Automatic Optical Inspection to Detect Missing Components in Surface Mount Assemblies

Hendawan Soebhakti
Electrical Engineering
State Polytechnic of Batam
Batam, Indonesia
hendawan@polibatam.ac.id

Farkhad Ihsan Hariadi
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
hariadi@ic-proc.paume.itb.ac.id

***Abstract*—Missing component is one of defects that occur during PCB assembly process, especially when using Surface Mount Technology (SMT). Manual visual inspection by an operator in assembly line is having a risk. Human fatigue is one of the main factor where operator missing the defects. In this paper, we develop AOI by employing neural networks. By extracting histogram value from PCB image, neural networks can classified component under inspection is exist or missing. Experiment result shows 93, 93% success rate can achieve by the system when inspecting small components.**

***Keywords*—Missing component; PCB assembly; Automatic Optical Inspection, Neural network**

I. INTRODUCTION

Using Surface Mounted Technology (SMT) in PCB assembly process can increase production speed but also the risk of defect. Some defects which can occur are missing component, misalignment component, side overhang, and end overhang. The defects are commonly detected by doing a visual inspection. In an assembly line, visual inspection usually done manually by an operator. Human fatigue is one of the main factor that possibly cause the operator missing to detect the defects. Automatic Optical Inspection (AOI) becomes one of the solutions to decrease misidentification of defects.

Some authors have converted LPKF prototyping machine for PCBs into an AOI machine [1]. For implementing the AOI program, they used IMAQ function existing in LABVIEW. The AOI can detect incorrect drill position on the pad, remain copper between traces, solder in excess and component value using OCR (Optical Character Recognition). The experimental results show that camera focus is an important thing in AOI. Bad recognition can happen if the camera is not focused on the component.

Other authors have been developing an AOI to detect defective solder in through-hole components [2]. A threshold value is applied on acquired image to produce a binary image. Solder size can be calculated based on the number of white pixels in each of ROI (Region of Interest). The solder is considered as OK if the solder size is in the defined range value. The test results for the PCB which has been classified as OK has 0% rate of false recognition. And for the PCB which has classified as NOK has 53, 33% rate of false recognition.

In this paper, we use neural network to develop AOI for detecting missing component defect. Webcam with autofocus function is using to avoid bad recognition. Histogram value is extracted from grayscale PCB images and used as input of neural networks.

II. FEATURE EXTRACTION

Fig.1 is a feature extraction process to get histogram value of the image. A camera is used to capture PCB image as shown in Fig.2. First, marking areas where component will be inspect. In this experiment, we use two areas: Area 1 marked by red rectangular and Area 2 marked by green rectangular. The next step is cropping the image inside those areas. Therefore the cropping area has variety of size based on component size; it must be resized to make sure final cropping images are in the same sizes, which are 320 x 240 pixels. Fig.3 is shown images after cropping and resizing.

The cropped image is still in RGB color, and need to be converted to grayscale image. From this grayscale image we can get histogram. Fig.4 and Fig.5 are shown histogram from images in Area 1 and Area 2.

To simplify computational process in neural network, the actual value of histogram (H_a) needs to be normalized and become normalized value (H_n) between 0 to 1 by using equation (1).

$$H_n[i] = H_a[i]/(X * Y) \quad (1)$$

Where X is width of resized image (320 pixels), Y is height of resized image (240 pixels), and i is grayscale level 0,1,…,255.

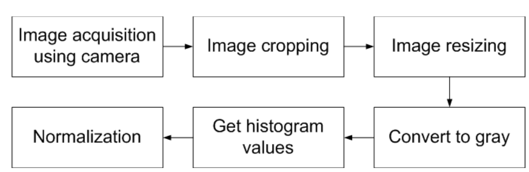

Fig. 1. Feature extraction process

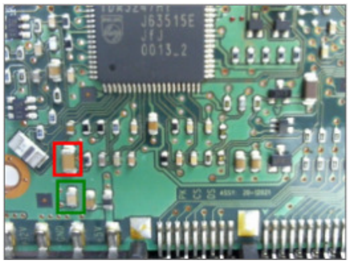

Fig. 2. Source image from camera

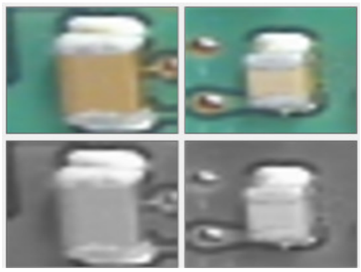

Fig. 3. Image cropping and resizing : Area 1 (Left); Area 2; (Right); RGB image (Top); Gray image (Bottom)

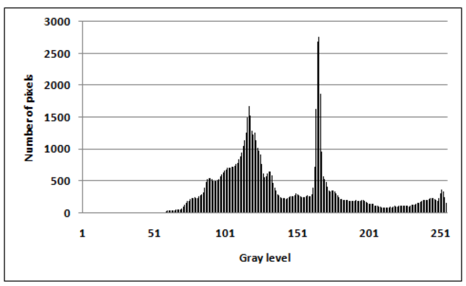

Fig. 4. Histogram of gray image at Area 1

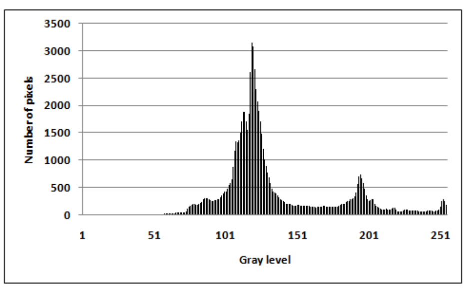

Fig. 5. Histogram of gray image at Area 2

III. Neural Network

Training the neural network by using the standard back propagation algorithm can be performed according to the following algorithm [3]:

Step 1. Initialize the network weights to small random values.

Step 2. From the set of training input/output pairs, present an input pattern and calculate the network response.

Step 3. Calculate all the local errors using (2) and (3).

Step 4. Update the network weights using (4).

Step 5. If convergence is achieved, stop; if not convergence go to step 2.

$$\delta_j^{(s)} = \left(d_{qh} - x_{out,j}^{(s)}\right) g\left(v_j^{(s)}\right) \quad (2)$$

$$\delta_j^{(s)} = \left(\sum_{h=1}^{n_{s+1}} \delta_h^{(s+1)} w_{hj}^{(s+1)}\right) g\left(v_j^{(s)}\right) \quad (3)$$

$$w_{ji}^{(s)}(k+1) = w_{ji}^{(s)}(k) + \mu^{(s)} \delta_j^{(s)} x_{out,i}^{(s)} \quad (4)$$

Where s designates appropriate network layer, $\delta_j^{(s)}$is local error, $d_{qh}$is desired networks output for the qth input pattern, $x_{out,j}^{(s)}$ is actual output networks, $g\left(v_j^{(s)}\right)$ is first derivative of the nonlinear activation function, $w_{hj}^{(s+1)}$is weights in s layer, $\mu^{(s)}$is learning rate parameter.

In this experiment, we use two neural networks as shown in Fig.6. Each neural network has 256 nodes at input layer, 150 nodes at hidden layer 1, 20 nodes at hidden layer 2, and 2 nodes at output layer.

For the first test, we try to inspect component in black color with size (length x width): 4 x 2 mm at Area 1 and 2, 5 x 1, 25 mm at Area 2. Neural network 1 is trained using images taken from Area 1 as shown in Fig.7. Four images on top are exist category and four images on bottom are missing category. We can also see the histogram in Fig. 8 and Fig.9 for both categories have different pattern. Fig.10 is trained images for neural network 2; they are taken from Area 2.

For the second test, we try to inspect brown components with smaller size: 3 x 1.25 mm at Area 1 and 2 x 1 mm at Area 2. Fig.11 and Fig.12 show the trained images for neural networks.

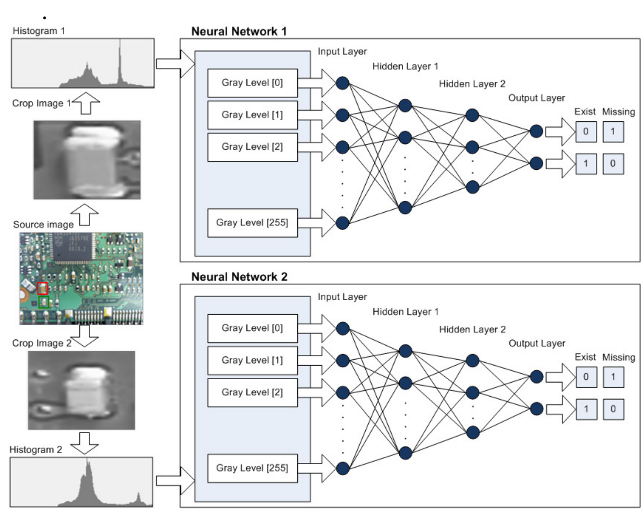

Fig. 6. Neural network structure

Fig. 7. Trained images for Area 1 : Exist category (Top); Missing category (Bottom)

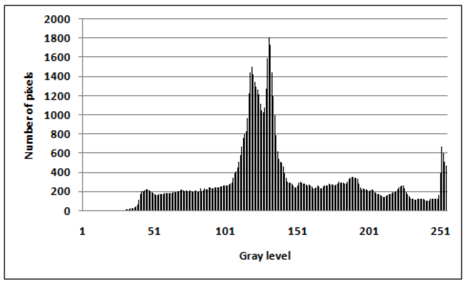

Fig. 8. Histogram of trained image at Area 1 for exist category

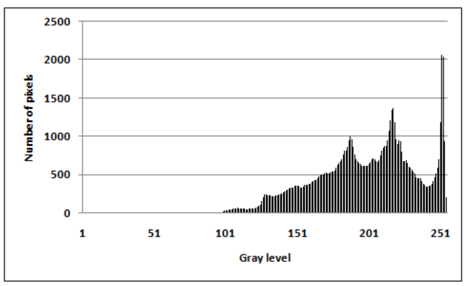

Fig. 9. Histogram of trained image at Area 1 for missing category

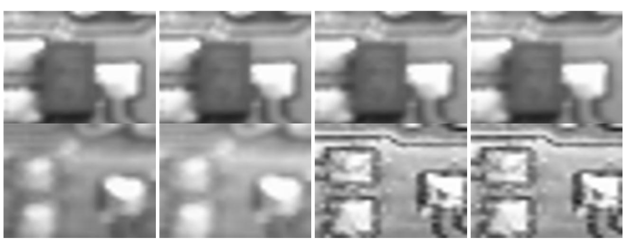

Fig. 10. Trained images for Area 2 : Exist category (Top); Missing category (Bottom)

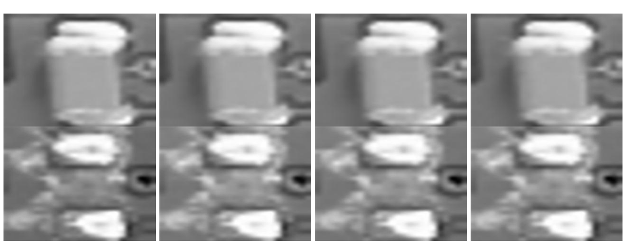

Fig. 11. Trained images for Area 1 : Exist category (Top); Missing category (Bottom)

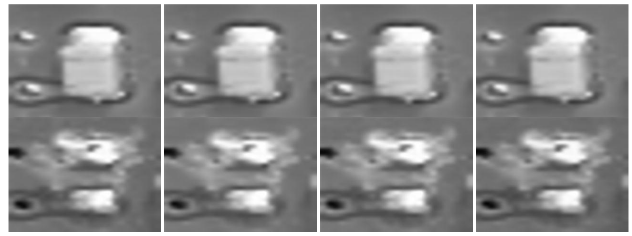

Fig. 12. Trained images for Crop Area 2 : Exist category (Top); Missing category (Bottom)

IV. Experimental Results

In this experiment, we use Logitech webcam c525 that can provide video capture up to 1280 x 720 pixels. The webcam is connected to a PC with an Intel Core i3 CPU 3.20GHz, 2 Gbytes of RAM, and Windows 7 professional 32 bit.

For the first test, we inspect 30 boards with actual components condition as shown in Table I. Neural networks recognize the components as same as actual condition. The test result has 100 % success rate.

For the second test, for smaller components, neural networks have 93, 93% success rate as shown in Table II. False recognition is occurring for test number 6 and 24. In both of them, neural networks make two false detections for the exist component at Area 2. Test image 6 and 24 in Fig. 13 show different component color than others. In this two test images, the color of component are white, not brown.

TABLE I. Recognition results for 1^{ST} test

Test No	**Actual Condition**		**NN Recognition**		**Status**
	Area 1	***Area 2***	***Area 1***	***Area 2***	
1	Exist	Exist	Exist	Exist	Success
2	Missing	Missing	Missing	Missing	Success
3	Exist	Exist	Exist	Exist	Success
4	Missing	Exist	Missing	Exist	Success
5	Exist	Exist	Exist	Exist	Success
6	Exist	Exist	Exist	Exist	Success
7	Exist	Exist	Exist	Exist	Success
8	Missing	Missing	Missing	Missing	Success
9	Exist	Missing	Exist	Missing	Success
10	Exist	Exist	Exist	Exist	Success
11	Exist	Exist	Exist	Exist	Success
12	Missing	Exist	Missing	Exist	Success
13	Missing	Missing	Missing	Missing	Success
14	Exist	Exist	Exist	Exist	Success
15	Exist	Missing	Exist	Missing	Success
16	Exist	Exist	Exist	Exist	Success
17	Exist	Exist	Exist	Exist	Success
18	Exist	Exist	Exist	Exist	Success
19	Missing	Missing	Missing	Missing	Success
20	Missing	Exist	Missing	Exist	Success
21	Exist	Missing	Exist	Missing	Success
22	Exist	Exist	Exist	Exist	Success
23	Exist	Exist	Exist	Exist	Success
24	Missing	Exist	Missing	Exist	Success
25	Exist	Exist	Exist	Exist	Success
26	Missing	Exist	Missing	Exist	Success
27	Exist	Exist	Exist	Exist	Success
28	Exist	Exist	Exist	Exist	Success
29	Exist	Missing	Exist	Missing	Success
30	Missing	Missing	Missing	Missing	Success
Success rate					100%

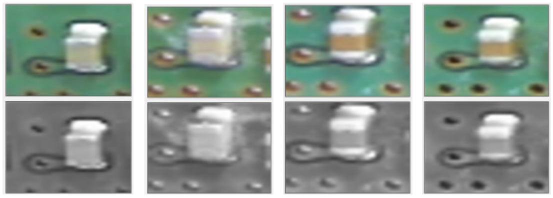

Fig. 13. Test images (From left to right): test image 6, test image 24, test image26, test image 3.

TABLE II. RECOGNITION RESULTS FOR 2ND TEST

Test No	Actual Condition		NN Recognition		Status
	Area 1	*Area 2*	*Area 1*	*Area 2*	
1	Missing	Exist	Missing	Exist	Success
2	Exist	Exist	Exist	Exist	Success
3	Exist	Exist	Exist	Exist	Success
4	Missing	Missing	Missing	Missing	Success
5	Exist	Missing	Exist	Missing	Success
6	Exist	Exist	Exist	Missing	Fail
7	Exist	Missing	Exist	Missing	Success
8	Exist	Missing	Exist	Missing	Success
9	Exist	Exist	Exist	Exist	Success
10	Missing	Missing	Missing	Missing	Success
11	Missing	Exist	Missing	Exist	Success
12	Exist	Missing	Exist	Missing	Success
13	Exist	Exist	Exist	Exist	Success
14	Exist	Missing	Exist	Missing	Success
15	Missing	Exist	Missing	Exist	Success
16	Missing	Missing	Missing	Missing	Success
17	Exist	Exist	Exist	Exist	Success
18	Missing	Missing	Missing	Missing	Success
19	Missing	Exist	Missing	Exist	Success
20	Exist	Exist	Exist	Exist	Success
21	Exist	Missing	Exist	Missing	Success
22	Missing	Exist	Missing	Exist	Success
23	Exist	Exist	Exist	Exist	Success
24	Exist	Exist	Exist	Unknown	Fail
25	Exist	Missing	Exist	Missing	Success
26	Exist	Exist	Exist	Exist	Success
27	Exist	Exist	Exist	Exist	Success
28	Exist	Exist	Exist	Exist	Success
29	Missing	Exist	Missing	Exist	Success
30	Exist	Exist	Exist	Exist	Success
Success rate					93,33%

V. CONCLUTIONS

By extracting histogram value from PCB image, neural network can classified component under inspection is exist or missing. The color and size of component affected to recognition success rate. Experiment result shows 93, 93% success rate can achieve by the system when inspecting small components.

REFERENCES

[1] M. M. Valentin, P. Dan, and C. Gabriel, "Automated Optical Inspection Tool Using the LPKF PCB Mechanical Prototyping Machine", 2nd Electronics Systemintegration Technology Converence Greenwich, UK, pp. 1359-1362, 2008.

[2] H. G. da Silva, T.G. Amaral and O. P. Dias, "Automatic Inspection for Detecting Defective Solders on Printed Circuit Boards", IEEE, pp. 1087-1091, 2010.

[3] F. M. Ham and I. Kostanic, *Principles of Neurocomputing for Science & Engineering*, International ed., New York, America: McGraw-Hill,pp.109-110,2001.