



# SMT defect classification by feature extraction region optimization and machine learning

Ji-Deok Song<sup>1</sup> · Young-Gyu Kim<sup>1</sup> · Tae-Hyoung Park<sup>1</sup>

Received: 1 June 2018 / Accepted: 11 November 2018 / Published online: 17 November 2018  
© Springer-Verlag London Ltd., part of Springer Nature 2018

## Abstract

In this paper, we propose a solder joint defect type classification method for automatic optical inspection machines in the manufacturing system of printed circuit boards. The inspection procedure for the solder joint defect type classification consists of an offline stage, which sets the optimal feature extraction region, and an online stage which classifies a defect type. In the offline stage, we use an optimization technique, namely, the genetic algorithm, to optimize the feature extraction region. In this stage, the optimal feature extraction region for defect type classification is constructed automatically. In the online stage, feature extraction regions are used to segment the solder joint image after component image acquirement. We then extract various color features from the segmented feature extraction regions. Next, we use support vector machine, which is one of the machine learning model's method to classify the solder joint defect type. To evaluate the performance of the proposed method, ten types of solder joint defects were used in an experiment. The experimental results verified the effectiveness of the method in terms of the recognition rate, and its convenience.

**Keywords** Automated optical inspection · Classification · Genetic algorithm · Optimization · Printed circuit board · Solder joint · Support vector machine

## 1 Introduction

The development of the electronics industry seems to have incurred a necessary decrease in the size of mobile electronic devices such as mobile phones, tablet PCs, smart watches, and head-mounted displays. Accordingly, components mounted on a printed circuit board (PCB) are also being miniaturized and the board density is increasing. In addition, defect inspections have become more important for the surface-mount technology (SMT) process, where components are mounted onto a PCB. The traditional manual method using the naked eye cannot guarantee the inspection quality or reliability for miniaturized high-density PCBs. According to some estimates, the defect rate of SMT assembly is about 5%, and the defect of solder joint can account for up to 80% of all SMT assembly defects [1]. This problem was addressed by applying an automated optical inspection (AOI) to the SMT process. An AOI

system inspects the defects of mounted components by obtaining images from an optical apparatus such as a camera.

There are two main methods for inspecting and classifying SMT defects. One is a modeling-based method, and the other method is based on a feature extraction region. The modeling-based method [2–5] creates a defect model through a feature map analysis [2], statistical modeling [3], or a Gaussian mixture model [4] for defect images of each component, and then inspects and classifies the defects. However, this method makes it difficult to analyze the details of defect images of components and requires a long inspection time.

The feature extraction region-based method [6–18] designates regions for feature extraction in defect images to classify the defect types. With this method, after features such as the average brightness and highlight values are extracted from a feature extraction region, machine learning techniques such as support vector machine (SVM) and multi-layer perceptron (MLP) are applied to these features to classify the defects. Compared to the modeling-based method, defects can be analyzed in detail and a high level of accuracy is achieved. Accordingly, this method is utilized for SMT defect type classification. The representative feature extraction region-based methods include H. W. Xie's [6] method using AdaBoost and

✉ Tae-Hyoung Park  
taehpark@cbnu.ac.kr

<sup>1</sup> Department of Control & Robot Eng., Chungbuk National University, Cheongju, South Korea

a decision tree, H. Wu's [7] method using a Bayes classifier and an SVM, and W. Hao's [8] application of a genetic algorithm and backpropagation network. Every feature extraction region-based method obtains component images from the RGB illumination system and sets a feature extraction region for the component images acquired. Features are extracted from the region and defects are classified. Some studies have used wavelet of the image for SMD defect detection and classification [17, 18]. Each of these methods was verified for its recognition rate and speed during the experiments.

However, the existing methods commonly use a feature extraction region, which was set manually by an operator [14–16]. Hundreds of component types are currently used in industrial sites. If the features and sizes of such numerous components are considered to set the feature extraction region manually, it will significantly degrade the efficiency of the SMT process. In addition, such a manual method is available for classifying only about five defect types, which is far below the actual number of defect types occurring in the SMT process. Thus, it is impossible to efficiently classify the defect types.

In this study, focusing on a determination of the feature extraction regions, we propose an SMT defect type classification method based on the optimization of the feature extraction regions. A genetic algorithm (GA) is used to set an optimal feature extraction region for all defect types to be classified, and color features are extracted from the feature extraction region. An SVM then classifies the defect types using the extracted features as inputs.

The remainder of this paper is organized as follows. Section 2 describes an illumination system used for obtaining component images for defect type classification and the SMT defect types that are to be classified. Section 3 describes the existing defect type classification methods, which adopt a manual setting of the feature extraction regions, and their results. Section 4 presents an optimization of the feature extraction regions, as proposed in the present study. Section 5 reports and analyzes the experimental results for defect type classification using the proposed method. Finally, Section 6 summarizes the characteristics of the proposed method and presents several conclusions.

## 2 Problem definition

### 2.1 Illumination system

In this study, the component images used for defect type classification were obtained using an RGB illumination system, as shown in Fig. 1 [6–14]. The RGB illumination system includes three LED lights of different colors (red, green, and blue). Because these lights were installed at different angles, each light was reflected on a different surface of the solder

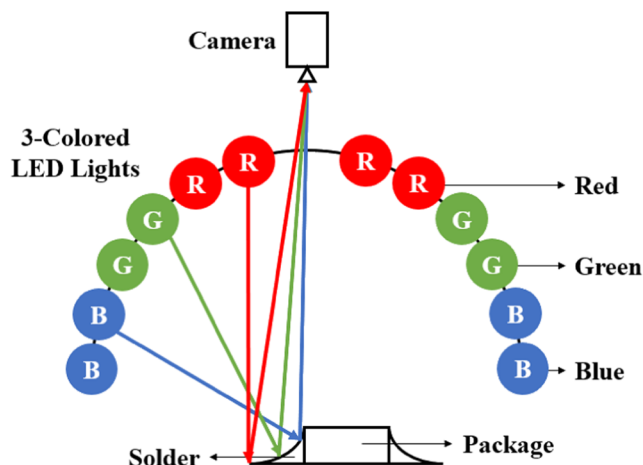


Fig. 1 RGB illumination system

joint. The red light was reflected on a flat surface, the green light on a gentle sloped surface, and the blue light on a steep, sloped surface. As shown in Fig. 2, the component images acquired using such an illumination system showed a red, green, and blue color distribution, and each color indicated different height data of the solder joint. In this way, three-dimensional height information of the components can be secured in two-dimensional images.

### 2.2 Defect type

We classified ten defect types occurring during the SMT process. These include a package defect determined based on the package direction and location, a solder defect based on the solder distribution and quantity, and a complex defect with features of both package and solder defect types. Table 1 shows component images for each defect type, which were obtained using the RGB illumination system.

In the case of a package defect type, if a different component than the target is positioned, it is a “Wrong” defect. If no component is inserted and only solder is present, it is a “Missing” defect. In addition, a rotation defect indicates the rotation of a component exceeding a specific angle; “Tombstone” defect occurs when one side of a component is normally soldered but the other side is not, thereby making the component erect; a “Manhattan” defect refers to a component that is soldered so as to be laterally erect; a shift defect indicates a horizontally slanted component.

In the case of a solder defect type, “No-solder” defect indicates a case in which a component is normally positioned but one solder pad is not soldered at all, and “Exceed” defect occurs when one solder pad is so excessively soldered that the polarity is covered. Finally, “Pseudo” defect, which belongs to a complex defect type, refers to a case in which a component is soldered to completely float and be laterally slanted.

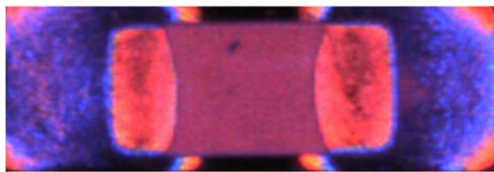


Fig. 2 Image acquired using RGB illumination system

### 3 Defect type classification

#### 3.1 Feature extraction region segmentation

A component image obtained using the RGB illumination system includes two types of region according to the color distribution: a package region and a solder pad region. The color distribution of the package region varies depending on the package defect type, whereas that of the solder pad region changes according to the solder defect type. The existing methods segment a component image into one package region and two solder pad regions, as shown in Fig. 3a, and thus a total of three regions are used for feature extraction [6–8]. Moreover, each package region and solder pad region are further divided into three regions. Consequently, as shown in Fig. 3b, a total of nine feature extraction regions are used [14]. To identify the effect of the segmentation of the feature extraction region on the defect type classification performance, we added another segmentation method using 15 feature

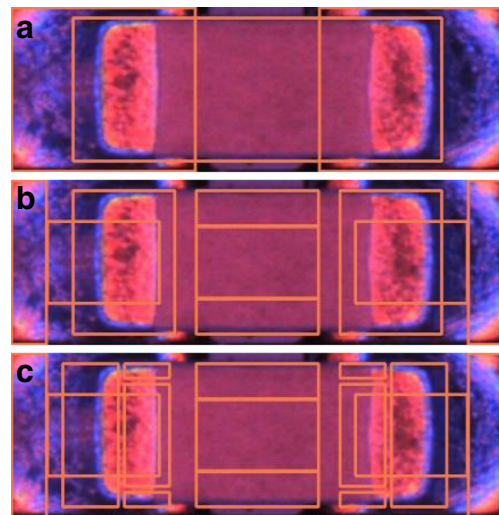




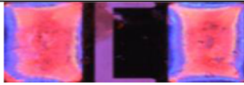

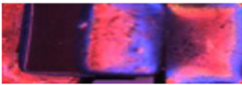





Fig. 3 Feature extraction region segmentation method using a 3-regions, b 9-regions, and c 15-regions

extraction regions, as illustrated in Fig. 3c, and conducted a comparative experiment.

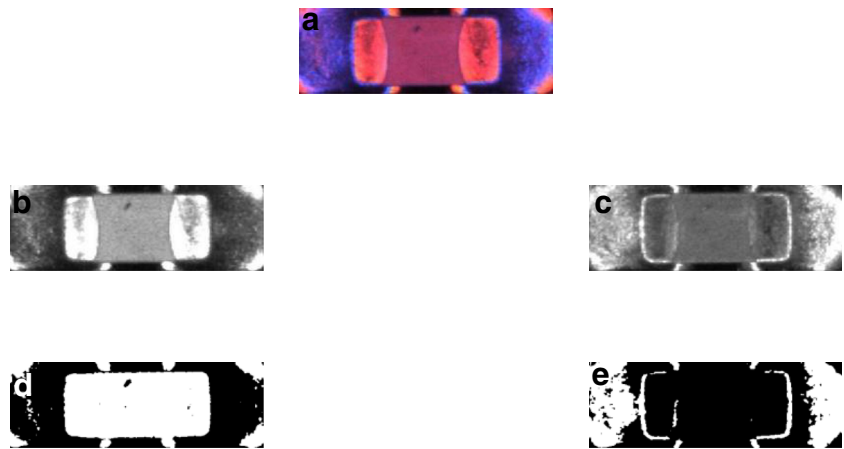
#### 3.2 Feature extraction

Figure 4a shows a component image acquired through the RGB illumination system. The color distribution of the image consists of three colors (red, green, and blue). To use the

Table 1 Component image according to defect type

Symbol	Image		
Normal Component			
	Normal		
Package Defect			
	Wrong	Missing	Rotation
			
	Tombstone	Manhattan	Shift
Solder Defect			
	No-Solder	Exceed	
Complex Defect			
	Pseudo		

**Fig. 4** Component image: **a** image acquired from RGB illumination system, **b** red channel image, **c** blue channel image, **d** binary image of red channel image, and **e** binary image of blue channel image



height information of the solder joint indicated by each color, the component image was divided into each color. Because the green light illuminated the gentle slope of the solder joint, the green region was smaller than the other two regions, as shown in Fig. 4a. Accordingly, excluding the green channel image, as shown in Fig. 4b and c, the red channel image, which illuminated the flat slope, and the blue channel image, which illuminated the steep slope, were separated. From these two channel images, the average intensity value was extracted [7, 9].

In addition, binarization was conducted for each channel image on the binary images of each channel, as shown in Fig. 4d and e. A highlight value was then extracted from the binary images of two channels [8, 14]. Because each feature is extracted from each feature extraction region, if a component image is segmented into  $s$  feature extraction regions, a total of  $4 \times s$  features are extracted. This is defined as  $f_1^i, f_2^i, f_3^i, f_4^i$  ( $i = 1, \dots, s$ ), and each feature is calculated through the following equations.

$$f_1^i = \frac{1}{m \times n} \sum_{x=1}^m \sum_{y=1}^n I_R(x, y) \quad (1)$$

$$f_2^i = \frac{1}{m \times n} \sum_{x=1}^m \sum_{y=1}^n I_B(x, y) \quad (2)$$

$$f_3^i = \frac{1}{m \times n} \sum_{x=1}^m \sum_{y=1}^n T_R(x, y) \quad (3)$$

$$f_4^i = \frac{1}{m \times n} \sum_{x=1}^m \sum_{y=1}^n T_B(x, y) \quad (4)$$

where  $i$  is the index of a feature extraction region from which a feature is extracted,  $m$  and  $n$  are the width and height of the  $i$ -th feature extraction region,  $x$  and  $y$  are the coordinates of the width and height for the region, and finally,  $I_R$  and  $I_B$  indicate the red and blue channel images, and  $T_R$  and  $T_B$  indicate otsu binarization images for  $I_R$  and  $I_B$ , respectively.

### 3.3 Machine learning algorithm

#### 3.3.1 Decision tree

A decision tree is a machine learning algorithm that is most widely used for classification problems. It has a similar architecture as human decision-making. A total of  $4 \times s$  features extracted from  $s$  feature extraction regions are used as input data, and the data are classified based on a specific value of the descriptor of each node. In a decision tree, the final node, where the input data arrive after passing every node, is called the terminal node, and the data are ultimately classified using the value of the terminal node.

#### 3.3.2 Multi-layer perceptron

A multi-layer perceptron (MLP) is a neural network that has one or more hidden layers between the input and output layers. This machine learning algorithm overcomes the limit of a single-layer perceptron, which enables only linear classification through a hidden layer. In particular, MLP has been widely utilized for PCB-related classification [14, 15]. As shown in Fig. 5, this study used an MLP with a single hidden layer. The input layer had  $4 \times s$  nodes with features extracted from  $s$  feature extraction regions as the input data. The hidden layer has 16 nodes including bias. Because the results of the activation function for the ten defect types to be classified are used as the output, the output layer has ten nodes.

The activation function of the MLP used in this study is a bipolar sigmoid function, which can be expressed as follows [19].

$$\tau(f_x) = \beta \times \frac{1 - e^{-\alpha f_x}}{1 + e^{-\alpha f_x}} \quad (5)$$

where  $f_x$  indicates the input feature, and  $\alpha$  and  $\beta$  are the slope parameters of the activation function, for which 0.5 and 2 are used based on the experimental results.



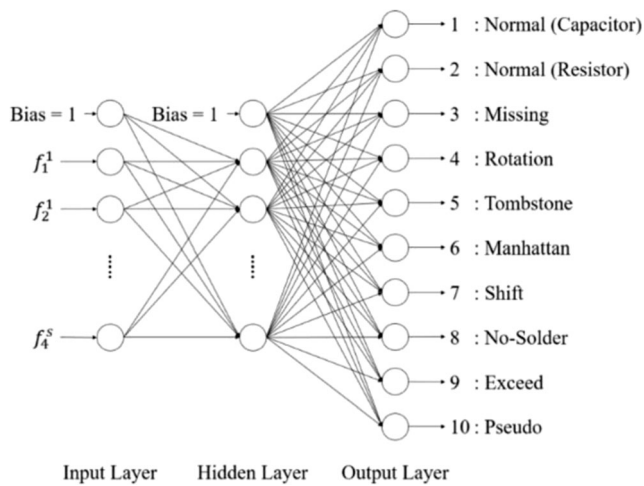


Fig. 5 Multi-layer perceptron

### 3.3.3 Support vector machine

A support vector machine (SVM) is a machine learning algorithm that has been designed to maximize the generalization capability based on statistical learning theory. The kernel function [20–22] maps a non-linear space including the input data, which are not available to linear classification, into a higher dimensional space where linear classification is possible. Next, data are classified by finding an optimal hyperplane that maximizes the margin between classes. Here, the margin is the minimum distance between the optimal hyperplane and each dataset. This study classified defects using an SVM with  $4 \times s$  input nodes, which is the same number as the features extracted from  $s$  feature extraction regions, and ten output nodes.

In this study, the chi-square  $\chi^2$  function is used as the kernel function of the SVM and can be expressed as follows [22].

$$K(f_x, f_y) = e^{-\gamma \frac{(f_x - f_y)^2}{(f_x + f_y)^2}}, \quad \gamma > 0 \tag{6}$$

where  $f_x$  and  $f_y$  are data of a non-linear space where linear classification is impossible, and  $\gamma$  is the scale parameter of the kernel function, which has a value of 0.001 based on the experimental results.

### 3.3.4 Classification experiments

To evaluate the defect type classification performance of the manual setting of the feature extraction regions, we conducted an experiment by obtaining images of real PCB components from AOI. A total of 1400 components of the ten defect types were used in this study, 1065 components of which were used for the training dataset, and the remaining 335 components were used for the test dataset.

Table 2 SMT defect dataset

Defect type	Train	Test
Normal (capacitor)	111	35
Normal (resistor)	95	21
Missing	110	36
Rotation	106	35
Tombstone	106	37
Manhattan	105	30
Shift	108	36
No-Solder	108	34
Exceed	108	36
Pseudo	108	35
<b>Total</b>	<b>1065</b>	<b>335</b>

Among the package defects, the wrong defect refers to a situation in which a resistor mistakenly takes the place of a capacitor or vice versa. Because a wrong defect includes normal soldering, with the exception of the mounting of a wrong component, it has the same color distribution as a normal component image. Accordingly, the defect types of normal and wrong defects were changed to normal (capacitor) and normal (resistor) for the defect type classification. As presented in Table 2, the defect types were classified for each dataset. All component images used for defect type classification in this study have the same pixel resolution of  $160 \times 480$ .

The defect type classification performance was evaluated based on the level of accuracy, which utilizes the following formula.

$$\text{Accuracy}(\%) = \frac{N_C}{N_T} \times 100(\%) \tag{7}$$

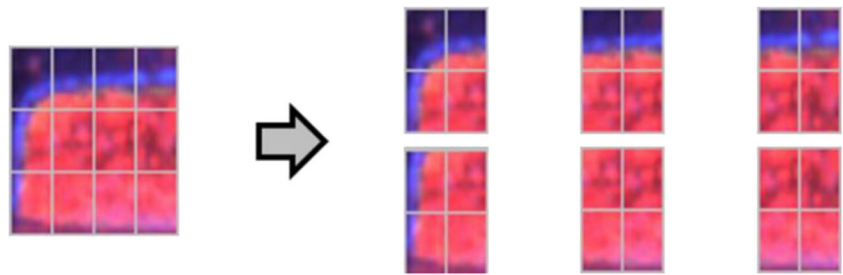
where  $N_T$  is the total number of test datasets, and  $N_C$  is the number of datasets that are classified as equal to real values.

Table 3 shows the result of defect type classification for test dataset of Table 2 using the feature extraction region of Fig. 3. It turns out that the SVM achieved higher classification accuracy for three feature extraction regions than the decision tree and MLP. This study also used the SVM to classify the ten defect types.

Table 3 Defect type classification accuracy according to machine learning methods and feature extraction regions (%)

Defect type	Number of regions		
	3 regions	9 regions	15 regions
Decision tree	44.8	43.6	45.7
SVM	69.6	94.9	45.1
MLP	44.8	94.6	94.0

**Fig. 6** Feature extraction region segmentation method



Using 9 feature extraction regions produced higher accuracy than using 3 feature extraction regions but using 15 feature extraction regions lowered the accuracy compared with the use of 9 feature extraction regions. This indicates that the shape and number of feature extraction regions significantly affect the defect type classification performance. In this regard, we propose a new method that applies an optimization technique to determine a feature extraction region.

#### 4 Optimization of feature extraction regions

During the phase of feature extraction region optimization, using a genetic algorithm, which is an optimization technique, the feature extraction regions are automatically set for the ten defect types. An initial population is created for the genetic algorithm, and the fitness of the population is calculated. After the fitness calculation, if the fitness does not converge, a genetic operation should be conducted, whereas if the fitness converges, the feature extraction region optimization is completed. Ultimately, the optimal feature extraction regions for the ten defect types are automatically set, and these regions are used for defect type classification.

##### 4.1 Initial population generation

The solution to be obtained using the genetic algorithm is expressed as a chromosome. The initial population is an entity that is needed only for later solutions, and thus does not have

to be composed of good solutions. Accordingly, it is created randomly.

First, an input component image is segmented using a  $15 \times 10$  lattice. The feature extraction region generated using the existing methods has overlapped areas, as shown in Fig. 6. This study uses four spaces of the lattice as a single feature extraction region, and thus there are some overlapped areas between regions. Consequently, 126 feature extraction regions, namely,  $14 \times 9$  in size, are created.

The chromosome for the genetic algorithm has the same  $14 \times 9$  size as the segmented feature extraction region and a two-dimensional integral arrangement. Each region is randomly set as a selected or unselected region. The selected region is designated as 1, and the unselected region is designated as 0. The chromosome has an ultimate shape, as shown in Fig. 7.

##### 4.2 Fitness calculation

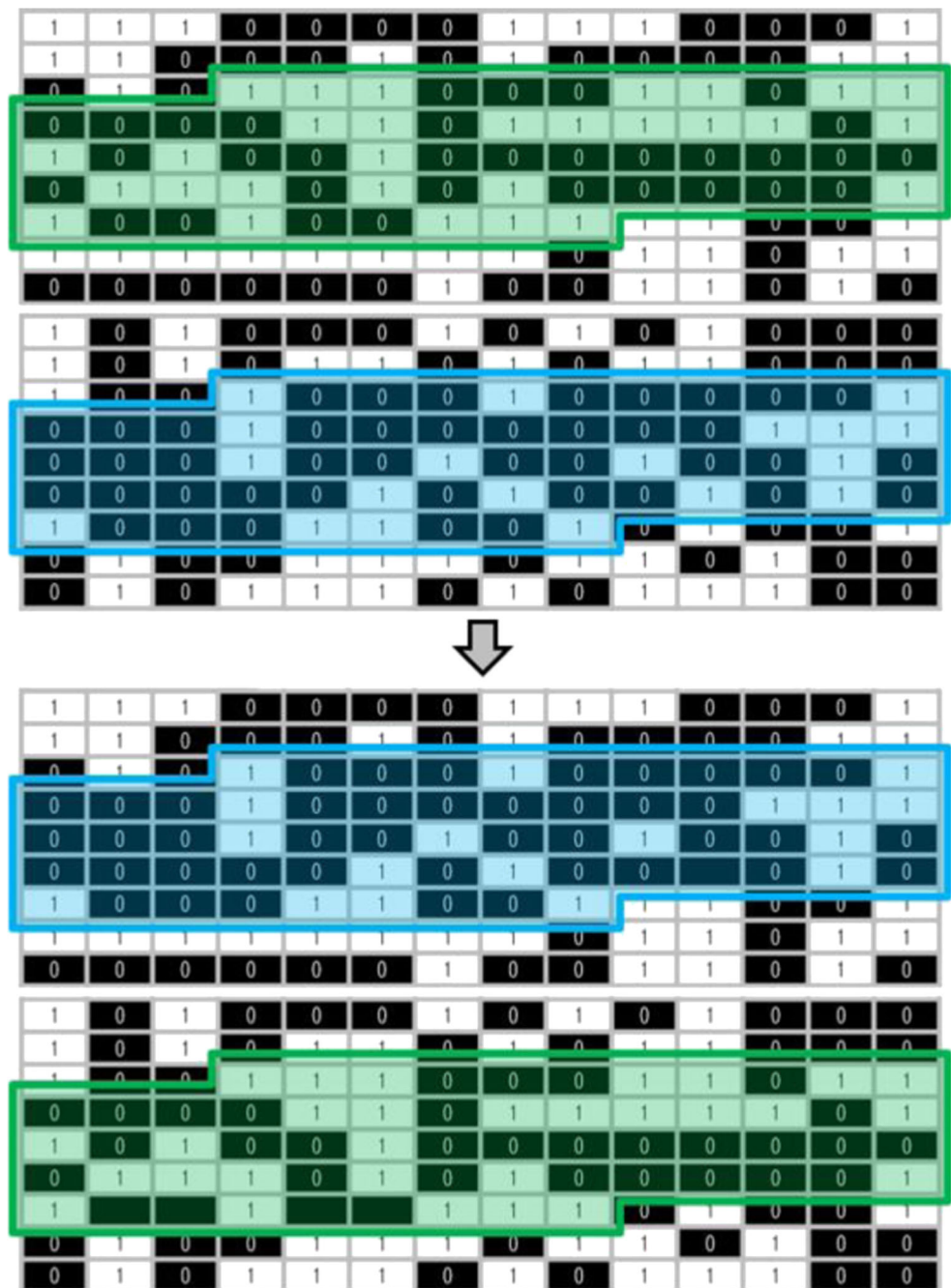
When a chromosome is judged with respect to the optimal solution, the higher the fitness calculation of the chromosome, the more probable it is that the chromosome will be passed down to the next generation. In this study, an optimal chromosome needs to show a good classification performance when the feature extraction region of this chromosome is used. Accordingly, a variance ratio for the extracted features is used for the fitness calculation [5].

As the variance ratio of the features increases, the classification performance is improved using the features. If every variance ratio for the four features has a high value, the best classification performance is achieved. In this case, the variance ratio of the features has the smallest standard deviation,

**Fig. 7** Chromosome

1	1	1	0	0	0	0	1	1	1	0	0	0	1
1	1	0	0	0	1	0	1	0	0	0	0	1	1
0	1	0	1	1	1	0	0	0	1	1	0	1	1
0	0	0	0	1	1	0	1	1	1	1	1	0	1
1	0	1	0	0	1	0	0	0	0	0	0	0	0
0	1	1	1	0	1	0	1	0	0	0	0	0	1
1	0	0	1	0	0	1	1	1	1	1	0	0	1
1	1	1	1	1	1	1	1	0	1	1	0	1	1
0	0	0	0	0	0	1	0	0	1	1	0	1	0

Fig. 8 Two-point crossover



which is used to calculate the fitness of the chromosome. Because a lower standard deviation results in a better defect type classification performance, the reciprocal of the standard deviation is used to calculate the fitness of the genetic algorithm used in this study.

In this study, the chromosome fitness calculation proceeds in the following manner.

Step 1. When  $s$  feature extraction regions are used, the matrix  $F_j \in R^{s \times 1}$ , ( $j = 1, \dots, 4$ ) is defined for four features extracted from each region.

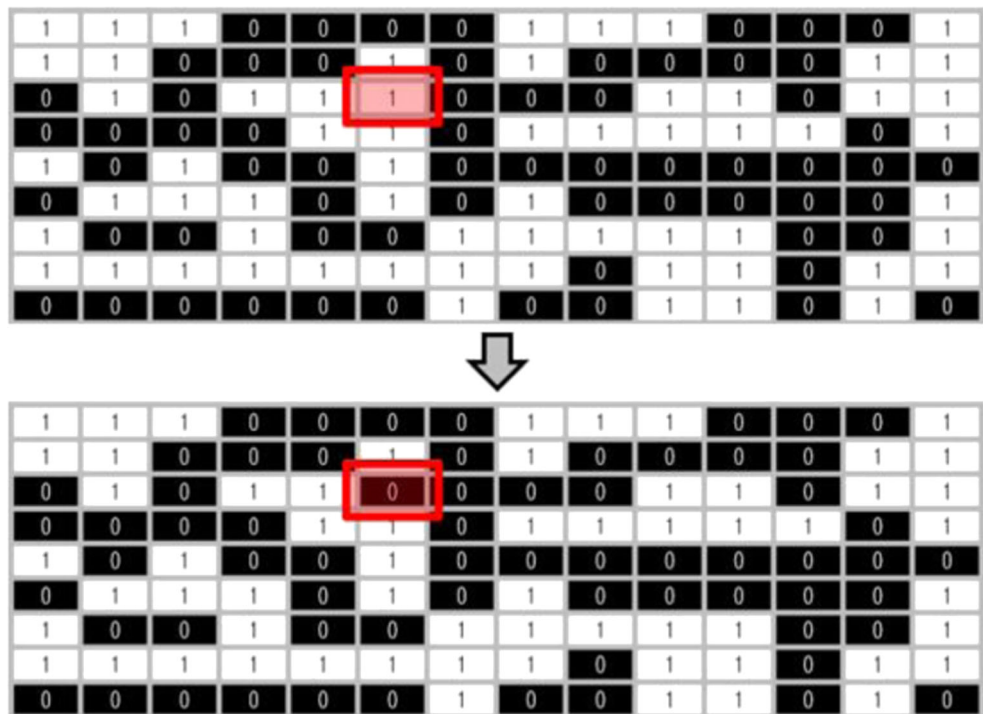
$$F_j = [f_j^1, f_j^2, \dots, f_j^s], \quad j = 1, 2, 3, 4 \tag{8}$$

Step 2. An  $s \times s$  dimensional variance matrix  $S_j \in R^{s \times s}$  or the  $j$ -th feature is calculated using  $F_j$  as follows.

$$S_j = (F_j - M_F) (F_j - M_F)^T \tag{9}$$

where  $M_F \in R^{s \times 1}$  is the average of the extracted feature matrices and is calculated using  $M_F = \frac{1}{4} \sum_{j=1}^4 F_j$ .

Fig. 9 Single-point mutation



Step 3. The feature variance matrix  $S_{Wj} \in R^{s \times s}$  for the same defect types is calculated using the variance matrix  $S_j \in R^{s \times s}$  [23].

$$S_{Wj} = \sum_{j=1}^4 S_j \tag{10}$$

Step 4. The feature variance matrix  $S_{Bj} \in R^{s \times s}$  for different defect types is calculated using the feature variance matrix  $S_j \in R^{s \times s}$  [23].

$$S_{Bj} = (M_F - M_D)(M_F - M_D)^T \tag{11}$$

where  $M_D \in R^{s \times 1}$  is the average of  $M_F$  calculated for each of the ten defect types to be classified and is calculated using  $M_D = \sum_{d=1}^{10} M_{Fd}$ .

Step 5. The variance ratio for the  $j$ -th feature is calculated using two feature variance matrices,  $S_{Wj}$  and  $S_{Bj}$ , and the standard deviation  $\sigma$  of the variance ratio for all features is calculated.

$$\sigma = \sqrt{\frac{1}{4} \sum_{j=1}^4 \left( \frac{\det(S_{Bj})}{\det(S_{Wj})} - m_s \right)^2} \tag{12}$$

where  $m_s$  is the average variance ratio of four extracted features and is calculated using  $m_s = \frac{1}{4} \sum_{j=1}^4 \frac{\det(S_{Bj})}{\det(S_{Wj})}$ .

Step 6. The reciprocal of the standard deviation is calculated and used to obtain the ultimate fitness calculation.

$$g = \frac{1}{\sigma} \tag{13}$$

### 4.3 Genetic operation

#### 4.3.1 Selection

In this phase, the candidate chromosomes that can be inherited in the next generation are selected. We used the remaining stochastic sampling with replacement as the selection method. The procedure is as follows.

Step 1. The selection probability  $p_i$  and expected quantity of duplicates  $e_i$  for each chromosome of the present generation are calculated.

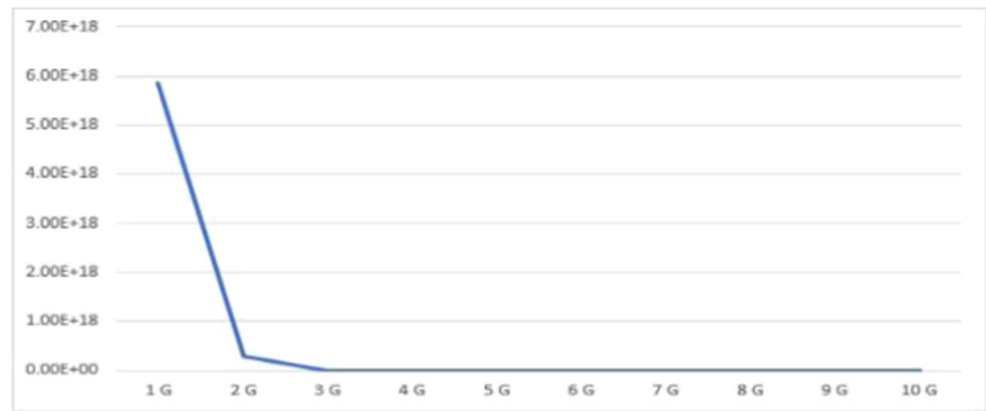
$$p_i = \frac{g_i}{\sum g_i} \tag{14}$$

$$e_i = \text{popsize} \times p_i \tag{15}$$

where  $g_i$  is the value of the fitness calculation for the  $i$ -th chromosome, and  $\text{popsize}$  is the total number of chromosomes in a single generation.



**Fig. 10** Fitness value according to generation number



- Step 2. The  $i$ -th chromosome is duplicated in the next generation into as many numbers as the integer part of  $e_i$ .
- Step 3. A roulette wheel is constructed using the fraction mantissa of  $e_i$  to randomly duplicate the insufficient genes.

#### 4.3.2 Crossover

A new chromosome is created by making two chromosomes cross over each other with a specific probability. This study used a two-point crossover. In this method, two cut points are set at the same location in two chromosomes, as shown in Fig. 8, and the genes between the two cut points cross over each other.

#### 4.3.3 Mutation

In this phase, one or more chromosomes are mutated with a specific probability to create a new chromosome. We applied a

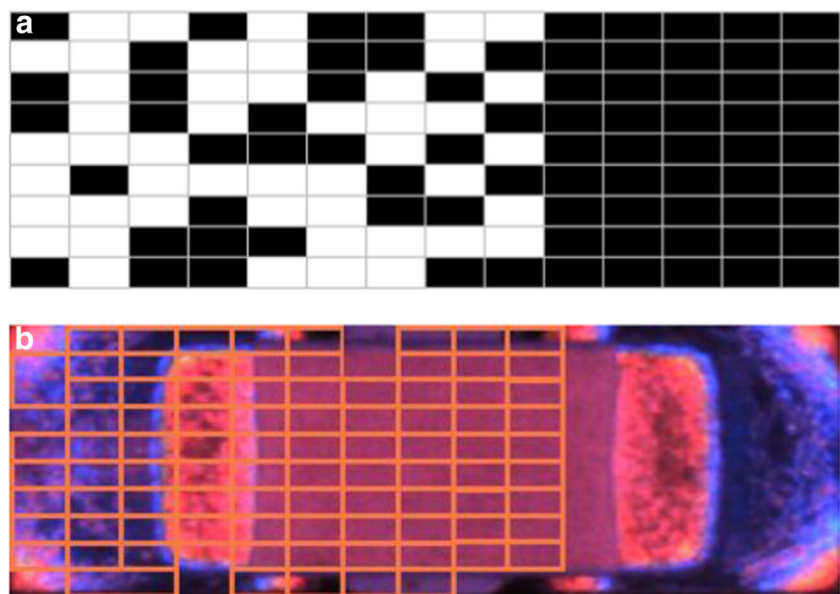
single-point mutation. As shown in Fig. 9, after a single gene is selected from a chromosome designated as either 0 or 1, if the value of the gene is 1, it is mutated into 0, and vice versa.

## 5 Experimental result

The genetic algorithm used for the feature extraction region optimization has the following specifications: a total of 100 generations, 50 genes per generation, a 0.25% crossover, and a 0.03% mutation. Components of the training datasets in Table 2 were used to set the optimal feature extraction regions for the ten defect types.

Figure 10 shows the fitness values of each generation during the genetic algorithm application. The optimal feature extraction region obtained using the genetic algorithm described in Section 4 is illustrated in Fig. 11a. The result of the optimal region applied to the component image acquired from the RGB illumination system is shown in Fig. 11b.

**Fig. 11** Result of feature extraction regions optimization: **a** optimal feature extraction region chromosome, **b** optimal region in component image



**Table 4** Defect type classification accuracy (%)

	Manual method		Automatic method (proposed)	
Number of regions	15	30	45	47
Normal (capacitor)	76.3	75.0	75.2	100.0
Normal (resistor)	97.9	97.6	97.7	100.0
Missing	90.9	92.9	92.1	97.2
Rotation	57.6	69.5	61.3	88.6
Tombstone	95.3	98.8	99.4	100.0
Manhattan	97.1	100.0	100.0	100.0
Shift	46.8	59.7	58.1	97.2
No-solder	74.2	75.8	77.3	94.1
Exceed	87.9	86.5	85.7	91.7
Pseudo	91.7	86.1	83.3	91.4
Average	81.6	84.2	83.0	96.0

Table 4 compares the manual feature extraction region selection method (15, 30, 45 regions) with the automatic feature extraction region selection method (proposed method) in classification accuracy. All methods used SVM for defect type classification and Eq. (7) is used for accuracy calculation. The proposed method classified 335 defects with 96.0% accuracy, which was better than the existing method.

Table 5 shows the average computation time for the manual methods and proposed method. This table shows that the computational time of the proposed method (3.70 ms/component) is not so larger than the other cases (1.54 to 4.42 ms/component). Actually, the advantage of the proposed method is not computational time but region setup time. The manual cases spend a lot of time to determine the feature extraction region, but our method determines the regions automatically.

## 6 Conclusion

We proposed a new method for SMT defect type classification. A genetic algorithm is implemented to determine the optimal feature extraction regions for SMT defect types to be classified. Features are extracted from the determined feature extraction regions and used as input data for an SVM applied to classify the defect types. The experimental results showed that the proposed method achieves a superior

**Table 5** Average computational time (ms/component)

	Manual method		Automatic method (Proposed)	
Number of regions	15	30	45	47
Feature extraction	1.28	2.21	3.29	2.81
SVM calculation	0.26	0.97	1.13	0.89
Total	1.54	3.18	4.42	3.70

classification performance in setting the feature extraction regions than the existing method.

We classified the defect types by optimizing the feature extraction regions. However, there are some problems with the proposed method. First, the components used for defect type classification are limited to the capacitor and resistor. Second, only component images obtained using RGB illumination are applicable to the defect type classification. Therefore, further studies will attempt to include other components such as an integrated circuit and tantalum capacitor and extend the illumination conditions to include general lighting.

**Acknowledgements** This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. R7117-16-0164, Development of eight wide area driving environment awareness and cooperative driving technology, which are based on V2X wireless communication)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

- Kong F (2008) A new method of inspection based on shape from shading. *Int Congress on Image and Signal Process*, pp. 291–294
- Jiang J, Cheng J, Tao D (2012) Color biological features based solder paste defects detection and classification on printed circuit boards. *IEEE Trans Compon Packag Manuf Technol* 2(9):1536–1544
- Cai N, Lin J, Wang H, Weng S, Ling BW-K, Ye Q, New A (2016) IC solder joint inspection method for an automatic optical inspection system based on an improved visual background extraction algorithm. *IEEE Trans Compon Packag Manuf Technol* 6(1):161–172
- N. Cai, Q. Ye, G. Liu, H. Wang, and Z. Yang, IC solder joint inspection based on the Gaussian mixture model. *International Journal of Soldering and Surface Mount Technology* 28(4):207–214 (2016)
- Cai N, Zhou Y, Ye Q, Liu G, Wang H, Chen X (2017) IC solder joint inspection via robust principle component analysis. *IEEE Trans Compon Packag Manuf Technol* 7(2):300–309
- Xie HW, Zhang XM, Kuang YC, Ouyang GF (2011) Solder joint inspection method for chip component using improved AdaBoost and decision tree. *IEEE Trans Compon Packag Manuf Technol* 1(12):2018–2027
- Wu H, Zhang X, Xie H, Kuang Y, Ouyang G (2013) Classification of solder joint using feature selection based on Bayes and support vector machine. *IEEE Trans Compon Packag Manuf Technol* 3(3): 516–522
- Hao W, Xianmin Z, Yongcong K, Gaoferi O, Hongwei X (2013) Solder joint inspection based on neural network combined with genetic algorithm. *Int J Light and Electron Optics* 124(20):4110–4116
- Sim KJ, Yun TS, Kim HJ (2000) Solder joint inspection system using support vector machine and circular illumination. *Proceed KIISE Spring Conference (in Korean)* 27(1B):607–609
- Xie H, Kuang Y, and Zhang X (2009) A high speed AOI algorithm for chip component based on image difference. *IEEE Int Conf Inform Autom*, pp. 969–974

11. Wu F, Zhang X (2011) Feature-extraction based inspection algorithm for IC solder joints. *IEEE Trans Compon Packag Manuf Technol* 1(5):689–694
12. Wu F, Zhang X, Kuan Y, and He Z (2008) An AOI algorithm for PCB based on feature extraction inspection. *Int Conf Intellig Control and Autom*, pp. 240–247
13. Wu F, Zhang X (2014) Inspection and classification method for chip solder joints using color grads and Boolean rules. *Int J Robotics Comp-Integr Manuf* 30(5):517–526
14. Song JD, Kim YG, Park TH (2017) Defect classification method of PCB solder joint by color features and region segmentation. *J Instit Control Robotics and Systems (in Korean)* 23(12):1086–1091
15. Lee JS, Park TH (2015) Defect classification of components for SMT inspection machines. *J Instit Control Robotics and Systems (in Korean)* 21(10):982–987
16. Youn SG, Lee YA, Park TH (2015) Automatic classification of SMD packages using neural network. *J Instit Control Robotics and Systems (in Korean)* 21(3):276–282
17. Youn SG, Kim YG, Park TH (2017) Wavelet transform based defect detection for PCB inspection machines. *Trans Korean Instit Electric Eng* 66(10):849–857
18. Li X, Tso SK, Guan X, Huang Q (2006) Improving automatic detection of defects in castings by applying wavelet technique. *IEEE Trans Ind Electron* 53(1):849–857
19. Shenouda A (2006) A quantitative comparison of different MLP activation functions in classification. *Int Symp Neural Networks*, pp. 849–857
20. Vedaldi A, Zisserman A (2011) Efficient additive kernels via explicit feature maps. *IEEE Trans Pattern Anal Mach Intell* 34(3): 480–492
21. Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods-Support Vector. Learning*, pp. 185–208
22. Fan R, Chen P, Lin C (2005) Working set selection using second order information for training support vector machines. *Int J Mach Learn Res* 6:1889–1918
23. Mika S, Ratsch G, Weston J, Scholkopf B, and Mullers R (1999) Fisher discriminant analysis with kernels. *Proceed 1999 IEEE Signal Process Soc Workshop*, pp. 41–48