Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(7): 11821–11846.
DOI: 10.3934/mbe.2023526
Received: 13 March 2023
Revised: 24 April 2023
Accepted: 27 April 2023
Published: 09 May 2023

Research article

A full-flow inspection method based on machine vision to detect wafer surface defects

Naigong Yu[1,2,3*], **Hongzheng Li**[1,2,3] **and Qiao Xu**[1,2,3]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2] Beijing Key Laboratory of Computing Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124, China
[3] Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China

* **Correspondence:** Email: yunaigong@bjut.edu.cn; Tel: +8613051913709.

Abstract: The semiconductor manufacturing industry relies heavily on wafer surface defect detection for yield enhancement. Machine learning and digital image processing technologies have been used in the development of various detection algorithms. However, most wafer surface inspection algorithms are not be applied in industrial environments due to the difficulty in obtaining training samples, high computational requirements, and poor generalization. In order to overcome these difficulties, this paper introduces a full-flow inspection method based on machine vision to detect wafer surface defects. Starting with the die image segmentation stage, where a die segmentation algorithm based on candidate frame fitting and coordinate interpolation is proposed for die sample missing matching segmentation. The method can segment all the dies in the wafer, avoiding the problem of missing dies splitting. After that, in the defect detection stage, we propose a die defect anomaly detection method based on defect feature clustering by region, which can reduce the impact of noise in other regions when extracting defect features in a single region. The experiments show that the proposed inspection method can precisely position and segment die images, and find defective dies with an accuracy of more than 97%. The defect detection method proposed in this paper can be applied to inspect wafer manufacturing.

Keywords: machine vision; wafer surface defect detection; machine learning; image segmentation; feature extraction

1. Introduction

Modern industry relies heavily on chips because they are the brains of contemporary electronics. Chip manufacturing requires multiple processes such as film deposition, etching, polishing, scribing and inversion [1]. As the main material for chip manufacturing, hundreds of chips will be

lithographed on wafers, and any abnormality in the manufacturing process may lead to wafer defects [2]. An individual chip in a wafer is called a die. Among the types of die surface defects are scratches, dots, broken, missing corner, missed welding and dirt dies, as shown in Figure 1. These wafer defects are costly for both electronic equipment suppliers and customers. Therefore, wafer surface defect detection plays a critical role in wafer manufacturing. On the one hand, developing a surface defect inspection process for wafer manufacturing can help avoid the unwanted cost caused by defect dies. On the other hand, it can ensure yield quality, ramp up the manufacturing process and boost enterprises' competitiveness in the market.

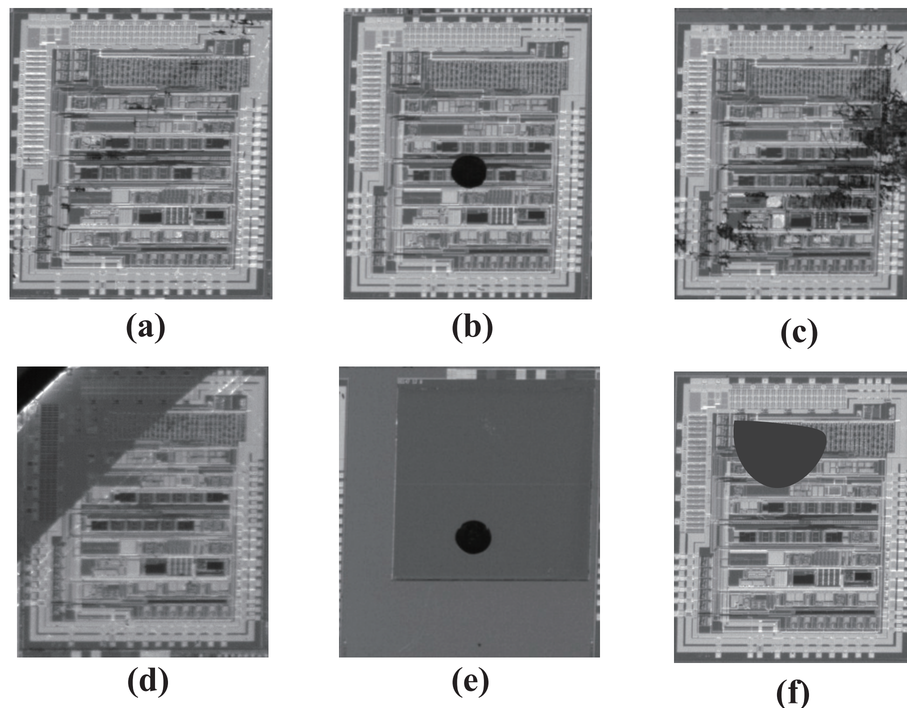


Figure 1. Types of die defects. (a) Scratches die. (b) Dots die. (c) Broken die. (d) Missing corner die. (e) Missed welding. (f) Dirt die.

Previously, wafer surface defect identification primarily relied on manual visual inspection, an inefficient technique [3] due to its expensive cost, time consumption and ineffective for massive production. At present, wafer detection mainly relies on probe detection. This method can only detect the electrical performance of the die on the wafer, and the contact detection method may cause mechanical damage. With the development of industrial cameras, automatic optical inspection (AOI) technology is more and more widely used [4]. Wafer nondestructive inspection technology based on machine vision has been widely used. AOI consists primarily of industrial cameras, light sources, image acquisition cards and PCs. Its workflow can be divided into three steps: acquisition of the target image, identifying the defect and analyzing the defect [5]. Usually, a typical AOI begins with an industrial camera with an illumination system for acquiring the corresponding image. Then, the area to be inspected is located in the acquired image. Finally, the corresponding algorithm is designed to complete the detection of defects in product surface defects.

Although AOI has been studied in wafer surface defect detection, there are still some problems with the existing methods. In the detection step, the existing method only detects whether there are defects

on the wafer surface, and does not locate the defects at the specific die location. That is to say, the existing methods do not focus on how to split the die from the wafer and segment each die one by one, which is an important step and an urgent need [6]. Marking defective dies in the wafer map can alert the factory to remove them in a timely manner. When defective dies are distributed at different locations in the wafer map, they will appear as different defect patterns [7]. However, existing methods do not mark the defective dies one by one in the wafer map.

To solve these problems, this paper proposes a full process detection method from wafer image acquisition to die defect detection. First, wafer images are captured by industrial cameras; second, each die image is segmented from the wafer image; and finally, each die is inspected for defects detection. To the best of our knowledge, this work provides the first wafer surface defect detection method used to locate the die defect location. The main contributions of this paper are as follows.

1) For the problem that the seriously damaged dies cannot be segmented from the wafer map, a die segmentation method based on candidate frame fitting and coordinate interpolation was proposed. This method can accurately segment all the dies from the wafer map and can be used to segment many types of wafer.

2) Aiming at the difficulty of sample acquisition of defect-free dies, a defect detection method based on regional defect feature clustering is proposed, which can complete the defect detection task without relying on standard model dies.

3) A complete solution for the whole process of wafer image detection from acquisition to defective wafer map generation. We designed algorithms for pose correction, image segmentation, image enhancement and defect detection for the inspection process and integrated the task of "acquisition-segmentation-inspection" of wafers.

The remainder of this paper is organized as follows. Section 2 reviews existing related methods. Section 3 details the steps in the wafer surface defect detection process and the required methods. Section 4 describes the construction of the experimental platform and the experimental validation of the proposed detection method. Finally, the work summary and prospective conclusions are given.

2. Related works

For wafer surface defect detection methods, there are many relevant researches in recent years and the existing methods can be divided into two main categories: methods based on image processing and methods based on deep learning. Early wafer surface defect detection methods were mainly based on image processing technology [8, 9]. This method relies on the calculation of the difference between the image to be tested and the reference image, and is greatly affected by the reference image accuracy. Shankar [10] proposes a template matching based method to inspect wafers. Yeh [11] applied wavelet transform to detect wafer appearance defects. Threshold segmentation [12] and principal component analysis [13] are used to detect surface defects of solar dies. Lin [14] and Zhang [15] studied the surface defects of LED packages and the color defects of polysilicon by building a visual inspection system. Saad [16] and Ko [17] used multilayer threshold segmentation and local binary mean segmentation to separate defects such as contamination, splinters and cracks from the background. Although these image-processing methods can detect defects on wafer surfaces, most are designed for specific defects.

Such methods are poorly generalized and cannot be extended enough to detect surface defects in other wafers. Because the wafers are not all the same, each wafer can be different.

In recent years, deep learning has been gradually applied to wafer surface defect detection by virtue of its strong feature extraction capability [18]. Chen [19] combined an improved k-mean clustering algorithm with morphological methods to enhance the robustness of the defect segmentation extraction process against noise. Haddad et al. [20] proposed a three-stage method for wafer surface defect detection, including candidate region generation, defect detection and refinement stages. Yang et al. [21] proposed a quantum classical hybrid model for wafer defect detection Chen [22] proposed a die image defect detection based on ResNet, and the die defect detection accuracy of simple circuit structure could reach 98.57%. Jared [23] classified the chemical composition of defect particles in semiconductor wafers. Classical target detection networks are also used in wafer surface defect detection, such as Faster-RCNN [24] and YOLO v3 [25, 26]. Although deep learning methods have improved the accuracy of detecting a defect, training the model still requires high computing power resources and a long training time. Besides, it requires a substantial amount of labeled data to train the model, and detecting other types of wafers requires the model to be trained again with new labeled samples. In addition, adapting to new iterations of semiconductor products is difficult because of heavy workloads. Unsupervised learning does not require labeled data to perform the task of recognition and classification [27, 28]. Cluster analysis has been widely used as a common unsupervised learning approach [29, 30]. For many types of wafer surface defect detection problems, we try to use clustering methods to detect defective dies on the surface.

When the above methods detect defects on the wafer surface, they do not detect the dies one by one. In other words, the above methods do not segment the dies to build a complete defect detection process. Bourgeat et al. [31] proposed a segmentation algorithm for semiconductor wafer images generated by optical inspection tools. Yang [32] proposed affine iterative closest approximation algorithm based on spatial feature point guidance for wafer segmentation and inspect each die. The above method completes the segmentation of the wafer by analysing the characteristics of the die surface. However, the surface of a severely damaged die no longer has distinctive features. The above method does not explain how to segment severely damaged dies. Our method segments each die and marks the defective die in the wafer map. The analysis of defective wafer maps helps the factory to identify the cause of defects in time [33].

3. Methodology

This section will elaborate on the inspection flow and corresponding implementation method designed in this paper for wafer surface defect detection. The overall detection strategy is described as follows. First, set up the wafer surface automatic optical inspection platform and complete the image acquisition of the wafer surface. Then, for the original wafer images acquired by the platform, we use the contents described in Sections 3.1–3.4 to complete pose correction, sample segmentation, image enhancement and defect detection in turn. Finally, the detected defective dies are marked on the wafer map to generate a defective wafer map. The specific detection process is shown in Figure 2.

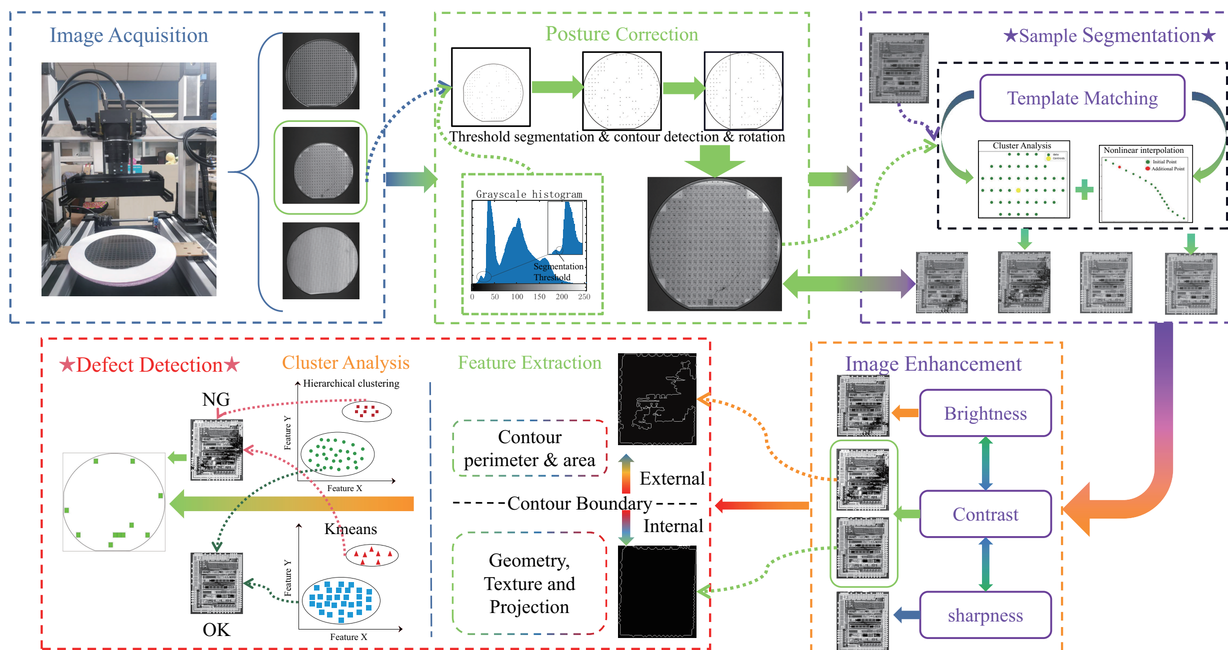


Figure 2. Wafer surface defect detection process.

3.1. Wafer image pose correction

In the production inspection process, the wafers on the conveyor belt cannot be placed at the same angle, so the wafer image captured by the image acquisition system will have different deflection angles. Therefore, it is necessary to perform positional correction on the acquired image and extract the wafers from the corrected image.

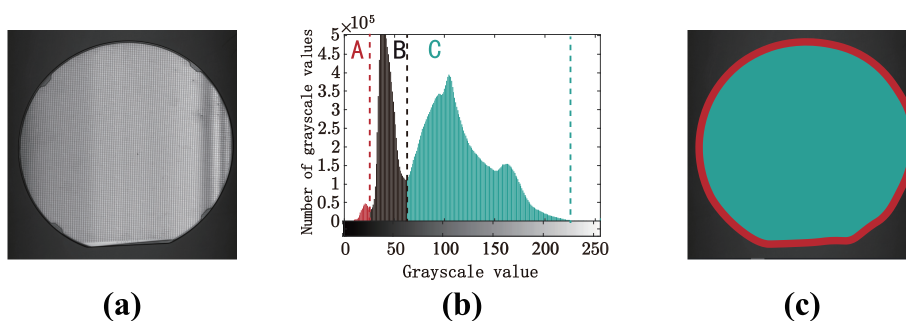


Figure 3. Determining the segmentation threshold. (a) Original wafer image. (b) Grayscale histogram of image (a). (c) The wafer image is divided into three areas.

Image pose correction is also known as the geometric transformation of an image. It maps the coordinate position of an image to a new coordinate position. The geometric transformation merely rearranges the pixel information in the image which does not alter the pixel values. We use threshold segmentation and morphological methods to process the outer contour part of the wafer. This is because

the wafer outer profile section already contains information on the overall wafer size and deflection angle. To obtain the outer contour of the wafer using threshold segmentation for the acquired image, first, we need to determine the segmentation threshold. As shown in Figure 3(a),the outer contour part of the wafer presents a small gray value and a certain amount of gray in the grayscale image. We can determine the segmentation threshold by analyzing the image grayscale histogram. The wafer image is shown in Figure 3(a), and its gray histogram is shown in Figure 3(b). Each color in Figure 3(b) corresponds to the area in the wafer image, as shown in Figure 3(c). In order to segment the wafer and the background, the value between the A and B parts can be selected as the optimal segmentation threshold.

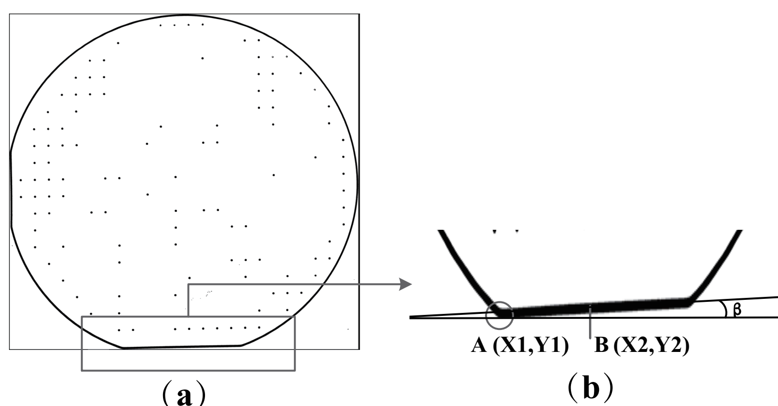


Figure 4. Determination of image rotation angle. (a) Wafer image after threshold segmentation. (b) Local enlargement of image.

Threshold segmentation is performed on the wafer image, and the segmented image is shown in Figure 4. The lowest part of the wafer outline image is a straight line, and the slope of the line can be detected to determine whether the image needs to be rotated for correction and to calculate the rotation angle. First, we search for the coordinates of the point with a gray value 0 from the last row upward and record the point as $(X1, Y1)$, as shown in Figure 4, point A. Then, we calculate the coordinates of the middle column of the image and record it as $Y2$. After that, we search for the row with a gray value of 0 from the bottom upward in column $Y2$ and record the point as $(X2, Y2)$, as shown in Figure 3, point B. Finally, the rotation angle β is calculated from $(X2 - X1)/(Y2 - Y1)$. Rotation correction is applied to the image based on the calculated rotation angle. The wafer image threshold segmentation and rotation correction results are shown in Figure 5.

3.2. *A die segmentation method based on candidate frame fitting and coordinate interpolation*

Wafer segmentation is the process of splitting each die from the wafer map. We segment the wafer map to obtain each die, which has the following advantages. 1) We can check each die more closely for defects. 2) Defective dies can be marked in the wafer map, and the factory can remove the defective dies in time. Existing segmentation methods can be divided into two categories: those based on template matching and those based on feature analysis. Image matching is the most common method used in wafer segmentation. It is the process of finding the region in an image most similar to the template image. The feature-based methods segment the wafers by analyzing the features in the die, such as right angles and straight lines. However, the above method does not enable the severely

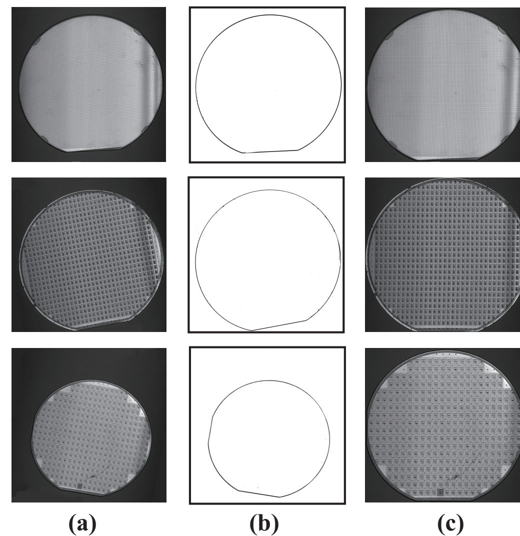


Figure 5. Wafer image pose correction results. (a) Original wafer image. (b) Wafer image after threshold segmentation. (c) Image after positional correction.

damaged dies to be split off from the wafer map. This is because the surface of a badly damaged die has lost its distinctive features. We propose a wafer segmentation method based on candidate frame fitting and coordinate interpolation. First, We use a template matching-based method to complete the segmentation of the majority of dies. Then, the severely damaged dies are identified by analyzing the characteristic relationships of the coordinates of the already segmented dies.

1) Template matching and coordinate fitting. This paper uses the normalized squared difference matching method to complete the template matching work. The principle is as follows: the template image is denoted by T, the target image is denoted by I and the confidence result is denoted by R. Before the image matching work, we also need to set the similarity threshold, τ. In the template matching process, if the calculated $R > \tau$, the die image is considered to exist in the region. The matching confidence formula $R(x, y)$ is calculated in Eq (3.1).

$$R(x, y) = \frac{\sum_{x',y'} T(x', y')I(x + x', y + y')}{\sqrt{\sum_{x',y'} T(x', y')^2 \sum_{x',y'} I(x + x', y + y')^2}} \tag{3.1}$$

In Eq (1), $T(x', y')$is the template image and $I(x, y)$ is the image to be matched. In this paper, the template similarity threshold is set to 0.6.

As shown in Figure 6, since duplicate redundancy detection boxes exist in the template matching results, we need to filter those based on the initial matching results and complete the best match result extraction. Currently, the above cases are often solved by non-maximum suppression (NMS) methods in image processing. However the NMS method has some disadvantages. 1) Threshold values must be set manually. 2) A missed detection may occur if the target sample is present in the overlapping area.

We collect the coordinates of the upper left corner of all candidate boxes, as shown in Figure 6, and record them as, $(x_1, y_1), (x_1, y_1), ..., (x_n, y_n)$. Use Eq (3.2) to calculate the geometric centre point (x_c, y_c) as the most suitable coordinate point. The calculation of (x_c, y_c) is shown in Figure 6(b).

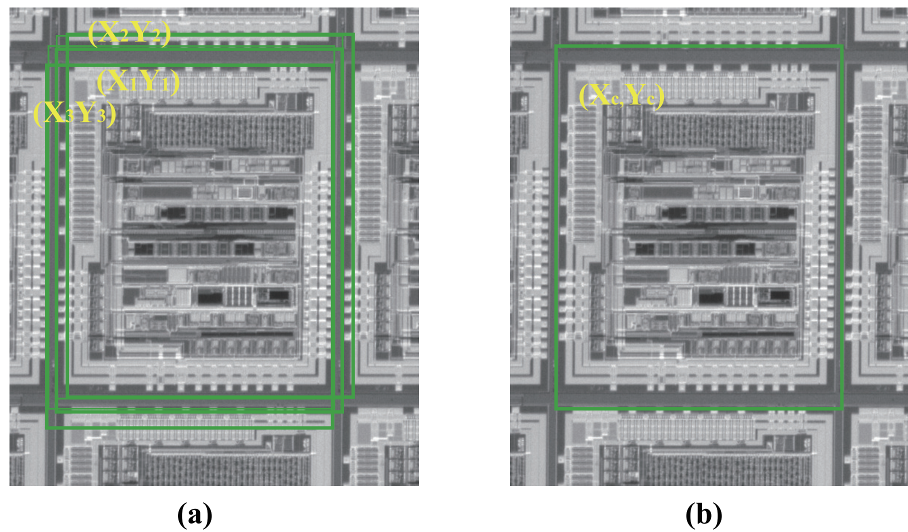


Figure 6. Template matching based wafer segmentation. (a) Results of template matching. (b) Calculation of (x_c, y_c).

$$(x_c, y_c) = \frac{1}{|n|} \sum_{i=1}^{n} (x_i, y_i) \tag{3.2}$$

2) Wafer segmentation method based on die coordinate relations. For normal dies, they can be detected successfully when the τ is set to 0.6. However, for damaged dies, as shown in Figure 7, the detection segmentation cannot be completed by the template matching method even when the τ is on a very low setting.

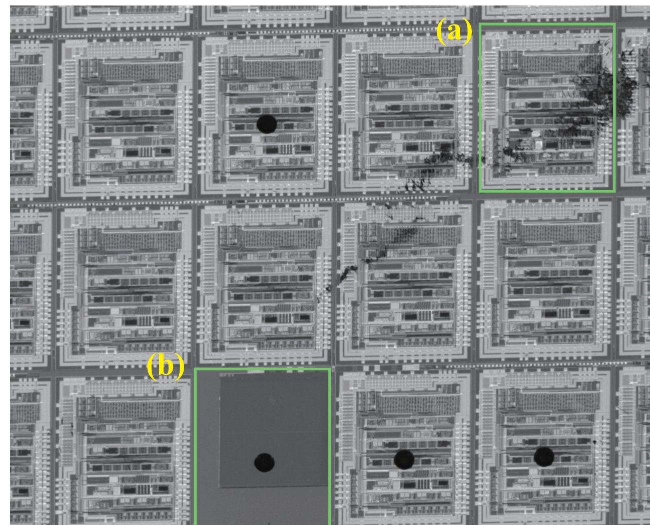


Figure 7. Damaged dies. (a) Die with severe surface fragmentation, (b) Missing to weld.

We segment the damaged dies by analyzing the relationship between the co-ordinate points. As shown in Figure 8(a), template-matching based methods have obtained the coordinates of some of the dies. The upper left-hand coordinate point of the die is denoted $p_i = (x_i, y_i)$, as shown in Figure 8(b).

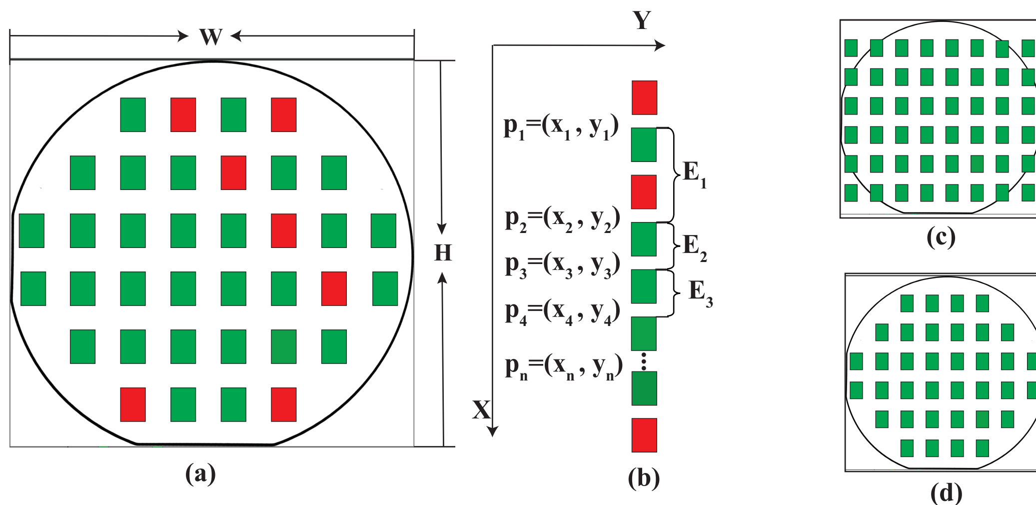


Figure 8. Damaged die detection. (a) Wafer segmentation results based on template matching. (b) Dies in the same column. (c) Possible grains in the wafer map. (d) The final result of wafer segmentation. The red square represent damaged dies and green square represent normal dies.

The set of coordinate points of the same column is calculated using Eq (3.3) and is denoted as P.

$$\begin{cases} P = \{p_1, p_2, ..., p_n\} \\ X = (x_1, x_2, ..., x_n) \end{cases} \tag{3.3}$$

where X represents all the X-coordinates of the same column of dies.

E_i^x is the difference between adjacent x_i and is calculated using Eq (3.4).

$$E_i^x = x_{i+1} - x_i \quad i \in \{0, 1, 2, ..., n-1\}. \tag{3.4}$$

As shown in Figure 8(b), when there are damaged dies, E_1 is significantly larger than E_2 and E_3. We can set a threshold value for E_i^x to detect the presence of damaged dies. If E_i^x is greater than the set threshold, there is a damage die. In order to reduce the human factor in the threshold setting and improve the anti-interference of the detection algorithm, we use the mean and variance in statistics to complete the threshold setting. We denote the mean of E_i^x as μ and the variance as σ. The 3-sigma algorithm assumes that in a normal distribution, when the data lies in $(\mu - 3\sigma, \mu + 3\sigma)$, then it is considered normal data. When the data value is greater than $\mu + 3\sigma$, the confidence level of the data decreases and it is regarded as a discrete value. So, we calculate the mean and variance of E_i^x using Eq (3.5).

$$\begin{cases} \mu = \frac{1}{n}\sum_{i=1}^{n} E_i^x \\ \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(E_i^x - \mu\right)^2} \end{cases} \tag{3.5}$$

where μ and σ are the mean and variance of E^x respectively.

Equation (3.6) holds indicating that there are damaged dies between x_{i+1} and x_i. We can use Eq (3.7) to calculate the number of damaged dies.

$$\forall E_i^x \in E^x, \quad E_i^x > u + 3\sigma \tag{3.6}$$

$$N_j = [(x_{i+1} - x_i)/E_{min}^x] \tag{3.7}$$

where N_j is the number of damaged dies. E_{min}^x is the minimum of E^x.

Equation (3.8) is used to calculate the x-coordinate $x_j^{'}$ of the damaged dies.

$$x_j^{'} = \sum_{j=0}^{N_j} x_i + j * E_{min}^x \tag{3.8}$$

where x_i is in Eq (3.7).

The least square method of Eq (3.9) can be used to fit the data and calculate the missing coordinates.

$$\min f(x) = \sum_{i=0}^{n} [y_i - f(x_i, w_i)] \quad n = 1 \tag{3.9}$$

where$f(x) = w_0 + w_1 x$.

The result of Eq (3.8) is carried over into Eq (3.10) to calculate the y-coordinate $y_j^{'}$ of the damaged dies.

$$y_j^{'} = f(x_j^{'}, w_0, w_1) \tag{3.10}$$

As shown in Figure 8(b), we also need to calculate the coordinates of the damaged dies above point P_1 and below point P_n. Equations (3.11) and (3.12) are used to calculate the number of damaged dies above point P_1 and below P_n, respectively.

$$N_k = \lfloor (x_1)/E_{min}^x \rfloor \tag{3.11}$$

where N_k is the number of damaged dies above point P_1.

$$N_l = \lfloor (H - x_n)/E_{min}^x \rfloor - 1 \tag{3.12}$$

where N_l is the number of damaged dies below point P_n. H is the height of the wafer image, as shown in Figure 8(a).

Equations (3.13) and (3.14) are used to calculate the x-coordinate of the damaged dies above point P_1 and the x-coordinate below point P_n.

$$x_k^{'} = \sum_{k=0}^{N_k} x_1 - k * E_{min}^x \tag{3.13}$$

where $x_k^{'}$ is the x-coordinate of the damaged dies above point P_1.

$$x_l^{'} = \sum_{l=0}^{N_l} x_n + l * E_{min}^x \tag{3.14}$$

where $x_l^{'}$ is the x-coordinate of the damaged dies below point P_n.

Equations (3.15) and (3.16) calculate the Y-coordinate of the damaged dies.

$$y_k' = f(x_k', w_0, w_1) \tag{3.15}$$

where y_k' is the y-coordinate of the damaged dies above point P_1.

$$y_l' = f(x_l', w_0, w_1) \tag{3.16}$$

where y_l' is the y-coordinate of the damaged dies below point P_n.

As shown in Figure 8(c), the set consisting of (x_i, y_i), (x_j', y_j'), (x_k', y_k'), (x_l', y_l') is denoted $S(p_i')$.

$$S(p_i') = \left\{(x_1, y_1), (x_2, y_2), ...(x_n, y_n), (x_1', y_1'), ..., (x_{D_{die}}', y_{D_{die}}')\right\} \tag{3.17}$$

where $S(p_i'))$ is the coordinates of all dies. p_i' is a die.

The outer contour of the wafer can be obtained by threshold segmentation and contour detection. The dies outside the wafer contour should be removed and the result is shown in Figure 8(d). The method proposed in this paper is not affected by artificially set parameters and has higher segmentation accuracy. It can be well applied to the task of multi-target sample segmentation in wafer images. The results of template matching of dies by the proposed method are shown in Figure 9.

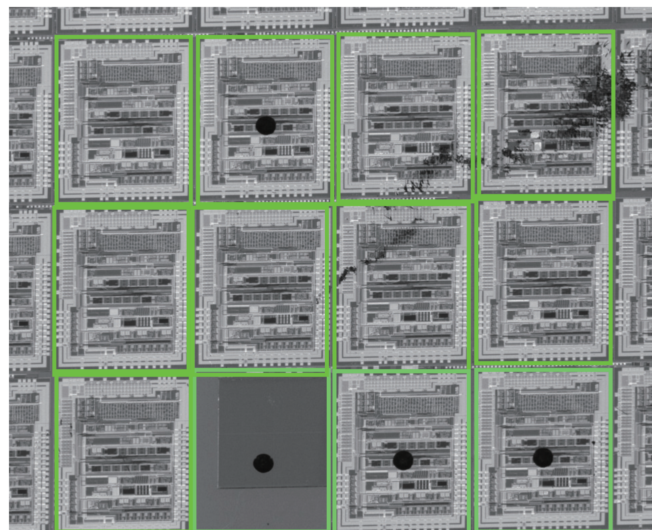


Figure 9. Results of wafer segmentation.

3.3. Die images enhancement

During the image acquisition process, there are shadow areas in the captured images because the light does not radiate evenly on the wafer. We hope that the brightness of the die image can be kept stable. Because too bright or too dark die image may be judged as a defect. Image enhancement can effectively solve this problem. Image enhancement can not only improve the brightness of the image, but also enhance its details, which is convenient for later defect detection.

First, all die images are converted to HSV space, and the H component of each die image is extracted to generate a one-dimensional array. The data are sorted from smallest to largest, and the first 30% of

the array is taken as the darker image, the middle 40–70% as the standard brightness image and the last 70–100% as the brighter image. Figure 10(a) shows the distribution of brightness values in the original dies' image. The data from the standard brightness images are summed and averaged and noted as V_{mean}. The other individual image brightness values are recorded as V_{sample}. Enhancement of brighter and darker images is done using Eq (3.18). The values of each parameter of the formula are shown in Table 1, and the enhancement effect is shown in Figure 11. Figure 10(b) shows the distribution of brightness values in the dies' image after brightness enhancement.

$$out = in0 * (1.0 - \alpha) + in1 * \alpha \tag{3.18}$$

where, out is the output image. $in0$ is the new image with the same image size and the same number of channels as the image to be enhanced. $in1$ is the image to be enhanced, and α is the enhancement factor.

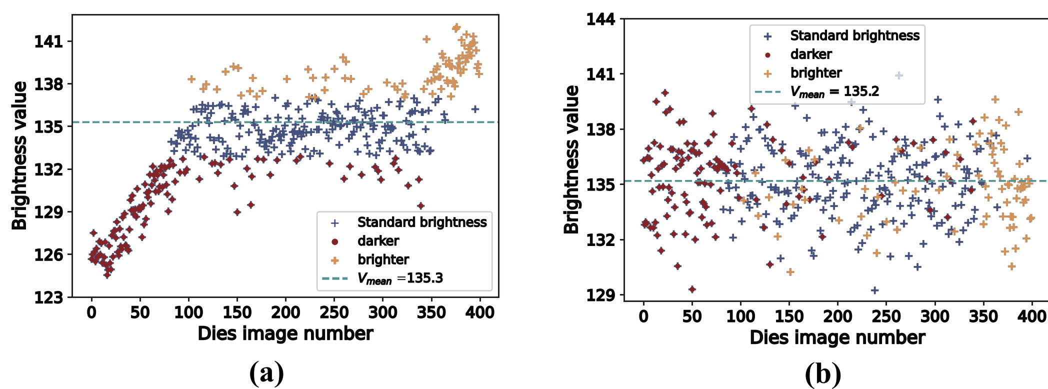


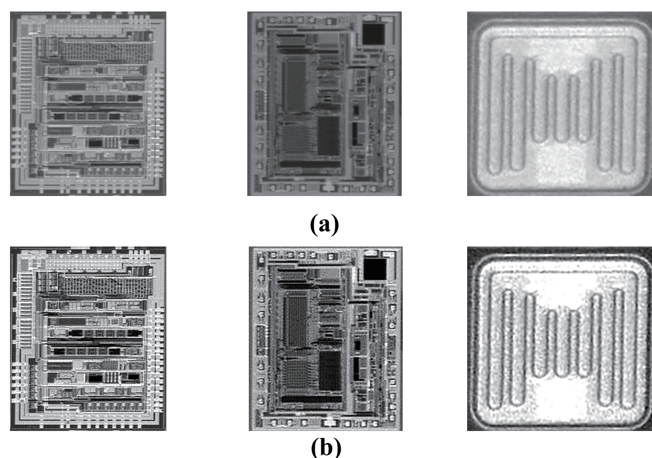
Figure 10. Image brightness enhancement. (a) Distribution of brightness values of unenhanced images. (b) Distribution of brightness values of enhanced images.

Table 1. Image enhancement parameters.

Enhancement parameter	in1	in0	α
Brightness	Original image	0	V_{mean}/V_{sample}
Contrast	Original image	Average value of in1	1.5
Sharpening	Original image	Smooth filtering of in1	3.0

3.4. Die surface defect detection

The micro defect size of the dies on the wafer is small. Due to the high dimension of image data, there is more redundant feature information, which inhibits the expression ability of useful features. The existing wafer surface defect detection algorithms are supervised learning methods, which rely on a large number of labeled samples to complete the detection task. Additionally, the existing methods cannot locate the dies with specific defects. Therefore, we propose a defect detection method based on regional defect feature clustering. This method first detects the maximum external contour of the die image, and then divides the entire image into the area within the contour and the area outside the contour, as shown in Figure 12.



(a)

(b)

Figure 11. Image enhancement results. (a) Unenhanced images. (b) Enhanced image.

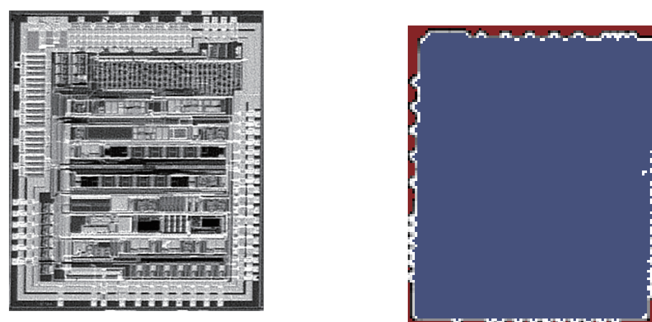


Figure 12. Contour detection. (a) Die images. (b) Dividing the die image into different areas.

When we detect the area within the outline, we need to ignore the details beyond the outline. Because of the irregular features of the gap between the dies on the wafer, it will affect the extraction of the dies' features, as shown in Figure 13. Then, we will extract geometric, projective and textural features for the areas within the outline. Followed by extracting the area feature from the contour. Finally, the extracted feature sets were clustered and classified to screen out the defect dies.

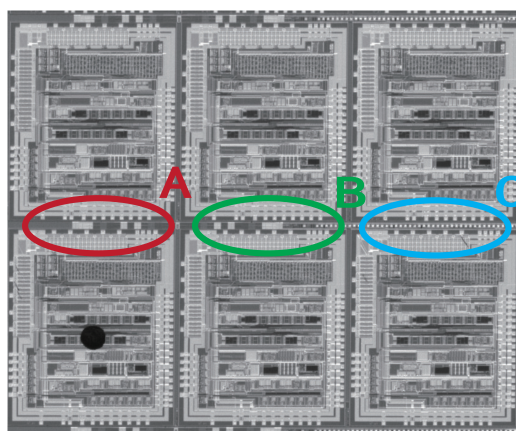


Figure 13. Gap between dies.

1) Internal feature extraction. In the extraction of internal features of die contours, the largest outer contour of the die image is searched first. After several experiments, the outline image produced by thresholding the initial image twice and adding filtering and morphological processing was the clearest.

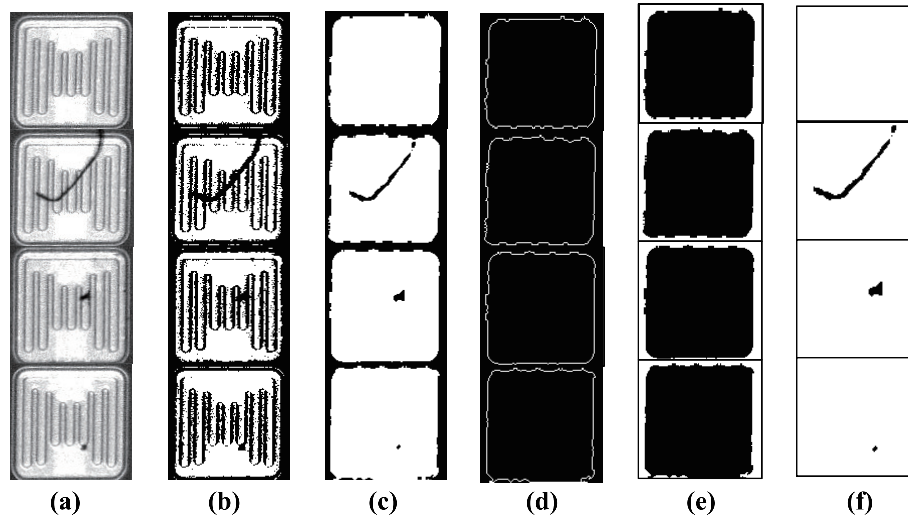


Figure 14. Internal feature extraction process. (a) The original grayscale image of the die. (b) and (c) are the results after two threshold segmentations, respectively. (d) The result of the outer contour search of the die boundary. (e) The area outside the largest contour in the die image with grayscale value 255. (f) The result of image fusion.

The die image is noted as $D(p_i)$. p_i is the pixel in the image. We use Eqs (3.19) and (3.20) to perform image thresholding on die images. The image segmentation results are shown in Figure 14(b),(c).

$$D_{s1}(p_i) = \phi(M(D), D) \tag{3.19}$$

$$D_{s2}(p_i) = \phi(M(D_{s1}), D_{s1}) \tag{3.20}$$

where D_{s1} is the result of thresholding the image D. D_{s2} is the result of thresholding the image D_{s1}. ϕ represents a function for image thresholding, the first parameter is the segmentation threshold and the second parameter is the image. $M(D)$ is the average of the pixels of image D.

Then, the contour search finds the maximum outer contour, as shown in Figure 14(d). Filling the area outside the largest contour in the die image with grayscale value 25 5and the result is noted as $D_f(p_i)$, as shown in Figure 14(e). This allows ignoring the defective features in the area outside the contour. Then, we use Eq (3.21) to fuse Figure 14(c),(e) to get the new Figure 14(f) .

$$D^{'}(p_i) = D_{s2} + D_f \tag{3.21}$$

Extraction of geometric, textural and projection features of the processed die images.

$$(\alpha_i, \beta_i, \gamma_i) = \psi_1(D^{'}(p_i)) \tag{3.22}$$

where α_i, β_i, γ_i for projection, geometric and textural features of image $D^{'}(p_i)$. ψ_1 is a function of the feature extraction of image $D^{'}(p_i)$.

2) External feature extraction. As shown in Figure 15, die defects such as missing corner defect and broken defect mainly damage the overall integrity of the die. The feature reflected in the image where the largest outer contour of the wafer is damaged. The outer contour of the defective die no longer has integrity and exhibits inward shrinkage. This feature can be used as a defect feature for defect detection. We use Eq (3.23) to calculate the area in green of Figure 15(d) as the defect feature.

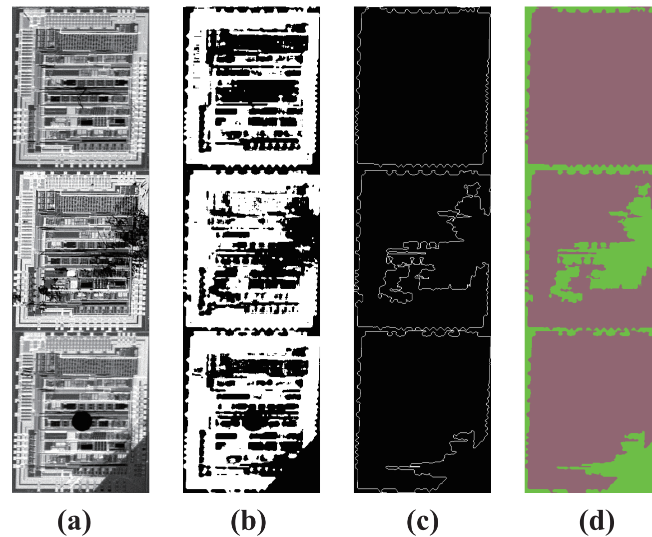


Figure 15. Boundary feature extraction process. (a) the original grayscale image of the die. (b) Image (a) after thresholding segmentation. (c) The outer contour of image (a). (d) Image outer contour anomaly detection.

$$\zeta_i = \psi_2(D^{'}(p_i)) \tag{3.23}$$

where ζ_i is the area feature of image $D^{'}(p_i)$. ψ_2 is a function that calculates the area outside the largest contour of image $D^{'}(p_i)$.

3) Defect detection method based on feature clustering. The features of defect-free dies and defective dies are different. In 1) and 2), we extracted several features, including projection, geometry, texture and the area out of contours. Clustering analysis is an unsupervised learning-based feature analysis method that we can use to distinguish different features. There are hundreds of dies on a wafer image, and most of them are defect-free dies. We set the clustering center to 2. The defective features will be clustered together, and the number will be small. Defect-free features will be clustered together and in larger numbers. We use Eq (3.24) to calculate the clustering results.

$$(C_1, C_2) = F_{cluster}(\alpha_i, \beta_i, \gamma_i, \zeta_i) \tag{3.24}$$

where (C_1, C_2) is the result of feature clustering, representing two clusters. $F_{cluster}$ is the clustering function.

Based on actual production experience, damaged dies are in the minority in wafer images. We use Eq (3.25) to calculate who has fewer grains in C1, C2. The result of the calculation is the defective dies, noted as C_{defect}.

$$C_{defect} = \underset{C_i \in (C_1, C_2)}{arg\min}(F_n(C_i)) \tag{3.25}$$

where C_{defect} is the defective dies. $F_n(C_i)$ is the number of dies in C_i.

The area of the contours is one-dimensional data and can be visualized for presentation. We cluster the area out of the contours and the results are shown in Figure 16.

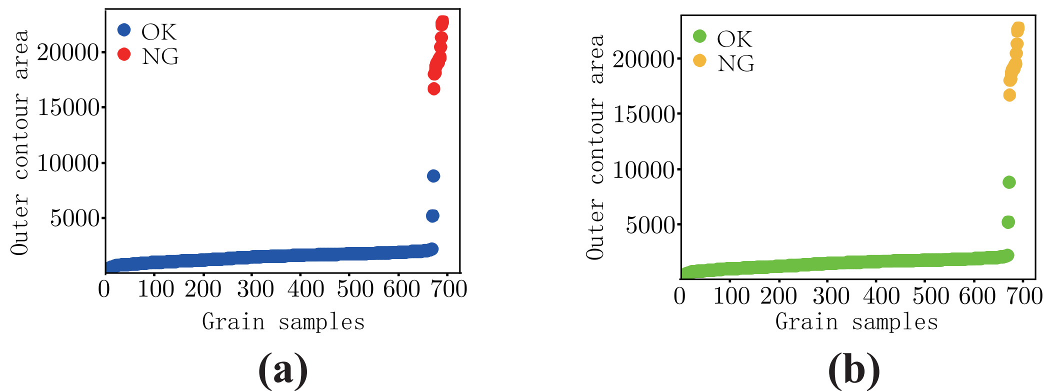


Figure 16. Cluster analysis to filter discrete samples. (a) Kmeans clustering. (b) Hierarchical clustering. OK stands for defect-free characteristics. NG stands for defective characteristics.

4. Experimental results and analysis

4.1. Experiment preparation

The wafer image acquisition system consists of line array camera(LA-CM-16K05A), motion stage, light sources and wafers, as shown in Figure 17. The methods in this paper are written in Python and uses the Opencv library. All experiments were run on AMD Ryzen 7 5800H 3.2GHz CPU and 16GB RAM. The used experimental data collected by the self-built wafer image acquisition system.

Table 2. SPECIFIC PARAMETERS FOR LA-CM-16K05A.

Parameters	Value
Resolution(pixels)	16,384
Pixel Size (μm)	3.52×3.52
Pixel Fill Factor (%)	100
Line Rate(kHz)	48
Bit Depth(bit)	8
Sensor type	CMOS
Camera size(mm)	76.0(W) × 76.0(H) × 36.7(D)

4.2. Die segmentation results based on candidate frame fitting and coordinate interpolation

1) Results of wafer image pose correction. We have quantified the results of the wafer positional correction in Figure 5. The quantified results are shown in Table 3.

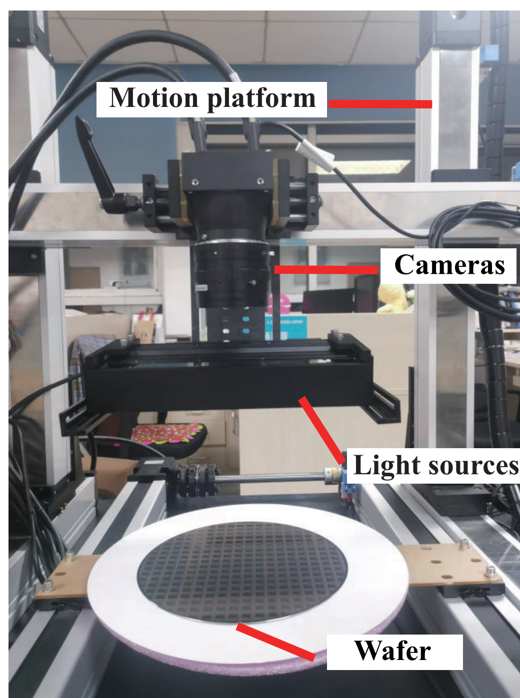


Figure 17. Wafer Image Acquisition Systems.

Table 3. Results of wafer image pose correction.

Angles	R_1	R_2	R_3
Actual angle	3.15°	10.02°	9.43°
Ours	3.12°	10.04°	9.45°

*Note: R_1, R_2 and R_3 are the three types of wafers in Figure 5.

As the table shows, our calculated rotation angle is close to the actual angle. The calculation error is less than 0.05°.

2) Identification of dies with surface feature damage. In this experiment, we use the template matching method to perform wafer segmentation on the wafer in Figure 3(a). Dies with severe damage cannot be identified by template matching, as shown in Figure 18. When there are damaged dies, the results of making differences between adjacent coordinate points are shown in Figure 19.

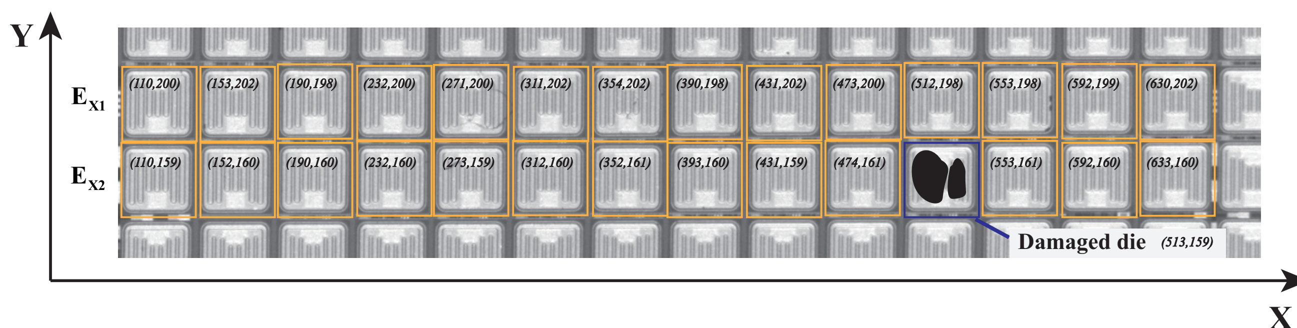


Figure 18. Local enlargement of a wafer with damaged dies.

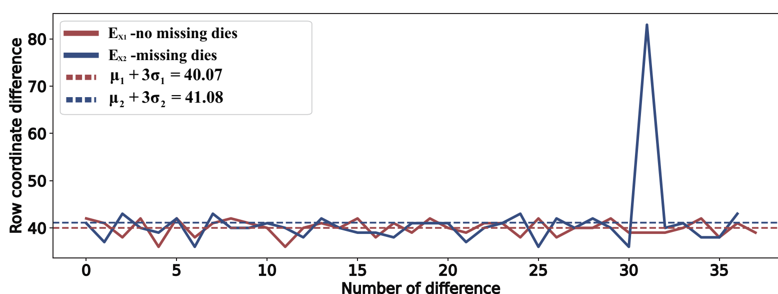


Figure 19. Missing coordinates detection. μ_1 and σ_1 are the mean and variance of E_{x1}. μ_2 and σ_2 are the mean and variance of E_{x2}. E_{x1} and E_{x2} represent the difference between adjacent X coordinates.

As seen in Figure 19, there is a large peak in E_{x2}, which represents a missing coordinate found between adjacent coordinate points. We calculated the coordinates of the damaged die using the method proposed in Section 3.2, as shown in Figure 18. This result shows that our method can identify in time the dies that have not been split off from the wafer map by analysing the relationship between the die coordinates.

3) Comparison and analysis of wafer segmentation methods. In this experiment, We compared the proposed method with Template Matching, LSD (Line Segment Detector) and AICP-FP methods [32]. We performed wafer segmentation experiments using three different types of wafers, and the number of missing dies was used as an evaluation index. The three types of wafers are shown in Figure 20 and we have used A,B,C to name the wafer types. The number of dies in the three types of wafers A,B,C is 339, 692, 1528, respectively. The quantification results are shown in Table 4.

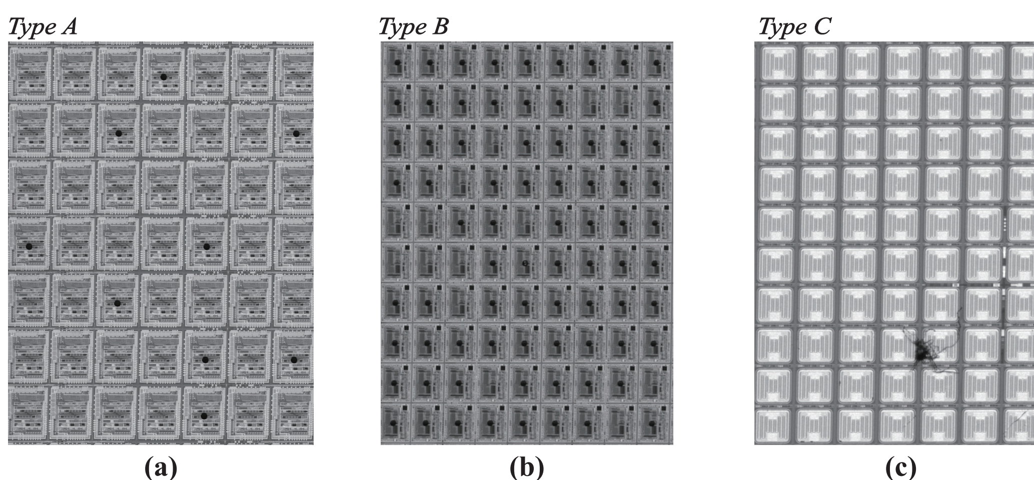


Figure 20. (a), (b) and (c) are the three types of wafers that are used for wafer segmentation.

As shown in Table 4, our method allows all dies to be segmented from different types of wafers. The template matching-based method relies on the quality of the template. When the surface of the die is dirty and broken, as shown in Figure 21, these dies will appear to differ significantly from the standard template. Therefore, the template matching method is unable to segment these dies. LSD and AICP-FP have similar drawbacks to template-based matching methods. They cannot identify dies that have been severely damaged on the surface. AICP-FP is more capable of extracting features than LSD.

Table 4. Wafer image segmentation miss detection results.

Methods	The number of missing dies		
	Type A	Type B	Type C
Template Matching	13	10	10
LSD	15	8	9
AICP-FP*	10	2	3
Our Method	0	0	0

*Note: stands for state-of-the-art method.
A, B and C are the three types of wafers in Figure 20.

When the damaged surface area is small, the AICP-FP can segment the die by analyzing the lines and corners of the die edges.

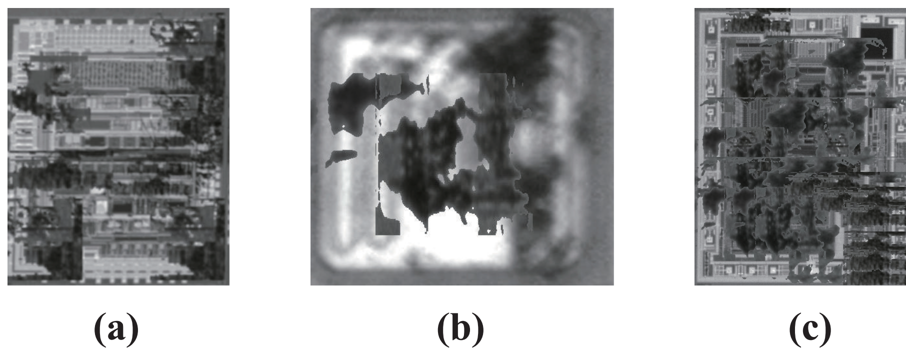


Figure 21. (a), (b) and (c) are dies with severe surface damage.

We have also analyzed the results of wafer segmentation methods when there are different degrees of damage to die surface features. We use Gaussian noise to blur the features of the die image.

$$I_\sigma = I * G_\sigma \tag{4.1}$$

where I_σ is the image with noise added. $*$ is for convolution. I is the raw image.

G_σ is the Gaussian convolution kernel, defined as:

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{4.2}$$

where σ is the standard deviation, $x^2 + y^2$ is the Gaussian fuzzy radius.

We randomly Gaussian blurred 100 dies in wafer A, wafer B and wafer C, respectively. When the standard deviation in the Gaussian convolution kernel is changed, the die image results are shown in Figure 22. Different wafer segmentation methods are then used to segment wafers A, B and C. The different methods are evaluated using the evaluation metrics of Eq (4.3). The quantification results are shown in Table 5.

$$AC_{Seg} = \frac{T_{seg}}{ALL_{seg}} \tag{4.3}$$

where T_{seg} is the Gaussian blurred die that was identified. ALL_{seg} is the total number of Gaussian blurred dies.

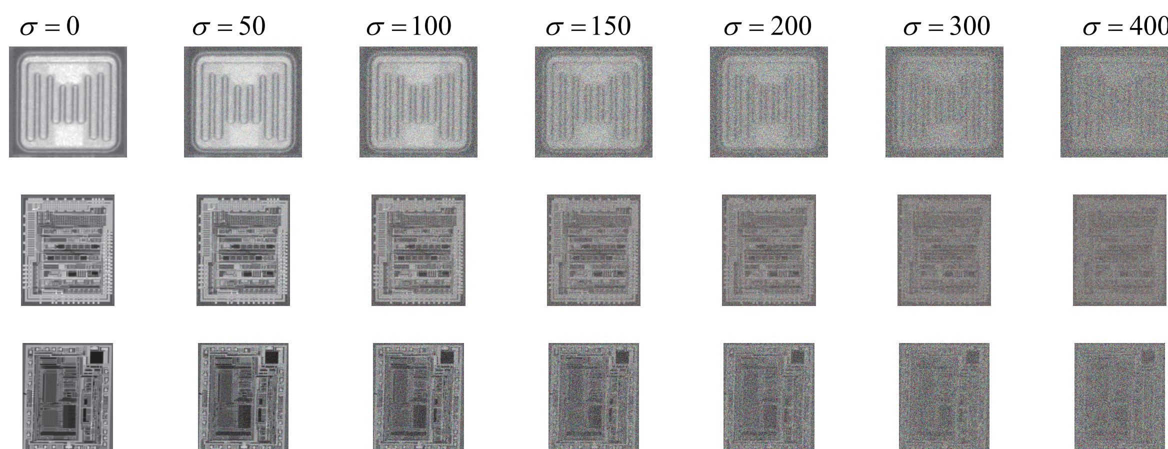


Figure 22. Die images with different Gaussian noise added.

Table 5. Wafer segmentation results for different σ values.

Method	AC_{Seg}					
	σ = 50	σ = 100	σ = 150	σ = 200	σ = 300	σ = 400
Template Matching	1.00	0.98	0.83	0.62	0.21	0.00
LSD	1.00	1.00	0.91	0.73	0.32	0.00
AICP-FP*	1.00	1.00	0.98	0.81	0.52	0.00
Our Method	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.00

The larger the value of σ, the more blurred the features of the dies in Figure 22. When σ is bigger than 150, Template Matching, LSD and AICP-FP cannot identify all the noisy dies in the wafer map. This is because the features on the die surface have been largely lost. When σ is greater than 400, all methods cannot identify the noisy dies. Our method first uses template matching to identify the dies on the wafer surface. Then, the coordinates of the unidentified dies are calculated based on the coordinate relationships of the identified dies. So our method can identify more noisy dies.

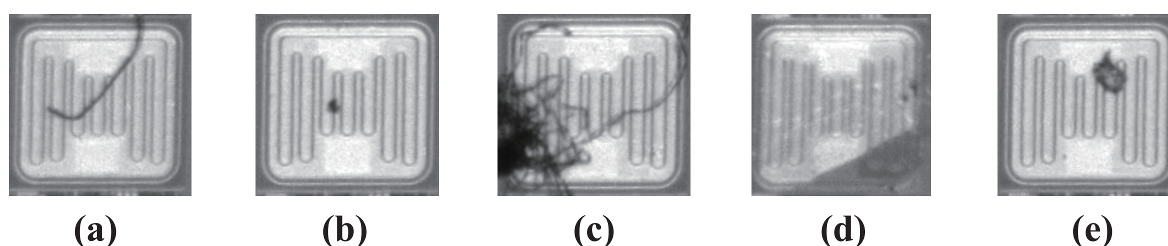


Figure 23. Die images with different defects. (a) Scratches die. (b) Dots die. (c) Broken die. (d) Missing corner.(e) Dire die.

In addition, we used wafer segmentation methods to identify dies with different defects, as shown in Figure 23. The results of the inspection are shown in Table 6. As shown in Table 6, only our method can accomplish the segmentation of all defect types of dies. Template Matching, LSD and AICP-FP are all based on analyzing features on the die surface to achieve wafer segmentation. These methods cannot segment all dies when the surface features are damaged.

Table 6. Results of the identification of dies with different defect types.

Methods	Scratches dies	Dots die	Broken die	Missing corner	Missed welding	Dirt die
Template Matching	✓	✓	×	×	×	×
LSD	✓	✓	×	×	×	×
AICP-FP*	✓	✓	×	×	×	×
Our Method	✓	✓	✓	✓	✓	✓

*Note: ✓means that all grains of this type can be identified. × means that this type of grain can be partially identified.
* stands for state-of-the-art method.

4.3. Analysis of die defect inspection results

1) Die defect detection results and analysis. We use the method proposed in Section 3.4 for defect detection of dies. The dies in Figure 20 were used as experimental data. Accuracy, Recall and Precision were used to evaluate the effectiveness of those methods. The detection results of our methods are shown in Table 7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.5}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.6}$$

TP (True Positive): No defective sample and also judged as a defect-free sample. FN (False Negative): No defective sample but judged as a defective sample. FP (False Positive): Defective sample but judged as a defect-free sample. TN (True Negative): Defective sample and also judged as a defective sample.

Table 7. Die defect detection results.

Wafer Type	Total number of dies	TP	FN	TN	FP	Accuracy	Recall	Precision
A	339	313	6	18	2	0.976	0.981	0.994
B	692	642	9	38	3	0.983	0.986	0.995
C	1528	1400	45	75	8	0.965	0.969	0.994

In order to verify the effectiveness of the proposed method, we performed defect detection on different types of dies. The accuracy can reach more than 96%, which can meet the practical requirements of wafer surface defect detection. However, the accuracy rate of C-type wafers in the inspection experiment is lower. This is due to the manual determination of minor defects on the die surface is a normal sample, while the algorithm determines minor defective dies as defective samples.

Image enhancement can enhance defective features in an image. We add experiments on the effect of image enhancement of defect detection results.

Table 8. Die defect detection results.

Image enhancement	A	B	C
Unenhanced	0.932	0.940	0.955
Enhanced	0.976	0.983	0.965

As shown in Table 8, image enhancement will improve the accuracy of die defects. This is because image enhancement will make defect features in the die image easier to detect.

Table 9. Detection process time (ms).

Wafer Type	Single die pixels	Minimum time	Maximum time	Average time	Time spent on defect detection only
A	246×290	25.82	48.66	28.97	4.058
B	174×208	14.90	23.42	18.09	3.560
C	40×40	3.10	4.48	3.50	0.587

We also did experiments on the time spent on defect detection. The inspection was repeated 20 epochs for each type of wafer independently. The total time from acquisition to completion of the inspection was recorded, and the time spent for individual die inspection was calculated. Minimum time, Maximum time and Average time are the maximum, minimum and average values of the time spent in the 20 epochs respectively. The inspection time statistics are shown in Table 9. Our method takes about 18ms to complete wafer acquisition, die segmentation, image enhancement and defect detection. The time consumed increases as the number of image pixels increases. For comparison with other methods, we also tested the time-consuming time in the defect detection phase. Both the time spent in the whole phase and the time spent in the defect detection phase can meet the industrial inspection needs.

2) Comparison of defect detection methods. In the defect detection experiments, template matching [4], Fourier transform [34] and Deep learning(SH-DNN) [35] are used for comparison with our method. We compared the accuracy, F1-score and inspection time of the different methods for die defect detection. The quantification results are shown in Table 10.

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{4.7}$$

Table 10. Quantitative results of wafer defect detection by different methods.

Methods	Accuracy	F1-score	Single die pixels	Time-consuming(ms)
Template matching	0.925	0.94	-	-
Fourier transform	0.908	0.95	-	-
SH-DNN*	**0.985**	**0.995**	192×192	5.000
Our method	0.975	0.978	246×290	**4.058**

*Note: stands for state-of-the-art method.

Most of the defects in the image frequency domain are a high-frequency part, the Fourier transform-based method can transform the image to the frequency domain for analysis. It can observe the high-frequency part of the image and then completely detect large-area defects such as scratches and broken. However, this method is not suitable for point defects and small soiled defects where the high-frequency component is not obvious. The detection method based on template matching also has some disadvantages. It is susceptible to factors such as the selected high-standard template and the rotation and stretching of the image to be detected, which eventually affects the detection results. As shown in Table 10, SH-DNN achieved the best accuracy and F1-score. Deep learning has better feature extraction capability, and SH-DNN can extract more discriminative defect features in die images. However, deep learning requires powerful computational resources and higher time consumption for defect detection. Although the accuracy of SH-DNN is better than our method, the

time consumption of our method is shorter. When detecting images with more pixels, we use 1 ms less time than SH-DNN.

3) Marking defective dies in the wafer map. In contrast to other methods, we have designed a complete wafer defect detection method. Our method can do everything from image acquisition to die segmentation to defect detection. Especially, our method detects defective dies and relabels them in the wafer map. Defective dies will show defective patterns in the wafer map. Analyzing the defect pattern helps to complete the detection of the cause of the defect. Figure 24 shows the results of our detection of defective dies.

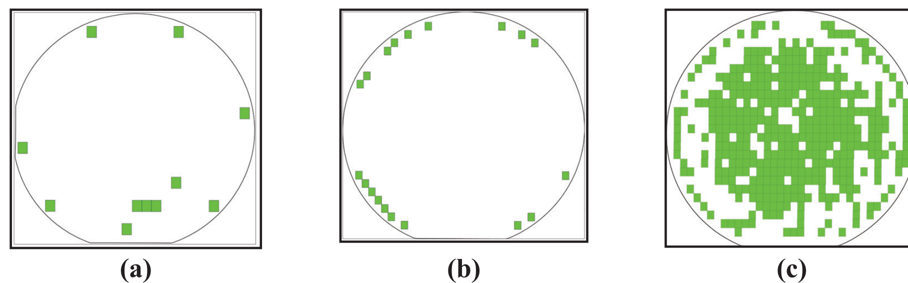


Figure 24. Marking of defective dies. The green squares in the images of (a), (b) and (c) represent defective dies.

5. Conclusions

For the problems of wafer surface defect detection, this paper proposes a full-flow inspection method based on machine vision. The inspection process consists of four parts: wafer image pose correction, wafer segmentation, image enhancement and defect detection. To be able to segment all the dies from the wafer map, we propose a die segmentation algorithm based on candidate frame fitting and coordinate interpolation. Unlike other methods, our method can still segment dies from the wafer surface when the surface features are severely damaged. In addition, we propose a die defect anomaly detection method based on defect feature clustering by region. More importantly, our proposed method does not require an existing labeled sample, which greatly saves manual sample marking time.

We provide a wealth of experiments on the proposed method. The results show that our method can perform wafer segmentation and die-by-die defect detection. The defect detection accuracy is above 97%, which meets the actual production requirements. Especially we also mark the defective dies individually in the wafer map and generate a defective wafer map. The defect wafer map represents the defect pattern and can be used to analyze the specific cause of the defect. In the future, we expect to improve the existing defect detection methods to achieve better accuracy rates.

Acknowledgments

This research was funded by Beijing Municipal Education Commission and Beijing Natural Science Foundation (No. KZ202010005004)

Conflict of interest

The authors declare there is no conflict of interest.

References

1. N. Yu, Q. Xu, H. Wang, J. Lin, Wafer bin map inspection based on densenet, *J. Cent. South Univ.*, **28** (2021), 2436–2450. https://doi.org/10.1007/s11771-021-4778-7
2. L. Alam, N. Kehtarnavaz, A survey of detection methods for die attachment and wire bonding defects in integrated circuit manufacturing, *IEEE Access*, **10** (2022), 83826–83840. https://doi.org/10.1109/ACCESS.2022.3197624
3. Y. Fu, X. Ma, H. Zhou, Automatic detection of multi-crossing crack defects in multi-crystalline solar cells based on machine vision, *Mach. Vision Appl.*, **32** (2021), 1–14. https://doi.org/10.1007/s00138-021-01183-9
4. Q. Hu, K. Hao, B. Wei, H. Li, An efficient solder joint defects method for 3d point clouds with double-flow region attention network, *Adv. Eng. Inform.*, **52** (2022), 101608. https://doi.org/10.1016/j.aei.2022.101608
5. W. Dai, A. Mujeeb, M. Erdt, A. Sourin, Soldering defect detection in automatic optical inspection, *Adv. Eng. Inform.*, **43** (2022), 101004. https://doi.org/10.1016/j.aei.2019.101004
6. T. Kim, K. Behdinan, Advances in machine learning and deep learning applications towards wafer map defect recognition and classification: a review, *J. Intell. Manuf.*, (2022), 1–33. https://doi.org/10.1007/s10845-022-01994-1
7. K. C. Cheng, L. L. Chen, J. Li, K. S. Li, N. C. Tsai, S. Wang, et al., Machine learning-based detection method for wafer test induced defects, *IEEE Trans. Semicond. Manuf.*, **34** (2021), 161–167. https://doi.org/10.1109/TSM.2021.3065405
8. J. Wang, Z. Yu, Z. Duan, G. Lu, A sub-region one-to-one mapping (som) detection algorithm for glass passivation parts wafer surface low-contrast texture defects, *Multimedia Tools Appl.*, **80** (2021), 28879–28896. https://doi.org/10.1007/s11042-021-11084-8
9. K. S. Li, P. Y. Liao, K. C. Cheng, L. L. Chen, S. Wang, A. Y. Huang, et al., Hidden wafer scratch defects projection for diagnosis and quality enhancement, *IEEE Trans. Semicond. Manuf.*, **34** (2020), 9–16. https://doi.org/10.1109/TSM.2020.3040998
10. N. Shankar, Z. Zhong, Defect detection on semiconductor wafer surfaces, Microelectron. Eng., **77** (2005), 337–346. https://doi.org/10.1016/j.mee.2004.12.003
11. C. Yeh, F. Wu, W. Ji, C. Huang, A wavelet-based approach in detecting visual defects on semiconductor wafer dies, *IEEE Trans. Semicond. Manuf.*, **23** (2010), 284–292. https://doi.org/10.1109/TSM.2010.2046108

12. Y. Chiou, J. Liu, Y. Liang, Micro crack detection of multi-crystalline silicon solar wafer using machine vision techniques, *Sensor Rev.*, **31** (2011), 154–165. https://doi.org/10.1108/02602281111110013

13. M. Yao, J. Li, X. Wang, Solar cells surface defects detection using rpca method, *Chin. J. Comput.*, **36** (2013), 1943–1952. https://doi.org/10.3724/SP.J.1016.2013.01943

14. H. Lin, S. Chiu, Flaw detection of domed surfaces in led packages by machine vision system, *Expert Syst. Appl.*, **38** (2011), 15208–15216. https://doi.org/10.1016/j.eswa.2011.05.080

15. Z. Zhang, Y. Liu, X. Wu, S. Kan, Integrated color defect detection method for polysilicon wafers using machine vision, *Adv. Manuf.*, **2** (2014), 318–326. https://doi.org/10.1007/S40436-014-0095-9

16. N. Saad, A. Ahmad, H. Saleh, A. Hasan, Automatic semiconductor wafer image segmentation for defect detection using multilevel thresholding, in *MATEC Web of Conferences*, EDP Sciences, 2016. https://doi.org/10.1051/matecconf/20167801103

17. J. Ko, J. Rheem, Defect detection of polycrystalline solar wafers using local binary mean, *Int. J. Adv. Manuf. Technol.*, **82** (2016), 1753–1764. https://doi.org/10.1007/s00170-015-7498-z

18. Z. Wang, X. Liu, Z. He, L. Su, X. Lu, Intelligent detection of flip chip with the scanning acoustic microscopy and the general regression neural network, *Microelectron. Eng.*, **217** (2019), 111127. https://doi.org/10.1016/j.mee.2019.111127

19. X. Chen, C. Zhao, J. Chen, D. Zhang, K. Zhu, Y. Su, K-means clustering with morphological filtering for silicon wafer grain defect detection, in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, **1** (2020), 1251–1255. https://doi.org/10.1109/ITNEC48623.2020.9084726

20. B. M. Haddad, S. F. Dodge, L. J. Karam, N. S. Patel, M. W. Braun, Locally adaptive statistical background modeling with deep learning-based false positive rejection for defect detection in semiconductor units, *IEEE Trans. Semicond. Manuf.*, **33** (2020), 357–372. https://doi.org/10.1109/TSM.2020.2998441

21. Y. Yang, M. Sun, Semiconductor defect detection by hybrid classical-quantum deep learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2022), 2323–2332. https://doi.org/10.1109/CVPR52688.2022.00236

22. H. Chen, Automated detection and classification of defective and abnormal dies in wafer images, *Appl. Sci.*, **10** (2020), 3423. https://doi.org/10.3390/app10103423

23. J. O'Leary, K. Sawlani, A. Mesbah, Deep learning for classification of the chemical composition of particle defects on semiconductor wafers, *IEEE Trans. Semicond. Manuf.*, **33** (2020), 72–85. https://doi.org/10.1109/TSM.2019.2963656

24. N. Yu, Q. Xu, H. Wang, Wafer defect pattern recognition and analysis based on convolutional neural network, *IEEE Trans. Semicond. Manuf.*, **32** (2019), 566–573. https://doi.org/10.1109/TSM.2019.2937793

25. P. P. Shinde, P. P. Pai, S. P. Adiga, Wafer defect localization and classification using deep learning techniques, *IEEE Access*, **10** (2022), 39969–39974. https://doi.org/10.1109/ACCESS.2022.3166512

26. S. Chen, C. Kang, D. Perng, Detecting and measuring defects in wafer die using gan and yolov3, *Appl. Sci.*, **10** (2020), 8725. https://doi.org/10.3390/app10238725

27. Y. Yi, S. Lai, W. Wang, S. Li, R. Zhang, Y. Luo, et al., Sdnmf: Semisupervised discriminative nonnegative matrix factorization for feature learning, *Int. J. Intell. Syst.*, **37** (2022), 11547–11581. https://doi.org/10.1002/int.23054

28. Y. Yi, J. Wang, W. Zhou, C. Zheng, J. Kong, S. Qiao, Non-negative matrix factorization with locality constrained adaptive graph, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2019), 427–441. https://doi.org/10.1109/TCSVT.2019.2892971

29. X. Yang, G. Lin, Y. Liu, F. Nie, L. Lin, Fast spectral embedded clustering based on structured graph learning for large-scale hyperspectral image, *IEEE Geosci. Remote Sens. Lette.*, **19** (2022), 1–5. https://doi.org/10.1016/j.ins.2023.03.035

30. L. He, N. Ray, Y. Guan, H. Zhang, Fast large-scale spectral clustering via explicit feature mapping, *IEEE Trans. Cybern.*, **49** (2018), 1058–1071. https://doi.org/10.1109/TCYB.2018.2794998

31. P. Bourgeat, F. Meriaudeau, K. W. Tobin, P. Gorria, Content based segmentation of patterned wafers, *J. Electr. Imaging*, **13** (2004), 428–435. https://doi.org/10.1117/1.1762518

32. J. Yang, Y. Xu, H. Rong, S. Du, H. Zhang, A method for wafer defect detection using spatial feature points guided affine iterative closest point algorithm, *IEEE Access*, **8** (2020), 79056–79068. https://doi.org/10.1109/ACCESS.2020.2990535

33. S. Wang, Z. Zhong, Y. Zhao, L. Zuo, A variational autoencoder enhanced deep learning model for wafer defect imbalanced classification, *IEEE Trans. Compon., Packag., Manuf. Technol.*, **11** (2021), 2055–2060. https://doi.org/10.1109/TCPMT.2021.3126083

34. W. Li, D. Tsai, Automatic saw-mark detection in multicrystalline solar wafer images, *Sol. Energy Mater. Sol. Cells*, **95** (2011), 2206–2220. https://doi.org/10.1016/j.solmat.2011.03.025

35. T. Schlosser, M. Friedrich, F. Beuth, D. Kowerko, Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks, *J. Intell. Manuf.*, **33** (2022), 1099–1123. https://doi.org/10.1007/s10845-021-01906-9

AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)