Automated Optical Inspection for Printed Circuit Board Assembly Manufacturing with Transfer Learning and Synthetic Data Generation

Syed Saad Saif[1,2], Kerem Aras[2,3], Alessandro Giuseppi[1,*]

***Abstract*— Automated Optical Inspection (AOI) is among the most common and effective quality checks employed in production lines. This paper details the design of a Deep Learning solution that was developed for addressing a specific quality control in a Printed Circuit Board Assembly (PCBA) manufacturing process. The developed Deep Neural Network exploits transfer learning and a synthetic data generation process to be trained even if the quantity of the data samples available is low. The overall AOI system was designed to be deployed on low-cost hardware with limited computing capabilities to ease its deployment in industrial settings.**

***Index Terms*— Automated Optical Inspection; AOI; Quality Control; Industrial Assembly Lines; Deep Learning; Neural Networks**

I. INTRODUCTION

Automated Optical Inspection (AOI) is an automatic process that consists in a no-contact, visual-based, quality inspection of the output of product lines. AOI has become a fundamental part of manufacturing, seeing applications in several industries such as aerospace, [1], machining [2], [3], recycling [4], food [5] and printing [6].

The most common use case for AOI is related to quality assurance in the electronics industry [7]–[9], for which specialized machines have been developed and are broadly integrated into production lines. The typical quality control delegated to AOI is related to checking the soldering quality on a circuit board and its components. In fact, it is not uncommon in Printed Circuit Board Assembly (PCBA) manufacturing to have misplaced/absent elements and their positioning, along with their alignment, plays a fundamental role in quality controls.

Despite the numerous advantages that integrated machines offer, their pricing and the complexity of their tuning may represent a barrier for their application in low-volume product lines, where specialized/custom circuits are produced.

In this direction, the present work presents a Deep Neural Network (DNN) solution for AOI, specifically tailored to an industrial case study under consideration at the company Token Financial Technologies. related to its PCBA quality controls. The main contributions of this work are:

- The design of a DNN for AOI on a PCBA case study involving the correct placement of a specific microchip on a real production line.
- The design of a data generation procedure with which it is possible to train a DNN tailored for a specific quality-assurance control, minimizing the amount of real data to be collected and enabling the application of the proposed framework on low-volume production lines.
- The proposed DNN-based AOI solution was designed in order to provide a low-cost solution that can be applied to compensate for the looser quality checks that are commonly employed for production lines with low returns.

The rest of the paper is organized as follows: Section II discusses some related works. Section III presents the specific task under consideration, while Section IV presents the proposed Deep Learning system for AOI. In particular, Section IV-A details the dataset creation process; IV-B develops the custom loss function used for the DNN training; section IV-C discusses the DNN architecture. Finally, Section V reports the results of our tests and Section VI draws the conclusions and discusses some future works.

II. RELATED WORKS

The interest and first developments in AOI solutions date back to the 90s [10], but the technology behind them has constantly evolved over the years thanks to the numerous advancements in computer vision [7], [11], inevitably leading to solutions that rely on Deep Learning (DL) [12], [13].

This study focuses on an AOI application designed for PCBA manufacturing. In this application, and in general in the electronics domain, the main advantages offered by AOI systems are [10], [14]:

- The possibility of optimizing the product design by removing for-testing elements, as AOI is a contact-free non-destructive process.
- The reduction of human checks of several orders of magnitude, as operators will be asked to check only a small portion of the PCBA components/solderings.
- A significant increase in the inspection, and consequently in the quality check, consistency and accuracy.
- A significant improvement in the quality control procedure, leading to more efficient product lines with increased throughput.
- The reduction of defective units that leave the production line, that lowers waste, costs and in general pollution.

One of the main drawbacks of data-driven solutions, such as DNNs, is the fact that to increase their performance

This study has been partially supported by the project "Avvio alla ricerca 2021" of the University of Rome "La Sapienza"

[1] Dept. of Computer, Control and Management Engineering (DIAG) of the University of Rome "La Sapienza", Via Ariosto 25, 00185, Roma, Italy
[2] Token Financial Technologies, Istanbul, Turkey
[3] Technical University of Berlin, Berlin, Germany
[*]Corresponding author: giuseppi@diag.uniroma1.it

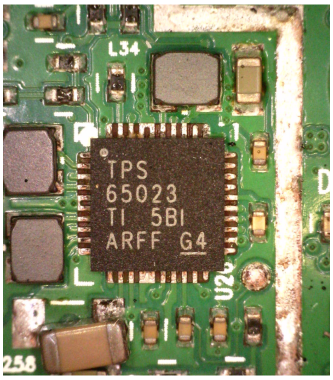

Fig. 1: PCBA example.

it is required to conduct extensive, and expensive, data collection campaigns that may also face difficulties related to the Intellectual Property (IP) of the circuit/product under analysis.

To address this issue, both Transfer Learning (TL) [15]–[18] and synthetic data generation [4], [19], [20] processes have been studied. TL [21] is a popular DL technique that aims at solving a specialized task for which only a small dataset is available by exploiting the knowledge acquired by solving a generic, yet similar, task (e.g., image classification on a large dataset such as ImageNet [22]). This "knowledge transfer" is implemented by deploying a neural network trained to solve the generic task and then fine-tune a few of its layers by re-training their parameters on the available dataset. TL has shown significant performance in many computer vision tasks [23]–[25], but its requirement regarding the availability of both a large dataset related to a task similar to the one under consideration and a smaller, yet still representative, dataset for the fine-tuning process may limit its application in heavily specialized applications such as PCBA AOI.

To compensate for these limitations and enable the study of complex specialized tasks, several studies proposed synthetic data generation approaches as done by the authors in the non-linear spectroscopy domain [26] and by Yue et al. for autonomous driving [27]. In order to generate synthetic data suitable to improve/enable the training of DNNs capable of generalizing also on real data, it is necessary to have extensive knowledge on the considered tasks and use it to develop a suitable data generation model. Such a model shall produce data samples that resemble closely real data, while also incorporating a large spectrum of typical noises/defects/characteristics that may be present in the real task setting.

The present work proposes a DNN-based system that relies on both TL and data generation to cope with the defects that more commonly affect the PCBA designed at Token Financial Technologies. The proposed solution has been designed to be easily tunable, as it can be adapted to similar displacement issues in PCBA manufacturing lines for arbitrary components. The architecture of the DNN was chosen to be not computationally demanding, so that once trained it may be deployed on low-cost/edge computing hardware without requiring connectivity to a cloud server.

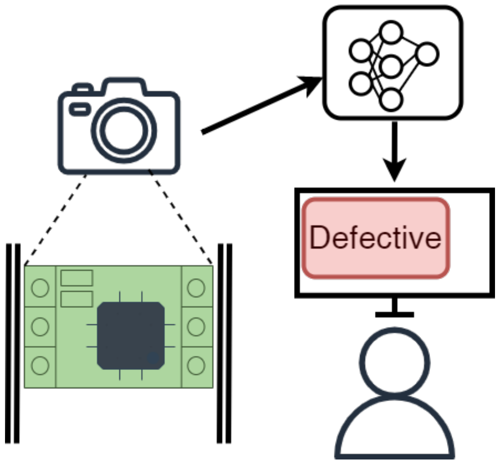

Fig. 2: Scheme for the DNN-based AOI system.

III. Task Description

It is common for low-volume and low-throughput PCBA orders to be produced on multiple production lines originally dedicated to higher-value orders during their idling/maintenance times, as this allows the manufacturer to fulfill lower-value orders without reserving any machinery or stations. This setup allows for a significant production cost reduction, but, at the same time, it causes quality checks to be significantly more complex. In fact, AOI machines, and in general most quality controls, should be entirely reconfigured to the new production, partially neglecting the advantages of employing non-reserved production lines. It is hence common for a manufacturer to deploy fewer and simpler quality control measures for orders of this kind, leading to higher defect rates.

This study proposes a low-cost and easily deployable solution that may be implemented to compensate for the unavailability of advanced AOI systems. In particular, the case study under analysis was identified starting from an analysis of the most common defect reported in a real PCBA line, that is related to the misplacement of the microchip reported in Figure 1 which depicts a defective PCBA where some capacitors were not properly soldered. The objective of the AOI system under development is then to detect any misalignment in the positioning of this specific component, as its defects are the most impactful on the overall PCBA production. We mention that the design process behind this study may be seamlessly adapted to arbitrary components, so that multiple checks may be deployed in parallel to reduce further the shipping of defective units.

IV. Deep Learning-based AOI system

This section will detail the DNN-based AOI system design. As in most AOI machines, the system under consideration envisages the placement of the PCBA under analysis in a controlled area (e.g., an enclosed space), where the PCBA is placed and lighted in a consistent way. An image acquisition system then takes a photo of the area of interest that is then processed by the proposed DNN-based system. The result of the analysis can then be forwarded to operators or automated systems for the disposal of defective units. A simplified block architecture of the setup is reported in Figure 2.

A. Synthetic Data Generation

As already mentioned, the major shortcoming when developing an ad hoc solution based on deep learning is the collection of a suitably large amount of high-quality data. The data collected, in fact, has to be representative of the specific problem under consideration, meaning that it may be required to design ad hoc data gathering campaigns and testings.

For the considered problem, we should provide the learning system with a large variety of examples of both defective and properly assembled PCBA, focusing specifically on the microchip under analysis. By designing a station with controlled illumination and PCBA alignment, as envisaged in Figure 2, we may limit the impact of environmental factors on our data so that our system may focus directly on the microchip positioning.

In order to limit the quantity of PCBA that would need to be manually examined, photographed and labeled, we resort to generating synthetic data that encodes the information sought by our DNN. In particular, starting from a few photographs of properly assembled microchips, it is trivial to create thousands of realistic images by slightly shifting and/or rotating the considered chip after some basic image masking. Some examples of fake defective PCBA generated with python are shown in Figure 3. Despite the fact that these images can be easily identified as edited, as it will be discussed they can still be used to train a DNN with an objective specialized for the AOI task.

An additional advantage of synthetic data generation is that the data can be automatically labeled. As it will be detailed in the following subsection, we reduced the AOI task to identifying the position of the edges of the microchip of Figure 1. Being the synthetic images produced by rotating and/or moving the chip, the required information on its edges can be trivially provided to the DNN for its training, significantly reducing the labour and costs dedicated to the data preparation process.

In our testing, starting from 10 different 256×256 PCBA images we generated 2000 synthetic ones for our training dataset.

B. Proposed Custom Loss function and Performance Metric

When developing any learning system, the choice of an appropriate loss function is among the most critical aspects, as the loss function defines the training objective and shall model the main aspects of interest to solve the task considered.

Being the main defect observed in the PCBAs under study related to the correct positioning of a specific chip, we decided to design the AOI system so that it may precisely locate its four edges. Supposing that correctly estimates

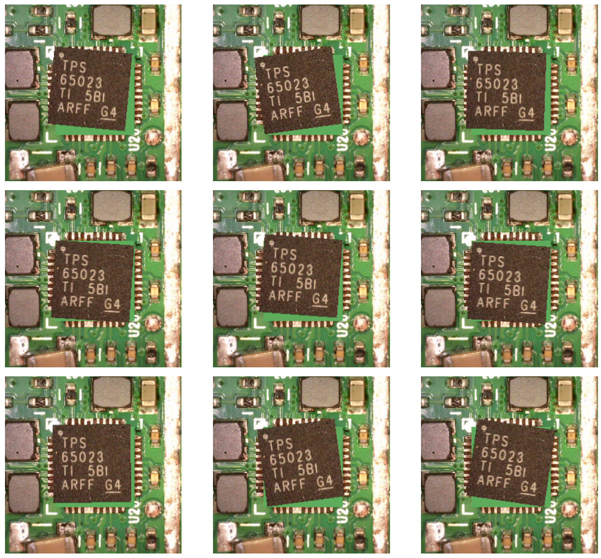

Fig. 3: Synthetic images generated from the original photo reported in Figure 1.

the positions of the edges, it is then possible to determine whether the alignment and placement of the chip are within some appropriate thresholds.

A first solution could be to implement a DNN that, given an image of a PCBA under analysis, estimates the position, in terms of x and y pixel coordinates, of the edges of the considered chip. In this setting, the output of the DNN would take the form of 4 (x, y) pairs, corresponding to the points that are visualized in Figure 5, and the main objective of the DNN would simply be to minimize the squared error between the predicted edge positions and their actual value.

Considering that it was observed that the typical PCBA defects are related to the positioning/alignment of the chip rather than deformations of its shape, we decided to exploit this information to improve the DNN training. For this reason, we added to the training process an additional objective that captures the shape of the 4-points prediction, so that the DNN is encouraged to identify the edges with coordinates that form a rectangle.

Considering the reference nomenclature reported in Figure 4, with $\vec{c_i} = [c_i^x, c_i^y]^T, \vec{e_i} = [e_i^x, e_i^y]^T \quad \forall i \in \{1, 2, 3, 4\}$ we can define $\tilde{P}$:

$$\tilde{P} = |cos(\alpha_1)| + |cos(\alpha_2)| + |cos(\alpha_3)| + |cos(\alpha_4)|. \quad (1)$$

It is immediate to see that $\tilde{P}$ is minimized when the $\vec{e}$s are perpendicular.

Recalling the properties of the scalar product between two column vectors, $< \vec{a_1}, \vec{a_2} >= \vec{a_1}^T \vec{a_2}$, we can write

$$\begin{aligned} \tilde{P} = & \frac{| < \vec{e_1}, \vec{e_2} > |}{||\vec{e_1}|| \cdot ||\vec{e_2}||} + \frac{| < \vec{e_2}, \vec{e_3} > |}{||\vec{e_2}|| \cdot ||\vec{e_3}||} + \\ & + \frac{| < \vec{e_3}, \vec{e_4} > |}{||\vec{e_3}|| \cdot ||\vec{e_4}||} + \frac{| < \vec{e_4}, \vec{e_1} > |}{||\vec{e_4}|| \cdot ||\vec{e_1}||}. \end{aligned} \quad (2)$$

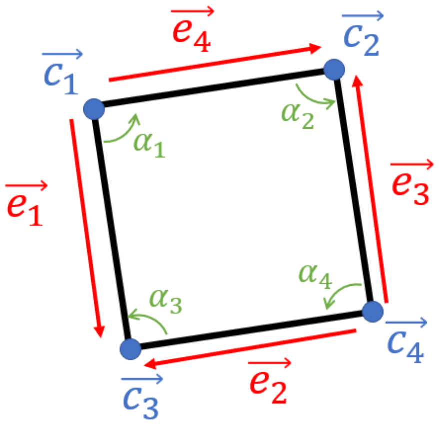

Fig. 4: Reference system for the custom loss function evaluation.

The inclusion of (2) as an additive term to the loss function makes so the DNN is encouraged to predict rectangles, in line with our objectives.

In order to assure that (2) is a proper term in a loss function, and that it is suitable for real-world deployment, we have to make sure that it can be evaluated with GPU-acceleration compliant operations. To do so, we introduce the matrices $C \in \mathbb{R}^{2\times4}$,$E \in \mathbb{R}^{2\times4}$:

$$\begin{aligned} C &= [\vec{c_1}, \vec{c_2}, \vec{c_3}, \vec{c_4}], \\ E &= [\vec{e_1}, \vec{e_2}, \vec{e_3}, \vec{e_4}]. \end{aligned} \tag{3}$$

One has that

$$\begin{aligned} \vec{e_1} &= \vec{c_3} - \vec{c_1} \\ \vec{e_2} &= \vec{c_4} - \vec{c_3} \\ \vec{e_3} &= \vec{c_2} - \vec{c_4}, \\ \vec{e_4} &= \vec{c_2} - \vec{c_1} \end{aligned} \tag{4}$$

meaning that

$$E = C \cdot \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 \end{bmatrix}. \tag{5}$$

Equation (5) states that, starting from the DNN predictions $\vec{c_1}, \vec{c_2}, \vec{c_3}, \vec{c_4}$], the vectors $\vec{e_1}, \vec{e_2}, \vec{e_3}, \vec{e_4}$ can be obtained with a simple (and fixed) matrix multiplication. Once the matrix E is available, evaluating (2) consists in straightforward vector products, transposes and basic arithmetic operations, meaning that it can be trivially *broadcasted* (i.e., vectorized over the entire training dataset) by standard DNN frameworks, such as TensorFlow and Keras. The overall training loss considered for our AOI system is the the sum between the mean squared error (MSE) on the edge coordinates and $\tilde{P}$.

Other than the loss function, it is a common practice to define additional quantities, known as *metrics*, to evaluate the training quality and the DNN performance. The main difference between losses and metrics is that metrics are not actively minimized during the DNN training, so they may capture additional aspects of interest for the task at hand and they may be used to assess whether the considered DNN correctly solves the given task and further evaluate the quality of its predictions. In our setting, we employed as a metric the Mean Absolute Error (MAE) on the rectangle center prediction, as the centering precision is an important sub-task for the AOI problem.

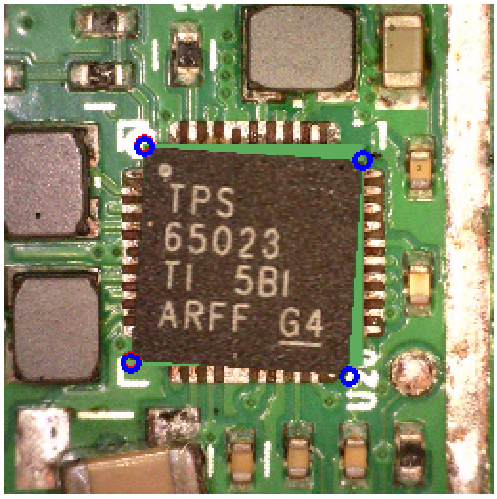

Fig. 5: Example of synthetic PCBA image with edge labels hghlghted. Blue circles represent the DNN edge location predicition, whereas red circles highlight their real location. Note that red circles are almost entirely covered by the the blue ones, as this figure reports the result of an evaluation done by the trained DNN)

C. *Proposed Neural Network Architecture*

The DNN proposed in this study employs TL to initialize the weights of most of its layers. As customary in most TL applications, the first layers of the proposed DNN are directly taken from a pre-trained DNN and are then “frozen” (i.e., kept constant) during the training. For our design, we employed for this purpose the MobileNetV2 [28] pre-trained on the ImageNet dataset. The reason for the choice of MobileNetV2 is that its design makes it particularly suitable for complex applications on low-computing hardware, as it employs both the so-called *inverted residual blocks* (to support the presence of several layers) and *separable convolutions* (to reduce the number of trainable weights and computations).

The last fully connected, or *dense*, layers from the MobileNetV2 are replaced by an additional separable convolutional layer (with 8 filters 5×5 filters), and a dense layer with 512 neurons. A final dense layer with 8 neurons computes the output of the DNN, that is a 8×1 vector containing the edges x and y coordinates. All neurons employed ReLU activation functions and the dense layer had a dropout of 0.1 to reduce overfitting.

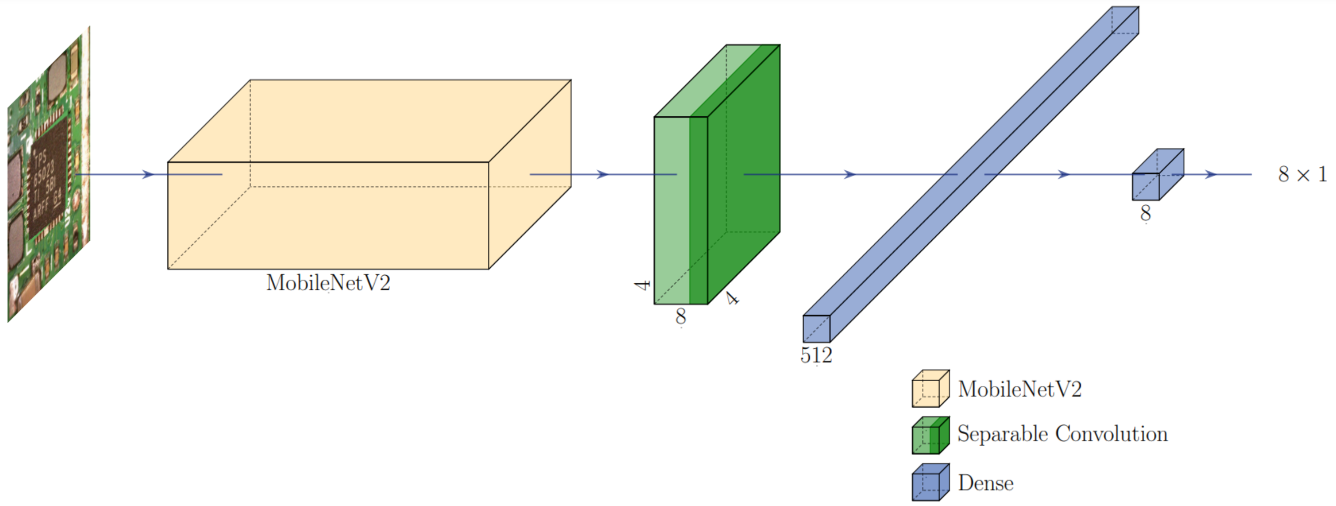

Fig. 6: Architecture of the proposed Deep Neural Network. The first layers, grouped in the yellow box, are taken from the MobileNetV2 network that was trained on the ImageNet dataset. The remaining layers were trained on the PCBA dataset synthesized in this work, hence following a transfer learning approach.

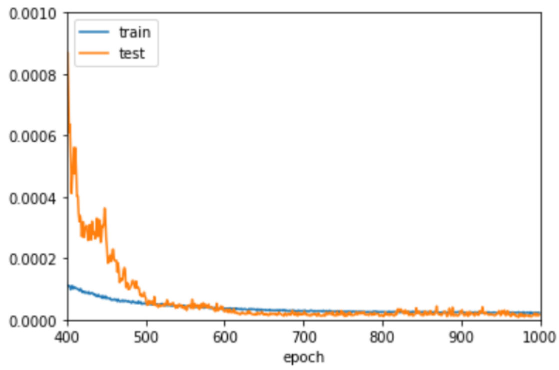

Fig. 7: Training and validation curves for the custom loss over the training epochs.

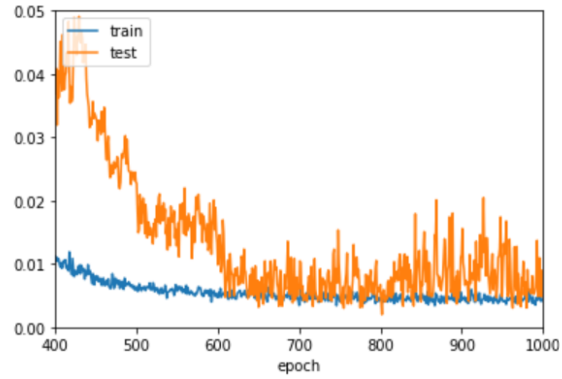

Fig. 8: Training and validation curves for the chosen metric (MAE on rectangle center) over the training epochs.

The resulting architecture is shown in Figure 6.

V. Testing and Results

For the training of the DNN we used 85% of the 2000 images generated from the 10 PCBA photos as our training set, keeping the rest for the DNN validation as its test dataset. In addition, we further augment the test dataset with 300 more images generated from two unused PCBA photos to better test the generalization capabilities of the DNN. We ran the training for 1000 epochs, using Adam [29] as the training optimizer with a mini-batch size of 32 images.

Figure 7 reports the training performance in terms of loss evaluations during the process. It can be seen that after about 600 epochs the training stabilizes, with the performance on the training and test set becoming very similar. A similar behaviour is observed in Figure 8, where, despite minor fluctuations, it can be seen how the trained DNN correctly identifies the center of the chip even on images that were not provided in its training dataset, further proving its generalization capabilities.

The entirety of the training was conducted on a standard desktop machine equipped with a 4GB GPU and the resulting trained DNN is suitable for deployment on low-power hardware. The code was developed in Python using the latest stable versions of Keras and Tensorflow.

VI. Conclusions and Future Works

The present paper presented the design process of a Deep Neural Network (DNN) for an Automatic Optical Inspection (AOI) process in Printed Circuit Board Assembly (PCBA) manufacturing. The developed system is tailored for a real industrial use case and represents a preliminary validation study for the development of a low-cost solution that may be deployed for ad hoc quality checks in assembly lines.

The case study considered involved the identification of defective boards by the automated visual inspection of the correct positioning of a critical microchip, as it represents the most common point of failure in the final product assembly.

The solution proposed can be easily adapted to other specific mounting/placement controls and does not require extensive data collection campaigns, as it exploits the results of both transfer learning and synthetic data generation. The DNN of choice for being the basis of the transfer learning process is the well-known MobileNetV2 [28], as its design is tailored for devices with low computing power, making it an ideal choice for being deployed on industrial hardware.

Future works envisage the realization of a physical prototype able to capture PCBA images automatically, so that it may employ the trained DNN to detect defects in a real production line. The extension of the quality checks to other components and types of defects is also under development.

REFERENCES

[1] P. Hogue and J. D. Coopersmith, "Development of an automated optical inspection system for determining percent area coverage for spacecraft contamination control," in *SPIE Proceedings*, P. T. C. Chen, J. C. Fleming, and M. G. Dittman, Eds. SPIE, Oct. 2004.

[2] A. Zbrowski, "Cleaning system for automated optical inspection of heads of bearing rollers," *Solid State Phenomena*, vol. 198, pp. 289–294, Mar. 2013.

[3] F.-S. Yang, C.-C. Ho, and L.-C. Chen, "Automated optical inspection system for o-ring based on photometric stereo and machine vision," *Applied Sciences*, vol. 11, no. 6, p. 2601, Mar. 2021.

[4] M.-G. Retzlaff, J. Stabenow, J. Beyerer, and C. Dachsbacher, "Synthesizing images using parameterized models for automated optical inspection (AOI)," *tm - Technisches Messen*, vol. 82, no. 5, pp. 251–261, Apr. 2015.

[5] S. Cubero, N. Aleixos, E. Moltó, J. Gómez-Sanchis, and J. Blasco, "Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables," *Food and Bioprocess Technology*, vol. 4, no. 4, pp. 487–504, Jul. 2010.

[6] H.-J. Kwon and T.-H. Park, "Automated optical inspection for holograms with mixed patterns," *The International Journal of Advanced Manufacturing Technology*, vol. 54, no. 1-4, pp. 215–221, Sep. 2010. [Online]. Available: https://doi.org/10.1007/s00170-010-2915-9

[7] A. A. R. M. A. Ebayyeh and A. Mousavi, "A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry," *IEEE Access*, vol. 8, pp. 183 192–183 271, 2020.

[8] S.-H. Huang and Y.-C. Pan, "Automated visual inspection in the semiconductor industry: A survey," *Computers in Industry*, vol. 66, pp. 1–10, Jan. 2015.

[9] T. Parakontan and W. Sawangsri, "Development of the machine vision system for automated inspection of printed circuit board assembl," in *2019 3rd International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, Jun. 2019.

[10] D. Raymond and D. Haigh, "Why automate optical inspection?" in *Proceedings International Test Conference 1997*. Int. Test Conference, 1997.

[11] V. Reshadat and R. A. Kapteijns, "Improving the performance of automated optical inspection (AOI) using machine learning classifiers," in *2021 International Conference on Data and Software Engineering (ICoDSE)*. IEEE, Nov. 2021.

[12] M. Metzner, D. Fiebag, A. Mayr, and J. Franke, "Automated optical inspection of soldering connections in power electronics production using convolutional neural networks," in *2019 9th International Electric Drives Production Conference (EDPC)*. IEEE, Dec. 2019.

[13] H. G. Nguyen, M. Meiners, L. Schmidt, and J. Franke, "Deep learning-based automated optical inspection system for crimp connections," in *2020 10th International Electric Drives Production Conference (EDPC)*. IEEE, Dec. 2020.

[14] H.-C. Liao, Z.-Y. Lim, Y.-X. Hu, and H.-W. Tseng, "Guidelines of automated optical inspection (AOI) system development," in *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*. IEEE, Jul. 2018.

[15] S. Kim, W. Kim, Y.-K. Noh, and F. C. Park, "Transfer learning for automated optical inspection," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, May 2017.

[16] B. Ghosh, M. K. Bhuyan, P. Sasmal, Y. Iwahori, and P. Gadde, "Defect classification of printed circuit boards based on transfer learning," in *2018 IEEE Applied Signal Processing Conference (ASPCON)*. IEEE, Dec. 2018.

[17] L. H. de S. Silva, G. O. de A. Azevedo, B. J. T. Fernandes, B. L. D. Bezerra, E. B. Lima, and S. C. Oliveira, "Automatic optical inspection for defective PCB detection using transfer learning," in *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. IEEE, Nov. 2019.

[18] I. Volkau, A. Mujeeb, D. Wenting, E. Marius, and S. Alexei, "Detection defect in printed circuit boards using unsupervised feature extraction upon transfer learning," in *2019 International Conference on Cyberworlds (CW)*. IEEE, Oct. 2019.

[19] A. Boikov, V. Payor, R. Savelev, and A. Kolesnikov, "Synthetic data generation for steel defect detection and classification using deep learning," *Symmetry*, vol. 13, no. 7, p. 1176, Jun. 2021. [Online]. Available: https://doi.org/10.3390/sym13071176

[20] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee, "Automated defect inspection system for metal surfaces based on deep learning and data augmentation," *Journal of Manufacturing Systems*, vol. 55, pp. 317–324, Apr. 2020. [Online]. Available: https://doi.org/10.1016/j.jmsy.2020.03.009

[21] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends*. IGI Global, 2010, pp. 242–264.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[23] X. Li, Y. Grandvalet, F. Davoine, J. Cheng, Y. Cui, H. Zhang, S. Belongie, Y.-H. Tsai, and M.-H. Yang, "Transfer learning in computer vision tasks: Remember where you come from," *Image and Vision Computing*, vol. 93, p. 103853, Jan. 2020.

[24] S. Chen, K. Ma, and Y. Zheng, "Med3d: Transfer learning for 3d medical image analysis," 2019. [Online]. Available: arXiv:1904.00625

[25] E. Deniz, A. Şengür, Z. Kadiroğlu, Y. Guo, V. Bajaj, and Ümit Budak, "Transfer learning based histopathologic image classification for breast cancer detection," *Health Information Science and Systems*, vol. 6, no. 1, Sep. 2018.

[26] C. M. Valensise, A. Giuseppi, F. Vernuccio, A. D. la Cadena, G. Cerullo, and D. Polli, "Removing non-resonant background from CARS spectra via deep learning," *APL Photonics*, vol. 5, no. 6, p. 061305, Jun. 2020.

[27] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A LiDAR point cloud generator," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ACM, Jun. 2018.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2018.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980