Journal of Imaging Science and Technology® 67(5): 1–10, 2023.

Automotive Paint Defect Classification: Factory-Specific Data Generation using CG Software for Deep-Learning Models

Kazuki Iwata[†] **and Haotong Guo**[†]
Graduate School of Science and Engineering, Chiba University, Chiba, Japan
E-mail: iwata.k.0811@gmail.com, k.kaku@chiba-u.jp

Ryuichi Yoshida and Yoshihito Souma
Sensing Business Headquarter, Konica Minolta, Inc., Osaka, Japan

Chawan Koopipat
Department of Imaging and Printing Technology, Chulalongkorn University, Bangkok, Thailand

Masato Takahashi
Graduate School of Science and Engineering, Chiba University, Chiba, Japan

Norimichi Tsumura
Graduate School of Science and Engineering, Chiba University, Chiba, Japan
Hiroshima University, Hiroshima University Hospital, Hiroshima, Japan

Abstract. *In recent years, the advances in technology for detecting paint defects on exterior surfaces of automobiles have led to the emergence of research on automatic classification of defect types using deep learning. To develop a deep-learning model capable of identifying defect types, a large dataset consisting of sequential images of paint defects captured during inspection is required. However, generating such a dataset for each factory using actual measurements is expensive. Therefore, we propose a method for generating datasets to train deep-learning models in each factory by simulating images using computer graphics.*
[DOI: 10.2352/J.ImagingSci.Technol.2023.67.5.050412]

1. INTRODUCTION

In the automobile industry, a range of defects may occur on automobile exterior surfaces during the external painting process. Such defects may include convex defects stemming from painting on areas where iron powder or dust adheres to the automobile exterior surfaces, and concave defects caused by oil or silicone adhering to the exterior surfaces and repelling the paint [1], as shown in Figure 1. Since defect detection requires significant visual concentration and expertise, efforts are being made to automate this process. In a previous study, Lou and Huang [2] advocated for a proactive approach to quality control utilizing artificial intelligence and engineering principles for addressing complex and uncertain manufacturing processes. They prioritized on-process inspection over post-process inspection and employed hierarchical decision making with fuzzy MIN-MAX algorithms and optimization to evaluate process performance and prevent defects in automotive topcoat applications. Meanwhile, Tanaka et al. [3] concentrated on identifying concave and convex defects on painted surfaces and developed a technique that employs a surface light source and video camera to detect defects and discriminate unevenness. In recent times, research has focused on developing a tunnel-type inspection system [4] (Figure 2) that autonomously detects paint defects, as well as enabling deep-learning-based classification of defect types. Training a deep-learning model for this purpose requires a large dataset consisting of images of parts with paint defects from an automobile exterior that has passed through the inspection system (hereafter referred to as “paint defect images”), as depicted in Figure 3. However, generating such a dataset by capturing images and building individual training models for each customer are expensive and time-consuming processes.

This study proposed a method for generating a learning dataset for each factory by simulating image generation using computer graphics (CG). Figure 4 shows the concept of the proposed method. The shape data of automobile exterior surface paint defects as well as the condition of the inspected object, equipment, and optical arrangement of the factory were simulated to create a CG object (Object A) and the environment (Factory A) that accurately reproduced the

[†]Equally contributed.

Received May 26, 2023; accepted for publication Aug. 29, 2023; published online Oct. 5, 2023. Associate Editor: Peter Nussbaum.
1062-3701/2023/67(5)/050412/10/$25.00

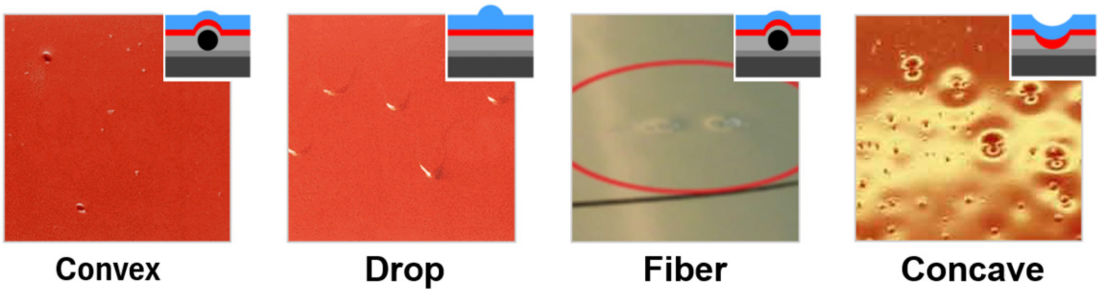

Figure 1. Schematic and image of paint defects.

Figure 2. In-line tunnel-type paint defects inspection system esφi [4].

defects of the painted automobile exterior surface observed in that factory. The simulation generated a dataset (Dataset A) of images of automobile exterior surface paint defects found in the factory. However, owing to differences in the conditions of the test objects and equipment, a deep-learning model trained on Dataset A cannot accurately classify measured data of defect images from a different factory (Test set B). Therefore, a virtual dataset (Dataset A in B) can be created by generating a reproduction environment of a different factory (Factory B) using CG software and simulating it with Object A' modified for the different factory test set conditions. The deep-learning model trained on Dataset A in B is expected to classify Test set B with higher accuracy.

The efficacy of this method was evaluated by generating two datasets, Dataset A and Dataset A in B, and comparing the classification accuracy in Test set B of deep-learning models trained on Dataset A and Dataset A in B.

2. RELATED WORKS

2.1 *Dataset Creation for Deep Learning with CG*

In recent years, with research using deep learning becoming popular, many methods for creating training data sets using synthetic data have been proposed [5]. In this section, we introduce some examples. An example of a dataset created using CG is a dataset for a deep-learning model to detect a specific object. Rajpura et al. proposed a method to create a dataset with composite images by rendering packaged foods reproduced in Blender in a virtual environment of a refrigerator to detect packaged foods in a refrigerator [6]. Furthermore, O'Byrne et al. proposed a method to generate synthetic images featuring biofouling in various virtual environments to detect biofouling [7] in marine structures [8]. To avoid privacy issues, several studies have also used CG to build synthetic models of humans when creating datasets containing human face images and videos. Queiroz et al. and Dong et al. proposed pipelines for generating synthetic images of faces using CG [9, 10] In addition, Ragheb et al., De Souza et al., and Varol et al. created a large dataset for human action recognition using CG [11–13].

2.2 *Analysis of 3D Data Using 3DCNN*

The 3D convolutional neural network (3DCNN) is a neural network architecture proposed by Ji et al. that has been expanded to support three-dimensional input [14]. By performing three-dimensional convolution in the convolution stage, the 3DCNN can compute features from both spatial and temporal dimensions. The 3D convolution is achieved by convolving the 3D kernel into a cube formed by stacking multiple consecutive frames. With this construction, the feature map of the convolutional layer is connected to the previous layer of consecutive frames, enabling the capture of motion information.

The first purpose of using 3DCNN is anomaly detection. Collins et al. constructed a 3DCNN model that detected colon and esophageal cancer with high accuracy by training on hyperspectral image datasets of mucosal tissue in the intracanal regions of the colon and esophagus of patients [15]. Wang et al. also developed a deep autoencoder network for detecting abnormal behavior in surveillance videos, which combines a 3DCNN that encodes short-term temporal and local spatial information and a ConvGRU [16] that encodes long-term temporal and global spatial information [17]. Other examples include fatigue behavior detection for train drivers [18], fall detection for elderly people in a home environment [19], and foul detection within a basketball game [20]. A method for building 3DCNN models to detect videos that exploit deep-faking techniques, which have become a problem in recent years, has also been proposed by Zhang et al. and Wang et al. [21, 22].

3. REPRODUCTION OF PAINT DEFECT IMAGES

This study used Blender [23], an open-source CG software, for generating images of paint defects and datasets. In this section, we present the methodology for generating these images and datasets using Blender.

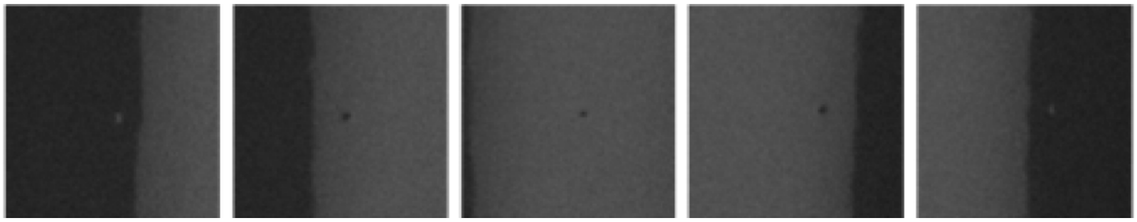

Figure 3. Example of the paint defect image.

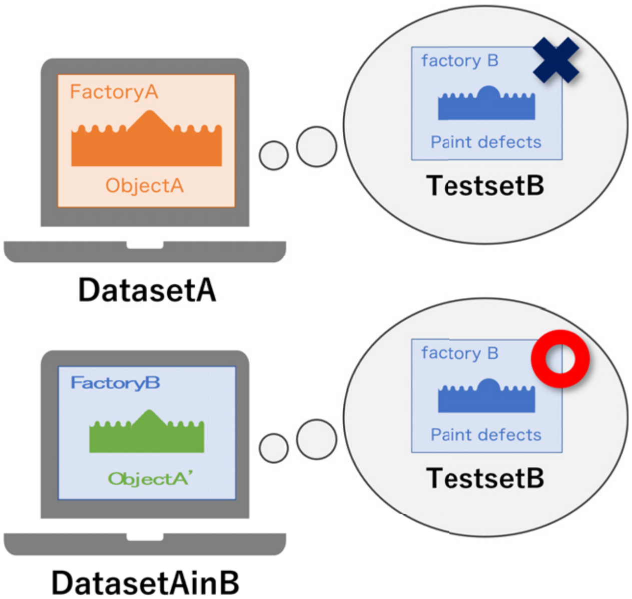

Figure 4. Conceptual diagram of the proposed method.

3.1 *Modeling of Defect Shapes*

In this study, we generated a dataset consisting of four classes: convex defects, concave defects, fiber defects, and no defects. The shape data for convex and concave defects were generated using the model equation for paint defects by Yoshida et al. [24], while whereas the shape data for fiber defects was based on measured data. The model equations for paint defects used to generate the shape data for convex and concave defects is presented below.

$$z = h \cdot e^{-\left(\frac{x^2+y^2}{0.1 \cdot s \cdot d^2}\right)^s}. \quad (1)$$

The parameters of height, diameter, and shape factor are denoted by h, d, and s, respectively. The shape data for convex and concave defects were obtained by generating random values within a predetermined range for each factory. The 3D model geometry resulting from the Eq. (1) computation is shown in Figure 5.

3.2 *Reproduction of Paint Defects in Factory A*

Figure 6 shows the process for reproducing paint defects. First, as depicted in Figure 7, the surface point-cloud data of a paint defect was generated by adding the shape data of the defect at Factory A to the measured shape data of the orange

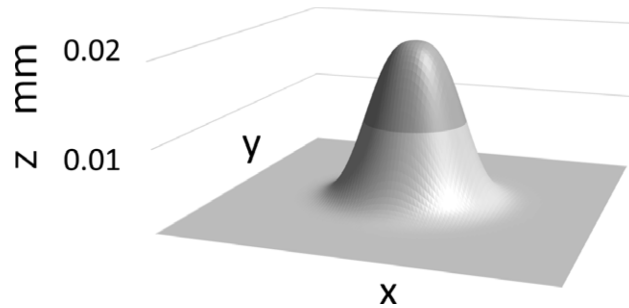

Figure 5. 3D shape of paint defects calculated using the model equation.

peel, which are minute irregularities appearing on the paint surface. Subsequently, MeshLab [25] software was employed to mesh the point-cloud data by establishing connections between lines and planes. The mesh data of the defect surface were then imported into Blender, where random curvature was introduced to the mesh data to produce variations in the reflection of the light from the inspection device source in the paint defect images. Finally, Object A was created by specifying the material (reflective properties).

3.3 *Reproduction of the Environment of Factory A*

The creation of Factory A involved the construction of a virtual environment based on the optical arrangement and conditions present in the real-world Factory A. As depicted in Figure 8, this involved the placement of a camera and an arch-shaped light source to simulate the real-world environment. Additionally, Object A, generated using the methodology outlined in Section 3.1, was placed within Factory A and given horizontal motion animation. A 101 $\times$ 101 $\times$ 33 (frame) image of the defect was produced by limiting the render area to the region surrounding the paint defect.

Figure 9 compares the simulated image in Dataset A generated using the above method with an image captured in a real environment. To demonstrate the similarity between the measured and generated images, a subjective evaluation experiment was conducted where 9 students evaluated three types of automotive paint defects: concave, convex, and fiber. The students were presented with 10 simulated videos generated from the measured videos of concave, convex, and fiber defects, respectively, and asked to evaluate each defect individually. A five-level evaluation index was used in the evaluation process as shown in Figure 10, with higher numbers indicating greater similarity. The evaluation

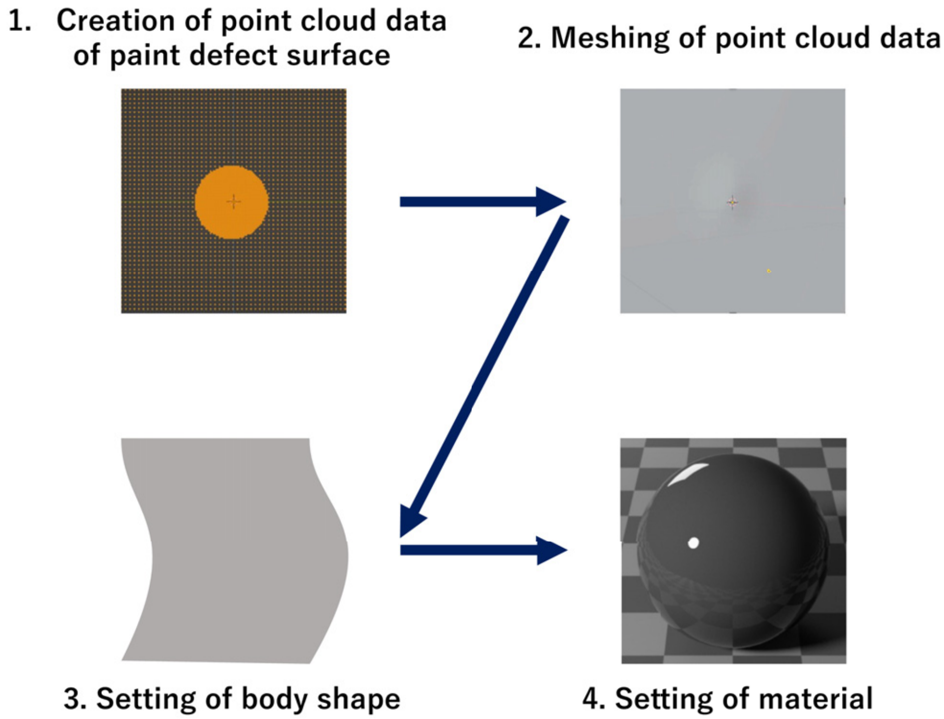

Figure 6. Procedure for reproducing paint defects.

Figure 7. Conceptual diagram of the procedure for generating a paint defect surface.

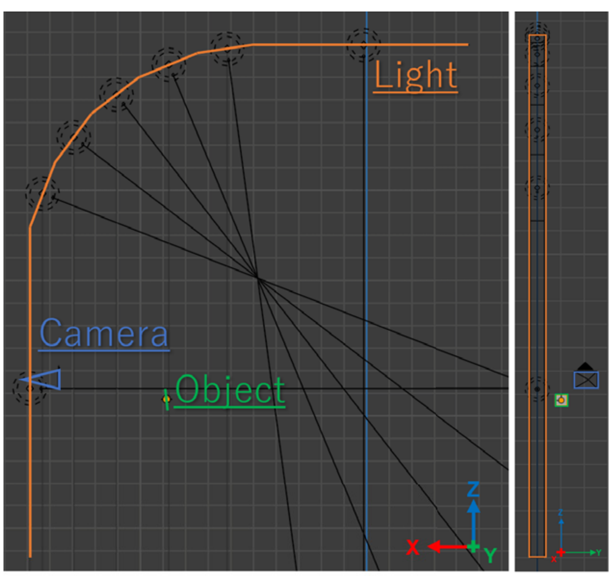

Figure 8. Optical arrangement.

results for concave, convex, and fiber defects are shown in Figures 11–13. On the vertical axis is the evaluation value and on the horizontal axis is the number of the presented defect movies. The average mean evaluation scores for concave, convex, and fiber defects were 3.36, 3.35, and 3.25 points, respectively. The particularly low evaluation score for fiber defects is thought to be due to the variation in the shape of the fibers falling on the surface of the coating and the differences between factories.

Although the student evaluation results were not ideal based on the average from the questionnaire, the fact that the evaluations by experts showed a remarkable similarity to the images captured confirmed the validity of the generated images.

3.4 *Change of Environmental Conditions to Factory B*

Table I shows the differences in the equipment and test object conditions at each factory. As shown in Figure 14, simulations can generate paint defect images with varying appearances by changing these conditions despite possessing similar defect shape data.

The procedure for generating Dataset A in B, a virtual dataset for Factory B, is described as follows. First, the shape data of paint defects used to generate Dataset A was prepared. Subsequently, the defect shape data of Factory A was used to generate a defect surface CG object; Object A' that reproduces the test object conditions of Factory B. Next, Factory B,

Table I. Differences between factories.

Category	Item	Factory A
Equipment	Focal length (mm)	9, 12.5, 16
	F number	7
	Number of pixels (px)	4097 × 3000
	Pixel sixe (μm)	3.45
	Object distance (mm)	600–1600
	Shooting pitch (mm/frame)	4
Test object	Body shape	Flat surfaces: 70% Curved surfaces: 30%
	Level of orange peel	Good (amplitude 1.4–5.0x)
	Distribution of paint colors	Black and white only
Category	**Item**	**Factory B**
Equipment	Focal length (mm)	16, 25, 35
	F number	7
	Number of pixels (px)	4096 × 2168
	Pixel size (μm)	3.45
	Object disance (mm)	600–1600
	Shooting pitch (mm/frame)	2.5
Test object	Body shape	Flat surfaces: 30% Curved surfaces: 70%
	Level of orange peel	Bad (amplitude 1.4–15.0x)
	Distribution of paint colors	Black and white plus solid colors

which reproduced the equipment conditions of Factory B, was generated. Finally, by placing Object A' in Factory B and executing a simulation, Dataset A in B was generated.

4. COMPARISON OF CLASSIFICATION ACCURACY USING 3DCNN

In this study, we used 3DCNN to compare the classification accuracy of Test set B when trained on two datasets. As there was no Test set B containing captured images in this study, we generated Test set B through simulation using Blender. Table II shows the number of data samples per class included in each dataset.

4.1 *Network Architecture*

The network architecture of the 3DCNN used in this study is depicted in Figure 15. The input comprises a 33-frame sequence of 101 × 101-pixel paint defect images, with a single channel. The input was subjected to a series of operations, including a 3D convolution layer, 3D maximum value pooling layer, and a dropout layer to alleviate overfitting. Subsequently, it went through additional coupling layers, resulting in the prediction of the probabilities of four classes: convex, concave, fiber, and no defects.

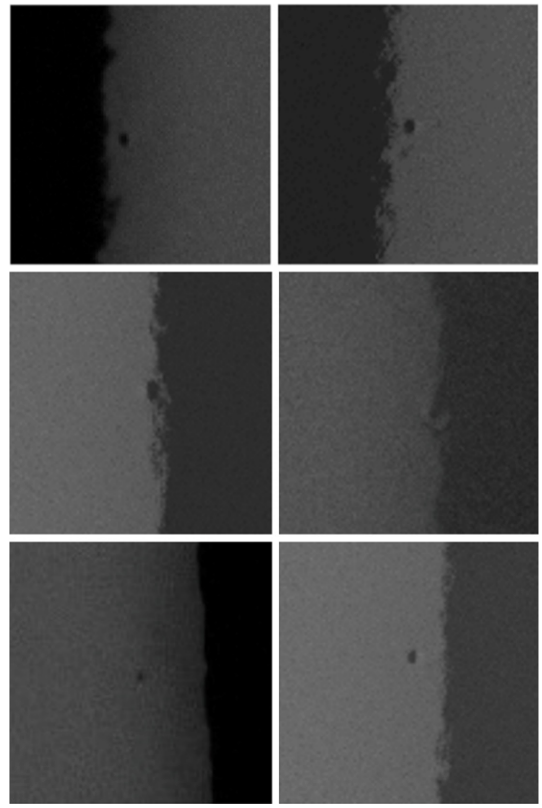

Figure 9. Comparison of captured and simulated images (right: captured, left: simulated).

Table II. Number of data samples for each class in datasets.

Data set A		Data set A in B		Test set B	
Class	Number of data	Class	Number of data	Class	Number of data
Convex	900	Convex	900	Convex	840
Concave	240	Concave	240	Concave	324
Fiber	60	Fiber	60	Fiber	36
No defects	240	No defects	240	No defects	240
Total	1440	Total	1440	Total	1440

4.2 *Training*

The 3DCNN model was trained on Dataset A (Model A) and Dataset A in B (Model A in B), which were divided into 80% for training and 20% for validation according to Pareto's law [26]. The 80/20 split ratio is one of the most common ratios in the deep-learning field [27, 28]. The training was performed for 40 epochs using the hardware environment specified in Table III, with the optimization function set to Adam, learning rate of 0.0001, and batch size of 8. The learning process took approximately 16 minutes. Figure 16 shows the changes in losses during the learning process.

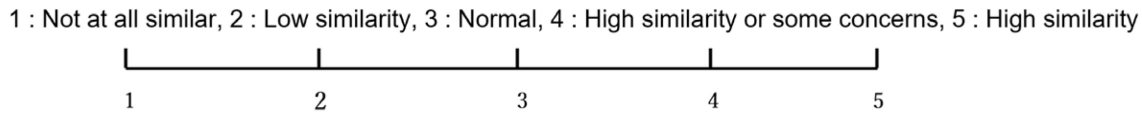

Figure 10. Five-level evaluation index of subjective evaluation experiments.

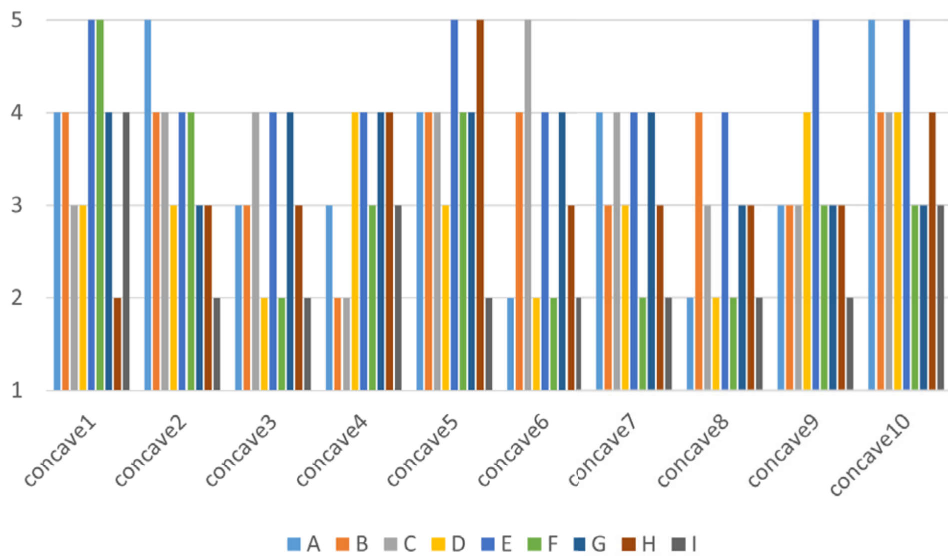

Figure 11. Subjective evaluation results for concave defects.

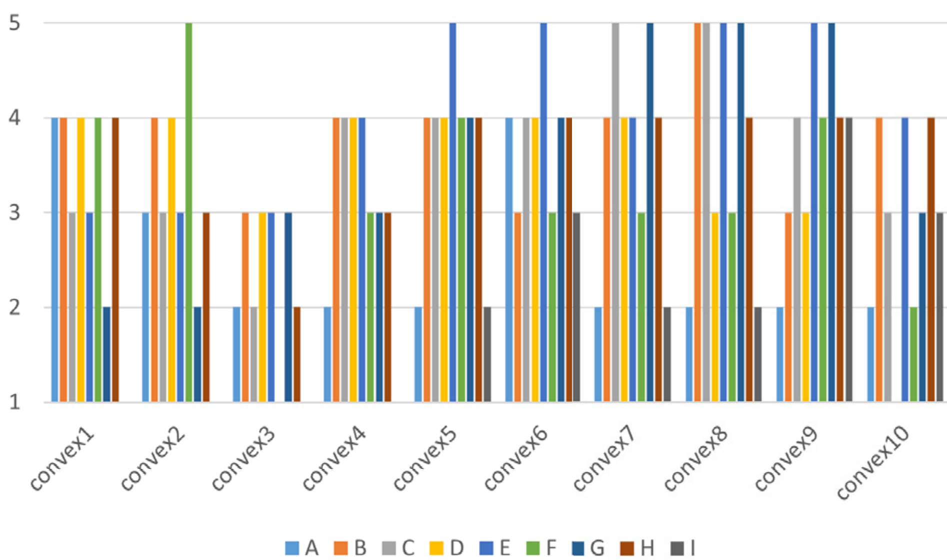

Figure 12. Subjective evaluation results for convex defects.

Table III. Hardware environment.

CPU	Intel® Xeon (R) Silver 4116 CPU @ 2.10 GHz × 48
RAM	93G DDR4
GPU	NVDIA GeForce RTX 2080 Ti × 4, Quadro P400

4.3 *Comparison of Classification Accuracy*

We compared the detection results of both models using four indicators: precision, true positive rate (TPR), F1 score, and accuracy. We further evaluated them using the area under the curve (AUC) indicator derived from the receiver operating characteristic (ROC) curve. AUC measures the classification performance and diagnosis rules that are widely used [29–31].

Table IV shows the confusion matrix, where true positive (TP) means that a positive sample was correctly identified, true negative (TN) means that a negative sample was correctly identified, false positive (FP) means that a negative

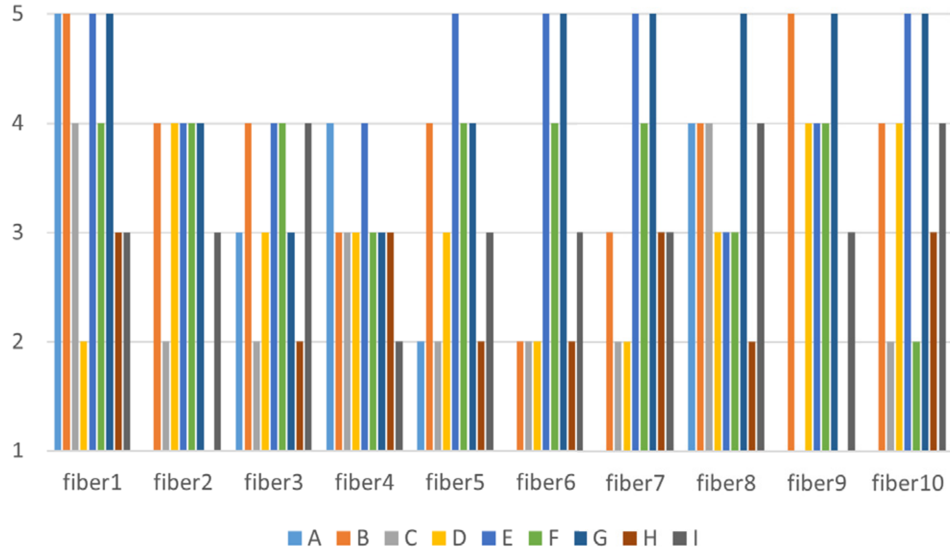

Figure 13. Subjective evaluation results for fiber defects.

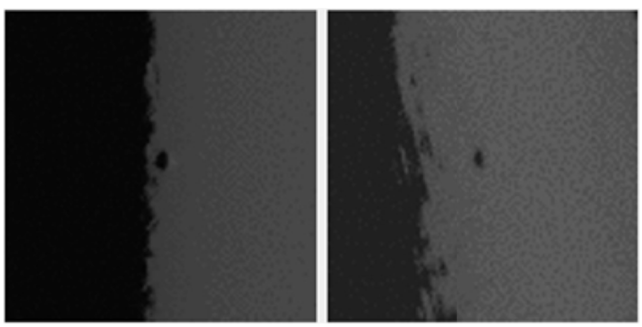

Figure 14. Comparison of simulation results (Right: Factory A, Left: Factory B).

Table IV. Confusion matrix.

		Predicted value	
		Positive	Negative
Correct value	Positive	TP True positive	FN False negative
	Negative	FP False positive	TN True negative

sample was incorrectly identified as positive, and false negative (FN) means that a positive sample was incorrectly identified as negative.

Precision is defined as in Eq. (2) and indicates the percentage of TP samples among the samples identified as positive.

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{2}$$

TPR is defined as in Eq. (3) and indicates the percentage of positive samples in the data that are correctly discriminated.

$$TPR = \frac{TP}{TP + FN}. \tag{3}$$

The precision of *TPR* and repeatability alone do not provide a good assessment of model performance. Therefore, we introduced the F1 score to consider fit and reproducibility together. Its definition is shown in Eq. (4).

$$F1 = 2\frac{\text{Precision} \cdot TPR}{\text{Precision} + TPR}. \tag{4}$$

Accuracy is generally used to evaluate the global accuracy of a model, which contains insufficient information and does not provide a comprehensive evaluation of the performance of the model. It is defined as in Eq. (5).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5}$$

Tables V and VI show the confusion matrices for Model A and Model A in B. Comparing the tables, Model A in B detects concave and convex defects more accurately than Model A. This shows that Model A in B more accurately identifies these defects. To further quantify and evaluate the performance of both models, the accuracy evaluation results from the confusion matrices of both models are presented in Tables VII and VIII. The tables show that Model A in B outperforms Model A in all evaluation indices except *TPR* for fiber defects and no defects. The relatively low *TPR* for fiber defects may be due to the difficult nature of accurately classifying fiber defects, which vary in shape between factories.

Figures 17 and 18 depict the ROC curves obtained from Model A and Model A in B, respectively. The horizontal and vertical axes of the ROC curve represent *TPR* and *FPR*, respectively. *FPR* signifies the percentage of negative samples misclassified as positive. Its definition is shown in Eq. (6).

$$FPR = \frac{FP}{TN + FP}. \tag{6}$$

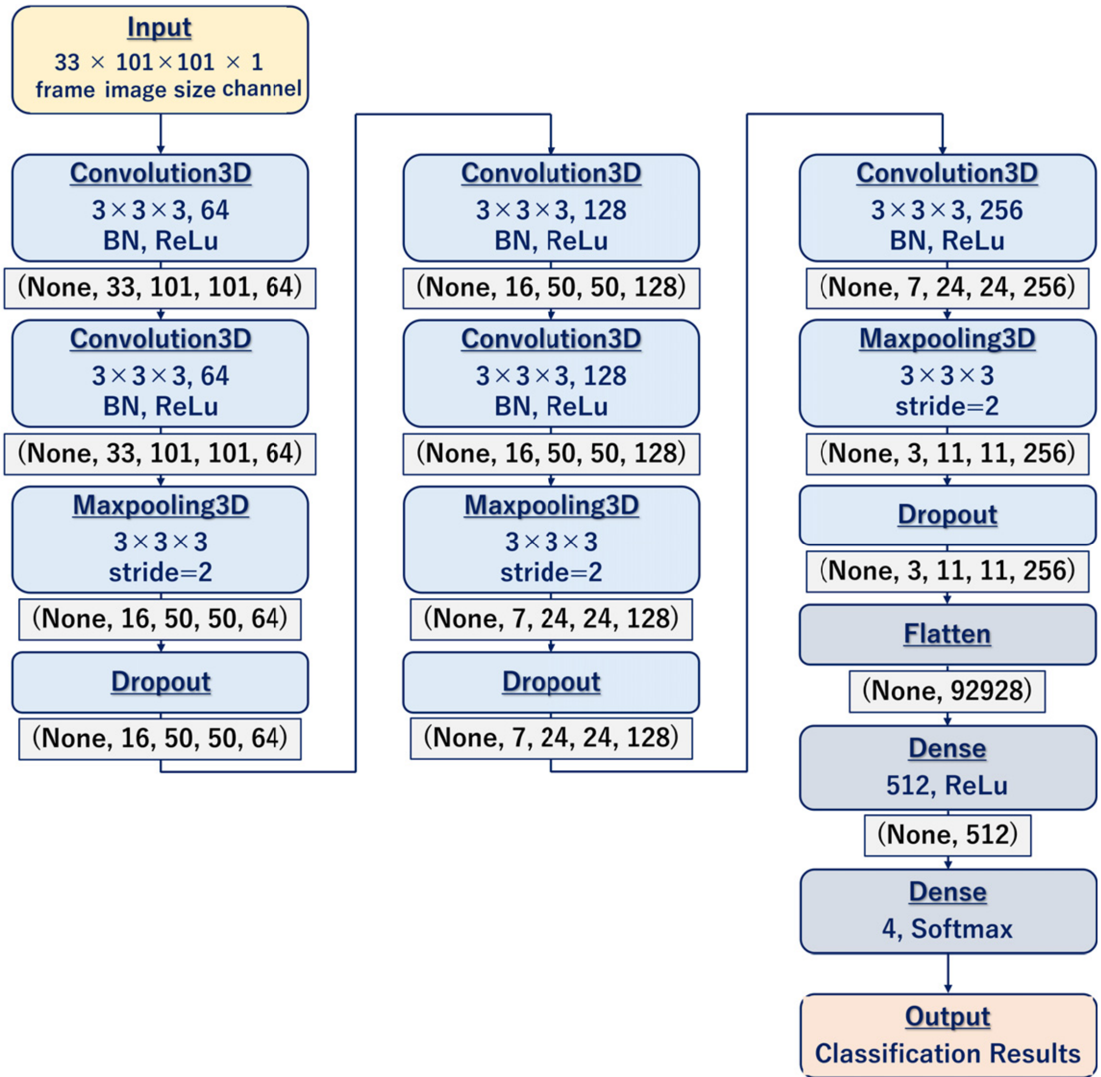

Figure 15. Network architecture.

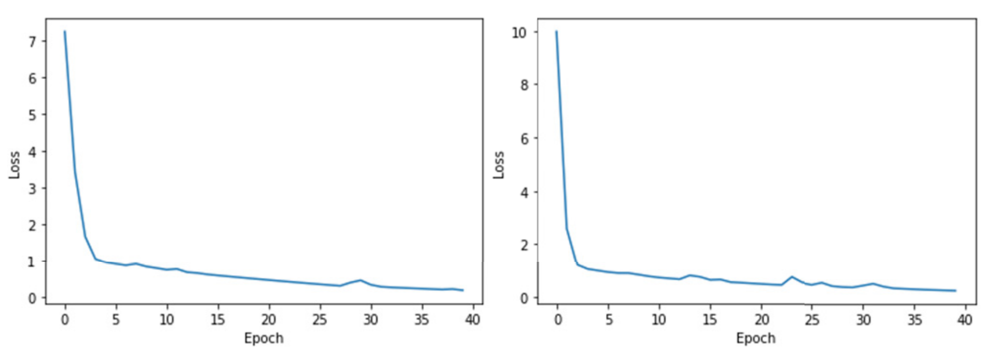

Figure 16. Changes in loss during the learning process (Right: Model A, Left: Model A in B).

The upward curvature in both cases signifies that these models achieved low *FPR* while maintaining high *TPR*, indicating their robust performance. This is indicative of the ability of the models to accurately predict positive samples while minimizing misclassifications of negative samples as positive. Notably, the ROC curve of Model A in B exhibits higher elevation compared to that of Model A, suggesting superior discriminative power. A higher ROC curve elevation signifies that Model A in B achieved a more balanced classification performance, making it better suited for defect detection in Factory B.

The AUC values, as presented in Table IX, further support the performance evaluation. AUC values closer to 1.0 indicate better model performance, close to the

Table V. Confusion matrix of Model A.

Model A					
		Predicted			
		Convex	Concave	Fiber	No defects
True	Convex	751	21	7	61
	Concave	17	256	14	37
	Fiber	1	2	30	3
	No defects	0	0	0	240

Table VI. Confusion matrix of Model A in B.

Model A in B					
		Predicted			
		Convex	Concave	Fiber	No defects
True	Convex	827	5	0	8
	Concave	5	319	0	0
	Fiber	5	3	28	0
	No defects	0	0	0	240

Table VII. Accuracy evaluation results for Model A.

Class	Convex	Concave	Fiber	No defects
Precision	0.977	0.918	0.588	0.704
TPR	0.894	0.790	0.833	1.00
F1 Score	0.933	0.849	0.690	0.826
Accuracy		0.886		

Table VIII. Accuracy evaluation results for Model A in B.

Class	Convex	Concave	Fiber	No defects
Precision	0.988	0.976	1.000	0.968
TPR	0.985	0.985	0.778	1.00
F1 Score	0.986	0.980	0.875	0.984
Accuracy		0.982		

ideal model with perfect predictive ability. Comparing both models, Model A in B outperforms Model A in all defect classifications, demonstrating its superiority. The comparisons collectively corroborate the higher accuracy of Model A in B over Model A, thereby affirming the efficacy of the proposed method.

5. SUMMARY AND FUTURE WORK

In this paper, we presented a novel method for generating factory-specific datasets for paint-defect classification of automotive exteriors. By using Blender to simulate paint defects and the factory environment, our approach eliminates the need for new data collection and thus provides a cost-effective and efficient solution. Experimental results show a significant improvement in test set classification accuracy, supporting the effectiveness of the proposed method. The ability to generate factory-specific datasets considers the variation in paint defects across different factories and improves the performance of the model for real-world applications. To the best of our knowledge, no similar study has used the proposed method, thus proving the novelty and uniqueness of our contribution, as this

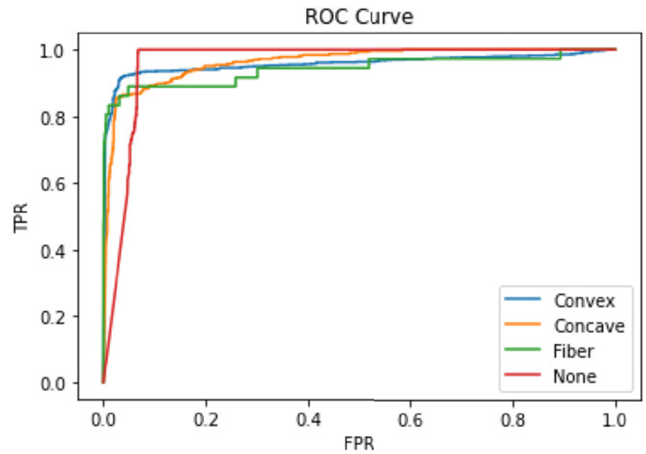

Figure 17. Receiver operating characteristic (ROC) curve of Model A.

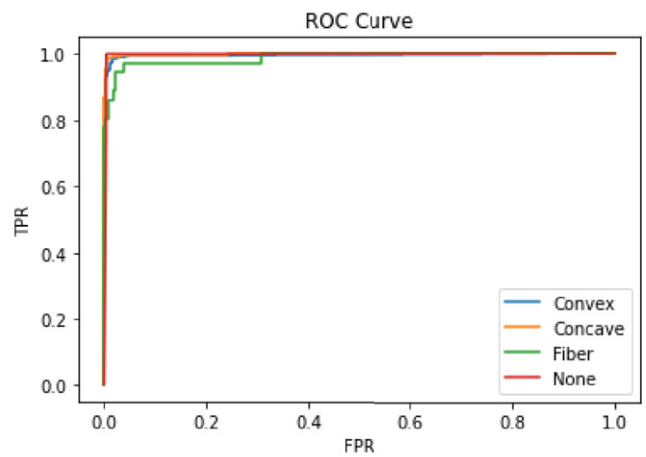

Figure 18. Receiver operating characteristic (ROC) curve of Model A in B.

Table IX. AUC comparison of two models.

Model A		Model A in B	
Class	AUC	Class	AUC
Convex	0.957	Convex	0.995
Concave	0.964	Concave	0.999
Fiber	0.942	Fiber	0.988
No defects	0.961	No defects	0.997

study is pioneering in its approach. Our study lays the foundation to facilitate the practice of defect classification in the automotive industry and optimize the quality control process.

Future work includes superimposing paint defects on the automobile model and mapping measured values to material parameters to generate simulations that more closely resemble inspections under real-world conditions.

REFERENCES

1 Nippon Kako Toryo Co., Ltd., "Nippon Kakoh Paint". (accessed 16 January).
2 H. H. Lou and Y. L. Huang, "Hierarchical decision making for proactive quality control: system development for defect reduction in automotive coating operations," Eng. Appl. Artif. Intell. **16**, 237–250 (2003).
3 K. Tanaka, Y. Shinbara, H. Ikeda, N. Yamada, H. Kiba, and K. Sasanishi, "Automated inspection method for painted surfaces," Trans. Jap. Soc. Mech. Eng. **60**, 866–873 (1994).
4 Konica Minolita Japan, INC., "Inline Tunnel Type Coating Defect Inspection System esφi | KONICA MINOLITA JAPAN, INC.- Powered by IPROS CORPORATION." (accessed 16 January).
5 S. I. Nikolenko, *Synthetic Data for Deep Learning.* (Springer, Cham, 2021), p. 174.
6 P. S. Rajpura, R. S. Hegde, and H. Bojinov, Object Detection Using Deep CNNs Trained on Synthetic Images. arXiv:abs/1706.06782 (2017).
7 T Kazuhisa, "Biofouling in seawater reverse osmosis desalination plant," Bull. Soc. Sea Water Sci., Jpn. **63**, 303–306 (2009).
8 M. O'Byrne, V. Pakrashi, F. Schoefs, and B. Ghosh, "Semantic segmentation of underwater imagery using deep networks trained on synthetic imagery," J. Marine Sci. Eng. **6**, 93 (2018).
9 R. Queiroz, M. Cohen, J. L. Moreira, A. Braun, J. C. Jacques Junior, and S. R. Musse, "Generating facial ground truth with synthetic faces," *2010 23rd SIBGRAPI Conf. on Graphics* (IEEE, Piscataway, NJ, 2010), pp. 25–31.
10 Y. Dong, M. Lin, J. Yue, and L. Shi, "A low-cost photorealistic CG dataset rendering pipeline for facial landmark localization," Multimedia Tools Appl. **78**, 22397–22420 (2019).
11 H. Ragheb, S. Velastin, P. Remagnino, and T. Ellis, "Vihasi: virtual human action silhouette data for the performance evaluation of silhouette-based action recognition methods," *2008 2nd ACM/IEEE Int'l. Conf. on Distributed Smart Cameras* (IEEE, Piscataway, NJ, 2008), pp. 1–10.
12 C. R. De Souza, A. Gaidon, Y. Cabon, and A. M. L'opez, "Procedural generation of videos to train deep action recognition networks," *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2017), pp. 2594–2604.
13 G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2017), pp. 4627–4635.
14 S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional neural networks for human action recognition," IEEE Trans. Pattern Anal. Mach. Intell. **35**, 221–231 (2013).
15 T. Collins, M. Maktabi, M. Barberio, V. Bencteux, B. Jansen-Winkeln, C. Chalopin, J. Marescaux, A. Hostettler, M. Diana, and I. Gockel, "Automatic recognition of colon and esophagogastric cancer with machine learning and hyperspectral imaging," Diagnostics **11**, 1810 (2021).
16 N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *Int'l. Conf. on Learning Representations (ICLR)* (2016), pp. 1–11.
17 X. Wang, W. Xie, and J. Song, "Learning spatiotemporal features with 3DCNN and ConvGRU for video anomaly detection," *2018 14th IEEE Int'l. Conf. on Signal Processing (ICSP)* (IEEE, Piscataway, NJ, 2018), pp. 474–479.
18 Y. Liu, T. Zhang, and Z. Li, "3DCNN-based real-time driver fatigue behavior detection in urban rail transit," IEEE Access **7**, 144648–144662 (2019).
19 Y. Z. Hsieh and Y. L. Jeng, "Development of home intelligent fall detection IoT system based on feedback optical flow convolutional neural network," IEEE Access **6**, 6048–6057 (2018).
20 C. H. Lin, M. Y. Tsai, and P. Y. Chou, "A lightweight fine-grained action recognition network for basketball foul detection," *2021 IEEE Int'l. Conf. on Consumer Electronics-Taiwan (ICCE-TW)* (IEEE, Piscataway, NJ, 2021), pp. 1–2.
21 D. Zhang, C. Li, F. Lin, D. Zeng, and S. Ge, "Detecting deepfake videos with temporal dropout 3DCNN," *Proc. 13th Int'l. Joint Conf. on Artificial Intelligence (IJCAI-21)* (IEEE, Piscataway, NJ, 2021), pp. 1288–1294.
22 Y. Wang and A. Dantcheva, "A video is worth more than 1000 lies. Comparing 3DCNN approaches for detecting deepfakes," *2020 15th IEEE Int'l. Conf. on Automatic Face and Gesture Recognition (FG 2020)* (IEEE, Piscataway, NJ, 2020), pp. 515–519.
23 Blender. Blender. https://www.blender.org/. (accessed 16 January).
24 R. Yoshida, Y. Nagai, and M. Kashiwabara, "A study of "scales" and "defect samples" for coating defect inspection," *37th Annual Conf. on Paint and Coating Research* (2022).
25 MeshLab. MeshLab. https://www.meshlab.net/. (accessed 16 January).
26 R. Dunford., Q. Su, and E. Tamang, "The pareto principle," Plymouth Stud. Sci. **7**, 140–148 (2014).
27 G. Jain, D. Mittal, D. Takur, and M. K. Mittal, "A deep learning approach to detect COVID-19 coronavirus with X-ray images," Biocybern. Biomed. Eng. **40**, 1391–1405 (2020).
28 S. Abdelaziz Ismael, A. Mohammed, and H. Hefny, "An enhanced deep learning approach for brain cancer MRI images classification using residual networks," Artif. Intell. Med. **102**, 101779 (2020).
29 A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," Pattern Recognit. **30**, 1145–1159 (1997).
30 D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," Mach. Learn. **45**, 171–186 (2001).
31 J. A. Hanley, "Receiver operating characteristic (ROC) methodology: the state of the art," Crit. Rev. Diagn Imag. **29**, 307–335 (1989).