Jurnal Teknik Industri: Jurnal Hasil Penelitian dan Karya Ilmiah dalam Bidang Teknik Industri

Vol. 12, No. 1, 2026

A Lightweight Machine Vision Pipeline for Screen-Printing Defect Detection in MSMEs Using Low-Cost Image Acquisition

Galih Mahardika Munandar[1], Tiyan Fatkhurrohman [2], Lazuardi Fatahilah Hamdi[3]
[1] Department of Industrial Engineering, Faculty of Science and Humaniora, Universitas Muhammadiyah Gombong
Jl. Yos Sudarso No. 461, Wonokriyo, Gombong, Kebumen Jawa Tengah, 54411
Email: galihmahardika@unimugo.ac.id
[2] Department of Management, Faculty of Science and Humaniora, Universitas Muhammadiyah Gombong
Jl. Yos Sudarso No. 461, Wonokriyo, Gombong, Kebumen Jawa Tengah, 54411
Email: tiyanfatkhurrokhman@gmail.com
[3] Department of Information Technology, Faculty of Science and Humaniora, Universitas Muhammadiyah Gombong
Jl. Yos Sudarso No. 461, Wonokriyo, Gombong, Kebumen Jawa Tengah, 54411
Email: lazuardi@unimugo.ac.id

ABSTRACT

This study addresses the need for affordable visual inspection support in micro, small, and medium enterprises (MSMEs) engaged in screen-printing production. Although machine vision and deep learning have been widely applied in manufacturing quality control, many existing systems are designed for relatively controlled industrial settings and require stable cameras, lighting, computing resources, and technical expertise. This condition limits direct adoption by small MSMEs, where image acquisition is often performed with operator-level devices under variable lighting and background conditions. This study designed and evaluated an initial low-resource visual inspection pipeline consisting of low-cost image acquisition, five-class defect labeling, MobileNetV3-based transfer learning, performance evaluation, and TensorFlow Lite conversion. The dataset consisted of 160 screen-printing images grouped into five classes: good, misalignment, bleeding, pinholes, and ghosting. The preliminary evaluation yielded 24.38% multiclass accuracy and a loss of 2.5635, indicating that the model could not yet reliably distinguish detailed defect categories. The converted TensorFlow Lite model was 5.43 MB, indicating that the technical conversion path was feasible. A binary quality-control interpretation produced 75.63% accuracy, but 27 defective images were still predicted as pass QC. Therefore, the pipeline cannot be treated as a final quality-control decision system. The main contribution of this study is empirical evidence that image-acquisition quality, dataset sufficiency, class separability, and training configuration are critical bottlenecks in developing lightweight deep-learning-based inspection for low-resource MSME environments.

Keywords: *Machine Vision; MSME; MobileNetV3; TensorFlow Lite; Quality Control; Screen-Printing Defect Detection*

Introduction

Micro, small, and medium enterprises (MSMEs) require continuous technological adaptation to remain competitive in production, marketing, and quality control [1] [2]. In small-scale screen-printing production, quality control is still commonly conducted through direct visual inspection by operators. This practice is simple and inexpensive, but it is highly dependent on operator attention, lighting conditions, visual consistency, and experience. Defects such as misalignment, bleeding, pinholes, and ghosting may go undetected until late, leading to rework, material waste, and reduced production efficiency.

Machine vision has become an important approach in manufacturing quality control because it supports image-based inspection, dimensional analysis, product classification, and defect detection [3], [4], [5], [6]. Several industrial studies show that machine vision can support more systematic quality evaluation in cyber-physical production systems, wood-plank inspection, and rotary-switch production [7], [8], [9]. However, these applications commonly assume more controlled acquisition conditions, including industrial cameras, stable illumination, dedicated computing resources, and technical expertise. These assumptions are not always compatible with MSMEs, especially small screen-printing businesses with limited investment capacity and non-standardized work environments.

Low-cost computer vision studies in embedded systems, agriculture, water-level measurement, and fruit sizing indicate that practical image-based systems can be developed with affordable hardware when acquisition constraints are explicitly considered [10], [11], [12], [13]. In parallel, image classification literature has evolved from conventional classifiers such as SVM to convolutional neural network approaches and transfer learning [14], [15], [16], [17]. Lightweight architectures such as MobileNetV3 and other compact deep-learning models are also relevant for resource-constrained inference because they can reduce computational requirements and support mobile-oriented applications [18], [19], [20], [21], [22]. TensorFlow Lite and edge-oriented optimization further enable trained models to be converted into formats suitable for mobile or edge deployment [23], [24], [25].

Nevertheless, previous studies do not fully address the methodological challenge faced in MSME screen-printing inspection: a lightweight model may be computationally efficient, but its performance still depends on acquisition consistency, dataset sufficiency, and visual separability among defect classes. Surface-defect detection using CNNs is particularly sensitive to image quality, defect scale, class imbalance, and similarity among defect patterns [26]. Therefore, the research gap is not simply the absence of a screen-printing defect classifier, but also the limited empirical evaluation of how a lightweight deep learning pipeline performs under low-resource MSME acquisition conditions.

To address this gap, this study designs and evaluates an initial visual inspection pipeline for screen-printing defect detection at Tonight Sablon. The pipeline combines operator-level image acquisition, five-class labeling, MobileNetV3 transfer learning, multiclass and binary QC evaluation, and TensorFlow Lite conversion. This study does not claim to develop a final quality-control system or to replace manual inspection. Instead, it is positioned as a proof-of-concept to examine the feasibility and limitations of lightweight visual inspection in a low-resource MSME context.

The contribution of this study is threefold. First, it documents an initial low-cost image-acquisition and labeling pipeline for screen-printing defect images in an MSME setting. Second, it evaluates the early performance of MobileNetV3 and TensorFlow Lite conversion under limited-data conditions. Third, it interprets low multiclass accuracy and false-pass results as empirical evidence that acquisition quality, dataset size, class imbalance, and training configuration are critical bottlenecks before lightweight deep learning can be developed into a reliable visual inspection support tool.

Research Methods

Research Design

This research used an experimental computational design focused on pipeline feasibility. The pipeline consisted of image acquisition, class labeling, preprocessing, data splitting, MobileNetV3-based transfer learning, performance evaluation, and TensorFlow Lite conversion. The unit of analysis was a screen-printing product image obtained from MSME production. The model output consisted of five classes: good, misalignment, bleeding, pinholes, and ghosting. The experiment was intentionally designed as a feasibility study rather than a complete industrial inspection system.

Image Acquisition

Image acquisition was designed to reflect the practical constraints of small MSMEs. Images were collected using operator-level devices rather than industrial cameras. This decision aligned with the goal of evaluating a realistic, low-cost visual inspection pipeline. The acquisition process retained several real-world variations, including differences in camera quality, image sharpness, focus, distance, angle, lighting, shadow, and background. These variations were not fully eliminated because they represent the conditions that may occur when MSME operators use available devices. Thus, image acquisition was treated not only as a technical step but also as a methodological factor influencing model performance.

Dataset Description

The dataset consisted of 160 images distributed into five classes. The distribution was not fully balanced because images were collected from available production conditions and observed defect occurrences. The dataset was used for proof-of-concept evaluation and was not intended to establish a final high-accuracy model. Table 1 presents the dataset distribution.

Table 1. Dataset distribution

Class	**Number of Images**	**Description**
Good	24	Products visually passing QC.
Misalignment	44	Shifted or nonaligned screen-printing patterns.
Bleeding	34	Ink spread beyond the intended boundary.
Pinholes	33	Small missing dots or hole-like defects.
Ghosting	25	Shadow-like or duplicated visual impressions.
Total	**160**	

Preprocessing, Data Split, and Model Configuration

Each image was resized to 224 x 224 pixels and normalized before being used as an input to MobileNetV3. The dataset was split into 70:20:10 for training, validation, and testing. This split was used to support initial model learning, validation monitoring, and early evaluation. However, because the total dataset was limited, the evaluation results were interpreted as an indicator of pipeline feasibility rather than final model performance.

MobileNetV3 was selected because lightweight convolutional neural networks are relevant for mobile-oriented and resource-constrained applications [18], [19], [20], [21], [22]. Transfer learning was applied because the target dataset was small, and pretrained visual representations can support initial classification learning [14], [16], [17]. The model was trained for three epochs as an initial experiment to verify the training pipeline. This limited duration is acknowledged as a methodological constraint, as three epochs are insufficient to form stable decision boundaries on a small, imbalanced, and visually similar multiclass dataset.
The detailed training configuration is summarized in Table 2.

Table 2. Experimental configuration

Component	**Configuration**
Framework	TensorFlow/Keras
Base model	MobileNetV3Small
Input size	224 × 224 × 3 pixels
Number of classes	5 classes
Class labels	Good, misalignment, bleeding, pinholes, ghosting
Label mode	categorical
Batch size	32
Optimizer	Adam
Learning rate	0.001
Loss function	categorical cross-entropy
Metrics	accuracy
Epochs	3
TFLite conversion	Standard TensorFlow Lite conversion using TFLiteConverter.from_keras_model()
Quantization	Not applied / no explicit quantization
TFLite model size	5.43 MB

TensorFlow Lite Conversion and Evaluation Metrics

After training, the model was converted into TensorFlow Lite format to verify the technical path toward mobile or edge deployment. The converted model was 5.43 MB. Conversion success was interpreted as technical feasibility only; it was not treated as evidence of deployment readiness. The model still requires further testing to assess predictive reliability, latency, memory consumption, and false-positive risk before practical use.

Evaluation used accuracy, loss, precision, recall, F1-score, and confusion matrix. Multiclass performance was evaluated for five original classes. In addition, a binary QC interpretation was calculated by grouping good as pass QC and all defect classes as not pass QC. This binary interpretation was used solely to assess the system's early screening potential and was not intended to claim that the system can replace manual inspection.

Results and Discussion

Dataset Characteristics and Visual Samples

The dataset contained 160 images with unequal class distribution. Misalignment had the most samples, whereas good had the fewest. This imbalance is relevant because deep learning models are sensitive to class imbalance, particularly when the dataset is small. The visual boundary between good and defective samples was also not always clear, as the images were obtained with operator-level devices and varied in acquisition quality. Representative available image samples are shown in Figure 1.

***Figure 1**. Representative screen-printing image samples illustrating available defect classes and acquisition variability in the low-cost image collection process.*

Training and TensorFlow Lite Conversion Results

The initial training and conversion results are presented in Table 3. The model was successfully trained and converted to TensorFlow Lite. However, the multiclass accuracy remained low, indicating that the model could not yet reliably distinguish the five classes. Therefore, the result should be interpreted as evidence that the pipeline is executable, not as evidence of a ready-to-use QC system.

Table 3. Training and conversion results

Component	Result	Interpretation
Model architecture	MobileNetV3	Lightweight image-classification architecture.
Number of epochs	3	Initial feasibility experiment.
Multiclass accuracy	24.38%	Low classification performance.
Loss	2.5635	Indicates unstable class separation.
TensorFlow Lite model size	5.43 MB	Converted model; further optimization required.
Conversion status	Successful	Pipeline technically executable.

With five classes, random guessing under a balanced condition would be approximately 20%. The obtained accuracy of 24.38% was only slightly above this baseline. This indicates that the model still had difficulty learning class-specific visual patterns. The main causes may include limited dataset size, unbalanced class distribution, visually similar defect characteristics, poor camera quality, inconsistent focus, unstable lighting, and insufficient training duration.

Multiclass Confusion Matrix and Error Analysis

Table 4 presents the multiclass confusion matrix. The diagonal values represent correct predictions. The model correctly predicted 39 of 160 available prediction records, yielding 24.38% accuracy.

Table 4. Multiclass confusion matrix

Actual / Predicted	good	misalignment	bleeding	pinholes	ghosting	Total
Good	12	5	3	3	1	24
Misalignment	11	11	5	9	8	44
Bleeding	4	8	7	9	6	34
Pinholes	5	12	5	5	6	33

Ghosting	7	3	7	4	4	25
Total prediction	**39**	**39**	**27**	**30**	**25**	**160**

The confusion matrix shows that errors were distributed across several classes, suggesting that the model had not formed stable decision boundaries among defect categories. The good class had the highest number of correct predictions, but 12 good samples were still classified as defect classes. Misalignment was frequently confused with good, pinholes, and ghosting. Pinholes were often predicted as misalignment, while bleeding was frequently predicted as pinholes.

These error patterns are consistent with the visual characteristics of screen-printing defects. Pinholes usually appear as small local defects and may disappear under blur, noise, or low sharpness. Ghosting has a shadow-like appearance that can resemble lighting artifacts or image blur. Misalignment may be difficult to distinguish from good samples if the shift is small or the reference boundary is not clearly visible. Therefore, low accuracy should not be interpreted only as model failure; it also reflects acquisition inconsistency, defect similarity, and limited data representation.

Class-Level Performance and Binary QC Interpretation

The class-level precision, recall, and F1-score are shown in Table 5. The results confirm that the model performance was low across all classes. The good class achieved the highest recall, but its precision remained low, indicating that many images labeled as good actually came from defect classes. This condition is critical because false-pass predictions may lead to defective products being treated as acceptable.

Table 5. Precision, recall, and F1-score by class

Class	**Precision**	**Recall**	**F1-score**
Good	30.77%	50.00%	38.10%
Misalignment	28.21%	25.00%	26.51%
Bleeding	25.93%	20.59%	22.95%
Pinholes	16.67%	15.15%	15.87%
Ghosting	16.00%	16.00%	16.00%
Macro Average	23.51%	25.35%	23.89%

Although detailed multiclass performance was low, the practical needs of small MSMEs may initially focus on separating products that pass visual inspection from those requiring further inspection. Therefore, the five-class prediction results were also interpreted as a binary QC scenario, as shown in Table 6.

Table 6. Binary QC interpretation

Actual Condition	**Predicted Pass QC**	**Predicted Not Pass QC**	**Total**
Pass QC	12	12	24
Not Pass QC	27	109	136
Total	39	121	160

Based on this binary grouping, the accuracy was 75.63%. The recall for not pass QC was 80.15%, while the precision for not pass QC was 90.08%. However, 27 defective images were still predicted as pass QC. This false-pass condition is unacceptable for a final QC decision because defective products may be released as acceptable. Thus, the binary result should be interpreted only as an exploratory indication of early screening potential, not as evidence that the system can replace manual inspection.

Discussion

The main finding of this study is that the technical pipeline can be executed from image acquisition to MobileNetV3 training and TensorFlow Lite conversion, but the current model performance is still insufficient for reliable defect classification. This finding supports the broader understanding that machine vision performance depends strongly on acquisition quality, lighting, object presentation, and data sufficiency [5], [6], [26]. In this study, the use of operator-level devices increased affordability and contextual realism but also led to unstable image quality. This creates a trade-off between affordability and acquisition consistency in MSME-oriented machine vision.

Compared with machine-vision studies conducted in more controlled industrial environments [3], [4], [7], [8], [9], this study intentionally retained part of the acquisition variability to reflect realistic MSME conditions. This approach provides a different contribution: it shows that low-cost image acquisition is not merely a supporting component but a decisive bottleneck in the learning pipeline. Even with a lightweight architecture, noisy or inconsistent images can prevent the model from learning stable class boundaries.

The TensorFlow Lite result supports the feasibility argument from a deployment-path perspective. A converted model size of 5.43 MB indicates that mobile-oriented conversion is feasible. However, edge deployment should not be evaluated only by model size. Predictive reliability, latency, memory consumption, user interaction, and false-pass risk must also be examined before implementation. In MSME settings, a human-centered approach is necessary because the operator should remain the final decision-maker when model uncertainty and false-pass risk remain high.

Limitations and Future Work

This study has several limitations. First, the dataset contained only 160 images, which is small for multiclass deep-learning classification. Second, the class distribution was not fully balanced because data were collected from available production conditions. Third, image acquisition used operator-level devices, causing variations in camera quality, lighting, focus, distance, angle, and background. Fourth, the model was trained for only three epochs, limiting the ability to form stable decision boundaries. Fifth, the study did not yet evaluate latency, memory consumption, quantization effects, or real-time performance on an edge device.

Future research should standardize image acquisition by using controlled lighting, a fixed camera distance, a stable background, and operator guidelines. The dataset should be enlarged and balanced across defect classes. Additional experiments should evaluate data augmentation, longer training duration, alternative lightweight architectures, TensorFlow Lite optimization strategies, and visual explanation methods such as Grad-CAM. Before practical implementation, the system must be tested for false-pass rate, prediction stability, latency, and usability in MSME production workflows.

Conclusion

This study designed and evaluated an initial lightweight machine vision pipeline for screen-printing defect detection in an MSME context using operator-level image acquisition, MobileNetV3 transfer learning, and TensorFlow Lite conversion. The multiclass evaluation yielded 24.38% accuracy and a loss of 2.5635, indicating that the model was not yet able to reliably classify the five defect categories. The TensorFlow Lite conversion was successful, producing a 5.43 MB model, demonstrating that the technical conversion path to mobile or edge deployment is feasible.

The contribution of this study is not a ready-to-use quality-control system but rather empirical evidence on the feasibility and limitations of lightweight visual inspection under low-resource MSME acquisition conditions. The findings show that image-acquisition quality, dataset size, class imbalance, defect similarity, and limited training configuration are critical factors for future development. The binary QC interpretation suggests potential for early screening, but the presence of defective images that are predicted as pass QC indicates that the pipeline cannot be used as a final QC decision tool. Further development must prioritize standardizing acquisition, using larger, more balanced datasets, adopting more robust training strategies, and evaluating edge devices before practical deployment.

References

[1] L. Setiyani and Y. Rostiani, “Analysis of E-Commerce Adoption by SMEs Using the Technology-Organization-Environment (TOE) Model: A Case Study in Karawang, Indonesia,” *Int. J. Sci.*, 2024, [Online]. Available: http://ijstm.inarah.co.id

[2] G. M. Munandar and A. Dicasani, “Hubungan Pelanggan dengan Pelaku UMKM Konveksi di Kebumen dengan Pendekatan e-CRM,” *J. Teknol. dan Manaj. Ind. Terap.*, vol. 4, no. 3, pp. 855–860, 2025, doi: 10.55826/jtmit.v4i3.792.

[3] T. Coito, M. S. E. Martins, B. Firme, J. Figueiredo, S. M. Vieira, and J. M. C. Sousa, “Assessing the impact of automation in pharmaceutical quality control labs using a digital twin,” *J. Manuf. Syst.*, vol. 62, no. December 2021, pp. 270–285, 2022, doi: 10.1016/j.jmsy.2021.11.014.

[4] C. El Hachem, G. Perrot, L. Painvin, and R. Couturier, “Automation of Quality Control in the Automotive Industry Using Deep Learning Algorithms,” *2021 Int. Conf. Comput. Control Robot. ICCCR 2021*, pp. 123–127, 2021, doi: 10.1109/ICCCR49711.2021.9349273.

[5] A. Akundi and M. Reyna, “A Machine Vision Based Automated Quality Control System for Product Dimensional Analysis,” *Procedia Comput. Sci.*, vol. 185, no. June, pp. 127–134, 2021, doi:

10.1016/j.procs.2021.05.014.

[6] S. Kim, T. P. Nguyen, and J. Yoon, “A Systematic Review of Machine Vision Applications in Factory and Manufacturing Processes: From Quality Control to Predictive Diagnostics,” *Int. J. Precis. Eng. Manuf. - Green Technol.*, 2025, doi: 10.1007/s40684-025-00769-2.

[7] A. V. Gurjanov, V. I. Babenkov, A. V. Shukalov, I. O. Zharinov, and O. O. Zharinov, “Total quality control of the cyber-physical production using machine vision technologies,” *J. Phys. Conf. Ser.*, vol. 1889, no. 5, 2021, doi: 10.1088/1742-6596/1889/5/052014.

[8] M. Cinal, A. Sioma, and B. Lenty, “The Quality Control System of Planks Using Machine Vision,” *Appl. Sci.*, vol. 13, no. 16, 2023, doi: 10.3390/app13169187.

[9] I. Karacan, I. Erdogan, M. C. Igdil, and U. Cebeci, “Machine Vision Supported Quality Control Applications in Rotary Switch Production by Using Both Process FMEA and Design FMEA,” *Nat. Appl. Sci. J.*, vol. 4, no. 2, pp. 16–31, 2021, doi: 10.38061/idunas.850545.

[10] J. Jaramillo, J. Vanden Heuvel, and K. H. Petersen, “Low-Cost, Computer Vision-Based, Prebloom Cluster Count Prediction in Vineyards,” *Front. Agron.*, vol. 3, no. April, pp. 1–10, 2021, doi: 10.3389/fagro.2021.648080.

[11] J. M. G. P. Isidoro, R. Martins, R. F. Carvalho, and J. L. M. P. de Lima, “A high-frequency low-cost technique for measuring small-scale water level fluctuations using computer vision,” *Meas. J. Int. Meas. Confed.*, vol. 180, no. February, 2021, doi: 10.1016/j.measurement.2021.109477.

[12] G. Bortolotti *et al.*, “A computer vision system for apple fruit sizing by means of low-cost depth camera and neural network application,” *Precis. Agric.*, vol. 25, no. 6, pp. 2740–2757, 2024, doi: 10.1007/s11119-024-10139-8.

[13] L. D. Ortega, E. S. Loyaga, P. J. Cruz, H. P. Lema, J. Abad, and E. A. Valencia, “Low-Cost Computer-Vision-Based Embedded Systems for UAVs,” *Robotics*, vol. 12, no. 6, pp. 1–20, 2023, doi: 10.3390/robotics12060145.

[14] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, “Transfer learning for medical image classification: a literature review,” *BMC Med. Imaging*, vol. 22, no. 1, pp. 1–13, 2022, doi: 10.1186/s12880-022-00793-7.

[15] M. A. Chandra and S. S. Bedi, “Survey on SVM and their application in image classification,” *Int. J. Inf. Technol.*, vol. 13, no. 5, 2021, doi: 10.1007/s41870-017-0080-1.

[16] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, “Review of Image Classification Algorithms Based on Convolutional Neural Networks,” *Remote Sens*, pp. 1–50, 2021, doi: https://doi.org/10.3390/rs13224712.

[17] M. Mehmood, A. Shahzad, B. Zafar, A. Shabbir, and N. Ali, “Remote Sensing Image Classification: A Comprehensive Review and Applications,” *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/5880959.

[18] S. Qian, Y. Hu, and C. Ning, “MobileNetV3 for Image Classification,” *6th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2022 - Proc.*, pp. 490–497, 2022, doi: 10.3390/proceedings2021072002.

[19] X. Kong and Z. Ge, “Deep PLS: A Lightweight Deep Learning Model for Interpretable and Efficient Data Analytics,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 11, pp. 8923–8937, 2023, doi: 10.1109/TNNLS.2022.3154090.

[20] Z. Y. Deng, H. H. Chiang, L. W. Kang, and H. C. Li, “A lightweight deep learning model for real-time face recognition,” *IET Image Process.*, vol. 17, no. 13, pp. 3869–3883, 2023, doi: 10.1049/ipr2.12903.

[21] A. A. A. El-Latif, S. A. Chelloug, M. Alabdulhafith, and M. Hammad, “Accurate Detection of Alzheimer’s Disease Using Lightweight Deep Learning Model on MRI Data,” *Diagnostics*, vol. 13, no. 7, pp. 1–21, 2023, doi: 10.3390/diagnostics13071216.

[22] Z. Li, X. Lei, and S. Liu, “A lightweight deep learning model for cattle face recognition,” *Comput. Electron. Agric.*, vol. 195, no. January, p. 106848, 2022, doi: 10.1016/j.compag.2022.106848.

[23] S. O. Bursa, O. D. Incel, and G. I. Alptekin, “Building lightweight deep learning models with TensorFlow Lite for human activity recognition on mobile devices,” *Ann. des Telecommun*, vol. 78, no. 11–12, pp. 687–702, 2023, doi: 10.1007/s12243-023-00962-x.

[24] I. N. K. Wardana, J. W. Gardner, and S. A. Fahmy, “Optimising deep learning at the edge for accurate hourly air quality prediction,” *Sensors*, vol. 21, no. 4, 2021, doi: 10.3390/s21041064.

[25] M. H. Akhtar, I. Eksheir, and T. Shanableh, “Edge-optimized deep learning architectures for classification of agricultural insects with mobile deployment,” *Information*, vol. 16, no. 5, 2025, doi: 10.3390/info16050348.

[26] E. Cumbajin *et al.*, “A systematic review on deep learning with CNNs applied to surface defect detection,” *J. Imagin*, vol. 9, no. 10, 2023, doi: 10.3390/jimaging9100193.