IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2019, accepted January 21, 2020, date of publication January 24, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969217

MarsNet: Multi-Label Classification Network for Images of Various Sizes

JU-YOUN PARK[1], YEWON HWANG[2], DUKYOUNG LEE[3], AND JONG-HWAN KIM[2], (Fellow, IEEE)

[1]School of Engineering and Applied Science, George Washington University, Washington, DC 20052 USA
[2]School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
[3]Kohyoung Technology, Inc., Seoul 08588, South Korea

Corresponding author: Jong-Hwan Kim (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 Since the Convolutional Neural Network (CNN) has surfaced and fascinated the world, many researchers have exploited CNN for image classification, object detection, semantic segmentation, etc. However, the conventional CNNs have a pyramidal structure and were designed to process images which have the same size. Although some CNNs can accept images of various sizes, performance is degraded for images smaller than the size of images used for training. In this paper, we propose MarsNet, a CNN based end-to-end network for multi-label classification with an ability to accept various size inputs. In order to allow the network to accept such images, dilated residual network (DRN) is modified to get higher resolution feature maps, and horizontal vertical pooling (HVP) is newly designed to efficiently aggregate positional information from the feature maps. Furthermore, multi-label scoring module and threshold estimation module are employed to serve the purpose of multi-label classification. We verify the effectiveness of the proposed network through two distinctive experiments. We first verify our model by inspecting and classifying multiple types of defects occurred in PCB screen printer using solder paste inspection (SPI) datasets. Secondly, we verify our network using VOC 2007 dataset. Our network is pioneering in that no research has attempted to accomplish multi-label classification for defects in addition to being able to take input images of various sizes in SPI field.

INDEX TERMS Convolutional neural networks, images of various sizes, multi-label classification, printed circuit board, solder paste inspection.

I. INTRODUCTION

Ever since the idea of deep learning has emerged, there has been booming research on deep learning due to its tremendous benefits. Exceptional breakthrough of CNN has made various computer vision applications such as image classification [1], [2], object detection [3], [4], and semantic segmentation [5], [6] possible today. Many industries have shifted their focus to how deep learning can be implemented in their existing technology for a better time efficiency as well as performance. Among many industries that have benefited from the development of deep learning, [7], [8] elevated existing technology by implementing a multi-label classification technique.

In this paper, we propose a novel end-to-end **m**ulti-label cl**a**ssification for images of va**r**ious **s**izes **net**work (MarsNet). Our network can be divided into two meaningful functionalities: an ability to receive various input sizes and multi-label classification. When working with images, CNN is usually utilized as a base model. Standard CNNs typically handle same size inputs. In real life, on the other hand, the size of images comes in a wide range; hence, in order to utilize a standard CNN, a user has to resize the input image. But, this is not ideal since it can cause loss or inaccuracy of contextual information in the original image. Although fully convolutional networks such as InceptionNet and ResNet work when small size inputs are fed in, they show poor performance.

The associate editor coordinating the review of this manuscript and approving it for publication was Jianqing Zhu.

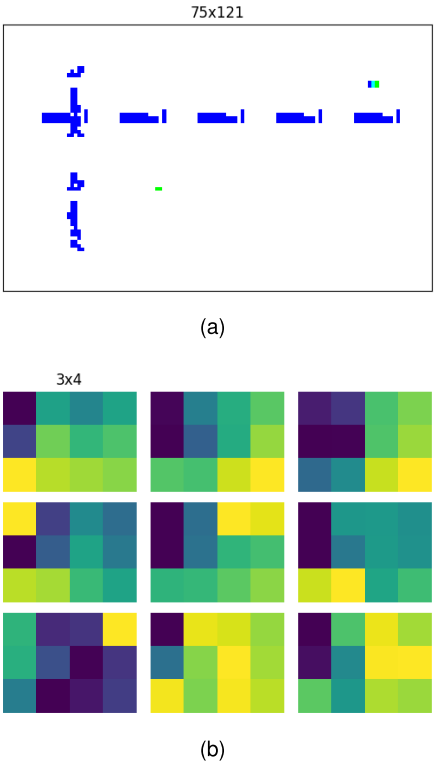


FIGURE 1. Examples of the feature maps. (a) Given the SPI image data with the size of 75 × 121, (b) the extracted feature maps with the size of 3 × 4 from nine randomly chosen filters among a total of 512 filters in the last convolutional layer of ResNet-18.

This is because CNNs generally adopt the structure that reduces the size of feature maps in pyramidal fashion to offset its drawback of using an enormous number of parameters. This way, as the size of the feature maps decreases, highly informative features can be extracted even with a smaller sized filter due to an increasing number of layers. However, this kind of structure diminishes the resolution of the feature maps extracted from the last convolutional layer which contains high-level features. This phenomenon has been brought up as an issue in image segmentation in which each pixel of the input image has to be labeled [9]–[11].

The pyramidal structure becomes particularly problematic when classifying inputs with the sizes smaller than 224×224 or 299×299 which are image sizes that were used in the original paper for ResNet and InceptionNet, respectively, because the size of the feature maps decreases in proportion to the size of an input image. For example, ResNet-18 provides feature maps with a size that is $1/32$ of the input image size for each dimension. Consequently, when the feature maps become too small, neither the proper learning nor the extraction of meaningful information can occur due to the low resolution as shown in Fig. 1. To compensate for this disadvantage, we adopt DRN [12]. DRN replaces the convolutional operators with dilated convolutions [13], [14] for some layers in ResNet. By doing this, the convolutional layers can examine wide receptive fields by using a small size filter. DRN generates reduced feature maps that are 8 times smaller than the input image size for each dimension. However, if the input image is smaller than 224×224, the resolution of the last feature maps is too low to contain positional information. Thus, we build a modified verison of DRN, mDRN, to resolve the low resolution problem by adding more dilated convolutional layers which result in feature maps that are half of the input image size for each dimension.

In order to enable classification regardless of input image size, typically global average pooling (GAP) is used by averaging out all the features extracted from each filter after the last convolutional layer. Later, to further investigate classification of images with multi-scale and multi-size, [15] proposed spatial pyramid pooling (SPP). Similar to other existing methods, SPP aggregates features into the same size vector to enable classification of various sized input. Though its remarkable achievement, their algorithm was verified with no more than two different size input images which is not sufficient enough and fails to address the problem of low resolution in input images. Our work, on the other hand, tackles the low resolution problem in input images which leads to a disappearance of information included in a microscopic part of a small image. The problem of low resolution in input images was often addressed in the field of image segmentation [12], [16], not particularly in image classification. Since the mDRN, which we use to solve low resolution problem, generates high resolution feature maps, our model is able to aggregate more detailed location information of the extracted feature maps. When preserving the detailed and spatial information is required, global average pooling is not appropriate since it can decimate detailed location information in the extracted features. Consequently, we newly design HVP which executes pooling by dividing the last feature maps in horizontal and vertical directions independently such that spatially meaningful features are preserved and extracted from feature maps of various sizes with higher computational efficiency. Higher computational efficiency is possible because HVP can aggregate precise positional information with lower model complexity than Spatial Pyramid Pooling (SPP) [15] for the same precision. Furthermore, by lowering the complexity of the model, MarsNet is able to avoid overfitting.

In order to achieve multi-label classification, our network consists of two modules; the multi-label scoring module and threshold estimation module. For the multi-label scoring module, we use a sigmoid cross entropy loss [17]. Output value of each node in the multi-label scoring module represents the score of the corresponding class. The output value is compared to a label confidence threshold, that is a reference value, and if the output value is greater than the threshold value, then the class is selected. We further improve the network by adding the threshold estimation module which estimates the optimal threshold for each label [18]. Instead of using the same predefined value for all classes such as 0.5, the threshold estimation module is trained to get different optimal values for each class to improve performance.

To demonstrate the effectiveness of MarsNet, it is applied to an SPI task which is usually added as a supplementary component in Surface Mount Technology (SMT) assembly to inspect whether the solder is applied properly on the PCB in a screen printer. To the best of our knowledge, no research has attempted to classify which components in a screen printer are dysfunctional by examining multiple types of defects occurred in a printed image of PCB. We validate our work not only on a customized PCB dataset, but also on VOC 2007 dataset. The superb performance demonstrates the effectiveness of our network and supports pervasive use of our network in a multi-label classification with various input sizes.

The rest of the paper is organized as follows. In Section II, we review the related work. Our proposed MarsNet is described in Section III followed by extensive experimental validation in Section IV. Discussion points follow in Section V, and we finally conclude our work in Section VI.

II. RELATED WORK

A. DILATED RESIDUAL NETWORK

Semantic segmentation has gained tremendous attention over the past few years because it provides the most precise information about the image by classifying every pixel in the image. However, several challenges have brought up in semantic segmentation, one of which is the poor feature resolution caused by pooling and striding which discard detailed spatial information. In order to overcome this challenge, [13], [14] suggested using dilated convolution, also known as atrous convolution, to aggregate multi-scale contextual information without losing resolution for dense prediction. Similarly, [19], [20] employed the atrous algorithm which expands receptive field without increasing the number of parameters by inserting holes into the filters. The low resolution of the feature maps extracted from the last convolutional layer can be detrimental to image classification or image segmentation. On the other hand, DRNs used dilated convolutions to increase the receptive field of the higher convolutional layers to make the last feature maps high resolution [12].

B. SPATIAL PYRAMID POOLING

Existing CNNs typically result in reduced recognition accuracy for various size images because they require same size input. For this reason, preprocessed images are fed into CNNs as opposed to the raw images. To eliminate this issue, [15] presented SPP which generates a fixed-length output regardless of image size or scale by using multi-level spatial bins. This property enables SPP to pool features extracted at various scales. [21] introduced the idea of ''pyramid match'' to find correspondences between unordered two sets of vectors while maintaining robustness to clutter. Later, [22] presented a kernel-based recognition method, Spatial Pyramid Matching, based on the work of [21]. Motivated by SPP, [20] developed Atrous Spatial Pyramid Pooling (ASPP) which adds atrous convolution layers with different dilation rate in parallel to capture the multi-scale information.

C. MULTI-LABEL CLASSIFCATION

The most classical approach to implement multi-label classification is a problem transformation method, where the multi-label problem is transformed into multiple single label problems. For instance, Label Powerset (LP) considers each set of labels as one class, which makes the new transformed problem a single label classification task. This method, however, suffers from an extremely complex computation. One of the alternatives for better computation include Binary Relevance (BR) [23] which works by decomposing the multi-label task into multiple independent binary learning tasks. One downside of BR is that it neglects correlation between labels. In order to compensate this downside, Classifier Chains (CC) was proposed [24]. In order to boost the performance of multi-label classification, some suggested combining several models instead of using just one, which is called the ensemble method. The ensemble methods which are improved upon LP and CC are RAKEL [25] and ECC [24], respectively. Others presented adapting existing single label algorithms to suit the multi-label classification, called adapted algorithm method. Some of the well-known adapted algorithms are ML-KNN [26] and ML-DT [27], algorithms developed based on kNN and decision tree for multi-label problem, respectively. Over the past few years, since deep learning has gained tremendous popularity, a lot of research on multi-label classification using neural network have been actively conducted. BP-MLL is the first algorithm that utilized neural network on multi-label classification problem [28]. In order to boost the performance of multi-label classification problems using neural networks, [17] investigated the limitations of BP-MLL by replacing the ranking loss with the cross entropy loss function. Later, [18] further improved multi-label classification by introducing a new loss function for pairwise ranking and incorporating a label decision module into the model. Recently, [29] showed state-of-the-art results by applying RNN in addition to CNN to exploit semantic label dependencies in an image.

D. SMT INSPECTION

There have been a wide range of studies on inspecting defects in PCB from using traditional machine learning methods such as random forest [30] to using more modern deep learning approach, utilizing multi-layer perceptron neural network and convolutional neural network [31], [32]. Another neural network based study incorporates fuzzy rule-based method to correct any possible misclassification made by the neural network module [33]. In addition, recently there was a research which attempts to detect capacitor in PCB using YOLO algorithm [34]. However, prior studies are confined to either merely evaluating whether or not a component is flawed after detecting the mounted components or identifying each defect using a single-label classification approach. This paper, on the other hand, tackles defect identification in SMT, specifically in SPI using multi-label classification.

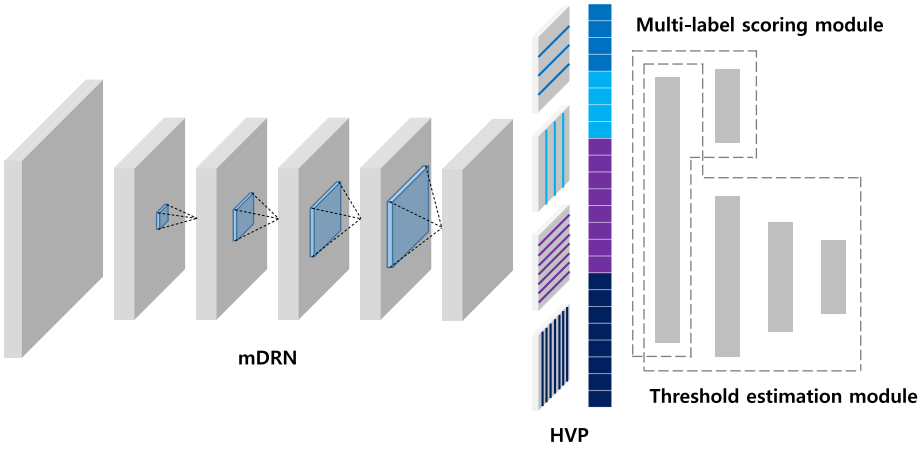


FIGURE 2. Structure of the proposed network.

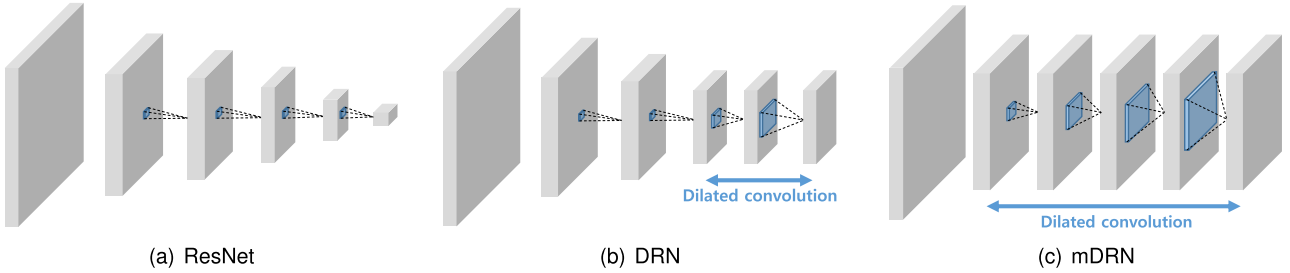


FIGURE 3. Structural comparison of (a) ResNet, (b) DRN, and (c) mDRN. Each gray cube represents the output feature maps for each level in a network. The blue region is the receptive field that is aggregated to the next level pixel connected by four dotted lines. (a) ResNet has progressively shrinking output feature maps while (b) DRN maintains the sizes of feature maps at the higher two levels. (c) The sizes of feature maps in mDRN are the same after downsampling once at the beginning.

III. PROPOSED NETWORK

In this section, we propose a novel network, MarsNet, that performs a multi-label classification for images of various sizes. Fig. 2 shows the structure of the proposed MarsNet which is built based on the mDRN. It is consisted of a Horizontal and Vertical Pooling (HVP), a multi-label scoring module, and a threshold estimation module. The mDRN and the newly designed HVP allow classification of images of various sizes. To perform a multi-label classification, a sigmoid cross entropy loss is employed to train our proposed network. The threshold estimation module is additionally applied to improve multi-label classification performance.

Let X_i be an input image and $Y_i = \{Y_{ij} \mid \forall j \in [1, M]\}$ be a set of corresponding ground-truth class labels, where Y_{ij} is the individual class label in Y_i, defined as follows:

$$Y_{ij} = \begin{cases} 1, & \text{when } j \text{ is relevant to } i \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

$i = 1, 2, \cdots, N, j = 1, 2, \cdots, M, N$ is the number of image data samples, and M is the number of classes. $\hat{Y}_i = \{\hat{Y}_{ij} \mid \forall j \in [1, M]\}$, on the other hand, denotes a set of $\hat{Y}_{ij}$ which is determined by MarsNet. For instance, suppose there are five classes for a given input image X_1. When the first and the third classes are relevant to the image, the set of ground-truth class labels can be denoted as $Y_1 = \{1, 0, 1, 0, 0\}$. MarsNet attempts to output $\hat{Y}_1$ that is equal to Y_1.

A. CLASSIFICATION FOR IMAGES OF VARIOUS SIZES

To classify images of various sizes, convolutional layers are built based on mDRN, and spatial information of features in the last feature maps is aggregated via HVP.

1) MODIFIED DILATED RESIDUAL NETWORK

An l-dilated convolution with the stride s, $*_{l,s}$ over the input feature map f for each location $\mathbf{n}$ in the output of the l-dilated convolution and a filter g can be defined as follows:

$$(f *_{l,s} g)[\mathbf{n}] = \sum_{\mathbf{m}} f[s\mathbf{n} - l\mathbf{m}]g[\mathbf{m}] \tag{2}$$

where l is the dilated factor. Note that the standard convolution, $*$ is the same as $*_{1,1}$.

DRN-A-16 [12] employs dilated convolutions at the fourth and fifth levels in ResNet-18 [35] to solve low resolution issue that occurs in traditional CNN models. DRN-C-26 further improves DRN-A-16 by eliminating gridding artifacts with the degridding scheme. DRN-D-22, a simplified version of DRN-C-26, is employed as our backbone network. Since DRN-D-22 utilizes 2 and 4-dilated convolutions at the first two levels of the four levels, the size of the last feature maps is reduced by a factor of 8 for each dimension compared to the input image size. For example, if the size of an SPI data image is 48×48, the size of the last feature maps becomes 6×6. The low resolution output feature

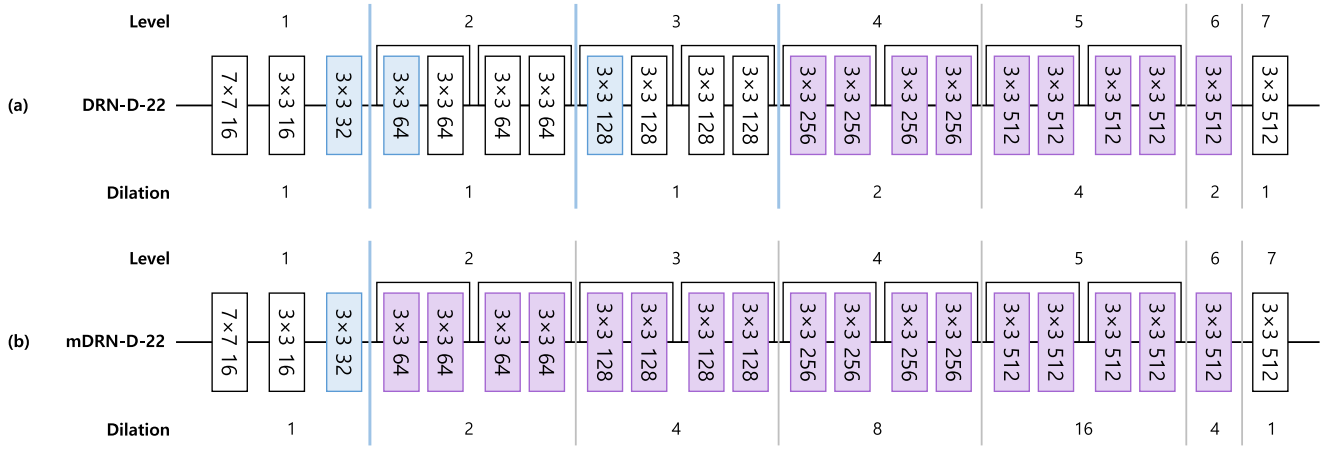


FIGURE 4. Layer architectures of (a) DRN-D-22 and (b) mDRN-D-22. Each rectangle represents a Conv-BN-ReLU group, and the numbers specify the filter size and the number of channels in that layer. Blue colored rectangles are Conv-BN-ReLU groups with stride 2, and downsampling occurs in blue lines. The purple colored rectangles adopt dilated convolutions rather than standard convolutions with the dilated factors described beneath each network architecture. The networks are divided into levels, such that all layers within a given level have the same dilation and spatial resolution.

maps will cause performance degradation. To deal with a wide range of PCB sizes, we modify DRN-D-22 to obtain higher resolution feature maps and call this modified network mDRN-D-22.

Fig. 3 depicts the structure of feature maps of different networks. The feature maps of ResNet are progressively scaled down by striding, and standard convolutions are applied to all four levels, while DRN adopts dilated convolutions at levels 4 and 5, and the size of feature maps is maintained at those levels. To get higher resolution feature maps, we modify DRN by adopting dilated convolutions at all four levels. Specifically, our mDRN-D-22 adopts dilated convolutions with the dilation factors 2, 4, 8, and 16 at the levels from 2 to 5, as shown in Fig. 4. Thus, the size of the final feature maps is half of the input image size for each dimension, and the resolution of the feature maps is high enough to contain spatial information, which will then be aggregated for classification. This allows classification even if the input image is smaller than 224 × 224. In addition, the dilation factor of the level 6 convolution layer is increased from 2 to 4, thereby mitigating gridding artifacts more appropriately for higher dilation factors of mDRN-D-22 than DRN-D-22.

The mDRN-D-22 appears to have the same network structure as the DRN-D-22, including the channel sizes and the number of layers. However, unlike the DRN-D-22, the mDRN-D-22 applies convolutional layers that use dilated convolution (the purple colored rectangles in Fig. 4) at levels 2, 3, 4, and 5 rather than standard convolution (the blue colored rectangles in Fig. 4). Furthermore, another difference between the mDRN-D-22 and the DRN-D-22 is that they use different values for dilation factors. Although the difference in the two models' structure as well as dilation factors may seem trivial in Fig. 4, the mDRN-D-22 empirically proves to generate higher resolution feature maps, resulting successful feature detection in smaller images. The positional information of the higher resolution feature maps, generated from the mDRN-D-22, can be more precisely aggregated using a newly designed pooling layer, which will be explained in the next section.

2) HORIZONTAL VERTICAL POOLING

We propose Horizontal and Vertical Pooling (HVP) to perform a pooling in horizontal and vertical directions alternatively to aggregate spatial feature information from the high resolution feature maps that we obtained from the mDRN. While an ordinary pooling divides the feature map into squares and performs pooling to pool the representative value of each divided space proportional to the feature map, HVP performs a pooling in two different directions: the first part of the pooling occurs in horizontal direction and the second part of the pooling occurs in vertical direction. We define the size of HVP as a vector $(p_1, p_2, \cdots, p_K)$, where a $p_k \times 1$ horizontal pooling and a $1 \times p_k$ vertical pooling are performed for each p_k, and K is the number of pooling processes. After pooling processes, the pooled outputs are concatenated into a vector. The output vector is $(p_1 + p_1 + p_2 + p_2 + \cdots + p_K + p_K) * D = 2D(p_1 + p_2 + \cdots + p_K)$-dimensional, where D is the depth of the input feature map. By performing HVP at multiple sizes, features at different scales can be extracted like SPP. An example of HVP with the size of (4, 8) is depicted in Fig. 5.

Since a horizontal pooling and a vertical pooling are performed in series, the processes can locate the position where the high-level feature is activated like SPP which executes pooling in a square form. Thus, HVP provides similar performance with less computational complexity than SPP. HVP also obtains an output vector that is smaller than that of SPP. Our mDRN-D-22 generates a high-resolution feature map, allowing precise locational information to be extracted from the last feature maps. Therefore, it is important to fully exploit this advantage, and in order to do that, we utilize HVP. The significance of using the proposed HVP after the mDRN is that the feature values are pooled while considering the

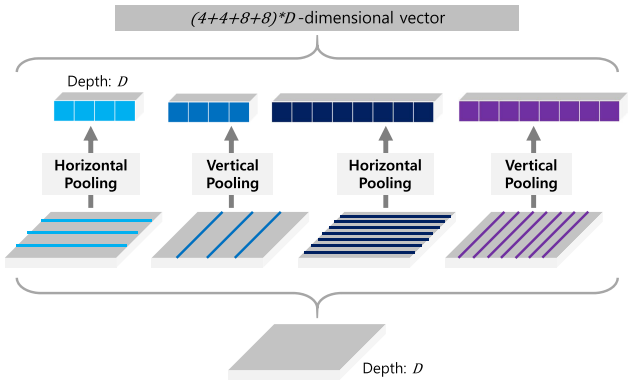


FIGURE 5. Process of horizontal vertical pooling (HVP) with the size of (4, 8) for the feature maps with the depth D. The pooled output vector is $(4+4+8+8)*D = 24D$-dimensional.

location information, allowing a classification for images of various sizes with high performance.

Since SPP has higher model complexity than HVP, SPP may suffer from the overfitting problem. SPP as well as HVP are expected to achieve improved performance when larger pooling size is used. However, for SPP, the number of parameters that is required to be optimized is the squared amount of the pooling size. Therefore, SPP has more difficulty in training with higher dimensions than HVP because it is prone to overfitting. HVP can aggregate finer spatial information from the feature maps with less parameters to be trained and is less likely to face the overfitting problem.

B. MULTI-LABEL CLASSIFICATION

The concatenated output vector from HVP undergoes a fully connected layer, and the output of the fully connected layer enters both multi-label scoring module and threshold estimation module. Then the final class labels are estimated after comparing the outputs from the two modules.

1) MULTI-LABEL SCORING MODULE

The multi-label scoring module includes a fully connected layer. This fully connected layer reduces the dimension of the output vector to match the number of classes. Output value of j-th node in the fully connected layer, $f_{ij} = f_j(X_i)$ represents the score of the corresponding class for the input X_i. The following sigmoid cross entropy loss is used for multi-label classification to train the proposed network:

$$L(X, Y) = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}[Y_{ij}\log(\sigma(f_{ij})) + (1 - Y_{ij})\log(1 - \sigma(f_{ij}))] \quad (3)$$

where $\sigma(\cdot)$ is a sigmoid function.

2) THRESHOLD ESTIMATION MODULE

A label confidence threshold is a reference value for determining whether or not a class should be labeled. To estimate the optimal threshold, the threshold estimation module which consists of three fully connected layers is provided. The first two layers are used to increase the complexity of the module and the last layer is used to reduce the dimension of the output vector to match the number of classes. The module is placed in parallel with the multi-label scoring module. The output of the fully connected layers which is initialized with random weights in the beginning is a vector of each label confidence threshold, $\theta_j \in \mathbb{R}$. When training MarsNet in which threshold estimation module is applied, the following sigmoid cross entropy loss is used to obtain the optimal threshold values:

$$L_{\text{threshold}}(X, Y) = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}[Y_{ij}\log(\delta_\theta^{ij}) + (1 - Y_{ij})\log(1 - \delta_\theta^{ij})] \quad (4)$$

where $\delta_\theta^{ij} = \sigma(f_{ij} - \theta_j)$.

3) DECISION MAKING

A set of class labels for each input image X_i, $\hat{Y}_i$ is decided as follows:

$$\hat{Y}_i = \{\hat{Y}_{ij} \mid \hat{Y}_{ij} = [f_{ij} > \theta_j], \forall j \in [1, M]\} \quad (5)$$

where $[\cdot]$ denotes the Iverson bracket. Based on the above equation, the proposed MarsNet selects the class j whose score value, f_{ij}, from the multi-label scoring module is greater than the corresponding threshold from the threshold estimation module, θ_j.

IV. EXPERIMENTS

It is important to note that Solder Paste Inspection (SPI) task is to classify multiple types of defects occurred in the screen printer by observing the entire PCB image at once. In practice, the size of PCB image which depends on the actual size of PCB comes in a wide range. Therefore, it is different from image segmentation and classification because their intent is to classify multi-scale objects in an image per pixel and object, respectively. Therefore, in addition to evaluating our model on VOC 2007 dataset, we also performed experiments using customized SPI image dataset as to verify MarsNet. The SPI image dataset of various sizes was used to examine differently conditioned models as an ablation study.

A. SPI IMAGE DATASET

Solder Paste Inspection (SPI) inspects the volume of solder paste that is printed on each pad, from which SPI determines whether the paste is printed excessively, insufficiently, or adequately. From this information, an excess map and an insufficient map are generated independently as binary maps. The following describes how excess and insufficient maps are created: first, SPI data is composed of the following information: x and y distances from the top left corner as well as the volume of the paste printed on each pad. Secondly, a 2-dimensional image is created by sorting x-distance/y-distance sequentially where pad exists and making the index of sorted x-distance/y-distance as x-coordinate/y-coordinate. We denoted the binary map as excess/insufficient map if the pixel value

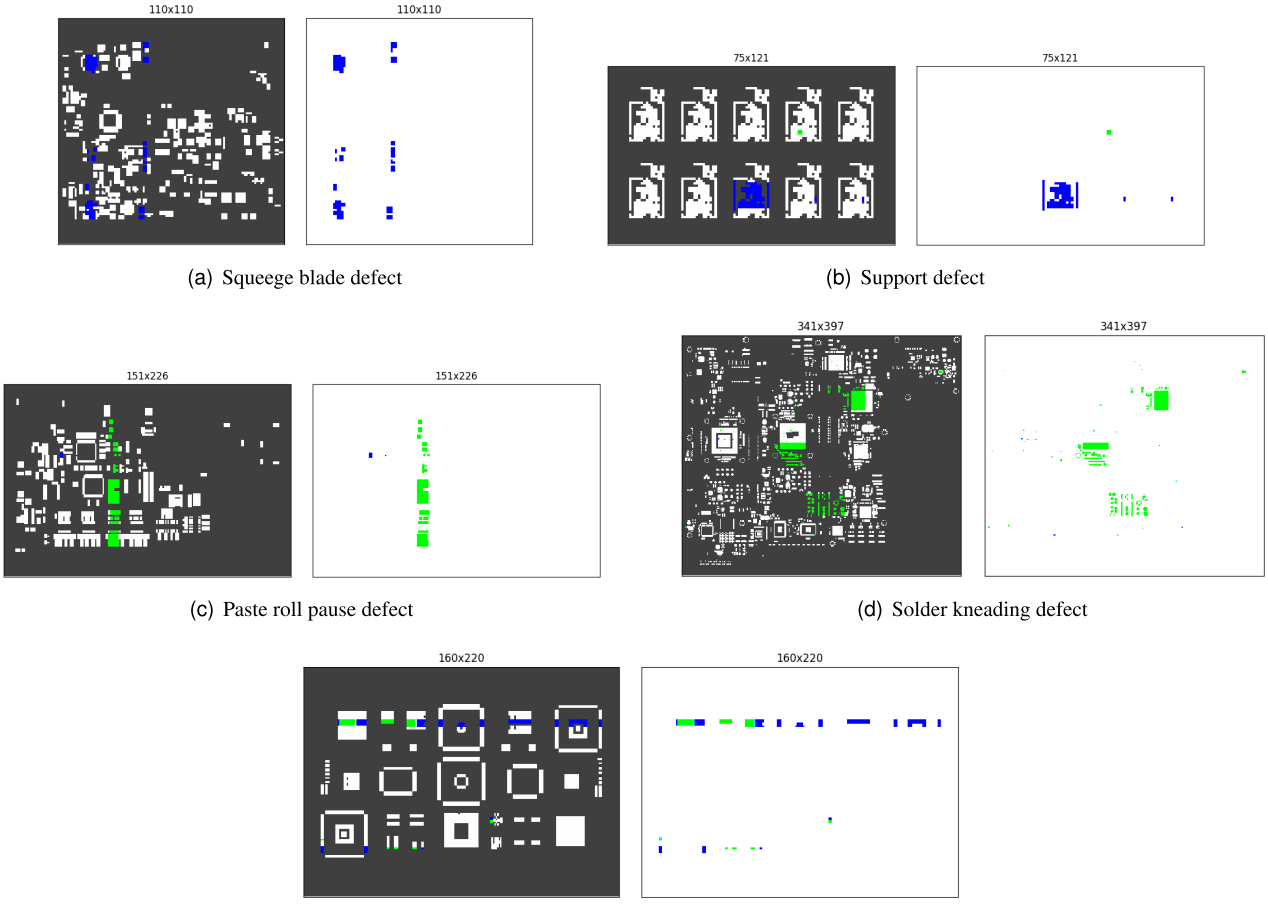


FIGURE 6. Sample SPI image data of each defect: (a) squeegee blade defect, (b) support defect, (c) paste roll pause defect, (d) solder kneading defect, and (e) clamp defect. For each defect, the left image shows the corresponding defect on the PCB and the right image visualizes the SPI image actually used as an input to MarsNet. Each image size is represented on the top of the image. (Blue channel: excess map, Green channel: insufficient map.)

of the pad with excess/insufficient paste volume is 1 and the rest is 0. The SPI image dataset consists of two channel images: one for an excess map and the other for an insufficient map.

Nine different datasets of PCB with various sizes were created, each of which consists of 8,400 SPI images. Four sets of dataset were used to train and the other five sets were used to test. Each image is labeled as the corresponding PCB printer defects. 5 defects which are the common errors that occur in the PCB screen printer were considered: squeegee blade defect, support defect, paste roll pause defect, solder kneading defect, and clamp defect. Sample images of each defect are shown in Fig. 6. We visualize the SPI image as an RGB image: the blue channel for the excess map and the green channel for the insufficient map. For SPI images, pixel in an image with all RGB values of 0 is displayed in white for convenient observation. It is worth noting that more than one defect can appear simultaneously in the same PCB. In this paper, we conducted experiments on SPI datasets which, at most, two defects were present at the same time.

As mentioned, the SPI image dataset consists of nine image sets of different sizes. Therefore, we configured each image batch with the same size images and used it for batch training. There are datasets consisting of images of various sizes other than the SPI image dataset; however, for those datasets, batch training can not be performed because each image has a different size. To the best of our knowledge, there is no dataset that can be organized as image batches with different sizes and has multiple labels on each image at the same time. For this reason, we conducted experiments on the SPI image dataset.

B. TRAINING SETUP

Multiple experiments on SPI datasets of various sizes were conducted for differently conditioned models for an ablation study, as shown in Table 1. All of these models were executed with the multi-label scoring module. The first two fully connected layers with the sizes of 100 and 10 were applied in the threshold estimation module. For the models that the threshold estimation module was not applied, each score from the multi-label scoring module was compared to the threshold

value of 0.0 to determine whether the corresponding defect occurred or not (see (5)). Each model was based on either ResNet-18, DRN-C-26, DRN-D-22, or mDRN-D-22. For the models with HVP, HVP with the size of (8) was performed using a max pooling, while for the models without HVP, global average pooling was applied. It is MarsNet that applies all the threshold estimation, mDRN-D-22, and HVP.

The purpose of the experiments is to detect multiple types of defects that have occurred in PCB. However, it is also important to determine which defects have not occurred. Therefore, the performance was evaluated in terms of $(1 - HL)$ in % where HL denotes Hamming loss defined as follows:

$$HL = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} Y_{ij} \oplus \hat{Y}_{ij} \tag{6}$$

where $\oplus$ denotes the exclusive-or, and Y_{ij} is the individual class label in a set of corresponding ground-truth class labels, Y_i. $\hat{Y}_i$ is a set of $\hat{Y}_{ij}$ which is determined from the network. N is the number of image data samples, and M is the number of classes. $(1 - HL)$ represents the percentage of the correctly determined labels among the total number of labels.

From the SPI image dataset, we formed image batches so that each batch consists of the same size images and used them for batch training. Adam optimizer was used to train the network with a learning rate of 0.01 for the layers in the multi-label scoring and threshold estimation modules, and learning rate of 0.001 was used for the rest of the layers. The learning rate was reduced by a factor of 10 for every 10 epochs. 20 epochs with the batch size of 40 were trained in total. Learning rate is not a significant factor in performance, but learning rate greater than the assigned learning rate can cause overfitting. During testing, each image batch is classified by the trained network, and the averaged accuracy is measured in terms of $(1 - HL)$ in % as mentioned.

C. EXPERIMENTAL RESULTS

1) PERFORMANCE COMPARISONS

The experimental results are summarized in Table 1 and visualized in Fig. 7 for performance comparison at a glance. We will use the term, baseline model, for the model that integrates ResNet-18 with global average pooling layer without the threshold estimation module, as noted in the footnote below Table 1. MarsNet refers to the model that integrates mDRN-D-22 with both HVP and the threshold estimation module. Performance changes based on each condition are clearly displayed. For various input sizes, ResNet-18 alone, without threshold estimation module and HVP, exhibited the lowest accuracy of 92.18%. By either adopting threshold estimation module or HVP in ResNet-18, the accuracy increased by 0.18% and 0.57%, respectively. However, when the threshold estimation module and HVP were applied to ResNet at the same time, accuracy was lower by 0.65% than when HVP was applied without threshold estimation module.

TABLE 1. Experimental results of differently conditioned models where the last model indicates MarsNet in terms of accuracy in % for the SPI image dataset of various sizes.

Threshold Estimation	Convolutional Layers	HVP	Accuracy (%)
×	ResNet-18	×	92.18*
○	ResNet-18	×	92.36
×	ResNet-18	○	93.75
○	ResNet-18	○	93.10
×	DRN-C-26	×	93.01
○	DRN-C-26	×	93.55
×	DRN-C-26	○	94.85
○	DRN-C-26	○	94.49
×	DRN-D-22	×	93.86
○	DRN-D-22	×	93.62
×	DRN-D-22	○	94.79
○	DRN-D-22	○	94.43
×	mDRN-D-22	×	93.96
○	mDRN-D-22	×	93.76
×	mDRN-D-22	○	94.85
○	mDRN-D-22	○	**95.11****

*The baseline model.

**The proposed model, MarsNet.

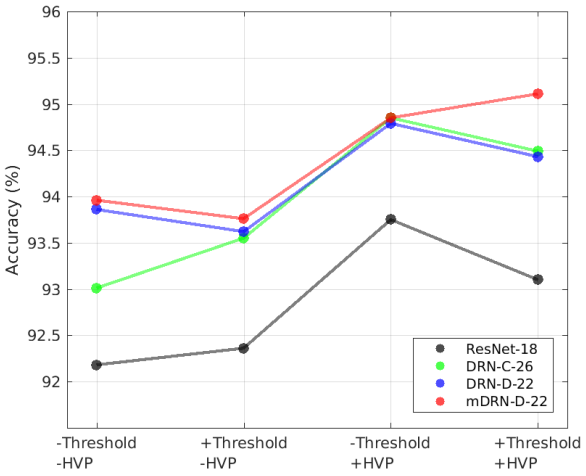


FIGURE 7. Visualization for performance comparison. Differently conditioned models are indicated by + and − signs for an ablation study. + indicates the presence of the feature (threshold estimation or/and HVP), whereas − indicates the absence of the feature. Performances of different convolutional layer architectures are plotted with different colors: black for ResNet-18, green for DRN-C-26, blue for DRN-D-22, and red for mDRN-D-22.

Overall, DRN-C-26 exhibited better performance than ResNet-18 as expected with 94.85% accuracy when HVP was implemented, which was the highest accuracy achieved by the DRN-C-26. DRN-D-22 showed similar accuracy to the DRN-C-26, while the DRN-D-22 showed approximately 1% better accuracy than the DRN-C-26 under the condition where neither HVP nor the threshold estimation module is employed. The mDRN-D-22 achieved similar performance to that of DRN-D-22, but attained slightly higher accuracy than

the DRN-D-22 in all conditions. Regardless of whether either threshold estimation module or HVP was applied or not, both DRN-D-22 and mDRN-D-22 showed strong performance.

However, when both threshold estimation module and HVP are applied, the performance of mDRN-D-22 increases, displaying the highest accuracy among all models, whereas performance of DRN-D-22 rather decreases. In fact, for all models except the following two models: mDRN-D-22 with HVP and ResNet-18 without HVP, applying threshold estimation module rather degraded the classification performance instead of enhancing it. We suspect that when applying HVP, it is possible to aggregate meaningful features in the last feature maps, and only mDRN can extract the last feature maps of high resolution, which contain enough precise positional information to be aggregated by HVP. The low resolution last feature maps of ResNet showed improved performance with the application of the threshold estimation module when HVP was not applied. However, the improved performance was lower than that of mDRN with HVP. This signifies that training an appropriate threshold value for each defect, after spatial features of each set of feature maps are pooled by the appropriate pooling method, improves the classification performance. For SPI tasks with images of various sizes, the best performance was achieved by pooling the high resolution feature maps of mDRN using HVP and applying the threshold estimation module. That is, MarsNet showed the best performance.

MarsNet, mDRN-D-22 with the addition of HVP and the threshold estimation module, showed an accuracy of 95.11%. When going down a row in Table 1, increasing trend in performance from the very first row, the baseline model, to the last row, the proposed MarsNet, can be seen. This is because a model in a row is an improved version of the model that is above by adding additional threshold estimation module or horizontal and vertical pooling layer or both. In addition, change in the convolutional layers also resulted in better performance in the downward direction. The final model, MarsNet, showed accuracy improvement of 2.93% compared to the baseline model which is built upon ResNet-18 with global average pooling without the threshold estimation module.

Experiments were also performed on each PCB SPI image dataset separately based on the size of SPI images. Five PCB datasets with different sizes were provided and the PCB image sizes are shown in Table 2. Each dataset consists of 8,400 PCB images. The baseline model and MarsNet, each with HVP and the threshold estimation module, were individually trained and tested for the given datasets. MarsNet outperformed the baseline model on the four out of the five datasets. MarsNet showed lower performance than the baseline model for the dataset of size 341×397, but the difference is insignificant. In particular, MarsNet showed significantly better performance than the baseline model for datasets with SPI image sizes of 75×121 and 110×110. This result confirmed that MarsNet resolved performance degradation due to the low resolution of the last feature maps of the baseline model for datasets with image sizes much smaller than 224×224.

TABLE 2. Classification accuracies for individual PCB SPI image datasets with different sizes.

SPI Image Size	Accuracy (%)	
	Baseline model*	MarsNet**
151×226	94.92	**95.80**
75×121	84.99	**91.27**
160×220	97.65	**98.13**
341×397	**97.10**	96.62
110×110	90.83	**93.74**

*ResNet-18 with global average pooling layer and without the threshold estimation module.

**mDRN-D-22 with both HVP and the threshold estimation module.

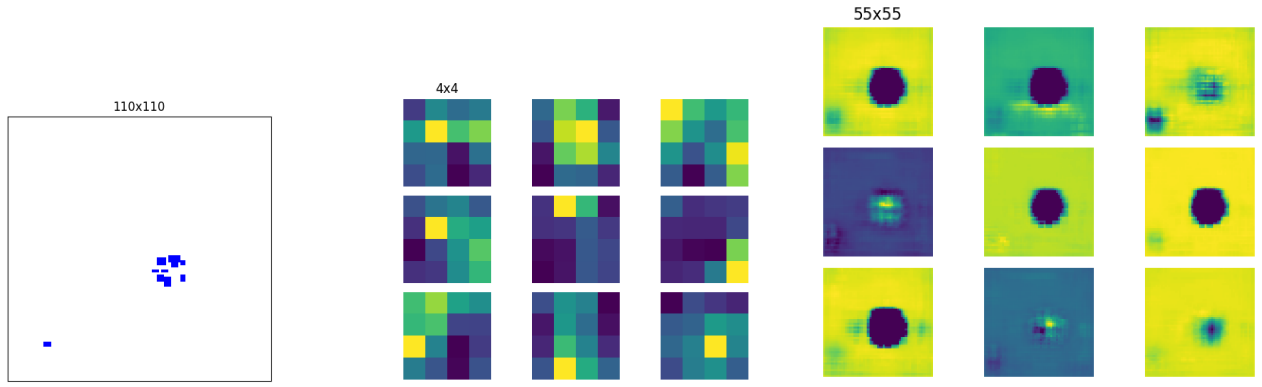
TABLE 3. Classification accuracies for the SPI image dataset to compare the performances of HVP and SPP under the following conditions.

Pooling Layer	Accuracy (%)
SPP $\{1 \times 1, 2 \times 2, 4 \times 4, 8 \times 8\}$	95.00
SPP $\{1 \times 1, 2 \times 2, 4 \times 4, 8 \times 8, 16 \times 16\}$	94.13*
HVP $(1, 2, 4, 8)$	93.80
HVP $(1, 2, 4, 8, 16)$	93.61
SPP $\{8 \times 8\}$	94.80
SPP $\{16 \times 16\}$	85.87*
HVP (8)	**95.11**
HVP (16)	94.83

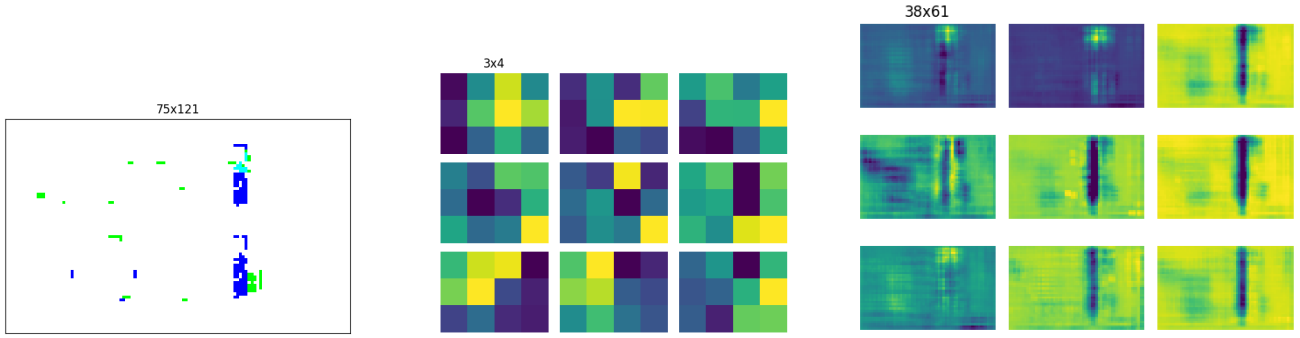
*Trained with the batch size of 30.

2) PERFORMANCE ANALYSES - POOLING LAYER COMPARISON

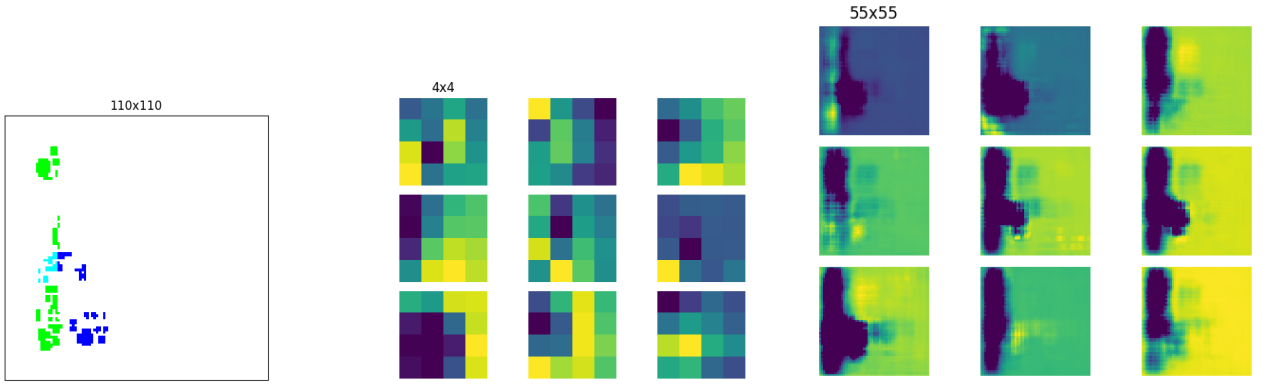
More experiments were conducted to further analyze and compare the performances of HVP and SPP. In Table 3, the first division compares the performances of HVP and SPP when multiple levels of pooling layer are involved, and the second division compares the performances of HVP and SPP when only one level of pooling layer is involved. As shown in the table, HVP with only one level pooling layer with pooling dimension of 8 achieved the highest accuracy. We initially expected performance to achieve better accuracy as the pooling dimension becomes higher. However, the performance decreased as the pooling dimension got bigger. This phenomenon can be seen predominantly in SPP where performance drops dramatically from $\{8 \times 8\}$ to $\{16 \times 16\}$. This is largely due to overfitting. Overall, based on empirical results, we can conclude that conserving the original structure of SPP, i.e. pooling in multiple levels in pyramidal fashion is indeed appropriate, while for HVP, it is more appropriate to perform pooling in just one level.



(a) Left: Ground truth: support defect, Middle: Baseline model's classification: no defect (incorrect), Right: MarsNet's classification: support defect (correct)



(b) Left: Ground truth: squeegee blade and solder kneading defects, Middle: Baseline model's classification: squeegee blade and paste roll pause defects (incorrect), Right: MarsNet's classification: squeegee blade and solder kneading defects (correct)



(c) Left: Ground truth: support and paste roll pause defects, Middle: Baseline model's classification: squeegee blade and paste roll pause defects (incorrect), Right: MarsNet's classification: support and paste roll pause defects (correct)

FIGURE 8. Visualized results of the last feature maps extracted by the baseline model and MarsNet in the middle and on the right, respectively, when each SPI image data on the left was given. Below each SPI image on the left, the ground truth defects are displayed. Below each set of feature maps in the middle and on the right, the defects determined by each model are displayed.

The reason for pyramidal pooling in SPP is to observe an image in multi-scale. Therefore, pyramidal pooling occurs in multiple levels in different sizes. However, pooling multiple times in different scales is not necessary when the mDRN-D-22 is used since it is able to take in multi-scale images as input to the network, which is why HVP is able to achieve higher accuracy in addition to the mDRN-D-22. From this, we can infer that the mDRN-D-22 and HVP complement each other in MarsNet.

Moreover, when training, the batch size was initially set to 40 for all conditions; however, it was later reduced to 30 for some trainings that involved SPP because the GPU memory was not sufficient enough to train when the batch size was set to 40. This indicates that HVP is computationally more efficient than SPP.

3) PERFORMANCE ANALYSES - FEATURE MAP VISUALIZATION

We also visualize how well MarsNet generates the last feature maps with high resolution in Fig. 8. In the figure, the images on the left represent SPI image data that are smaller than 224×224. Given these images as input, the images in the

middle and on the right show the feature maps extracted from nine randomly chosen filters among a total of 512 filters in the last convolutional layer of the baseline model and MarsNet, respectively. The ground truth class labels of the input are defined below each image on the left, and the predicted class labels from the baseline model and MarsNet are respectively defined below each image in the middle and on the right. As shown in Fig. 8, it is clear that the last feature maps of MarsNet which is based on the mDRN-D-22 contains more useful spatial information with higher resolution. The resolution of the last feature maps in the middle column is too poor to precisely extract any features of interest in the images. On the contrary, the images on the right were able to precisely locate where the inadequate amount of solder paste was printed on the PCB. This is the reason why MarsNet was able to correctly detect all the defects, while the baseline model was not able to. This kind of phenomenon could be easily found in small PCBs.

Specifically, Fig. 8(a) shows the example of 'support' defect which is manifested in a circular part in blue. Since the original input SPI image has 110×110 size, the feature maps generated from the baseline model have the size of 4×4 which has insufficient information to correctly classify defects. On the other hand, the feature maps from MarsNet were able to retain the circular features, allowing the 'support' defect to be correctly classified. Likewise, as shown in Fig. 8(b) of the 'squeegee blade' and 'solder kneading' defects example, most feature maps generated from MarsNet, contrary to the baseline model, contain features of 'squeegee blade' defect in the blue vertical line on the right-hand side of the map. Also, the 'solder kneading' defect, which is evidently shown in small green and blue dots in the left half of the input image, is well extracted in the feature maps of MarsNet. Particularly, it can be seen vividly in the left half of of the feature maps of MarsNet as a 180 degree rotated (CW) L-shape in blue and green colors. For the last example of 'support' and 'paste roll pause' defects in Fig. 8(c), the 'support' defect is presented in circular parts in blue and the 'paste roll pause' defect is presented in a green vertical line on the left-hand side of the SPI image. MarsNet precisely aggregated those featured parts in the feature maps of size 55×55.

4) PERFORMANCE ANALYSES - VISUAL PERFORMANCE EXPLANATION

For in-depth analysis, we used Grad-CAM [43] to visualize the performance of HVP and SPP. The images in Fig. 9 are feature maps that are extracted from the Grad-CAM algorithm, where red highlights indicate activated areas by Grad-CAM. The top of the figure is the output of Grad-CAM using HVP and the bottom is the output of Grad-CAM using SPP. Each output image contains five subset images, each of which is the output of Grad-CAM looking for a 'squeegee blade' defect, 'support' defect, 'remove area' defect, 'solder no kneading' defect, and 'clamp' defect from left to right. Each set of image contains results of HVP (top) and SPP (bottom) of the same input PCB image.

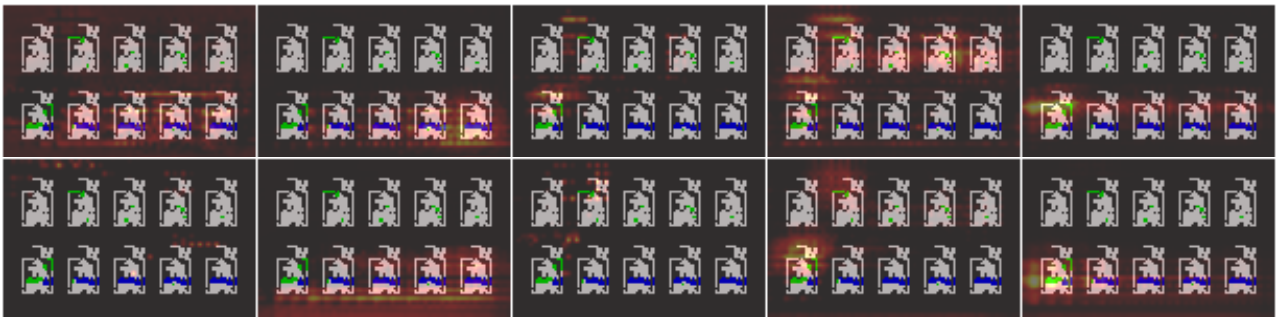
The correct classification for the first set is both of 'solder no kneading' defect and 'clamp' defect, as shown in Fig. 9(a). Looking at the fourth image in particular, Grad-CAM image using HVP focused on the small green areas in the upper half of the image which is crucial for classifying 'solder no kneading' defect accurately. Grad-CAM image using SPP, on the other hand, failed to focus on those areas adequately. Because of this, HVP was able to classify both correctly, while SPP was only able to classify 'clamp' defect correctly. This example manifests HVP's strong performance when small image or feature is given.

Furthermore, the correct classification for the second set is 'squeegee blade' defect, as shown in Fig. 9(b). However, looking at the first and the second subset images, SPP focuses on the areas that are unnecessary for correct classification, whereas HVP only focuses on the area necessary for correct classification. Specifically, for the correct classification, the model should have focused on the blue vertical area in the left hand side in the first image, which both HVP and SPP were able to do; however, SPP not only focused on the blue region, but also other irrelevant regions. For instance, the second image contains no feature that indicates the presence of 'support' defect; however, SPP focused on the meaningless area, leading to an incorrect classification. Yet again, in Fig. 9(c), SPP makes a similar mistake. The correct classification is 'solder no kneading' defect. However, by focusing on the insignificant bottom area in the second image, the model using SPP incorrectly classified the defect as 'support' and 'solder no kneading' defects. Meanwhile, the model using HVP was able to correctly classify the defect by only focusing on the green areas in the upper half of the fourth image. Lastly, the correct classification for the fourth set is both 'remove area' and 'solder no kneading' defects, as shown in Fig. 9(d). However, the model using SPP was unable to classify 'remove area' defect correctly because it missed the long green vertical feature which is pivotal feature for classifying 'remove area' defect in the third image. Contrarily, as shown in the figure, HVP successfully concentrated on the long vertical green area. This example demonstrates that HVP performs more effectively than SPP when either long vertical or horizontal feature is present in the image.

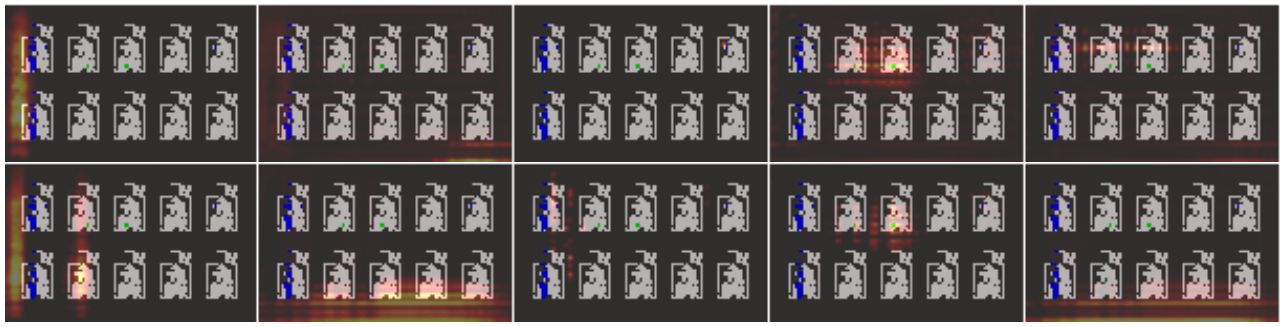
V. DISCUSSIONS

A. MULTI-LABEL IMAGE CLASSIFICATION

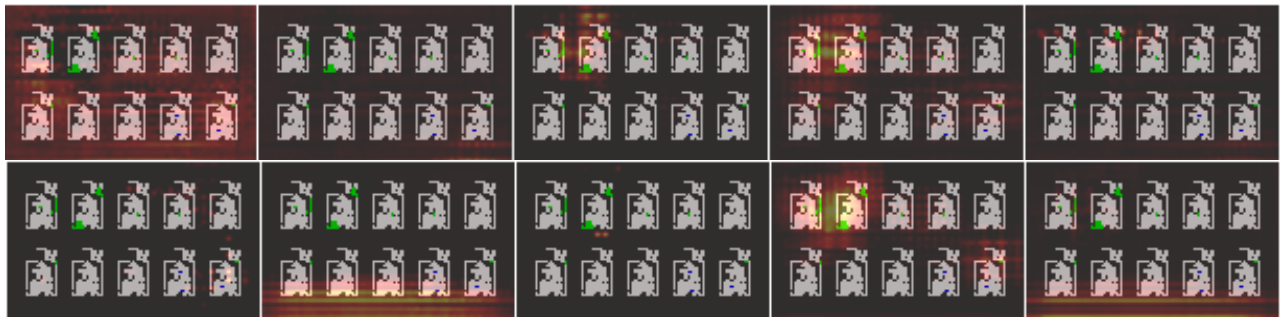
The experiments on the SPI image dataset were conducted through testing the models that were trained with the images having more than 2 different sizes, and such testing process is different from the other benchmark tests publicly available. Nonetheless, in order to compare our proposed network with the other methods, we conducted additional experiments on the VOC 2007 multi-label image classification dataset [44]. Models were trained on the *trainval* dataset of the VOC 2007 dataset and tested on the *test* dataset of the dataset. For data augmentation, resizing, cropping, and horizontal flipping were randomly applied to the *trainval* dataset, while



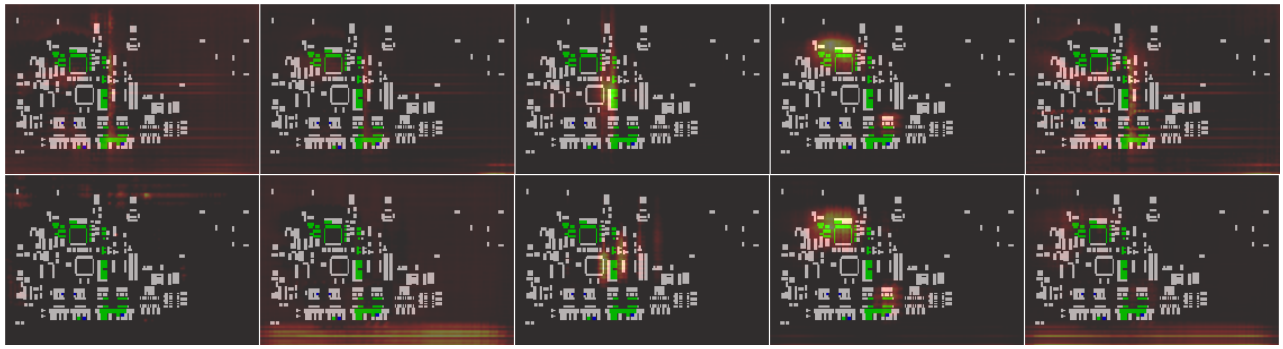
(a) Top: HVP's classification: solder no kneading and clamp defects (correct), Bottom: SPP's classification: clamp defect (incorrect)



(b) Top: HVP's classification: squeegee blade defect (correct), Bottom: SPP's classification: squeegee blade and support defects (incorrect)



(c) Top: HVP's classification: solder no kneading defect (correct), Bottom: SPP's classification: support and solder no kneading defects (incorrect)



(d) Top: HVP's classification: remove area and solder no kneading defects (correct), Bottom: SPP's classification: solder no kneading defect (incorrect)

FIGURE 9. Images of feature map that are extracted from the Grad-CAM algorithm, each of which is the output of Grad-CAM looking for 'squeegee blade', 'support', 'remove area', 'solder no kneading', and 'clamp' defects from left to right. Red highlights show where each model focused on when detecting each defect.

cropping and random resizing were applied to the *test* dataset. For each object class, performances were compared in terms of the average precision as well as the mean of all the average precision, as listed in Table 4. Since our network employs the threshold estimation module to train the best threshold values which obtain the best precision and recall, the performance of our network was measured by the precision for each object class instead of the average precision. Then, we compared the calculated precision with the average precision calculated in the other methods.

TABLE 4. Comparisons on the VOC 2007 multi-label classification dataset with the other methods.

Methods	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
CNN-SVM [36]	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.4	71.8	73.9
I-FT [37]	91.4	84.7	87.5	81.8	40.2	73.0	86.4	84.8	51.8	63.9	67.9	82.7	84.0	76.9	90.4	51.5	79.9	54.3	89.5	65.8	74.5
HCP-2000C [37]	96.0	92.1	93.7	93.4	58.7	84.0	93.4	92.0	62.8	89.1	76.3	91.4	95.0	87.8	93.1	69.9	90.3	68.0	96.8	80.6	85.2
CNN-RNN [29]	96.7	83.1	94.2	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	93.2	75.3	**99.1**	78.6	84.0
RLSD [38]	96.4	92.7	93.8	94.1	71.2	92.5	94.2	95.7	74.3	90.0	74.2	95.4	96.2	92.1	97.9	66.9	93.5	73.7	97.5	87.6	88.5
FeV+LV [39]	97.9	97.0	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	**97.7**	**95.9**	**98.6**	77.6	88.7	78.0	98.3	89.0	90.6
HCP [40]	**98.6**	97.1	**98.0**	95.6	75.3	**94.7**	95.8	**97.3**	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	**94.7**	76.2	97.9	91.5	90.9
RNN-Attention [41]	**98.6**	**97.4**	96.3	**96.2**	75.2	92.4	96.5	97.1	76.5	92.0	**87.7**	96.8	97.5	93.8	98.5	**81.6**	93.7	82.8	98.6	89.3	91.9
Atten-Reinforce [42]	**98.6**	97.1	97.1	95.5	**75.6**	92.8	96.8	**97.3**	**78.3**	**92.2**	87.6	96.9	96.5	93.6	98.5	**81.6**	93.1	**83.2**	98.5	89.3	**92.0**
MarsNet	96.3	91.2	95.0	69.9	60.8	78.2	**99.8**	94.6	68.3	87.3	80.0	**97.6**	93.1	73.2	93.9	76.6	89.7	66.6	90.1	**91.7**	84.7

Since our network needs to be trained under the same experimental setting as the other methods, we used a different backbone network and training setup. The VOC 2007 dataset consists of images of objects, and the trained model classifies multiple objects presented on each image. Objects in an image represent more complicated features than excess or insufficient map in a SPI image, hence we used a deeper network, mDRN-D-38, which is the modified version of DRN-D-38. Similar to the mDRN-D-22, the mDRN-D-38 employs dilated convolutions to the levels from 2 to 5 while DRN-D-38 employs dilated convolutions to the 4 and 5 levels. mDRN-D-38 has 6, 8, 12, and 6 numbers of Conv-BN groups for the levels 2, 3, 4, and 5, respectively. Such structure can extract more complicated features from object images. Since all the images have size of 224×224, the size of the last feature maps from our network is 112×112. We used the (4) size HVP. During training with the batch size of 40, the best performing model up to 50 epochs was stored. The learning rate was set to 0.00001 for the feature layers and 0.0001 for the other layers, and the value of learning rate was decreased by 0.1 every 20 epochs.

All the compared methods are based on CNN. Recently proposed methods for multi-label classification adopt a subsidiary process such as a region proposal strategy or consideration of label correlations. In this section, some compared methods use the region proposal strategy to find several possible regions containing a certain object in an image and classify the object in each region [37], [39]–[41]. Some other methods consider the correlations between class labels to enhance multi-label classification performance [29], and the rest methods use both region proposal strategy and the strategy considering label correlations [38], [42]. Since we focus on the specific purpose of the SPI image datasets, which is to classify multiple defects on SPI images having various sizes, our network used neither the region proposal strategy nor the label correlation strategy when designing the network. Although the multi-label image classification experiments were performed for the image dataset consisting of the same size images, which is different from our purpose, our network showed competitive performance. Moreover, our network outperformed the other methods, for classes like *car*, *dog*, and *tv*. The region proposal strategy is not suitable for our purpose because the features of the defects are presented not only in specific parts of the SPI image, but also in the whole part of the SPI image. We consider adding a label correlation strategy to enhance classification performance for the future work.

VI. CONCLUSION

We proposed MarsNet, a CNN based end-to-end network for multi-label classification which is also able to take inputs of various sizes. In order to make this possible, the mDRN was first applied to preserve the resolution of the feature maps throughout the convolutional layers by allowing a small filter to have a wide receptive field. Then, HVP was newly implemented to extract spatially meaningful information as well as to increase the computational efficiency during the process of pooling. Lastly, the multi-label scoring module and the threshold estimation module were employed for multi-label classification. We conducted experiments with input images of various sizes to verify the effectiveness of our network. The experimental results on the customized SPI dataset as well as VOC 2007 dataset demonstrated excellent performance of the proposed network. Recently, understanding the correlation between labels in multi-classification problems has been recognized as an important research topic. For the future work, we plan to exploit the correlation between the classes present in the image to improve the performance of our work.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2015, pp. 91–99.
[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[7] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[8] X. Luo and A. N. Zincir-Heywood, "Evaluation of two systems on multi-class multi-label document classification," in *Proc. Int. Symp. Methodologies Intell. Syst.* New York, NY, USA: Springer, 2005, pp. 161–169.

[9] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 519–534.

[10] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 75–91.

[11] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[12] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 472–480.

[13] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–13.

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2014, pp. 346–361.

[16] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[17] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification–revisiting neural networks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, 2014, pp. 437–452.

[18] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3617–3625.

[19] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 390–399.

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[21] K. Grauman and T. Darrell, "The pyramid match Kernel: Discriminative classification with sets of image features," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Oct. 2005, pp. 1458–1465.

[22] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2006, pp. 2169–2178.

[23] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[24] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, 2009, pp. 254–269.

[25] G. Tsoumakas and I. Vlahavas, "Random K-labelsets: An ensemble method for multilabel classification," in *Proc. Eur. Conf. Mach. Learn.* Springer, 2007, pp. 406–417.

[26] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.

[27] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Springer, 2001, pp. 42–53.

[28] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2285–2294.

[30] E. Yuk, S. Park, C.-S. Park, and J.-G. Baek, "Feature-learning-based printed circuit board inspection via speeded-up robust features and random forest," *Appl. Sci.*, vol. 8, no. 6, p. 932, Jun. 2018.

[31] G. Acciani, G. Brunetti, and G. Fornarelli, "A multiple neural network system to classify solder joints on integrated circuits," *Int. J. Comput. Intell. Res.*, vol. 2, no. 4, pp. 337–348, 2006.

[32] J. Richter, D. Streitferdt, and E. Rozova, "On the development of intelligent optical inspections," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 1–6.

[33] K. Won Ko and H. Suck Cho, "Solder joints inspection using a neural network and fuzzy rule-based classification method," *IEEE Trans. Electron. Packag. Manuf.*, vol. 23, no. 2, pp. 93–103, Apr. 2000.

[34] Y.-L. Lin, Y.-M. Chiang, and H.-C. Hsu, "Capacitor detection in PCB using YOLO algorithm," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Jun. 2018, pp. 1–4.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[36] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 806–813.

[37] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "CNN: Single-label to multi-label," 2014, *arXiv:1406.5726*. [Online]. Available: https://arxiv.org/abs/1406.5726

[38] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, "Multilabel image classification with regional latent semantic dependencies," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2801–2813, Oct. 2018.

[39] H. Yang, J. T. Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai, "Exploit bounding box annotations for multi-label object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 280–288.

[40] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1901–1907, Sep. 2016.

[41] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 464–472.

[42] T. Chen, Z. Wang, G. Li, and L. Lin, "Recurrent attentional reinforcement learning for multi-label image recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 6730–6737.

[43] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[44] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 38, no. 2, pp. 303–338, Sep. 2009.

JU-YOUN PARK received the B.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2015 and 2019, respectively. She worked as a Postdoctoral Researcher with the Information and Electronics Research Institute, KAIST. She is currently a Postdoctoral Scientist with the School of Engineering and Applied Science (SEAS), George Washington University, Washington, DC, USA. Her current research interests include machine learning, artificial intelligence, and robotics.

YEWON HWANG received the B.S. degree in mechanical engineering from Pennsylvania State University at University Park, in 2018. She is currently pursuing the M.S. degree in electrical engineering with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. Her current research interests include computer vision, machine learning, human–robot interaction.

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 (Fellow, IEEE) 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 Professor. He is currently the Director of the KoYoung-KAIST AI Joint Research Center and the Machine Intelligence and Robotics Multisponsored Research Platform. He has authored five books and five edited books, two journal special issues, and around 400 refereed articles in technical journals and conference proceedings. His research interests include intelligence technology, machine intelligence learning, ubiquitous and genetic robots, and humanoid robots. He served as an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the *IEEE Computational Intelligence Magazine*.