Dirt Classification of Silicon Wafers Based on Deep Learning

Xingxing Li*, Chao Duan, Yan Zhi, Panpan Yin
Department of electronics and information engineering, Guangzhou College of Technology and Business, Foshan, 528138, China.

*Corresponding author: lixingxing@gzgs.edu.cn

Abstract. Deep learning technology is now widely used in the industry, and some irregular blemishes and stains can be solved by deep learning methods. In this paper, the deep learning resnet network is used to classify and judge the dirt that appears on the silicon chip. It is a residual network structure. The mapping relationship between layers is realized through jump connections to ensure that the number of network layers increases. In the process of gradient backpropagation, there will be no gradient disappearance and gradient explosion, and the goal of training deep networks is achieved. This article is to apply the resnet deep neural network on the silicon wafer image to make a certain area on the silicon wafer make the right judgment for the dirty problem.

1. Introduction
As the deep learning vision technology shines in the industry, some difficult problems that have appeared in the industry in the past can be solved. For example, in the industry, there are often defects of indefinite shape, color, size, and texture, such as cracks, white spots, black spots, internal cracks, internal collapse, and dirt. These abnormal flaws will consume a lot of manpower and material resources if they only rely on the human eye to observe one by one. If a part of the human eye judgment task is handed over to the computer with the help of computer vision, it will greatly liberate the labor cost. However, if the defects that appear only rely on the previous manual features, some of the irregular shapes and weak imaging defects are difficult to solve, and traditional algorithms have problems such as low performance, high error detection rate, and noise sensitivity. With the help of deep learning algorithms and relying on a large amount of image data, the image is handed over to the network to automatically learn and extract the deep features of the image, which can make the judgment result more robust.

The method based on deep convolutional neural network has made great progress at present. In 2012, Alex Net [1] proposed by Krizhevsky et al. broke through the record of image classification accuracy in the large-scale visual recognition challenge (ILSRVC). In 2014, Simonyan and Zisserman proposed the VGG series model, which extracted image features through a deep convolutional neural network, and finally used a fully connected layer for classification, which achieved great success in the ImageNet Challenge that year. Subsequently, Google proposed the InceptionNet [2] series of models, which through careful design of the network results to reduce model parameters while improving the expressive ability of the model. In 2015, Kaiming He et al. proposed ResNet, which solved the gradient by introducing a residual module the problem of disappearance greatly increased the depth of the network [3]. In 2016, GAO Huang Et Al. proposed DenseNet [4], which uses Dense Block to multiplex the feature maps of each layer, strengthen the transfer of features in the network, and improve network performance while reducing the amount of network parameters.
The contribution of this paper is to use the classic renset50 network to determine whether there is dirt on the silicon wafer image. Since the shape and size of the dirt image are different, it is necessary to classify the dirt image first, and observe the dirt shape the image can be classified as large dirt, dirt on line marks, and spot dirt. In order to distinguish the dirty and normal samples, it is necessary to subdivide the normal samples, which can be divided into line marks, image chamfers, and normal silicon wafers. Then the data is enhanced for the 6 types of subdivided targets, and then the classification model is trained through the renset50 network, and finally the accuracy of the model is verified on the test set.

2. Principle

2.1. Dirty image data enhancement.

Dirt and flaws often appear in silicon wafer images. These dirt can have no fixed shape and location, and can appear anywhere in the image. By observing the shapes of dirt and normal samples, samples and dirt can be divided into 6 categories. As shown below:

![Figure 1. 6 different forms data set.](image-url)

The figure1 shows 6 different data forms, of which the upper part is a normal image, including ordinary silicon wafer images, silicon wafers with line marks, and silicon wafers with a black background. The bottom half is a dirty image, including large stains, linear stains, and dot stains. The purpose of subdividing dirty and non-stained images into 6 categories instead of simply dividing them into two categories is to allow the deep learning network to learn different types of dirty and non-stained features, so that the network is not easy to confuse the dirty dirt and non-dirt. In the second aspect, in order to improve the distinction between dirty and non-stained images, it is necessary to artificially refine and classify the image types, and finally summarize them based on the categories predicted by the network.

In terms of data enhancement, because the dirty image is a grayscale image, data enhancement on the color channel misleads the original grayscale information of the image. If the shape of the image is enhanced, for example, image distortion and deformation will make normal samples such as line marks become abnormal. At the same time, the dirty image itself is a kind of noise. If some artificial Gaussian noise and salt and pepper noise are added, it will weaken the characteristics of the normal sample. Therefore, the data enhancement methods finally adopted in this article mainly include rotation, mirroring, cropping and scaling, as shown in Figure 2:
The use of data enhancement technology to expand the sample can avoid the phenomenon of sample imbalance and ensure that each of the six types of samples can be roughly the same order of magnitude.

2.2. Resnet residual structure.
For the deep network structure, Kaiming He proposed residual learning to solve the degradation problem [3]. For a stacked layer structure (stacked in several layers) when the input is x, the learned feature is recorded as H(x), and now we hope it can learn the residual F(x) = H(x)-x, in this way, the original learning feature is H(x). The reason for this is that residual learning is easier than direct learning of original features. When the residual is F(x) = 0, the accumulation layer only does the identity mapping at this time, at least the network performance will not decrease, in fact the residual error will not be 0, which will also make the accumulation layer based on the input features learn new features to have better performance. The structure of residual learning is shown in Figure 4. This is similar to a "short circuit" in a circuit, so it is a short-circuit connection.

Why is the residual learning relatively easier? From an intuitive point of view, the residual learning requires less content, because the residuals are generally smaller and the learning difficulty is less. But we can analyze this problem from a mathematical point of view. First of all, the residual unit can be expressed as:

\[
y_i = h(x_i) + F(x_i, W_i)
\]

\[
x_{i+1} = f(y_i)
\]

(1)
Among them, \( x_l \) and \( x_{l+1} \) respectively represent the input and output of the L-th residual unit. Note that each residual unit generally contains a multilayer structure. \( F \) is the residual function, which means the learned residual, and \( h(x_l) = x_l \) means the identity mapping, and \( f \) is the ReLu activation function. Based on the above formula, we find the learning features from shallow \( l \) to deep\( L \).

\[
x_l = x_l + \sum_{i=1}^{L-1} F(x_i, W_i)
\]

We can know that for traditional CNN, the directly stacked network is equivalent to doing affine transformation-non-linear transformation layer by layer, and the affine transformation step is mainly matrix multiplication [3]. So in general, a directly stacked network is equivalent to a multiplicative calculation. In ResNet, compared to the directly stacked network, the nature of calculation has changed from multiplication to addition due to the emergence of shortcuts. The calculation becomes more stable, and the gradient can be propagated lossless. Another residual gradient needs to pass through a layer with weights, and the gradient is not directly transferred. The residual gradients are not so coincidentally all -1, and even if it is relatively small, the existence of 1 will not cause the gradient to disappear. In this way, the attenuation of the gradient is further suppressed, and the calculation of addition improves the stability and ease of training. So the number of layers of the trainable network has also been greatly increased.

2.3. Resnet (Bottleneck) module.

As shown in Figure 4 below, the left image is a very primitive conventional module (Residual block). In actual use, the residual module and the Inception module hope to reduce computational consumption. Therefore, the paper further proposed the "BottleNeck" module. The idea is the same as Inception. By using 1x1 conv to subtly reduce or expand the feature map dimension so that the number of filters of our 3x3 conv is not affected by the external input, that is, the previous layer of input. Naturally, its output will not affect the next module. However, it is designed purely to save calculation time and reduce the time required for the entire model training, and has no effect on the final model accuracy [3].

![Figure 4. BottleNeck module.](image)
The table, for the 18-layer and 34-layer ResNet, the residual learning between the two layers is carried out. When the network is deeper, it is the residual learning between the three layers, and the three-layer convolution the cores are 1x1, 3x3 and 1x1 respectively. It is worth noting that the number of feature maps in the hidden layer is relatively small, and it is 1/4 of the number of output feature maps.

3. Experiment
The experiment in this article uses dirty images enhanced by 6 types of data. The training level of the image and the distribution of the verification set are shown in the figure below.

**Table 1. The distribution of the number of samples in the image training set and test set.**

|                 | Normal | Line mark | Angel | Big stain | Line stain | Point stain |
|-----------------|--------|-----------|-------|-----------|------------|-------------|
| Train dataset   | 1094   | 985       | 1088  | 1011      | 947        | 1129        |
| Test dataset    | 231    | 455       | 440   | 531       | 431        | 742         |

After using the resnet50 network to train for 150 cycles, test the classification accuracy of each category on the test set. The test results are shown in Table 2 below:

**Table 2. Accuracy of the test results.**

|                 | Normal | Line mark | Angel | Big stain | Line stain | Point stain |
|-----------------|--------|-----------|-------|-----------|------------|-------------|
| Accuracy        | 0.97   | 0.94      | 0.98  | 0.96      | 0.93       | 0.97        |

If Normal, Line mark, and Angel are classified as normal samples, and big stain, Line stain, and Point stain are used as dirty samples, the overall classification accuracy of the test is 0.965. From the classification results, it is mainly difficult to distinguish between the line mark sample and the dirty line mark sample. Because the two features are relatively close, some images cannot even be judged by the human eye. By dividing the dirt into 6 categories, and outputting as 2 categories, the network extraction features can be more refined and the classification effect is more significant.

4. Summary
To use deep networks to classify blemish samples such as dirt, you first need to manually subdivide the samples, instead of simply dividing them into two categories, you need to refine the features so that the network can learn better. The training samples are enhanced by data to ensure the balance of the sample distribution. This article uses the resnet50 network for classification and residual learning to solve the degradation problem. The calculation can improve the stability and ease of training, and the number of layers of the network is also can be greatly deepened.

Acknowledgments
This work was financially supported by fund project, that is, Young Talents in Higher Education of Guangdong, China, (No. 2019KQNCX231).

References
[1] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60(6):84-90.
[2] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions [J]. 2014.
[3] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition [J]. 2015.
[4] Huang G, Liu Z, Laurens V D M, et al. Densely Connected Convolutional Networks [J]. 2016.