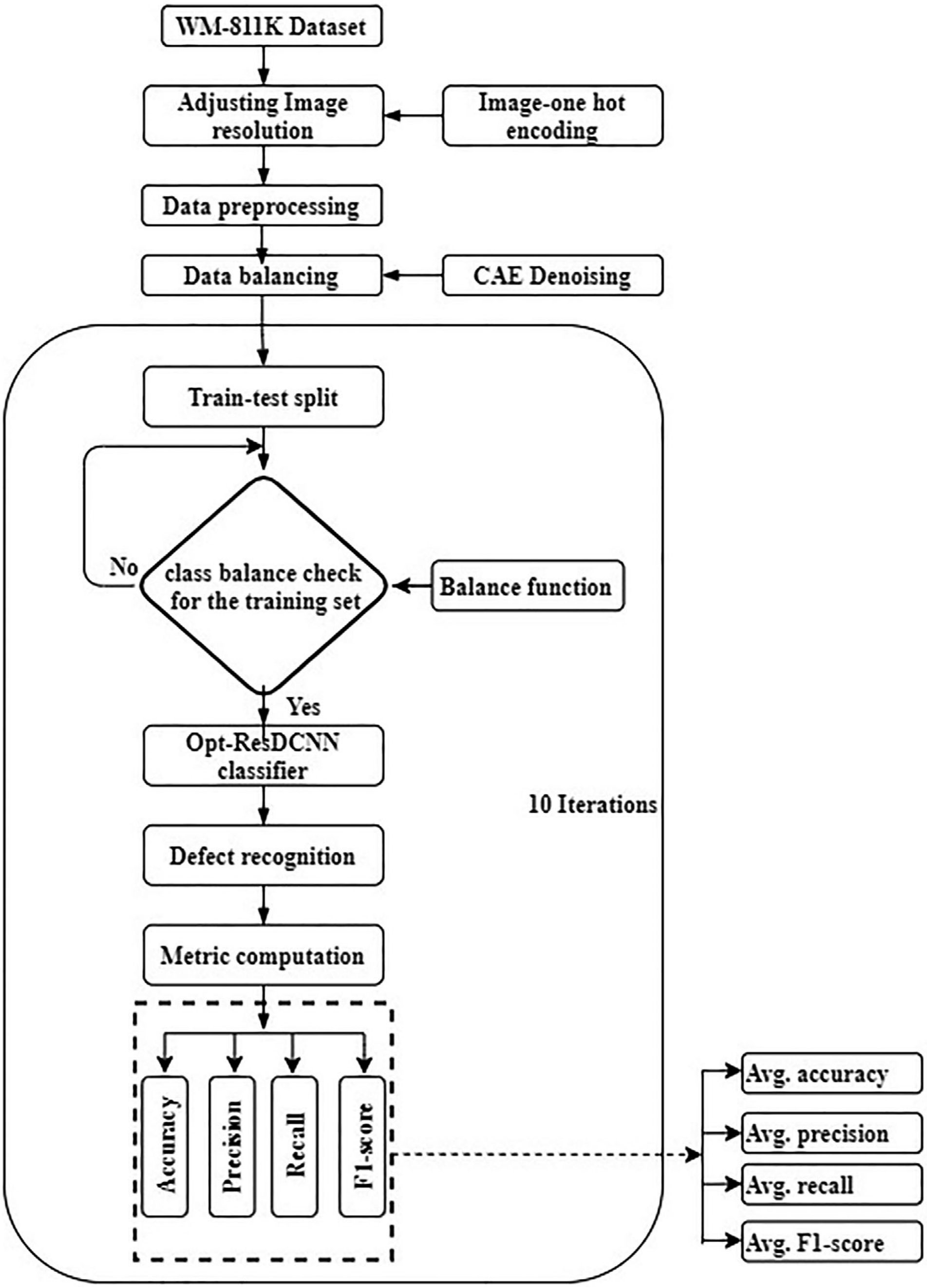

FIGURE 4 Flowchart for the proposed model

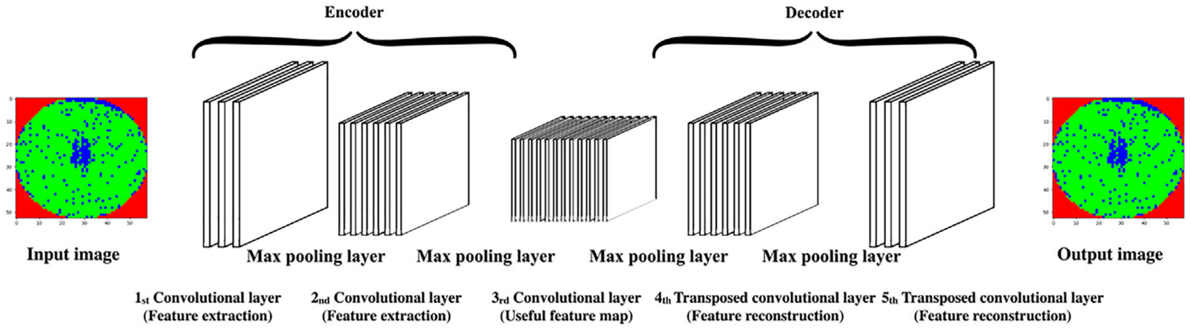
Encoder
Decoder
Input image
Max pooling layer
Max pooling layer
Max pooling layer
Max pooling layer
Output image
1st Convolutional layer (Feature extraction)
2nd Convolutional layer (Feature extraction)
3rd Convolutional layer (Useful feature map)
4th Transposed convolutional layer (Feature reconstruction)
5th Transposed convolutional layer (Feature reconstruction)

FIGURE 5 The general Convolutional autoencoder (CAE) model structure

TABLE 2 Details of different layers of the CAE model

Layer	Layer type	Kernel weight size	Stride	Padding	Activation function
1	Conv	(3, 3)	1	(1, 1)	ReLU activation
2	Max pool	(2, 2)	1	–	–
3	Conv	(3, 3)	1	(1, 1)	ReLU activation
4	Max pool	(2, 2)	1	–	–
5	Transposed Conv	(3, 3)	2	–	ReLU activation
6	Transposed Conv	(2, 2)	2	–	Sigmoid activation

Note: Conv = convolutional layer; Max pool = max pooling layer; Transposed Conv = Transposed Convolutional layers.

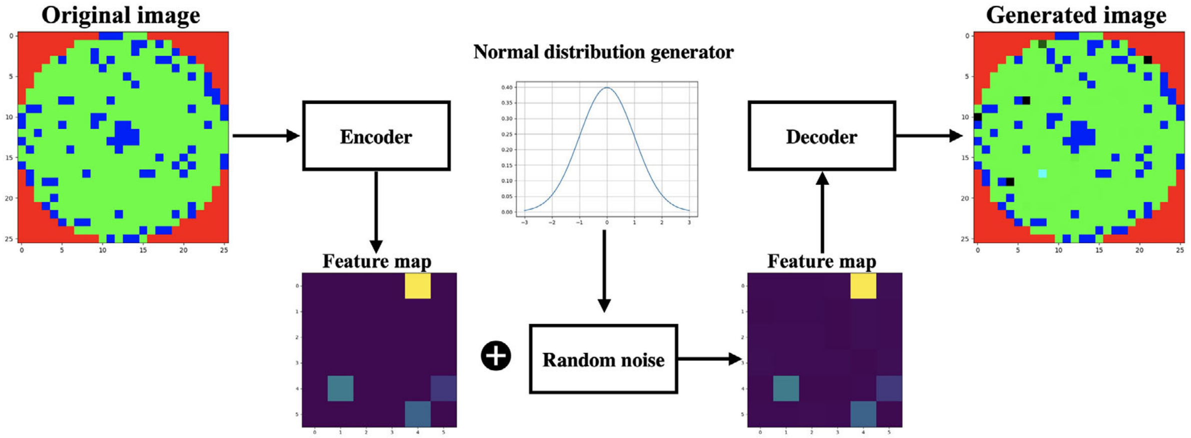

FIGURE 6 The data balancing process for 26 × 26 input image resolution

where σ is the activation function, which is the ReLU activation function, n is the number of hidden layers in the encoder, and $*$ represents the convolution operator. W^l and b^l are defined as the convolution kernel weight and bias of hidden layer l, respectively.

The decoder tries to reconstruct the input x by the last hidden layer's feature maps, and the transpose convolution process is used in the decoder to reduce the feature maps. The output is the restored feature maps of the *k-th* layer (TrH^k) given by (3).

$$TrH^k = \sigma\left(H^n * \hat{W}^k + b^k\right),\ k = 1,\ 2,\ \dots\dots,\ m \tag{3}$$

where σ is the activation function. In the decoder structure, the ReLU activation function and Sigmoid activation are conduct in the hidden layers and output layer, respectively, H^n is the output of the *n-th* hidden layer in the encoder, m is the number of hidden layers in the decoder, and $*$ in the decoder represents the transposed convolution operator. $\hat{W}^k$ and b^k are defined as the transposed convolution kernel weight and bias of hidden layer k, respectively.

To minimize the difference between the input x and the output of the last decoder layer (TrH^m) in training the CAE model, the MSE is used as the loss function. The MSE loss function is defined in Equation (4), where the training batch size (s)is the samples in the mini-batch.

$$MSE_{Loss} = \frac{\sum_{i=0}^{n}\left(TrH_i{}^n - x_i\right)^2}{s},\ n = 0,\ 1, \dots, s \tag{4}$$

For the data balancing process, the trained CAE model is divided into encoder and decoder. The encoder is used for reducing the feature dimensionality from the input wafer maps pattern. Some noise values generated from the random normal distribution generator will be added to the output feature map from the encoder. After adding noise to the model, the decoder tries to reconstruct the input images by removing noise. The CAE model will try to generate output images with the same defect pattern as the input image through the data balancing process. Figure 6. shows the data generation process by using the encoder and decoder for a 26 × 26 input image resolution. This study utilizes a CAE model for data

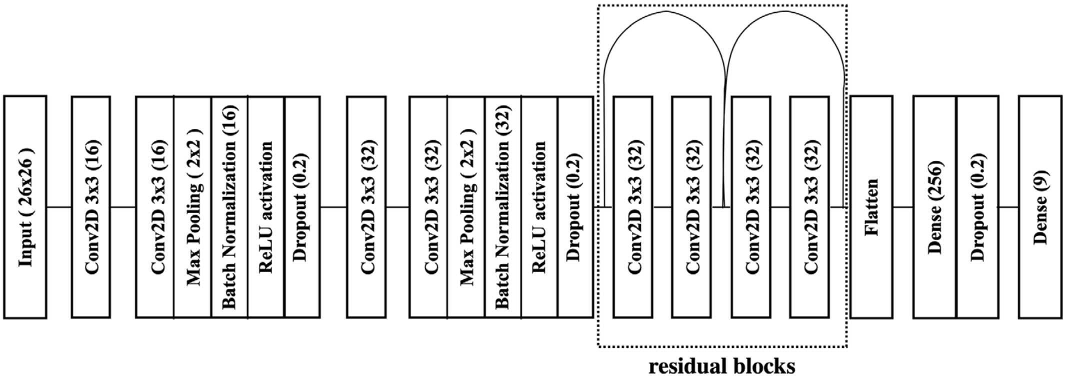

FIGURE 7 The architecture of the Opt-ResDCNN model

generation purpose only. Over-sampling and under-sampling data-level methods are used to deal with the data imbalance issue.

3.2 | Class balance check for splitting the dataset

Using a balanced training set is essential to train a good DL model for the classification problem. The previous related studies did not mention anything related to this issue. From the experiments in this study, it is found that balancing the classes in the training set can provide a better classification performance. The performance comparison between balanced and imbalanced training sets will be discussed in section IV. A balance function (B_i) for each class i is used to balance the classes in the training set using the following equation.

$$B_i\ (x,\ c) = \frac{x}{c} \tag{5}$$

where x is the total number of the training samples, c is the total number of defect classes, which is nine in this case. This study proposed this simple balance function only for checking the class balance during training and test set split. Using these balance function, we can make sure that each class has an equal amount of training samples for better performance of the proposed model.

3.3 | Proposed DL classifier: Opt-ResDCNN model

The proposed model called Opt-ResDCNN is a unique model which combines the concept from the DCNN model and residual blocks. Previously, Shawon et al. [16] proposed a DCNN model with wafer map pattern recognition accuracy of 99.26% for the WM-811K dataset. This study tries to improve WMDPI and classification based on the DCNN idea and add residual blocks to connect more convolutional layers to increase the neural network's depth. The Opt-ResDCNN model also includes dropout layers to solve the overfitting problem and optimize the model's performance. The architecture of the proposed model is shown in Figure 7.

The proposed model includes two residual blocks (Figure 8), a powerful structure for training a very deep DL model without any complex computation, and can solve the gradient vanishing problem.[13] It tried to learn the identity mapping residual from the additive residual function as in Equation (6), and the gradient vanishing problem can be solved by the identity mapping during backpropagation as shown in Equation (7).

$$X_{L+1} =\ h\left(X_L\right) + F\left(X_L,\ W_L\right) \tag{6}$$

$$\begin{cases} X_{L+1} =\ h\left(X_L\right) + F\left(X_L,\ W_L\right),\ forward\ pro. \\ \frac{\partial\varepsilon}{\partial X_L} = \frac{\partial\varepsilon}{\partial X_{L+1}}\ \frac{\partial X_{L+1}}{\partial X_L} = \frac{\partial\varepsilon}{\partial X_{L+1}}\ \left(1 + \frac{\partial}{\partial X_L} F\left(X_L,\ W_L\right)\right),\ backpro. \end{cases} \tag{7}$$

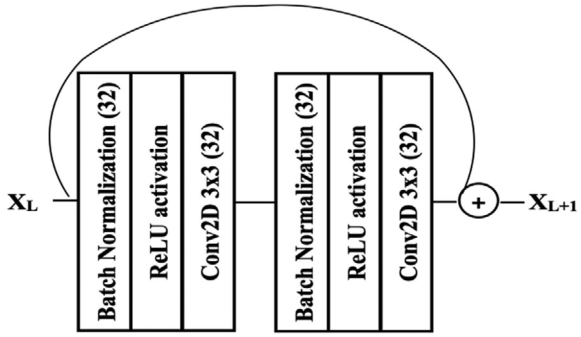

FIGURE 8 The architecture of the pre-activation residual block (X_L and X_{L+1} are input and output)

where X_L represents the input feature, and X_{L+1} is the output feature of the residual block. h is set as the identity mapping in which the input will be the same as the output. F represents a residual function that consists of ReLU activation function, batch normalization, and convolutional operation. The W_L is the convolution kernel weight. In Equation (7), the backpropagation is calculated using the chain rule, and the loss function is represented as ε. The partial differential of the identity mapping term is one. It means that the input feature information can propagate back directly to any layers to solve the gradient vanishing problem, and the residual block will try to learn residual from the residual function. Two residual blocks are connected in the proposed model before the fully connected layer of the CNN-based structure to increase the model depth and extract more features. The proposed model used the pre-activation function residual block structure to improve the Opt-ResDCNN model performance.

The 1st and 3rd layers of the proposed model utilize only convolutional layers for feature extraction. The 2nd and 4th layers use a convolution operation consisting of max pooling, batch normalization, and dropout layers. The max-pooling layer can randomly keep the maximum value from the feature map, and those maximum values represent the critical features. The batch normalization layer keeps the input and output in the same distribution and helps the model converge quickly. The dropout layer is used to deal with the overfitting problem, and it is a useful technique to increase the proposed model's classification performance. In the classification part of the proposed model, the fully connected layer is linked with the output of the last residual block to classify the class labels. The final output of the proposed model is nine different kinds of defect classes, including center, donut, edge-local, edge-ring, local, near-full, random, scratch, and none.

3.4 | Performance metrics computation

To evaluate the classification performance of the proposed model, accuracy, precision, recall, and F1 score metrics are used in this study. Those metrics are calculated from a confusion matrix. The x-axis and y-axis of the confusion matrix represent the predicted class labels and the actual class labels. The measurements are computed based on true positive (TP), true negative (TN), false positive (FP), and false negative (FN), as shown in Equation (8).

$$\begin{cases} Accuracy_{class} = \frac{TP_{class}+TN_{class}}{TP_{class}+TN_{class}+FP_{class}+FN_{class}} \\ Precision_{class} = \frac{TP_{class}}{TP_{class}+FP_{class}} \\ Recall_{class} = \frac{TP_{class}}{TP_{class}+FN\ {}_{class}} \\ F1\ score_{class} = \frac{2\times(Precision_{class}\times Recall_{class})}{Precision_{class}+Recall_{class}} \end{cases} \tag{8}$$

4 | EXPERIMENT ANALYSIS AND RESULTS

In this section, the same training and test samples or fewer training samples and same test samples are utilized to compare each previous study's proposed model's performance under the four different image resolutions, including 26 × 26, 64 × 64, 96 × 96, and 256 × 256. The proposed model initializes in each iteration to avoid the model training again, and the final

TABLE 3 Comparison of Opt-ResDCNN model performance with and without balance function

Metrics	Without balance function	With balance function
Accuracy	99.87%	**99.90%**
Precision	99.89%	**99.91%**
Recall	99.59%	**99.91%**
F1 Score	99.58%	**99.91%**

matric result is the average of 10-iterations matric (see Figure 4). The system runs on the PyChram 2020.1.5 environment with the Torch library. The computer configuration is as follows: Memory: 64 GiB; Processor: Intel Core i9-10900k and NVIDIA GeForce RTX 3090; OS: Ubuntu 20.04.

4.1 | Hyperparameters setting for CAE and Opt-ResDCNN model

The hyperparameter setting is essential for training DL models since it will affect the model performance by avoiding the overfitting and gradient vanishing problems. The tuned hyperparameters in the CAE and Opt-ResDCNN models are epoch, batch size, and learning rate. For the CAE model, the batch size and learning rate are set as 1024 and 0.01, respectively, for all four different resolution cases. However, for the proposed Opt-ResDCNN model, the learning rate of 0.01 is set for 26 × 26 and 64 × 64 input image resolutions, whereas the learning rate of 0.001 is set for 96 × 96 and 256 × 256 input image resolutions. The proposed model uses a batch size of 1024 for 26 × 26 and 64 × 64 input image resolution, whereas the batch size of 512 and 64 is used for 96 × 96 and 256 × 256 input image resolution, respectively. The learning rate and batch size are set based on the training samples' input image resolution and size. A lower learning rate and batch size is used for 96 × 96 and 256 × 256 input image resolutions by considering the device's memory limit. In addition, the dropout parameter used in all cases is 0.2.

4.2 | The effect of the balance function

The experiment uses 26 × 26 input image resolution to test the effect of the balance function while splitting the train-test set. The proposed model's training samples and test samples with and without balance function are 12,730 and 705, respectively. The measurement results in Table 3 are the average measurements of 10 repetitions. It can be seen that the accuracy, precision, recall, and F1-score of the proposed model with balance function are 99.89%, 99.91%, 99.91%, and 99.91%, respectively. These values are slightly greater than those of the proposed model without balancing. We can conclude that the model with a balance function can improve the performance of the DL classifier.

4.3 | The Opt-ResDCNN model comparison results

To show the effectiveness of the proposed model in WMDPI, we compare the proposed model with previously published results with different input image resolutions. For 26 × 26 input image resolution, the proposed model is compared with Shawon et al.[16] The train and test samples used in Shawon et al.[16] are 12,730 and 705, respectively. The proposed model uses the same training and test samples and repeats 10 times to obtain performance values. Table 4 shows the performance of the proposed model in terms of accuracy precision, recall, and F1 score. Regarding accuracy, the value of the Opt-ResDCNN model is 99.90%, which is better than the 99.29% of Shawon et al.[16] In addition, the four performance metrics of the proposed model used to identify donut, random, scratch, and near-full defect types are all 100%.

For the second resolution, 64 × 64, the proposed model is compared with the DMC1 model[36] and the GAN-CNN model.[34] The train and test samples used in Tsai and Lee[36] are 103,770 and 518,855, respectively. The train and test samples used in Ji and Lee[34] are 8160 and 1000, respectively. Here, the proposed model uses the same number of training and test samples in Tsai and Lee[36] and repeats 10 times to obtain performance values. Table 5 shows the classification results of the proposed model for each class, including accuracy, precision, recall, and F1 score. Regarding accuracy, the value of Opt-ResDCNN model is 99.90%, which is better than 97.10% of Tsai and Lee[36] and 98.30% of Ji and Lee.[34]

TABLE 4 Class level classification performance results under 26 × 26 resolution

Failure Type	Accuracy	Precision	Recall	F1 Value
Center	99.76%	99.79%	99.985%	99.884%
Donut	100.00%	100.00%	100.00%	100.00%
Edge-Loc	99.99%	99.99%	99.422%	99.699%
Edge-Ring	99.33%	99.42%	99.985%	99.699%
Loc	99.99%	99.99%	99.808%	99.897%
Random	100.00%	100.00%	100.00%	100.00%
Scratch	100.00%	100.00%	100.00%	100.00%
Near-Full	100.00%	100.00%	100.00%	100.00%
None	99.99%	99.99%	98.98%	99.985%
Average	99.90%	99.91%	99.91%	99.91%

TABLE 5 Class level classification performance results under 64 × 64 resolution

Failure Type	Accuracy	Precision	Recall	F1 Value
Center	99.12%	99.11%	98.60%	98.85%
Donut	99.99%	99.99%	99.97%	99.98%
Edge-Loc	97.23%	97.31%	94.96%	96.12%
Edge-Ring	99.62%	99.60%	99.70%	99.65%
Loc	97.71%	97.79%	96.02%	96.89%
Random	99.91%	99.91%	99.99%	99.95%
Scratch	99.86%	99.86%	99.86%	99.86%
Near-full	99.06%	99.03%	99.06%	99.04%
None	92.65%	92.25%	96.77%	94.44%
Average	98.35%	**98.32%**	**98.32%**	**98.31%**

In the third case with a resolution 96 × 96, we compare the proposed model with the results of Maksim et al.[37] The train and test samples used in MobileNetV2, ResNet-34, ResNet-50, and VGG-19 models[37] are 11,760 and 9421, respectively. The proposed model uses the same number of training and test samples and repeats 10 times to obtain performance values. Regarding accuracy, the value of the Opt-ResDCNN model is 90.28% (see Table 6), which is better than Maksim et al.'s[37] 81.91%, 85.39%, 87.84%, and 84.81%.

In the fourth case with a resolution 256 × 256, we compare the proposed model with the results of Jin et al.[35] The train and test samples used in the CNN-ECOC-SVM model[35] are 18,000 and 2,000, respectively. The proposed model uses the same samples for training and test and repeats 10 times to obtain performance values. Table 7 shows the performance of the

TABLE 6 Class level classification performance results under 96 × 96 resolution

Failure type	Accuracy	Precision	Recall	F1 Value
Center	95.78%	95.41%	94.34%	94.86%
Donut	99.30%	99.14%	99.99%	99.56%
Edge-Loc	78.01%	76.69%	78.13%	77.17%
Edge-Ring	94.97%	94.54%	99.59%	96.99%
Loc	79.70%	82.48%	67.12%	73.79%
Random	99.61%	99.60%	100.00%	99.80%
Scratch	97.71%	97.47%	99.48%	98.46%
Near-full	88.75%	87.61%	89.64%	88.58%
None	78.66%	74.61%	83.83%	78.54%
Average	**90.28%**	**89.73%**	**90.24%**	**89.75%**

TABLE 7 Class level classification performance results under 256 × 256 resolution

Failure Type	Accuracy	Precision	Recall	F1 Value
Center	99.07%	99.09%	99.56%	99.32%
Donut	**100.00%**	**100.00%**	**100.00%**	**100.00%**
Edge-Loc	98.25%	97.98%	98.17%	98.06%
Edge-Ring	99.23%	99.30%	99.38%	99.34%
Loc	96.50%	96.06%	98.11%	96.99%
Random	**100.00%**	**100.00%**	**100.00%**	**100.00%**
Scratch	99.83%	99.83%	100.00%	99.92%
Near-full	99.78%	99.79%	99.68%	99.73%
None	97.26%	97.14%	94.79%	95.82%
Average	**98.88%**	**98.80%**	**98.80%**	**98.80%**

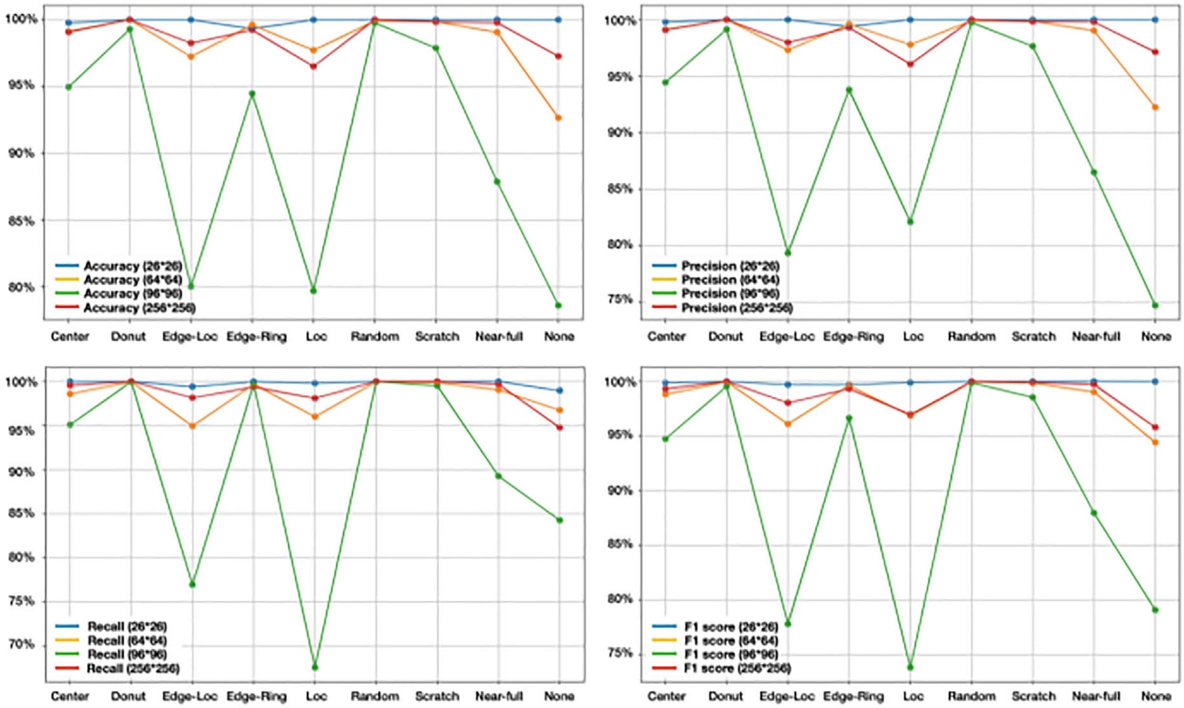

FIGURE 9 Class level classification performance metrics under different input image resolutions

proposed model in terms of accuracy precision, recall, and F1 score. Regarding accuracy, the value of the Opt-ResDCNN model is 98.88%, which is better than the Jin et al.'s[35] 98.43%.

The summary of the class level classification performance of the proposed model for the four different cases discussed above is shown in Figure 9. Relative to other classes, the proposed model well learns the Center, Donut, Edge-Ring, Random, and Scratch classes with accuracy, precision, recall, and F1 values of higher than 95% most of the time. The None and Loc classes have the lowest percentage values in all input image resolution cases. From the figure, we can see that the class level classification performance metrics curves for the four input image resolution cases show similar trends indicating that the proposed model is robust and suitable for WM-811K dataset.

The comparison results of the proposed model with previously published results for four different cases (input image resolutions) show that the proposed model outperforms the previously published results in all performance metrics for the four input image resolution cases. Therefore, we can conclude that introducing residual block and dropout layer in the DCNN architecture will help get a better classification result by solving the gradient vanishing problem.

5 | CONCLUSION

Aiming at the wafer map data set of unbalanced defects, we propose a DCNN model with a residual block with the optimal dropout for wafer map pattern identification and classification. The WM-811K data set is used to verify the proposed model's performance. The four input image sizes, including 26 × 26, 64 × 64, 96 × 96, and 256 × 256, are considered. The results show that the proposed model with a balance function and convolutional autoencoder can improve the model's classification performance. Besides, the model's credibility was verified by performing 10 times and calculating the average of the measured performance indicators.

Regarding accuracy, the value of the Opt-ResDCNN model on resolution 26 × 26 is 99.90%, which is better than the 99.29% of [16]. In addition, the four performance metrics of the proposed model used to identify donut, random, scratch, and near-full defect types are all 100%. For the resolution 64 × 64, the value of the Opt-ResDCNN model is 99.90%, which is better than 97.10% of Tsai and Lee[36] and 98.30% of Ji and Lee.[34] For the resolution 96 × 96, the value of the Opt-ResDCNN model is 90.28%, which is better than Maksim et al.'s[37] 81.91%, 85.39%, 87.84%, and 84.81%. The value of the Opt-ResDCNN model is 98.88%, which is better than the Jin et al.'s[35] 98.43%. These results show that the proposed model is a reliable classifier for wafer map pattern identification and classification.

ACKNOWLEDGMENTS

The Ministry of Science and Technology in Taiwan funded this research under Grant MOST-107-2221-E011-100-MY3.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in LSWMD.pkl at https://www.kaggle.com/qingyi/wm811k-wafer-map.

ORCID

Fu-Kwun Wang https://orcid.org/0000-0003-4563-945X

REFERENCES

1. Huang W, Kao N, TSMC Recognized with 2021 IEEE Corporate Innovation Award. 2020 Dec 09 [cited 2021 Apr 06].In: tsmc [Internet]: Available from: https://pr.tsmc.com/english/news/2765
2. Wang R, Chen N. Defect pattern recognition on wafers using convolutional neural networks. *Qual Reliab Eng Int*. 2020;36(4):1245-1257.
3. Yuan T, Kuo W, Bae SJ. Detection of spatial defect patterns generated in semiconductor fabrication processes. *IEEE Trans Semicond Manuf*. 2011;24(3):392-403.
4. Chen FL, Liu SF. A neural-network approach to recognize defect spatial pattern in semiconductor fabrication. *IEEE Trans Semicond Manuf*. 2000;13(3):366-373.
5. Gleason SS, Tobin KW, Karnowski TP, Lakhani F, Rapid yield learning through optical defect and electrical test analysis. In: *Proceeding. volume 3332 Metrology, Inspection, and Process Control for Microlithography*. Santa Clara, CA, USA, 1998:232-242.
6. Ishida T, Nitta I, Fukuda D, Kanazawa Y, Deep learning-based wafer-map failure pattern recognition framework. In: *Proceedings of 20th International Symposium on Quality Electronic Design (ISQED)*, Santa Clara, CA, USA, 2019:291-297.
7. Wu M, Jang JR, Chen J. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Trans Semicond Manuf*. 2015;28(1):1-12.
8. Yu J, Lu X. Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. *IEEE Tran Semicond Manuf*. 2016;29(1):33-43.
9. Fan M, Wang Q, Van der Waal B, Wafer defect patterns recognition based on OPTICS and multi-label classification. In: *Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. Xi'an, China, 2016:912-915.
10. Piao M, Jin CH, Lee JY, Byun JY. Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features. *IEEE Trans Semicond Manuf*. 2018;31(2):250-257.
11. Saqlain M, Jargalsaikhan B, Lee JY. A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. *IEEE Trans Semicond Manuf*. 2019;32(2):171-182.
12. Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks. *Pattern Recognit*. 2018;77:354-377.
13. He K, Zhang X, Ren S, Sun J, Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, 2016:770-777.
14. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ, Densely connected convolutional networks. In: *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017:2261-2269.
15. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278-2324.

16. Shawon A, Faruk MO, BinHabib M, Khan AM, Silicon wafer map defect classification using deep convolutional neural network with data augmentation. In: *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. Chengdu, China, 2019:1995-1999.
17. Yu J, Zheng X, Liu J. Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map. *Comput Ind*. 2019;109:121-133.
18. Johnson JM, Khoshgoftaar TM. Survey on deep learning with class imbalance. *J Big Data*. 2019;6(27).
19. Lee H, Park M, Kim J, Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In: *2016 IEEE International Conference on Image Processing (ICIP)*. Phoenix, AZ, USA, 2016:3713-3717.
20. Pouyanfar S, Tao Y, Mohan A, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In: *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. Miami, FL, USA, 2018:112-117.
21. Buda M, Maki A, Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw*. 2018;106:249-259.
22. Wang S, Liu W, Wu J, Cao L, Meng Q, Kennedy PJ, Training deep neural networks on imbalanced data sets. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, BC, Canada, 2016:4368-4374.
23. Khan SH, Hayat M, Bennamoun M, Sohei FA, Togneri R. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans Neural Netw Learn Syst*. 2018;29(8):3573-3587.
24. Wang H, Cui Z, Chen Y, Avidan M, BenAbdallah A, Kronzer A. Predicting hospital readmission via cost-sensitive deep learning. *IEEE/ACM Trans Comput Biol Bioinforma*. 2018;15(6):1968-1978.
25. Zhang C, Tan KC, Ren R, Training cost-sensitive deep belief networks on imbalance data problems. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, BC, Canada, 2016:4362-4367.
26. Huang C, Li Y, Loy CC, Tang X, Learning deep representation for imbalanced classification. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, 2016:5375-5384.
27. Lin TY, Goyal P, Girshick R, He K, Dollar P, Focal loss for dense object detection. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy, 2017:2999-3007.
28. Zhang Y, Shuai L, Ren Y, Chen H, Image classification with category centers in class imbalance situation. In: *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. Nanjing, China, 2018:359-363.
29. Ando S, Huang CY, Deep over-sampling framework for classifying imbalanced data. [Internet]. 2017 Jul [cited 2021 May 19]. Available from: http://arxiv.org/abs/1704.07515v3
30. Dong Q, Gong S, Zhu X. Imbalanced deep learning by minority class incremental rectification. *IEEE Trans Pattern Anal Mach Intell*. 2019;41(6):1367-1381.
31. Lee H, Kim J, Kim B, Kim S, Convolutional autoencoder based feature extraction in radar data analysis. In: *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*. Toyama, Japan, 2019:81-84.
32. Wang J, Yang Z, Zhang J, Zhang Q, Chien WK. AdaBalGAN: an improved generative adversarial network with imbalanced learning for wafer defective pattern recognition. *IEEE Trans Semicond Manuf*. 2019;32(3):310-319.
33. Yu J, Liu J. Two-dimensional principal component analysis-based convolutional autoencoder for wafer map defect detection. *IEEE Trans Ind Electron*. 2021;68(9):8789-8797.
34. Ji Y, Lee JH, Using GAN to improve CNN performance of wafer map defect type classification: yield enhancement. In: *2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. Saratoga Springs, NY, USA, 2020:1-6.
35. Jin CH, Kim HJ, Piao Y, Li M, Piao M. Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes. *J Intell Manuf*. 2020;31(8):1861-1875.
36. Tsai TH, Lee YC. A light-weight neural network for wafer map classification based on data augmentation. *IEEE Trans Semicond Manuf*. 2020;33(4):663-672.
37. Maksim K, Kirill B, Eduard Z, et al. classification of wafer maps defect based on deep learning methods with small amount of data. In: *2019 International Conference on Engineering and Telecommunication (EnT)*. Dolgoprudny, Russia, 2019:1-5.
38. Simonyan K, Zisserman A, Very deep convolutional networks for large-scale image recognition. [Internet]. 2015 Apr [cited 2021 May 12]. Available from: https://arxiv.org/abs/1409.1556
39. Howard AG, Zhu M, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. [Internet]. 2017 Apr [cited 2021 May 12]. Available from: https://arxiv.org/abs/1704.04861v1
40. Shen Z, Yu J, Wafer map defect recognition based on deep transfer learning. In: *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. Macao, China, 2019:1568-1572.
41. Hou B, Yan R, Convolutional auto-encoder based deep feature learning for finger-vein verification. In: *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. Rome, Italy, 2018:1-5.

AUTHOR BIOGRAPHIES

Fu-Kwun Wang received the M.S. degree in statistics and Ph.D. degree in industrial engineering from Arizona State University, Tempe, USA, in 1990 and 1996, respectively. Currently, he is a Distinguished Professor in the Department

of Industrial Management at the National Taiwan University of Science and Technology, Taiwan. His fields of interest are reliability engineering, quality control, and predictive analytics.

Jia-Hong Chou received the M.S. degree in industrial management from the National Taiwan University of Science and Technology, Taiwan, in 2021. His fields of interest are predictive maintenance and deep learning models.

Zemenu Endalamaw Amogne received the B.S. and M.S. degrees in industrial engineering from Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia, in 2012 and 2015, respectively. He is currently a Ph.D. candiate in the Department of Industrial Management at the National Taiwan University of Science and Technology, Taiwan. His fields of interest are predictive maintenance, fault detection and classification, and quality management.

How to cite this article: Wang F-K, Chou J-H, Amogne ZE. A deep convolutional neural network with residual blocks for wafer map defect pattern recognition. *Qual Reliab Eng Int*. 2021;1–15. https://doi.org/10.1002/qre.2983