Synthetic Data Generation for Indoor Scene Understanding using BIM

Yeji Hong[a] , Somin Park[a], and Hyoungkwan Kim[a]

[a]Department of Civil and Environmental Engineering, Yonsei University, South Korea
E-mail: hongyeji@yonsei.ac.kr, somin109@yonsei.ac.kr, hyoungkwan@yonsei.ac.kr

Abstract –

Visual facility inspections performed manually are tasks that can be automated. Segmentation of facility image data is one of the automated methods of identifying problems in facilities. However, the machine learning methodology that is mainly used to train the segmentation model requires a large amount of training dataset. Preparing training dataset accompanies laborious manual labeling. To address this issue, we present a new method for generating synthetic data that do not require manual labeling. The method is to create photograph-style images from the BIM images; a generative adversarial network called CycleGAN is used to enable style transfer between the two different domains.

Keywords –

BIM; Cycle-Consistent Adversarial Networks (CycleGAN); Facility Management; Scene Understanding; Synthetic Data

1 Introduction

Facility management aims to effectively operate the facility for a long period of time. Whether the facility is functioning can be understood by comparing the ideal state and the current state. Recent advances in imaging devices have caused image data to be widely used to monitor the current state of facilities. Images show the appearance of the facilities so that people can understand their condition. However, in a large-scale infrastructure, it takes time for a person to check images or videos. Therefore, it is necessary to automatically extract valuable information that provides the current state from the image. Segmentation extracts information such as spatial context [[1], [2]] and structure installation [3] from the image, enabling the identification of the current state of the facility on behalf of the human.

There is a difficulty when carrying out the segmentation. The difficulty is that large-scale training dataset is needed to train machine-learning models for segmentation (such as supervised-learning). Preparing a training dataset involves collecting and annotating multiple images. While image acquisition is effortless, annotating operation requires a lot of time and effort [4]. To deal with this, we propose a novel method for generating synthetic data similar to photograph using Building Information Model (BIM) designed during infrastructure construction.

BIM is the digital twin of the infrastructure, representing geometry, material and time information of the construction entities, in electronic form. Therefore, the images captured from BIM are similar to the photograph, but not completely identical. It is our proposal to create synthetic data that can be used as training data in segmentation models by applying the style of the photograph to the image captured in the BIM. Style transfer between real-world domain and BIM domain is performed using Cycle-consistent adversary networks (CycleGAN[5]). An evaluation of whether the generated virtual data is suitable for scene understanding will be conducted in further research. The framework of the research is shown in Figure 1.

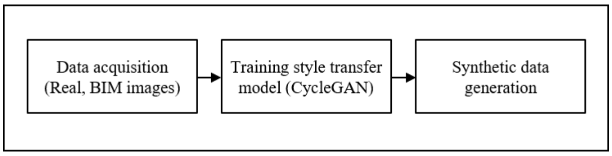
Data acquisition (Real, BIM images)
Training style transfer model (CycleGAN)
Synthetic data generation

Figure 1. Research framework

2 Related work

In the field of construction, vision-based analysis has been used as an important instrument for understanding the situation. In particular, vision-based analysis using deep learning has recently been actively conducted due to the performance enhancement of computer (CPU and GPU). Among the 2D image-based analysis techniques using deep learning are classification, detection, and segmentation. Among them, segmentation, the technology that classifies the class of each pixel and includes the two preceding methods, are widely used in the latest research in the construction sector. It was used for various construction management purposes including worker safety enhancement [3], progress monitoring [6], defect identification [[7], [8], [9]], and to update the status of the building [[10], [11]].

The aforementioned deep leaning-based method has limitation that requires a lot of training data to train the model. Preparation for the training dataset includes a labeling task, which is labour-intensive. In an effort to overcome this limitation, the previous studies created databases (ImageNet[12], SUN[13], COCO[14], Cityscape[4]). However, infrastructure facility classes such as HVAC system, plumbing system, structural element and construction materials often do not exist in the mentioned databases. Therefore, it is necessary to create training datasets that can be labelled easily on the desired objects.

To address the above issue, research have been conducted to generate training data using a 3D model [[15], [16]]. In the field of construction, Soltani et al. generated a synthetic image by adding background images and images captured from various angles on the 3D expansion model for visual reconstruction services [17]. Kim et al. took photographs of the concrete mixer trucks with a UAV camera, then created a 3D point cloud with a structure from motion technique, and created 2D synchronous images by projecting the point cloud on the plane [18].

Recently, research has also been conducted to generate synthetic data using BIM, the digital twin of infrastructure. Jong Won Ma et al. generated synthetic point clouds using 3D BIM to train the semantic segmentation model based on deep learning [19]. Although the synthetic point clouds are analogous to real point clouds, the differences in detail and volumetric information of some objects were cited as limitations. It is also difficult to train the model with the 2D synthetic images generated with unprocessed BIM. Inhae Ha et al. showed that differences exist in the feature map of the BIM images and the photographs for the same scene [20]. As such, the gap between real world and BIM is a factor that makes it difficult to use BIM image as training data. In order to reduce this gap, we create synthetic datasets that is transferred with CycleGAN so that the image obtained from BIM has a style similar to photographs.

CycleGAN [5] is a network that transfers images of different domain by learning the two mapping functions. To achieve the objective, the adversarial loss as defined in the Generative Adversarial Networks (GAN [21]) and the cycle consistency loss are used.

3 Methodology

The proposed methodology consists of a step of training CycleGAN and a step of generating synthetic data utilizing the trained network. Details are explained in the following paragraphs.

3.1 Training CycleGAN

CycleGAN [5] is a network that trains mapping function between two different domains. The network include two generators, $\mathrm{G_{AB}}$ mapping A to B and $\mathrm{G_{BA}}$ mapping B to A and two adversarial discriminators $\mathrm{D_A}$ and D_B , which enable mapping between source domain A and target domain B. In the adversarial loss $\mathcal{L}_{\mathrm{GAN}}(G_{AB}, D_B, A, B)$, $\mathrm{G_{AB}}$ tries to make $\mathrm{G_{AB}}(A)$ similar to B, and $\mathrm{D_B}$ tries to distinguish B from $\mathrm{G_{AB}}(A)$. $\mathcal{L}_{\mathrm{GAN}}(G_{BA}, D_A, B, A)$ also works the same. The additional loss due to the possibility that the correct mappings are not achieved with adversarial loss alone is a cycle consistency loss ($\mathcal{L}_{\mathrm{cyc}}$). In $\mathcal{L}_{\mathrm{cyc}}(G_{AB}, G_{BA})$, $G_{BA}(G_{AB}(A))$ is encouraged to have the same value as A, and $G_{AB}(G_{BA}(B))$ is encouraged to have the same value as B. As a result, the network operates to find G_{AB}^* and G_{BA}^* that satisfy the following expressions:

$$G_{AB}^*, G_{BA}^* = \min_{G_{AB}, G_{BA}} \max_{D_A, D_B} \mathcal{L}(G_{AB}, G_{BA}, D_A, D_B) \tag{1}$$

where,

$$\mathcal{L}(G_{AB}, G_{BA}, D_A, D_B) = \mathcal{L}_{\mathrm{GAN}}(G_{AB}, D_B, A, B) + \mathcal{L}_{\mathrm{GAN}}(G_{BA}, D_A, B, A) + \lambda \mathcal{L}_{\mathrm{cyc}}(G_{AB}, G_{BA}) \tag{2}$$

In this study, real world is the source domain and BIM is the target domain. The structure of CycleGAN is illustrated in Figure 2. The generator G_{AB} receives the real world image (photograph) $\mathrm{A_1}$ as an input and generates $G_{AB}(A_1)$ similar to the BIM image as an output. G_{BA} that receives $G_{AB}(A_1)$ as an input generates $G_{BA}(G_{AB}(A_1))$ similar to the original image A_1. L1-norm between A_1 and $G_{BA}(G_{AB}(A_1))$ is calculated in cycle consistency loss. Conversely, G_{BA} generates $G_{BA}(B_1)$ that is similar to the photograph from BIM image B_1, and the discriminator D_A discriminates whether the input image is from domain A or B.

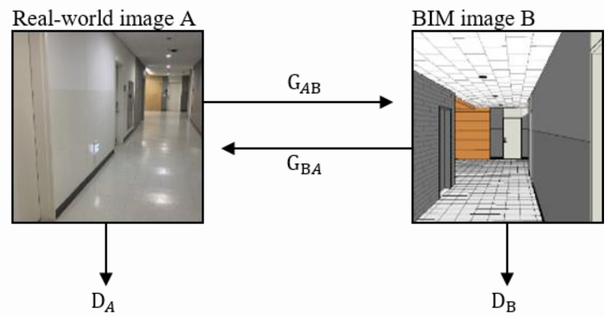

Figure 2. The structure of CycleGAN

3.2 Generating synthetic data

In this study, CycleGAN is set to train for 200 epochs. We used two learning rate settings. One sets the learning rate to 0.0002 for 200 epochs. The other one sets the same learning rate during the first 100 epochs, and linearly decreases the learning rate from 0.0002 to 0 for the remaining 100 epochs. Then every 40 of the 200 epochs, G_{BA} takes the BIM images as the input and generates the synthetic datasets. Figure 3 shows the process in which synthetic datasets are generated from each generator. We call the generator by concatenating the generator name with the number of epochs and adding the suffix "d" if the learning rate decays.

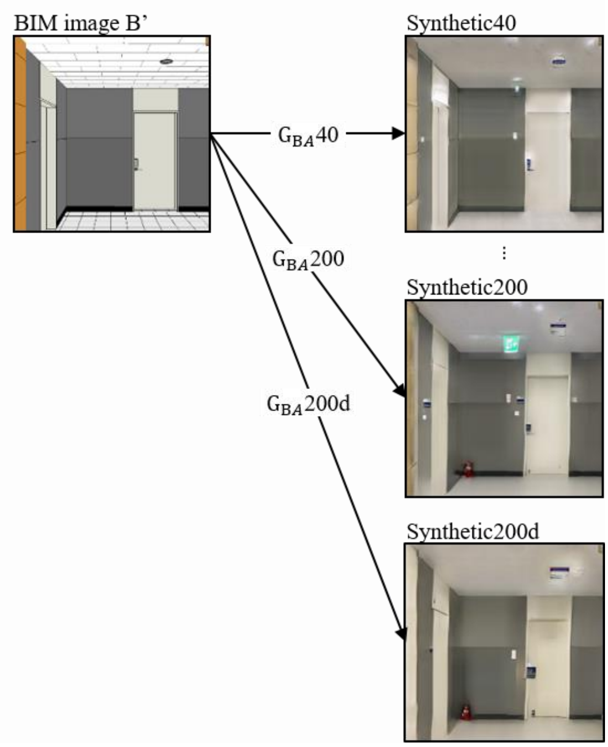

Figure 3. The process of synthetic data generation

4 Experimental Study

4.1 Dataset

The real-world domain training dataset was taken in the corridor of the north wing on the fifth floor of Yonsei University's first engineering building. 550 images were scaled from 3024 × 3024 to 512 × 512 pixels. Another training dataset, BIM domain, was extracted from Yonsei University's first engineering building BIM, which was implemented in Revit software. This BIM is the same as that used in Inhae Ha et al. [20]. 564 images with image size 512 × 512 were obtained by creating 3D views and extracting views as image files. The 3D views were also constructed in the north wing on the fifth floor of BIM.

The dataset, used as an input of generator G_{BA} to generate synthetic datasets, is a set of 100 512 × 512-sized images that do not overlap with training dataset.

4.2 Implementation

CycleGAN was trained with λ of 10 in Equation (2) and a batch size of 1. As mentioned in section 3.2, two settings of the learning rate were used: (1) non-decay setting and (2) decay setting.

4.3 Results and Discussion

This section compares one BIM dataset and eight synthetic datasets. The notation for the synthetic dataset is the same as for the generator. Figure 4 shows the image examples of BIM and synthetic datasets.

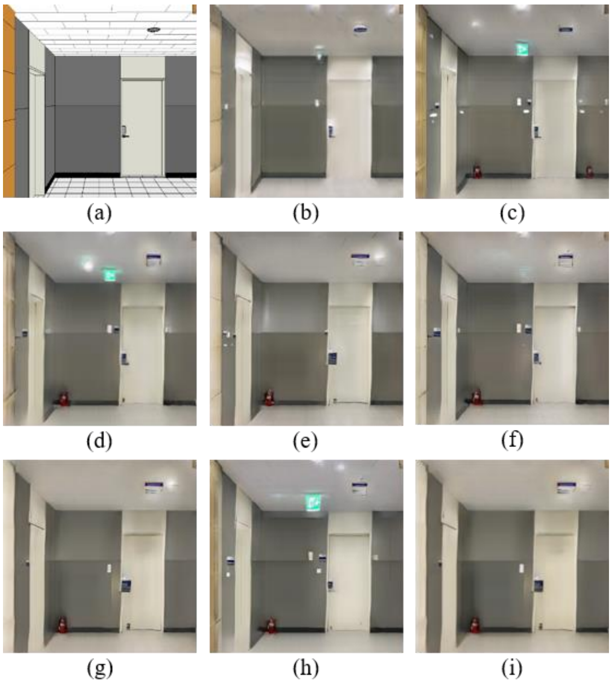

Figure 4. Examples of results from different datasets; (a) BIM, (b) Synthetic40, (c) Synthetic80, (d) Synthetic120, (e) Synthetic120d, (f) Synthetic160, (g) Synthetic160d, (h) Synthetic200, (i) Synthetic200d

As shown in Figure 4(b), synthetic40 looks similar to BIM. The color changed similar to the photograph, and the light shape and part of the exit marking in the BIM were generated. Some noise generated patterns of doorplates that appeared in the photograph. The grid patterns of floor and ceiling in the BIM have not yet disappeared. Synthetic80 shows the disappearance of a number of doorplate patterns in Figure 4(c). The position of the light is similar to the photograph, and the reflection of the light on the walls looks realistic in synthetic80. As shown in Figure 4(h), BIM's floor pattern and repetitive patterns (especially in wood and brick area) are almost eliminated in synthetic200.

As a result of the style transfer, it was found that the network produced small objects such as the fire extinguisher and exit marker, and caused color changes to produce values similar to those of photograph. In that process, small patterns such as exit marker appeared and disappeared, depending on the number of epochs. It is assumed that the synthetic200 and synthetic200d, the models trained with the 200 epochs, did not converge.

Cosine similarity was used as an indicator of evaluating synthetic datasets. Cosine similarity is a measure of the similarity of two vectors. The cosine similarity was calculated between the photographs having the same scene and generated synthetic datasets. Table 1 shows the average and standard deviation of cosine similarity. All the synthetic datasets have cosine similarity greater than BIM dataset.

Table 1. Average and standard deviation of cosine similarity with same scene real dataset

Dataset	Cosine Similarity Avg.	Cosine Similarity Std.
BIM	.9145	.0220
Synthetic40	.9488	.0198
Synthetic80	.9495	.0193
Synthetic120	.9497	.0204
Synthetic120d	.9479	.0211
Synthetic160	.9492	.0217
Synthetic160d	.9454	.0232
Synthetic200	.9500	.0209
Synthetic200d	.9442	.0237

The synthetic datasets with photograph style and BIM content were created. The datasets, whose label can be easily obtained from BIM, are valuable as the training datasets of the segmentation model. In further research, the segmentation model will be trained with synthetic datasets and the performance of the model will be evaluated to assess whether the datasets are adequate for scene understanding.

5 Conclusion

This study proposed a method for generating synthetic image data that can be used as a training set for scene understanding in the field of facility management. In the proposed method, CycleGAN is used to train a mapping function such that transfer between BIM images and their corresponding real-world images is performed. The generator of the proposed model produces synthetic data similar to real images using the BIM images. Cosine similarity values calculated using corresponding scene-photographs showed that the synthetic data are more realistic than BIM images. The label of synthetic data is the same as that of the corresponding BIM image, so labelled data can be obtained without any annotating works. This method could be highly beneficial for collecting data to train deep learning models in that the models usually require a large amount of data. In the experiment, noise patterns on the synthetic data appeared and disappeared repeatedly as the training of CycleGAN progressed. The noise patterns are expected to adversely affect the training of deep learning models when synthetic data, including the patterns, are used as training data for the models. To tackle this problem, further research should be conducted to improve the stability of training. In a future study, synthetic datasets generated in consideration of the aforementioned problem will be used as training sets for scene understanding.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (No. 2018R1A2B2008600) and the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 20SMIP-A156488-01).

References

[1] Asadi, K., Ramshankar, H., Pullagurla, H., Bhandare, A., Shanbhag, S., Mehta, P., … and Wu, T. Vision-based integrated mobile robotic system for real-time applications in construction. *Automation in Construction*, 96: 470-482, 2018.

[2] Atkinson, G. A., Zhang, W., Hansen, M. F., Holloway, M. L. and Napier, A. A. (2020). Image segmentation of underfloor scenes using a mask regions convolutional neural network with two-stage transfer learning. *Automation in Construction*, 113: 103118, 2020.

[3] Fang, W., Zhong, B., Zhao, N., Love, P. E., Luo, H., Xue J., and Xu, S. A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network. *Advanced Engineering Informatics*, 39: 170-177, 2019.

[4] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., … and Schiele B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213-3223, Las Vegas, USA, 2016.

[5] Zhu, J. Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer*

vision, pages 2223-2232, Venice, Italy, 2017.

[6] Rahimian, F. P., Seyedzadeh, S., Oliver, S., Rodriguez, S., and Dawood, N. On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning. *Automation in Construction*, 110: 103012, 2020.

[7] Dung, C. V. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99: 52-58, 2019.

[8] Bang, S., Park, S., Kim, H., and Kim, H. Encoder–decoder network for pixel-level road crack detection in black-box images. *Computer-Aided Civil and Infrastructure Engineering*, 34(8): 713-727, 2019.

[9] Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., and Wang, S. DeepCrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3): 1498-1512, 2018.

[10] Ferguson, M., Jeong, S., and Law, K. H. Worksite Object Characterization for Automatically Updating Building Information Models. In *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*, pages 303-311, Atlanta, USA, 2019.

[11] Ying, H. Q. and Lee, S. A Mask R-CNN Based Approach to Automatically Construct As-is IFC BIM Objects from Digital Images. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, pages 764-771, Banff, Canada, 2019.

[12] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248-255, Miami, USA, 2009.

[13] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485-3492, San Francisco, USA, 2010.

[14] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740-755, Zurich, Switzerland, 2014.

[15] Peng, X., Sun, B., Ali, K., and Saenko, K. Learning deep object detectors from 3d models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1278-1286, Santiago, Chile, 2015.

[16] Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., and Cipolla, R. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077-4085, Las Vegas, USA, 2016.

[17] Soltani, M. M., Zhu, Z., and Hammad, A. Automated annotation for visual recognition of construction resources using synthetic images. *Automation in Construction*, 62: 14-23, 2016.

[18] Kim, H. and Kim, H. 3D reconstruction of a concrete mixer truck for training object detectors. *Automation in Construction*, 88: 23-30, 2018.

[19] Ma, J. W., Czerniawski, T., and Leite, F. Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. *Automation in Construction*, 113: 103144, 2020.

[20] Ha, I., Kim, H., Park, S., and Kim, H. Image retrieval using BIM and features from pretrained VGG network for indoor localization. *Building and Environment*, 140: 23-31, 2018.

[21] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672-2680, Montreal, Canada, 2014.