Smart Structures and Systems, *Vol. 27, No. 5 (2021) 783-793*
DOI: https://doi.org/10.12989/sss.2021.27.5.783

Detection and quantification of bolt loosening using RGB-D camera and Mask R-CNN

Junyeon Chung [a] and Hoon Sohn*

Department of Civil and Environmental Engineering, Korea Advanced Institute for Science and Technology, Daejeon 34141, South Korea

(*Received August 27, 2020, Revised December 3, 2020, Accepted January 15, 2021*)

Abstract. Bolt loosening is one of the most common types of damage for bolt-connected plates. Existing vision techniques detect bolt loosening based on the measurement of bolt rotation or the exposure of bolt threads. However, these techniques examine bolt tightness only in a qualitative manner, or require a reference measurement at the initially tightened state of the bolt for quantitative estimation. In this study, the exposed shank length of a bolt is quantitatively measured using an RGB-depth camera and a mask-region-based convolutional neural network but without requiring any measurement from the initial state of the bolt. The performance of the proposed technique is validated by conducting lab-scale experiments, in which the angle and distance of the camera are varied with respect to a target inspection area. The proposed technique successfully detects bolt loosening at exposed shank length over 3 mm with a resolution of 1 mm and 97% accuracy at different camera angles (40°–90°) and distances (up to 65 cm).

Keywords: bolt-loosening detection; bolt-loosening quantification; RGB-depth camera; Mask R-CNN; deep learning

1. Introduction

Bolting is one of the most common approaches for connecting steel plate components for aerospace, mechanical, and civil applications. Bolting connects two or more components by applying axial force on them using fasteners and the mating of screw threads. The Korea Express Corporation has reported that 33.3% of the steel bridges under their stewardship have bolt-related defects, such as bolt loosening, bolt failure, and missing bolts, and bolt loosening accounts for 58.1% of these defects (Korea Expressway Corporation 2013). Bolt loosening is caused by abrupt mechanical shocks, continuous vibrations, and/or thermal loading. Bolt loosening can result in the loss in preload and eventually cause system failure. Therefore, it is important to inspect the tightness of bolt connections to ensure the safety and integrity of bolted components.

Conventionally, trained inspectors periodically inspect bolt connections using a torque wrench. The inspectors check the tightness of bolts by comparing measured torque values with the specifications in a structural design. This manual inspection is quite accurate; however, it is time consuming and labor intensive. Furthermore, it is difficult to apply manual inspection to continuous monitoring and in difficult-to-reach areas (Wang *et al*. 2013a).

To overcome these limitations, several sensing technologies are introduced for online monitoring. For example, acoustic sensors, piezoelectric sensors, and electromechanical impedance sensors have been developed for monitoring bolt loosening (Wang *et al*. 2013b, Suda *et al*. 1992, Huynh *et al*. 2018, Huynh and Kim 2017, 2018). However, these techniques require the installation of numerous discrete sensors with all bolts that must be inspected. It is considerably difficult to supply power to these sensors and transmit data from these sensors. More importantly, the long-term reliability of these sensors has not yet been proven (Wang *et al*. 2013b). Furthermore, measured data change depending on temperature and humidity because these techniques are sensitive to environmental conditions (Wang *et al*. 2013b).

Vision techniques have been developed for bolt inspection with the goal of integrating these techniques with drones and robots. For example, Park *et al*. (2015) employed image processing algorithms, such as the Hough transform and Canny edge detector, to quantify the rotation angle of a loosened bolt. Zhao *et al*. (2019) estimated the rotation angle of a bolt by applying a machine learning algorithm referred to as the single-shot multibox detector (SSD). Huynh *et al*. (2019) proposed a bolt rotation detection technique using a regional convolution neural network (R-CNN) and the Hough line transform. Lab-scale tests were conducted to validate the performance of these techniques, and the results demonstrated that rotations as low as 5° could be detected. However, these vision techniques identify bolt loosening by comparing the current rotational angle of a bolt with respect to its initially measured angle or by detecting abnormal deviation of the current bolt angle with respect to all the other bolts that are all initially aligned in the same orientation. Furthermore, they can underestimate bolt loosening when bolt rotation is over 360°.

*Corresponding author, Ph.D., Professor,
E-mail: hoonsohn@kaist.ac.kr
[a] Graduate Student

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

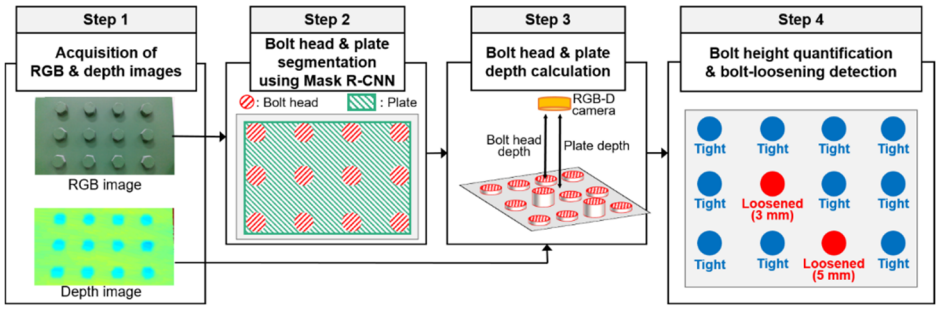


Fig. 1 Overall scheme of the proposed bolt-loosening detection and quantification technique

The threads of loosened bolts are captured by vision sensors and utilized to detect bolt loosening. For example, Cha *et al*. (2016) used a bolt image to determine the ratio of the exposed thread length to the bolt head radius and detected bolt loosening using a support vector machine. Ramana *et al*. (2019) advanced Cha's technique by integrating the Viola–Jones algorithm with a support vector machine to automate the bolt-loosening detection process. Zhang *et al*. (2019) adopted a faster regional convolutional neural network (Faster R-CNN) to localize and classify bolt loosening. These techniques can identify bolt loosening only when it is over 5 mm. Furthermore, the angle of a camera should be low to capture the images of bolt threads.

In this study, a new bolt-loosening detection and quantification technique is developed using a low-cost RGB-depth camera (RGB-D camera) and a mask regional convolutional neural network (Mask R-CNN). First, bolt heads and base plates are recognized using Mask R-CNN from the components of the RGB image obtained using the RGB-D camera. Then, the normal distances of the identified bolt heads and plates from the RGB-D camera are estimated using the depth components of the RGB-D images. Finally, the presence of loosened bolts is identified, and the level of bolt loosening is quantified by estimating the exposed shank length of each bolt. The major advantages of the proposed technique compared to the existing vision-based techniques are as follows: (1) Bolt loosening is detected, and the exposed shank length of the bolt is quantified. (2) Because the operational angle of the RGB-D camera used in this study is between 40°–90°, the proposed technique has a wider operational angle than the other existing techniques. Furthermore, the camera does not have to be placed closed to the inspection surface, making the proposed technique more suitable for integration with drones and robots.

The rest of the paper is organized as follows: The proposed bolt-loosening detection and quantification technique is presented in Section 2. The training of Mask R-CNN in the proposed technique is described in Section 3. The performance of the proposed technique is validated in Section 4 through lab-scale experiments. The concluding remarks and discussion are presented in Section 5.

2. Bolt-loosening detection and quantification technique

The proposed technique is composed of four steps, as presented in Fig. 1.

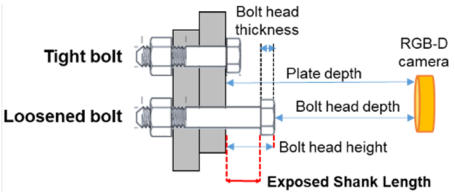


Fig. 2 Definition of exposed shank length

First, RGB and depth images are captured by an RGB-D camera and aligned with each other. Second, bolt heads and plate objects are extracted from the RGB images using Mask R-CNN. Third, the extracted bolt heads and plate objects are converted to point cloud data, and the depths (the normal distances from the RBG-D camera) of the bolt heads and plates are calculated. Fourth, loosened bolts are identified based on the exposed shank length, which is calculated by subtracting the bolt head thickness from the bolt head height, as depicted in Fig. 2.

2.1 Acquisition of RGB and depth images

The Intel RealSense D435i camera is used to simultaneously capture RGB and depth images. The camera consists of an RGB camera for capturing RGB images and a depth camera for acquiring depth images. This camera captures RGB and depth images using a global shutter, a maximum range of approximately 10 m, and a wide field of view. The specifications of the camera are presented in Table 1 (Grunnet-Jepsen *et al*. 2018). The depth accuracy of the camera improves as the distance between the camera and target object decreases. For example, the depth accuracy of the camera is approximately 2 mm at a distance of 30 cm and 4 mm at a distance of 1 m (Benkhoui *et al*. 2019).

Decimation and temporal filters are applied to the raw depth image to reduce noise and improve the precision of the measured depth. The decimation filter runs on a 2 × 2 kernel with a nonzero median, which samples the median value in the kernel while neglecting the zero values. The decimation filter reduces the size of the depth image to half of the original size (from 1280 × 720 to 640 × 360). In addition, holes (zero values) are filtered out because the filter uses pixels with only nonzero values. The temporal filter improves the precision of the depth image by averaging the pixel values over several frames with an exponential moving average, as shown below (Dubois and

Table 1 Specifications of Intel RealSense D435i (Grunnet-Jepsen *et al*. 2018, Benkhoui *et al*. 2019)

Shutter type	Global shutter	**Depth resolution**	1280 × 720 pixels
RGB field of view (horizontal × vertical)	69.4° × 42.5°	**Depth technology**	Active infrared stereo
Depth field of view (horizontal × vertical)	87° × 58°	**Depth accuracy**	2 mm at 30 cm 4 mm at 100 cm
Usage environment	Indoor/outdoor	**Maximum range**	10 m

Sabri 1984).

$$S_t = \begin{cases} Y_1 \\ \alpha Y_t + (1-\alpha) S_{t-1} \\ Y_t \end{cases} \quad \begin{array}{l} t = 1 \\ t > 1 \quad \text{and} \quad \Delta = |S_t - S_{t-1}| < \delta \\ t > 1 \quad \text{and} \quad \Delta = |S_t - S_{t-1}| > \delta \end{array} \tag{1}$$

where Y_t and S_t are the raw and filtered pixel values of the depth image at time t, respectively, α is the weight of each frame, and δ is a threshold for determining whether the difference between the pixel values of two consecutive frames is due to noise or not. In this study, α and δ are set as 0.5 and 20, respectively.

As the RGB and depth images are obtained from separate modules, these images should be aligned considering their relative positions (Hartley and Zisserman 2003).

$$\begin{bmatrix} X^d \\ Y^d \\ Z^d \end{bmatrix} = Z^d \begin{bmatrix} f_x^d & 0 & c_x^d \\ 0 & f_y^d & c_y^d \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u^d \\ v^d \\ 1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{bmatrix} = \begin{bmatrix} R_d^c & T_d^c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X^d \\ Y^d \\ Z^d \\ 1 \end{bmatrix} \tag{3}$$

$$\begin{aligned} &depth(u^c, v^c) \\ &= Depth\left(\frac{f_x^c \cdot X^c}{Z^c} + c_x^c, \frac{f_y^c \cdot Y^c}{Z^c} + c_y^c \right) = Z^c \end{aligned} \tag{4}$$

where subscripts/superscripts d and c denote the depth and RGB cameras, respectively. X, Y, and Z are the coordinates of the point cloud data in the corresponding camera (RGB camera or depth camera) coordinate system. u and v are the horizontal and vertical pixel coordinates in the corresponding image (RGB image or depth image) coordinate system. f_x and f_y are the focal lengths and c_x and c_y are the optical centers in the horizontal and vertical directions, respectively. R_d^c and T_d^c are the rotation and translation matrices from the depth camera to the RGB camera, respectively.

2.2 Bolt head and plate segmentation using Mask R-CNN

After the alignment of the RGB and depth images, Mask R-CNN is applied to the RGB image for bolt head extraction (He *et al*. 2017). Among various deep learning techniques, such as Faster R-CNN, YOLO, and SSD, Mask R-CNN is employed because it detects objects at the pixel level with shape information (He *et al*. 2017, Ren *et al*. 2015, Redmon and Farhadi 2018, Liu *et al*. 2016). Mask R-CNN consists of four steps, as shown in Fig. 3. The detailed architectures of Mask R-CNN are presented in Appendix A. First, a backbone network extracts feature maps from input images. Second, a region proposal network (RPN) is applied to the feature maps to generate regions of interest (RoIs). Third, RoIAlign is used to adjust the sizes of the feature maps corresponding to the RoIs such that all sizes are the same. Finally, the object detection network in Mask R-CNN identifies bolt heads by placing rectangular boxes,

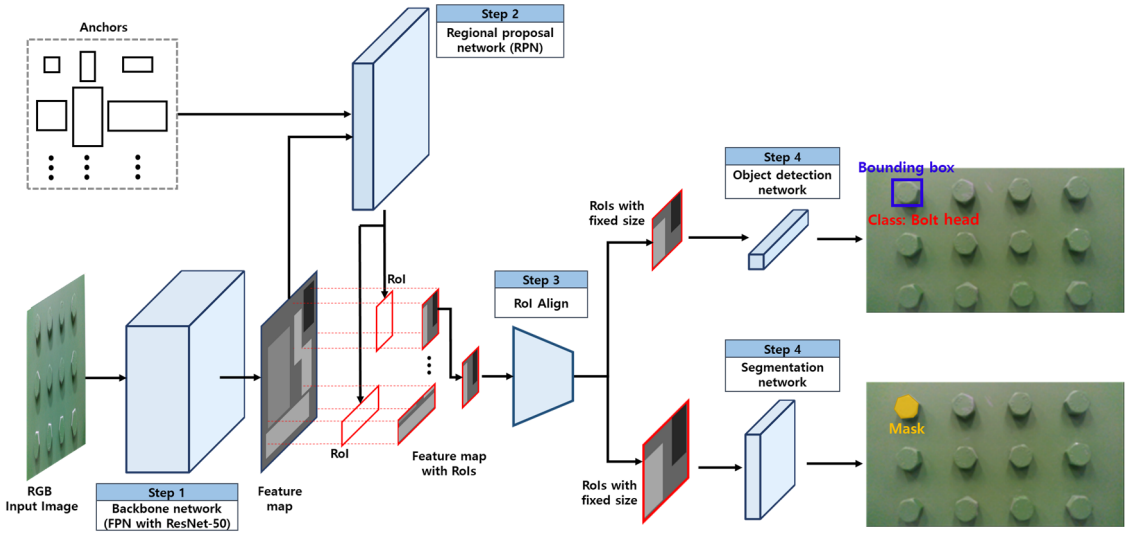


Fig. 3 Overall process of Mask R-CNN

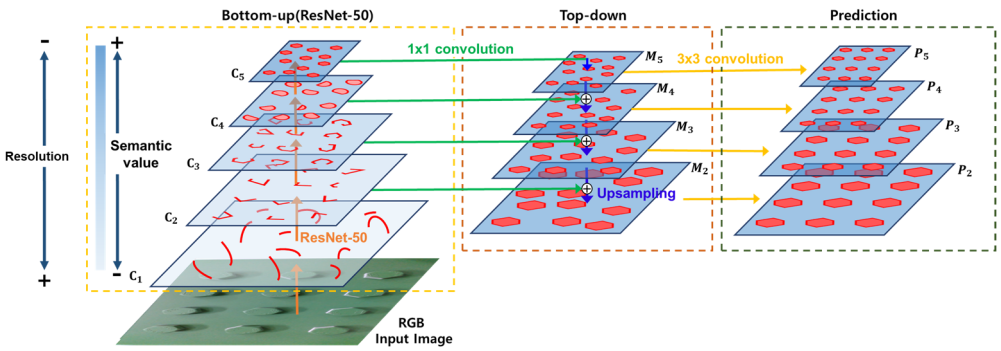


Fig. 4 Architecture of the backbone network (FPN with ResNet-50)

which are referred to as bounding boxes, around the bolt heads. A segmentation network quantifies the shapes and sizes of the identified bolt heads at the pixel level.

The objective of the backbone network is to extract feature maps with rich semantic values and a high pixel resolution for bolt head detection. In this study, the feature pyramid network (FPN) architecture with residual network-50 (ResNet-50) is adopted as the backbone network (Lin *et al*. 2017, He *et al*. 2016).

The backbone network operation consists of three steps, as shown in Fig. 4: (1) bottom-up mapping, (2) top-down mapping, and (3) prediction. Bottom-up mapping is implemented using ResNet-50. As the feature maps move to the upper layers of ResNet-50, the semantic values of the feature maps increase at the cost of decreased pixel resolution. The objective of top-down mapping is to improve pixel resolution while maintaining the previously achieved semantic values by merging two adjacent feature maps (C_i and M_{i+1}) into a new feature map (M_i) through 1 × 1 convolution and upsampling. Here, the reconstructed feature maps (M_i) can be distorted owing to the aliasing effect of upsampling. 3 × 3 convolution is applied to the feature map (M_i) to reduce the aliasing effect, and the final feature map (P_i) is generated in the prediction step.

Next, the RPN considers the final feature map (P_i) as the input and outputs the RoIs that indicate the locations and sizes of potentially meaningful objects within the feature map. The RoIs are obtained using anchors, which are a set of boxes with predefined heights and widths. Each of these anchors is scanned over the feature map to obtain its objectness score and location adjustment value. Objectness is defined as the possibility of capturing a meaningful object within an anchor, and the location adjustment value indicates how much the location and size of the anchor should be adjusted to match the object. The locations and sizes of the anchors with high objectness scores are adjusted according to the location adjustment values, and the selected anchors are defined as RoIs.

Then, RoIAlign considers the feature map segments corresponding to the RoIs as the inputs and resizes all of them to the same size because the subsequent object detection and segmentation networks require inputs with a fixed size.

The object detection network classifies any object within the RoIs into either bolt heads or backgrounds (no objects) and slightly adjusts the locations of the identified bolts. The segmentation network performs the pixel-level classification of the identified bolts to quantify their shapes and areas.

The base plate is identified after bolts are identified, as shown in Fig. 5. First, the four vertices (top-left, top-right, bottom-right, and bottom-left corners) of the plate boundary are extracted. For example, the top-left corner of the bounding box of the top-left bolt constitutes the top-left corner of the plate boundary. The other corners of the plate boundary are defined in a similar manner. The plate area is extracted by subtracting the bolt areas from the area enclosed by the four vertices.

2.3 Depth calculation for bolt head and plate

The 3D coordinates of the extracted bolt heads and plate are obtained from the depth information, as follows (Hartley and Zisserman 2003)

$$\begin{bmatrix} X_{obj}^c \\ Y_{obj}^c \\ Z_{obj}^c \end{bmatrix} = depth(u_{obj}^c, v_{obj}^c) \begin{bmatrix} f_x^c & 0 & c_x^c \\ 0 & f_y^c & c_y^c \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_{obj}^c \\ v_c^c \\ 1 \end{bmatrix} \quad (5)$$

where X_{obj}^c, Y_{obj}^c, and Z_{obj}^c represent the 3D coordinates of an object (bolt head or plate) in the RGB camera coordinate system, and u_{obj}^c and v_{obj}^c are the horizontal and vertical pixel values of the object in the RGB image coordinate system. f_x^c, f_y^c are the focal lengths of the RGB camera module, and c_x^c, c_y^c are the optical centers of the RGB camera module in the horizontal and vertical directions, respectively.

Next, the plane surfaces that represent each bolt head and plate are extracted using the well-established random sample and consensus (RANSAC) model fitting algorithm (Fischler and Bolles 1981). The RANSAC algorithm consists of two iterative steps. In the first hypothesis generation step, a hypothesis plane that connects three data points is generated. The points are randomly selected from the target object. In the second verification step, each of the remaining points is tested to determine if it is an inlier with respect to the hypothesis plane or not. Through the iterations of these two steps, the plane that consists of the largest number of inliers is selected as the fitted plane. The planes obtained using the RANSAC algorithm are shown in Fig. 6.

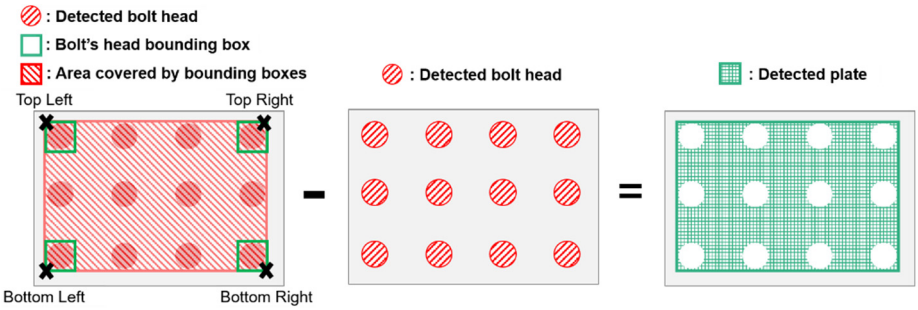


Fig. 5 Plate segmentation

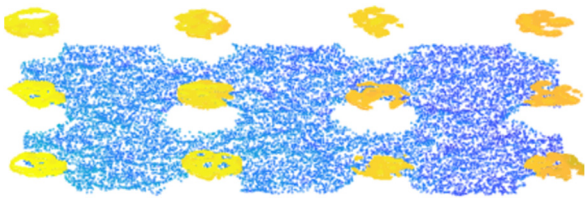


Fig. 6 Fitted planes of the plate and bolt heads

The depths of the bolt heads and plate are defined as the normal distance from the camera center to their fitted planes.

2.4 Exposed shank length quantification and bolt-loosening detection

As shown in Fig. 2, the bolt head height is defined as the normal distance from the plate to the top of the bolt head. The bolt head height is determined by subtracting the bolt head depth from the plate depth. Then, the exposed shank length, which represents the level of bolt loosening, is estimated by subtracting the bolt head thickness from the bolt head height. As the bolt size is standardized according to the American National Standards Institute, the relation between the bolt head thickness and the outer diameter of the bolt head can be estimated using polynomial regression. Therefore, the bolt head thickness can be determined by measuring the outer diameter of the bolt head.

Bolt loosening is identified by comparing the estimated exposed shank length with the threshold defined in Eq. (6). The bolt is determined as loosened if the exposed shank length is larger than the predefined threshold value. Assuming that the exposed shank length of a tight bolt follows a normal distribution, the upper limit corresponding to a 97% two-sided confidence level is set as the threshold value.

$$\text{exposed shank length} > \text{threshold} = 2.17\sigma \tag{6}$$

where σ is the standard deviation of the exposed shank lengths estimated from tight bolts.

3. Training of Mask R-CNN

3.1 Loss functions

The total training loss (L_{total}) of Mask-RCNN is the sum of the training losses of the RPN (L_{RPN}), object detection network (L_{OBJ}), and segmentation network (L_{SEG}) (He *et al*. 2017).

$$L_{TOTAL} = L_{RPN} + L_{OBJ} + L_{SEG} \tag{7}$$

where L_{RPN} and L_{OBJ} consist of classification loss (L_{cls}) and regression loss (L_{reg}) and L_{SEG} consists of mask loss (L_{mask}). Each loss function is computed as follows (He *et al*. 2017)

$$L_{RPN} = \frac{1}{N}\sum_i L_{cls}(p_i, p_i^*) + \lambda_1 \frac{1}{N}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{8}$$

$$L_{OBJ} = \frac{1}{N}\sum_i L_{cls}(p_i, p_i^*) + \lambda_2 \frac{1}{N}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{9}$$

$$L_{SEG} = \gamma_1 \frac{1}{N}\sum_i L_{mask}(s_i, s_i^*) \tag{10}$$

where hyperparameters λ_1, λ_2, and γ_1 balance the training losses of the three parts. N represents the number of bounding boxes. The bounding boxes represent the anchors in L_{RPN} and the RoIs in L_{OBJ} and L_{SEG}. p_i and p_i^* represent the predicted and ground truth classification probabilities of the i^{th} bounding box, respectively. t_i represents the predicted average difference between the i^{th} bounding box and ground truth box in terms of (1) the horizontal and vertical coordinates of the bounding box center and (2) the width and height of the bounding box. Similarly, t_i^* represents the ground truth value of the average difference between the i^{th} bounding box and ground truth box. s_i and s_i^* represent the predicted and ground truth binary matrices, respectively. L_{cls}, L_{reg}, and L_{mask} are derived from the following equations (He *et al*. 2017)

$$L_{cls}(p_i, p_i^*) = -p_i^* \log p_i \tag{11}$$

$$L_{reg}(t_i, t_i^*) = \text{smooth}_{L1}(t_i^* - t_i)$$
$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{12}$$

$$L_{mask}(s_i, s_i^*) = -(s_i^* \log(s) + (1 - s_i^*)\log(1 - s)) \tag{13}$$

3.2 Dataset

The dataset used in this study consists of 100 RGB

Fig. 7 Representative images of bolt head dataset (bolt heads indicated by polygon shapes)

images including 782 bolt head images. The images were captured from actual steel bridges or test specimens. 80% of the dataset is used for training, and the remaining 20% is used for validation. The representative images of the dataset are shown in Fig. 7.

3.3 Training

Transfer learning and data augmentation are implemented to improve the efficiency of network training with a limited training dataset (Torrey and Shavlik 2010, Van Dyk and Meng 2001). Transfer learning first borrows the pretrained weights of the backbone network (FPN + ResNet-50) and then fine tunes the weights using the bolt head dataset. The weights of the backbone network pretrained with the COCO dataset are initially assigned to the backbone network of the Mask R-CNN model. The weights trained with the COCO dataset can effectively extract features from various objects because the dataset consists of 318,000 images including 91 categories (Lin *et al*. 2014). Then, the remaining networks (RPN, object detection network, and segmentation network) are trained with the bolt head dataset to specifically detect the bolt heads and generate the corresponding polygon masks. Once the remaining networks are trained, the backbone network is also trained with the bolt head dataset. To prevent overfitting, the number of images during training is increased by data augmentation, including random rotation and flips at each iteration (Van Dyk and Meng 2001).

All algorithms are implemented in Python, and the training is carried out on a workstation with NVIDIA GTX 1080 Ti graphic cards and 16 GB RAM. The network is trained for 35 epochs at 100 iterations per epoch. The learning rate, weight decay, and momentum are selected as 0.001, 0.0001, and 0.9, respectively, based on the work by He *et al*. (2017).

Fig. 8 shows the training of the proposed Mask R-CNN model. The decrease in the training loss slows down after approximately 1,000 iterations. After 3,000 iterations, the improvement in the training loss becomes negligible and the validation loss converges to approximately 0.7. Finally, the mean intersection over union rate for validation dataset is 82% indicating that the predicted class, bounding box, and mask are well matched to actual labels.

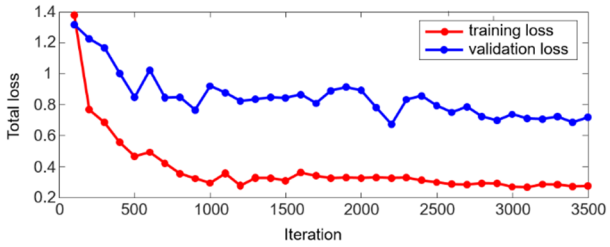


Fig. 8 Training progress of Mask R-CNN

4. Experimental validation

Lab-scale experiments were performed under four different conditions to validate the performance of the proposed technique. In Case 1, different camera angles and distances were considered to examine the operating range of the proposed technique in terms of camera angles and distances and to establish a threshold exposed shank length for bolt-loosening identification. In Case 2, bolts with various exposed shank lengths were tested to validate the accuracy of bolt-loosening detection. In Case 3, different lighting conditions were considered to examine the stability of the performance of the proposed technique. In Case 4, the proposed technique was investigated during eight different time points to detect gradual bolt loosening with exposed shank length estimation.

4.1 Experimental setup

The overall experimental setup is shown in Fig. 9. The dimensions of the specimen plate were 60 cm (width) × 45 cm (height), and twelve M30 bolts were placed on the plate. The outer diameter and head thickness of an M30 bolt were 53.1 mm and 19 mm, respectively. The horizontal and vertical spacing between the bolts were 12 cm and 11 cm, respectively. The camera angle represents the angle between $\overline{C_P C_L}$ and $\overline{C_P H_P}$, and the camera distance represents the length of $\overline{C_P C_L}$. Intel RealSense D435i was used for capturing RGB and depth images. The camera was placed on a stand, and images were captured by varying the camera angle and distance. The ground truth exposed shank length was measured using Vernier calipers.

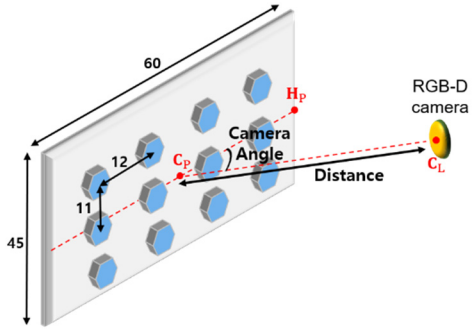
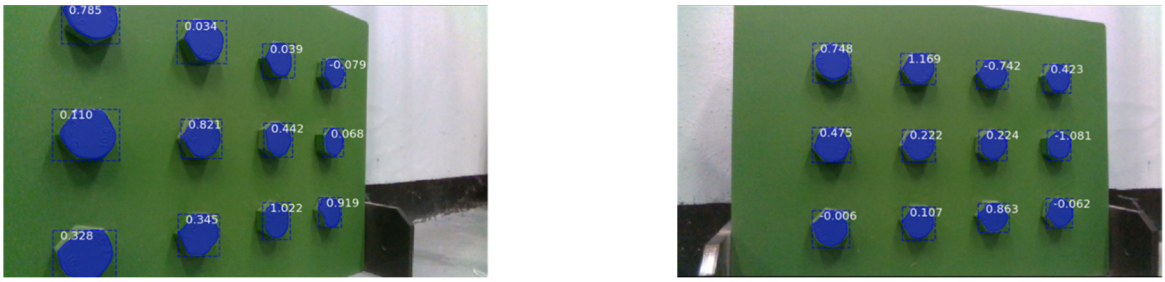


Fig. 9 Experimental setup (unit: cm)



(a) Angle: 40°/Distance: 50 cm (b) Angle: 80°/Distance: 65 cm

Fig. 10 Representative images obtained by RGB-D camera at different angles and distances

Table 2 RMSE of estimated exposed shank length at different camera angles

Angle (°)	Distance (cm)	RMSE (mm)
40	35–80	0.91
50	35–80	0.90
60	35–80	0.93
70	35–80	0.74
80	35–80	0.97
90	35–80	0.95

Table 3 RMSE of estimated exposed shank length at different camera distances

Distance (cm)	Angle (°)	RMSE (mm)
35	40–90	0.49
40	40–90	0.52
45	40–90	0.46
50	40–90	0.49
55	40–90	0.53
60	40–90	0.88
65	40–90	0.99
70	40–90	1.24
75	40–90	1.48
80	40–90	2.36

4.2 Experimental results

4.2.1 Case 1: Different camera angles and distances

In Case 1, the images of twelve tight M30 bolts (exposed shank length was 0 mm) were obtained by varying the camera angle from 40° to 90° at intervals of 10° and the camera distance from 35 cm to 80 cm at intervals of 5 cm. Next, exposed shank lengths were estimated based on the proposed technique, and the root mean square errors (RMSEs) between the estimated exposed shank length and true exposed shank length were calculated.

Figs. 10(a) and (b) show the representative images obtained under two different conditions (angle of 40° and distance of 50 cm, and angle of 80° and distance of 65 cm). The area of the tight bolts is colored in blue, and the estimated exposed shank lengths are provided in mm.

Table 2 shows that the RMSEs for different camera angles are similar. In contrast, Table 3 shows that the RMSE increases with the camera distance. The RMSE is approximately 0.5 mm up to a camera distance of 55 cm and reaches 2.36 mm at a camera distance of 80 cm. Based on these results, we conclude that the proposed technique can be applied under camera angles ranging from 40° to 90° and camera distances of up to 65 cm to achieve a exposed shank length estimation accuracy of 1 mm.

Considering all camera angles and distances, the mean and standard deviation of the estimated exposed shank length of the tight bolts are -0.05 mm and 0.62 mm. Therefore, the threshold exposed shank length for loosened bolt detection is set to 1.34 mm, which is the upper limit of a 97% two-sided confidence level.

4.2.2 Case 2: Different exposed shank lengths

In Case 2, the images of the M30 bolts with exposed shank lengths of 0 mm, 3 mm, and 6 mm were obtained under various camera angles (40°–90°) and distances (35 cm–65 cm). The RMSE between the estimated and true exposed shank lengths was computed, and the bolts whose exposed shank lengths were larger than 1.34 mm were identified as loosened.

Fig. 11(a) shows an RGB image with ground truth exposed shank lengths. Fig. 11(b) shows the loosened bolts and their exposed shank lengths estimated by the proposed technique. The loosened and tight bolts are marked in red and blue, respectively. These figures show that all loosened bolts with exposed shank lengths of 3 mm or more are successfully detected using the proposed technique.

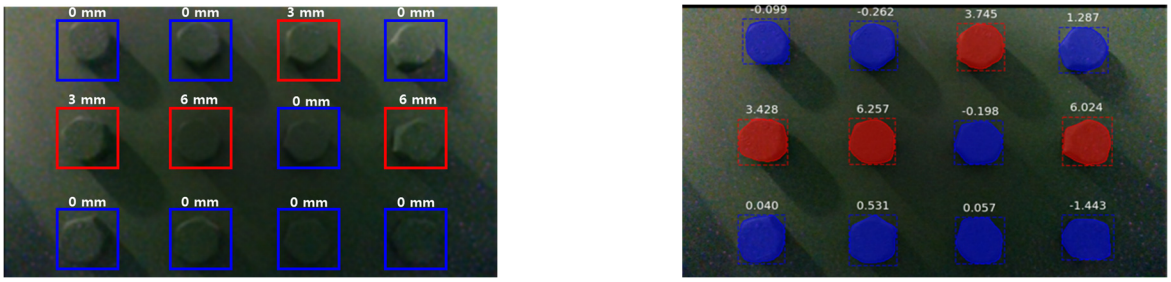
The results for Case 2 are summarized in the first three rows of Table 4. The RMSEs of the estimated exposed shank lengths are 0.62 mm, 0.66 mm, and 0.53 mm for the bolts with exposed shank lengths of 0 mm, 3 mm, and 6 mm, respectively. The accuracies of bolt-loosening detection are 99.19%, 97.99%, and 100% for the bolts with exposed shank lengths of 0 mm, 3 mm, and 6 mm, respectively.

4.2.3 Case 3: Different lighting conditions

The experiments for Cases 1 and 2 were conducted under bright lighting conditions (140 lx). In Case 3, additional experiments were conducted under darker lighting conditions (60 lx and 20 lx) to examine the

(a) Ground truth (b) Result of the proposed technique

Fig. 11 Identification of loosened bolts under bright lighting condition (140 lx)



(a) Ground truth (b) Result of the proposed technique

Fig. 12 Identification of loosened bolts under dark lighting condition (20 lx)

applicability of the proposed technique under different lighting conditions. Except for the lighting conditions, all other experimental parameters were identical to the previous cases.

Fig. 12(a) shows the RGB images captured under a dark lighting condition (20 lx), and Fig. 12(b) shows the loosened bolts and their exposed shank lengths estimated by the proposed technique. Similar to Case 2, the bolts with exposed shank lengths more than 1.34 mm are identified as loosened.

The results for Case 3 are summarized in the last six rows of Table 4. The RMSEs for Cases 2 and 3 are similar. Under an illumination of 60 lx, the RMSEs of the estimated exposed shank lengths are 0.69 mm, 0.53 mm, and 0.68 mm for the bolts with exposed shank lengths of 0 mm, 3 mm, and 6 mm, respectively. Additionally, the accuracies of bolt-loosening detection are 97.99%, 98.39%, and 100% for the bolts with exposed shank lengths of 0 mm, 3 mm, and 6 mm, respectively. Under an illumination of 20 lx, the RMSEs of the estimated exposed shank lengths are 0.59 mm, 0.60 mm, and 0.67 mm for the bolts with exposed shank lengths of 0 mm, 3 mm, and 6 mm, respectively. Moreover, the accuracies of bolt-loosening detection are 98.59%, 98.79%, and 100% for the bolts with exposed shank lengths of 0 mm, 3 mm, and 6 mm, respectively.

Table 4 Bolt-loosening detection results under different lighting conditions

Illumination (lx)	Exposed shank length (mm)	No. of bolts (EA)	Estimated exposed shank length		Loosened bolt detection		
			Mean (mm)	RMSE (mm)	No. of tight bolts (EA)	No. of loosened bolts (EA)	Accuracy (%)
Bright (140)	0	498	-0.05	0.62	494	4	99.19
	3	498	2.88	0.66	10	488	97.99
	6	498	6.10	0.53	0	498	100
Middle (60)	0	498	0.15	0.69	488	10	97.99
	3	498	3.05	0.53	8	490	98.39
	6	498	5.85	0.68	0	498	100
Dark (20)	0	498	0.08	0.59	491	7	98.59
	3	498	2.94	0.60	6	492	98.79
	6	498	5.91	0.67	0	498	100

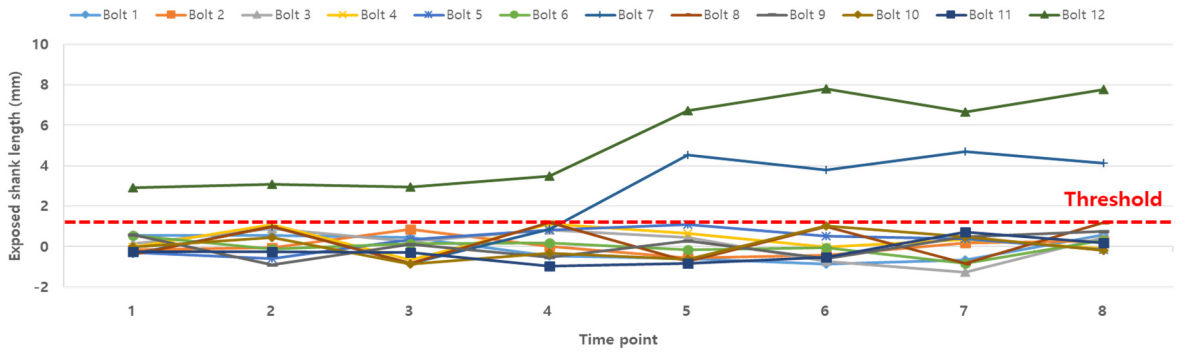


Fig. 13 Exposed shank length measurements under 90° camera angle, 65 cm camera distance and 140 lx brightness

(a) 1st inspection time point (b) 5th inspection time point

Fig. 14 Identification of loosened bolts for case 4 shown in Fig. 13

4.2.4 Case 4: Gradual bolt loosening

In Case 4, the RGB-D images of the twelve bolts on the connected plate were captured eight times to detect gradual loosening of the bolts over a certain time period. These images were obtained with under 90° camera angle, 65 cm camera distance and 140 lx brightness. Among the twelve bolts, one bolt (bolt 12) was initially loosened with 3 mm exposed shank length, while the other bolts (bolt 1-11) were initially tightened. Then, between 4th and 5th inspection time points, the exposed shank lengths of bolts 7 and 12 were increased from 0 mm to 4 mm and from 3 mm to 7 mm, respectively.

Fig. 13 shows that the exposed shank length of bolt 12 initially ranged between 2.90 mm and 3.48 mm up to the 4th inspection time point and increased to 6.72 mm after the 5th inspection time point, exceeding the threshold of 1.34 mm. The exposed shank length of bolt 7 was initially below the threshold and increased to 4.54 mm after the 5th inspection time point. The exposed shank length of all the other bolts remained below the threshold for the entire period. Fig. 14 display the detected loosened bolts, bolt 12 at the 1st inspection time point, and bolts 7 and 12 at the 5th inspection time point.

5. Conclusions

This paper proposed a bolt-loosening detection and quantification technique using a low-cost RGB-D camera and Mask R-CNN. First, the bolt heads in a plate were identified by applying Mask R-CNN to the RGB images captured by the RGB-D camera. Second, the exposed shank length was estimated using the depth images obtained from the RGB-D camera. A bolt was identified as loosened if the exposed shank length exceeded a user-specified threshold value.

The experiments performed on the test specimens indicated the following: (1) The RMSEs of the estimated exposed shank lengths were within 1 mm when the camera angle was varied from 40° to 90° and the camera distance was increased up to 65 cm. (2) When the exposed shank length was more than 3 mm, the bolt was detected as loosened with 97% accuracy regardless of lighting conditions.

It should be noted that the loss in preload cannot be estimated simply based on the exposed shank length. The rotation angle of the bolt should be considered along with the exposed shank length to quantify the preload loss of bolt connections. This should be investigated in the future with a field test.

Acknowledgments

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2019R1A3B3067987).

References

Benkhoui, Y., El Korchi, T. and Reinhold, L. (2019), "UAS-based crack detection using stereo cameras: a comparative study", *Proceedings of International Conference on Unmanned Aircraft Systems* (*ICUAS*), Atlanta, GA, USA, June.

Cha, Y.-J., You, K. and Choi, W. (2016), "Vision-based detection of loosened bolts using the Hough transform and support vector

machines", *Autom. Constr.*, **71**(2), 181-188. https://doi.org/10.1016/j.autcon.2016.06.008

Dubois, E. and Sabri, S. (1984), "Noise reduction in image sequences using motion-compensated temporal filtering", *IEEE Trans. Commun.*, **32**(7), 826-831. https://doi.org/10.1109/TCOM.1984.1096143

Fischler, M.A. and Bolles, R.C. (1981), "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Commun. ACM*, **24**(6), 381-395. https://doi.org/10.1145/358669.358692

Grunnet-Jepsen, A., Sweetser, J.N., Winer, P., Takagi, A. and Woodfill, J. (2018), "Projectors for Intel® RealSenseTM Depth Cameras D4xx", Intel.

Hartley, R. and Zisserman, A. (2003), *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, United Kingdom.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017), "Mask R-CNN", *IEEE International Conference on Computer Vision*, Venice, Italy, October.

Huynh, T.-C. and Kim, J.-T. (2017), "Quantification of temperature effect on impedance monitoring via PZT interface for prestressed tendon anchorage", *Smart Mater. Struct.*, **26**(12), 125004. https://doi.org/10.1088/1361-665X/aa931b

Huynh, T. and Kim, J. (2018), "RBFN-based temperature compensation method for impedance monitoring in prestressed tendon anchorage", *Struct. Control Health Monit.*, **25**(6), e2173. https://doi.org/10.1002/stc.2173

Huynh, T.-C., Dang, N.-L. and Kim, J.-T. (2018), "Preload monitoring in bolted connection using piezoelectric-based smart interface", *Sensors*, **18**(9), 2766. https://doi.org/10.3390/s18092766

Huynh, T.-C., Park, J.-H., Jung, H.-J. and Kim, J.-T. (2019), "Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing", *Autom. Constr.*, **105**, 102844. https://doi.org/10.1016/j.autcon.2019.102844

Korea Expressway Corporation (2013), "Improvement of bridge inspection system by the damage analysis", Korea Expressway Corporation.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J, Perona, P., Ramanan, D., Dollár, P. and Zitnick C.L. (2014), "Microsoft COCO: common objects in context", *Proceedings of European Conference on Computer Vision*, Zurich, Switzerland, September.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017), "Feature pyramid networks for object detection", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. and Cheng-Yang, F. (2016), "SSD: single shot multibox detector", *Proceedings of European Conference on Computer Vision*, Amsterdam, The Netherlands, October.

Park, J.-H., Huynh, T.-C., Choi, S.-H. and Kim, J.-T. (2015), "Vision-based technique for bolt-loosening detection in wind turbine tower", *Wind Struct., Int. J.*, **21**(6), 709-726. https://doi.org/10.12989/was.2015.21.6.709

Ramana, L., Choi, W. and Cha, Y.-J. (2019), "Fully automated vision-based loosened bolt detection using the Viola–Jones algorithm", *Struct. Health Monit.*, **18**(2), 422-434. https://doi.org/10.1177/1475921718757459

Redmon, J. and Farhadi, A. (2018), "YOLOv3: An incremental improvement", arXiv Prepr. arXiv1804.02767.

Ren, S., He, K., Girshick, R. and Sun, J. (2015), "Faster R-CNN: towards real-time object detection with region proposal networks", *Neural Inform. Process. Syst.*, Montreal, Canada, December.

Simonyan, K. and Zisserman, A. (2014), "Very deep convolutional networks for large-scale image recognition", arXiv Prepr. arXiv1409.1556.

Suda, M., Hasuo, Y., Kanaya, A., Ogura, Y. Takishita, T. and Suzuki, Y. (1992), "Development of ultrasonic axial bolting force inspection system for turbine bolts in thermal power plants", *JSME Int. J. Ser. 1, Solid Mech. Strength Mater.*, **35**(2), 216-219. https://doi.org/10.1299/jsmea1988.35.2_216

Torrey, L. and Shavlik, J. (2010), "Transfer learning", in Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, IGI Global, PA, USA.

Van Dyk, D.A. and Meng, X.-L. (2001), "The art of data augmentation", *J. Comput. Graph. Stat.*, **10**(1), 1-50. https://doi.org/10.1198/10618600152418584

Wang, T., Song, G., Liu, S., Li, Y. and Xiao, H. (2013a), "Review of bolted connection monitoring", *Int. J. Distrib. Sens. Netw.*, **9**(12), 871213. https://doi.org/10.1155/2013/871213

Wang, T., Song, G., Wang, Z. and Li, Y. (2013b), "Proof-of-concept study of monitoring bolt connection status using a piezoelectric based active sensing method", *Smart Mater. Struct.*, **22**(8), 87001. https://doi.org/10.1088/0964-1726/22/8/087001

Zhang, Y., Sun, X., Loh, K.J., Su, W., Xue, Z. and Zhao, X. (2019), "Autonomous bolt loosening detection using deep learning", *Struct. Health Monit.*, **19**(1), 105-122. https://doi.org/10.1177/1475921719837509

Zhao, X., Zhang, Y. and Wang, N. (2019), "Bolt loosening angle detection technology using deep learning", *Struct. Control Health Monit.*, **26**(1), e2292. https://doi.org/10.1002/stc.2292

HJ

Appendix A

Table 5 Parameters of anchors for RPN in Fig. 3

Parameter	Value
Anchor size	16, 32, 64, 128
Anchor stride	4, 8, 16, 32
Anchor ratio	0.5:1, 1:1, 2:1

Table 6 Architecture of RPN in Fig. 3

Type	Filter size	Stride	Padding	Depth
Shared RPN layer				
Convolution + ReLU	3 × 3	1	1	512
Classification RPN layer				
Convolution + Softmax	1 × 1	1	0	6
Bounding box adjustment RPN layer				
Convolution + Linear	1 × 1	1	0	12

Table 7 Architecture of object detection network in Fig. 3

Type	Filter size	Stride	Padding	Depth
Shared layer				
Convolution + ReLU	7 × 7	1	3	1024
Fully connected+ ReLU	-	-	-	1024
Classification layer				
Fully connected + Softmax	-	-	-	2
Bounding box adjustment layer				
Fully connected + Linear	-	-	-	4

Table 8 Architecture of segmentation network in Fig. 3

Type	Filter size	Stride	Padding	Depth
Segmentation layers				
Convolution + ReLU	3 × 3	1	1	256
Convolution + ReLU	3 × 3	1	1	256
Convolution + ReLU	3 × 3	1	1	256
Convolution + ReLU	3 × 3	1	1	256
Deconvolution + ReLU	2 × 2	2	0	256
Convolution + Softmax	1 × 1	1	1	2