IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

D^{3}PointNet: Dual-level Defect Detection PointNet for Solder Paste Printer in Surface Mount Technology

JIN-MAN PARK[1], YONG-HO YOO[2], UE-HWAN KIM[1], DUKYOUNG LEE[2], AND JONG-HWAN KIM[1] (FELLOW, IEEE)

[1]School of Electrical Engineering, KAIST, Republic of Korea
[2]Kohyoung Technology, Inc., Seoul 08588, Republic of Korea

Corresponding author: Jong-Hwan Kim (e-mail: johkim@rit.kaist.ac.kr).

This work was supported in part by the Industrial Strategic Technology Development Program (10077589, Machine Learning Based SMT Process Optimization System Development) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea), and in part by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00563, Research on Adaptive Machine Learning Technology Development for Intelligent Autonomous Digital Companion).

ABSTRACT In the field of surface mount technology (SMT), early detection of defects in production machines is crucial to prevent yield reduction. In order to detect defects in the production machine without attaching additional costly sensors, attempts have been made to classify defects in solder paste printers using defective solder paste pattern (DSPP) images automatically obtained through solder paste inspection (SPI). However, since the DSPP images are sparse, have various sizes, and are hardly collected, existing CNN-based classifiers tend to fail to generalize and over-fitted to the train set. Besides, existing studies employing only multi-label classifiers are less helpful since when two or more defects are observed in the DSPP image, the location of each defect can not be specified. To solve these problems, we propose a dual-level defect detection PointNet (D^{3}PointNet), which extracts point cloud features from DSPP images and then performs the defect detection in two semantic levels: a micro-level and a macro-level. In the micro-level, a type of printer defect per point is identified through segmentation. In the macro-level, all types of printer defects appearing in a DSPP image are identified by multi-label classification. Experimental results show that the proposed D^{3}PointNet is robust to the sparsity and size changes of the DSPP image, and its exact match score was 10.2% higher than that of the existing CNN-based state-of-the-art multi-label classification model in the DSPP image dataset.

INDEX TERMS Defect detection, multi-label classification, PointNet, segmentation, solder paste printer defects.

I. INTRODUCTION

WITH the rapid development of the electronics industry, multiple studies have been actively carried out to improve production yield by detecting defects in the surface mount technology (SMT) process [1] in earlier stages. For this, inspection machines are placed at the end of each step of SMT whose process consists of solder paste printing, pick-and-place, and reflow. Solder paste inspection (SPI) machines monitor the outputs of the solder paste printer, and automated optical inspection (AOI) machines check the products of pick-and-place or reflow procedure. On the other hand, early detection of defects in the machines can help the process manager respond quickly and minimize the yield reduction. Nonetheless, there has been little progress in detecting defects in the SMT machines and we focus on this in this work.

Early studies [2], [3] tried to detect defects in machines by attaching additional sensors directly to the production machines. These methods have the disadvantages that the required sensors are expensive to install and cannot be applied to existing pre-installed production machines. In order to eliminate the sensor installation costs, a recent study [4] attempted to diagnose a solder paste printer defect using a defective solder paste pattern (DSPP) image[1] automatically

[1]In [4], it is referred to as an SPI Image. In this paper, we call it a DSPP image to distinguish it from a solder volume map.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

obtainable from a pre-installed SPI machine[2].

As shown in Fig. 1, a DSPP image is obtained by post-processing a solder volume map measured in the process of solder paste inspection. Each pixel in a solder volume map consists of a measurement of the solder volume printed on a printed circuit board (PCB), where each measurement is represented as relative proportions based on predetermined criteria, as illustrated in Fig. 1 (a). If a volume value in a solder volume map falls within a predefined normal range[3], it is classified as 'normal,' otherwise classified as 'excessive' or 'insufficient,' resulting in a DSPP image composed of three binary channels, as depicted in Fig. 1 (b).

DSPP images are generally used by human experts to detect defects in solder paste printers. Since this manual analysis is time-consuming and expensive, attempts are underway to automate this process using neural networks. However, training a neural network with DSPP images is more challenging than training one with typical RGB images because of the following characteristics of DSPP images:

- **Sparseness**: The ratio of non-zero elements in DSPP images is generally less than 1.0%, which makes it difficult for a neural network to aggregate meaningful features. In addition, the sparsity varies from data to data even within the same class. Therefore, the neural network must be able to deal with various levels of the sparsity of input data.
- **Various sizes**: The size of DSPP images is very diverse, as the size of the PCB varies (e.g., from a PCB of an earphone to that of a computer). Resizing a DSPP image can cause some loss of data, which leads to a decrease in defect detection performance. Therefore, the neural network must be able to handle DSPP images without resizing them.
- **Limited data**: It is not easy to collect enough data for training since the cost of acquisition and annotation of DSPP images is highly expensive. In other words, the prepared training set does not have as much variety as the actual SMT field data for characteristics such as size and sparsity. Therefore, the neural network must have the generalization ability to deal with unseen data with only a small amount of training data distribution.

Due to those characteristics of DSPP images, conventional CNNs [5], [6] fail to generalize and often over-fitted to training set dealing with DSPP images. In [4], a multi-label classification network to deal with DSPP images of various sizes, named MarsNet, was proposed. MarsNet can handle small DSPP images by increasing the resolution of feature maps employing an improved dilated residual network [7] as its backbone. However, MarsNet, like other CNN-based models, significantly reduces its classification performance

[2]SPI machines generate DSPP images automatically using the PCB information stored in SPI database.

[3]A PCB contains multiple chips, and each chip type has its own normal range. Since this information is provided by PCB producers and is usually pre-built into the SPI Machine, no human intervention is required for an SPI machine to create DSPP images.

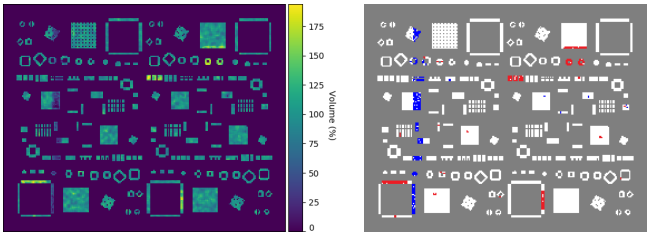


FIGURE 1: (a) Solder volume map. (b) Corresponding DSPP image processed from (a). In (b), pixels with excessive, normal, and insufficient solder paste are marked with red, white, and blue, respectively. Pixels with no solder paste are marked as grey.

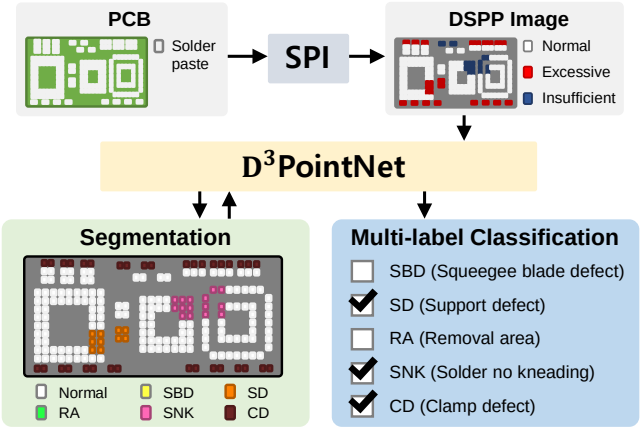


FIGURE 2: Dual-level defect detection PointNet (D^{3}PointNet) for solder paste printers in SMT. Given a DSPP image, D^{3}PointNet performs segmentation (micro-level defect detection) and multi-label classification (macro-level defect detection) for the five defects of the solder paste printer.

when the size of the input DSPP image is large enough to exceed the distribution of its training dataset.

To overcome the above-mentioned challenges and boost the detection performance, we propose the dual-level defect detection PointNet (D^{3}PointNet) as illustrated in Fig. 2. The proposed D^{3}PointNet consists of three components designed in this work: 1) conversion of DSPP images into point clouds to deal with sparseness, 2) two hand-crafted features for generalization ability, and 3) a set of a single encoder and two decoders for dual-level defect detection.

First of all, we convert defective regions of DSPP images into point clouds. By converting DSPP images into point clouds as proposed, we can employ effective networks robust to sparseness and size change. Next, we design two hand-crafted features that earns more generalization ability: the edge feature (EF) and the prior feature (PF). The EF is designed to prevent the loss of position information caused when converting a DSPP image into a point cloud and the PF is derived from prior knowledge of DSPP images.

The set of a single encoder and two decoders recognizes defects of a solder paste printer in two semantic levels: the micro-level and the macro-level. In the micro-level, it seg-

ments the defects in DSPP images that are caused by defects of solder paste printers and identifies the type of defects for each point. In the macro-level, it detects the occurrence of the defects and classifies the multi-label of the solder paste printer defects. Moreover, the set of a single encoder and two decoders is designed by following the multi-task learning scheme [8] [9]. The encoder is shared across two levels and two different decoders are used for each level, and additional skip concatenation paths between the two decoders are designed.

The key contributions of our work are as follows:

- We design a dual-level defect detection model, D^{3}PointNet, inspecting solder paste printer defects through segmentation and multi-label classification.
- We introduce the problems of sparseness and various sizes of DSPP images and solve them by converting DSPP images to point cloud form and applying a point cloud processing network.
- We define two hand-crafted features, EF and PF, using prior knowledge regarding PCB patterns to boost the generalization performance of the solder paste printer defect detection.

The remainder of this paper is organized as follows. Section II describes related works including SMT products inspection, SMT machine inspection, and deep learning on point clouds. Section III briefly reviews the mechanism of PointNet. Section IV presents the proposed D^{3}PointNet, including problem statement, image to point cloud conversion, hand-crafted feature generation, and the multi-task learning architecture. Section V describes the datasets used in this paper and reports experimental results including ablation studies. Finally, Section VI summarizes the proposed method.

II. RELATED WORKS

Our goal is to detect defects in SMT machines given DSPP images representing defects of SMT products, using neural networks processing point clouds. Therefore, our work relates to three areas: 1) defect detection of SMT products, 2) defect detection of SMT machines, and 3) deep learning on point clouds. These three areas are briefly described in the following.

A. DEFECT DETECTION IN SMT PRODUCTS

The detection of defects in SMT products are divided into three stages depending on the inspection point of the products: before solder paste printing, between solder paste printing and pick-and-place, and after pick-and-place and reflow. First, the defect detection before solder paste printing inspects the condition of bare PCBs using vision sensor in two steps: defect region proposal [10] and defect classification on proposals [11]–[14]. Additionally, microwave sensors are also used to capture defects such as strains that are difficult to detect with cameras and vision algorithms [15].

Next, the defect detection between solder paste printing and pick-and-place checks the quality of the solder paste printed on bare PCBs. The quality of solder paste is assessed by measuring the volume, the shape, the coplanarity of solder pastes on each pad [16], [17], [18]. The defect detection at the last stage inspects the quality of solder joints in two steps, where a solder joint is a solidified solder connecting a pad and a component in a PCB. First, defect region proposals are extracted. Then, a detection model classifies the type of defects. For the detection model, Bayesian network [19], multi-layer perceptrons (MLPs) with hand-crafted feature extraction methods [20] or CNN-based end-to-end models [21], [22] are used.

B. DEFECT DETECTION OF SMT MACHINES

The defect detection methods of SMT products have a limitation in that they only detect the occurrence of defects and cannot infer the cause of the defects. Studies that try to overcome the limitations are underway. Their primary goal is to detect defects of SMT machines. One preliminary study attempted to detect defects of solder paste printer stencils [2]. They collected an image dataset of solder paste printer stencils with optical equipment, trained a CNN using stencil images, and classified the defects. However, this method has a disadvantage that the stencil must be photographed using separate photographic equipment. In another study, a method to detect the defect of SMT machines using the sound from the machines [3] was presented. The method models the steady-state of SMT machines using an auto-encoder. The reconstruction error of the auto-encoder can classify the defect of the SMT machines. A recent study [4] attempted to classify defects in solder paste printers using DSPP images that can be obtained by solder paste inspection without additional sensor attachment to SMT machines. A variant of CNN-based multi-label classification network, MarsNet, was proposed in [4] to handle various sizes of DSPP images. MarsNet employs dilated residual network [7], hierarchical vertical pooling (HVP), and two MLPs for a classifier and a threshold estimator, respectively.

C. DEEP LEARNING WITH POINT CLOUDS

Point clouds compactly represent the sparse data acquired by 3D sensors. The resulting representations are highly irregular in that the distributions of points are uneven. Typical convolutional architectures, however, require highly regular input data formats such as 3D voxels and image grids. The point clouds need to be rendered as a sequence of images [23] or 3D voxels [24] in order to be fed into the convolutional architectures, but the rendered representations become voluminous and contain quantization artifacts. To resolve this problem, researchers have attempted to extract features directly from point clouds. PointNet [25] achieves the invariance of input order by the use of symmetric functions over point clouds. PointNet2 [26] and SO-Net [27] apply PointNet hierarchically and capture local structures with improved accuracy.

III. PRELIMINARIES

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

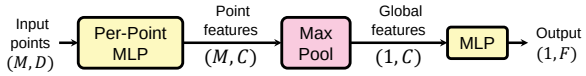


FIGURE 3: The architecture of basic PointNet. PointNet extracts a fixed-size feature vector from a set of points.

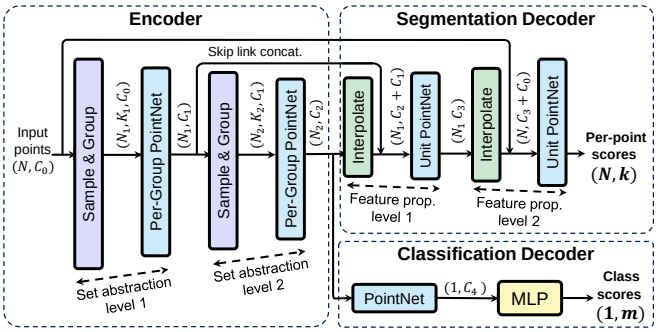


FIGURE 4: The architecture of PointNet2. Pointnet2 is a hierarchically extended version of PointNet, which enables high performance segmentation and classification of point clouds.

A. POINTNET

PointNet [25] extracts a fixed-size feature vector from a set of unordered points, as shown in Fig. 3. After receiving M points as input, PointNet converts each D-dim point into a vector of C-dim through a per-point MLP. Then, the M point vectors are transformed into a global feature vector of C-dim by the gloabal max-pooling. Finally, another MLP is applied to the global feature vector to map it to the output vector of F-dim. Since the per-point MLP and global max-pooling are symmetric, the output of PointNet is invariant to the order of input points. Due to its global max-pooling, however, PointNet alone cannot capture the local structure of the input.

B. POINTNET2

PointNet2 is extended from PointNet to a model in the form of an encoder-decoder, and extracts features hierarchically utilizing a set of basic PointNets. Due to its hierarchical feature learning architecture, PointNet2 overcomes the limitation of the basic PointNet. In this paper, we exploit both basic PointNet and PointNet2 for the defect detection of solder paste printer.

The encoder of PointNet2 consists of L set abstraction levels where each abstraction level consists of a sample&group layer and a per-group PointNet layer, as shown in Fig. 4. At the l-th set abstraction level ($l = 1, ..., L$), given an input point set of size $N_{l-1} \times C_{l-1}$, N_l points are chosen as centroids of local regions where $N_l \leq N_{l-1}$. Then for each centroid point, the k-nearest neighbors (kNN) are grouped as a local region by the sample&group layer. Through the sample&group layer, N_l groups are formed, where a group is a subset of the input point cloud of size $K_l \times C_{l-1}$, and $k = K_l$ is the number of points in a group which corresponds to neighbors of a centroid point. Then, the per-group PointNet layer, where weights are shared across different local regions, is applied to the groups of point sets to extract an output point set of size $N_l \times C_l$ where C_l is the dimension of each point. Finally, the output is fed into the next abstraction level.

Next, PointNet2 has two types of decoders: a segmentation decoder and a classification decoder. The segmentation decoder is composed of stack of feature propagation levels, where a feature propagation level consists of an interpolation layer and an unit PointNet layer. Once the point features are fed to an interpolation layer, they are propagated from $N_l \times C_l$ to $N_{l-1} \times C_{l-1}$. Here, the features of the newly created points are computed through the inverse distance weighted average of their surrounding points selected via kNN. Then, the interpolated features of size $N_{l-1} \times C_l$ are concatenated with the point features from the set abstraction level $l-1$ through the skip link. Then, the features of size $N_{l-1} \times (C_l + C_{l-1})$ are fed into a unit PointNet layer, where PointNet is applied for each point, updating feature vector of each point. Repeating this feature propagation process, the number of points in the resulting point set becomes that of the original input point set. Finally, per-point softmax is applied to extract an output point cloud containing per point probabilities for each class. In the classification decoder, point features are abstracted to a global feature vector by applying a basic PointNet to the whole point sets. Then, an MLP with a softmax layer is applied to extract an output vector containing probabilities for each and every class.

IV. PROPOSED D^{3}POINTNET

A. OVERVIEW

The proposed D^{3}PointNet takes a DSPP image, $I \in \mathbb{R}^{3\times w\times h}$, as input, and outputs per-point scores, $S^{seg} \in \mathbb{R}^{N\times k}$, and class scores, $S^{cls} \in \mathbb{R}^{1\times m}$, as illustrated in Fig. 5. Here, w and h are respectively width and height of the DSPP image, m is the number of defect classes, k=m+1 is the number of defect classes and a normal class, and N is the number of defective non-zero pixels in the DSPP image, which is equivalent to the number of points in a point cloud representation of the DSPP image.

The DSPP image is converted to a set of points where each point is a 3-dim vector containing 2D position (2-dim) and excessive/insufficient information (1-dim) of defective pixel. Each converted point is regarded as a position feature. Then, the two hand-crafted features are extracted by the PF and EF extraction. A PF and an EF for a point are a 2-dim vector and a 4-dim vector, respectively. The three features are concatenated, resulting the dimension of 9 for each point. Next, PointNet2 encoder is applied to the set of points to extract hierarchical features. The hierarchical features are fed to segmentation decoder and multi-label classification decoder to generate S^{seg} and S^{cls}, respectively. Additional input and score concatenation paths are designed to boost multi-task learning efficiency, which further increases the performance of both tasks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

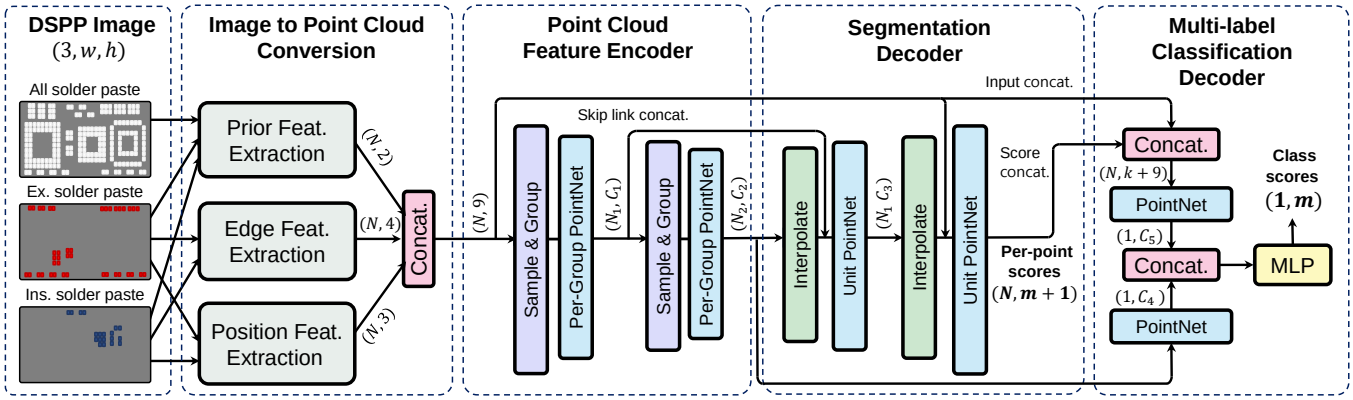


FIGURE 5: An overview of D^{3}PointNet. It employs 1) position feature extraction converting DSPP images into point cloud, 2) effective hand-crafted features designed only for DSPP images: edge feature (EF) and prior feature (PF), and 3) multi-task learning structures: input and score concatenations.

B. IMAGE TO POINT CLOUD CONVERSION

We first convert a DSPP image to a point cloud, where a point is equal to a position feature. Then, we extract PF and EF along with the basic position feature. Procedure of converting a DSPP image into a point cloud is described in the following.

1) Position Feature Extraction

A DSPP image, I consists of three binary channels: $I_{ma} \in \mathbb{R}^{1\times w\times h}$, $I_{ex} \in \mathbb{R}^{1\times w\times h}$, and $I_{in} \in \mathbb{R}^{1\times w\times h}$. I_{ma} is a mask containing all of the solder paste patterns which includes every normal and defective solder paste, I_{ex} is an excessive solder paste pattern, and I_{in} is an insufficient solder paste pattern. We extract the 2D coordinates of the defective solder pastes respectively from I_{ex} and I_{in}. The defective solder pastes are non-zero pixels in each channel.

Let (x_i, y_i) denote a 2D coordinate of a defective solder paste, d_i, where $0 \le x_i < w$, $0 \le y_i < h$ and $i = 1, ..., N$. We first translate the 2D coordinate so that the center of the coordinate system matches the center of the DSPP image, and normalize the coordinate so that the area of the DSPP image equals to one, as follows:

$$\hat{x_i} = \frac{x_i - w/2}{\alpha_x}, \quad \hat{y_i} = \frac{y_i - h/2}{\alpha_y}, \tag{1}$$

where α_x and α_y are scaling parameters for normalization. There are two options for α_x and α_y. For the first option, we keep the original aspect ratio of the DSPP image in the resulting point cloud by setting

$$\alpha_x = \alpha_y = \sqrt{wh}. \tag{2}$$

For the second option we change the aspect ratio of the original DSPP image to 1:1 in the resulting point cloud by setting

$$\alpha_x = w, \qquad \alpha_y = h. \tag{3}$$

The resulting 3D position feature, f_i^{pos}, corresponding to d_i is the concatenation of the normalized 2D coordinate, $(\hat{x_i}, \hat{y_i})$, of the defective solder paste, d_i, and the type of channel it belongs to, and it can be expressed as follows:

$$f_i^{pos} = \begin{cases} [\hat{x_i}; \hat{y_i}; +c] & d_i \in I_{ex} \\ [\hat{x_i}; \hat{y_i}; -c] & d_i \in I_{in}, \end{cases} \tag{4}$$

where $i = 1, ..., N$ and c, a constant, is a z-axis coordinate.

The position features represent only non-zero elements of its corresponding DSPP image, which makes neural networks robust to sparsity. Thus, the proposed D^{3}PointNet could extract relevant features even when the sparseness of DSPP images increases. On the other hand, image-processing neural networks, such as CNNs, could not extract meaningful feature from DSPP images as the ratio of non-zero elements decreases (increasing sparseness). In addition, resizing of images causes loss of information, while the position feature does not suffer from it and does not lose any information.

2) Edge Feature Extraction

Since the a position feature, f_i^{pos}, cannot express the edges of the DSPP image, the distance information from the i-th defective solder paste, d_i, from the four edges of the top, bottom, left, and right of the DSPP image disappears during the position feature extraction process, as illustrated in Fig. 6 (a). However, The distance information of each point is crucial in the domain of DSPP images as it may affect the defect class of the DSPP image, as shown in Fig. 6 (b). Therefore, we design EF to keep the distance information.

The EF, $f_i^{edg} \in \mathbb{R}^4$, extracted from d_i, is defined as follows:

$$f_i^{edg} = \left[\frac{x_i}{w}; \frac{y_i}{h}; \frac{w - x_i}{w}; \frac{h - y_i}{h}\right], \tag{5}$$

where $i = 1, ..., N$, and each element of f_i^{edg} is the normalized distance between d_i and the four edges of the DSPP image.

3) Prior Feature Extraction

It is a well-known prior knowledge that given a region containing defective solder pastes, r, in a DSPP image, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

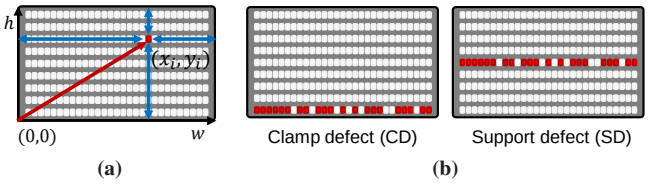


FIGURE 6: (a) Missing information (marked as blue arrows) when representing a defective solder paste as a point vector (marked as a red arrow). (b) An example when the same defective solder paste pattern with different locations resulting in different defect types. Here, each grid represents a DSPP image, where defective solder pastes are marked as red.

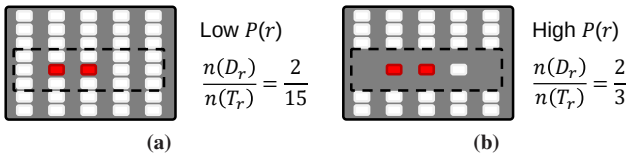


FIGURE 7: A piece of prior knowledge from DSPP images. A region, r, is represented as a dashed line, and each defective and normal solder paste are marked as red and white, respectively. (a) If defective solder pastes are sparse compared to normal ones in a region, the cause of the defects in the solder paste is likely to be noise, not a malfunction of the solder paste printer. (b) Otherwise, the cause of the defects in the solder paste is likely the failure of the solder paste printer.

probability that the defect is due to a defect in the solder paste printer, $\mathrm{P}(r)$, is proportional to the number of defective solder pastes divided by the number of total solder pastes as follows:

$$\mathrm{P}(r) \propto n(D_r)/n(T_r), \tag{6}$$

where D_r is a set of defective solder pastes in r, and T_r is a set of total solder pastes in r. Fig. 7 illustrates the prior knowledge with two examples. We use this prior knowledge to formulate the PF. Given a defective solder paste, d_i, and its position, (x_i, y_i), we set R_i as a union of R_i^{ver} and R_i^{hor}, where $R_i^{ver} = \{(x_1, y_1), ..., (x_{N_v}, y_{N_v})\}$ and $R_i^{hor} = \{(x'_1, y'_1), ..., (x'_{N_h}, y'_{N_h})\}$ are a set of vertically located solder paste positions and a set of horizontally located solder paste positions, respectively. Here, $(x_1, y_1), ..., (x_{N_v}, y_{N_v})$ are solder paste positions located on the vertical line of d_i, $(x_1 = ... = x_{N_v})$ where N_v is the number of vertically located solder pastes, and $(x'_1, y'_1), ..., (x'_{N_h}, y'_{N_h})$ are solder paste positions located on the horizontal line of d_i, $(y_1 = ... = y_{N_h})$ where N_h is the number of the horizontally located solder pastes.

Then, we define the PF, $f_i^{pri} \in \mathbb{R}^2$, as a concatenation of the two defect ratios:

$$f_i^{pri} = \left[\frac{n(D_i^{ver})}{n(R_i^{ver})}; \frac{n(D_i^{hor})}{n(R_i^{hor})}\right], \tag{7}$$

where $D_i^{ver} \subset R_i^{ver}$ and $D_i^{hor} \subset R_i^{hor}$ are the defective solder paste positions in R_i^{ver} and R_i^{hor}, respectively. D_i^{ver} and D_i^{hor} are obtained from the defective channels, I_{ex} and I_{in}, and R_i^{ver} and R_i^{hor} are obtained from the mask channel I_{ma}.

4) Union of All Point Cloud Features

We concatenate the three proposed features to create the final input point cloud, $P^{in} = \{p_1, ..., p_N\}$, where $p_i = [f_i^{pos}; f_i^{edg}; f_i^{pri}] \in \mathbb{R}^9$.

C. POINTNET2 ENCODER

The input point cloud, P^{in}, is fed to the PointNet2 encoder which has two abstraction levels to extract hierarchical point cloud features, $P^{h_1} \in \mathbb{R}^{N_1 \times C_1}$ and $P^{h_2} \in \mathbb{R}^{N_2 \times C_2}$ as follows:

$$\begin{aligned} G^1 &= \text{Sample\&Group}_1(P^{in}) \\ P^{h_1} &= \text{PerGroupPointNet}_1(G^1) \\ G^2 &= \text{Sample\&Group}_2(P^{h_1}) \\ P^{h_2} &= \text{PerGroupPointNet}_2(G^2), \end{aligned} \tag{8}$$

where a set abstraction level consists of a sample and group layer and a per-group PointNet layer. $G^1 \in \mathbb{R}^{N_1 \times K_1 \times 9}$ and $G^2 \in \mathbb{R}^{N_2 \times K_2 \times C_1}$ are grouped point clouds, where N_j is the number of groups in G^j and K_j is the number of points in G^j $(j = 1, 2)$. A group, G^j, is aggregated to a point by the per-group PointNet layer, resulting in a set of N_j points.

D. DUAL-LEVEL DECODERS

The proposed model detects printer defects in two semantic levels: the micro-level and the macro-level. In the micro-level, for every point, the segmentation decoder identifies the type of the solder paste printer defect that has made the point excessive/insufficient. In the macro-level, the multi-label classification decoder identifies the types of solder paste printer defects that appear throughout the set of points. The two decoders, described below, share their encoder, which is the hard parameter sharing architecture for multi-task learning [8]. Furthermore, we add two additional paths: input concatenation and score concatenation in order to increase the multi-task performance.

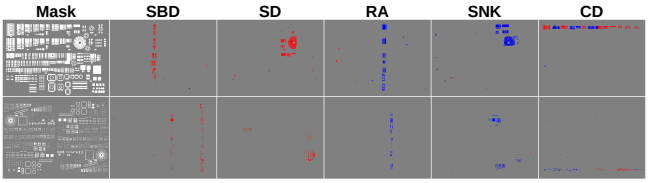
1) Segmentation Decoder

As described in Section III, the segmentation decoder propagates point features through its feature propagation layers. A feature propagation layer consists of a sequence of a interpolate layer and a unit PointNet layer as follows:

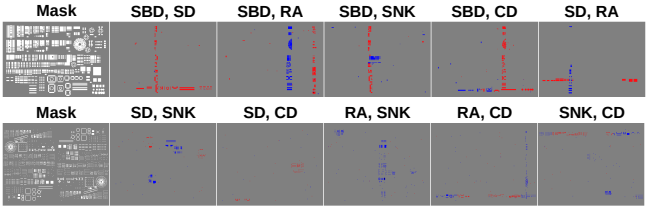
$$\begin{aligned} P^{e_1} &= \text{Interpolate}_1(P^{h_2}) \\ P^{u_1} &= \text{UnitPointNet}_1([P^{e_1}; P^{h_1}]) \\ P^{e_2} &= \text{Interpolate}_2(P^{u_1}) \\ S^{seg} &= \text{UnitPointNet}_2([P^{e_2}; P^{in}]), \end{aligned} \tag{9}$$

where $P^{e_1} \in \mathbb{R}^{N_1 \times C_2}$ and $P^{e_2} \in \mathbb{R}^{N \times C_3}$ are the interpolated set of points from P^{h_2} and P^{u_1}, respectively, and $P^{u_1} \in \mathbb{R}^{N_1 \times C_3}$ and $S^{seg} \in \mathbb{R}^{N \times k}$ are the outputs of the unit PointNet layers. We set the number of class k as $m+1$ where m is the number of printer defect types and the $+1$ is for the normal class.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access



(a) DSPP images when there is one solder paste printer defect.



(b) DSPP images when there are two solder paste printer defects.

FIGURE 8: Defective solder paste pattern (DSPP) image dataset. Excessive solder pastes are marked as red while insufficient solder pastes are marked as blue.

2) Multi-label Classification Decoder

We extract a global fixed-size feature, $P^{c_2} \in \mathbb{R}^{1\times C_4}$, from the hierarchical point cloud feature, P^{h_2}, using a basic PointNet layer as follows:

$$P^{c_2} = \text{PointNet}_2(P^{h_2}). \tag{10}$$

Then, we extract another global fixed-size feature, $P^{c_1} \in \mathbb{R}^{1\times C_5}$, from the concatenation of the input set of points, P^{in}, and the segmentation output, S^{seg}, using another PointNet layer as follows:

$$P^{c_1} = \text{PointNet}_1([P^{in}; S^{seg}]). \tag{11}$$

Class scores, $S^{cls} \in \mathbb{R}^{1\times m}$, can be obtained by applying an MLP to the concatenation of P^{c_1} and P^{c_2} as follows:

$$S^{cls} = \text{MLP}([P^{c_1}; P^{c_2}]). \tag{12}$$

Finally, a sigmoid layer, ϕ, is applied to S^{cls} to represent the result as an m-dim vector of probabilities.

3) Multi-task Loss

The total loss is the weighted sum of the segmentation loss, $\mathcal{L}^{seg}$, and the multi-label classification loss, $\mathcal{L}^{cls}$, where the two losses are equally weighted. Here, $\mathcal{L}^{seg}$ is obtained by a cross entropy loss between S^{seg} and a segmentation ground-truth, $T^{seg} \in \mathbb{R}^{N\times k}$, while $\mathcal{L}^{cls}$ is obtained by a binary cross entropy loss between S^{cls} and a multi-label classification ground-truth, $T^{cls} \in \mathbb{R}^{1\times m}$.

V. EXPERIMENTS

We evaluated the proposed D^{3}PointNet on two customized datasets: a various-size DSPP image set and a fixed-size DSPP image set. In the various-size DSPP image set, there are nine types of DSPP image sizes from nine different PCBs where the image size varies from 75×121 to 341×397. For each PCB type, 8,400 DSPP images were collected where a DSPP image contains a maximum of two solder paste printer defects. Following the same experimental setting in [4], we used four of them as a train set and the rest as a test set, resulting $8,400 \times 4 = 33,600$ images for training and $8,400 \times 5 = 42,000$ images for testing. The experimental setting for the fixed-size DSPP image set was identical to that of the various-size DSPP image set, except that the size of all DSPP images in the fixed-size DSPP image set is 299×299. Comparing results from these two datasets, we showed that the proposed model is able to deal with various image sizes better than the CNN-based models including the previous state-of-the-art, MarsNet [4]. For the various-size DSPP image set, we used the dataset used in [4]. For the fixed-size DSPP image set, we obtained all the DSPP images from an on-site SPI machine manufactured by Koh Young Technology, KY8030-2. As shown in Fig. 8(a), there are five types of defects in the solder paste printer that can be detected from DSPP images: squeegee blade defect (**SBD**), support defect (**SD**), removed area of the solder paste (**RA**), solder no kneading (**SNK**), and clamp defect (**CD**).

SBD is a phenomenon in which a solder paste is excessively deposited on a cracked portion of a squeegee blade when the squeegee blade rolls over a stencil. **SD** occurs when a part of the supports, which hold the PCBs and keep them horizontal during the solder paste printing, are broken. The solder paste is excessively deposited at the region where the supports are broken. **RA** is a defect caused by the difference in density in the solder paste. In the process of solder paste printing, if the density of a specific part of the solder paste is high, the solder paste of that part is consumed faster than the other parts. At this time, the part with the high solder paste density is insufficiently deposited. **SNK** is a defect that occurs since the solder paste is not sufficiently kneaded. If the solder paste is not sufficiently kneaded, the solder paste will become hard, and the solder paste will not spread properly when the squeegee blade rolls. Therefore, the solder paste is deposited insufficiently on the hardened part of the solder paste. **CD** is a phenomenon in which the PCB is so tightly clamped that strain occurs on the PCB, causing solder paste to be deposited excessively or insufficiently near the clamped area, which usually happens at the top or bottom edge of the PCB. As shown in Fig. 8(b), there are 10 cases in which two of the five defects can appear together. In the datasets, each DSPP image is labeled in both pixel-level and image-level for segmentation and multi-label classification, respectively.

A. EVALUATION METRICS

To evaluate the segmentation performance, we measured the per-point accuracy and the mean intersection over union (mIoU), following the conventional settings in the point cloud segmentation field [12], [25], [26]. Per-point accuracy is an average of per-point classification accuracy over whole points. Also, mIoU is an average of IoUs, where an IoU is a per-class metric, which is equal to the number of true positives divided by the number of sum of true positives, false positives, and false negatives.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

TABLE 1: Ablation Study on Multi-task Learning Architectures.

Model				#	Segmentation Metric				Multi-label Classification Metric					
Input	Score	Seg.	Cls.	of	Per Point Acc.		mIoU		Exact Match		F1		mAP	
concat.	concat.	dec.	dec.	Parameters	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various
			✓	1.8 M	N/A	N/A	N/A	N/A	91.86	90.01	96.16	95.35	99.07	98.77
		✓		3.0 M	87.26	87.47	**73.41**	73.68	N/A	N/A	N/A	N/A	N/A	N/A
		✓	✓	3.8 M	**87.29**	86.77	73.37	72.83	91.60	90.38	95.99	95.46	99.13	98.79
	✓	✓	✓	4.2 M	87.22	87.82	73.33	74.26	91.76	90.48	96.12	95.55	99.12	98.85
✓		✓	✓	4.2 M	86.88	87.36	72.60	73.67	92.07	91.71	96.30	96.00	99.15	99.00
✓	✓	✓	✓	4.2 M	87.27	**88.61**	73.19	**75.90**	**92.25**	**92.36**	**96.38**	**96.27**	**99.16**	**99.07**

To evaluate the multi-label classification performance, we mainly measured exact match (EM), and in some experiments we additionally measured F1, accuracy, mean average precision (mAP), precision and recall, following the conventional settings in the multi-label classification field [28]. EM and F1 are threshold dependent measures, whereas mAP is not. To get the EM and the F1, probabilities of model output were classified by a predefined confidence threshold of 0.7. Then, the EM was obtained by measuring the percentage of predictions that exactly match their corresponding ground-truths as follows:

$$EM = \frac{1}{n}\sum_{i=1}^{n} I(S_i = T_i), \tag{13}$$

where n is the number of DSPP images, S_i and T_i denotes model outputs for i-th sample and its corresponding labels, respectively, and $I(\cdot)$ equals 1 when its input equation is true, otherwise 0.

Note that given a sample, EM allows a score only if every label matches, whereas metrics such as mAP, F1, and accuracy allow partial scores even if only a subset of labels matches. Thus, EM is the most suitable metric for evaluating the multi-label classification of defects because when multiple defects occur, it is highly required to detect all of them. Therefore, we selected EM as the primary metric in all multi-label classification experiments.

To get F1, for each prediction, F1_i was computed, which is the harmonic mean of the precision and the recall, where $i = 1, ..., O$, and O is the number of predictions. Then, they were averaged to get the F1. Here, F1 measures the average overlap between the ground truths and the predictions, which is more generous than EM. The mAP is the mean of the average precisions (APs), where an AP is the area under the precision-recall curve of a class. The APs were approximated by taking an average of the 11-point interpolated AP following the conventional approximation methods [29].

In addition, we measured accuracy for the comparison with MarsNet. The accuracy is defined as $(1-\mathcal{L}^{ham})$, where $\mathcal{L}^{ham}$ denotes Hamming loss defined as follows:

$$\mathcal{L}^{ham} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} \oplus(S_{i,j}, T_{i,j}), \tag{14}$$

where n is the number of DSPP images, m is the number of defect classes, $\oplus$ denotes exclusive-or, and $S_{i,j}$ and $T_{i,j}$ are the individual model output and its corresponding label for the j-th class of i-th sample, respectively. Here, accuracy means the average ratio of the correctly determined labels among the total number of labels for each prediction.

TABLE 2: Impact of EF and PF on Segmentation.

Features		Per Point Acc.		mIoU	
PF	EF	Fixed	Various	Fixed	Various
		79.38	77.58	62.75	60.76
	✓	81.03	83.79	64.08	68.51
✓		86.33	87.26	72.15	73.92
✓	✓	**87.26**	**87.47**	**73.41**	**73.68**

TABLE 3: Impact of the EF and PF on Multi-label Classification.

Features		Exact Match		F1		mAP	
PF	EF	Fixed	Various	Fixed	Various	Fixed	Various
		89.16	85.09	94.67	92.76	98.32	97.10
	✓	89.90	85.58	95.28	93.46	98.82	98.03
✓		91.52	86.86	95.81	93.26	98.59	97.69
✓	✓	**91.86**	**90.01**	**96.16**	**95.35**	**99.07**	**98.77**

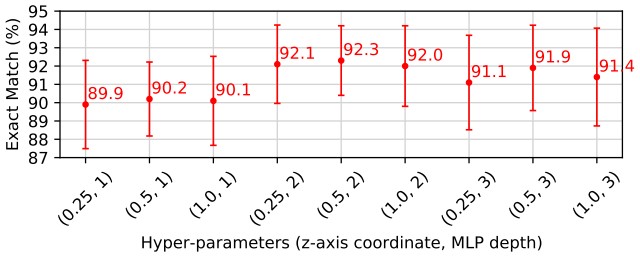


FIGURE 9: Cross-validation for hyper-parameter tuning.

B. IMPLEMENTATION DETAILS

For all experiments, we used the Adam optimizer [30] where the number of epochs, batch size, and learning rate were set to 150, 32, and 10^{-4}, respectively. The learning rate was multiplied by 0.1 after 80 and 120 epochs. In position feature extraction, we kept the aspect ratio for all experiments except Table 4. For the hyper-parameter setting of the point cloud feature encoder and the segmentation decoder, we applied the same setting as in the PointNet2 paper [26]. In each MLP layer, we used a 512-dim hidden layer with ReLU activation and a dropout layer with the probability of 0.5. Without early

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

TABLE 4: Impact of the Normalization Methods.

Normalization method	Segmentation Metric				Multi-label Classification Metric					
	Per Point Acc.		mIoU		Exact Match		F1		mAP	
	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various
Fix aspect ratio	87.27	87.75	73.19	74.80	92.25	90.30	96.38	95.26	99.16	98.89
Keep aspect ratio	**87.27**	**88.61**	**73.19**	**75.90**	**92.25**	**92.36**	**96.38**	**96.27**	**99.16**	**99.07**

TABLE 5: Comparison of Model Sizes (Number of Param. and Feature Map Sizes) of the Proposed Model with CNN-based Models.

Model	# of Param.	Feature Map Size Before MLP
ResNet50 + SPP(1) + MLP	1.8 M	$1 \times 1 \times 1024$
ResNet50 + SPP(3) + MLP	3.8 M	$3 \times 3 \times 1024$
ResNet50 + SPP(1, 3) + MLP	4.0 M	$(3 \times 3 + 1 \times 1) \times 1024$
InceptionV4 + SPP(1) + MLP	25.2 M	$1 \times 1 \times 512$
InceptionV4 + SPP(3) + MLP	29.4 M	$3 \times 3 \times 512$
InceptionV4 + SPP(1, 3) + MLP	30.0 M	$(3 \times 3 + 1 \times 1) \times 512$
MarsNet [4]	18.0 M	$(4 + 4 + 8 + 8) \times 512$
D^{3}PointNet (w/o EF, PF, seg.)	**1.8M**	1×512
D^{3}PointNet (w/o seg.)	1.8M	1×512
D^{3}PointNet	4.2M	1×512

stopping, we measured the performance of each model after training ends.

For the hyper-parameter tuning of the z-axis coordinate and MLP depth, we performed a 4-fold cross-validation on the various-size training set, as shown in Fig. 9. We split the training set by their PCB type, so that each fold contains one type of PCB. In the experiment, we evaluated nine models varying the z-axis coordinate value from 0.25 to 1 and increasing the number of MLP layers from 1 to 3. As a result, the optimal hyper-parameters, z-axis coordinate value of 0.5 and the number of MLP layers of 2, were used in the subsequent experiments. All experiments were conducted on a workstation with a Intel Core i7 8700K, 64 GB of RAM and an Nvidia GTX2080Ti GPU. It took approximately 15 hours to train our model to convergence.

C. ABLATION EXPERIMENTS

1) Multi-task Learning Architecture

Table 1 reports the impact of multi-task learning architectures. We compared several cases for multi-task learning: when only one of the two tasks, segmentation and multi-label classification, was performed using one decoder (rows 1-2), when both tasks were performed using two decoders (row 3), and when the proposed input and score concatenations were used with two decoders (rows 4-6). The number of parameters of the model is reported for each case. Compared to using separate models for each task, the multi-task learning baseline, which shares the encoder and uses two decoders, saved approximately 1.0 M parameters. The performance, however, was similar or slightly lower. The performance improved using the proposed architectures, input and scores concatenations, at the expense of extra 0.4 M parameters. The improvement was greater in multi-label classification than in segmentation, and as well as for the various-size dataset than the fixed-size dataset.

2) Impact of the Hand-crafted Features

Tables 2 and 3 show the impact of EF and PF. In this experiment, we excluded the influence of multi-task learning and measured the effects of the EF and PF by using only the segmentation decoder in Table 2, and using only the multi-label classification decoder in Table 3. Both proposed EF and PF improved performances of the segmentation and the multi-label classification. When PF was used alone, the performance enhanced more than when only EF was used. The performance was at the peak when EF and PF were used together. Similar to the first experiment, when the proposed features were used, the performance improvement was greater for the various-size dataset than for the fixed-size dataset. This proves the utility of the proposed features in handling the inputs of various sizes.

3) Impact of the Normalization Methods

We compared the two options for normalizing the input point clouds: keeping the aspect ratio and fixing aspect ratio to 1:1. As seen in Table 4, keeping aspect ratio showed superior performance to that of fixing aspect ratio to 1:1, in both segmentation and multi-label classification of various-size dataset. Meanwhile, for the fixed-size dataset, there was no difference between the two normalization methods because the aspect ratio had been already 1:1 from the beginning.

D. COMPARISON WITH CNN-BASED MULTI-LABEL CLASSIFICATION MODELS

In Tables 6 and 7, we compared the multi-label classification performance[4] of the proposed model with CNN-based models, including the state-of-the-art in DSPP image dataset, MarsNet [4]. The CNN models have a structure of a CNN + a pooling layer + an MLP, which is a typical structure in the defect classification fields [2], [11], [12], [21], [22]. CNNs extract feature maps, pooling layer transforms the feature maps into a fixed-size feature vector, and MLPs act as a classifier. In this experimental analysis, we used either ResNet50 [6] or InceptionV4 [5] as the backbone CNN pre-trained on the ImageNet-1k image classification dataset [31]. For the pooling layer, we used the spatial pyramid pooling (SPP) [32] with max pooling sizes of either 1×1 or 3×3, or both. For the MLP, we used a 512-dim hidden layer with

[4]We conducted five trainings and tests for each model for reporting the average performance and standard deviation in the test set.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

TABLE 6: Comparison of Classification Results (AP in %) of the Proposed Model with CNN-based Models.

Model	SBD		SD		RA		SNK		CD		mAP	
	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various
ResNet50 + SPP(1) + MLP	99.84	97.84	93.51	84.74	99.79	98.84	95.82	80.09	92.90	95.08	96.37±0.21	91.32±0.79
ResNet50 + SPP(3) + MLP	99.68	98.08	96.87	90.02	99.84	99.62	96.18	92.75	96.85	92.32	97.88±0.14	94.56±0.68
ResNet50 + SPP(1, 3) + MLP	99.84	98.98	97.91	88.27	99.88	99.64	96.53	91.38	96.39	94.35	98.11±0.10	94.52±0.59
InceptionV4 + SPP(1) + MLP	99.88	98.01	97.44	90.10	99.92	98.44	97.67	92.50	97.92	91.17	98.57±0.20	94.04±0.40
InceptionV4 + SPP(3) + MLP	99.90	97.97	97.02	94.09	99.93	99.03	97.89	92.68	97.83	94.67	98.51±0.18	95.69±0.45
InceptionV4 + SPP(1, 3) + MLP	99.89	98.75	98.22	93.83	99.92	99.18	97.76	92.10	98.79	94.54	98.91±0.15	95.68±0.32
MarsNet**	99.80	98.84	98.31	94.78	99.77	99.19	97.70	93.42	98.98	95.24	98.91±0.13	96.29±0.23
D^{3}PointNet (w/o EF, PF, seg.)	99.89	99.75	98.11	97.80	99.88	99.79	98.18	96.42	98.96	96.29	99.00±0.10	98.01±0.16
D^{3}PointNet (w/o seg.)	99.96	99.94	98.17	97.72	99.96	99.95	98.26	97.72	99.02	98.51	99.07±0.08	98.77±0.09
D^{3}PointNet	**99.97**	**99.96**	**98.55**	**98.40**	**99.97**	**99.95**	**98.30**	**98.10**	**99.04**	**98.95**	**99.16±0.08**	**99.07±0.08**

** Our re-implementation.

TABLE 7: Comparison of Classification Results (Exact Match, F1, Accuracy, Precision, Recall in %) of the Proposed Model with CNN-based Models.

Model	Exact Match		F1		Accuracy		Precision		Recall	
	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various	Fixed	Various
ResNet50 + SPP(1) + MLP	81.48±0.91	71.11±0.83	91.29±0.68	84.94±0.33	93.51±0.43	88.54±0.33	94.07±0.60	89.19±0.30	88.67±0.70	81.08±0.36
ResNet50 + SPP(3) + MLP	84.74±0.89	75.55±0.71	93.18±0.63	88.66±0.27	93.12±0.37	91.09±0.27	95.56±0.59	92.43±0.24	90.92±0.67	85.19±0.29
ResNet50 + SPP(1, 3) + MLP	86.06±0.82	74.35±0.61	93.72±0.58	87.72±0.24	93.87±0.32	91.92±0.24	95.97±0.51	91.80±0.21	91.58±0.60	84.00±0.24
InceptionV4 + SPP(1) + MLP	88.67±0.78	72.22±0.70	94.32±0.44	86.12±0.34	94.12±0.18	93.84±0.17	96.21±0.40	90.38±0.31	92.51±0.46	82.24±0.37
InceptionV4 + SPP(3) + MLP	88.32±0.65	79.24±0.55	94.43±0.36	89.96±0.30	94.51±0.16	94.19±0.15	96.37±0.32	93.33±0.28	92.57±0.39	86.82±0.28
InceptionV4 + SPP(1, 3) + MLP	89.26±0.63	78.89±0.54	94.79±0.36	89.42±0.27	94.78±0.15	94.55±0.14	96.45±0.31	92.86±0.24	93.19±0.38	86.22±0.27
MarsNet*						95.11*				
MarsNet**	89.11±0.41	81.40±0.60	94.78±0.27	91.10±0.32	94.72±0.13	95.20±0.18	96.49±0.24	93.83±0.29	93.13±0.29	88.52±0.34
D^{3}PointNet (w/o EF, PF, seg.)	89.16±0.35	85.09±0.47	94.67±0.22	92.76±0.22	94.80±0.13	95.91±0.11	96.51±0.20	95.03±0.22	92.89±0.21	90.59±0.28
D^{3}PointNet (w/o seg.)	91.86±0.31	90.01±0.27	96.16±0.18	95.35±0.14	96.84±0.07	97.41±0.09	97.56±0.16	96.88±0.13	94.73±0.20	93.87±0.17
D^{3}PointNet	**92.25±0.27**	**92.36±0.21**	**96.38±0.14**	**96.27±0.14**	**97.17±0.09**	**97.87±0.08**	**97.58±0.12**	**97.28±0.13**	**95.20±0.16**	**95.28±0.16**

* As reported in [4]. ** Our re-implementation.

ReLU activation and a dropout layer with the probability of 0.5, which is the same structure used in D^{3}PointNet.

We re-implemented MarsNet, employing a modified dilated residual network with 22 layers (mDRN-D-22) as a backbone, hierarchical vertical pooling (HVP) with pooling sizes of 8×1, 1×8, 4×1 and 1×4, and two MLPs as a multi-label scoring module and a threshold estimation module, respectively, as proposed in [4]. The structure of MLPs in MarsNet equals to that of other models. All the training strategies for CNN-based models, including the optimizer, were the same as those of the proposed model as mentioned in V-B.

Note that the MLP structures of all the models used in the experiments are the same, but the size of the MLP's input feature (the output feature of the pooling layers) is different for each model. Table 5 lists the size of the MLP input feature and the overall size of the models in terms of the number of parameters.

We compare the CNN-based models with our proposed model, D^{3}PointNet, in which all the proposed techniques are applied (row 10). For a fair comparison, we also report the result of our proposed model without EF, PF, and segmentation decoder (rows 8-9), which eliminates the benefits from the hand-crated features and the multi-task learning.

Table 6 shows the comparison on AP for each defect class and mAP for an overall evaluation. For CNN-based models except MarsNet, the performance tends to increase as the number of parameters increases. However, even when they used approximately 15 times the parameters of the proposed model, they showed lower performance than the proposed model. MarsNet showed the best performance among CNN-based models. However, unlike the proposed model which shows high AP for all classes, CNN-based models including MarsNet show low AP for the CD class and the SNK class.

Table 7 shows the comparison on EM, F1, accuracy, precision, and recall. We also report the accuracy of MarsNet on the various-size dataset, 95.11%, reported in [4]. CNN-based models have significant performance differences in the various-size dataset and the fixed-size dataset. MarsNet, which had the smallest difference among CNN-based models, showed 91.1% performance in the various-size dataset compared to the fixed-size dataset in terms of EM. On the other hand, the proposed D^{3}PointNet showed little difference in performance between the two datasets. The EM of MarsNet, which showed high scores when evaluated with other metrics, is 81.40% in the various-size dataset, implying that it misclassifies roughly 19 out of 100 defect-containing samples. On the other hand, the proposed D^{3}PointNet achieved the EM of 92.3%, a level that can be applied to on-site SMT defect detection. Furthermore, D^{3}PointNet has 3.45 point greater precision and 6.76 points greater recall than MarsNet in the various-size dataset, which implies that D^{3}PointNet makes less false positives than MarsNet.

E. TOLERANCE TO SPARSENESS

We verified that the proposed D^{3}PointNet is robust to sparse data by adjusting the sparsity of the input data. As presented in Fig. 10, we artificially generated more sparse data by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

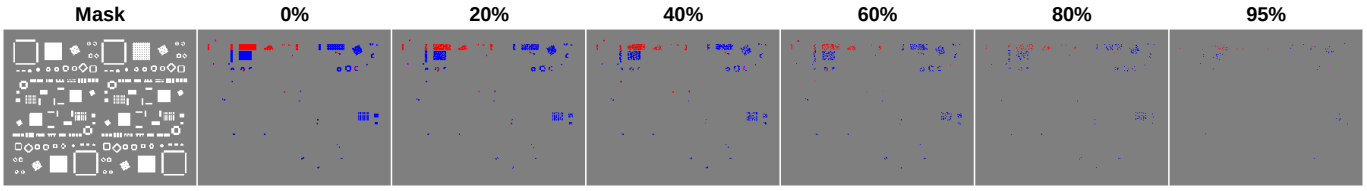


FIGURE 10: Generation of sparse DSPP images using input data dropout. Defective pixels were removed from the DSPP images with the dropout probability, up to 95%.

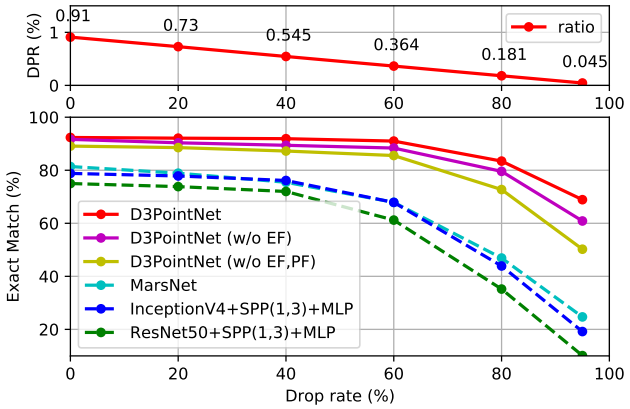
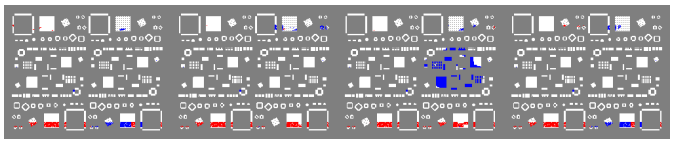
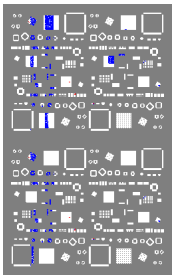


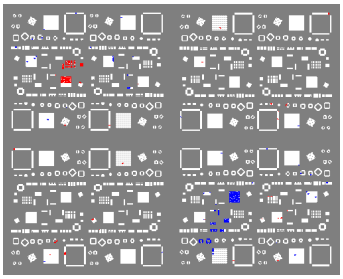
FIGURE 11: Tolerance to input sparsity. The performance changes of the models are shown while increasing input data sparsity by applying dropout to the DSPP image dataset. The degree of sparsity, according to the increase in dropout probabilities, was expressed as a defective pixel ratio (DPR), where a DPR represents the ratio of defective pixels to all pixels in DSPP images.



(a) 400% area, defect class: SNK + CD.



(b) 200% area, defect class: RA + SNK.



(c) 400% area, defect class: SD + SNK.

FIGURE 12: Examples of augmented DSPP images. We generated 105 images and and annotated them for verifying generalization ability of the proposed D^{3}PointNet.

applying dropout up to 95% to the excessive/insufficient channel of the DSPP image. Fig. 11 shows the performance of the models by increasing the dropout probability in the dataset. In addition, sparsity for each dropout probability was measured and indicated by a defective pixel ratio (DPR). Here, a DPR is the average ratio of defective pixels to the total number of pixels in a DSPP image.

As the dropout probability increased from 0% to 95%, the DPR of the dataset decreased from 0.91% to 0.045%. The EM of MarsNet, which showed the best among the CNN-based models, decreased by 69.9%. The EM of the proposed D^{3}PointNet, without EF and PF, decreased by 43.6%, which implies that the network structure of PointNet2 [26] with the proposed position feature extraction is robust to sparse data in DSPP image dataset. Moreover, in the case of D^{3}PointNet using the devised EF and PF, EM only decreased by 25.4%. This indicates that the proposed feature extraction methods, EF and PF, help the proposed network to become more robust to sparse data in DSPP image dataset.

F. GENERALIZATION TO VARIOUS SIZES

We confirmed the generalization ability of D^{3}PointNet through performance evaluation on images with a larger scale than the DSPP images in the training dataset. As displayed in Fig. 12, we created the three augmented test sets from test splits of various-size datasets. Each test set consists of 105 augmented DSPP images, where an augmented DSPP image was created by attaching n source DSPP images with the same PCB but different defect classes ($n = 2, 4, 8$ for each test set). The attachments between the images were made such that the resulting image is rectangular, such as 1×2, 2×2, and 2×4. When the attached source DSPP images belong to different defect classes, the combined DSPP image belongs to all of the defect classes of the source DSPP images. For example, in Fig. 12 (c), a DSPP image of the SD class, a DSPP image of the SNK class, and two DSPP images of the normal class are combined to create a new DSPP image of 400% area, belonging to both the SD and SNK classes. The augmentation was only applied when creating a test set, not when training, to measure the generalization ability of each model.

In the case of CNN-based models, including MarsNet, their performance drastically decreased by merely doubling the area of the test set. In particular, EM of MarsNet decreased by 69.4% in the 800% area dataset. On the other hand, EM of D^{3}PointNet, without EF and PF, fell 19.8% in the 800% area dataset. Moreover, the decrease of EM was

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

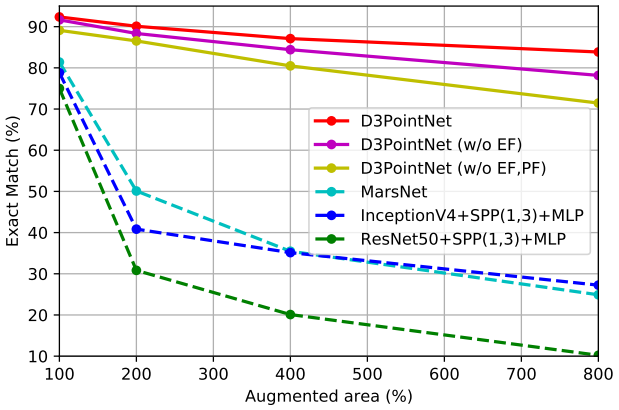


FIGURE 13: Generalization to various sizes. The EMs of D^{3}PointNet and CNN-based models were measured for the datasets augmented to increase the area of the image by 200%, 400%, and 800% compared to the original. The horizontal axis represents the area of the DSPP images generated after augmentation, compared to their original images. Augmented area = 100% indicates that the original test set was used without augmentation.

only 9.2% using D^{3}PointNet with the devised EF and PF, which implies that the proposed feature extraction methods help to improve the generalization ability for various image sizes.

G. QUALITATIVE RESULTS IN SEGMENTATION

We visualized the segmentation results to verify the segmentation performance of our proposed D^{3}PointNet, as shown in Fig. 14. From the leftmost, each column represents the mask channels, defective solder paste pattern channels, ground-truths, and segmentation outputs. The proposed model took the left two columns as inputs and generated the rightmost column as output.

The outputs are almost identical to the ground-truths except for a few failure cases. In the second sample in the figure (row 2), the model incorrectly segmented some of the support defects (SDs) as clamp defects (CDs). However, considering that the SDs appeared near the bottom edge, there are possibilities of both SDs and CDs. Therefore even a human expert cannot distinguish between them. In the fifth sample (row 5), the model incorrectly segmented the insufficient solder paste near the solder no kneading (SNK) area as SNK. This was a reasonable prediction considering the characteristics of the SNK appearing insufficient solder paste patterns in a wide area, but it was actually noise independent of the printer defect. Regardless of whether the input data is relatively dense (rows 1-5) or sparse (rows 6-10), the proposed D^{3}PointNet performed segmentation task successfully.

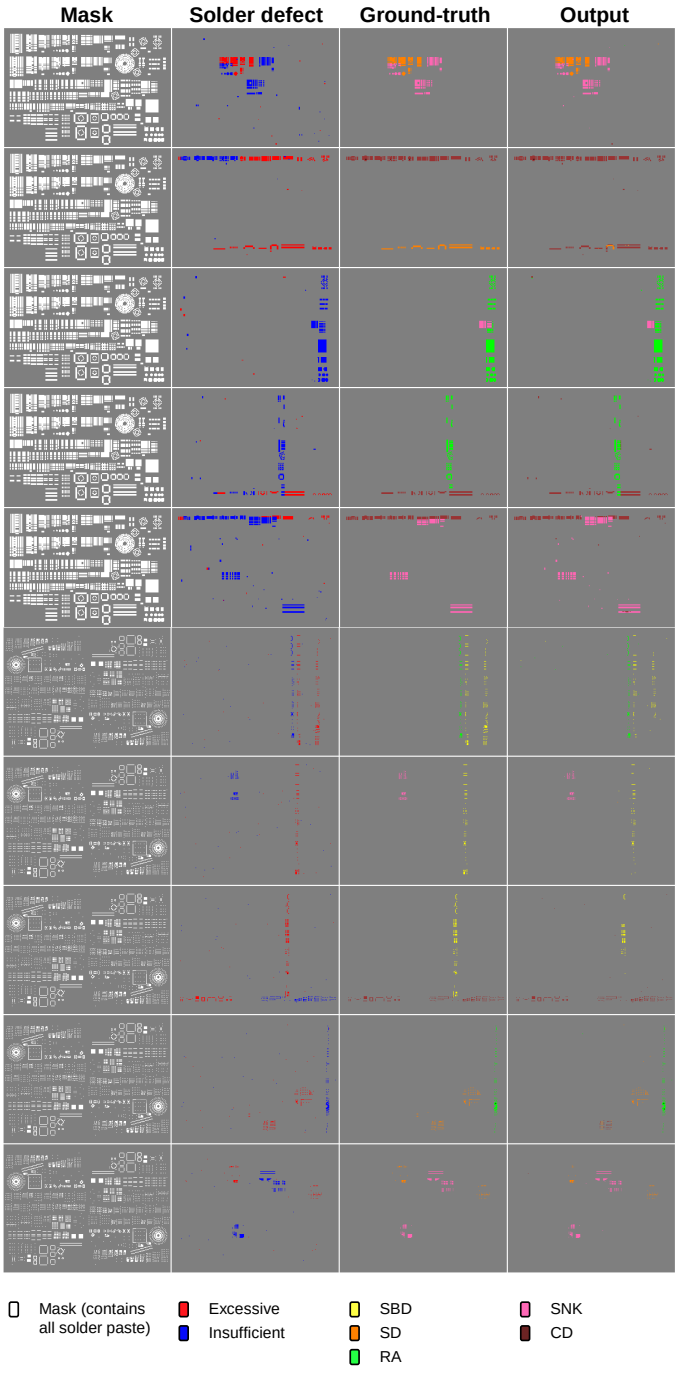


FIGURE 14: Visualization of the segmentation result. From left to right, each column represents masks, defective solder patterns, ground-truths, and segmentation outputs, respectively.

VI. CONCLUSION

In this paper, the problem of defect detection for solder paste printer using DSPP images was addressed. It was pointed out that conventional CNNs are not suitable for the DSPP images due to their sparseness, various sizes, and limited number of data. As a solution, a dual-level defect detection PointNet, D^{3}PointNet, was proposed, where a DSPP image

is first converted into a set of feature points, then the defect detection is performed in two semantic levels: the micro-level and the macro-level. Compared to CNN-based models including MarsNet, which is a state-of-the-art model, the proposed D^{3}PointNet showed more robustness to changes in sparsity and size of input data. Moreover, since D^{3}PointNet provides segmentation as an intermediate result in multi-label classification, it is more useful than the existing method performing only the multi-label classification of DSPP images, in that the segmentation of the D^{3}PointNet indicates the location of the defects.

REFERENCES

[1] R. Prasad, Surface mount technology: principles and practice. Springer Science & Business Media, 2013.
[2] X. Yin, K. Yang, Q. Zhang, and X. Zhang, "Stencil imaging and defects detection using artificial neural networks," in 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing (DASC), pp. 129–134, IEEE, 2018.
[3] D. Oh and I. Yun, "Residual error based anomaly detection using auto-encoder in smd machine sound," Sensors, vol. 18, no. 5, p. 1308, 2018.
[4] J.-Y. Park, Y. Hwang, D. Lee, and J.-H. Kim, "Marsnet: Multi-label classification network for images of various sizes," IEEE Access, vol. 8, pp. 21832–21846, 2020.
[5] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, IEEE, 2016.
[7] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 472–480, 2017.
[8] M. Long, Z. Cao, J. Wang, and S. Y. Philip, "Learning multiple tasks with multilinear relationship networks," in Advances in neural information processing systems, pp. 1594–1603, 2017.
[9] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7482–7491, 2018.
[10] A. P. S. Chauhan and S. C. Bhardwaj, "Detection of bare pcb defects by image subtraction method using machine vision," in Proceedings of the World Congress on Engineering, vol. 2, pp. 6–8, 2011.
[11] L. Zhang, Y. Jin, X. Yang, X. Li, X. Duan, Y. Sun, and H. Liu, "Convolutional neural network-based multi-label classification of pcb defects," The Journal of Engineering, vol. 2018, no. 16, pp. 1612–1616, 2018.
[12] P. Wei, C. Liu, M. Liu, Y. Gao, and H. Liu, "Cnn-based reference comparison method for classifying bare pcb defects," The Journal of Engineering, vol. 2018, no. 16, pp. 1528–1533, 2018.
[13] W.-C. Wang, S.-L. Chen, L.-B. Chen, and W.-J. Chang, "A machine vision based automatic optical inspection system for measuring drilling quality of printed circuit boards," IEEE Access, vol. 5, pp. 10817–10833, 2016.
[14] D. Li, S. Li, and W. Yuan, "Flexible printed circuit fracture detection based on hypothesis testing strategy," IEEE Access, vol. 8, pp. 24457–24470, 2020.
[15] V. S. Ramalingam, M. Kanagasabai, and E. F. Sundarsingh, "Transit time dependent condition monitoring of pcbs during testing for diagnostics in electronics industry," IEEE Transactions on Industrial Electronics, vol. 65, no. 1, pp. 553–560, 2017.
[16] T.-W. Hui and G. K.-H. Pang, "Solder paste inspection using region-based defect detection," The International Journal of Advanced Manufacturing Technology, vol. 42, no. 7-8, p. 725, 2009.
[17] C. Benedek, O. Krammer, M. Janóczki, and L. Jakab, "Solder paste scooping detection by multilevel visual inspection of printed circuit boards," IEEE Transactions on Industrial Electronics, vol. 60, no. 6, pp. 2318–2331, 2013.
[18] J. Li, B. L. Bennett, L. J. Karam, and J. S. Pettinato, "Stereo vision based automated solder ball height and substrate coplanarity inspection," IEEE Transactions on Automation Science and Engineering, vol. 13, no. 2, pp. 757–771, 2015.
[19] C. W. Mak, N. V. Afzulpurkar, M. N. Dailey, and P. B. Saram, "A bayesian approach to automated optical inspection for solder jet ball joint defects in the head gimbal assembly process," IEEE Transactions on Automation Science and Engineering, vol. 11, no. 4, pp. 1155–1162, 2014.
[20] G. Acciani, G. Brunetti, and G. Fornarelli, "Application of neural networks in optical inspection and classification of solder joints in surface mount technology," IEEE Transactions on Industrial Informatics, vol. 2, no. 3, pp. 200–209, 2006.
[21] Y.-G. Kim, D.-U. Lim, J.-H. Ryu, and T.-H. Park, "Smd defect classification by convolution neural network and pcb image transform," in 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), pp. 180–183, IEEE, 2018.
[22] N. Cai, G. Cen, J. Wu, F. Li, H. Wang, and X. Chen, "Smt solder joint inspection via a novel cascaded convolutional neural network," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 8, no. 4, pp. 670–677, 2018.
[23] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in Proceedings of the IEEE international conference on computer vision, pp. 945–953, 2015.
[24] L. Zhang, G. Zhu, P. Shen, J. Song, S. Afaq Shah, and M. Bennamoun, "Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition," in Proceedings of the IEEE International Conference on Computer Vision, pp. 3120–3128, 2017.
[25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660, IEEE, 2017.
[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in Advances in Neural Information Processing Systems, pp. 5099–5108, 2017.
[27] J. Li, B. M. Chen, and G. Hee Lee, "So-net: Self-organizing network for point cloud analysis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9397–9406, IEEE, 2018.
[28] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Oregon State University, Corvallis, vol. 18, pp. 1–25, 2010.
[29] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2285–2294, IEEE, 2016.
[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.

JIN-MAN PARK received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2015, and the M.S. degree in the robotics program from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2017, where he is currently pursuing the Ph.D. degree. His current research interests include data analytics, anomaly detection, and computer vision.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3013291, IEEE Access

YONG-HO YOO received the M.S. and Ph.D. degrees on anomaly detection for industrial fields in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2014 and 2019, respectively. After joining Kohyoung Technology, in 2019, he has been developing diagnosis systems in SMT field. His current research interests include optimization methods for industrial fields, and anomaly detection using spatio-temporal data.

UE-HWAN KIM received the M.S. and Ph.D. degrees on task intelligence for service robots in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2015 and 2020, respectively. He has been a postdoctoral researcher at Robot Intelligence Technology Lab., KAIST since 2020. His current research interests include service robot, cognitive Internet of Things, computational memory systems, and learning algorithms.

DUKYOUNG LEE received the M.S. and Ph.D. degrees on robotics and machine vision in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1999 and 2010, respectively. He joined Samsung Electronics, in 2004, where he had hands-on all rounded experience on control system and SW platform. After joining Kohyoung Technology, in 2015, he has been leading Smart AI Solution Team. His current research interests include SW product line design, distributed data analytics systems, and machine learning.

JONG-HWAN KIM (F'09) received the Ph.D. degree in electronics engineering from Seoul National University, Seoul, South Korea, in 1987. Since 1988, he has been with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, where he is leading the Robot Intelligence Technology Laboratory as a KT Endowed Chair Professor. He is the director of Kohyoung-KAIST AI Joint Research Center and Machine Intelligence and Robotics Multi-Sponsored Research and Education Platform, KAIST. He has authored 5 books and 5 edited books, 2 journal special issues, and around 400 refereed papers in technical journals and conference proceedings. His current research interests include intelligence technology, machine intelligence learning, and AI robots.