EasyChair Preprint

№ 10226

2D X-Ray Solder Joint Segmentation Based on K-Means Clustering and Deep Learning

Chukiat Boonkorkoer, Phayung Meesad and Maleerat Maliyaem

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 22, 2023

2D X-Ray Solder Joint Segmentation based on K-Means Clustering and Deep Learning

(1)Chukiat Boonkorkoer, (2)Phayung Meesad, and (3)Maleerat Maliyaem
Faculty of Information Technology and Digital Innovation
King Monkut's University of Technology North Bangkok
1518 Pracharat 1 Rd., Wongsawang, Bangsue, Bangkok
(1)s6407011810027@email.kmutnb.ac.th, (2)phayung.m@itd.kmutnb.ac.th, (3)maleerat.m@itd.kmutnb.ac.th

***Abstract*—Identifying defective solder joints is crucial in printed circuit board assembly manufacturing, but doing so accurately can be challenging. Image segmentation techniques like the threshold method may not accurately identify defects in X-ray images of solder joints. This study aimed to identify a simple segmentation technique used for X-ray images exported from the automatic X-ray inspection machine (AXI) without compromising the accuracy of the results. The proposed approach combines the threshold method and k-means clustering to segment individual pins of the solder joint. We then use the four kinds of padding shade of segmented images from our proposal to train on the YOLOv7, a novel object detection model. When testing with a test set on X-ray images obtained from identified defective boards solder joint, the model best performs on a replicated border with the gray padding training set.**

Keywords—Solder joint segmentation, K-Means clustering, Automated X-ray inspection

I. Introduction

Identifying defective solder joints from known good ones is challenging for this study [1]. One of the crucial steps in achieving this is accurately identifying the Region of Interest (RoI) - the X-ray image of the solder joint. Various image segmentation techniques, including the Threshold Method, K-Means Clustering, and other advanced techniques [2], have been introduced to accomplish this task. Our study aimed to identify a simple segmentation technique that could be used for X-ray images exported from the Automatic X-ray Inspection machine (AXI) without compromising the accuracy of the results.

II. Problem formulation

A. Type of Solder Joint Defects

X-ray imaging typically detects defects in the Printed Circuit Board Assembly (PCBA) process, such as insufficient solder in through-holes, solder bridging on Small Outline Integrated Circuit (SOIC) and Ball Grid Array (BGA) components, and voids in solder balls of BGA [3]–[7].

Various solder joint defects can compromise the reliability and functionality of the final product in the PCBA process [5], [7]. These defects include insufficient solder due to these factors like inadequate solder paste deposition, incorrect reflow soldering temperatures or durations, or poor wetting of the solder to the pad or component lead. Solder bridging may occur due to the factors such as incorrect component alignment, excessive solder paste application, or misalignment or registration during the solder paste printing process, which hinder proper solder wetting and lead to bridge formation between adjacent pins or pads.

The defect, such as solder voids in BGA balls, on the other hand, can be attributed to various factors such as the presence of contaminants, solder voids or bubbles in the solder paste, or improper reflow soldering temperatures or duration. Other factors that can contribute to solder voids in BGA balls include the type of BGA package, solder mask roughness, stencil shape, and the design of the circuit board [8].

B. Techniques Used for Solder Joint Segmentation

After conducting experiments using the Thresholding method alone to find the solder joint contours, we found that it produced good results for the well-formed solder joints but not for some defective solder joints, such as solder bridging, including image dimming or image out-of-focus (Fig. 2a and Fig. 2b). To address this issue, we introduced a combination of the Thresholding method and K-Means clustering to solve.

III. Methodology

A. Solder Joint Collection

The solder joints obtained from the AXI machine are in segmented and unsegmented forms. The machine recombined them into a shape that resembled the component outline to facilitate human recognition. We need to split them into individual solder joints for easy model training.

The X-ray images we were collecting from the identified defective boards, defective components, and defects on each pin were recorded as Python objects, providing a structured description of the nature and location of each defect.

Figure 1. shows the proposed solder joint segmentation workflow. It outlines the process of segmenting solder joints from the X-ray images from the AXI machine.

B. Solder Joint X-ray Image Segmentation Pre-Processing

In some cases, image blurring works as noise reduction by smoothing out sharp edges and fine details. It can reduce noise and unwanted artifacts. Choosing the appropriate level of blurring can help achieve a desirable threshold image while retaining the features of an image.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

C. Finding the Contours Using OpenCV

In digital image processing, people often use the OpenCV findContours function as a popular technique for detecting contours. Many image processing applications use it to extract boundaries or "objects contours" in an image. A vector of points represents the contours, with each point corresponding to the pixel coordinates. The "findContours function" can also be used to extract hierarchical "contours" where each "contour" is a set of nested contours. Computer vision applications widely use this function for object recognition, segmentation, and shape analysis. The finding Contours technique detects object boundaries in various digital images, including X-ray images of printed circuit boards [4].

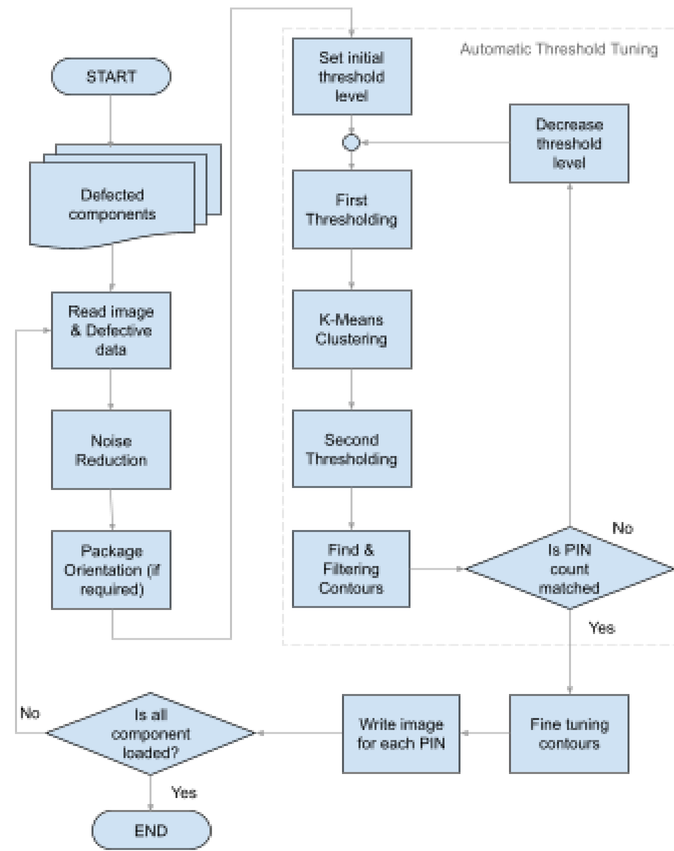


FIGURE 1
The proposed solder joint segmentation algorithm workflow

D. Combining of Threshold Technique and K-Means Clustering to Achieve the Proper Segmentation

In one instance, we encountered a pair of shorted SOIC pins in which the threshold image failed to identify individual pins due to bridging. Using the k-means clustering function, we achieved more distinct segmentation. The algorithm then created correct bounding boxes and split the image into individual pins, as shown in Figure 2. We then applied the classification to each solder joint, identifying them as either known "good" or "defective" locations. The solder joints were described as Python objects, such as {'BRD': [1, 2], 'GD': [7, 8]}, indicating that pins 1 and 2 were bridging solder joints, while pins 7 and 8 were good solder joints. Since the number of defective pins was less than the number of good pins, we selected a suitable number of good pins for the dataset balancing purpose, which we will utilize as the defect detection model dataset.

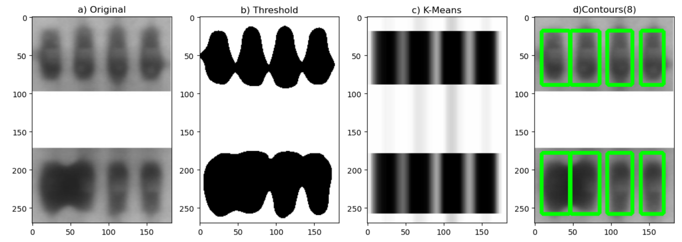


FIGURE 2
Result from each stage of segmentation

The first image (Fig. 2a) shows the original grayscale image of the solder joint with the shorted SOIC pins. The second image (Fig. 2b) displays the result after applying the thresholding method, which helps identify the pixels that belong to the solder joint but not all are located correctly. The third image (Fig. 2c) shows the result of applying the k-means clustering on the thresholded image. This process helps to separate the pixels that belong to the individual pins, allowing for the creation of bounding boxes for each pin. The fourth image (Fig. 2d) displays the original image with each pin bounding box created during the segmentation process. Note that these images illustrate the different stages of the segmentation process for a specific example. However, the actual results may vary depending on the quality of input X-ray images.

E. Automatic Threshold Tuning.

This study utilized the automatic threshold technique, which involved reverse looping from the high-level threshold. Then the detected bounding box will be compared with the known input pin count, and the loop will terminate after they determine that the pin count and the bounding box count are equal.

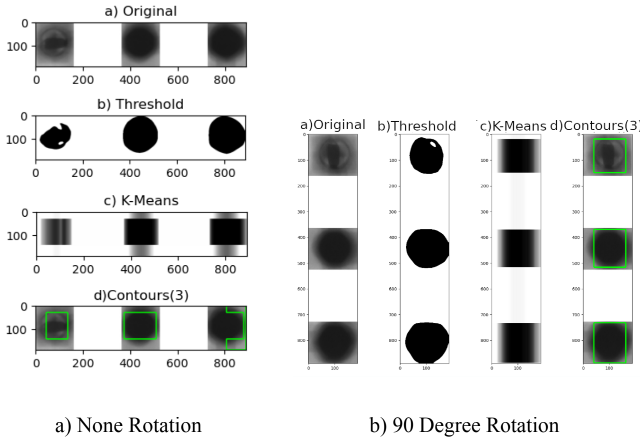
F. Investigating the Role of Background Color in Defect Detection Model Training

We encountered the issue of tall rectangular pins in some components, such as the SOIC package, which are not ideal for training a defect detection model. To mitigate this, we applied padding to the images to make them square. However, the selection of padding color could potentially impact the model's accuracy. Thus, we presented four options for the background color: the highest gray level, the replicated border with gray padding, white, or black.

G. Enhancing Pin Alignment in K-Means Clustering with the Right Angle Rotation Parameter

Based on our experimental findings, we concluded that pin alignment is a crucial factor in achieving successful results when utilizing K-Means clustering. Specifically, we observed that one alignment of the pins resulted in better performance than the other (Fig. 3). Therefore, to address the issue of poor segmentation results, we introduced a "Right Angle Rotation"

parameter that allows for a 90-degree rotation of the package. This rotation option enables users to adjust the orientation of the image and potentially improve segmentation performance.



a) None Rotation b) 90 Degree Rotation

FIGURE 3
Comparison of normal and 90 degree rotation result

H. Data Augmentation Techniques

Due to the rarity of defective images, we recognized the need to prevent an imbalanced dataset. To achieve this, we employed data augmentation techniques. Specifically, we utilized sampling rotation, flipping, brightness adjustment, and padding with various background options on the original images. We applied these techniques to help to increase the diversity of the dataset and improve the model's ability to detect defects accurately.

To tackle the challenge of imbalanced data, we employed data augmentation techniques by increasing the number of samples. We augmented the initial dataset of 241 solder joints, consisting of 134 normal solder joints, 44 joints with insufficient solder, 31 joints with solder bridging, and 32 joints with voids in BGA balls, by a factor of 20.

I. Improving SOIC Bridging Detection through Adaptation of Segmentation Methodology

Our initial segmentation approach was to separate the solder joints individually, which failed to detect SOIC bridging in the original X-ray image. To overcome this limitation, we adapted our algorithm to verify the long rectangular solder joint and expand the region of interest (RoI) to include neighboring solder joints. This modification allowed the model to be trained on a "wider view" of the solder joints, resulting in improved detection performance for SOIC bridging.

J. Splitting the Dataset into Training, Validation and Testing Set

In this experiment, we split the dataset into a training set, a validation set, and a test set. The split ratios were 70%, 20%, 10% for training, validation, and testing, respectively. We used the ‘training’ set to train the model, the validation set to tune the model's hyperparameters and avoid overfitting, and the test set to evaluate the final performance of the model. By splitting the dataset in this way, we can ensure that the model's performance evaluation is on unseen data and that it was not simply memorizing the training set.

K. Labeling for YOLO, the *Object Detection Model*

YOLO (You Only Look Once) is an object detection algorithm that divides an input image into a grid of cells and predicts bounding boxes and class probabilities for containing objects within each cell. We must label the dataset with bounding boxes and corresponding object classes before training a YOLO model.

Labeling for YOLO involves creating annotation files that contain information about each object in the image. Each annotation file corresponds to the specific image and has one row for each object in the target image. Each row includes the object class and its bounding box coordinates, represented as normalized values between 0 and 1 relative to the size of the target image.

For the YOLO object detection, we defined four classes in this study: GD (0-normal joint), INS (1-insufficient solder joint), BRD (2-pin shorting), and BVD (3-void in BGA ball). The segmentation process located the solder joint at the center and applied the surrounding padding. Hence, we annotated each image in the format of "C 0.500000 0.500000 0.500000 0.500000". The "C" value represents the class label, in which "0" is the "GD" class, "1" is the "INS" class, "2" is the "BRD" class, and "3" is the "BVD" class. The values "0.500000 0.500000" represent the center of the bounding box, and the next "0.500000 0.500000" is the size of the bounding box, which was half of the image size.

L. Train the YOLO v7 Object Detection Model

To evaluate the effectiveness of our solder joint segmentation results, we employed them as a dataset for training the YOLO v7 model. The YOLO v7 model was pre-trained with the COCO image dataset and used for this purpose.

IV. Results and Discussion

A. The YOLOv7 Training Result

From the results in Table I, we notice that the black background provides the highest precision and recall values. However, when considering the "mean Average Precision" ("mAP"), which is the averaging of the "average precision" (AP) scores for each class over a range of Intersections over Union (IoU) thresholds, the back background gives a mAP_0.5 value of 0.996690, which is the highest among all the other applied background colors. However, the mAP_0.5:0.95 value is 0.977091, the lowest among all the others. This result indicates that the model may not be able to detect objects with various sizes and positions, and additional datasets containing a variety of solder joint sizes may be required to improve the model's performance.

TABLE I. The training result of different background color

Background	**Matrix**			
	mAP_0.5	***mAP_0.5:0.95***	***precision***	***recall***
Black	0.9968	0.9771	0.9964	0.9939

Background	Matrix			
	mAP_0.5	***mAP_0.5:0.95***	***precision***	***recall***
White	0.9922	0.9862	0.9592	0.9849
Gray	0.9954	0.9936	0.9906	0.9919
Replicated	0.9954	0.9844	0.9934	0.9842

According to the "F1", which considers both precision and recall, it shows that the black, gray, and replicated border with a gray background yielded the highest scores. These scores were 1.0 at a threshold of 0.479, 0.99 at a confidence threshold of 0.338, and 0.99 at a confidence threshold of 0.387. In contrast, the white background resulted in the lowest score of 0.97 at a confidence threshold of 0.614. Furthermore, when examining the confidence bandwidth, we noticed that the black background provided the flattest and highest F1 value along the confidence threshold range.

The black background also resulted in the highest accuracy for correctly predicting good solder joints, insufficient solder joints, bridging solder joints, and BGA ball voids at 0.99%, 1.0%, 0.99%, and 1.0%, respectively. However, it did make some incorrect predictions for the background as good and bridging joints at 0.5%. This issue could be due to the solder joint cropping being too narrow, which sometimes only captured the dark area of the joint without its surroundings.

B. Prediction of The Actual Image from the AXI

As previously mentioned regarding the four various padding shades, we used them to train the model and then tested it on the separately prepared dataset. We crop the exported image from the AXI machine to make it a suitable size for testing without further manipulation. The testing result is shown in TABLE II.

TABLE IIa. Applying black border with black blackground

Black	BRD	INS	BVD	ALL
Recall	0.5714	0.9630	0.4667	0.7000
Precision	0.9412	0.9630	1.0000	0.9608
F1	0.7111	0.9630	0.6364	0.8099

TABLE IIb. Applying white border with white blackground

White	BRD	INS	BVD	ALL
Recall	0.5714	0.7037	1.0000	0.7143
Precision	0.8421	0.9048	0.6818	0.8065
F1	0.6809	0.7917	0.8108	0.7576

TABLE IIc. Applying gray border with gray blackground

Gray	BRD	INS	BVD	ALL
Recall	0.6786	0.9630	1.0000	0.8571
Precision	0.7037	0.9630	1.0000	0.8696
F1	0.6909	0.9630	1.0000	0.8633

TABLE IId. Applying replicated border with gray blackground

Rep	BRD	INS	BVD	ALL
Recall	0.6429	1.0000	1.0000	0.8571
Precision	0.9000	0.8438	0.8824	0.8696
F1	0.7500	0.9153	0.9375	0.8633

The gray background (TABLE IIc) and the replicated border with gray padding (TABLE IId) give the same "overall F1" score. However, almost all models predict well for "Insufficient solder" detection but not for bridging. Therefore, we select the replicated border with gray padding, which provides the best score for bridging, as the winning model.

C. Conclusion and considerations for Multi-Slice Image Analysis in Solder Joint Inspection

In this study, we analyzed the images obtained from the AXI machine, which comes with varying slices for a single solder joint. There are four slices for each through-hole solder joint, three for BGA ball inspection, and only one for the solder bridging. In real-world defect-detecting scenarios, it is difficult to determine the defect from individual X-ray slices. The solder joint inspection requires the combination of all its image slices. Therefore, it is necessary to conduct further research on this subject to improve the overall model accuracy.

Although the achieved segmentation may not be optimal, it adequately prepares the x-ray image solder joints for object detection training. Future research will include conducting multiple runs with various random splits to mitigate bias and enhance the results.

References

[1] H. Jayasekara, Q. Zhang, C. Yuen, M. Zhang, C.-W. Woo, and J. Low, “Detecting Anomalous Solder Joints in Multi-sliced PCB X-ray Images: A Deep Learning Based Approach,” *SN Comput. Sci.*, vol. 4, no. 3, p. 307, Apr. 2023, doi: 10.1007/s42979-023-01765-6.

[2] “Fig Fruit Image Segmentation using Threshold, K-means Clustering, and Sharp U-Net Techniques.” https://ieeexplore.ieee.org/document/9935411/?arnumber=9935411 (accessed Apr. 08, 2023).

[3] Q. Zhang *et al.*, “Deep Learning Based Defect Detection for Solder Joints on Industrial X-Ray Circuit Board Images.” arXiv, Mar. 25, 2021. Accessed: Feb. 23, 2023. [Online]. Available: http://arxiv.org/abs/2008.02604

[4] C. AKDENİZ, Z. Dokur, and T. Ölmez, “Detection of BGA solder defects from X-ray images using deep neural network,” *Turk. J. Electr. Eng. Comput. Sci.*, vol. 28, pp. 2020–2029, Jul. 2020, doi: 10.3906/elk-1910-135.

[5] M.-C. Chen *et al.*, “A PCBA Solder Joint Defects Inspection System Based on Deep Learning Technology,” in *2023 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2023, pp. 1–3. doi: 10.1109/ICCE56470.2023.10043589.

[6] S. Liao, C. Huang, Y. Liang, H. Zhang, and S. Liu, “Solder Joint Defect Inspection Method Based on ConvNeXt-YOLOX,” *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 12, no. 11, pp. 1890–1898, Nov. 2022, doi: 10.1109/TCPMT.2022.3224997.

[7] J. Zhou, G. Li, R. Wang, R. Chen, and S. Luo, “A Novel Contrastive Self-Supervised Learning Framework for Solving Data Imbalance in Solder Joint Defect Detection.,” *Entropy*, vol. 25, no. 2, p. 268, Feb. 2023, doi: 10.3390/e25020268.

[8] M. Kozak, P. Vesely, and K. Dusek, “Analysis of solder mask roughness and stencil shape influence on void formation in solder joints,” *Weld. World*, vol. 67, no. 5, pp. 1347–1355, May 2023, doi: 10.1007/s40194-023-01505-7.