IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 16, 2020, accepted January 28, 2020, date of publication January 30, 2020, date of current version February 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970461

A Light-Weighted CNN Model for Wafer Structural Defect Detection

XIAOYAN CHEN[1], JIANYONG CHEN[1], XIAOGUANG HAN[1], CHUNDONG ZHAO[1], DONGYANG ZHANG[1], KUIFENG ZHU[2], AND YANJIE SU[2]

[1]College of Electronic Information and Automation, Tianjin University of Science and Technology, Tianjin 300222, China
[2]Tianjin FLY Tech Company, Ltd., Tianjin 300350, China

Corresponding author: Xiaoyan Chen (cxywxr@tust.edu.cn)

This work was supported in part by the Tianjin Municipal Science and Technology Bureau under Grant 18YFZCGX00360, and in part by the National Natural Science Foundation of China under Grant 61903274 and Grant 41704131.

ABSTRACT Silicon wafer is the raw material of semiconductor chip. It is important and challenging to research a fast and accurate method of identifying and classifying wafer structural defects. To this end, we present a novel detection method in terms of the convolution neural networks (CNN), which achieve more than 99% detection accuracy. Due to the wafer images are not available by open datasets, a set of imaging acquisition system is designed to capture wafer images. Digital image preprocessing technology is utilized to split a wafer image into thousands of silicon grain images. The proposed model, called WDD-Net, uses depthwise separable convolutions and global average pooling to reduce parameters and calculations, adopts multiple 1*1 standard convolutions to increase the network depth. Specifically, two types of CNN models, VGG-16 and MobileNet-v2, are adopted for comparative analysis. Using the aforementioned three models, the comparative experiments are implemented on data sets that consisting of more than ten thousand grain images. The experimental results show that compared with VGG-16 and MobileNet-v2, the detection speed of the WDD-Net is 105.6FPS, which is 5 times faster. The model size of the WDD-Net is 307KB, which is much smaller than the other two. Furthermore, the WDD-Net directly completes the data collection and defect detection process through the local computing equipment, which is suitable for edge computing.

INDEX TERMS Image classification, neural networks, semiconductor manufacturing, machine learning.

I. INTRODUCTION

With the development of science and technology, chip has become the basic energy for industrial production, and its quality is of vital importance. Silicon wafer processing is the basis of semiconductor manufacturing [1], [2]. Wafer defect detection is one of the key challenges facing the semiconductor manufacturing companies. Different wafer defects have unique patterns for identification and classification due to their spatial dependence on wafer images. Inspectors identify wafer defects by inspecting wafer images. Since the inspection task requires extreme concentration, the time that an inspector can continue the task is quite limited, and still, it tends to be quite slow and inaccurate [3]–[5].

With the development of inspection technology, automatic wafer defect detection (WDD) has become a research hotspot. Because silicon wafers are soft and fragile, they must be detected by non-contact measurements. In recent studies, the machine vision system has the advantages of wide measurement range, no contact and high stability [6], [7]. Building machine vision system and implementing defect detection based on machine learning methods is very suitable for the non-contact detection task [8]–[10]. The machine vision systems include the following processes mostly: image acquisition, image processing, judgment and recognition, and automatic marking. The machine learning algorithms extract edge feature, surface texture and pattern information from the collected images, process and output the image recognition results, which are the core of machine vision systems [11]–[13].

Machine learning algorithms can be divided into two categories, CNN-based methods and non-CNN-based methods. Non-CNN-based methods such as k-means [14], multi-frame differential image summation [15], template matching [16] and spectral subtraction [17], [18] are effective for detecting defects in some larger size wafers.

The associate editor coordinating the review of this manuscript and approving it for publication was Ying Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

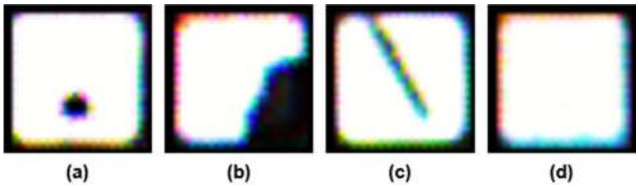


FIGURE 1. Three structural defects and defect-free(d). Redundant(a) is a wafer surface defect caused by tiny particles, dust, residues, etc. Crystal defects(b), also known as slip line defects, caused by uneven heating during crystal growth. Mechanical damage(c) is generally caused by chemical mechanical grinding in the steps of polishing, slicing, etc.

With the in-depth study, convolutional neural networks (CNN) are invoked to find high accuracy detection approaches. CNN, as an excellent feature extraction module, becomes an indispensable part for image classification. They have achieved a great success on kinds of tasks, such as weather recognition [19], industrial components defect detection [20], video surveillance [21] and so on.

In the wafer defect detection, CNN also achieved a good performance. Hui Han proposed a defect segmentation method for polycrystalline silicon wafer by means of the deep convolutional networks, which can segment various defects in silicon wafer by training with small amount of roughly marked defect images [22]. Nakazawa presented a method for wafer map defect detection by using CNN. A total of 28600 synthetic wafer maps were used for CNN training, validation, and testing. This method has an overall classification accuracy of 98.2% for 6600 test dataset [23]. Kyeong Kiryong used the CNN-based method to classify mixed-type defect patterns in wafer bin maps in the framework of an individual classification model for each defect pattern [24]. Jianbo Yu proposed stacked convolutional sparse denoising auto-encoder (SCSDAE) for wafer map pattern recognition (WMPR), in which the features can be extracted from maps directly [25]. Sejune Cheon given an automatic defect classification method based on deep learning that automatically classifies various types of wafer damage by adopting a single CNN model to extract features without additional feature extraction algorithms [26]. Unsupervised learning networks such as encoder-decoder neural network [27], and generative adversarial network [28] are used for wafer defect feature segmentation recently.

The aforementioned methods focused mainly on the detection of wafer mixed-type defect and wafer map. However, the classification of wafer structure defects is rarely considered. Wafer structure defect mainly refers to the component defects on the surface of the wafer. A wafer usually consists of thousands of chips, called grains. There are three common structural defects on grains in practice, namely redundancy, crystal defects and mechanical damage, as shown in Fig.1(a), (b), (c). A normal grain should be crystal-complete, regular in shape and free of turbidity, as shown in Fig.1(d).

This paper focuses on the detection of structural defects in wafer. A novel model, named WDD-Net is proposed and specified. VGG-16, a high-precision CNN model, and MobilenNet-v2, a lightweight CNN model, are selected as comparison research. Comparative experiments are implemented on the detection speed, detection accuracy and model size respectively to verify the performance of the models.

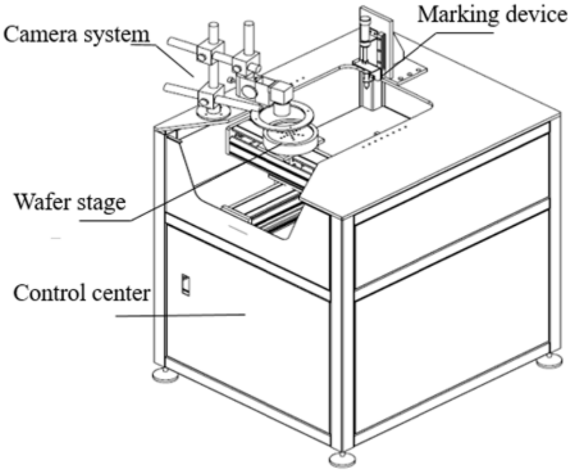


FIGURE 2. The wafer inspection device.

The remainder of this paper is organized as follows. In Section 2, we introduce the machine vision system and data preprocessing process. In Section 3, the components of WDD-Net and two comparison models VGG-16 and MobileNet-b2 are specified. In Section 4, data augmentation methods are introduced to solve the data unbalance problem. In Section 5, comparative experiments are implemented in the aspects of detection speed, detection accuracy and model size, respectively. Section 6 summarizes this work and briefly discusses possible future extensions.

II. MACHINE VISION SYSTEM AND DATA PREPROCESSING

A. MACHINE VISION SYSTEM

Due to the wafer images are not available by open datasets, a set of wafer inspection device shown in Fig.2 is designed and manufactured. It is composed of four parts: camera system, wafer stage, control center and marking device.

The schematic diagram is shown in Fig.3. A MD-UB1000 CMOS camera is used to capture wafer images. The camera is fixed by the lead screw directly above the detection device. The lens of the camera is MV-JT08. The FJI-RL150-A00-W ring light source is selected to ensure that all grains in the wafer receives light evenly. The resolution of the camera is 3664*2748. As shown in Fig.4(b), dozens of ventilation holes evenly distributed on the wafer stage. The wafer is gently adsorbed by small negative pressure from an air pump. The control center consists of two programmable logic controllers (PLCs) and a computer. The PLCs control the servo motors to move the wafer stage by the coordinates inferred from detection results. The computer is in charge of processing images, running algorithms, and outputting

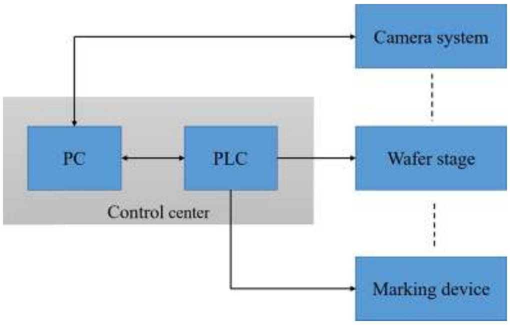


FIGURE 3. The schematic diagram of the wafer inspection device.

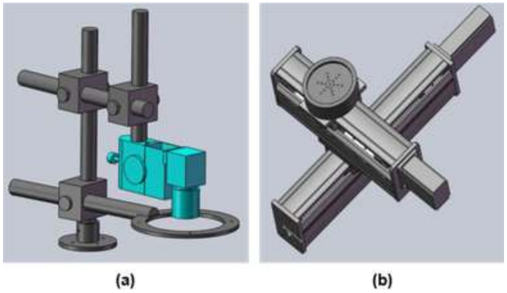


FIGURE 4. Image acquisition system. camera system (a), wafer stage (b).

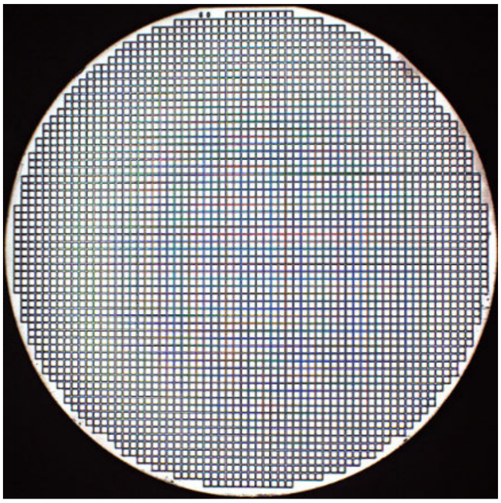


FIGURE 5. A standard wafer image.

detection results. The marking device marks the defect grains. It is assembled and debugged in our lab.

The wafers in this article are four inches, as shown in Fig.5. Each grid in the circular wafer image contains a grain. The size of a grain is 60 mils, which is 1.524 mm. One pixel in the standard wafer image is approximately equal to 0.0544mm.

B. DATA PREPROCESSING

As shown in Fig.5, the image acquisition system captures entire wafer images. The image we need for CNN-based detection is the grains shown in Fig.1, so we partition the entire wafer image into separated grain images. As shown

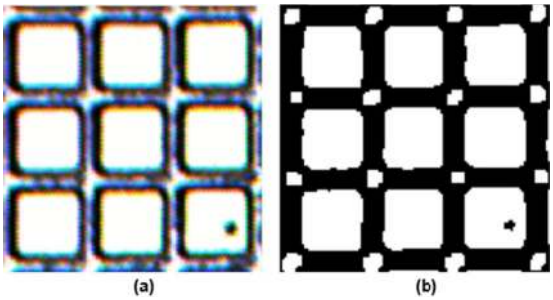


FIGURE 6. Local area before and after processing. Before processing(a), after processing(b).

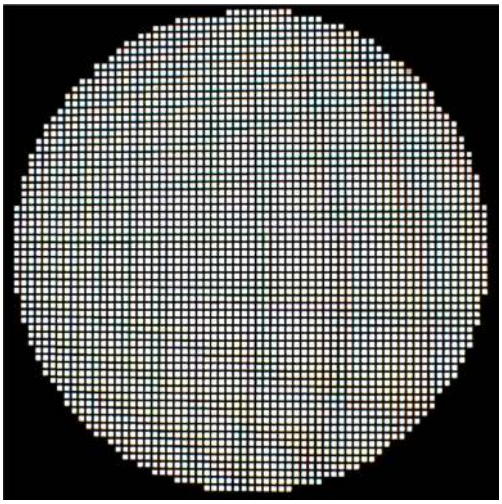


FIGURE 7. Contour of all contours.

in Fig.6(a), it is clear to see that there are some noise dots between the gaps, which will cause the contours of the grains to be no longer independent and complete. The open operation has the effect of smoothing the contour of the grains, breaking the narrow connection and removing the small protruding parts. Choosing the appropriate threshold and mask ensure the noise information in the grid gap be removed while the contour information is retaining. It can be seen that the edges of grains are obviously clear and most noise dots are eliminated in Fig.6(b).

We take the Findcontours function in open source computer vision library (OpenCV) to implement contour extraction. The hierarchical tree indexing and the rectangular contour approximation method are used to find all contours. The actual grain size is 1.524mm, corresponding to 28*28 pixels in the wafer image. The contours that far beyond and below 28 pixels are not advisable. Thus we delete the contours whose width and height are less than 20 pixels or more than 35 pixels. Wafer image as shown in Fig.5 can be extracted 4324 contours, 3240 contours are retained with the restriction. To filter the background noise, the 3240 contours are displayed in a blank template as shown in Fig.7.

The returned array of Findcontours function stores the width, height, and coordinate information of each contour.

Cutting the original image shown in Fig.7 in accordance with the width, height, and coordinate information of each contour. In this way we cut 3240 contours into 3240 grain images. By the preprocessing, we get the grains without reduction or redundancy. Each grain image is labelled by the coordinates of the left-up corner of the grid.

III. METHODOLOGY

CNN are widely used in computer vision to extract translation-invariant features. As shown in Fig.8, CNN is composed of input layer, hidden layer and output layer. The hidden layer is usually composed of convolutional layer, pooling layer and fully connected layer, which is the core part of CNN. The front part of CNN extracts the representative high and low frequency features in the image. In the deeper hidden layer, more general and complete features are extracted. The outstanding characteristics of CNN-based methods exists in the following three aspects: (i) local connection ensures the filters response to local input sensitively. (ii) weight sharing reduces the parameters quantity greatly. (iii) pooling reduces dimensions of the data while retaining useful information.

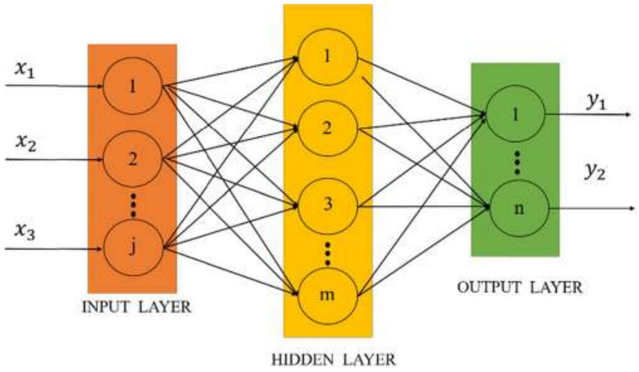


FIGURE 8. The basic structure of CNN.

Using the CNN-based methods, we transformed the wafer structural defect detection into an image classification of four classes (defect-free, redundancy, crystal defects and mechanical damage).

A. WDD-NET

Detection accuracy and detection speed are two of the most important indicators for wafer inspection task. Considering these two factors, we proposed a novel CNN-based model WDD-Net. The structure of the model is shown in Fig.9, and the layer outputs and parameters are shown in Table 1. The convolution part includes a 3*3 standard convolution and three depthwise separable convolution (3*3 separation convolution and 1*1 standard convolution). Considering that the brightness of the wafer foreground target (wafer defects) is lower than the background, we use the average pooling at the last layer to ensure the integrity of the information, preventing the loss of shallow features. We take global average pooling(GAP) [29] replace the fully connected layer. A very fatal weakness of the fully connected layer is that the number of parameters is too large, especially the fully connected layer connected to the last convolutional layer. The huge amount of parameters on the one hand leads to a reduction in the speed of training and testing. On the other hand, it may lead to over-fitting. The GAP can effectively reduce the parameter quantity, and is more robust to the transformation of spatial information. Finally, we use Softmax to output the classification results.

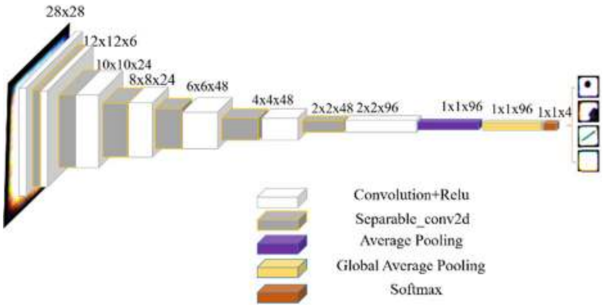


FIGURE 9. The architecture of WDD-Net.

TABLE 1. Layer outputs and parameters of WDD-Net.

Layer (Type)	Output Shape	Parameters
conv2d_1 (Conv2D)	(None, 14, 14, 6)	168
separable_conv2d_1 (SeparableConv2D)	(None, 12, 12, 6)	96
conv2d_2 (Conv2D)	(None, 12, 12, 6)	42
separable_conv2d_2 (SeparableConv2D)	(None, 10, 10, 12)	138
conv2d_3 (Conv2D)	(None, 10, 10, 24)	312
separable_conv2d_3 (SeparableConv2D)	(None, 8, 8, 24)	816
conv2d_4 (Conv2D)	(None, 8, 8, 24)	600
separable_conv2d_4 (SeparableConv2D)	(None, 6, 6, 24)	816
conv2d_5 (Conv2D)	(None, 6, 6, 48)	1200
separable_conv2d_5 (SeparableConv2D)	(None, 4, 4, 48)	2784
conv2d_6 (Conv2D)	(None, 4, 4, 48)	2352
separable_conv2d_6 (SeparableConv2D)	(None, 2, 2, 48)	2784
conv2d_7 (Conv2D)	(None, 2, 2, 96)	4704
average_pooling2d_1 (AveragePooling2D)	(None, 1, 1, 96)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 96)	0
dense_1 (Softmax)	(None, 4)	388

The core idea of WDD-Net is to reduce the amount of parameters and calculations while preserving the depth of the network. It has the following characteristics:

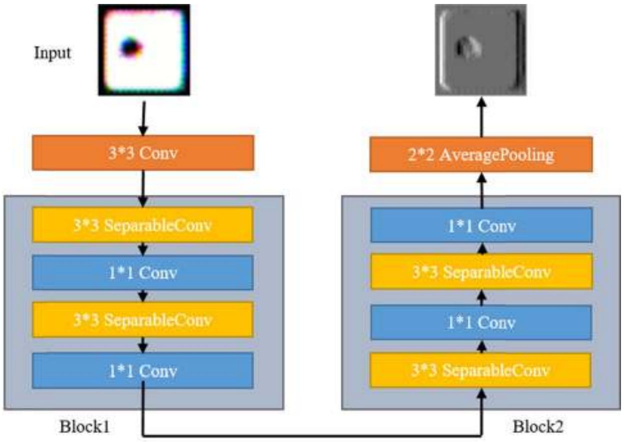


FIGURE 10. The convolutional combination structure of WDD-Net.

(i) Depthwise separable convolution combined with standard convolution. WDD-Net uses depthwise separable convolutions [30], [31] to reduce the amount of parameters and calculations, and uses multiple 1*1 standard convolutions to increase the network depth and improve model performance.

(ii) The feature complexity of the four types of grains is limited. In order to preserve shallow features such as edges and angles, the average pooling is adopted at the last layer.

(iii) Use the global average pooling layer instead of the fully connected layer.

Fig. 10 shows the convolutional combination structure of WDD-Net. The size of all depthwise separable convolution kernels is 3*3. After using the standard convolution of 3*3 once, the size of the rest of the standard convolution kernel is 1*1.

1) DEPTHWISE SEPARABLE CONVOLUTION

Fig.11 shows the composition of the depthwise separable convolution. For input (D_F, D_F, M), the standard convolution K(D_K, D_K, M, N), the standard convolution calculation formula is:

$$\mathrm{G}_{\mathrm{k,l,n}} = \sum_{\mathrm{i,j,m}} \mathrm{K}_{\mathrm{i,j,m,n}} \times \mathrm{F}_{\mathrm{k+i-1,l+j-1,m}} \tag{1}$$

The amount of calculation N_c is:

$$\mathrm{N_c} = \mathrm{D_K} \times \mathrm{D_K} \times \mathrm{M} \times \mathrm{N} \times \mathrm{D_F} \times \mathrm{D_F} \tag{2}$$

Split the standard convolution K(D_K, D_K, M, N) into a depthwise convolution(D_K, D_K, 1, M) and a pointwise convolution(1, 1, M, N). The separable convolution calculation formula is:

$$\dot{\mathrm{G}}_{\mathrm{k,l,n}} = \sum_{\mathrm{i,j}} \dot{\mathrm{K}}_{\mathrm{i,j,m,}} \times \mathrm{F}_{\mathrm{k+i-1,l+j-1,m}} \tag{3}$$

The amount of calculation is reduced:

$$\frac{\mathrm{D_K} \times \mathrm{D_K} \times \mathrm{M} \times \mathrm{D_F} \times \mathrm{D_F} + \mathrm{M} \times \mathrm{N} \times \mathrm{D_F} \times \mathrm{D_F}}{\mathrm{D_K} \times \mathrm{D_K} \times \mathrm{M} \times \mathrm{N} \times \mathrm{D_F} \times \mathrm{D_F}} = \frac{1}{\mathrm{N}} + \frac{1}{\mathrm{D_K^2}} \tag{4}$$

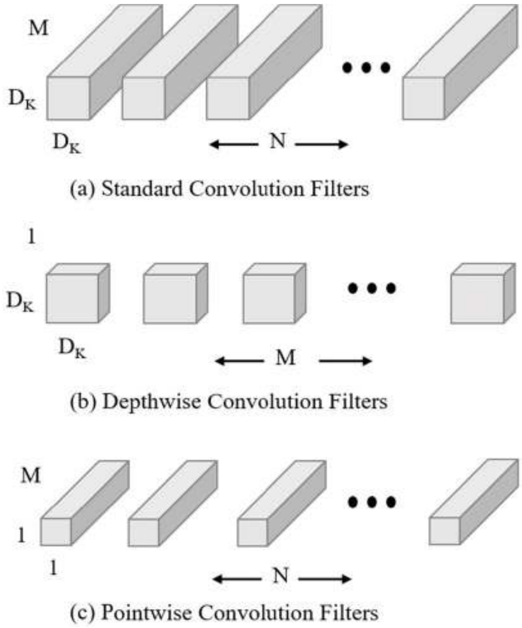

M
D_K
D_K
N
(a) Standard Convolution Filters
1
D_K
D_K
M
(b) Depthwise Convolution Filters
M
1
1
N
(c) Pointwise Convolution Filters

FIGURE 11. The composition of the depthwise separable convolution. The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c).

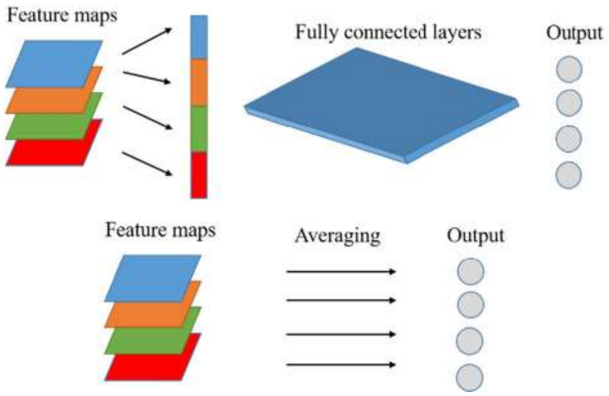


FIGURE 12. Global average pooling instead of fully connected layer.

2) GLOBAL AVERAGE POOLING(GAP)

Global average pooling calculates an average of all pixels in the feature map of each channel output from the previous layer, obtains a feature vector with the same dimensions and number of categories, and then inputs it to the Softmax layer. As shown in Fig.12, the global averaging pooling takes each feature graph as the confidence output corresponding to that category. A large number of parameters and calculations of the fully connected layer are saved.

3) Softmax

As shown in Fig.13, Softmax layer as the output layer. The Softmax layer maps the input to a range of values from 0 to 1, the sum of these values is 1.

$$\sum\nolimits_{\mathrm{i=1}}^{\mathrm{J}} \sigma_{\mathrm{i}}(\mathrm{z}) = 1 \tag{5}$$

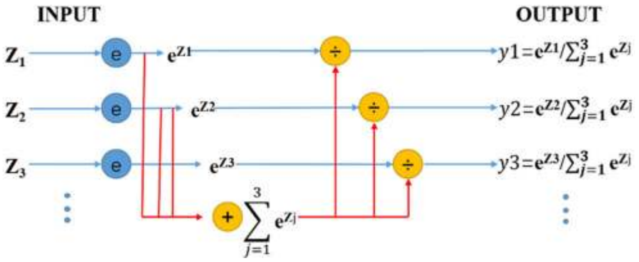

INPUT
OUTPUT
Z_1
Z_2
Z_3
e
e
e
e^{Z1}
e^{Z2}
e^{Z3}
÷
÷
÷
+
$\sum_{j=1}^{3} e^{Zj}$
$y1=e^{Z1}/\sum_{j=1}^{3} e^{Zj}$
$y2=e^{Z2}/\sum_{j=1}^{3} e^{Zj}$
$y3=e^{Z3}/\sum_{j=1}^{3} e^{Zj}$

FIGURE 13. The structure of the Softmax layer.

i is the subscript order of the nodes. The output node with the highest probability is selected as the prediction target. Each output node represents a classification, and the excitation function of each node is:

$$y_i = \sigma_i(z) = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_i}} \tag{6}$$

Cross-entropy is used in combination with Softmax to evaluate the difference between the probability distribution and the real distribution. It characterizes the distance between the actual output and the expected output. The smaller the value of the cross-entropy, the closer the two probability distributions are. The expression in the case of binary classification is:

$$\mathrm{L} = -[\mathrm{y} \times \log(\mathrm{p}) + (1-\mathrm{y}) \times \log(1-\mathrm{p})] \tag{7}$$

In the formula, y represents the label of the sample, the positive class is 1, and the negative class is 0. p represents the probability that the sample is predicted to be positive. Multi-classification is an extension of binary classification:

$$\hat{\mathrm{L}} = -\sum_{\mathrm{c}=1}^{\mathrm{M}} \mathrm{y_c} \log(\mathrm{p_c}) \tag{8}$$

In the formula, M is the number of categories. y_c is the indicator variable (0 or 1), 1 if the category is the same as the sample category, 0 otherwise. p_c is the prediction probability that the observation sample belongs to category c.

B. COMPARISON METHOD

1) VGG-16

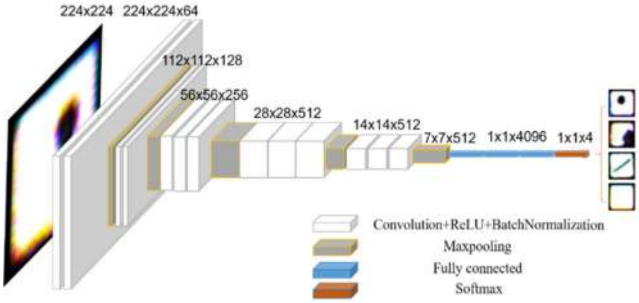


FIGURE 14. The architecture of VGG-16.

TABLE 2. Layer outputs and parameters of VGG-16.

Layer(Type)	Output Shape	Parameters
conv2d_1, conv2d_2 (Conv2D)	(None, 224, 224, 64)	38720
max_pooling2d_1(MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_3, conv2d_4 (Conv2D)	(None,112, 112, 128)	139520
max_pooling2d_2(MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_5, conv2d_6, conv2d_7 (Conv2D)	(None, 56, 56, 256)	951040
max_pooling2d_3 (MaxPooling2D)	(None, 28, 28, 256)	0
conv2d_8, conv2d_9, conv2d_10 (Conv2D)	(None, 28, 28, 512)	3802624
max_pooling2d_4 (MaxPooling2D)	(None, 14, 14, 512)	0
conv2d_11, conv2d_12, conv2d_13 (Conv2D)	(None, 14, 14, 512)	4982272
max_pooling2d_5 (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_1 (Dense)	(None, 4096)	102764544
dense_2 (Dense)	(None, 4096)	16781312
dense_3 (Dense)	(None, 1000)	4097000
dense_4 (Softmax)	(None, 4)	4004

VGG-16 performs well in image classification and target detection tasks [32]. VGG-16 consists of 13 convolution layers (divided by five maximum pooling layers) and three fully connected layers. VGG-16 has five convolution parts, each of which is composed of multi-layer convolution and maximum pooling. All the convolution layers have the same configuration: the size of the convolution core is 3*3, the step size is 1, and the filling size is 1. The size of maximum pooling is 2*2 and the step size is 2. The first two layers of the full connection layer are 4096 channels, and the third layer is 1000 channels representing 1000 label categories respectively. The last layer is Softmax layer. The number of convolution channels in the first layer is 64. The number of channels doubles with each maximum pooling. By stacking small convolution cores of 3*3 and maximum pooling of 2*2 repeatedly, the network performance is improved. The structure of the model is shown in Fig.14, the layer outputs and parameters are shown in Table 2.

2) MobileNet-v2

MobileNet was presented by the Google team and published on CVPR-2017 [33]. MobileNet-v2 is a network designed for mobile and embedded deep learning applications. The basic unit of MobileNet-v2 is depthwise separable convolution. The depthwise separable convolution can be broken down into two smaller operations: depthwise convolution and pointwise convolution. Standard convolution uses a standard convolutional convolution kernel on all input channels, while depthwise convolution uses a different convolution kernel for each input channel. Pointwise convolution is the 1*1 standard convolution. First, the depthwise convolution convolve the different input channels separately, and then the output is combined by pointwise convolution. This method reduces the amount of calculations and the amount of model parameters greatly. MobileNet-v2 first improves the dimensionality to extract features, and then reduces the dimensionality.

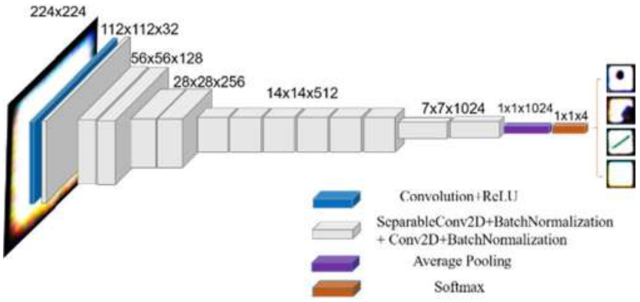


FIGURE 15. The architecture of MobileNet-v2.

TABLE 3. Layer outputs and parameters of MobileNet-v2.

Layer(Type)		Output Shape	Parameters
conv2d_1 (Conv2D)		(None, 112, 112, 32)	896
mobile_block_1	separable_conv2d_1 (SeparableConv2D)	(None, 112, 112, 32)	1344
	batch_normalization_1 (Batch Normalization)	(None, 112, 112, 32)	128
	conv2d_2 (Conv2D)	(None, 112, 112, 32)	1056
	batch_normalization_2 (Batch Normalization)	(None, 112, 112, 32)	128
	separable_conv2d_2 (SeparableConv2D)	(None, 56, 56, 64)	2400
	batch_normalization_3 (Batch Normalization)	(None, 56, 56, 64)	256
	conv2d_3 (Conv2D)	(None, 56, 56, 128)	8320
	batch_normalization_4 (Batch Normalization)	(None, 56, 56, 128)	512
5*mobile_block_1		(None, 7, 7, 1024)	3894912
separable_conv2d_13 (SeparableConv2D)		(None, 7, 7, 1024)	1058816
batch_normalization_25 (Batch Normalization)		(None, 7, 7, 1024)	4096
conv2d_14 (Conv2D)		(None, 7, 7, 1024)	1049600
batch_normalization_26 (Batch Normalization)		(None, 7, 7, 1024)	4096
average_pooling2d_1 (Average Pooling2D)		(None, 1, 1, 1024)	0
flatten_1 (Flatten)		(None, 1024)	0
dense_1 (Softmax)		(None, 4)	4100

The structure of the model is shown in Fig.15, and the layer outputs and parameters are shown in Table 3.

3) APPLICABILITY COMPARISON

VGG-16 is one of the most popular CNN models for image classification tasks. It has the advantages of high adaptability and high precision. It can process high dimensional data comprehensively and solve the problem of image classification and positioning of 1000 categories. However, the training process of VGG-16 needs a large number of data samples to support. And it requires a lot of computing and memory resources.

Mobilenet-v2 is an improved lightweight CNN model. The purpose of the lightweight model is to solve two problems of CNN. (i) Storage issues. Hundreds of layers of networks have a large number of weight parameters. Saving a large number of weight parameters has high requirements on the memory of the device. (ii) Speed issues. In practical applications, the speed standard is often in the millisecond level. Therefore, in addition to improving the performance of the processor, the amount of computation must be reduced. Mobilenet-v2 reduces parameters without losing network performance. It is well suited for deployment on mobile devices with priority on memory usage.

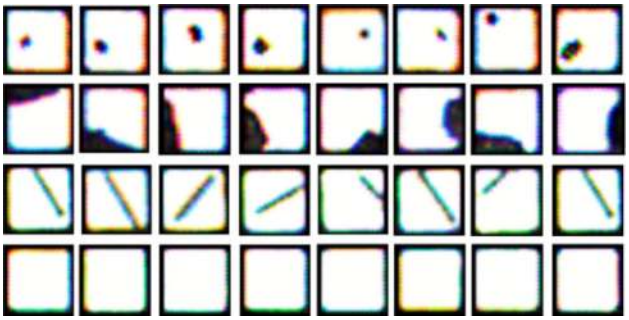


FIGURE 16. Four types of grain images.

WDD-Net is a model proposed in this paper in order to improve the detection speed and further reduce the model size. The complexity of wafer defect features is not high, so we pay more attention to the extraction of shallow features such as lines and angles. Compared to VGG-16 and MobileNet-v2, the network structure of WDD-Net is simpler. WDD-Net removes the fully connected layer and replaces it with GAP layer. The use of the maximum pooling layer is also omitted. It uses smaller convolution kernels and has shallower layers.

IV. DATA AUGMENTATION

In Section 2, we took the wafer image and segmented it into grain images. Fig.16 shows four types of grain images. Due to qualified grains (positive data) are occupied over 95% of a wafer and the defect grains (negative data) are very lack, the dataset is unbalanced in quantities of positive and negative data. In order to expand the number of defect images in the dataset and enhance the generalization ability of the model, two data augmentation methods were used.

A. AFFINE TRANSFORMATION

Affine transformation can increase the synthetic data and improve the robustness of the model. In this paper, the following three transformation methods are used.

Rotation. The images are randomly rotated along the X axis and Y axis. The rotation matrix is shown in Formula (9), (10). θ is the rotation factor.

$$\begin{bmatrix} X^T \\ Y^T \\ Z^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (9)$$

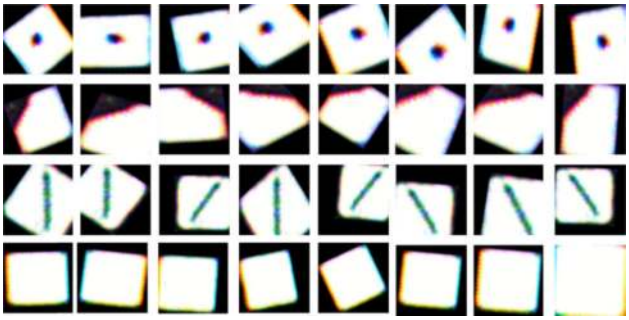


FIGURE 17. **Generated grain images.**

$$\begin{bmatrix} X^T \\ Y^T \\ Z^T \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{10}$$

Translation. The images are randomly shifted along the X axis and Y axis. The translation matrix is shown in Formula (11). T_X , T_Y are the translation factors.

$$\begin{bmatrix} X^T \\ Y^T \\ Z^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_X \\ 0 & 1 & T_Y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{11}$$

Scaling. The images are scaled randomly. The scaling matrix is shown in Formula (12). S is the scaling factor.

$$\begin{bmatrix} X^T \\ Y^T \\ Z^T \end{bmatrix} = \begin{bmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & S \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{12}$$

Fig.17 shows the generated grain images. It can be seen that the generated images enrich the feature structure of the wafer dataset to a certain extent.

B. GENERATIVE ADVERSARIAL NETWORK

Generating adversarial networks (GAN) is an important method for image generation. GAN mainly includes two parts, generator and discriminator. The generator is used to learn the real image distribution so as to make the generated image more real and fool the discriminator. The discriminator determines whether the generated image is true or false. Throughout the process, the generator worked hard to make the generated image more real, while the discriminator worked hard to identify the authenticity of the image. The generator and discriminator continued to fight each other, and finally the two networks reached a dynamic equilibrium: The image generated by the generator is close to the real image distribution. The discriminator cannot recognize the true and false images and the probability of the true prediction is basically close to 0.5 (equivalent to a random guess category).

Fig.18 shows the process of generating grain images. The first-generation generator generates poor images, and the first-generation discriminator can accurately distinguish the generated images from real images. Then, the quality of the pictures generated by the trained second-generation generator is improved, which can deceive the discriminator of the first generation. Subsequently, the trained second-generation discriminator can accurately distinguish the generated pictures. By analogy, when the nth-generation discriminator cannot distinguish the generated picture from the real picture, the network fits.

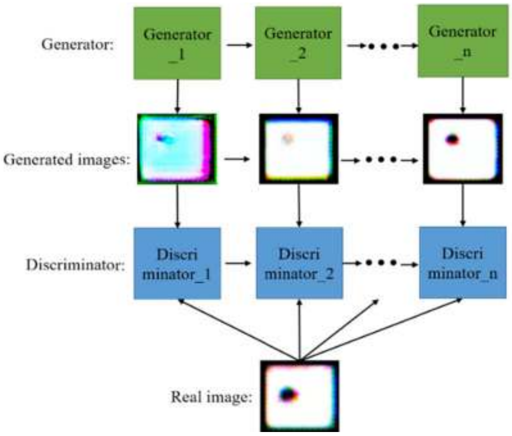


FIGURE 18. **The process of generating grain images.**

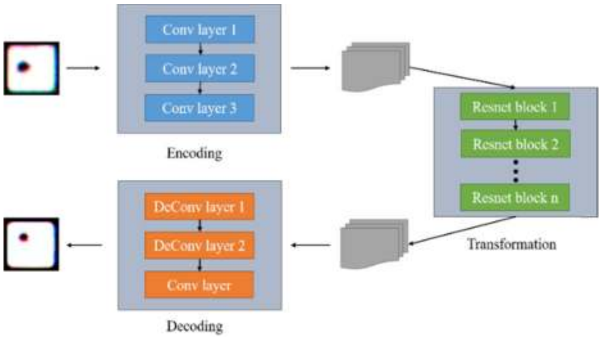


FIGURE 19. **The structure of the generator.**

The generator used in this paper consists of an encoder, a converter, and a decoder, as shown in Fig.19. The encoder uses a convolutional neural network to extract features from the input image, and compresses the image into multiple feature vectors. The converter combines multiple features of the image. The decoder uses the deconvolution layer to restore the low-level features from the feature vector, and finally obtains the generated image.

The discriminator itself belongs to a convolutional neural network. It needs to extract features from the image, and then determines whether the extracted features belong to a specific category by adding a convolution layer that generates a one-dimensional output, as shown in Fig.20.

x is the input picture, and its distribution is $P_{data}(x)$. The distribution of the generator is $P_g(x)$. For the fixed generator G, the optimal discriminator D is:

$$D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)} \tag{13}$$

According to the virtual training criterion, the global minimum is reached only when $P_g(x) = P_{data}(x)$. The objective

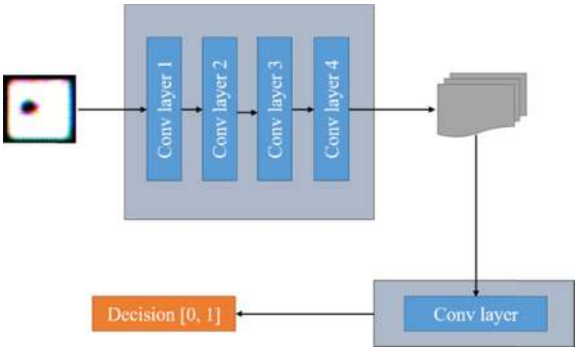


FIGURE 20. The structure of the discriminator.

TABLE 4. The distribution of the dataset.

	Redundant	Crystal defects	Mechanical damage	Defect-free
Total	3568	6059	1208	2679
Training set	2142	3635	724	1607
Validation set	713	1212	242	536
Test set	713	1212	242	536

TABLE 5. Model hyper-parameter settings.

Model	Input Size	Batch Size	Epochs	Activation	Optimizer	Learning Rate
VGG-16	224*224	64	50	ReLU	Adam	0.0001
MobileNet-v2	224*224	32	50	ReLU	RMSProp	0.0004
WDD-Net	28*28	32	50	ReLU	Adam	0.001

function of GAN is $V_{(D,G)}$.

$$V_{(D,G)}{}^{\max,\min} = E_{x\sim P_{data}(x)}\left[\log D_{(x)}\right] + E_{z\sim P_{z}(z)}[\log(1-D(G_{(z)}))] \quad (14)$$

E represents the mathematical expectation of real data x and noise data z. P_z is the Gaussian distribution of the noise data.

V. EXPERIMENTATIONS AND RESULT ANALYSIS

A. TRAINING

As shown in Table 4, a total of 13514 grain images were selected for experiments, including 3568 pieces of redundant, 6059 pieces of crystal defects, 1208 pieces of mechanical damage, 2679 pieces of defect-free. The ratio of training set, validation set, and test set was 6:2:2.

The hyper-parameter settings of three models are shown in Table 5. The experimental software environment was Python3.6, TensorFlow-Gpu1.8.0, Cuda9.0, Keras2.1.4, and the hardware platform was NVIDIA TITAN Xp GPU, Intel Core I7-9700K CPU @3.60GHz.

In section 2, the size of the grain images we obtained is 28 * 28. WDD-Net removes the fully connected layer, and there is no requirement for the input size. We directly input the grain images of 28*28. The input size requirements of VGG-16 and MobileNet-v2 are 224*224. The size change of the input vector results in the weight parameters change of the full connected layer. This leads to dynamic variation in the network and the parameter training impossible. So we resized the input grain images of VGG-16 and MobileNet-v2 to 224 * 224. For VGG-16, the activation function was ReLU and the optimizer was Adam. It was found that a large learning rate led to the non-convergence of VGG-16. So we set the learning rate to 0.0001. For MobileNet-v2, we chose the RMSProp optimizer with a learning rate of 0.0004. For WDD-Net, the activation function was ReLU and the optimizer was Adam. The learning rate is 0.001.

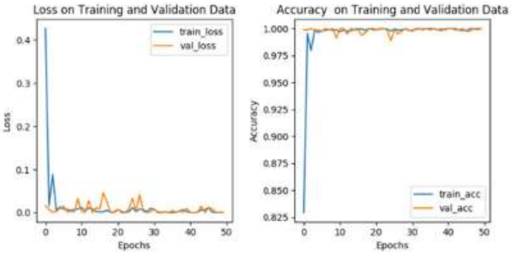


FIGURE 21. The loss and accuracy curves of VGG-16 on the training and validation data.

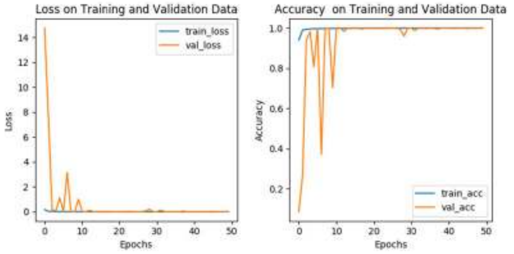


FIGURE 22. The loss and accuracy curves of MobileNet-v2 on the training and validation data.

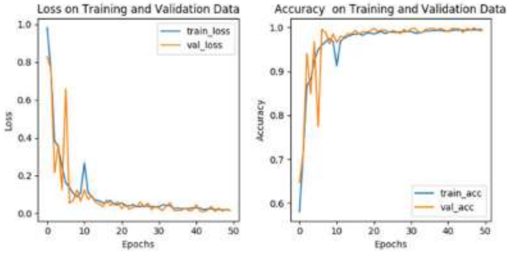


FIGURE 23. The loss and accuracy curves of WDD-Net on the training and validation data.

Fig.21, Fig.22, and Fig.23 show the loss and accuracy curves of the three models on the training and validation data, respectively. After 50 epochs, the loss dropped to below 0.01, and the accuracy increased to nearly 1. This proves that the three models converge well, and there is no overfitting.

TABLE 6. The detection speed of the three models.

Model	Detection speed(FPS)
VGG-16_224*224	17.5
MobileNet-v2_224*224	17.3
WDD-Net_28*28	8719.3
WDD-Net_224*224	105.6

B. DETECTION SPEED COMPARISON

In practice, the number of grains is huge. Therefore, the most preferred evaluation index is detection speed. We count the number of grains that can be detected per second of three methods. The detection speeds (FPS) of the three methods are obtained, as shown in Table 6. We find that the detection speed of VGG-16 and MobileNet-v2 are basically the same and the 28*28 input WDD-Net is significantly faster than them. Due to the different computational complexity, the processing speed of small input size is faster, which is part of the reason why the detection speed of WDD-Net 28 *28 is much faster than that of VGG-16 and MobileNet-v2.

For further comparison, we adjust the input of WDD-Net to 224*224, which is consistent with that of VGG and MobileNet-v2. It can be seen that, compared with VGG-16 and MobileNet-v2, the detection speed of WDD-Net 224*224 is still the fastest, about 5 times as fast as them. This indicates that the simplified structure of WDD-Net improves the computing speed to some extent, the detection speed improved.

C. DETECTION ACCURACY COMPARISON

Detection accuracy is the core evaluation index of wafer defect detection method. We verify the accuracy of the three methods on the test set. The detection accuracy of the three methods is shown in Table 7. We find that on the whole test set, the detection accuracy of the three models is basically the same, reaching over 99%. This indicates that all three CNN wafer defect detection methods can effectively extract defect features and classify defect patterns.

The detection accuracy of WDD-Net is slightly lower than that of VGG-16 and MobileNet-v2 in the discrimination of the crystal defects and mechanical damage. This indicates that the recognition capability of VGG-16 and MobileNet-v2 is better than that of WDD-Net under limited data sets. The simplified structure of WDD-Net reduces the number of parameters and computation, but slightly reduces the identification accuracy.

For comparison, we resize the 28 * 28 grain images to 224 * 224 and input them to WDD-Net. We find that the resize operation do not improve detection accuracy. In contrast, the detection accuracy of WDD-Net is reduced. Although the resize operation enlarges the input size, the filling method of interpolation loses the image information to a certain extent, which affects the recognition result of the model.

TABLE 7. The detection accuracy of the three models.

Model	Redundant	Crystal defects	Mechanical damage	Defect-free	Whole test set
VGG-16	99.86%	100%	100%	100%	99.96%
MobileNet-v2	100%	100%	100%	100%	100%
WDD-Net_28*28	100%	99.63%	97.52%	100%	99.70%
WDD-Net_224*224	100%	99.63%	94.63%	100%	99.44%

D. MODEL SIZE COMPARISON

Model size is also an important evaluation index for wafer detection methods. The parameters of deep convolutional neural network are huge, and the calculation of convolutional layer and fully connected layer requires a large number of floating-point matrix multiplication, resulting in very high computational overhead. Although some networks can run in real time on the GPU, they cannot be directly applied to embedded edge computing devices. Small size model takes up less disk space, reduces the memory used during inference, and makes calculation faster.

The parameters and model size of the three models are shown in Table 8. We found that VGG-16 has more than 100 million parameters and its model size is very large (1.5GB). MobileNet-v2 has fewer parameters than VGG-16 and a smaller model size of 46.2MB. WDD-Net has the fewest parameters (17200) and the model size is minimal (307KB). The structure of VGG-16 and MobileNet-v2 are complex, and the amount of parameters and model size are huge. WDD-Net has a compact structure with fewer parameters and smaller model sizes.

TABLE 8. The parameters and model size of the three models.

Model	Total parameters	Trainable parameters	Non-trainable parameters	Hdf5 Model size
VGG-16	133561036	133561036	0	1.50GB
MobileNet-v2	6030660	6008900	21760	46.2MB
WDD-Net	17200	17200	0	307KB

E. COMPREHENSIVE COMPARISON

We comprehensively compare the three evaluation indicators to determine the most effective wafer detection method. As shown in Fig.24, the model sizes of MobileNet-v2 and WDD-Net are 3% and 0.02% of VGG-16, respectively. The detection speed of VGG-16 and MobileNet-v2 is 0.2% of WDD-Net. As shown in Fig.25, the detection accuracy of WDD-Net 28 * 28 is slightly lower than that of VGG-16 and MobileNet-v2.

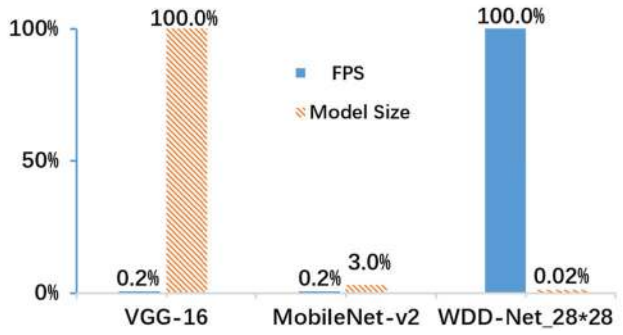


FIGURE 24. FPS and model size of three models.

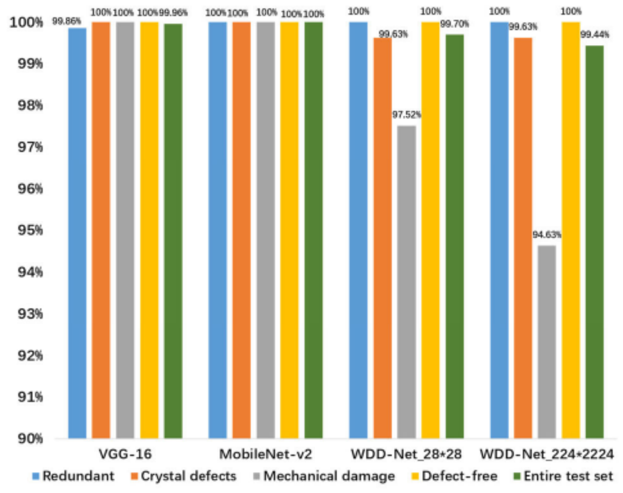


FIGURE 25. The detection accuracy of three models.

The model size of VGG-16 is too large and the detection speed is slow. VGG-16 method is not suitable for wafer defect detection. Using MobliNet-v2 method can obtain the highest detection accuracy, however, its detection speed cannot meet the needs of practical applications. The detection accuracy of WDD-Net_28*28 method is slightly lower than that of MobileNet-v2, the detection speed is extremely fast. In addition, the model size of WDD-Net is only 307KB, which makes it very convenient to transplant on micro-devices. Overall, compared with VGG-16 and MobileNet-v2, WDD-Net lost tiny detection accuracy, increased detection speed by more than 5 times, and reduced model size by more than 100 times. This proves that WDD-Net is a more effective wafer defect detection method in practical applications.

VI. CONCLUSION

In this paper, we investigate a fast and accurate wafer defect detection method by using CNN. The experimental results shown that although the VGG-16 and MobileNet-v2 detection methods have higher detection accuracy, the detection speed still need to improve. For this reason, we proposed a novel CNN-based model, called WDD-Net, to improve detection speed and reduce the model size. The proposed method has a detection accuracy of more than 99% and can identify a minimum defect area of 0.06 mm^2. This means that the CNN-based method is very effective to solve the problem of automatic defect detection and pattern classification for semiconductor silicon wafers. During the research, there are two problems still need promoting:

(i) Data sets. The number of categories in the existing dataset is uneven. Although the data enhancement method in the paper alleviates the problem to some extent, the data capacity of the mechanical damage category is still insufficient. Therefore, establishing wafer defect detection datasets with large amounts of data, wide-type coverage and balanced sample numbers of defect categories becomes a top priority in the field of wafer defect detection.

(ii) The CNN methods researched in this paper are supervised learning models. Data need to be labeled manually. The adjustment of model hyper-parameters also requires a lot of empirical knowledge. Unsupervised transfer learning can apply the knowledge or patterns learned in a certain field or task to different but related fields or problems, which is an exciting research direction.

ACKNOWLEDGMENT

The authors would like to appreciate Tianjin FLY Tech Co., Ltd. for offering samples and for Intelligent Edge Computing Joint LAB of TUST for the testing result analyzing.

REFERENCES

[1] S. Bengtsson, "Semiconductor wafer bonding: A review of interfacial properties and applications," *J. Electron. Mater.*, vol. 21, no. 8, pp. 841–862, 1992.

[2] H. J. Möller, "Chapter two—Wafering of silicon," *Semicond. Semimetals*, vol. 92, pp. 63–109, Apr. 2015.

[3] F. Adly, O. Alhussein, P. D. Yoo, Y. Al-Hammadi, K. Taha, S. Muhaidat, Y.-S. Jeong, U. Lee, and M. Ismail, "Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1267–1276, Dec. 2015.

[4] R. Neubecker, J. E. Hon, "Automatic inspection for surface imperfections: Requirements, potentials and limits," *Proc. SPIE*, vol. 10009, Jun. 2016, Art. no. 1000907.

[5] N. Shankar and Z. Zhong, "Defect detection on semiconductor wafer surfaces," *Microelectron. Eng.*, vol. 77, nos. 3–4, pp. 337–346, Apr. 2005.

[6] C.-L. Tien, Q.-H. Lai, and C.-S. Lin, "Development of optical automatic positioning and wafer defect detection system," *Meas. Sci. Technol.*, vol. 27, no. 2, Feb. 2016, Art. no. 025205.

[7] T.-F. Yao, L. G. Connolly, and M. Cullinan, "Expanded area metrology for tip-based wafer inspection in the nanomanufacturing of electronic devices," *J. Micro/Nanolith. MEMS MOEMS*, vol. 18, no. 3, p. 1, Sep. 2019.

[8] Z. Zhen, "Wafer defects detecting and classifying system based on machine vision," presented at the 8th Int. Conf. Electron. Meas. Instrum., Xi'an, China, Aug. 2007.

[9] D. Kim, P. Kang, S. Cho, H.-J. Lee, and S. Doh, "Machine learning-based novelty detection for faulty wafer detection in semiconductor manufacturing," *Expert Syst. Appl.*, vol. 39, no. 4, pp. 4075–4083, Mar. 2012.

[10] Z.-F. Zhang, Y. Liu, X.-S. Wu, and S.-L. Kan, "Integrated color defect detection method for polysilicon wafers using machine vision," *Adv. Manuf.*, vol. 2, no. 4, pp. 318–326, Dec. 2014.

[11] W. Ho, A. Tay, Y. Zhou, and K. Yang, "*In situ* fault detection of wafer warpage in microlithography," *IEEE Trans. Semicond. Manuf.*, vol. 17, no. 3, pp. 402–407, Aug. 2004.

[12] Y. Sugihara, T. Honda, Y. Urano, M. Watanabe, S. Noguchi, and H. Igarashi, ''Classification and size estimation of wafer defects by using scattered light distribution,'' *Electron. Commun. Jpn.*, vol. 98, no. 6, pp. 36–43, Jun. 2015.

[13] R. Buengener, ''Defect inspection strategies for 14 nm semiconductor technology,'' *Proc. SPIE*, vol. 8466, Oct. 2012, Art. no. 846607.

[14] M. Liukkonen and Y. Hiltunen, ''Recognition of systematic spatial patterns in silicon wafers based on SOM and K-means,'' *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 439–444, 2018.

[15] Q. Tian, S. Xiao, Y. Duan, X. Gao, and W. Zhou, ''Semiconductor wafer surface defect inspection algorithm based on multi-frame differential image summation,'' *Proc. SPIE*, vol. 10819, Nov. 2018, Art. no. 108190J.

[16] A. Vacca, B. Eynon, and S. Yeomans, ''Improving wafer yields at low k1 with advanced photomask defect detection,'' *Solid State Technol.*, vol. 41, no. 6, pp. 185–193, Jun. 1998.

[17] G. Udupa, B. K. A. Ngoi, H. C. F. Goh, and M. N. Yusoff, ''Defect detection in unpolished Si wafers by digital shearography,'' *Meas. Sci. Technol.*, vol. 15, no. 1, pp. 35–43, Jan. 2004.

[18] H. Liu, W. Zhou, Q. Kuang, L. Cao, and B. Gao, ''Defect detection of IC wafer based on spectral subtraction,'' *IEEE Trans. Semicond. Manuf.*, vol. 23, no. 1, pp. 141–147, Feb. 2010.

[19] B. Zhao, X. Li, X. Lu, and Z. Wang, ''A CNN–RNN architecture for multi-label weather recognition,'' *Neurocomputing*, vol. 322, pp. 47–57, Dec. 2018.

[20] Y. Zhang, D. You, X. Gao, N. Zhang, and P. P. Gao, ''Welding defects detection based on deep learning with multiple optical sensors during disk laser welding of thick plates,'' *J. Manuf. Syst.*, vol. 51, pp. 87–94, Apr. 2019.

[21] H. Yous, A. Serir, and S. Yous, ''CNN-based method for blotches and scratches detection in archived videos,'' *J. Vis. Commun. Image Represent.*, vol. 59, pp. 486–500, Feb. 2019.

[22] H. Han, C. Gao, Y. Zhao, S. Liao, L. Tang, and X. Li, ''Polycrystalline silicon wafer defect segmentation based on deep convolutional neural networks,'' *Pattern Recognit. Lett.*, to be published, doi: 10.1016/j.patrec.2018.12.013.

[23] T. Nakazawa and D. V. Kulkarni, ''Wafer map defect pattern classification and image retrieval using convolutional neural network,'' *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018.

[24] K. Kyeong and H. Kim, ''Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks,'' *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 3, pp. 395–402, Aug. 2018.

[25] J. Yu, X. Zheng, and J. Liu, ''Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map,'' *Comput. Ind.*, vol. 109, pp. 121–133, Aug. 2019.

[26] S. Cheon, H. Lee, C. O. Kim, and S. H. Lee, ''Convolutional neural network for wafer surface defect classification and the detection of unknown defect class,'' *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 163–170, May 2019.

[27] T. Nakazawa and D. V. Kulkarni, ''Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing,'' *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 250–256, May 2019.

[28] J. Wang, Z. Yang, J. Zhang, Q. Zhang, and W.-T.-K. Chien, ''AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition,'' *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 310–319, Aug. 2019.

[29] M. Lin, Q. Chen, and S. Yan, ''Network in network,'' 2014, *arXiv:1312.4400*. [Online]. Available: https://arxiv.org/abs/1312.4400

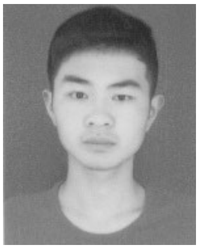
[30] F. Chollet, ''Xception: Deep learning with depthwise separable convolutions,'' 2017, *arXiv:1610.02357*. [Online]. Available: https://arxiv.org/abs/1610.02357

[31] L. Sifre and S. Mallat, ''Rigid-motion scattering for texture classification,'' 2014, *arXiv:1403.1687*. [Online]. Available: https://arxiv.org/abs/1403.1687

[32] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

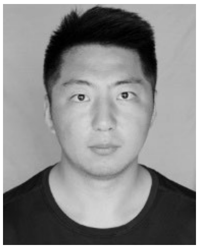
[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

XIAOYAN CHEN received the M.S. degree from the Tianjin University of Science and Technology, in 1999, and the Ph.D. degree in measurement technology and automation devices from Tianjin University, in 2009. From 2009 to 2015, she held a postdoctoral position at Tianjin University. She was invited by RPI, USA, as a Visiting Scholar, from 2009 to 2010, and Kent, U.K., in 2012. She is currently a Professor and an Advisor of Postgraduate and Doctorate students with the Tianjin University of Science and Technology. Her current interests include pattern recognition, biomedical impedance tomography, and advanced measurement technology.



JIANYONG CHEN received the B.S. degree in automation from the Tianjin University of Science and Technology, China, in 2018, where he is currently pursuing the M.S. degree in control science and engineering with the Department of Electronic Information and Automation, under the supervision of Dr. Chen. His current research interests include digital image processing, pattern recognition, and machine learning.

XIAOGUANG HAN received the B.S. and M.S. degrees in mathematics from Harbin Normal University and Harbin Engineering University, Harbin, China, in 2007 and 2009, respectively, and the Ph.D. degree in control science and engineering from Nankai University, Tianjin, China, in 2017. Since 2017, he has been with the Tianjin University of Science and Technology, Tianjin, where he is currently a Lecturer with the College of Electronic Information and Automation. His research interests include the supervisory control of discrete-event systems, formal methods and cyber-physical systems, and Boolean networks.



CHUNDONG ZHAO received the B.S. degree in automation from the Tianjin University of Science and Technology, China, in 2018, where he is currently pursuing the M.S. degree in control science and engineering with the Department of Electronic Information and Automation, under the supervision of Dr. Chen. His current research interests include digital image processing, pattern recognition, and machine learning.

DONGYANG ZHANG received the B.S. degree in automation from the Tianjin University of Science and Technology, China, in 2018, where he is currently pursuing the M.S. degree in control science and engineering with the Department of Electronic Information and Automation, under the supervision of Dr. Chen. His current research interests include pattern recognition and intelligent systems.

KUIFENG ZHU is currently the General Manager of FLY Tech Company, Ltd., Tianjin, China. From 1996 to 2005, he was the purchaser of Daewoo Electronics Company, Ltd., Tianjin. From 2005 to 2010, he was responsible for the supply chain integration of Samsung Aeshang International Logistics Company, Ltd., Tianjin. He founded FLY Tech Company, Ltd., in 2011. His research interests include automation equipment and packing material.

YANJIE SU is currently the Support Center Manager of FLY Tech Company, Ltd., Tianjin, China. From 2007 to 2012, she was with the Sales Department of Sanyuan Sensing Technology Company, Ltd., Qingdao, China. From 2012 to 2015, she was responsible for the Comprehensive Management Department of Keland Automobile Trading Company, Ltd., Tianjin. Her research interests include automation equipment and packing material.