Smart Structures and Systems, *Vol. 32, No. 2 (2023) 111-121*
https://doi.org/10.12989/sss.2023.32.2.111

Assembly performance evaluation method for prefabricated steel structures using deep learning and k-nearest neighbors

Hyuntae Bang [1], Byeongjun Yu [2] and Haemin Jeon*[3]

[1] *Department of Autonomous Vehicle System Engineering, Chungnam National University, Yuseong-gu, Daejeon, 34134, Republic of Korea*
[2] *StradVision, Seoul, 06621, Republic of Korea*
[3] *Department of Civil and Environmental Engineering, Hanbat National University, Yuseong-gu, Daejeon, 34158, Republic of Korea*

(*Received March 10, 2023, Revised July 28, 2023, Accepted August 14, 2023*)

Abstract. This study proposes an automated assembly performance evaluation method for prefabricated steel structures (PSSs) using machine learning methods. Assembly component images were segmented using a modified version of the receptive field pyramid. By factorizing channel modulation and the receptive field exploration layers of the convolution pyramid, highly accurate segmentation results were obtained. After completing segmentation, the positions of the bolt holes were calculated using various image processing techniques, such as fuzzy-based edge detection, Hough's line detection, and image perspective transformation. By calculating the distance ratio between bolt holes, the assembly performance of the PSS was estimated using the k-nearest neighbors (kNN) algorithm. The effectiveness of the proposed framework was validated using a 3D PSS printing model and a field test. The results indicated that this approach could recognize assembly components with an intersection over union (IoU) of 95% and evaluate assembly performance with an error of less than 5%.

Keywords: assembly performance evaluation; *k*-nearest neighbors; machine learning; prefabricated steel structure; semantic segmentation; vision sensor

1. Introduction

In the Republic of Korea, the construction industry is facing a severe shortage of workforce due to the aging of construction workers, with over 60% of construction sites comprising workers aged 50 and above (Construction workers mutual aid association 2021). Furthermore, social efforts to reduce working hours, along with related laws and regulations, are accelerating the decrease in available manpower for the construction industry. In particular, the shortage of skilled manpower is emerging as a significant problem for the entire construction sector (Rahman *et al.* 2013). To solve this problem, a need for increased automation and efficiency in construction is emerging, and smart construction is attracting attention as a way to realize this goal. Smart construction is defined as the application of advanced digital technologies such as robotics, artificial intelligence (AI), building information modeling (BIM), and the Internet of Things (IoT) to traditional construction technologies for the purpose of shortening the construction period, reducing the required manpower, and enhancing onsite safety (Ministry of Land, Infrastructure, and Transport 2021). Due to its high efficiency and low cost, smart construction has attracted considerable attention in recent years (Štefanič and Stankovski 2018). Prefabricated structures, where modular components are fabricated offsite and then assembled onsite, are an important aspect of smart construction (Boafo *et al.* 2016). The use of prefabricated steel structures (PSSs) is popular because doing so maximally reduces the construction period, saves resources, reduces energy consumption, and protects the environment (Deng *et al.* 2020).

For automatic construction and maintenance of modular structures, studies have been conducted in recent decades to satisfy various social demands (Zonta *et al.* 2007, Koem *et al.* 2016, Wasim *et al.* 2020, Nguyen *et al.* 2022). However, human-based visual inspection is typically performed after the production of modular structures about the PSSs. Inspectors check the dimensions and shapes of the PSSs according to the standard specifications (Ministry of Land, Infrastructure, and Transport 2013). Most of the research to automate this process use laser scanning to reconstruct three-dimensional (3D) structural models and monitor deformations (Park *et al.* 2007, Kim *et al.* 2014, 2016, Xu *et al.* 2020). For example, Park *et al.* (2007) proposed a structural health monitoring system that uses terrestrial laser scanning technology to measure structure deflection with high accuracy (Park *et al.* 2007). Kim *et al.* (2014, 2016) proposed a coordinate transformation algorithm for noncontact dimensional quality inspections (Kim *et al.* 2014, 2016). Moreover, Xu *et al.* (2020) proposed a 3D reconstruction and measurement system for monitoring surface defects in prefabricated components (Xu *et al.* 2020). They used the *k*-nearest neighbors (kNN) algorithm

*Corresponding author, Professor,
E-mail: hjeon@hanbat.ac.kr

ISSN: 1738-1584 (Print), 1738-1991 (Online)

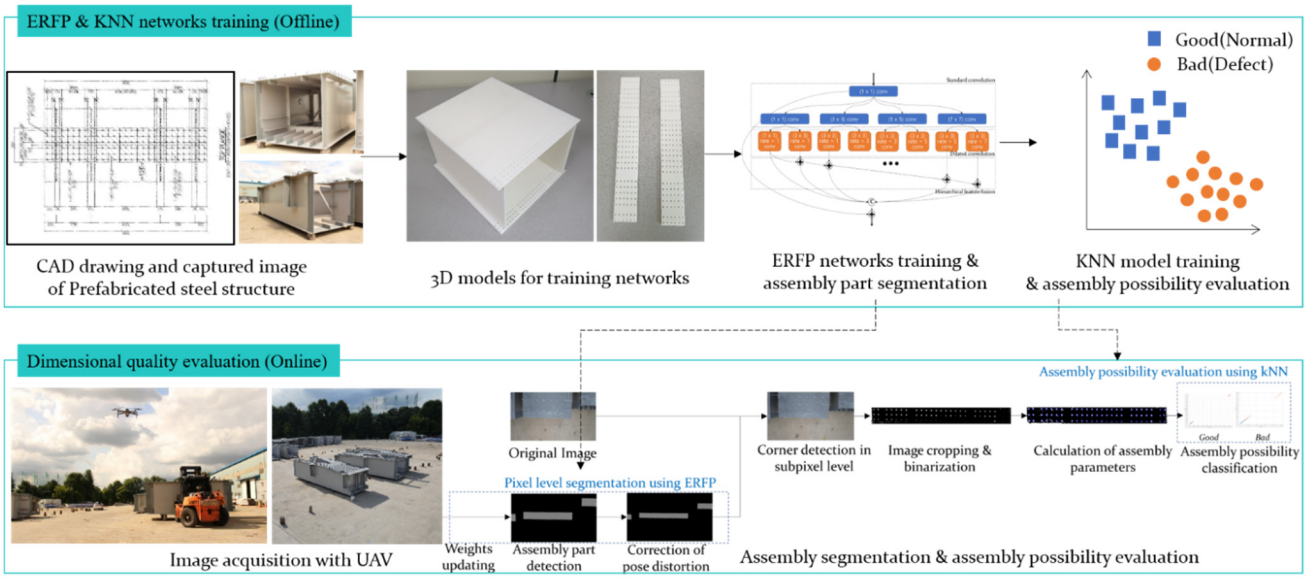


Fig. 1 Schematic diagrams of PSS assembly performance evaluation methods using machine learning techniques

to reduce data density and the alpha shape algorithm to effectively extract boundaries.

This study proposes a method for PSS assembly evaluation using a low-cost vision sensor and machine learning (Jeon *et al.* 2014, Choi *et al.* 2017). Machine learning is a process by which a machine uses information from a training dataset for a specific purpose and uses data to form or improve the system (Mitchell *et al.* 2007). Machine learning techniques can be divided into supervised and unsupervised approaches according to the characteristics of the input dataset (Géron 2019). Supervised learning is performed based on the trends and distribution of data by mapping an input to an output. It can be further divided into classification and regression methods according to the type of the target value. Classification methods involve learning and prediction using classified target values, whereas regression methods use continuous numerical values and trends. In contrast, unsupervised learning approaches learn patterns from unlabeled data (Hinton and Sejnowski 1999). The results are derived by creating clusters or analyzing relationships according to the distribution or characteristics of the data.

The proposed method uses supervised machine learning techniques, such as semantic segmentation and the kNN classifier. Semantic segmentation, a pixel-level object classification technique, is used to detect assembly components. After completing segmentation, the bolt holes are extracted, and their positions are calculated using image processing techniques such as fuzzy-based edge detection, Hough's line detection, and image perspective transformation. To ensure the accuracy of the assembly performance evaluation model with minimal differences between normal and defective images, we adopted a two-step approach: 1) using deep learning to detect assembly parts, and 2) deriving the input variables from the detected assembly parts for classification. The calculated positions of the bolt holes are converted to distance ratios, which are used as input data for the kNN classifier. Utilizing the kNN algorithm, the assembly performance is classified into two classes, normal and defective. The kNN results are compared with those of other classification learning techniques, such as decision trees, discriminant analysis and support vector machines (SVMs) classifiers, to evaluate the performance of assembly. The proposed framework for assembly quality evaluation is verified using 1:10 scaled 3D printing models and real-scale PSSs with an unmanned aerial vehicle (UAV). The remainder of this paper is organized as follows. Section 2 describes PSS assembly performance evaluation methods that use machine learning techniques. An introduction to deep learning methods for semantic segmentation is described in Section 2.1. The developed semantic segmentation model for assembly part classification and the assembly performance evaluation method based on the positions of bolt holes are introduced in Sections 2.2 and 2.3, respectively. The performance of the proposed method is validated with a 3D printing model in Section 3. Experimental tests with PSSs are conducted, and the results are discussed in Section 4. Finally, the conclusions and future research directions are discussed in Section 5.

2. Dimensional quality evaluation of PSS using machine learning

To evaluate the assembly performance of PSSs, two-stage processes combining different machine learning techniques have been adopted, as shown in Fig. 1. In an offline study, a newly proposed semantic segmentation network was trained to classify the assembly parts in captured images. The bolt holes in the segmented area were detected using various image processing techniques, including fuzzy-based edge detection, Hough's line detection, and homography transformation. In the filtered image, the locations of the bolt holes were calculated. By calculating the distance ratios between bolt holes, the

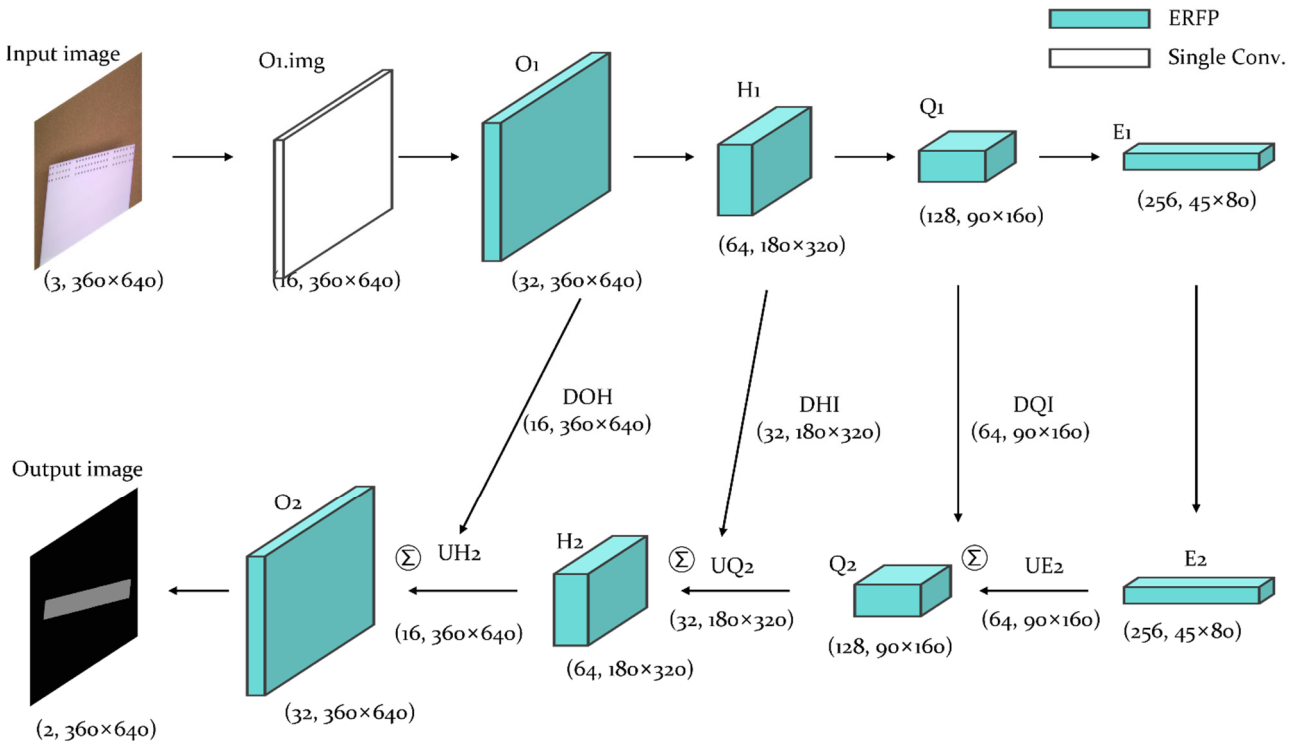


Fig. 2 Encoder-decoder network with ERFP convolution modules for assembly segmentation

assembly performance was binarized using the kNN classifier. Once the semantic segmentation and classification learning networks were trained, they were used to evaluate the assembly performance of PSSs from newly captured images in real-time.

2.1 Deep learning methods for semantic segmentation

Deep learning techniques have been widely used in the field of data analysis due to the development of computer hardware and deep learning networks with various forms and uses. In the field of computer vision, convolutional neural networks (CNNs) have emerged for classification, object detection, and segmentation. A CNN consists of a convolution layer and a pooling layer, and it extracts features using filters with constant heights, widths and strides. The convolution layer outputs a feature map by performing a convolution operation, and the pooling layer reduces the number of required computations by compressing the result output by the convolution layer.

In addition to recognizing and classifying the existence of objects in an image, many studies have been conducted on object segmentation and semantic segmentation to detect the area where an object exists at the pixel level. A fully convolutional network (FCN), which converts all fully connected layers (the final classifier layers of an image classification network) to convolutions, has achieved remarkable performance in the field of semantic segmentation (Krizhevsky *et al.* 2012, Simonyan and Zisserman 2014, Szegedy *et al.* 2015, Long *et al.* 2015). However, most FCNs that replace fully connected layers with convolutions, such as AlexNet, VGG16, and GoogLeNet, have the following drawbacks: 1) object details disappear due to the use of a simple upsampling procedure, and 2) location information is lost due to the continuous use of max pooling.

Network structures were proposed to solve this problem, featuring an encoder that extracts features without using a fully connected layer, and a decoder that progressively increases in size (Ronneberger *et al.* 2015, Noh *et al.* 2015, Badrinarayanan *et al.* 2017, Jégou *et al.* 2017). Ronneberger *et al.* (2015) proposed a U-net that combines high-resolution features from the encoding pathway into the upsampled output to achieve more precise results (Ronneberger *et al.* 2015). Noh *et al.* (2015) proposed a deconvolution approach with a stride of two to associate multiple outputs in a single activation step (Noh *et al.* 2015). Badrinarayanan *et al.* (2017) applied max pooling as an upsampling technique. Since the pooling method of copying the same value loses spatial information, Jégou *et al.* (2017) used an upsampling method that stores the indices at the time of pooling (Badrinarayanan *et al.* 2017, Jégou *et al.* 2017).

In addition to research on network structures, the use of kernels to improve the performance of CNNs is also being studied. The wider and more diverse features were extracted with a small number of parameters through dilated convolution (Yu and Koltun 2015, Chen *et al.* 2017, Yu *et al.* 2017, Mehta *et al.* 2018). Furthermore, a method was proposed to extract features from various domains based on flexible receptive domains using a non-fixed data-driven kernel (Dai *et al.* 2017, Liu *et al.* 2017). In this paper, a receptive field pyramid is proposed in an encoder- decoder framework with two dilation rates at each convolution module to detect assembly parts with high accuracy.

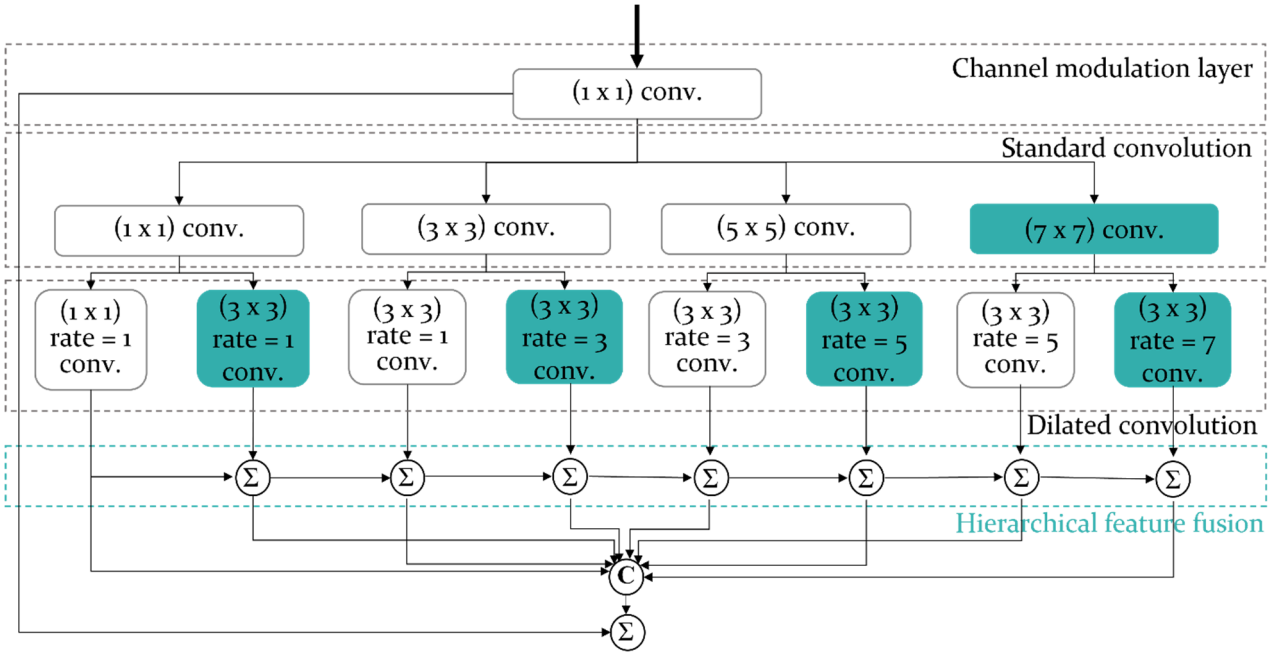


Fig. 3 ERFP convolution modules

2.2 Efficient receptive field pyramid for the segmentation of assembly parts in PSSs

To segment PSS assembly parts which are the rectangular regions including the bolt holes, a modified version of the convolution module from the receptive field block (RFB) network, namely the efficient receptive field block pyramid (ERFP) and inspired by the eccentric functionality of the human visual system, was adopted (Liu *et al.* 2018, Yu *et al.* 2022). The encoder-decoder structure of the deep learning network for assembly part segmentation is shown in Fig. 2. The highlighted convolution modules in the figure are shown in Fig. 3. The convolution module begins with a channel modulation layer, which enhances performance by adjusting feature characteristics along the channel dimension. To compensate for the reduction in detection accuracy caused by channel modulation, two dilation convolutions with different dilation rates were applied to each standard convolution to diversify the pathway and extract features of various sizes. As the captured image consisted of a limited number of channels, a front layer (O1.img) that warmed up the network has been added in the front. After extending the number of channels to more than eight, the ERFP consisted of eight convolutions that could be applied (Yu *et al.* 2022). The proposed network provides higher accuracy than other approaches due to the use of 1) a light convolution module with channel modulation, incorporating various kernel sizes and two different dilation rates, and 2) hierarchical feature fusion, which eliminates grid artifacts. The modified parts in the proposed ERFP are indicated with colored boxes in Fig. 3

The proposed ERFP module searches a wide area using the input feature map, which is composed of a standard receptive field convolution with various kernel sizes and a dilated convolution with a dilation rate that is proportional to the kernel size. The proposed ERFP can be expressed by the following equations

Table 1 Efficiency of the network with the channel modulation layers

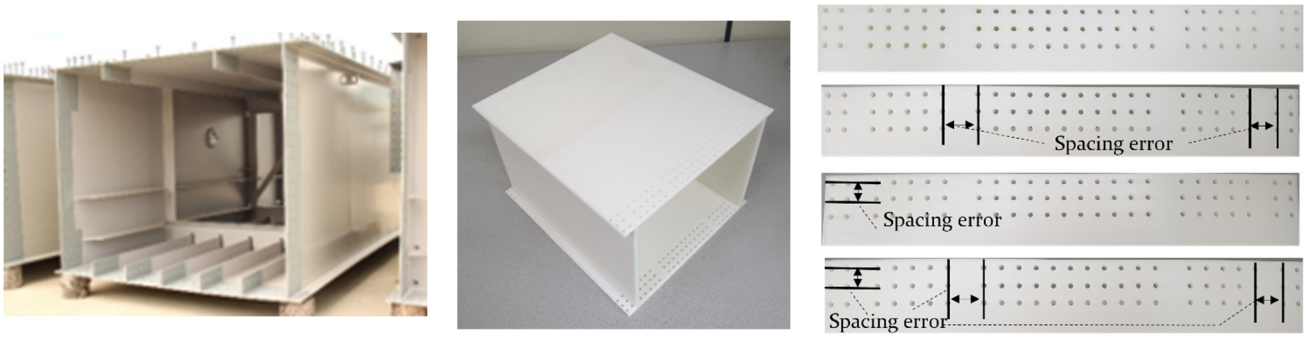
Channel Modulation	Computational complexity (GMac)	Number of parameters
Including	26.33	2.4 M
Excluding	34.71	3.36 M

$$ERFP = f_{K,D}(f_{(2p-1),1}(f_{1,1}(x))) \tag{1}$$

$$f_{K,D}(x_i) = \sum_{k=1}^{K} x[i + D * k] \cdot w[k] \tag{2}$$

$$D = |2(p + b) - 5| \tag{3}$$

where x and p are the input feature map and branch in the ERFP, respectively; $f_{k,D}(x)$ is a convolution with a kernel size of (K×K) and a dilation rate of D; and $x[i]$ and $w[i]$ are the i^{th} elements of the input feature map and kernel weight, respectively. Here, p = [1,2,3,4] and b = [1,2] are the orders of the standard and dilated convolutions, respectively. In Eq. (1), we set K = 1 when both p and b are 1, and K = 3 for all other cases. In addition, a convolution module was constructed to appropriately sum the features extracted from kernels with various sizes and to add hierarchical feature fusion to the last output layer for degridding (Wang *et al.* 2019). Table 1 demonstrates the efficiency of ERFP by comparing computational complexities and parameter numbers of networks with and without channel modulation layers. These layers are located in front of the pyramid and reduce the channels of the input feature map to minimize complexities. With the channel modulation layers, the



(a) (b)

Fig. 4 3D printing models. (a) The captured images of PSSs and (b) 3D-printed models (1:10 scale)

computational complexity and the number of parameters were reduced by 24% and 29%, respectively.

2.3 Assembly performance evaluation with kNN

To evaluate the assembly performance of a PSS, various image processing techniques were applied to calculate the positions of the bolt holes on the segmented image. From the locations of the bolt holes, the distance ratios between the bolt holes in the longitudinal and latitudinal directions were calculated. The distance ratios between the detected bolt holes were used as input variables for the kNN classifier. The kNN classifier algorithm has been widely used because of 1) its conceptual simplicity, 2) its ability to perform a nonparametric analysis with a significant reduction in the computational load, and 3) the fact that its error is bounded by twice the Bayesian error when the size of the training data approaches infinity (Cover and Hart 1967). This method calculates the distance of a new point from a set of previously classified points and assigns the unclassified data to the nearest point. Various distance metrics, such as Euclidean distance, cosine similarity, and Mahalanobis distance, can be applied. In this paper, the cosine similarity method was applied to calculate the distances of all data during the learning process. Rather than estimating a numerical model through learning, it iteratively classifies sample points so that the k closest training data can be contained in a single feature space (Bhatia *et al.* 2010).

The results obtained from the kNN algorithm were compared with those of other widely known classification learning methods, such as decision trees, discriminant analysis and SVMs classifiers. A decision tree is a nonparametric supervised learning method that is used for regression and classification. Regression and classification are performed by dividing the input data to create branches based on specific numeric values or conditions in the data in the same way that a tree creates multiple branches from a single stem (Myles *et al.* 2004). The discriminant analysis finds a linear combination of features to classify objects or events (McLachlan 2005). The SVM model calculates support vectors among divided clusters of the training dataset and performs optimization to find the linear decision boundary that most effectively differentiates the input data based on the calculated support vectors (Suthaharan 2016, Noble 2006). The performance contribution of each model was evaluated by using the root mean square error. In addition, k-fold cross-validation was used to improve the model's performance and prevent over fitting. The verification procedure was performed five times with the k value set to 5 for all the tests.

3. Performance validation with a 3D printing model

3.1 Assembly part segmentation with the ERFP

The production of actual PSS components with binary cases (a part can or cannot be assembled) is challenging due to its high cost. Instead of manufacturing real-scale PSS components, 3D printed models with normal and defective cases were produced for segmentation and assembly performance evaluation (see Fig. 4). As shown in the figure, 3D models that could and could not be assembled, referred to as normal and defective cases, respectively, were produced on a 1:10 scale of the actual model. In this study, defective cases were created by changing the positions of the bolt holes. The defective parts were manufactured to have errors at intervals of 0.2 to 0.3 mm, which is one-tenth of the bolt-hole tolerance stipulated in the standard specification for road bridges provided by the Ministry of Land, Infrastructure and Transport in the Republic of Korea (Ministry of Land, Infrastructure, and Transport 2013). Defective and normal bolt-hole cases were manufactured, as shown in Fig. 4(b). We created two sets of box-shaped PSS models, with each set containing a total of four assembly parts. As shown in the figure, we designed 3D models to include a normal assembly part and three defective assembly parts, each with different positions of the bolt holes that constitute the assembly.

To verify the performance of the ERFP segmentation network, a total of 300 images of the defective and normal PSS components were collected and labeled. Among the images, 240 were defective, and 60 were images of normal components. Of the original and labeled images, 60 (20% of the total) were reserved for use as test data. Augmented images were generated by applying elastic distortion, horizontal flipping, color correction, and affine transformation to the remaining 240 images. This provided 1,200 training images. PyTorch was used as a deep learning

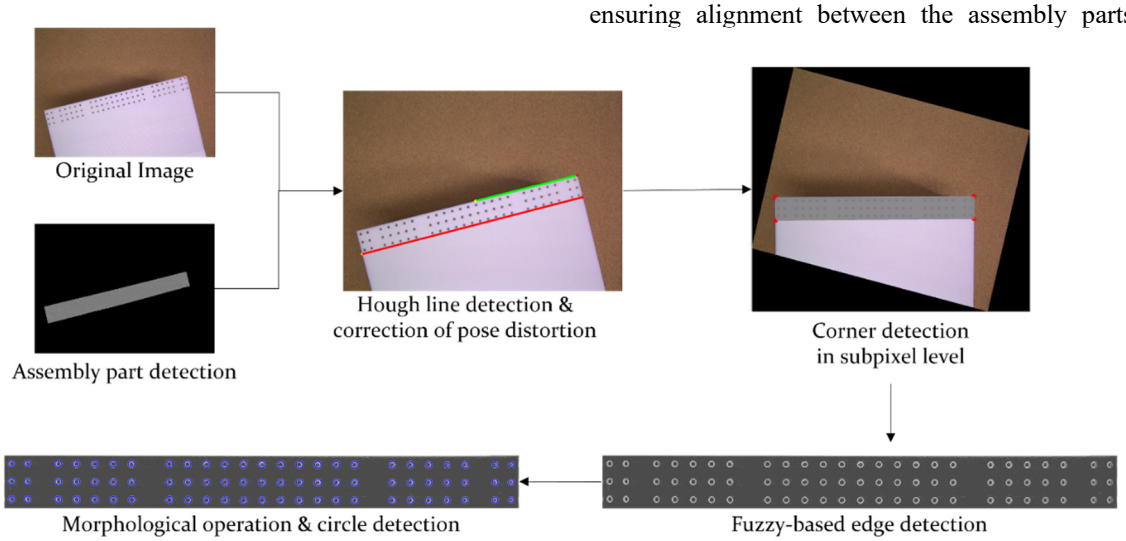


Fig. 5 Image processing for locating bolt holes

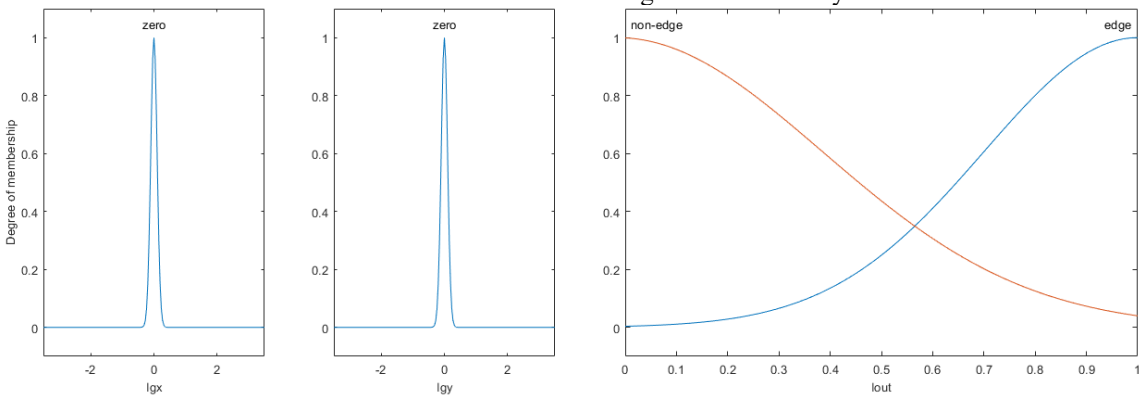


Fig. 6 Fuzzy membership function for edge detection

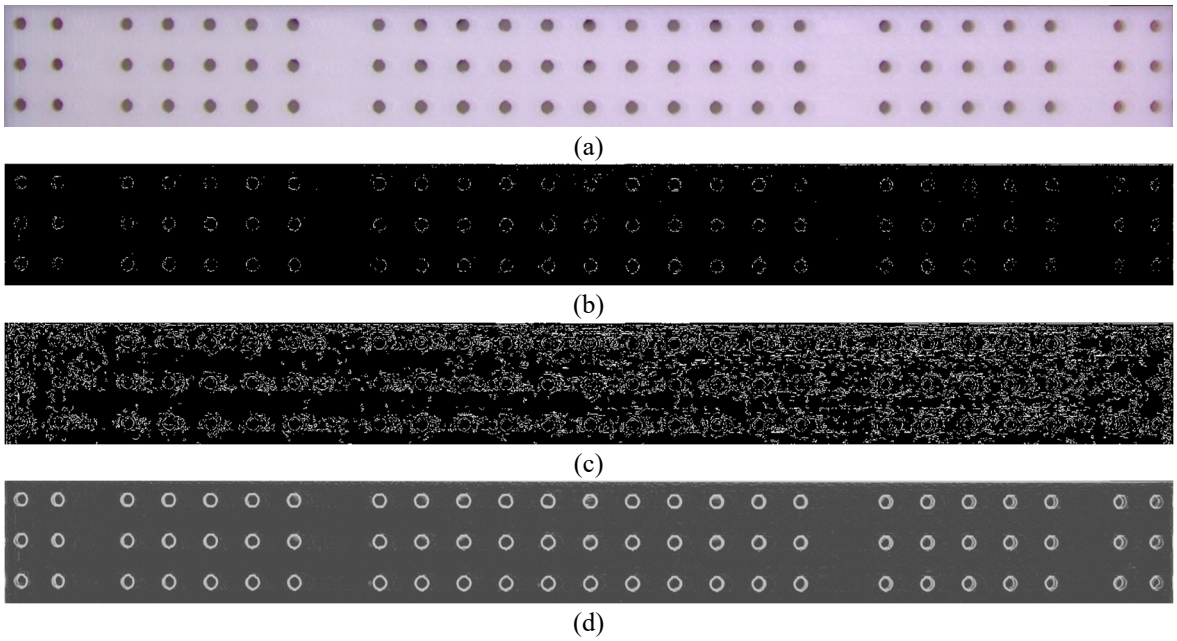
framework for model building, and training and testing were conducted using a workstation equipped with an Intel Core i7-9700K CPU at 3.60 GHz, 32 GB of RAM, and an Nvidia GeForce GTX 1080 GPU. The batch size for training the model was set to 3, and the number of epochs for learning the entire dataset was set to 300. We set the batch size considering the image resolution (3840×2748) and the limited hardware specifications. With the test dataset containing 60 images, the learning performance of the assembly part detection approach produced an intersection over union (IoU), F1 score, precision, recall, and run time of 96.64%, 98.28%, 98.43%, 98.16%, and 39.27 frames per second (FPS), respectively.

3.2 Assembly performance evaluation based on the positions of bolt holes

The image processing techniques used to calculate the positions of the bolt holes are shown in Fig. 5. As seen in the figure, the misalignment of the image was corrected, ensuring alignment between the assembly parts and the image's coordinate system. From the transformed image, the corners were detected at the sub-pixel level using the segmented results provided by the proposed deep learning algorithm. Due to the vagueness of the captured images, a fuzzy-based edge detection operator that does not specify a threshold technique was applied (Versaci and Morabito 2021). In the fuzzy logic operator, membership functions for edge detection were empirically designed through trial and error, as shown in Fig. 6. The fuzzy rule for determining whether an edge exists is as follows:

- IF I_{gx} is zero and I_{gy} is zero THEN I_{edge} is false
- IF I_{gx} is not zero or I_{gy} is not zero THEN I_{edge} is true

The images applying edge detection algorithms such as the Canny, Sobel, and fuzzy edge detectors with gradients in the X and Y axes are shown in Fig. 7. As shown in the



(a) (b) (c) (d)

Fig. 7 Results of different edge detection algorithms. (a) Original images and the results of the (b) Sobel, (c) Canny and (d) fuzzy edge detectors

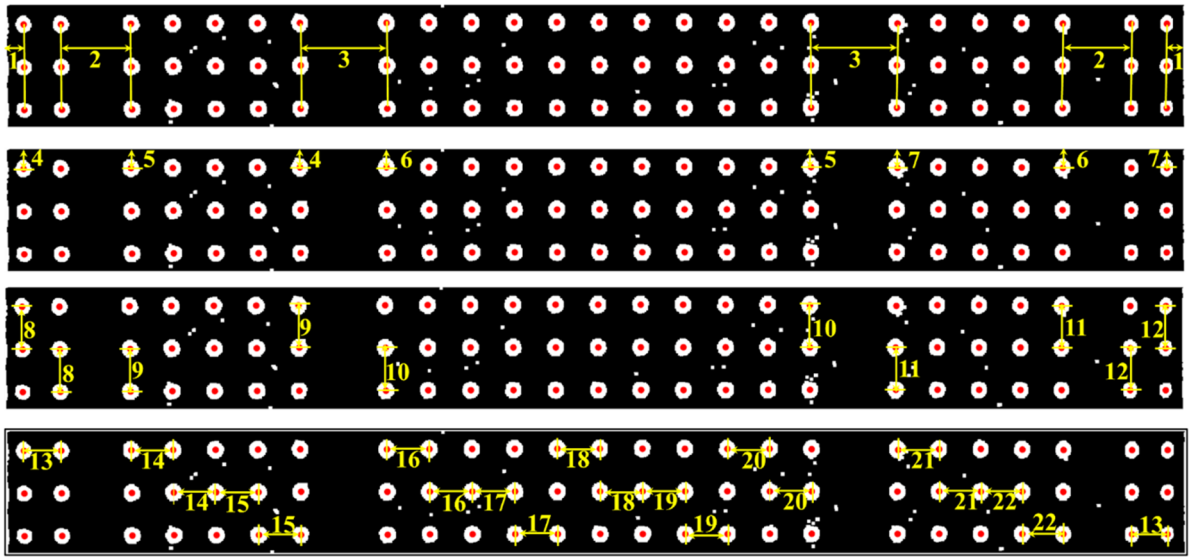


Fig. 8 Input parameters for the assembly performance evaluation network

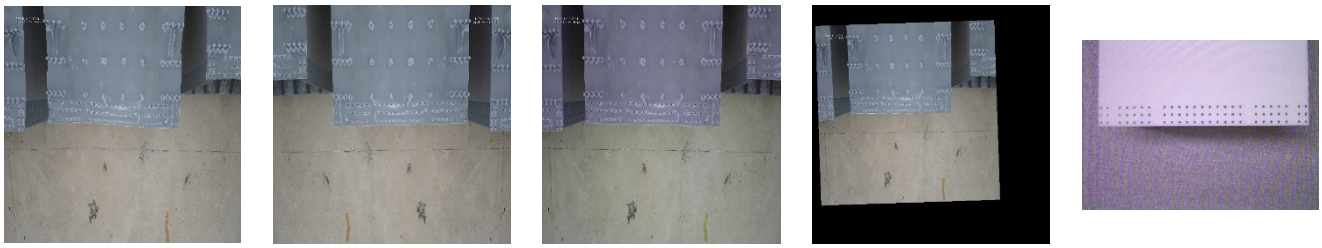
figure, the edge detection algorithm with the fuzzy rules achieved the best performance regardless of the background noise.

After performing fuzzy-based edge detection, dimensional information, such as the ratios of the distances between the bolt holes in the longitudinal and latitudinal directions, was calculated (see Fig. 8) using Hough's circle detection and morphological operations such as erosion and dilation. Utilizing the dimensional parameters of the bolt holes, the images were classified as normal or defective by calculating the ratios between the segmented bolt holes.

Once the assembly components were segmented, the differences between the calculated and predefined relative positions of the bolt holes were used to evaluate the assembly performance of the PSSs. In other words, the ratios of the distances between the bolt holes were used as the input parameters for the dimensional quality evaluation, as shown in Fig. 8. For example, the number 1 was derived from the ratio of the horizontal distances on the far left and far right. By utilizing ratios rather than absolute distances for normalization, we ensure comparable values regardless of variations in image dimensions.

The location of the input parameters can vary depending on the position of the assembly's bolt holes, and this paper is written specifically for a single case where the information from the drawing is provided. By considering allowable errors of ±2 mm between the outermost bolt holes in the group and ±0.5 mm for each bolt hole, 500 bolt hole combinations for normal and defective PSSs were randomly generated, and the machine learning network was trained

Fig. 9 Experimental test with a UAV



(a) (b) (c) (d) (e)

Fig. 10 Training dataset for assembly part segmentation in UAV captured images. The UAV-captured images with applied augmentation of (a) elastic distortion, (b) horizontal flipping, (c) color jitter, and (d) affine transformation; and (e) the captured image of 3D modeling structures

Table 2 Dimensional quality evaluation accuracy achieved with a 3D printing model for normal and defective assembly parts

Method	Decision tree	Discriminant analysis	SVM	kNN
Accuracy	95.8	93.7	95.8	**97.9**

*The best score is highlighted in bold.

based on these combinations. The kNN classifier produced the best results, as shown in Table 2. The kNN classifier, which did not make any assumptions concerning the fundamental data distribution, performed slightly better than other classifiers in terms of pattern recognition for binary categories.

4. Field test with a UAV

An experimental test with a full-scale PSS and a UAV was conducted to verify the performance of the proposed assembly evaluation algorithm using supervised machine learning methods. The UAV employed in this study was a DJI Mavic 2 Enterprise Advanced, as shown in Fig. 9. The UAV was equipped with an RGB camera with a maximum resolution of 48 megapixels and a 32x digital zoom. A total of 100 images were captured through the UAV, and 20 of these images were used to augment the previously collected 3D printing model images for training the segmentation network. The augmented dataset was utilized to construct a deep learning detection model to enhance the accuracy of assembly part segmentation. However, for the assembly

Table 3 Accuracy of the assembly performance evaluation with a real-scale PSS.

Method	Decision tree	Discriminant analysis	SVM	kNN
Accuracy	93.7	91.2	95.0	**96.2**

*The best score is highlighted in bold.

performance evaluation using the machine learning models, the pre-trained model, originally trained on 3D printing model assembly performance data, was relied upon without the need for additional data from the UAV. The performance of the proposed ERFP was verified through the remaining 80 images (20% of the 400 total images). As in the performance verification approach using 3D models, the training images were augmented by applying elastic distortion, horizontal flipping, color correction, and affine transformation, as shown in Figs. 10(a)-(d). The augmented images were combined with existing 3D model images as shown in Fig. 10(e) to construct the training dataset.

The original, labeled, and segmented images obtained using the conventional RFB and the proposed ERFP are shown in Fig. 11. The segmentation performance was evaluated by calculating the IoU, F1 score, precision, recall, and runtime, which were 95.74%, 97.82%, 97.73%, 97.93%, and 38.31 FPS, respectively. The conventional RFB method achieved IoU, F1 score, precision, recall, and runtime values of 92.93%, 96.33%, 95.32%, 97.38%, and 59.56 FPS, respectively. As shown in the figure, the proposed ERFP yielded better segmentation results for most PSS images.

By using the trained kNN algorithm and the image processing techniques described in Section 3, the assembly

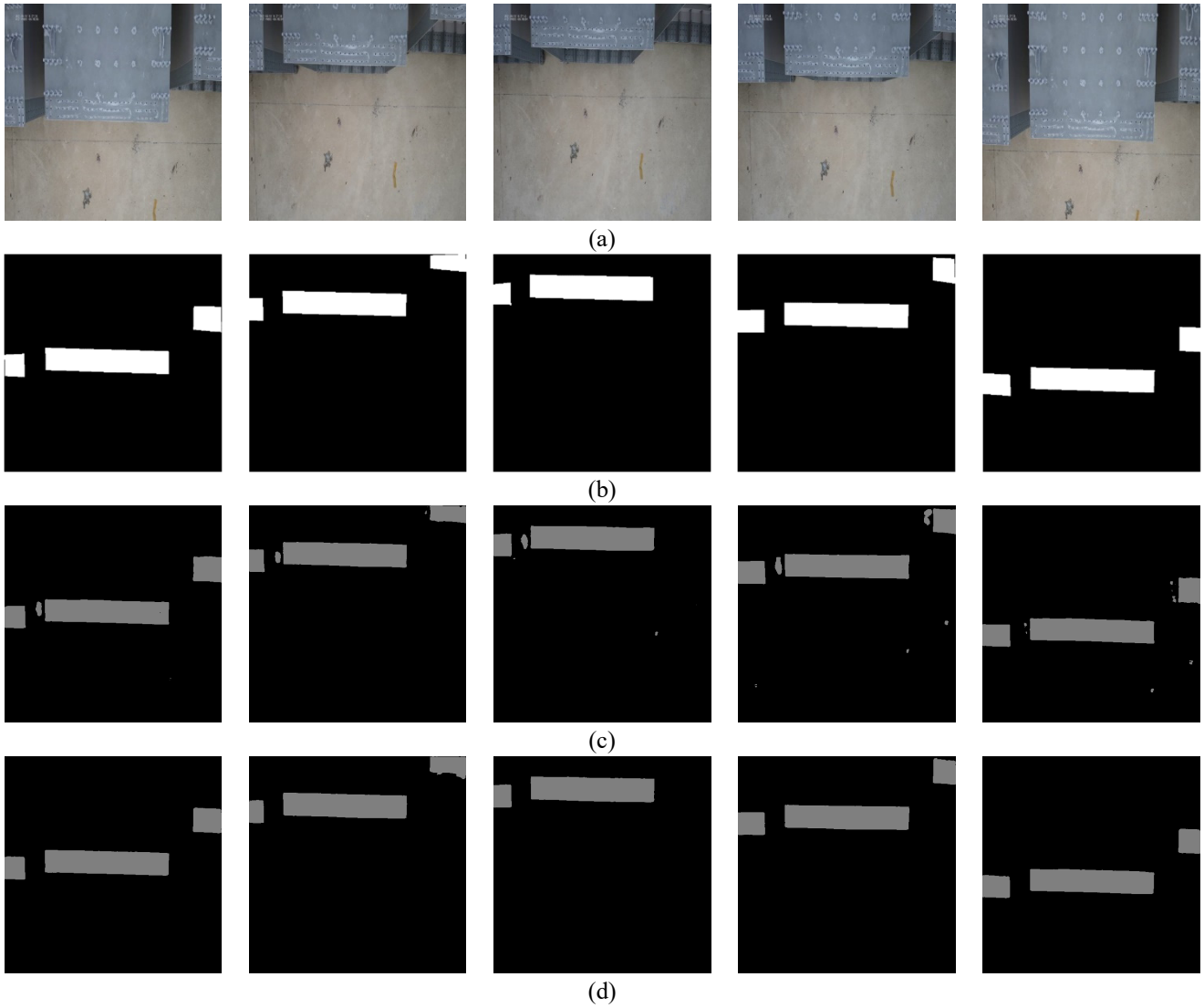


Fig. 11 Segmentation results of the field test. (a) Original images, (b) labeled images, and segmented PSS images obtained using (c) the conventional RFB (IoU: 92%) and (d) the proposed ERFP (IoU: 95%)

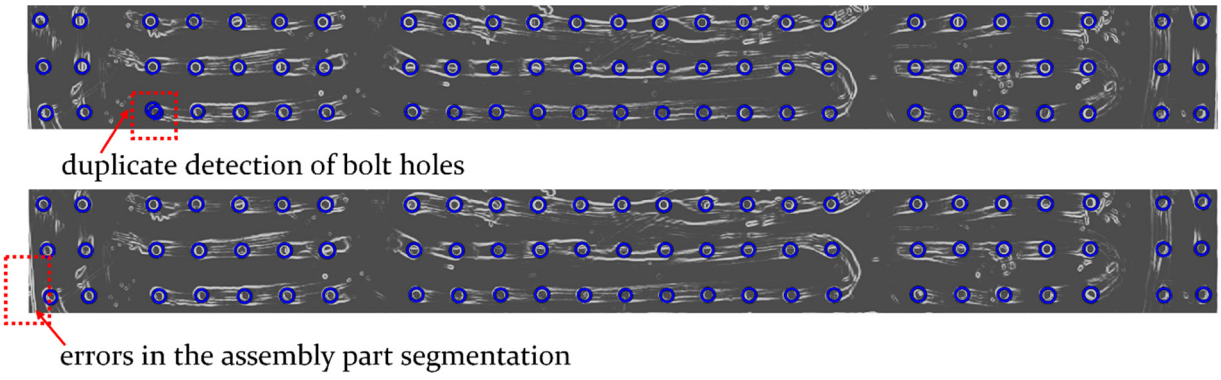


Fig. 12 Instances of assembly performance classification errors

performance achieved with the real-scale structure was evaluated. A kNN classifier, which attained the best performance for 3D models, was applied, and it produced a discrimination performance of 96% in the assembly quality evaluation. In cases of misclassified assembly results, errors may arise due to the assembly part segmentation errors or inaccuracies in detecting bolt holes such as duplicate detections (see Fig. 12). These inaccuracies can affect the calculation of input variables used for assembly performance evaluation. The field applicability of the

proposed algorithm for automating assembly performance evaluation in construction was demonstrated by the fact that the assembly segmentation (IoU) and dimensional quality evaluation results both had errors of less than 5% (see Table 3).

5. Conclusions

This study proposed a framework for assessing the assembly performance of PSSs using a vision sensor and machine learning with the aim of simplifying the preliminary assembly steps. The proposed method consisted of two steps: 1) assembly component segmentation using a modified version of the RFB method and 2) assembly performance evaluation using the positions of bolt holes calculated via various image processing methods and kNN. A convolution pyramid, which was factorized through channel modulation and an exploration layer with a large receptive field, was introduced to segment assembly components with high accuracy and to maintain real-time computing speed. An assembly quality evaluation for determining the performance of assembly was achieved by calculating the distance ratios between the bolt holes in the segmented components. By applying the kNN learning method, the assembly performance, which involved the classification of normal and defective PSS components, was evaluated. The results obtained with the kNN classifier were compared with those of other supervised learning classification methods, such as decision trees, discriminant analysis and SVMs classifiers.

To evaluate the performance of the proposed method, experimental tests were conducted with a 3D model and real-scale PSSs. These tests showed that the assembly components were detected with an IoU greater than 95%, and normal/defective structures were classified with an error of less than 5%. For the images obtained from the 3D model and onsite PSSs, the kNN classifier, which did not make any assumptions concerning the fundamental data distributions, achieved the best assembly performance classification results, with an accuracy exceeding 95%. Thus, the proposed framework with two different machine learning methods for semantic segmentation and quality classification can provide accurate dimensional quality evaluation results, enabling automated assembly performance evaluations for PSSs.

In the future, the proposed method will be improved by 1) improving its segmentation accuracy via the fusion of 3D point cloud and image data, 2) developing a bolt-hole position detection algorithm that is applicable to various types of PSS components, such as nonrectangular or curved surfaces, and 3) facilitating practical evaluations with full-scale PSSs for automation in construction scenarios.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1F1A1057721). It was also supported by the "National R&D Project for Smart Construction Technology (RS-2020-KA156007)" funded by the Korea Agency for Infrastructure Technology Advancement under the Ministry of Land, Infrastructure and Transport, and managed by the Korea Expressway Corporation. The authors extend special thanks to Daewoo ST for their cooperation.

References

Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017), "Segnet: A deep convolutional encoder- decoder architecture for image segmentation", *IEEE Trans. Pattern Anal. Mach. Intel.*, **39**(12), 2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615.

Bhatia, N. (2010), Survey of Nearest Neighbor Techniques, arXiv preprint arXiv:1007.0085.

Boafo, F.E., Kim, J.H. and Kim, J.T. (2016), "Performance of modular prefabricated architecture: Case study-based review and future pathways", *Sustain.*, **8**(6), 558. https://doi.org/10.3390/su8060558.

Chen, L.C., Papandreou, G., Schroff, F. and Adam, H. (2017), Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv preprint arXiv:1706.05587.

Choi, S., Myeong, W., Jeong, Y. and Myung, H. (2017), "Vision-based hybrid 6-DOF displacement estimation for precast concrete member assembly", *Smart Struct. Syst.*, **20**(4), 397-413. https://doi.org/10.12989/sss.2017.20.4.397.

Construction Workers Mutual Aid Association (2021), Trends in Construction Experienced Workers and Construction Skilled Workers, https://cwma.bigzine.kr/. (in Korean)

Cover, T. and Hart, P. (1967), "Nearest neighbor pattern classification", *IEEE Trans. Inform. Theory*, **13**(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H. and Wei, Y. (2017), "Deformable convolutional networks", *Proceedings of the IEEE International Conference on Computer Vision*, 764-773.

Deng, E.F., Zong, L., Ding, Y., Zhang, Z., Zhang, J.F., Shi, F.W., Cai, L.M. and Gao, S.C. (2020), "Seismic performance of mid-to-high rise modular steel construction-A critical review", *Thin Wall. Struct.*, **155**, 106924. https://doi.org/10.1016/j.tws.2020.106924.

Géron, A. (2019), *Hands-on Machine Learning with Scikit-Learn*, *Keras*, *and TensorFlow: Concepts*, *Tools*, *and Techniques to Build Intelligent Systems*, O'Reilly Media, Inc..

Hinton, G. and Sejnowski, T.J. (1999), *Unsupervised Learning: Foundations of Neural Computation*, MIT Press.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A. and Bengio, Y. (2017), "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 11-19.

Jeon, H., Kim, Y., Lee, D. and Myung, H. (2014), "Vision-based remote 6-DOF structural displacement monitoring system using a unique marker", *Smart Struct. Syst.*, **13**(6), 927-942. https://doi.org/10.12989/sss.2014.13.6.927.

Kim, M.K., Sohn, H. and Chang, C.C. (2014), "Automated dimensional quality assessment of precast concrete panels using terrestrial laser scanning", *Auto. Constr.*, **45**, 163-177. https://doi.org/10.1016/j.autcon.2014.05.015.

Kim, M.K., Wang, Q., Park, J.W., Cheng, J.C., Sohn,H. and Chang, C.C. (2016), "Automated dimensional quality assurance of full-scale precast concrete elements using laser scanning and BIM", *Auto. Constr.*, **72**, 102-114. https://doi.org/10.1016/j.autcon.2016.08.035.

Koem, C., Shim, C.S. and Park, S.J. (2016), "Seismic performance of prefabricated bridge columns with combination of continuous

mild reinforcements and partially unbonded tendons", *Smart Struct. Syst.*, **17**(4), 541-557. https://doi.org/10.12989/sss.2016.17.4.541.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Adv. Neur. Inform. Proc. Syst.*, **25**, 1.

Liu, S. and Huang, D. (2018), "Receptive field block net for accurate and fast object detection", *Proceedings of the European Conference on Computer Vision (ECCV)*, 385-400.

Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H. and Kautz, J. (2017), "Learning affinity via spatial propagation networks", *Advances in Neural Information Processing Systems*, **30**.

Long, J., Shelhamer, E. and Darrell, T. (2015), "Fully convolutional networks for semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431-3440.

McLachlan, G.J. (2005), *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley & Sons.

Mehta, S., Rastegari, M., Caspi, A., Shapiro, L. and Hajishirzi, H. (2018), "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation", *Proceedings of the European Conference on Computer Vision (ECCV)*, 552- 568.

Ministry of Land, Infrastructure, and Transport (2013), Road Bridge Standard Specification, https://www.codil.or.kr/.

Ministry of Land, Infrastructure, and Transport (2021), Guidelines for Activation of Smart Construction Technology (Notice No. 2021-1283), https://www.law.go.kr/.

Mitchell, T.M. (2007), *Machine Learning*, Volume 1, McGraw-Hill, New York.

Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A. and Brown, S.D. (2004), "An introduction to decision tree modeling", *J. Chemometr.: J. Chemometr. Soc.*, **18**(6), 275-285. https://doi.org/10.1002/cem.873.

Nguyen, D.C., Park, S.J. and Shim, C.S. (2022), "Digital engineering models for prefabricated bridge piers", *Smart Struct. Syst.*, **30**(1), 35-47. https://doi.org/10.12989/sss.2022.30.1.035.

Noble, W.S. (2006), "What is a support vector machine?", *Nat. Biotechnol.*, **24**(12), 1565-1567. https://doi.org/10.1038/nbt1206-1565.

Noh, H., Hong, S. and Han, B. (2015), "Learning deconvolution network for semantic segmentation", *Proceedings of the IEEE International Conference on Computer Vision*, 1520-1528.

Park, H.S., Lee, H., Adeli, H. and Lee, I. (2007), "A new approach for health monitoring of structures: terrestrial laser scanning", *Comput.-Aid. Civil Infrastr. Eng.*, **22**(1), 19-30. https://doi.org/10.1111/j.1467-8667.2006.00466.x.

Rahman, I.A., Memon, A.H. and Karim, A.T.A. (2013), "Relationship between factors of construction resources affecting project cost", *Modern Appl. Sci.*, **7**(1), 67-75.

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: Convolutional networks for biomedical image segmentation", *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference*, Munich, Germany, October.

Simonyan, K. and Zisserman, A. (2014), Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint arXiv:1409.1556.

Štefanič, M. and Stankovski, V. (2018), "A review of technologies and applications for smart construction", *Proc. Inst. Civil Eng.-Civil Eng.*, **172**, 83-87. https://doi.org/10.1680/jcien.17.00050.

Suthaharan, S. (2016), "Machine learning models and algorithms for big data classification", *Integr. Ser. Inf. Syst.*, **36**, 1-12. https://doi.org/10.1007/978-1-4899-7641-3.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015), "Going deeper with convolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1- 9.

Versaci, M. and Morabito, F.C. (2021), "Image edge detection: A new approach based on fuzzy entropy and fuzzy divergence", *Int. J. Fuzzy Syst.*, **23**(4), 918-936. https://doi.org/10.1007/s40815-020-01030-5.

Wang, H., Su, D., Liu, C., Jin, L., Sun, X. and Peng, X. (2019), "Deformable non-local network for video super-resolution", *IEEE Access*, **7**, 177734-177744. https://doi.org/10.1109/ACCESS.2019.2958030.

Wasim, M., Vaz Serra, P. and Ngo, T.D. (2020), "Design for manufacturing and assembly for sustainable, quick and cost-effective prefabricated construction-a review", *Int. J. Constr. Manage.*, 1-9. https://doi.org/10.1080/15623599.2020.1837720.

Xu, Z., Kang, R. and Lu, R. (2020), "3D reconstruction and measurement of surface defects in prefabricated elements using point clouds", *J. Comput. Civil Eng.*, **34**(5), 04020033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000920.

Yu, B., Jeon, H., Bang, H., Yi, S.S. and Min, J. (2022), "Fender segmentation in unmanned aerial vehicle images based on densely connected receptive field block", *Int. J. Nav. Arch. Ocean Eng.*, **14**, 100472. https://doi.org/10.1016/j.ijnaoe.2022.100472.

Yu, F. and Koltun, V. (2015), Multi-scale Context Aggregation by Dilated Convolutions, arXiv preprint arXiv:1511.07122.

Yu, F., Koltun, V. and Funkhouser, T. (2017), "Dilated residual networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 472-480.

Zonta, D., Pozzi, M. and Bursi, O.S. (2007), "Performance evaluation of smart prefabricated concrete elements", *Smart Struct. Syst.*, **3**(4), 475-494. https://doi.org/10.12989/sss.2007.3.4.475.

HJ