An Incremental Meta Defect Detection System for Printed Circuit Boards

Jia-Jiun Gung, Chia-Yu Lin, Pin-Fan Lin, and Wei-Kuang Chung
Department of Computer Science and Engineering, Yuan Ze University, Taiwan.
Email: sallylin0121@saturn.yzu.edu.tw

2022 IEEE International Conference on Consumer Electronics - Taiwan | 978-1-6654-7050-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCE-TAIWAN55306.2022.9869108

***Abstract*—Defect detection is essential in production lines to guarantee the quality of products. However, detecting tiny defects is difficult. Besides, as the variety of products increases, the variety of defects also increases. Models take much time to retrain. In this paper, we propose an "Incremental Meta Defect Detection (IMDD) System," which utilizes incremental meta-learning to detect tiny defects. We decompose the model into feature pyramids and use feature alignment to improve the sensitivity of minor defects. Incremental learning utilizes knowledge distillation but this affects the learning of new categories, so the model is quickly adapted to new categories. We further combine incremental learning with meta-learning to increase the generality of the model. In experiments, the proposed model is 1.14 times more accurate than previous techniques. Therefore, the proposed system can enhance the ability to identify minor defects and quickly adapt to new defect types.**

I. Introduction

The production line for defective testing requires AI computer visual inspection. The data needs to be labeled before the training of the model. Furthermore, the defect type cannot be discovered and defined simultaneously, and it is necessary to retrain the model. Nevertheless, suppose we add new categories directly to the old model to fine-tune training. In that case, the accuracy of the past training categories may have dropped sharply, which is "catastrophic forgetting." Fortunately, incremental learning can figure out this problem.

When using incremental learning, knowledge distillation or regularization is used to retain previously learned knowledge. However, this can cause the model to fail to adapt fast as it learns new knowledge. To effectively fast adapt to the way, we thank the research of [1] and use a meta-learning method. Allows training to generalize to new classes more generally.

This paper proposes Incremental Meta Defect Detection (IMDD) System," which combined with incremental meta-learning to detect tiny defects. The production line may have defect types that have not been encountered before, so we must introduce incremental learning, characterized by retaining previously learned knowledge and increasing the ability to identify newly discovered defect types. Many studies [1], [2] have shown that meta-learning methods are suitable for solving this problem because meta-learning uses unique gradient descent methods. Even though many meta-learning ways propose different gradient descent methods, the gradient descent calculated by θ_1 and θ_2 is used to update the original gradient θ_0, enabling the model to effectively discriminate previous types and even generalize to New unknown categories. In addition, meta-learning can also have good results in the case of less new data. This coincides with incremental learning.

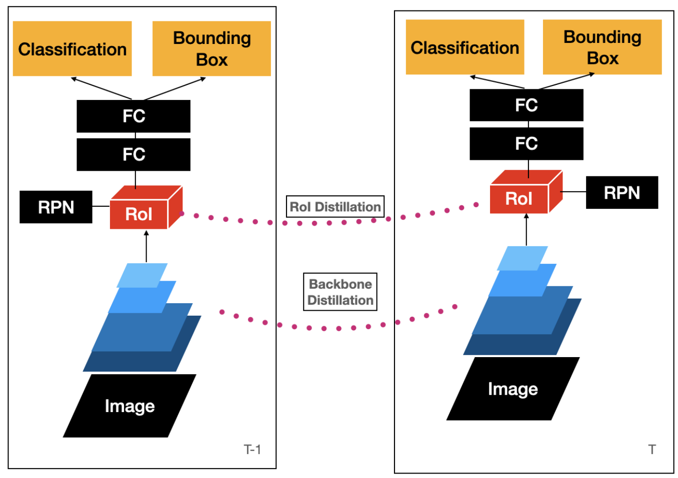

Figure 1. IMDD architecture.

According to [3], [4], [7], we want to find the idea of tiny defects by overlapping bounding boxes during prediction. So that the model can be more sensitive and accurate in detecting slight flaws. In this regard, we must fuse the model at multiple scales and align the extracted feature pyramid network (FPN) to improve the accuracy of tiny flaws. This research is conducted through the data of cooperative manufacturers, and the type of data will not be discussed in detail.

II. IMDD method

As shown in Fig. 1, the following describes the IMDD architecture diagram designed to enhance the detection ability of tiny defects. We refer to [3], [4], [7] to introduce the concept of FPN. After feeding the image into the model. Features are extracted using i-layer convolution and aligned with other scale layers through the FPN. Furthermore, the scale of each level of our pyramid is set to {32, 64, 128, 256, 512} respectively. After the model outputs the features, the region of interest (RoI) and region proposal networks (RPN) are used to predict defects and bounding box categories. When adding new types requires incremental learning, we perform knowledge distillation on the RoI layer and feature of FPN to preserve previous knowledge.

We use Faster R-CNN [6] as the main model. It is used in the model to determine the correctness of the Bounding Box by calculating the RoI loss. Let $\boldsymbol{p}$ = {p0,..., pk} be the probability of K+1 classes (K classes and background). Let b be the predicted bounding box position for each RoI, where $\boldsymbol{b}$ = {b_x, b_y, b_w, b_h}. Furthermore, set the ground truth class probability and bounding box regression targets as $\boldsymbol{p}^*$ and $\boldsymbol{b}^*$ as the following equations (1). Where $\mathcal{L}_{cls}$ and $\mathcal{L}_{bbox}$ are defined follows (2), (3). Smooth$_{\mathcal{L}1}$ is defined as follows (4). The advantage of using the L1 loss is that the value can be stably converged to the ideal value. Not using the L2 loss is that if there are outliers, the L2 loss is more easily affected, resulting in very different values. Furthermore, inspired by [3], we define the distillation loss as follows (5). In the above distillation loss function, we use $\mathcal{L}_2$ loss for $\mathcal{L}_{\mathcal{R}eg}$ and add a λ_n in front of each term to increase the weight for each loss. From [5], it is known that the weight will be adjusted according to the degree of emphasis. That allows our loss function to increase its λ_n for differences in importance. F represents the model backbone. The RPN loss is defined in Faster R-CNN, but we have to detect tiny defects in this work. We refer to the RPN defined in [1]. It can effectively predict the presence or absence of objects in multi-task and regression bounding boxes. The loss of one image is defined as follows (6).

$$\mathcal{L}_{RoI} = \mathcal{L}_{cls}\left(\boldsymbol{p_i}, \boldsymbol{p_i^*}\right) + \lambda\left[\boldsymbol{p_i^*} \geq 1\right] \mathcal{L}_{bbox}\left(\boldsymbol{b_i}, \boldsymbol{b_i^*}\right), \tag{1}$$

$$\mathcal{L}_{cls}\left(\boldsymbol{p}, \boldsymbol{p}^*\right) = -log\boldsymbol{p_i^*}\boldsymbol{p_i} + \left(1 - \boldsymbol{p_i^*}\right)\left(1 - \boldsymbol{p_i}\right), \tag{2}$$

$$\mathcal{L}_{bbox} = \sum_{i \in \{x,y,w,h\}} Smooth_{\mathcal{L}1}\left(\boldsymbol{b_i} - \boldsymbol{b}_i^*\right), \tag{3}$$

$$Smooth_{L1} = \begin{cases} 0.5x^2 & if\ \ |x| < 1 \\ |x| - 0.5 & otherwise, \end{cases} \tag{4}$$

$$\begin{aligned}\mathcal{L}_{Distill} = \lambda_1 \mathcal{L}_{\mathcal{R}eg}\left(F_t, F_{t-1}\right) + \lambda_2 \mathcal{L}_{\mathcal{KL}}\left(p_t, p_{t-1}\right) \\ + \lambda_3 \mathcal{L}_{\mathcal{R}eg}\left(b_t, b_{t-1}\right),\end{aligned} \tag{5}$$

$$\mathcal{L}_{RPN} = \frac{1}{N_{cls}} \sum_i \left(\boldsymbol{p_i}, \boldsymbol{p_i^*}\right) + \lambda \frac{1}{N_{Reg}} \sum_i \boldsymbol{p_i^*} \mathcal{L}_{Reg}\left(\boldsymbol{b_i}, \boldsymbol{b_i^*}\right), \tag{6}$$

$$\mathcal{L}_{task} = \alpha \mathcal{L}_{Distill} + (1 - \alpha)(\mathcal{L}_{RoI} + \mathcal{L}_{RPN}), \tag{7}$$

$$\begin{aligned}\mathcal{L}_{warp} = \sum_{(\boldsymbol{f}, \boldsymbol{p}^*, \boldsymbol{b}^*) \sim F_{store}} \mathcal{L}_{cls}\left(\boldsymbol{p}, \boldsymbol{p}^*\right) + \lambda\left[\boldsymbol{p}^* \geq 1\right] \mathcal{L}_{bbox}\left(\boldsymbol{b}, \boldsymbol{b}^*\right), \\ \text{s.t.}, (\mathbf{p}, \mathbf{b}) = \text{FRoI}\ \ (\mathbf{f})\end{aligned} \tag{8}$$

Where $\mathcal{L}_{Reg}$ is the same as previously defined Smooth$_{L1}$, overall, the loss has to be computed in multiple tasks. Tiny defect detection more easily by aligning feature pyramids. Its structure diagram is shown in Fig. 1. And the definition of $\mathcal{L}_{task}$ is as follows (7). $\mathcal{L}_{warp}$ is defined as follows (8).

The F in the equation represents a queue for storing features and bounding boxes. The primary purpose of $\mathcal{L}_{warp}$ is to balance the number of image category instances and features. That also exists and is necessary on the PCB. θ in Meta-learning has expressed as $\theta = \emptyset \cup \varphi$. Where $\emptyset$ is the parameter of $\mathcal{L}_{task}$, and φ is the parameter of $\mathcal{L}_{warp}$.

Table I
EXPERIMENT RESULT. UNIT:%

Model	***MH***	***MB***	***OC***	***Short***	***Spur***	***SC***	***mAP***
iOD	53.85	47.65	46.13				49.21
	51.64	37.33	36.94	45.18	43.09	47.09	43.54
IMDD	55.76	52.77	49.78				52.77
	53.48	50.62	47.18	50.47	47.54	49.56	***49.81***

III. EXPERIMENT

We trained the model on a Tesla P100-PCIE-16GB and used the [3] augmentation dataset for training. The augmented dataset has six categories: open circuit(OC), short, mouse bit(MB), spurious copper(SC), spur, and missing hole(MH). Its dataset has 10,668 printed circuit board images. The size of the image is 600 x 600. The result of mean average precision(mAP) measured by the previous [1] is 43.5499%. Compared with the former, our experimental result is 49.8138%. The results are shown in Table I. Our results are 1.14 times better than the last state-of-the-art experiments.

IV. CONCLUSION

We proposed an incremental meta defect detection (IMDD) system to detect tiny defects on PCBs. Using incremental learning and tiny defect detection at the same time is challenging. Sigh [1] combines incremental Object detection and meta-learning. This is used on the PCB and combined with FPN to improve performance. Furthermore, according to the current experiments, mAP results are 1.14 times better than the state-of-the-art model. This is because FPN is more sensitive to detecting tiny defects than general models.

This article mainly applies incremental meta-learning to tiny defect detection on printed circuit boards. Because the previous work primarily uses incremental meta-learning to train PASCAL-VOC and MS COCO, less research is used on PCB. However, during the implementation process, it was found that the model's ability to train for tiny defect detection is limited. We refer to the [3], [7] to solve this problem, which can significantly improve the detection ability.

REFERENCES

[1] Kj, Joseph, et al. "Incremental Object Detection via Meta-Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[2] Rajasegaran, Jathushan, et al. "itaml: An incremental task-agnostic meta-learning approach," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[3] Ding, Runwei, et al. "TDD-net: a tiny defect detection network for printed circuit boards," *CAAI Transactions on Intelligence Technology 4.2*(2019): 110-116.

[4] Shi, Wei, et al. "Single-shot detector with enriched semantics for PCB tiny defect detection," *The Journal of Engineering* 2020.13 (2020): 366-372.

[5] Akcay, Samet, Amir Atapour-Abarghouei, and Toby P. Breckon. "Ganomaly: Semi-supervised anomaly detection via adversarial training," *Asian conference on computer vision*. Springer, Cham, 2018.

[6] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems* 28 (2015).

[7] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017.