Advanced Engineering Informatics 43 (2020) 101004

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Full length article

Soldering defect detection in automatic optical inspection

Wenting Dai[a,*], Abdul Mujeeb[b], Marius Erdt[c], Alexei Sourin[a]

[a] School of Computer Science & Engineering, Nanyang Technological University, Singapore
[b] School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
[c] Fraunhofer Research Center, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Automatic Optical Inspection (AOI)
Localization and classification of solder joint defects
Semi-supervised learning
YOLO
Clustering
Active learning

ABSTRACT

This paper proposes an integrated detection framework of solder joint defects in the context of Automatic Optical Inspection (AOI) of Printed Circuit Boards (PCBs). Both localization and classifications tasks were considered. For the localization part, in contrast to the existing methods that are highly specified for particular PCBs, we used a generic deep learning method which can be easily ported to different configurations of PCBs and soldering technologies and also gives real-time speed and high accuracy. For the classification part, an active learning method was proposed to reduce the labeling workload when a large labeled training database is not easily available because it requires domain-specified knowledge. The experiments show that the localization method is fast and accurate. In addition, high accuracy with only minimal user input was achieved in the classification framework on two different datasets. The results also demonstrated that our method outperforms three other active learning benchmarks.

1. Introduction

In Printed Circuit Boards (PCBs) assembly, Surface Mount Technology (SMT) is widely used. Three steps are commonly included [1]: (1) Solder printing: machine prints solder paste (fusible metal alloy) on the surface of the board for connecting the components and boards; (2) Picking and placing: components are properly placed on the board; and

(3) Reflowing: the solder joints take shape by reflowing of the solder paste.

During the production of PCBs, various defects may occur at each stage. For example, at the second stage, defect cases of missing, wrong, or doubled components may occur. In terms of possible soldering defects, most of them happen after the reflowing stage, such as the defects at the IC package components (Pseudo joint, Excess solder, Insufficient solder, Shifting Solder bridge) and the defects at the non-IC components (Side termination, Tombstoned components, Raised components, Pseudo joint, Excess solder, Insufficient solder, Shifting, Solder bridge).

In modern factories, *basic electric tests* such as the in-circuit test and the *functional board test* like the manual visual inspection [2] are still commonly used. But, due to the miniature size of the joints and due to huge numbers of possible defects, conventional inspection methods reached its limits. Therefore, more attention is paid to Automatic Optical Inspection (AOI).

There are three steps included in AOI, which are (1) image acquiring, (2) solder joint localization, and (3) classification. In our research project, we focused on the last two steps, because capturing had already been performed by an RGB camera with a three-color tiered illumination system placed above the production station [3]. After that, a front-end process—the localization of solder joints—must be completed before defect classification. Typically, the solder joints are outlined by rectangles which are called the regions of interest (ROI), as it is illustrated in Fig. 1. Then, commonly, tedious manual localization of solder joints is used based on conventional image processing algorithms customized for specific soldering technologies and particular PCBs. In contrast, we were motivated to propose a generic deep learning solution to the localization problem, which can be used with real-time speed for different configurations of PCBs and soldering technologies.

Regarding the second task, since learning-based methods achieved excellent performance in many computer vision tasks previously, they have been widely adopted in AOI. However, insufficiently annotated datasets impact their performance. Simultaneously, annotating thousands of images is usually costly, time-consuming, and often requires domain specified knowledge. Therefore, we propose a novel method of combining active learning and semi-supervised learning. It is not restricted to the active suggestion of relevant samples for manual annotation, but it is also able to label parts of the unlabeled data automatically.

* Corresponding author.
E-mail address: daiw0004@e.ntu.edu.sg (W. Dai).

https://doi.org/10.1016/j.aei.2019.101004
Received 12 February 2019; Received in revised form 7 August 2019; Accepted 26 September 2019

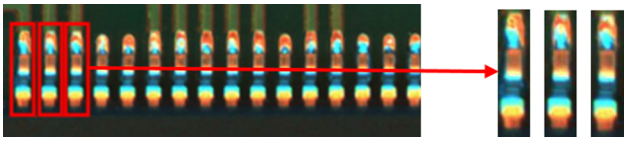

Fig. 1. Individual solder joint segmentation.

We review existing related methods in Section 2. We also state the motivation and hypothesis of solder joint localization and classification. The details of the proposed method are presented in Section 3, followed by the validation experiments in Section 4. Eventually, Section 5 concludes the paper.

2. Related works and their motivations

2.1. Existing solder joint localization methods

The existing solder joint localization methods are based on three groups of image processing algorithms using (1) template-matching, (2) image histogram, and (3) clustering of pixels.

Template matching is an approach used in image processing for identifying small parts of the tested image by matching with the predefined templates. The templates are created based on qualified samples, and then the similarity values at each position of the tested image will be calculated by sliding the templates. Thus, for example, [4] uses the normalized cross-correlation as the measure of similarity, which is directly calculated based on pixel intensities of PCB images. [5] utilizes template matching in IC images and extracts the edge features as the input by taking the inconsistent shape and non-uniform color distribution into account. However, template matching approaches are very sensitive to small differences in the appearance of the target structure, which cannot be avoided when manufacturing solder joints as they are tiny and the manufacturing process is not entirely consistent. It also becomes inefficient when the large computational cost of a global traversal search for each position is incurred.

Many *histogram-based methods* use a calculated global threshold to identify the ROIs in PCB images. Thus, [6] uses the mean and standard deviation of pixel values to compute the threshold. The adaptive threshold is more commonly used. [7] and [8] use the Otsu method to calculate the optimum threshold for pre-processed IC images using the bi-modal histogram. Except for one global threshold, [9] measures the threshold values for each individual pixel against their neighborhood. However, the histogram-based methods cannot work well when other highlighted components are included, such as markings and via-holds.

Clustering of pixels constitutes the third category of methods to identify solder joints. A multilevel thresholding approach in [10,11] clusters the highlighted pixels of the PCB images according to their secularity properties. After that, the components are recognized based on the color distribution measured by the Gaussian Mixture Model. However, it is time-consuming and makes real-time identification problematic.

We concluded that there are no methods that are able to provide a generic real-time solution which can be easily ported to different configurations of PCBs and soldering technologies. In addition, the solder images may also vary significantly due to image capturing conditions (e.g., lighting). Furthermore, the electronic components on the PCB images introduce additional complexity to the task of solder joints detection.

2.2. Existing solder joint classification methods

The existing classification methods for solder joints can be categorized into two groups based on the types of components they inspect. They are solder joint classification for non-Integrated Circuits (non-IC) component [12–18] and IC package [19–22].

2.2.1. Non-IC solder joints classification methods

In these methods, for acquiring more useful information features are often extracted from several sub-areas after region division methods. Then, several types of algorithms are used. The first type is *ensemble learning*. A Random Forest-based classification algorithm based on the color and geometry was presented in [12]. In [13], an inspection algorithm first trained several AdaBoost classifiers and made the decision based on a classification and regression tree. A similar boosting inspection method was also developed in [14], which integrates the results of specific single-layer artificial neural networks. For solder joint inspection, another popular choice is Support Vector Machine (SVM). A two-stage classifier used a Bayes classifier and a multi-class SVM in [15]. Another approach combining a neural network with the genetic algorithm was proposed in [16]. Besides, a prototype-based supervised algorithm named Learning vector quantization (LVQ) was also adopted. Instead of only feeding the extracted features from orthogonal view direction images in LVQ, like in [17], solder joint images from an oblique view are also used where special equipment (mirror pyramid) is needed in [18].

2.2.2. IC solder joints classification methods

Since inspecting defects on the smaller IC solder joints is more difficult than that on the non-IC solder joints, the number of relevant publications is relatively small. In paper [19] fuzzy rules combined the independent results of several LVQ classifiers on sub-regions. A Multiple Neural Network System presented in [20] feeds the extracted features into an MLP and an LVQ. If their results are inconsistent, a new MLP will reclassify it based on the new extracted wavelet feature together with previous features. Model-based methods are also commonly used. An appearance model-based method trains its qualified model on the low-rank components after a Robust Principle Component Analysis decomposition in [21]. Cai et al. [22] proposed a statistical model-based method based on an improved visual background extraction algorithm. Deep learning-based approaches are also applied in this area. Paper [23] adopts three cascaded convolutional neural networks (CNNs) where a CNN first learns the ROIs of solder image, and the other two classify the results in the learned ROIs and the entries image.

Some of the mentioned approaches gained satisfactory performances, while they are not free from drawbacks. The feature-based methods ([12–16,20]) are hard to generalize for both solder joints of IC or non-IC components, even with sufficient dataset used. Because their hand-crafted features are only designed for one type of solder joint, and a specific region division is included. As for the model-based approaches ([17–19,21,22]), they do not face problems caused by feature extraction, while they cannot work well if the samples are not correctly registered and need location prior knowledge that limits the generalization ability to different solder joint types.

Moreover, attaining labels in many real applications is not an easy task. However, satisfactory performance of the supervised classifier, especially for deep learning-based methods, cannot be guaranteed if labeled samples are insufficient. Therefore, if the algorithm is able to annotate parts of the samples automatically and suggests a little number of irredundant and informative samples to query annotators, the amount of labeling workload will be significantly decreased. However, they do not show this ability.

2.3. Research hypothesis

We then studied whether machine learning algorithms could be useful for solving our problem. Machine learning methods have been successfully used in various domains to detect objects of fuzzy shapes in busy backgrounds. [24–26] are examples of using deep learning-based object detection methods to identify humans, animals, and man-made objects. Inspired by the success of these methods, we decided to apply deep learning for automatic solder joint detection. As for solder joint classification, we had to find out if there would be a machine learning-

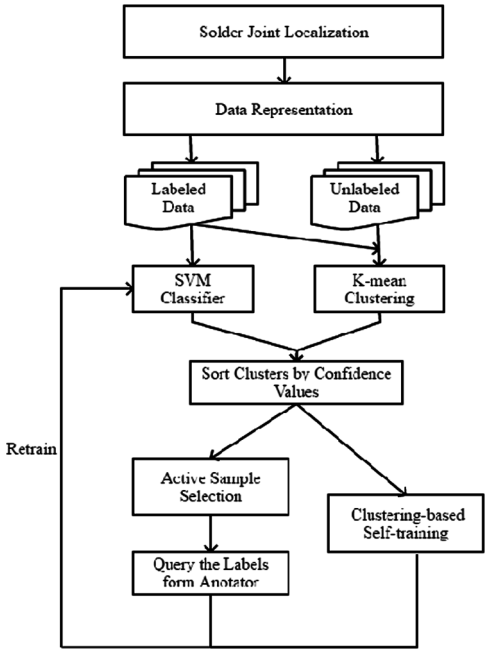

Fig. 2. The framework of the localization and clustering-based active semi-supervised classification.

based approach, including active learning, semi-supervised learning, and a general image representation method.

Therefore, we proposed to work on an integrated detection framework that consists of localization and classification of solder joints. At the first stage, the ROI of solder joints are localized by a deep convolutional neural network (ConvNet)-based method that can be easily ported to different configurations of PCBs and can work quickly. Secondly, we use a clustering-based classification method in which a deep ConvNet, trained on a large dataset, extracts general features based on the concept of transfer learning. Active and semi-supervised learning are integrated for reducing the annotation workload. Furthermore, to provide a more helpful suggestion, it takes data distribution into account. The underlying clustering structure of the dataset computed by the K-means method is adopted in our proposed method.

3. The proposed method

The integrated defect detection framework is illustrated in Fig. 2. It begins at the localization of individual solder joints. The small joints will be transformed into a well-classified representation space. After that, an SVM classifier is used based on the small portion of the labeled samples. Simultaneously, K-means clustering analyzes the clustering structure of the whole dataset.

According to two different types of learning rules, parts of the unlabeled samples are automatically and manually annotated, respectively. After that, the classifier is retrained on the enlarged labeled dataset followed by new samples annotation iteration. Finally, it stops when meeting the stopping criterion. The details of the proposed method are introduced step by step in the subsequent subsections.

3.1. Solder joint localization

Inspired by the current outstanding results of object detection, we considered the deep ConvNet-based methods that can be easily ported to different configurations of PCBs and can work quickly. Amongst them, the YOLO algorithm [26] was selected due to its excellent accuracy and speed. Even though there are other methods which have shown similar accuracy on various datasets, YOLO shows the fastest speed in many object detection applications. Before the invention of YOLO, the previous detection algorithms usually separate the task into two parts: region proposal and classification of these multiple proposals. Finally, the high probability regions will be treated as the detection results of objects. Oppositely, YOLO is an end-to-end detection network. It divides images into a grid and predicts the bounding boxes and class probability for each grid cell. Therefore, it makes the method extremely faster at the testing phase compared to other approaches. Regarding our localization problem, there are only two classes in total, which are solder joint and PCBs' background.

Because YOLO is a supervised method, the training dataset should be first manually created, which would contain bounding box information of solders joints. Four numbers are used to descript the bounding box information: Center X, Center Y, Width and Height. Before training, one parameter has to be changed. Because YOLO is known to be less accurate for the detection of small objects such as the solder joints we are working on, the image size in the configuration file was increased from 416 to 1024 pixels. Then, the image sizes will be automatically adjusted to this value. The training took two days to complete 20,000 iterations on an NVIDIA V100 GPU. After that, the gained model localizes the solder joints for testing PCB images in real time. The results are shown in Fig. 3.

3.2. Data representation

Data representation includes two steps, which are feature extraction and dimension reduction. In order to represent the data in a more generic way, transfer learning is used in feature extraction. Commonly, it transfers the knowledge learned from the source domain to the target domain. Paper [27] and [28] proved that in small datasets pre-trained CNNs could transfer the mid-level features learned on large-scale datasets into AOI tasks even without fine-tuning. Thus, we used a pre-trained VGG-16 model [29], which consists of 13 convolutional layers and three fully-connected layers, as the generic feature extractor. Through this way, the dimensions of the extracted feature vector will be 4,096 for each sample.

However, the performance of the following clustering algorithm will probably be impacted by such huge dimensions, because relatively low-dimensional data is more suitable to K-means. Therefore, for decreasing the feature dimensions, Principal Component Analysis (PCA) will first orthogonally transform the data to a new coordinate system where all variables are uncorrelated and then discards the small variances principal components [30].

3.3. Clustering analysis

For taking the underlying data distribution into account, K-means method measures the clustering structure of the whole dataset. The dataset $X = \{x_1, x_2, \cdots, x_N\}$ is separated into K clusters $S = \{S_1, S_2, \cdots, S_K\}$ where each sample belongs to the cluster S_* whose centroid C_* is closest to the sample. At the same time, an SVM classifier trained on the labeled data predicts the classes for all unlabeled samples. According to the results, the confidence to label the samples within each cluster can

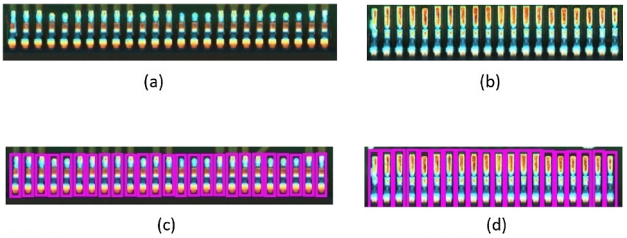

Fig. 3. Two PCBs with different layouts and solder patterns. (a) and (b) original PCBs images, (c) and (d) solder bounding boxes.

be evaluated. Here, it uses a value called homogeneity [31] which calculates the variance of the label distribution. With reference to the binary classification problem, it is defined as:

$$\text{homogeneity}(S_i) = \frac{1}{N_i}|n_p^i - n_n^i| \quad (1)$$

where n_p^i and n_n^i are the numbers of the predicted positive and negative samples in this cluster S_i, N_i is the size of the cluster. The range of homogeneity is between 0 and 1. Intuitively, if positive and negative labels are equally probable in a cluster, it means that in the data space it covers a more difficult part to classify than a cluster having a very skewed distribution, in which there is a clear predominant class value [31].

3.4. Active learning

Active learning, also called the sample-query-suggestion algorithm, tries to suggest classifiers with parts of unlabeled samples for annotation actively when the majority of samples is unlabeled and there is a lack of labeled data. It is based on the prediction of the current classifier to select a small part of "informative" samples and query their labels for human annotators. The performance of classifier will be improved after getting more labeled samples. Therefore, training a classifier with high accuracy but only annotating a small number of unlabeled samples becomes possible.

We used the *uncertainty sampling* strategy. It hypothesizes that getting the labels of the least certain samples is more helpful to improve the classifier's performance than others. Based on this rule, the least r_a certain clusters (decided by their confidence values) are select at first. From them, since SVM classifier is non-probabilistic, active samples having the smallest distance to the decision hyperplane will be suggested $\text{Value}(x_i) = w\bullet\text{certainty}(x_i) + (1 - w)\bullet\text{representative}(x_i, C_*)$[32]. $\text{Value}(x_i) = w\bullet\text{certainty}(x_i) + (1 - w)\bullet\text{representative}(x_i, C_*)$.

3.5. Semi-supervised learning

For taking advantage of hidden information in the unlabeled dataset, semi-supervised learning methods train their classifiers from both labeled and unlabeled datasets. Instead of manual labeling, it is able to automatically label unannotated samples based on certain rules for improving the performance. Many approaches were proposed in this area. Fung and Mangasarian [33] proposed to recast semi-supervised SVM as a concave minimization that can be quickly solved by a few linear programs. However, the results are only proved in the linear case. An SVM-based relevance feedback scheme [34] was proposed for content-based image retrieval, which adopts a semi-supervised biased maximum margin analysis to combine the distinct properties of two types of feedback and learn from the unlabeled data. Here, we used one of the methods named self-training method. Conventionally, the unannotated samples having the highest confidence values will be directly labeled as their predicted class labels by the current classifier. Since the increase of the labeled data, the classifier will achieve higher performance. In addition, paper [35] analyzes the results of semi-supervised classifier based on its clustering structure. It proved that the distributions of labeled samples in clusters would affect the result. Inspired by this, we considered the clustering structure as well. First, the most r_s confident clusters are selected. Within each cluster, the unannotated samples with the largest certainty values (distance to decision hyperplane) are annotated as their predicted labels. After these two newly obtained datasets (D_A and D_S) are added into the labeled dataset, retaining of the current SVM classifier begins.

3.6. Re-clustering

Because the result of K-means algorithm is not very accurate, an excellent result of measured clustering structure cannot be guaranteed.

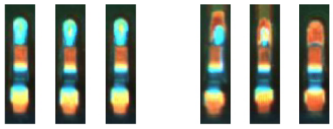

Fig. 4. Qualified (a, b, c) and insufiicient (d, e, f) solder joint samples in dataset_1.

For releasing the influence of the included errors, after certain iterations, our method will re-cluster the whole dataset.

4. Validation

Here, we first evaluate the performance of the localization algorithm and then validate the proposed classification method. At the first part, because an implementation of YOLO released by its authors is available, our starting point is the original code YOLOv3. 30 PCB images were used in the training of YOLO, and 110 solder joints for each PCB. In the testing set, 10 different PCBs with 1100 individual bins are included. After getting the localization model, we used it to create datasets for classification framework.

We focused on two solder joint datasets: they are insufficient solder (dataset_1) and shifting solder (dataset_2). The insufficient solder joints usually happen when the amount of solder paste printed on PCB is much less than the requirement like it is shown in Fig. 3. The shifting of solder joints, as in Fig. 4, happens when their components are not placed exactly where they are supposed to be. There are 2,610 positive and 2,427 negative in dataset_1, while 1,276 and 1,267 are in dataset_2. Each sample is represented as a 1×500 vector after feature extraction and dimension reduction (see Fig. 5).

We then trained the K-means clustering. Because our goal is measuring the distribution of the dataset, it is better to set K to be larger than the actual number of classes (15 is used in all experiments). In addition, both the numbers of labeled samples from active learning and self-training are also 15 per iteration. After every five iterations, we will re-clustered the dataset and finally averaged 20-run results. According to the size of our datasets, the setting of dataset splitting is 70% vs. 30%. Furthermore, for simulating the real application scenario, 10% of data in the training set are labeled at the beginning. The training stage of classification will stop until the manually annotated data grown to 40% of the whole dataset for saving the computational cost.

4.1. Influence of the configuration of image resolution in YOLO

The original resolution of PCB images used is $1{,}261 \times 1{,}331$ pixels. They have to be scaled to the configured value during testing and training. The original setting of YOLO is 416x416. We compared the detection performances by two evaluation matrices that are Intersection of Union (IoU) and mean Average Precision (mAP) and processing time under different configurations of image resolution. The results are shown in Table 1. We found that the performance of YOLO for the detection of small objects are not satisfactory when using its initial resolution setting. Therefore, we increased the resolution to 1024×1024, and the accuracy then grew as it was expected. Even though the processing time also increased, it still meets the requirement.

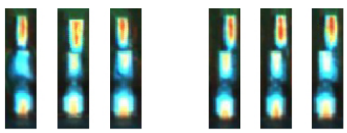

Fig. 5. Qualified (a, b, c) and shifting (d, e, f) solder joint samples in dataset_2.

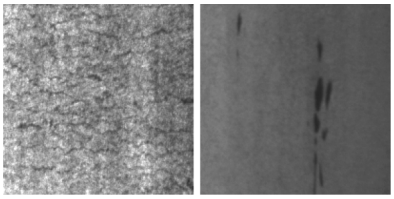

Fig. 6. Inclusion defect and crazing defect in NEU surface defect dataset.

Table 1
Accuracy and speed under different image resolution.

Image resolution	IoU	mAP	Processing time
1024 × 1024	0.836	0.951	0.34
512 × 512	0.822	0.884	0.27
416 × 416	0.318	0.367	0.25

4.2. Results of the different numbers of selected clusters in active learning

The performances of choosing a fixed number (15) of samples for active learning in different numbers of clusters were compared. Taking $r_a = 15$ as an example, it means one sample will be chosen in each cluster. The experiment results are shown in Fig. 7. We can find different optimal strategies achieved on two datasets. For dataset_1, the largest r_a obtains the best performance, while the best choice for dataset_2 is relatively small. It is probably because in dataset_1 insufficient solder defect has various appearances, which it is more irregular than the shifting solder defect, therefore getting an accurate unsupervised clustering result in this dataset is much harder.

4.3. Results of the different numbers of selected clusters in self-training

For semi-supervised learning processing, we similarly evaluate the influence of $r_s$different numbers of the selected clusters (r_s), whose result is shown in Fig. 8. Even though self-training benefits from saving the annotation workload, it also causes inevitable degradation of the performance by the incorrectly labeled samples. Thus, the average error rate of automatically labeled samples over 20 runs are compared in Table 2. From these experiments, we concluded that if r_s is small, the samples that are incorrectly labeled by self-training can be largely avoided.

4.4. Comparisons with benchmarks

The proposed active and semi-supervised classification method is also compared with three different benchmarks. For gaining more information, Precision and Recall are adopted rather than classification accuracy, which are shown in Fig. 9. For fair comparisons, random, representative [32] and uncertainty samplings [36] are selected, because of the similar SVM classifier they used. Samples which are being closest to the current decision boundary are suggested by uncertainty sampling, whereas representative sampling queries the cluster medoids within the margin of the SVM classifier. The numbers of manual labeling samples are also 15 in the other three benchmarks. At the same time, the values of areas under types of curves are also compared in Table 3.

In dataset 1, larger values are achieved in precision, and the differences between them are small. Unlikely, our method outperforms random sampling and representative sampling significantly in recall value at 4.65% and 3.40%, while slightly better results are achieved by our method than uncertainty sampling method.

Oppositely, dataset 2 shows a different trend. Specifically, their precisions that have large variations inside are smaller than recalls. Overall, compared with these benchmarks, our method achieves the best result in both datasets.

4.5. Experiment on a public dataset

To evaluate the generalization capability of the proposed method, we also made comparisons with other benchmarks in public dataset. We used NEU surface defect dataset that collects six classes of defects on the metal surface, which is introduced in [37]. For comparison, we select two types of defects (inclusion and crazing) for classification. Two examples are illustrated in Fig. 6. After training, the gain results are shown in Table 4. It shows that the proposed method achieved a relative result with uncertainty sampling method, whereas it still outperforms the other two methods with large margins.

5. Conclusion

We proposed to use the deep learning method YOLO to locate hundreds of small and closely spaced solder joints in PCB images automatically. Our experiment shows that it can be easily ported to different PCB layouts and also achieved accurate detection results with a very high speed of less than 0.34 s per PCB image (110 solder joints).

Then, based on active and semi-supervised learning concepts, a novel solder defects classification framework was proposed. During the training phase, parts of unannotated samples are automatically labeled with a small error rate (less than 1.5% of all automatically labeled samples), at the same time irredundant and informative samples can also be suggested to human annotators. The results proved that the performance of the classifier is continuously improved, but the annotation cost will be largely reduced compared with the other active learning strategies. Thus, the proposed framework has the potential to be applied in real-world AOI solder joint inspection settings. However, it only can solve the two-class (qualified and defective) classification now. In the future, we will generalize the algorithm for detecting more specific types of defects.

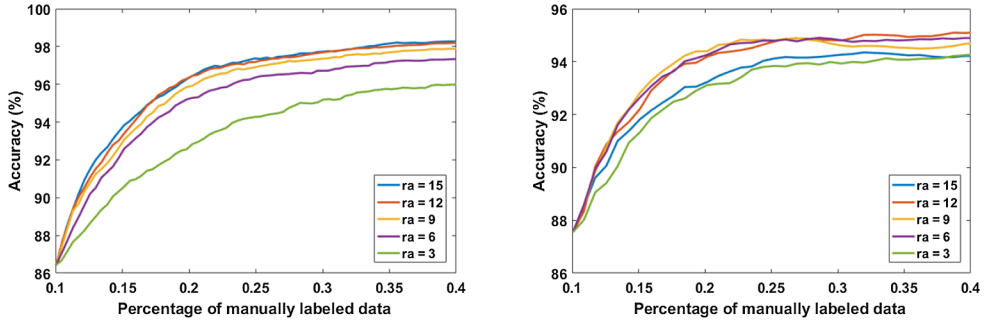

Fig. 7. Comparing the change of accuracies of the proposed method under different numbers of selecting clusters in active learning on (a) dataset_1 (b) dataset_2.

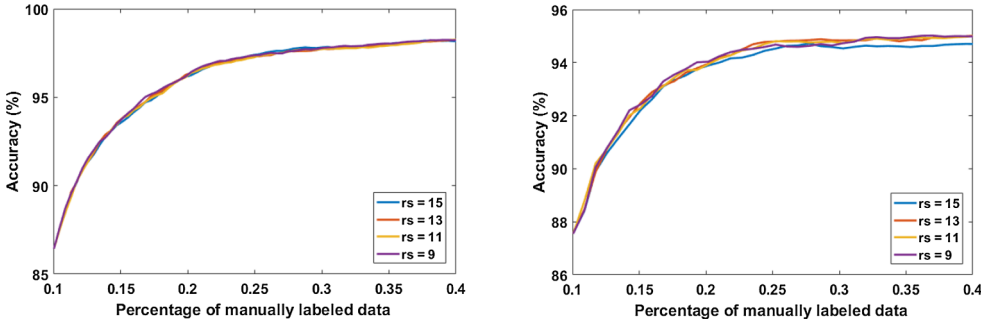

Fig. 8. Comparing the change of accuracies of the proposed method under different numbers of selecting clusters in self-training on (a) dataset_1 (b) dataset_2.

Table 2
The average error rate of automatically labeled samples.

	Dataset1 (%)	Dataset2 (%)
rs = 15	0.572	1.444
rs = 13	0.239	0.481
rs = 11	0.117	0.093
rs = 9	0.047	0.056

Table 3
The average values of areas under curves on two datasets.

	Dataset1 (%)		Dataset2 (%)	
	Precision	Recall	Precision	Recall
Our method	95.16	94.77	87.68	94.68
Random sampling	94.61	90.12	81.06	93.90
Uncertainty sampling	95.08	94.23	87.66	93.64
Representative sampling	94.63	91.37	81.46	94.42

Table 4
The average values of areas under curves on nEU surface defect dataset.

	Precision (%)	Recall (%)
Proposed method	93.33	87.41
Random sampling	93.33	77.84
Uncertainty sampling	93.33	87.83
Representative sampling	93.33	80.94

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was conducted within the Delta-NTU Corporate Lab for Cyber-Physical Systems, Singapore with funding support from Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

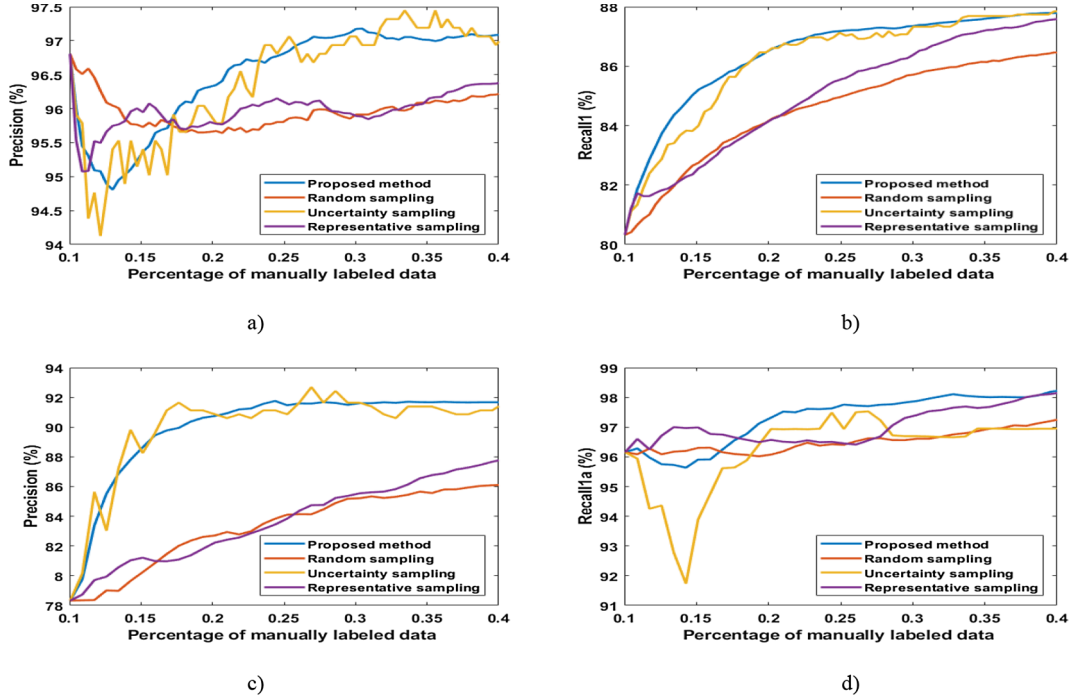

Fig. 9. Comparing the performances of the proposed method with three different benchmarks our datasets. The figures are: (a) Precision on dataset_1, (b) Recall on dataset_1, (c) Precision on dataset_2, (d) Recall on dataset_2.

References

[1] D. Walsh, J. Arena, Seeing AOI for six-sigma PCB manufacturing, Electronic Packaging & Production (2001).

[2] M. Janóczki, Á. Becker, L. Jakab, R. Gróf, T. Takács, Automatic optical inspection of soldering, materials science - advanced topics, in: Dr. Erika Hodulova (Ed.), InTech, 2013 DOI: 10.5772/51699.

[3] N.S.S. Mar, C. Fookes, P.K.D.V. Yarlagadda, Design of automatic vision based inspection system for solder joint segementation, J. AMME 34 (2) (2009).

[4] A.J. Crispin, V. Rankov, Automated inspection of PCB components using a genetic algorithm template-matching approach, Int. J. Adv. Manuf. Technol. 35 (3) (2007) 293–300, https://doi.org/10.1007/s00170-006-0730-0.

[5] X. Hao, W. Li, Z. Sun, S. Zhu, S. Yan, Z. Zhao, Detection of ball grid array solder joints based on adaptive template matching, Int. J. Heat Technol. 36 (2018) 189–194. 10.18280/ijht.360125.

[6] A. Sanguannam, J. Srinonchat, Analysis ball grid array defects by using new image technique, in: Proc. 9th Int. Conf. Signal Process., Beijing, China, Oct. 2008, pp. 785–788.

[7] M.M. Abdelhameed, M.A. Awad, H.M. Abd El-Aziz, A robust methodology for solder joints extraction, in: 2013 8th International Conference on Computer Engineering & Systems (ICCES), Cairo, 2013, pp. 268–273. doi: 10.1109/ICCES.2013.6707217.

[8] W.-Y. Wu, C.-C. Chen, A system for automated BGA inspection, in: Proc. IEEE Conf. Cybern. Intell. Syst., Singapore.

[9] H. Gao, W. Jin, X. Yang, O. Kaynak, A line-based-clustering approach for ball grid array component inspection in surface-mount technology, IEEE Trans. Ind. Electron. 64 (4) (April 2017) 3030–3038, https://doi.org/10.1109/TIE.2016.2643600.

[10] J. Intell. Manuf. 22 (6) (2011) 919–932, https://doi.org/10.1007/s10845-009-0367-6.

[11] Z. Zeng, L. Ma, M. Suwa, Algorithm of locating PCB components based on colour distribution of solder joints, Int. J. Adv. Manuf. Technol. 53 (5) (Mar 2011) 601–614, https://doi.org/10.1007/s00170-010-2850-9.

[12] Soldering & Surf. Mount Tech 29 (3) (2017) 164–170, https://doi.org/10.1108/SSMT-08-2016-0016.

[13] X. Hongwei, Z. Xianmin, K. Yongcong, O. Gaofei, Solder joint inspection method for chip component using improved adaboost and decision tree, IEEE Trans. Compon. Packag. Manuf. Technol. 1 (12) (Dec 2011) 2018–2027.

[14] B. Luo, Y. Zhang, G. Yu, X. Zhou, ANN ensembles based machine vision inspection for solder joints, in: 2007 IEEE International Conference on Control and Automation, pp. 3111–3115, May 2007.

[15] H. Wu, X. Zhang, H. Xie, Y. Kuang, G. Ouyang, Classification of solder joint using feature selection based on bayes and support vector machine, IEEE Trans. Compon. Packag. Manuf. Technol. 3 (3) (Mar 2013) 516–522.

[16] W. Hao, Z. Xianmin, K. Yongcong, O. Gaofei, X. Hongwei, Solder joint inspection based on neural network combined with genetic algorithm, Optik – Int. J. Light Electron Opt. 124 (20) (2013) 4110–4116. Available: http://www.sciencedirect.com/science/article/pii/S0030402613000478.

[17] J.K. Kim, H.S. Cho, Neural network-based inspection of solder joints using a circular illumination, Image Vision Comp. 13 (1995) 479–490.

[18] T. Ong, Z. Samad, M. Ratnam, Solder joint inspection with multi-angle imaging and an artificial neural network, Int. J. Adv. Manuf. Technol. 38 (2008) 455–462.

[19] K.W. Ko, H.S. Cho, Solder joints inspection using a neural network and fuzzy rule-based classification method, IEEE Trans. Electron. Packag. Manuf. 23 (2) (Apr 2000) 93–103.

[20] G. Acciani, G. Brunetti, G. Fornarelli, A multiple neural network system to classify solder joints on integrated circuits, Int. J. Comput. Intelligence Res. 2 (2006) 337–348.

[21] N. Cai, Y. Zhou, Q. Ye, G. Liu, H. Wang, X. Chen, IC solder joint inspection via robust principle component analysis, IEEE Trans. Compon. Packag. Manuf. Technol. 7 (2) (Feb. 2017) 300–309, https://doi.org/10.1109/TCPMT.2016.2638503.

[22] N. Cai, J. Lin, Q. Ye, H. Wang, S. Weng, B.W.K. Ling, A new IC solder joint inspection method for an automatic optical inspection system based on an improved visual background extraction algorithm, in: IEEE Transactions on Components, Packaging and Manufacturing Technology, Manuf. Technol., vol. 6, no. 1, pp. 161–172, Jan. 2016.

[23] N. Cai, G. Cen, J. Wu, F. Li, H. Wang, X. Chen, SMT solder joint inspection via a novel cascaded convolutional neural network, IEEE Trans. Compon. Packag. Manuf. Technol. 8 (2018) 670–677.

[24] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Region-based convolutional networks for automatic object detection and segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 142–158 (2016).

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, R.E. Scott, F. Cheng-Yang, B.C. Alexander, SSD: single shot multibox detector, CoRR, 2015.

[26] J. Redmon, S. Divvala, R. Grishick, A. Farhad, You only look once: unified, real-time object detection. arXiv: 1506.02640v5, 2015.

[27] S. Kim, W. Kim, Y. Noh, F.C. Park, Transfer learning for automated optical inspection, Neural Networks (IJCNN) (2017), https://doi.org/10.1109/IJCNN.2017.7966162.

[28] V. Natarajan, T. Hung, S. Vaikundam, L. Chia, Convolutional networks for voting-based anomaly classification in metal surface inspection, IEEE International Conference on Industrial Technology (2017), https://doi.org/10.1109/IJCNN.2016.7727522.

[29] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[30] I.T. Jolliffe, Principal component analysis, series: Springer series in statistics, second ed., Springer, NY, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4, 2002.

[31] D. Ienco, A. Bifet, I. Žliobaitė, B. Pfahringer, Clustering based active learning for evolving data streams, in: J. Fürnkranz, E. Hüllermeier, T. Higuchi (Eds.), Discovery Science. DS 2013. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2013.

[32] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, in: Proceedings of the International Conference on Machine Learning (ICML). Morgan Kaufmann, 2000, pp. 999–1006.

[33] G. Fung, O.L. Mangasarian, Semi-supervised support vector machines for unlabeled data classification, Optimization Methods and Software, 15(1), 2001.

[34] Lining Zhang, Lipo Wang, Weisi Lin, Semi-supervised biased maximum margin analysis for interactive image retrieval, IEEE Trans. Image Process. 21 (4) (2011) 2294–2308.

[35] Knowl.-Based Syst. 143 (2018) 65–80, https://doi.org/10.1016/j.knosys.2017.12.006.

[36] Z. Xu, K. Yu, V. Tresp, X. Xu, J. Wang, Representative sampling for text classification using support vector machines, in: F. Sebastiani (Ed.), Advances in Information Retrieval. ECIR 2003. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2003.

[37] K. Song, Y. Yan, A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects, Appl. Surf. Sci. 285 (Nov. 2013) 858–864.