ReProInspect: Framework for Reproducible Defect Datasets for Improved AOI of PCBAs

Ahmad Rezaei[1[0000-0002-9232-6934]], Johannes Nau[1[0000-0001-7538-2283]], Detlef Streitferdt[1[0000-0003-0878-0614]], Jörg Schambach[2[0009-0003-6398-4860]], and Todor Vangelov[1]

[1] Technische Universität Ilmenau, Helmholtzplatz 5, 98693 Ilmenau, Germany
{ahmad.rezaei, johannes.nau, detlef.streitferdt todor.vangelov}@tu-ilmenau.de
[2] GÖPEL electronic GmbH, Jena, Germany j.schambach@goepel.com

Abstract. Today, the process of producing a printed circuit board assembly (PCBA) is growing rapidly, and this process requires cutting-edge debugging and testing of the boards. The Automatic Optical Inspection (AOI) process detects defects in the boards, components, or solder pads using image processing and machine learning (ML) algorithms. Although state-of-the-art approaches for identifying defects are well developed, due to three main issues, the ML algorithms and datasets are incapable of fully integrating into industrial plants. These issues are privacy limitations for sharing data, the distribution shifts in the PCBA industry, and the absence of a degree of freedom for reproducible and modifiable synthetic datasets.
This paper addresses these challenges and introduces "ReProInspect", a comprehensive framework designed to meet these requirements. ReProInspect uses fabrication files from the designed PCBs in the manufacturing line to automatically generate 3D models of the PCBAs. By incorporating various techniques, the framework introduces controlled defects into the PCBA, thereby creating reproducible and differentiable defect datasets. The quality data produced by this framework enables an improved detection and classification scenario for AOI in industrial applications. The initial results of ReProInspect are demonstrated and discussed through detailed instances. Finally, the paper also highlights future work to improve the current state of the framework.

Keywords: Automated Optical Inspection · Machine Learning · 3D Rendering

1 Introduction

The global Printed Circuit Board (PCB) market size is currently growing from 72 billion USD in 2022 to an estimated 89 billion by 2028.[3] This rapid growth in

[3] MarketWatch, The Prospects of Printed Circuit Board (PCB) Market 2023: Industry Trends and Challenges till 2030

Table 1: PCBA defect categories covered by AOI systems.

Category	Area	Short explanation
Incorrect placement (IP)	Component	The component is positioned in the false angle or place. The connection with the solder pad can be weak or lost.
Missing component (MC)	Component	The component is not present in its specified place.
Tombstone (T)	Component	Due to solder heat setting or placement the component loses solder connection on one side.
Textual failure (TF)	Component	The text on the component refers to a false component or is distracted.
Extra or insufficient solder (EIS)	Solder pad	The amount of solder is too low or too much.
Not soldered (NS)	Solder pad	The soldering paste is removed before melting.
Short circuit (SC)	Solder pad and Board	An undesirable electrical connection.
Missing solder pad (MSP)	Board	Incorrect manufacturing of PCB solder pads.
Open circuit (OC)	Board	The loss of connection between copper lines, which should be connected.
Spurious or mouse bite on copper (SMC)	Board	damaged copper lines without loss of electrical connection.
Pseudo defects (PD)	All areas	Environmental particles are observed.

PCB production requires advancements for fast, reliable, and cost-effective production chains. A major challenge in production chains is the detection, repair, or elimination of defective components or boards.

AOI is a process for detecting defective PCBA boards (listed in Table 1) by processing the industrial camera images taken from the boards in the production line. AOI contains three factors; the camera and light setup, the image processing algorithm used for better highlighting the defects, and the machine learning (ML) techniques for decision-making. These factors are constantly studied and improved in the literature. For instance, defective areas and components are not observable without having the proper light setting and camera setup. Shadows or reflections of other components or solder pads should be tackled for a clear view of the defects. Authors in [3], [8], and [9] cover these issues regarding capture angle and light settings, and they propose various settings for AOI systems. Moreover, the image processing techniques are mathematically well developed and intertwined with the need for ML techniques used for decision-making. For instance, the large industrial images should be re-scaled and divided into several smaller sets for faster decision-making [12], [17]. Subsequently, ML techniques for the detection and classification of PCBA defects are constantly improving by using cutting-edge object detectors and Convolutional Neural Networks (CNNs); e.g. improved versions of the "You Only Look Once" (YOLO) algorithm for the detection of defects in PCBA surface [2], [12] or CNN models used for component defect classification [8].

Although the state-of-the-art (SOTA) shows great improvement for AOI systems, three challenging factors (discussed in details in section 2) in PCBA AOI datasets such as privacy-criticality, quality of synthetic data, and degree of freedom persist. Former datasets include a limited number of defect categories (due to privacy issue) without considering the distribution shifts present in the industry, and moreover, they produce synthetic or partly real data with constant light and camera setting for all images, which does not adhere to industry preferences and is not practical for detection of all defect types. In this paper, these factors are discussed and tackled with the proposal of a framework for reproducible

PCBA AOI defect dataset. The framework adheres to industrial privacy constraints, and gives the users a degree of freedom in producing their own PCBA AOI datasets from 3D PCBA models with a degree of freedom in environmental settings, distribution shifts, and the quality of captured data.

The following sections in this paper are organized as follows. Section 2 elaborates on PCBA defects and covered defects in the former works. Next, the technical setup for the framework to produce synthetic data is covered in subsection 2.2. The section 3 presents our proposed framework and describes its architecture. Using this architecture, visual results for the captured data by the framework are discussed in section 4.

2 Related Works

PCBA defects are abnormalities in the manufacturing process, which can lead to malfunction or complete breakdown of a PCBA. These defects in PCBA happen in three main areas such as electrical components, soldering pads, and board surface. More than 100 defects are categorized in National Physical Laboratory Industry Defects Database[4], which are structured into eleven categories (shown in Table 1) classified based on AOI system detection (categorized based on publications in [17] and [14]).

A summary on former datasets is provided in Table 2, which is used for the identification of research gaps discussed in sub-section 2.1. The literature consists of datasets for PCBA AOI for defect detection or normal board inspection [10], [13], and [7]. The latter is out of scope for this paper; however, the ReProInspect framework can help enhance the datasets in the normal board inspections. In 2019, DeepPCB [18] and synthesized PCB [6] datasets focused on five categories of defects on board or solder pads. The use case for these datasets is for the defect detection with object detectors, but they lack other common defects. Another issue in these datasets is the quality of synthetic data (manual insertion of defects) and the quality of captured data which does not conform to industrial settings. Moreover, in 2020, the authors in [10] used a nonpublic dataset, including real PCBA images captured in an industrial setting, that includes more defects focusing on solder pads. Furthermore, in 2021, CD-PCB [4] was published with 20 image pairs consisting of synthetic board and solder pad defects. In this dataset the manual synthesized data and lacking industrial capturing environment are persistent. Authors in [21] published PCBNet dataset (2022) that conversely to previous datasets focuses more on component defects. This dataset contains real images taken in an industrial setting, but it lacks other categories of defects. Subsequently, in 2023, the HU-Solder [20], capturing some of the defect categories on components and solder pads, is released. This dataset contains real images taken in a manual setting using a 13-megapixel camera. Finally, in the same year, a non-public dataset is used in [1] that includes real data taken under an industrial setting but with only a focus on some solder pad defects.

[4] Available (last seen on 18.08.2023): `http://defectsdatabase.npl.co.uk/`

Table 2: Summary on the datasets used in previous literature.

	Dataset Name	DeepPCB [18]	Synthesized PCB [6]	Advantech Data [11]	CD-PCB [4]	PCBNet Data [21]	HU-Solder [20]	Vision Automobile [1]
	Available	✓	✓	✗	✓	✓	✓	✗
	Industrial Capture	✓	✗	✓	✗	✓	✗	✓
	Real Data	✗	✗	✓	✗	✓	✓	✓
Covered Defects	**IP**	✗	✗	✗	✗	✓	✓	✗
	MC	✗	✗	✗	✗	✓	✗	✗
	T	✗	✗	✗	✗	✓	✓	✗
	TF	✗	✗	✗	✗	✓	✗	✗
	EIS	✗	✗	✓	✗	✗	✓	✓
	NS	✗	✗	✓	✓	✗	✗	✗
	SC	✓	✓	✓	✓	✗	✓	✓
	MSP	✓	✓	✗	✓	✗	✗	✗
	OC	✓	✓	✗	✓	✗	✗	✗
	SMC	✓	✓	✗	✓	✗	✗	✗
	PD	✓	✓	✓	✓	✗	✗	✗
	#Defect images	1,500	1,386	834	20	-	655*	600*

(*=snapshot images, -=no information given)

2.1 A Recap on The Privacy Issue

Privacy is the main culprit for the lack of unified and well-developed datasets in PCBA AOI domain. However, the privacy of companies' intellectual properties contributes to the reliability of the products, the companies' competition in open market, and resilience to security threats. In other domains, the authors try to leverage privacy by redefining privacy-preserving data exchange [15], but due to lack of viable solution for PCBA AOI, this paper proposes a solution adhering to the privacy constraints in the industry.

The quantity of produced images for PCBA defect datasets in [18] and [6] shows the great potential to create synthetic data from fewer actual PCBAs. However, the production of the synthetic data in former approaches is done manually and in an uncontrolled way. First, they include a subset of defect categories, which can be extended for a more realistic dataset. Second, the defects are manually inserted to a 2D image without adhering to their actual rationale (heat, physical fluid dynamics, and malfunction of pick-and-place arms) behind the formation of defects. Thus, larger quantity of higher quality data is producible with solving these two issues. Furthermore, in industrial AOI scenarios, the capturing angle and the light settings for images are adjustable. Henceforth, it is valuable to allow the researchers having the same degree of freedom while producing their synthetic dataset.

Subsequently, in realistic AOI scenarios, distribution shifts are available [19]. So the test data (PCBA images taken from the production line) can be different from training data (already-made defect dataset), which can degrade the performance of ML algorithms severely. In the PCBA AOI, the source of distribution shift is basically the difference in color of boards or in the components used during the assembly. The distribution shifts affect the defect detection greatly, although the PCBA possesses the same functionality. Creation of synthetic de-

Table 3: Ordered factors for selection of rendering software (highest priority first)

Priority	Factor	Description
1.	Complexity Handling	Managing complex scenes efficiently.
2.	Speed	Rendering speed impacts dataset production timeline.
3.	GPU Rendering	Utilizing GPUs for fast rendering.
4.	Ease of Use	User-friendly interface for a smooth workflow.
5.	Cost-effectiveness	Balancing capabilities with costs.
6.	Render Farm Support	Integration with distributed rendering.
7.	Integration	Seamless collaboration with modeling tools.
8.	Light Control	Precise control over lighting settings/scenarios.
9.	Quality Control	Fine-grained adjustments for visual fidelity.
10.	Supported Platforms	Compatibility with multiple operating systems.
11.	Community Support	Active user community and resources.
12.	Updates and Development	Regular software improvements.

fects with distribution shifts for improving the AOI systems or evaluating the robustness of proposed AOI systems is beneficial.

In conclusion, the ReProInspect framework enables the users to reproduce a controlled defect dataset with a degree of freedom in modifying the image capture settings, distribution shifts, and adhering to industrial settings. The following sub-section explains the technical setup for generation of 3D defect data.

2.2 Technical Requirements and Setup

An important part of the ReProInspect framework is the rendering software which creates the PCBA 3D model from the fabrication data. In Table 3, twelve selection factors for producing sheer volume of high quality data are prioritized based on the authors' experiences and research.

After applying the first factor to the search for different rendering software packages, results were limited to 8 rendering softwares with similar characteristics for factors 6 to 12. These software packages are shown in Table 4 retaining high integration capacity, support for rendering farms, flexible control on light and quality, various OS support, and active maintenance and community support. Thus, Table 4 only lists the comparison factors of the subset between 2 to 5. Based on a comparison in this Table, Blender and LuxCoreRender allow a user-friendly, high speed, and open-source experience with GPU compatibility.

The 3D computer-aided design (CAD) data for components (in OBD++, DXF, and Blend formats) are usually accessible via open source libraries such as Kicad software package [5]. These 3D data enable manual user modifications, which are automated in ReProInspect framework with production of components with irregular shapes (due to manufacturing problems), broken parts, different colors or textures, and also, with simulated dust and other environmental particles as pseudo defects.

Furthermore, the fabrication data (Gerber files, PCB NC files, and Pick and placement files) are required for production and assembly of PCB and CAD components in the ReProInspect. Next section proposes the architecture of the ReProInspect based on the explained technical setup and requirements.

[5] Available: `https://www.kicad.org/`

Table 4: Rendering software comparison for PCBA defect dataset production

Software	Speed	GPU Rendering	Ease of Use	Cost-effectiveness
Blender	Fast	Yes	User-friendly	Free
Autodesk Maya	Moderate	Yes	Moderate	Costly
Maxon Cinema 4D	Moderate	Partial	User-friendly	Costly
Chaos Group V-Ray	Fast	Yes	Moderate	Costly
Redshift	Fast	Yes	Less User-friendly	Reasonably Priced
LuxCoreRender	Fast	Yes	User-friendly	Free
Octane Render	Fast	Yes	Moderate	Reasonably Priced
Arnold	Fast	Yes	Less User-friendly	Costly

3 Proposed Framework

The architecture is based on an enhanced tool chain as described in [14]. The core architectural style of the tool chain is Pipes & Filters, to support several operations on each PCBA-element. As modelled in Fig. 1, the filters operate on a central repository. Thus, the original Pipes of the Pipes & Filters style had to be adapted, in favor of a data model supporting the later introduction of defects for the ML training, the ultimate goal of this paper.

As seen in Fig. 1, the input for the tool-chain are PCBA fabrication data (e.g. from Kicad) and the 3D component files for each of the PCBA components. In the diagram, the repository holds all the PCBA geometric layer data combined with the 3D component data items, which is essential for production of complete PCBA 3D model in the rendering step. Next, the Renderer is an independent component for final image production, and the tool-chain's filters are responsible for the introduction of defects (listed in Table 1) into the dataset. Eventually, the resulting and rendered image can contain correct or erroneous images of PCBA.

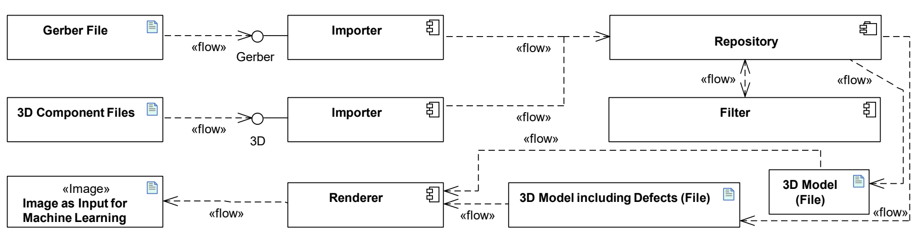


Fig. 1: Software Architecture, Tool Chain

Based on the detailed architecture of the repository (Fig. 2), each of the filters adhere to this software architecture. The filter has access to the complete data model of the PCBA and its components, which is modeled as a list of all *virtual* components. This list will be read and serialized back to the repository, e.g. a database.

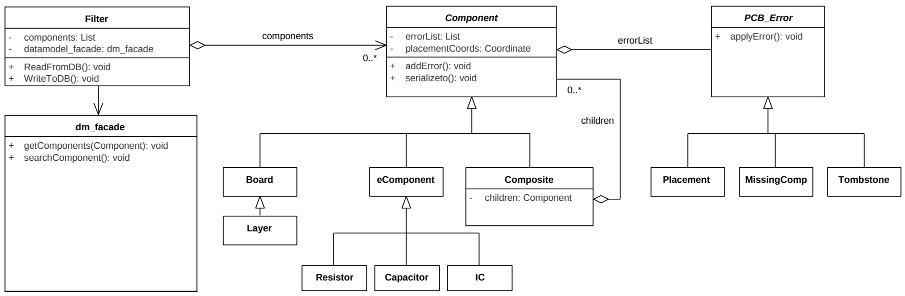


Fig. 2: Software Architecture, Data Model

The *virtual* components are the core part of the composite design pattern [5] for PCBA and its components, and it reflects the PCBA system structure and allows the introduction of defects at any layer or component of the structure. The *Board*, a leaf of the composite pattern, with its layers is the central building block of an electronic system, and the *eComponent* can then be placed on the board and have types like Resistor, Capacitor, or Integrated Circuit (IC). Based on this architecture a list of possible defects is held with any level of granularity, and these defects can be implanted sequentially as it results in different rendered output image (e.g. sequential order of a broken colored resistor or a colored broken resistor).

The main emphasis was put on the extensibility of the software architecture. The composite pattern allows to add any number of components for and layers of the PCBA. In addition, it would also be possible to derive classes from composite and introduce special composites, to allow a further grouping of components. The same argument holds for the PCBA defects which are also extensible. The method *applyError()* will be implemented in any new defect class. Thus, this method works on a component level and can change the components parameters (e.g. change the material, reflectiveness of the surface, or the written type on the component) or change its geometric form (e. g. to simulate mechanical damages or the simply move or misplace the component).

Although each filter will have to implement the architecture of Fig. 2, it will be specifically developed for a given defect use case/scenario. This allows to use the software framework within a command line scripting environment. Finally, we create the desired and fully labeled pictures for the following ML training. The scripts hold all the necessary information for the generated picture set and thus, will be stored in a versioning system.

4 Results and discussion

Produced images with *ReProInspect* framework underline the capabilities discussed in the last sections. In Fig. 3 an example PCBA[6] with black theme and

[6] Available (last seen on 18.08.2023):`https://github.com/dmitrystu/Nucleo2USB`

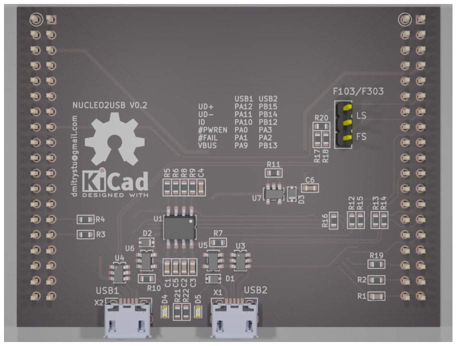


Fig. 3: Correct PCBA Image

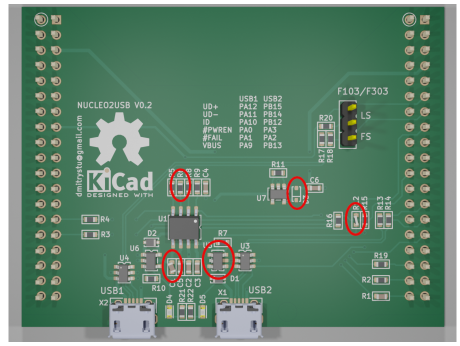
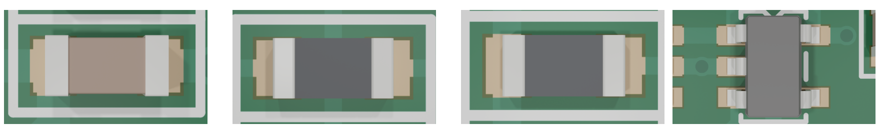
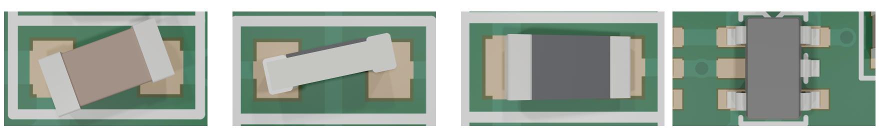


Fig. 4: PCBA Image With Defects

without any defects is shown, whereas in Fig. 4, the same PCBA (green theme) has defects. Of course, in these views it is hard for humans to identify the defects on the board. Thus, an AOI system inspects components for the defect detection. Five defects are "implanted" on Fig. 4. Detailed pictures of four of them, generated with ReProInspect, are shown in Fig. 5.



(a) C1 capacitor (b) R12 resistor (c) R6 resistor (d) U5 IC



(e) Rotated C1 (f) Billboard R12 (g) Tombestone R6 (h) Misplaced U5

Fig. 5: Snapshots taken from normal and defective components.

First, a rotated capacitor in Fig. 5e with its correct positioning in Fig. 5a is shown. This defect might happen when the airflow in the soldering oven moves components with the liquefied soldering paste. Second, the resistor R12, correctly placed in Fig. 5b, has a billboard defect in Fig. 5f. This defect might be due to the placement of resistor upwards instead of its flat side. These two defects may result in fully functioning PCBA with probability of malfunctions, which is the reason for manufacturers to sort out such boards as *not ok*. Next defect is yet another resistor R6, placed correctly in Fig. 5c. The tombstone defect in Fig. 5g is hard to detect for humans from this above the component viewpoint. Tombstones are formed by a time delay of the liquefaction of the two solder paste sides. The right side liquefied earlier and its surface tension lifted the component. Fourth defect is a 180 degrees misplaced IC as in Fig. 5h (correct placement as in Fig. 5d). The last two defects cause malfunctioning PCBA in the end.

All in all the ReProInspect framework has been successfully set up a tool chain. It is capable of processing the original PCBA data as input and can generate the desired pictures as output and input for the following ML training. Its generality and scalability are given by the scalable architecture (any defects can be added). Across Operating systems only Python and the availability of the rendering engine (currently: Blender) are the requirements.

5 Future works

Despite the current working state of the framework and its core software architecture, several of the architectural features have to be further enhanced.

As shown in the figures above, the soldering points, although available [14], need to be integrated into the picture generation process. Due to high computation and rendering time , the parallelization (discussed in section 3) together with soldering computation could be further improved. Currently we experienced a speedup of 35 times from a four core CPU to a CPU + GPU setup.

The validation of PCBA dataset generated by ReProInspect will be done through a comparison with real industrial data. The important difference here is the validation on the premises of the picture owner. Thus, the actual data stays on owner's site and the privacy is ensured. The same applies for enhancement of AOI systems with distribution shifts so the ReProInspect can be used on sites for further enhancing and refining approaches (e.g. in [16]), again on the premises The same argumentation is taken for the defect in PCBAs which are different amongst PCBA production sites. Here, the defect classification as started in [16] can be further enhanced and refined, again on the premises of AOI systems.

Acknowledgments: Thüringer Aufbaubank (TAB, 2021 FE 9036) provided financial support for this study.

References

1. Chen, M.C., Chang, W.J., Chen, Z.H., Hsu, C.H., Chiu, Y.C., Su, J.P., Cheng, T.S.: A pcba solder joint defects inspection system based on deep learning technology. In: 2023 IEEE International Conference on Consumer Electronics (ICCE). pp. 1–3. IEEE (2023)
2. Du, B., Wan, F., Lei, G., Xu, L., Xu, C., Xiong, Y.: Yolo-mbbi: Pcb surface defect detection method based on enhanced yolov5. Electronics **12**(13), 2821 (2023)
3. Du, Y., Chen, J., Zhou, H., Yang, X., Wang, Z., Zhang, J., Shi, Y., Chen, X., Zheng, X.: An automated optical inspection (aoi) platform for three-dimensional (3d) defects detection on glass micro-optical components (gmoc). Optics Communications p. 129736 (2023)
4. Fridman, Y., Rusanovsky, M., Oren, G.: Changechip: A reference-based unsupervised change detection for pcb defect detection. In: 2021 IEEE Physical Assurance and Inspection of Electronics (PAINE). pp. 1–8. IEEE (2021)

5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns: elements of reusable object-oriented software. Pearson Education (1994)
6. Huang, W., Wei, P.: A pcb dataset for defects detection and classification. arXiv preprint arXiv:1901.08204 (2019)
7. Jessurun, N., Dizon-Paradis, O.P., Harrison, J., Ghosh, S., Tehranipoor, M.M., Woodard, D.L., Asadizanjani, N.: Fpic: A novel semantic dataset for optical pcb assurance. ACM Journal on Emerging Technologies in Computing Systems **19**(2), 1–21 (2023)
8. Kim, Y.G., Park, T.H.: Smt assembly inspection using dual-stream convolutional networks and two solder regions. Applied Sciences **10**(13), 4598 (2020)
9. Lavrik, E., Panasenko, I., Schmidt, H.R.: Advanced methods for the optical quality assurance of silicon sensors. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **922**, 336–344 (2019)
10. Li, J., Gu, J., Huang, Z., Wen, J.: Application research of improved yolo v3 algorithm in pcb electronic component detection. Applied Sciences **9**(18), 3750 (2019)
11. Li, Y.T., Kuo, P., Guo, J.I.: Automatic industry pcb board dip process defect detection system based on deep ensemble self-adaption method. IEEE Transactions on Components, Packaging and Manufacturing Technology **11**(2), 312–323 (2020)
12. Liao, X., Lv, S., Li, D., Luo, Y., Zhu, Z., Jiang, C.: Yolov4-mn3 for pcb surface defect detection. Applied Sciences **11**(24), 11701 (2021)
13. Lu, H., Mehta, D., Paradis, O., Asadizanjani, N., Tehranipoor, M., Woodard, D.L.: Fics-pcb: A multi-modal image dataset for automated printed circuit board visual inspection. Cryptology ePrint Archive (2020)
14. Nau, J., Richter, J., Streitferdt, D., Kirchhoff, M.: Simulating the printed circuit board assembly process for image generation. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference. pp. 245–254. IEEE (2020)
15. Pennekamp, J., Buchholz, E., Lockner, Y., Dahlmanns, M., Xi, T., Fey, M., Brecher, C., Hopmann, C., Wehrle, K.: Privacy-preserving production process parameter exchange. In: Annual Computer Security Applications Conference. pp. 510–525 (2020)
16. Richter, J., Nau, J., Kirchhoff, M., Streitferdt, D.: Koi: An architecture and framework for industrial and academic machine learning applications. In: International Conference on Modelling and Development of Intelligent Systems. pp. 113–128. Springer (2020)
17. Shi, W., Lu, Z., Wu, W., Liu, H.: Single-shot detector with enriched semantics for pcb tiny defect detection. The Journal of Engineering **2020**(13), 366–372 (2020)
18. Tang, S., He, F., Huang, X., Yang, J.: Online pcb defect detector on a new pcb defect dataset. arXiv preprint arXiv:1902.06197 (2019)
19. Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., Schmidt, L.: Measuring robustness to natural distribution shifts in image classification. Advances in Neural Information Processing Systems **33**, 18583–18599 (2020)
20. Ulger, F., Yuksel, S.E., Yilmaz, A., Gokcen, D.: Solder joint inspection on printed circuit boards: A survey and a dataset. IEEE Transactions on Instrumentation and Measurement (2023)
21. Wu, H., Lei, R., Peng, Y.: Pcbnet: A lightweight convolutional neural network for defect inspection in surface mount technology. IEEE Transactions on Instrumentation and Measurement **71**, 1–14 (2022)