ICCEPM 2024 https://dx.doi.org/10.6106/ICCEPM.2024.0311
The 10th International Conference on Construction Engineering and Project Management
Jul. 29-Aug.1, 2024, Sapporo

HYBRID DATA SET GENERATION METHOD FOR COMPUTER VISION-BASED DEFECT DETECTION IN BUILDING CONSTRUCTION

Seung-mo Choi[1], Heesung Cha[2], Bo-sik, Son[3]

[1] *Department of Architectural Engineering, Ajou University, Suwon, Republic of Korea,, E-mail address: mo126713@ajou.ac.kr*
[2] *Department of Architectural Engineering, Ajou University, Suwon, Republic of Korea,, E-mail address:hscha@ajou.ac.kr*
[3] *Department of Architectural Engineering, Namseoul University, Cheonan, Republic of Korea,, E-mail address:bsson@nsu.ac.kr*

Abstract: Quality control in construction projects necessitates the detection of defects during construction. Currently, this task is performed manually by site supervisors. This manual process is inefficient, labor-intensive, and prone to human error, potentially leading to decreased productivity. To address this issue, research has been conducted to automate defect detection using computer vision-based object detection technologies. However, these studies often suffer from a lack of data for training deep learning models, resulting in inadequate accuracy.

This study proposes a method to improve the accuracy of deep learning models through the use of virtual image data. The target building is created as a 3D model and finished with materials similar to actual components. Subsequently, a virtual defect texture is produced by layering three types of images: defect information, area information, and material information images, to fabricate materials with defects. Images are generated by rendering the 3D model and the defect, and annotations are created for segmentation. This approach creates a hybrid dataset by combining virtual data with actual site image data, which is then used to train the deep learning model. This research was conducted on the tile process of finishing construction projects, focusing on cracks and falls as the target defects. The training results of the deep learning model show that the F1-Score increased by 12.08% for falls and cracks when using the hybrid dataset compared to the real image dataset alone, validating the hybrid data approach. This study contributes not only to unmanned and automated smart construction management but also to enhancing safety on construction sites. To establish an integrated smart quality management system, it is necessary to detect various defects simultaneously with high accuracy. Utilizing this method for automatic defect detection in other types of construction can potentially expand the possibilities for implementing an integrated smart quality management system.

Key words: Virtual data, Defect detection, Computer vision, Object detection, Construction quality management

1. INTRODUCTION

1.1 Background

The construction industry is facing a continuous decline in its labor force, which impacts the quality of construction projects. To mitigate this, research into automation in construction and management has been conducted, recognized as an efficient solution to replace the labor workforce [1]. Among the most promising fields is automation through computer vision [2]. Quality control in construction, especially construction quality management, requires significant labor and is prone to errors due to subjective judgments, making image analysis techniques based on computer vision and deep learning hold much potential [2-4]. However, the accuracy limitations of these technologies have been pointed out, hindering their adoption [5].

Image-based object detection technologies rely on the image data used for training to detect objects. The nature of these object detection technologies requires a diversity and quantity of data to achieve high accuracy [6,7]. However, due to the characteristics of the construction industry, it is challenging to collect defect data under various conditions, leading to an unresolved issue of data scarcity [5].

Therefore, the purpose of this study is to explore the foundational use of virtual image data to address the shortage of actual image data, by generating virtual images through 3D modeling software for use as data. Therefore, this study aims to examine the feasibility of applying a hybrid dataset, combining both actual and virtual image data, for the detection of defects in tile construction.

Before proceeding with the study, the type of construction work to be targeted was selected. Finishing works account for 51.62% of the defects in apartment building constructions, with tile defects alone constituting 19.15%, representing a significantly large portion [8]. Thus, tile work was chosen as the targeted construction work, and detachment and cracking were selected as the defect types for this study.

1.2 Research Flow

This study is conducted in four stages as shown in Figure 4: image collection and analysis, virtual 3D model creation, image data generation and processing, and model training and performance comparison.

The actual image collection and analysis stage involves collecting images of defects that have occurred on-site and analyzing their shapes to inform the creation of virtual models. Following this, the 3D model creation stage involves designing tiles with various defect types and creating walls containing these tiles to generate 3D models. These models are then rendered to produce images. To make the generated images suitable for deep learning model training, a labeling process is carried out. The datasets created through these stages are trained under the same conditions using a deep learning model, and the results are compared in the model training and performance comparison stage.

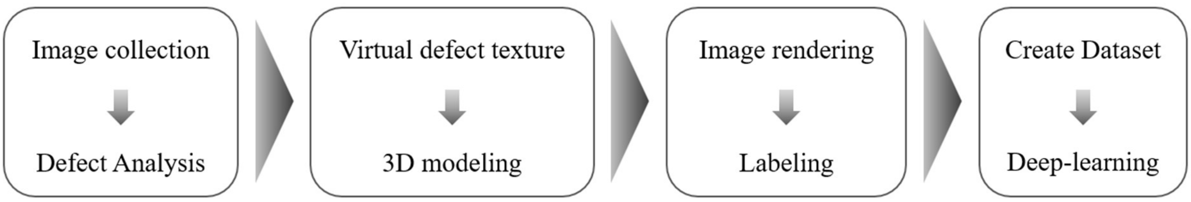

Figure 1. Research flow

2. Literature Reveiw

To alleviate the data scarcity issue in computer vision (CV) deep learning models, research has been undertaken to generate new data. The objectives of CV are varied, including scene understanding, safety monitoring, and defect detection, among others. Moreover, the methodologies for generating new data exhibit differences: there are approaches that generate new data from existing data through Generative Adversarial Networks (GANs), methods that extract images from Building Information Modeling (BIM) models and create data through style transfer techniques, and strategies that produce data from virtual environments [9-13].

Hong utilized BIM models for generating data for deep learning models aimed at scene understanding for buildings and bridges [14]. This study extracted necessary scenes from the BIM models and rendered the images more realistic through Cycle-GAN. Additionally, leveraging the advantages of BIM, automatic annotations were generated by assigning unique colors to each element, enabling the Mask R-CNN model to achieve an average accuracy of 71.6% for buildings and 84.9% for bridges, respectively.

Lee generated data for models used for on-site safety monitoring through a game engine [15]. Safety equipment used on construction sites was created in a virtual environment and synthesized with actual site photos for model training. Subsequently, a hybrid dataset was formed, leading to a performance improvement of up to 30.4%.

Siu created images for a model designed to detect cracks inside sewer pipes by synthesizing images that closely resemble real-world defects in a virtual environment with real images using style transfer [16]. The head of the deep learning model was also modified to a contrastive head, achieving a 7.7% improvement in detection performance.

All these studies have successfully generated virtual data to enhance accuracy. However, unlike previous research, this study focuses on generating defects for which actual field data could not be obtained, aiming to create them more conveniently in a virtual environment. Instead of style transfer, this research efficiently proceeded with rendering image data realism using the native rendering system of 3D software.

3. Methodology

3.1 Real image collection

For this study, real images were collected using image crawling based on the Python language, utilizing the Requests and BeautifulSoup libraries. Google Images was used as the search engine for this process, and the environment was set up in VS Code. The search terms used for crawling included "Cracked tile", "Broken tile", "Fallen tile", "Tile defect". Through this process, a total of 1,000 images were collected.

The images collected via image crawling were randomly gathered, which means some were irrelevant to the learning or duplicates. Thus, duplicates were removed, and images unrelated to the study were also excluded. Since this research targets defects occurring during construction rather than defects in a single product, the collection focused on images of tile walls. Consequently, only images containing at least six tiles were selected. Image-based analysis technologies dissect and analyze images at the pixel level, learning the characteristics of the target object for detection. Therefore, images with low resolution were eliminated to avoid training errors. Additionally, images containing text or frames, which could interfere with the training process, were either edited to remove these elements or excluded if such editing was impractical. Through this filtering process, a total of 307 images were collected for the study.

3.2 3D modeling

3.2.1 Tile texture

To create virtual 3D models suitable for the objectives of this study, the process begins with the creation of virtual tiles. The software used for creating 3D models is Blender. This program allows the use of various 3D assets or textures available online, but acquiring textures or assets that include specific defects such as cracks or fall-offs is challenging due to limited availability. Therefore, the production was carried out through image synthesis. This phase is conducted in three main steps.

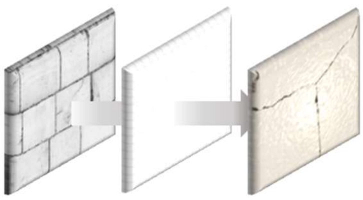

Figure 2a. Image layering

Figure 2b. Tile with defects

Figure 2. Defect texture creation

The first step involves the synthesis of defect information images and material information images, combining images of other materials with defects or images of similar forms with the texture of tile material as shown in Figure 2a. In the second step, areas without defects are removed from the defect image placed under the tile texture, and an area information image is created to remove the defect areas from the tile material image. This is done by creating a black background image and painting the necessary defect areas in white, which is then combined. This task is performed through the connection of each image's nodes in Shading, resulting in the tile image being partially erased and the underlying defect image appearing on the tile texture. Figure 2b is the result of combining images.

The third step involves applying a sense of depth to the images during rendering to mimic the effects that change according to the angle and position of each material in reality. This involves applying a sense of depth to both the tile and the defect separately. In Shading, this is achieved by linking the Normal Map and Displacement with the previously conducted node connections, giving each their own sense of depth.

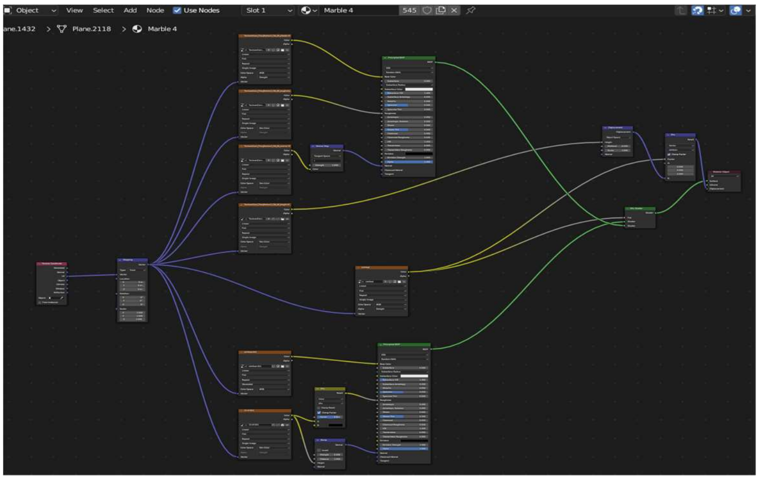

Figure 3. Blender node connention

3.2.2 3D Modeling

Through these processes, defect tile textures are created, allowing the model to learn scenarios where defects must be identified on a wall, similar to an actual construction site. This is achieved by combining defective tiles with standard tiles to construct walls as shown in Figure 4a. This task was applied to tiles of various materials and shapes to enhance diversity. Subsequently, these constructed tile walls are used to create 3D models. Considering that tiles are commonly used in bathrooms and living rooms in residential buildings, they were placed in similar locations within the 3D models. Walls of identical tile materials with different defects were arranged, and four 3D models were positioned in all directions with Sun Light placed in the center to add lighting conditions as shown in Figure 4b. Through this method, a total of 28 models were created using tiles of seven different materials.

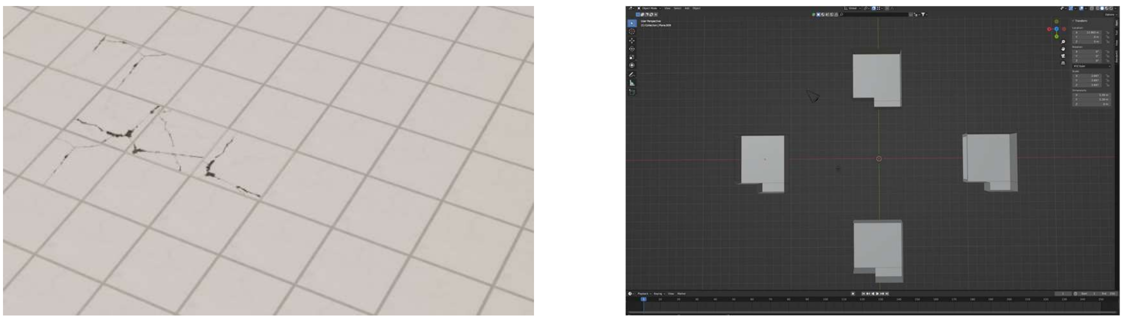

Figure 4a. Tile wall with defects　　　Figure 4b. Model positioning

Figure 4. Tile wall and model

3.3 Image data generation

3.3.1 Image rendering

For image generation, the rendering of the 3D model is conducted. When rendering, the position and height of the camera used in Blender are set, considering the shooting positions during actual quality inspections. In the research of automation of quality control through CV, the devices used for defect detection include UAVs such as drones and robots like Spot for monitoring [17-20]. These two types differ in height; UAVs shoot from a relatively high position, while robots like Spot shoot from a lower height. The shooting angle considered shooting from the horizontal plane of the target surface. Based on this, three cameras were installed at heights of 0.5m and 1.7m, totaling six cameras, and considering manual inspections, an additional camera was placed at the center of the wall, making a total of seven cameras as shown in Figure 5a, 5b. As a result, a total of 28 images are rendered for one material type. Through this process, a total of 784 images were generated.

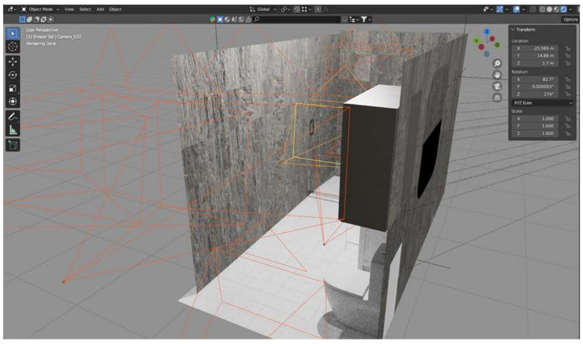

Figure 5a. Camera displacement example 1

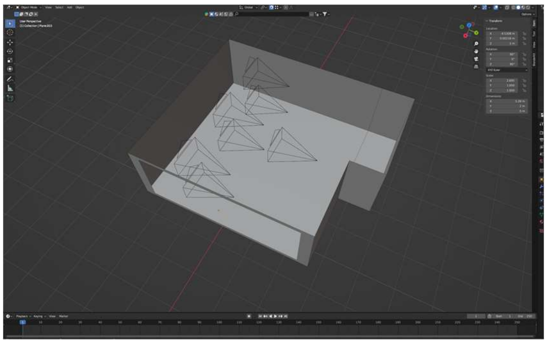

Figure 5b. Camera displacement example 2

Figure 5. Camera displacement

3.3.2 Image labeling

For object detection conducted through deep learning models, it's necessary to create annotation files through labeling images to transform them into a trainable format. Labeling identifies the target object on the image, informing the deep learning model about the location of the object it needs to learn. In this study, segmentation-based labeling was carried out. Unlike the general labeling method of bounding boxes, segmentation precisely marks the target object with polygon-shaped lines, distinguishing it from other objects. Tile defects were labeled using segmentation to avoid the error of recognizing grout lines as cracks or detachments with bounding boxes, specifically using instance segmentation to individually distinguish each defect as shown in Figure 6. Labels were classified into three categories: tile, crack, and detachment, creating annotation files for each image file.

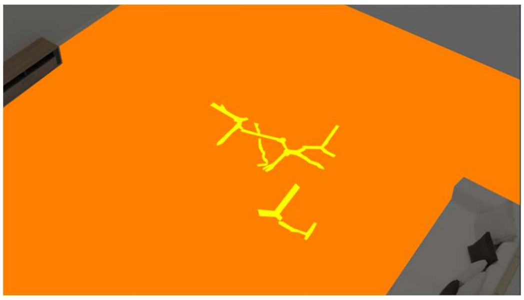

Figure 6a. Labeling layer

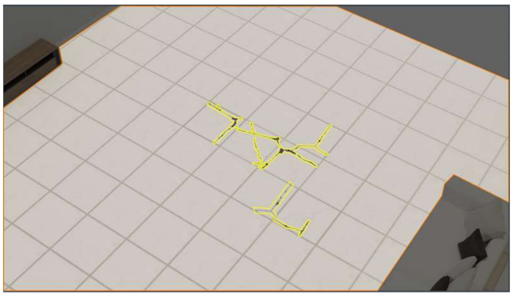

Figure 6b. Labeling result

Figure 6. Segmentation labeling

4. Dataset creation and Deep-learnging experiment

4.1 Dataset creation

To enhance accuracy for training deep learning models aimed at object detection, augmentation is applied to ensure the diversity of training data. Augmentation enriches the dataset's variety and increases the quantity of data by introducing changes such as color variation, exposure adjustment, blurring, or rotation to the original image data. In this study, augmentation was performed to secure data diversity, utilizing techniques such as Rotate, Shear, Crop, Brightness, MixUp, CutMix, Flip, GrayScale, Saturation, and GaussianBlur. This process was applied to both real and virtual image data, generating 2100 real image data and 3300 virtual image data.

Subsequent analyses were conducted to verify the efficacy of virtual image data. This research aimed to assess the performance of learning outcomes using virtual image data; hence, a comparison was made between hybrid datasets that combined real and virtual images and datasets comprising only real images. Additionally, to evaluate the performance of virtual image data as standalone datasets, learning outcomes from virtual image datasets were also compared. Consequently, the generated image data was classified into three categories for deep learning training.

Table 1. Dataset

Dataset	Number of data
Real Image Data	2100
Virtual Image Data	3300
Hybrid Image Data	5400

The deep learning model used for training is the Yolov8 model for Object Classification and Detection, specifically the Yolov8x-seg model for Segmentation training. The training conditions were as follows: OS: Linux, Open source Platform: Anaconda Jupyter, GPU: RTX3090, Software: Pytorch, Model: Yolov8x-seg, Epochs: 400, Patience: 50. The comparison of training results was conducted through F1-score - Confidence graphs. The F1-score is obtained through the harmonic mean of Precision and Recall values, while Confidence represents the model's reliability. In other words, it indicates how confident the model is in its predictions.

4.2 Experimental analysis

The results of the training are shown in the following Figure 7. In the real image dataset grph, the blue graph represents Tile, the orange graph represents Crack, and the red graph represents Fall. The green graph is an exception, including objects such as toilets, sofas, and windows, which are excluded from the target objects.

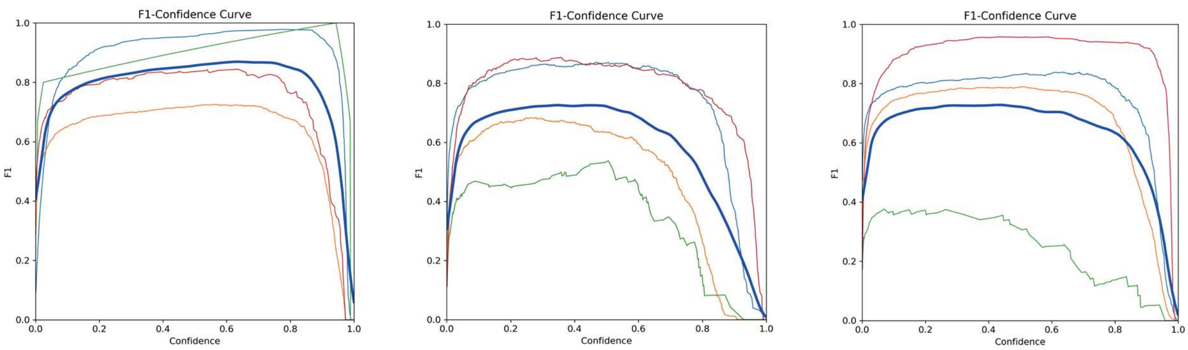

Figure 7a. Real image dataset Figure 7b. Virtual image dataset Figure 7c. Hybrid dataset

Figure 7. F-1 score of each dataset

The thick blue line represents the overall average. From this, we can see that the detection ability for tiles is generally very high, with Fall showing an F1 score of about 0.8, and Crack being just under 0.7.

For the virtual image dataset and the hybrid dataset, the red graph represents Tile, the blue represents Fall, and the orange represents Crack, with the green graph being identical to the real image dataset. The virtual image dataset shows that, except for Fall, the performance in detecting other objects lags behind the real data. However, the hybrid dataset shows an overall improvement in detection performance compared to the real dataset.

Figure 8 shows the Comparison results between each class. V0 represents real image data set, V1 represents virtual image data set, and V2 represents hybrid data set. Comparing the results for each class, the maximum F1 scores for Tile class learning results are similar between using only real images and the hybrid dataset. However, the learning results through the Hybrid dataset record higher F1 scores across a broader range of Confidence values. This implies that while the Tile class is included in all images in the real image dataset, thus showing high learning results, supplementing this with virtual image data allows for learning with image data that includes diversity, achieving high F1 scores even at wide Confidence levels.

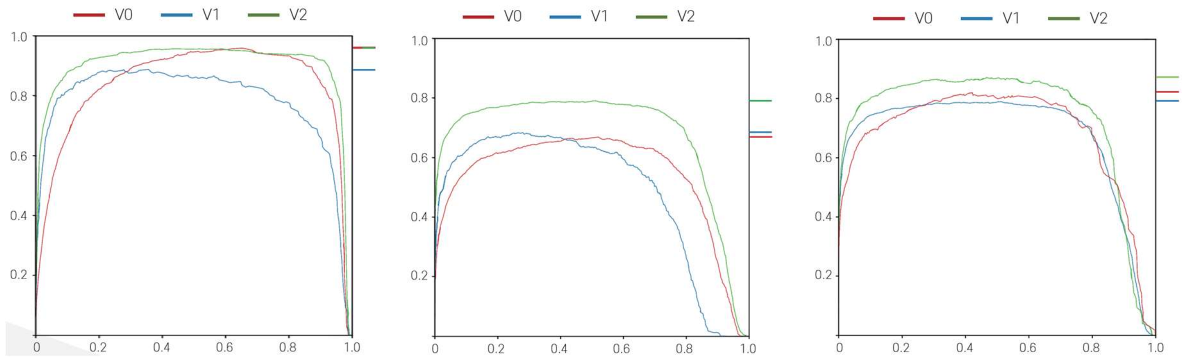

Figure 8a. Tile class   Figure 8b. Crack class   Figure 8c. Fall class

Figure 8. F-1 score comparison between each class

Comparing the learning results for the Crack class, the Hybrid dataset's learning results were about 10% higher in F1 score than the other datasets. For the Fall class, the results were 8% higher than the real image dataset and 5% higher than the virtual image dataset. This proves the effectiveness of the Hybrid dataset with sufficient figures.

5. Conclusion

In the current construction industry, the application of image analysis-based technology for construction quality management is rare. This scarcity is due to the technology's lack of accuracy, making practical application challenging. Nevertheless, efforts to apply this technology in real-world settings continue, driven by the significant benefits that digitalization and automation can bring to tasks. This study has validated the improvement in deep learning model accuracy through the creation of virtual image data and proposed a methodology for generating such images. The defect detection model using the Hybrid dataset proposed in this study showed superior results in all three classes: tile, crack, and fall, compared to models using only real image data or only virtual image data, as evidenced by F1-score comparisons.

This research used a relatively small set of 307 original real images. Increasing the number of real images would likely change the learning outcomes, as real images are considered the best data for training. Therefore, for practical field application, it seems ideal to use real image data as the main dataset and virtual data as supplementary for model training. Nonetheless, this research is meaningful as it contributes to improving the accuracy of model learning outcomes when real images are scarce.

The method proposed in this study can be applied not only to tile defects but also to other types of construction work. By creating defect textures in the same manner and applying them to 3D models, it is possible to generate image data for desired defects. This demonstrates that the methodology can be significantly used in developing a single model for detecting defects across all types of work through the construction of an integrated smart quality management system. Furthermore, this research contributes to expanding the spectrum of data used in deep learning by verifying that images created through 3D modeling tools can also improve model accuracy.

ACKNOWLEGEMENTS

This work was supported by the Ajou University Research Fund.

REFERENCES

[1] D. Hui, H. Hao, L. Dehuan, D. Yichuan, S. Cheng, Automatic Indoor Construction Process Monitoring for Tiles Based on BIM and Computer Vision, J. Constr. Eng. Manage. 146 (2020) 04019095.
[2] P. Martinez, M. Al-Hussein, R. Ahmad, A scientometric analysis and critical review of computer vision applications for construction, Autom. Constr. 107 (2019) 102947.
[3] H. Perez, J.H. Tah, Deep learning smartphone application for real-time detection of defects in buildings, Structural Control and Health Monitoring. 28 (2021) e2751.
[4] D. Thikra, Z. Zhenhua, Z. Tarek, Computer Vision–Based Model for Moisture Marks Detection and Recognition in Subway Networks, J. Comput. Civ. Eng. 32 (2018) 04017079.
[5] H. Luo, L. Lin, K. Chen, M.F. Antwi-Afari, L. Chen, Digital technology for quality management in construction: A review and future research directions, Developments in the Built Environment. 12 (2022) 100087.
[6] Z. Gong, P. Zhong, W. Hu, Diversity in machine learning, Ieee Access. 7 (2019) 64323-64350.
[7] F.E. Nowruzi, P. Kapoor, D. Kolhatkar, F.A. Hassanat, R. Laganiere, J. Rebut, How much real data do we actually need: Analyzing object detection performance using synthetic and real data, arXiv preprint arXiv:1907.07061. (2019).
[8] D. Kim, The priority analysis of defect type through tenant preliminary research in apartment house, Master's Thesis, Korea University. (2016).
[9] H. Ying, R. Sacks, A. Degani, Synthetic image data generation using BIM and computer graphics for building scene understanding, Autom. Constr. 154 (2023) 105016.
[10] K. Park, S. Ergan, C. Feng, Quality assessment of residential layout designs generated by relational Generative Adversarial Networks (GANs), Autom. Constr. 158 (2024) 105243.
[11] L. Eversberg, J. Lambrecht, Generating images with physics-based rendering for an industrial object detection task: Realism versus domain randomization, Sensors. 21 (2021) 7901.
[12] J. Kim, D. Kim, S. Lee, S. Chi, Hybrid DNN training using both synthetic and real construction images to overcome training data shortage, Autom. Constr. 149 (2023) 104771.
[13] T. Hodaň, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S.N. Sinha, B. Guenter, Photorealistic image synthesis for object instance detection, (2019) 66-70.
[14] Y. Hong, S. Park, H. Kim, H. Kim, Synthetic data generation using building information models, Autom. Constr. 130 (2021) 103871.
[15] H. Lee, J. Jeon, D. Lee, C. Park, J. Kim, D. Lee, Game engine-driven synthetic data generation for computer vision-based safety monitoring of construction workers, Autom. Constr. 155 (2023) 105060.
[16] C. Siu, M. Wang, J.C.P. Cheng, A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection, Autom. Constr. 137 (2022) 104213.
[17] T. Tschickardt, F. Kaufmann, C. Glock, Lean and BIM based flight planning for automated data acquisition of bridge structures with LiDAR UAV during construction phase, 3 (2022) 0.
[18] P. Mittal, R. Singh, A. Sharma, Deep learning-based object detection in low-altitude UAV datasets: A survey, Image Vision Comput. 104 (2020) 104046.
[19] D. Wallace, Y.H. He, J.C. Vaz, L. Georgescu, P.Y. Oh, Multimodal teleoperation of heterogeneous robots within a construction environment, (2020) 2698-2705.
[20] Y. Tan, W. Yi, P. Chen, Y. Zou, An adaptive crack inspection method for building surface based on BIM, UAV and edge computing, Autom. Constr. 157 (2024) 105161.