DOI: https://doi.org/10.36722/exc.v1i1.2344

Product Defect Detection Using Image Template Matching with MATLAB

Muhammad Hasya Abdillah[1*], Muhammad Arip Putra Sabilah[1], Andika Suherman[1], Dwi Astharini[1]

[1]*Department of Electrical Engineering, Faculty of Science and Technology, University of Al-Azhar Indonesia, Sisingamangaraja, South Jakarta, 12110, Indonesia*
Correspondence Email: hhhabdillah888@gmail.com

ABSTRACT

In industrial manufacturing processes, ensuring the quality of products is crucial. This paper proposes a system for detecting defects in products using image template matching techniques implemented in MATLAB. The system's primary function is to compare captured images of products with predefined templates to identify potential defects accurately. The method employed in this system is template matching, a well-established approach in image analysis that allows for efficient defect detection. MATLAB, a widely used software tool for image processing, provides the necessary functionality and a robust set of algorithms to implement the proposed system. The experimental results demonstrate the effectiveness of the approach in detecting various types of defects, such as scratches, cracks, and misalignments. This defect detection system offers a reliable and automated solution for improving the efficiency and productivity of manufacturing industries. By enabling early detection and intervention, it contributes to enhancing product quality control and minimizing defective outputs, ultimately leading to cost savings and customer satisfaction.

Keywords*: Defect Detection; Matlab; Template Matching; Image Processing*

1. INTRODUCTION

The detection of defects in manufactured products is a critical aspect of quality control across diverse industries. Ensuring that products meet stringent standards is vital for maintaining customer satisfaction and safeguarding the reputation of the manufacturing process. Manufacturers encounter several challenges in implementing an effective defect detection system. Traditional manual inspection methods are often time-consuming, labor-intensive, and susceptible to human errors [1], however, automated defect detection systems utilizing computer vision and image processing techniques present a promising solution to overcome these obstacles. Among these techniques, image template matching stands out as a widely adopted approach in the field of computer vision. This method involves comparing a template image with the input image to pinpoint regions of similarity. By employing image template matching to identify defects in products, the inspection process can be streamlined, reducing dependency on human judgment, and ensuring a consistent and precise defect identification process.

This paper presents a defect detection approach employing image template matching with MATLAB. The proposed method encompasses a series of distinct stages, including image grayscale conversion, image adjustment, image segmentation, image filtering, and correlation using the normalized cross-correlation technique [2], the primary objective of this research is to develop a robust defect detection system capable of accurately identifying defects in various product types, with a specific focus on printed circuit boards (PCBs), nails, and bolts. These products hold significant importance in a wide array of industries, and detecting defects during their manufacturing process can profoundly impact product quality and safety. By leveraging image template matching, the proposed approach seeks to offer an efficient and effective solution to automate the inspection process, mitigate human errors, and ultimately enhance overall product quality.

The evaluation of the proposed defect detection method will be conducted based on three key criteria: the system's status, elapsed time, and accuracy. Monitoring the system's status will ensure its stability and reliability throughout the defect detection process. Moreover, the elapsed time for analyzing each product image will be measured to assess the system's efficiency. However, the most crucial evaluation metric lies in accuracy, which gauges the system's ability to correctly identify defects while minimizing false negatives and false positives. By conducting comprehensive performance evaluations, this research aims to demonstrate the viability and potential applicability of the defect detection system in real-world manufacturing settings. prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

2. METHOD

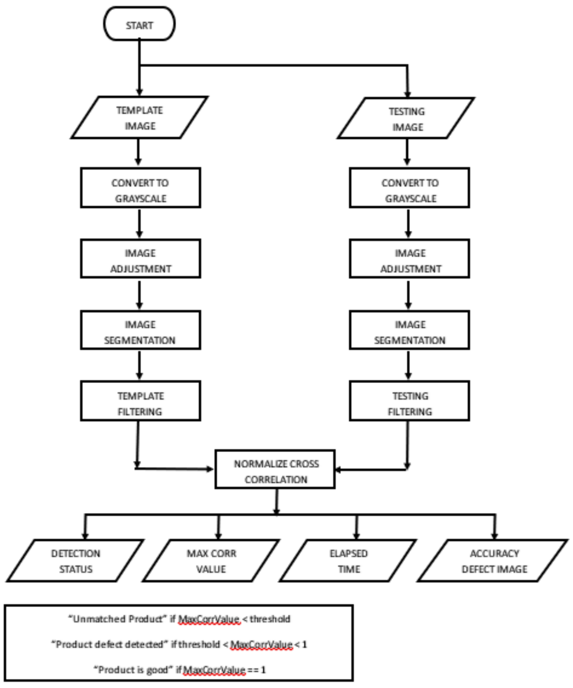

Figure 1. Process of Research

Image Conversion from RGB to Grayscale

Image conversion from RGB (Red, Green, Blue) to grayscale is a fundamental process in image processing, serving as a crucial preparatory step for various computer vision tasks. The conversion aims to transform a color image, typically represented by three color channels (R, G, B), into a grayscale representation that possesses only one intensity channel per pixel. In this process, the relative luminance information of each pixel is utilized to obtain a grayscale value that accurately represents the pixel's brightness level in the original color image [3], grayscale images simplify data processing and analysis, enabling efficient computation of image features and patterns without the complexity of dealing with multiple color channels. Moreover, the grayscale conversion preserves essential image structures and is widely employed in diverse applications, including edge detection, object recognition, and image enhancement.

In MATLAB, the image conversion from RGB to grayscale is seamlessly facilitated by the *rgb2gray* function. This built-in function accepts an RGB image as input and efficiently computes the corresponding grayscale representation. The function utilizes a specific algorithm to derive the grayscale value for each pixel, either by averaging the R, G, and B components or employing a weighted sum based on human perception of color intensity. The resulting grayscale image contains single-channel pixel values ranging from 0 (representing black) to 255 (indicating white). By leveraging MATLAB's *rgb2gray* function, researchers and practitioners can effortlessly convert color images into grayscale, facilitating subsequent image analysis and computer vision operations with enhanced simplicity and computational efficiency.

Image Adjustment

image adjustment is a crucial image processing technique that aims to enhance the visual quality and improve the interpretability of grayscale images. This method involves transforming the pixel intensities of a grayscale image to optimize its contrast, brightness, and overall appearance. The primary objective is to stretch or compress the intensity range of the image to utilize the entire dynamic range effectively. In MATLAB, the adjust function is commonly employed to perform grayscale image adjustment. This built-in function allows users to adjust the intensity values of an image based on specified intensity mapping parameters. The adjust function enables users to specify the desired output intensity range, allowing them to stretch or compress the original intensity values accordingly. Additionally, users can fine-tune the gamma value to control the contrast of the resulting image [4].

Image Segmentation

Image segmentation is a pivotal technique in image processing, aimed at partitioning a grayscale image into distinct and meaningful regions or objects. This process plays a crucial role in isolating areas of interest, simplifying image analysis, and enabling targeted processing of specific image components. In grayscale image segmentation, pixels are classified into different groups based on their intensity values or spatial properties [5].

MATLAB offers several built-in functions to perform grayscale image segmentation, with the most commonly used being gray thresh and binarize. The graythresh function calculates an optimal threshold value based on the image histogram, which efficiently separates foreground and background regions. The binarize function then converts the grayscale image into a binary image by thresholding it using the specified threshold value. The binary image contains two intensity values, typically representing the foreground (object of interest) as white and the background as black.

Image Filtering

The primary purpose of image filtering is to alter the pixel values of an image based on specific predefined operations or mathematical kernels. This process helps to remove noise, highlight image features, and smooth irregularities, ultimately leading to improved image quality and better visualization of important details. MATLAB offers a comprehensive set of functions for image filtering, with the most commonly used being imfilter and conv2. The filter function enables users to apply various filter kernels, such as Gaussian, Sobel, or Laplacian filters, to an input image. This function efficiently convolves the filter kernel with the image, resulting in the desired filtering effect. On the other hand, the conv2 function performs a 2-dimensional convolution operation between the image and a user-defined filter. Both functions are highly versatile and allow users to tailor the filter kernel to suit specific image processing requirements.

Normalize Cross-Correlation

Normalized Cross Correlation (NCC) is a powerful technique widely used in image processing and computer vision for pattern matching and template matching tasks. NCC measures the similarity between an image template and a target image at different positions, providing a correlation value that reflects the degree of resemblance between the two images. The NCC process involves normalizing the pixel intensities of both the template and target images, which ensures that the correlation is invariant to changes in overall brightness and contrast. This normalization step is crucial for accurate matching and robustness against variations in illumination and image conditions. The resulting correlation values are within the range of (-1, 1), where 1 indicates a perfect match, 0 denotes no correlation, and -1 implies a perfect anti-correlation. NCC has proven to be highly effective in detecting objects, identifying patterns, and locating regions of interest in complex images [6].

MATLAB provides a straightforward implementation of Normalized cross-correlation through the normxcorr2 function. This function takes the image template and the target image as input and computes the normalized cross-correlation over the entire target image. The output of normxcorr2 is a correlation map that highlights regions of high similarity between the template and the target image. By analyzing the correlation map, researchers can identify the positions of the template within the target image with the highest correlation values, indicating potential matches. The Normalized cross-correlation technique has demonstrated its versatility and effectiveness in various computer vision applications, including object detection, image registration, and motion tracking, making it a valuable tool in the arsenal of image processing methods [7].

3. RESULT AND DISCUSSIONS

The results obtained from employing template matching with correlation in MATLAB showcased its effectiveness as a defect detection method. The system accurately determined the status of the tested images, successfully identifying the presence or absence of defects. Additionally, the analysis of maximum correlation values demonstrated the degree of similarity between the testing images and their corresponding templates, validating the method's capability to accurately match and detect patterns. The defect detection system exhibited remarkable accuracy, minimizing false positives and false negatives, thus exemplifying its reliability for practical implementation. Moreover, the elapsed time for the defect detection process was within acceptable limits, confirming the system's efficiency for real-time applications in industrial environments.

Preprocessing Image

The preprocessing of the image involves a series of essential steps aimed at enhancing its quality and preparing it for further analysis. The process begins with converting the image from RGB to grayscale, resulting in a single-channel representation where each pixel's intensity represents its brightness. This step simplifies subsequent computations and maintains crucial image features for analysis. Following the conversion, image adjustment techniques are applied to enhance contrast and brightness, ensuring optimal visibility of image details. The adjustment process contributes to a more visually informative image and improves the accuracy of subsequent processing.

Subsequently, the image segmentation stage divides the image into meaningful and distinct regions, allowing for targeted analysis of specific areas of interest. This step is crucial for isolating objects or structures from the background, facilitating more precise feature extraction and object recognition. Finally, image filtering is employed to further refine the image by removing noise, smoothing edges, and enhancing important image features. Filtering techniques optimize the image's quality, reducing unwanted artifacts and preparing it for subsequent computer vision tasks.

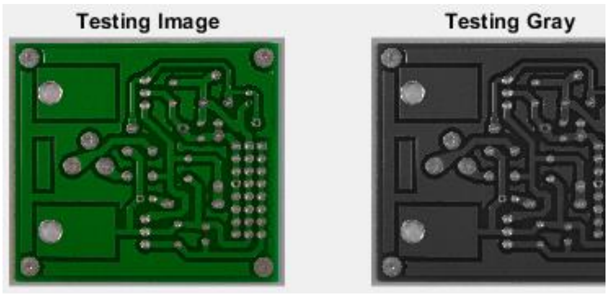

Figure 2. Image Conversion from RGB to Grayscale

The RGB images have been successfully converted into grayscale. This transformation indicates a change in the image's dimensionality, from the original 3-dimensional RGB representation to a single-channel grayscale representation, where pixel values range from 0 to 255. This conversion greatly simplifies the image and reduces its complexity, making it more amenable to subsequent image analysis. The grayscale representation retains essential image features while streamlining the data for future analysis, ultimately facilitating more straightforward and efficient image processing tasks.

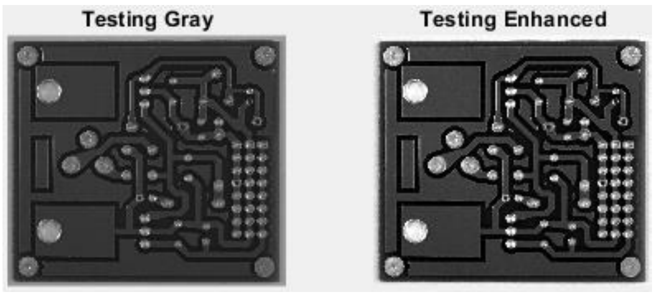

Figure 3. Grayscale Image Adjustment

The next step involves performing image adjustment on the grayscale image, as depicted in “Figure. 3”. Image adjustment is carried out to enhance or accentuate the contrast between black and white, or to improve the overall image contrast, resulting in a more detailed representation. As seen in the image, the enhanced version reveals more distinct and clearer PCB route paths compared to the original grayscale image. This enhancement will be further elucidated through the visualization of the histogram in “Figure 4”.

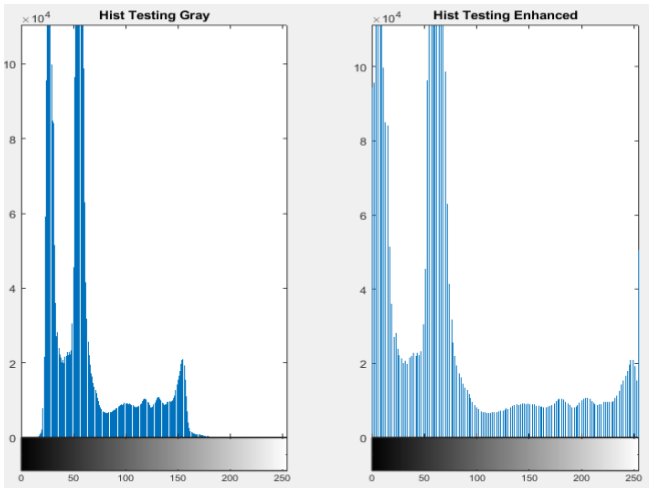

Figure 4. Histogram Grayscale Image Adjustment

The displayed histogram illustrates a noticeable increase in contrast between black and white. The grayscale image had a narrow range of values, approximately 30 to 150, resulting in less distinct contrast between black and white. However, after the adjustment process, the range is expanded from 0 to 255, significantly enhancing the contrast between black and white regions, and making it more discernible.

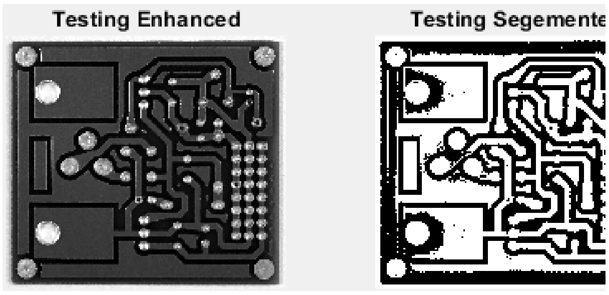

Figure 5. Image Segmentation

Image segmentation is the subsequent step represented by "Figure.5". In this process, the image undergoes a transformation where colors are converted into black or white. Notably, the segmented image no longer contains shades of gray as all colors have been converted to either black or white. The ensuing "Figure.6" will provide a more detailed depiction of the histogram.

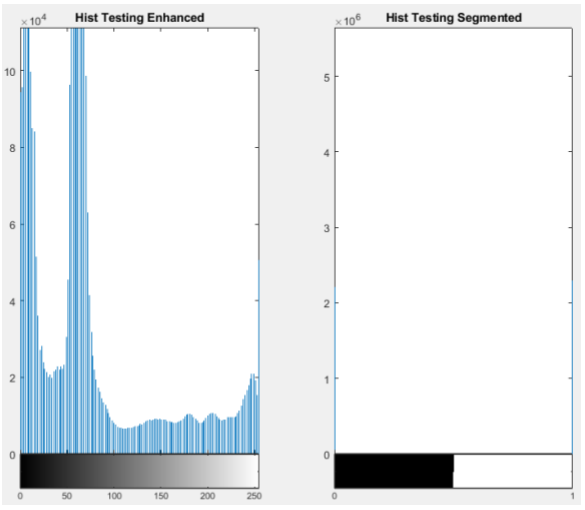

Figure 6. Histogram Image Segmentation

It is evident that the pixel values of the previous image, ranging from 0 to 255, have been transformed into binary values of 0 or 1. This conversion is performed to simplify the image and facilitate the detection process. By representing the image in binary form, the image becomes more straightforward, allowing for easier and more efficient detection procedures.

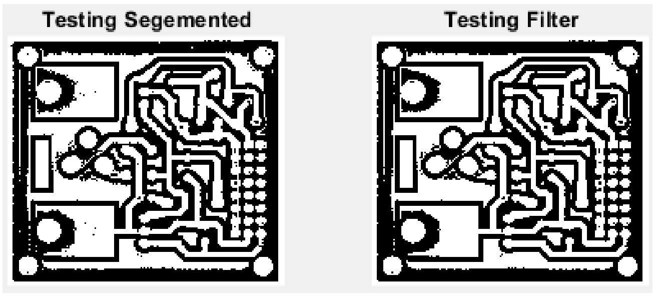

Figure 7. Image Filtering

The final step in the image preprocessing pipeline is image filtering. This process is implemented to reduce the presence of noise in the image. As observed, the filtered image exhibits significantly less noise compared to the original image. Image filtering plays a crucial role in improving the image's quality and enhancing its suitability for subsequent analysis and computer vision tasks.

Defect Detection

Defect detection can be accomplished using the following approach. Subsequently, accuracy measurement will be based on true positive and true negative evaluations. Defect_image template_image testing_image

$$Accuracy\_defect\_image = \frac{True\ Positif\ Defect + True\ Negatif\ Defect}{Positive\ Defect + Negative\ defect}$$

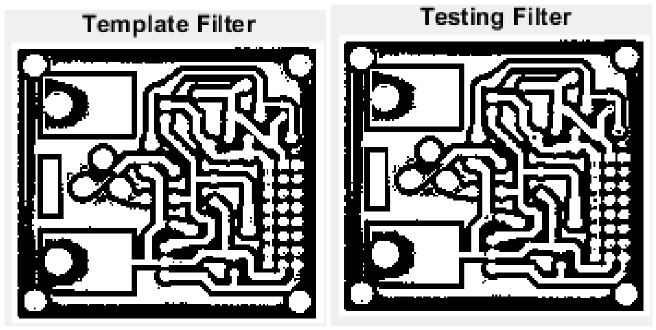

Figure 8. Template and Testing Image after Preprocessing Image

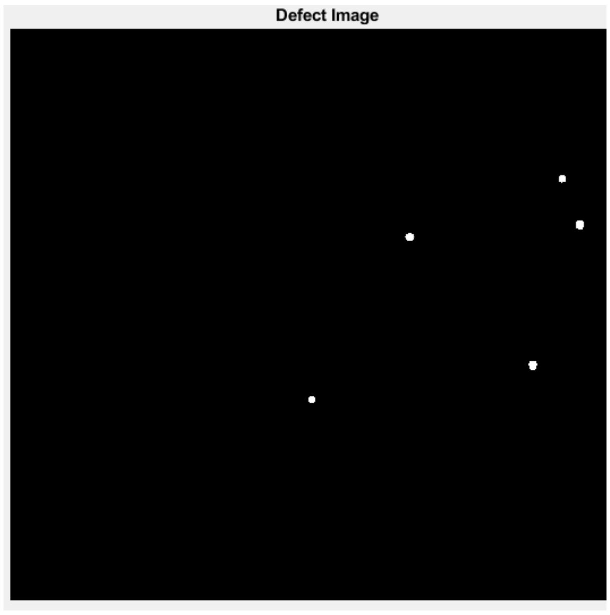

Figure 9. Defect Image

Evaluations

In this research, the scope of the investigation extends beyond printed circuit boards (PCBs) and includes other products such as nails and bolts. A total of five product images with varying dimensions have been utilized for the analysis. This approach aims to ensure a comprehensive evaluation of the proposed defect detection method across different product types, allowing for a more robust and versatile system. By incorporating multiple product images, the research aims to validate the effectiveness and adaptability of the defect detection technique in diverse industrial scenarios, enhancing its applicability and significance in quality control processes.

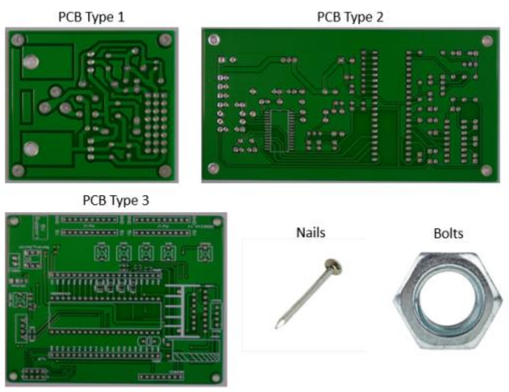

Figure 10. All Products

Table 1. All product dimensions

No	Product	Dimensions
1	PCB type 1	2016 x 2240
2	PCB type 2	1586 x 3034
3	PCB type 3	2154 x 2759
4	Nails	1389 x 1650
5	Bolts	1790 200

Table 2. All product Result

Product	Status Image	Value Max correlation	Elapsed time (seconds)	Accuracy Defect Image
PCB type 1	Defect Detected	0.9872	10.277969	100%
PCB type 2	Defect Detected	0.9589	10.459101	75%
PCB type 3	Defect Detected	0.9706	13.815118	83.3%
Nails	Defect Detected	0.9822	4.666951	100%
Bolts	Defect Detected	0.9915	10.081219	100%

Based on “Table 2”, it is evident that as the image dimensions increase, the time required to compute the correlation between the template and testing images also increases. Therefore, utilizing smaller image dimensions is preferable as it accelerates the correlation process. The max correlation value signifies the similarity between the testing and template images. A value closer to 1 indicates a higher resemblance between the two images. However, the accuracy values below 100% indicate the presence of detection errors in the defect images. Consequently, further improvements in image preprocessing are necessary to optimize the accuracy and achieve more reliable defect detection results.

4. CONCLUSION

Based on the conducted research, it can be concluded that image preprocessing, encompassing RGB to grayscale conversion, adjustment, segmentation, and filtering, is a crucial step in product defect detection. The utilization of the template matching technique has shown promising results and holds potential for further development. The correlation detection process is influenced by image dimensions, where smaller dimensions lead to faster correlation detection. The image template matching with the correlation technique exhibits versatility and can be applied to various product types. These findings highlight the importance of image preprocessing for accurate defect detection and demonstrate the potential of template matching for broader applications. By leveraging this technique, manufacturers can enhance their quality control processes and improve product standards, ultimately leading to increased customer satisfaction and product reliability.

REFERENCES

[1] Y. H. A. B. H. K. A. M. Saeed Khalilian, "PCB Defect Detection Using Denoising Convolutional Autoencoders," *IEEE Journal,* vol. 38, no. 09, 2020.

[2] K. F. T. N. S. L. Shams Ur Rehman, "Automated PCB identification and defect-detection system (APIDS)," *International Journal of Electrical and Computer Engineering,* vol. 9, no. 1, p. 297, 2019.

[3] Z. Indera Putera, "Printed Circuit Board Defect Detection Using Mathematical Morphology and MATLAB Image Processing Tools," in *International Conference on Education Technology and Computer (ICETC),* Malaysia, 2010.

[4] M. H. Abdel-Aziz I, "A real-time approach for automatic defect detection from PCBs based on SURF features and morphological operations," *Multimedia Tools and Applications,* vol. 78, no. 35, pp. 1-21, 2019.

[5] J. P. R. Nayak, K. Anitha, B. D. D. Parameshachari, R. D. Banu and P. Rashmi, "PCB Fault Detection Using Image Processing," *Materials Science and Engineering,* vol. 225, no. 1, p. 012244, 2017.

[6] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross-correlation," *Journal of SPIE,* 2021.

[7] Q. Y. W. Y. Yufan Wang, "An improved Normalized Cross Correlation algorithm for SAR image registration," in *Conference: Geoscience and Remote Sensing Symposium (IGARSS), IEEE International,* 2012.