

Research Article

Detection of the Bolt Loosening Angle through Semantic Key Point Extraction Detection by Using an Hourglass Network

Yu Ying ^{1,2,3} Wei Shichuan ¹ and Zhao Wei ⁴

¹Department of Civil Engineering, Shantou University, Shantou, Guangdong Province, China

²Guangdong Engineering Center for Structure Safety and Health Monitoring, Shantou University, Shantou, Guangdong Province, China

³Key Laboratory of Structure and Wind Tunnel of Guangdong Higher Education Institutes, Shantou, Guangdong Province, China

⁴College of Road and Bridge, Zhejiang Institute of Communications, Hangzhou, Zhejiang Province, China

Correspondence should be addressed to Zhao Wei; zhaowei7811@126.com

Received 30 November 2022; Revised 3 April 2023; Accepted 5 April 2023; Published 30 April 2023

Academic Editor: Ka-Veng Yuen

Copyright © 2023 Yu Ying et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Damage in bolts, which are used as connecting fasteners in steel structures, affects structural safety. Sophisticated machine vision methods have been formulated for the detection of loose bolts, but their accuracy remains an area for improvement. In this paper, a method based on a stacked hourglass network is proposed for automatically extracting the key points of a bolt and for obtaining the bolt loosening angle by comparing the rotations of the key points before and after the bolt is damaged. A data set containing 100 images of key bolt loosening points was collected, and rotation was performed as data augmentation to yield 1800 images. Moreover, a method was designed for automatically annotating the augmented image data set. In this study, 70%, 10%, and 20% of the annotated image data set were used for training, validation, and testing, respectively. Subsequently, a neural network model based on a stacked hourglass network was established to train the annotated image data set. The detection results were evaluated in terms of normalized errors (NEs), percentage of correct key points (PCK), detection speed, and training time. In testing, the proposed method accurately and efficiently identified the bolt loosening angle, with a PCK value as high as 99.3%. The accuracy of the proposed method was also highly robust to different shooting distances, viewing angles, and illumination levels.

1. Introduction

Bolt connections are key in steel structures, and loose bolts compromise the safety of a structure [1, 2]. Therefore, the effective and reliable real-time monitoring of bolted connections can improve the safety of steel structures and prevent structural damage [3, 4].

Currently, manual visual inspection is the most commonly used method for examining bolt connections in steel structures, but it is labor-intensive and prone to error. Graybeal et al. [5] found that inspectors accurately identified loose bolts in structural components at an accuracy rate of less than 50%.

Many studies have developed sensor-based bolt looseness detection methods, such as the acoustic-elastic method [6–8] and electromechanical impedance method [9–12].

The acoustic-elastic method involves measuring the stress acting on a solid by using the ultrasonic wave velocity, which changes with this stress [8]. Piezoelectric ceramic patches are pasted to both ends of a bolt to generate and collect ultrasonic waves, and the bolt looseness is judged by analyzing the changes in the acoustic wave characteristics. Different measurement methods can be selected according to the characteristics of the stress field to be measured. In the electromechanical impedance method, the mechanical impedance of a structure is used. First, a piezoelectric patch is bonded to a bolted joint and excited by a high-frequency voltage. Subsequently, impedance signatures, which reflect the local dynamics of the joint, are obtained and used to assess the degree of bolt looseness. This method is highly sensitive to changes in stress. However, the aforementioned two methods rely on cables and precision instruments and

require manual intervention to extract damage parameters, which are required for damage detection. Under complex environmental conditions (such as drastic changes in temperature and humidity), the performance of a sensor on a bolt might deteriorate, which negatively affects the monitoring results [13]; therefore, the sensor-based bolt looseness detection method has not been widely applied.

Many researchers have combined machine vision technology with deep learning methods to detect bolt loosening, where loose bolts can be identified by classifying them into loose and tight states (screw out the length or whether the drawing line coincides) or the specific angle at which the bolt is loose. Park et al. [14] proposed a pioneering vision-based approach for estimating bolt rotations and signals of looseness in bolted connections. Their method involves capturing digital images of the target connection and then based on the Hough transform [15] and Canny edge detection [16] to calculate the loosening angle.

In using the estimated loose and tight state methods, Cha et al. [17] used the Viola–Jones algorithm to detect bolts in images and estimate their screw out lengths and classified bolt tightness by using a support vector machine. Zhang et al. [18] proposed a bolt loosening evaluation method based on autonomous deep learning. In this method, a faster region-based convolutional neural network (R-CNN) is used to identify and classify the heights of bolts and screws in real time. Yuan et al. [19] proposed an automatic method based on a mask R-CNN for detecting bolt looseness. In this method, bolt looseness is identified through a change in screw height. The trained model can detect bolt looseness from images with different backgrounds. Gong et al. [20] proposed a method based on deep learning and geometric imaging theory for quantitatively calculating the exposed length of bolts. The average measurement error obtained with this method was 0.61 mm. Yang et al. [21] combined the traditional manual torque method commonly used in engineering with the YOLO v3 [22] and YOLO v4 [23] to reduce the cost of manual inspection and the rate of missed inspection. Deng et al. [24] proposed a novel vision-based method to conduct the loosening detection of three threaded joints in a T-junction pipe; a new generative adversarial network-based segmentation module was constructed to accurately segment marked bars, and the average detection accuracy was 94.7%. Deng et al. [25] proposed an automated method for detecting the loosening angle of mark bolted joints by integrating computer vision and geometric imaging theory. In lab-scale and real-scale environments, the average relative detection error was 3.5%. The aforementioned method is suitable for monitoring high bolt loosening damage but not low bolt loosening damage; the traditional manual torque method needs to draw lines in advance, which is not convenient for actual monitoring.

In using calculated loosening angle methods, Park et al. used an R-CNN [26] and a faster R-CNN [27] to identify bolts in images and adopted the Hough transform to determine the rotation angles of bolt edges. However, their method cannot detect end-to-end bolt loosening damage and can only identify loosening angles from 0° to 60° . Zhao et al. [28] used a single-shot multibox network to identify

bolts in images. They located the “bolt” (the real bolt in the picture) and “num” (a number mark on the bolt) in an image by using a regression box and then extracted the central coordinates for bolt angle calculation. Yu et al. [29] used three methods to mark bolts and adopted a single-shot multibox detector to determine the bolt loosening angle, which increased the bolt identification accuracy. However, in the approach of Yu et al., the center position deviation of the identified rectangular frame was high, which resulted in a high error in the identification of the bolt loosening angle. This method is effective only when the bolt rotation angle is large, and pasting marks are unsuitable for use in extensive bolt detection. Therefore, the current method based on the bolt loosening angle has insufficient accuracy. In this method, each bolt must be examined separately to obtain its loosening angle, which is inconvenient when investigating numerous bolts.

Stacked hourglass networks [30] have achieved high performance in applications such as human posture estimation [31], vehicle positioning and recognition [32], construction robots [33], and construction site safety maintenance [34]. These networks can be used to find the contours of a detected object accurately and rapidly. In the present study, a stacked hourglass network was used for automatically and rapidly determining the bolt loosening angle. This method can overcome the limitation of only using rectangular box regression positioning to calculate the bolt loosening angle.

The practicality of the method is one of important issues; in real applications, the structure is painted with the same color. When the contrast between the object of interest and the background is low, it may be difficult to distinguish the key points from the surrounding areas. This can lead to errors in the detection process and reduce the accuracy of the method. However, there are some techniques that can be used to improve the contrast between the object of interest and the background. In order to better simulate the actual construction site environment, Nguyen et al. [35] used the bolted connection model coated with gray anti-corrosion paint to simulate a real steel bridge. Pham et al. [36] used solidworks to create a bolt model sprayed with gray paint. Another approach to addressing the issue of low contrast is to use a different type of key point detector that is less reliant on the contrast between the object of interest and the background. For example, some key point detectors rely on the shape of the object rather than the contrast and may be more suitable for applications where the contrast is low.

The proposed method has been specifically designed and optimized for hexagon bolts. In general, the feasibility of the proposed method for other types of bolts would depend on the geometry and surface features of the bolt head, which determine the quality and robustness of the image processing and analysis algorithms. For instance, if the bolt head has similar geometry and surface features to a hexagon bolt, our method could potentially be adapted with minor modifications. On the other hand, if the bolt head has a significantly different geometry or surface features, it may only require a new geometric analysis, but the overall

process remains the same. For example, for round bolts, we can paste symbols on the bolt heads as key points.

The rest of this paper is organized as follows. Section 2 introduces the proposed method for determining the bolt loosening angle. Section 3 describes the adopted experimental setup, data set, and data acquisition system. Section 4 presents the experimental results of this study. Finally, Section 5 details the conclusions and limitations of this study and provides recommendations for future research.

2. Methodology

The method proposed in this paper for automatically determining the bolt loosening angle involves three steps: (a) data preparation, (b) model training, and (c) model performance evaluation. Data preparation involves image collection, the definition of key points, manual annotation, data augmentation, and automatic annotation. Model training mainly involves using a deep learning module and an optimization function to calculate the bolt loosening angle. Finally, model performance evaluation involves using the normalized error (NE), the percentage of correct key points (PCK), and other evaluation indicators to evaluate the model performance to obtain reference standards for model use. The overall workflow of the proposed method is displayed in Figure 1. The first red box introduces the general steps of data preparation, key point extraction, and loosening angle calculation. The second red box is the specific structure of the hourglass network; “res” is the abbreviation of residual module. The third red box is the calculation method of bolt looseness and model performance evaluation; θ is the bolt loosening angle. This method is described in detail in the following text.

2.1. Data Preparation. Captured using the camera of an iPhone12 (12-MP vision sensor, $f/2.4$ aperture, and focal length of 26 mm), 100 images of a single bolt with a resolution of 4000×3000 pixels were collected in a laboratory environment under different shooting distances, viewing angles, and illumination levels. To augment the data set, 25 of these images were rotated every 5° by using the image rotate function [37] of the Python Imaging Library to obtain 1800 augmented images. The augmented images had the same shape as the original images but different rotation angles. Moreover, the rotation angles were known, and the problem of inaccurate rotation angle measurement was thus avoided. A total of 70%, 10%, and 20% of the augmented data set were used for training, verification, and testing, respectively. To balance both computational efficiency and accuracy, images from all the data sets were cut into a size of 512×512 pixels. The object in the images (i.e., bolts) was annotated using Labelme3.16.7 [38]. Each bolt had the mark of a letter in the English alphabet; thus, these marks could be used as reference points, and each corner was numbered to make it regular. The corner to the left of the letter on the bolt was annotated as P_1 , and the other corner points were sequentially annotated as P_2, P_3, P_4, P_5 , and P_6 in the clockwise direction. JavaScript Object Notation (JSON) [39]

(a lightweight data exchange format) files were generated for model training.

For the automatic annotating augmented images, the corner coordinates can be obtained by the rotating angle according to the coordinate points marked in the original picture, taking the center of the image as the center of the circle and the distance between the coordinate point and the center of the circle as the radius. The coordinates in the JSON file of an original image were transformed into the coordinates of the rotated images by using the process described in the following text.

As displayed in Figure 2, (x_0, y_0) are the center coordinates of each image, (x_i, y_i) are the corner coordinates of each bolt, and the distance l_i between each coordinate point and the center point of the original image is calculated. Subsequently, the angle q_i between each coordinate point and the center point is determined by comparing with the x -axis. The q_i value obtained using equation (2) only ranges from 0° to 90° ; therefore, the real angle between the aforementioned points (q'_i) should be determined by identifying the quadrant in which q_i lies.

$$l_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}, \quad (1)$$

$$q_i = \arccos \frac{|x_i - x_0|}{l_i}. \quad (2)$$

In image processing applications in Python, the positive direction of the y -axis points downward; correspondingly, the following inequalities are obtained:

$$\begin{aligned} x_i - x_0 \geq 0 \cup y_i - y_0 < 0; q'_i &= q_i, \\ x_i - x_0 < 0 \cup y_i - y_0 < 0; q'_i &= 180 - q_i, \\ x_i - x_0 < 0 \cup y_i - y_0 \geq 0; q'_i &= 180 + q_i, \\ x_i - x_0 < 0 \cup y_i - y_0 \geq 0; q'_i &= 360 - q_i. \end{aligned} \quad (3)$$

Finally, rotation is performed as follows to obtain the rotated horizontal and vertical coordinates:

$$\begin{aligned} x'_i &= l_i \times \cos(q'_i + i) + x_0, \\ y'_i &= y_0 - l_i \times \sin(q'_i + i). \end{aligned} \quad (4)$$

2.2. Hourglass Network. An hourglass network has a symmetric structure and contains a residual module. The residual module can extract high-level features through the convolution operation, and it can retain the original information by using a skip route function. The aforementioned module changes the depth of data without changing their size. Therefore, this module can be considered an advanced convolutional layer. The structure of the residual module is displayed in the dotted box at the bottom of Figure 3.

The structure of a fourth-order hourglass module is displayed in the dotted box at the top of Figure 3. This module performs four downsampling and four upsampling processes, and it allows the network to extract feature information at different scales. The right and left blocks

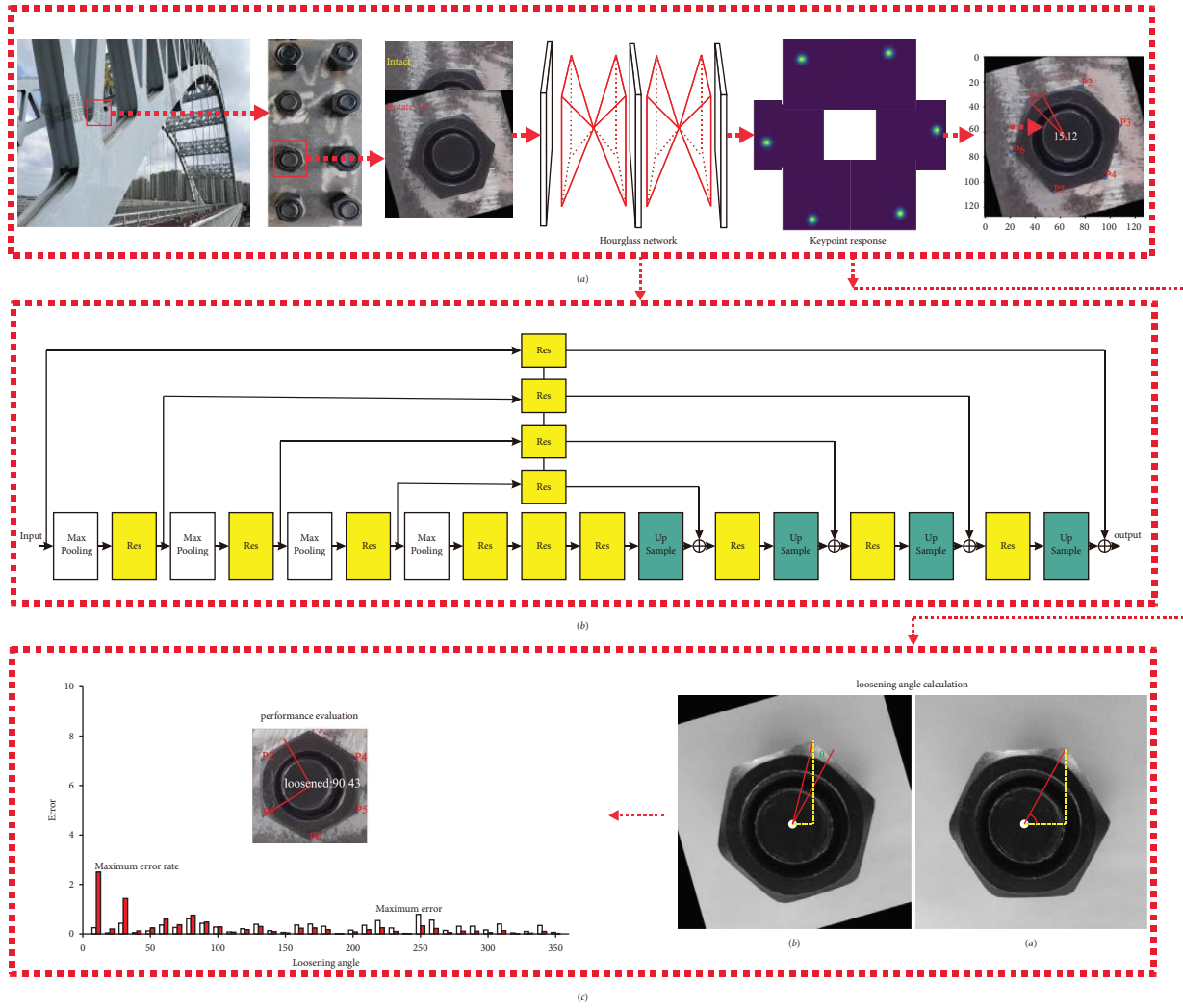


FIGURE 1: Overall workflow of the proposed method for automatically determining bolt loosening angle. (a) Data preparation. (b) Model training. (c) Model performance evaluation.

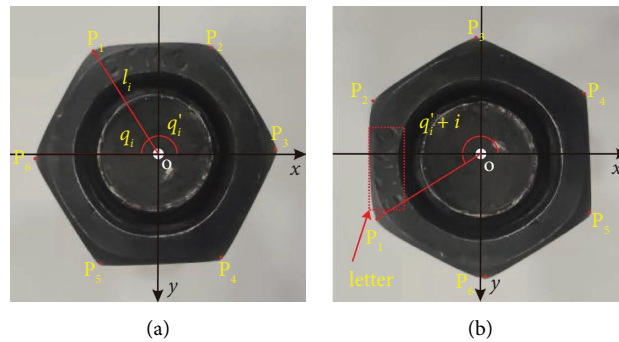


FIGURE 2: Automatic annotation process. (a) Calculate the angle of each coordinate point. (b) Rotate by set angle.

(C1–C4 and C4b–C1b, respectively) of the hourglass module are mirror images of each other. A similar set of blocks (C4a–C1a) also exists at the top of this module. Each block is combined with the block to its right through the plus sign. Because the network is symmetrical, its appearance is similar to that of an hourglass. After the feature layers are

superimposed, the output feature map C1b not only retains the information of all layers but also has the same size as the input map. Heatmaps representing the probabilities of key points are generated through 1×1 convolution. The output results of the first hourglass module are then summed and input to the second hourglass module. Finally, the

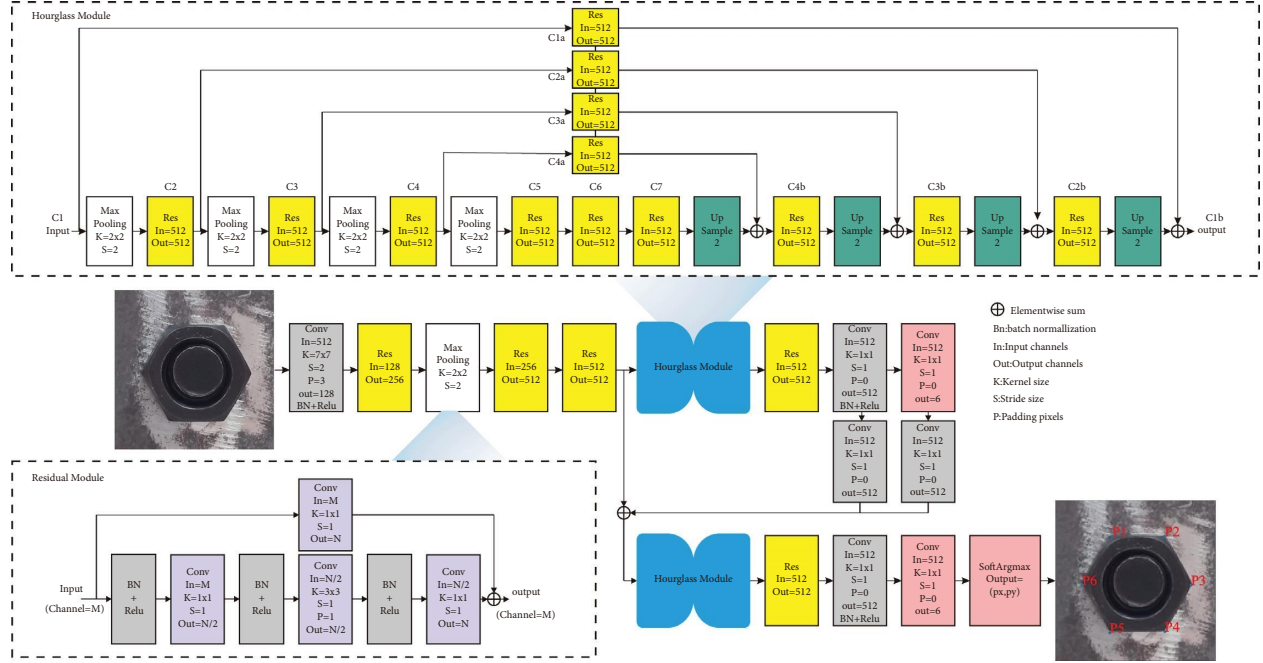


FIGURE 3: Key point detection for bolts by using a stacked hourglass network.

coordinates with the highest final prediction probability for each key point are obtained using the softargmax function.

To detect key points for bolts rapidly and accurately, a detector based on stacked hourglass networks was established in this study. In a deep learning network, the extracted features are gradually abstracted from the deep layers. For example, in a deep learning network designed for detecting bolts, the first layer extracts low-level features, such as contours, the second layer extracts some higher-level semantic features on the basis of the first layer, and the third layer extracts the most relevant features for the detection task, such as the complete bolt contour. In a convolutional neural network, a layer only retains the feature map of the layer immediately preceding it, which results in the loss of some information. In corner estimation, the highest recognition accuracy might be achieved for different corners in different feature maps. For example, P_1 points might be easily identified in the feature map of the second layer, whereas P_4 points might be easier to identify than P_1 points in the feature map of the fourth layer. Therefore, an hourglass network can perform improved key point detection if it contains multiple feature graphs.

2.3. Objective Function Optimization. In network training, an hourglass network usually adopts a loss function based on the l_2 norm that is commonly used in regression problems (i.e., the mean square error loss [40]). Such a loss function is expressed as follows:

$$L_{L2} = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^p)^2, \quad (5)$$

where y_i represents the target true value, y_i^p represents the corresponding value predicted by the network, and n is the number of target key points.

The squaring operation of the mean square error loss results in the model producing high losses at noise points and causes noise points to have high weights. When the model is optimized to reduce the losses at noise points, the overall model performance deteriorates. Therefore, in this study, the smooth l_1 loss function [40] was used to optimize the original loss function, reduce the weight assigned by the network to the noise point, and enhance the network's generalization ability. The smooth l_1 loss function is expressed as follows:

$$L_{\text{smooth}_l1} = \frac{1}{n} \sum_{i=1}^n d_i, \quad (6)$$

$$d_i = \begin{cases} \lambda(y_i - y_i^p)^2 & |y_i - y_i^p| < \delta, \\ \delta|y_i - y_i^p| - (1 - \lambda)\delta^2 & \text{others,} \end{cases}$$

where y_i represents the target true value, y_i^p represents the corresponding value predicted by the network, n is the number of target key points, λ is the loss adjustment coefficient (which is usually 0.5), and δ is the critical point between the square loss and the absolute loss.

2.4. Calculation of the Bolt Loosening Angle. The process used in this study for calculating the bolt loosening angle is illustrated in Figure 4. The six corners of a bolt are detected using an hourglass network, and the center coordinates of the bolt are then obtained from the coordinates of the six corners. Because the hourglass network can use the relationship between key points for positioning, the letter on the bolt surface can be used as a mark to locate the bolt. The difference in these angles is the bolt rotation angle.

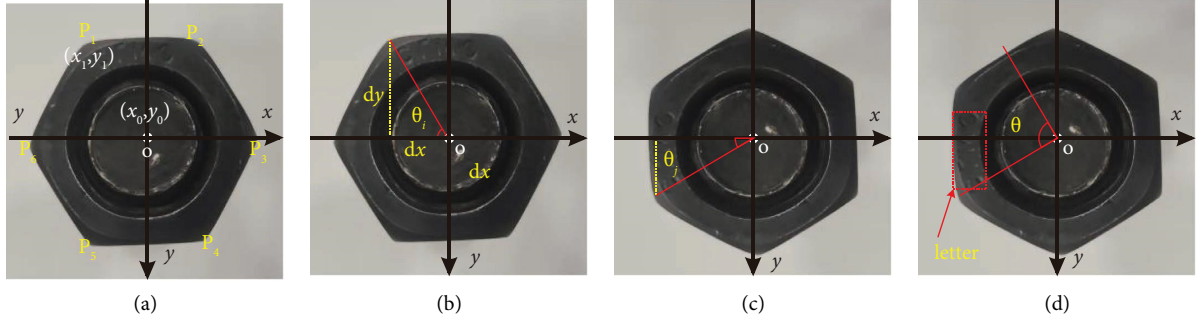


FIGURE 4: Process for calculating bolt loosening angle. (a) Determine the coordinates of each corner point and center point. (b) Calculate the angle between each corner point and the center point. (c) Calculate the angle between each corner point and the center point after loosening. (d) Calculate loose angle.

The corner coordinates of each bolt are (x_i, y_i) . On the basis of these coordinates, the center coordinates (x_0, y_0) of each bolt can be calculated as follows:

$$\begin{aligned} x_0 &= \frac{\sum_{i=1}^6 x_i}{6}, \\ y_0 &= \frac{\sum_{i=1}^6 y_i}{6}. \end{aligned} \quad (7)$$

The distance between each corner and the x -axis is calculated as follows:

$$\begin{aligned} dx_i &= x_i - x_0, \\ dy_i &= y_i - y_0. \end{aligned} \quad (8)$$

The angle formed by an unrotated corner, the x -axis, and the center point is calculated using the following equation:

$$\theta_i = \arctan\left(\frac{dy_i}{dx_i}\right). \quad (9)$$

Similarly, the angle θ_j formed by a rotated corner, the x -axis, and the center point is obtained by using the relevant rotated coordinates in equation (9).

The angle between the line and x -axis is calculated in Python, and the obtained angle is $[-180^\circ, 180^\circ]$, which is positive in quadrants 1 and 2 and negative in quadrants 3 and 4. The default bolt rotation is counterclockwise; thus, when θ_i and θ_j are combined, the bolt loosening angle θ can be determined as follows:

$$\begin{aligned} \theta_i \geq 0; \theta &= \begin{cases} |\theta_j - \theta_i| & \theta_j \geq \theta_i, \\ 360 - |\theta_j - \theta_i| & \theta_j < \theta_i, \end{cases} \\ \theta_i < 0; \theta &= \begin{cases} |\theta_j - \theta_i| & \theta_j \geq \theta_i \text{ and } \theta_j < 0, \\ |\theta_j| + |\theta_i| & \theta_j \geq \theta_i \text{ and } \theta_j > 0, \\ 360 - |\theta_j - \theta_i| & \theta_j < \theta_i. \end{cases} \end{aligned} \quad (10)$$

To minimize the error in the rotation angle of a corner, the rotation angles of the six corners of a bolt are averaged to obtain final loosening angle.

2.5. Model Evaluation. Four indices, including the NE and PCK, were used to evaluate the performance of the developed hourglass model. The NE and PCK are typical evaluation indicators for the solutions of most key point problems, such as facial key point estimation [41] and human posture estimation based on key points [31]. When the NE and PCK were calculated, a distance normalization parameter had to be used because the area occupied by a bolt differed between images. In this study, the normalization parameter was set as the average distance between a bolt corner and the center point.

2.5.1. Normalized Error. The NE is the average normalized distance between the predicted key point and the ground truth [42]. The normalized distance is obtained by dividing the predicted Euclidean distance and ground truth value by a distance normalization parameter. A smaller NE indicates better model performance. The NE is calculated using the following equation:

$$NE = \frac{1}{M} \sum_{m=1}^M \frac{\sum_{k=1}^K ((d_{m,k}/S_m) \cdot \delta)}{\sum_{k=1}^K \delta}, \quad (11)$$

where M is the total number of test images, K is the total number of key points, $d_{m,k}$ is the Euclidean distance between the predicted position and the ground truth of the k_{th} key point in the m_{th} image, and s_m is the normalization parameter in the m_{th} image.

2.5.2. Percentage of Correct Key Points. The PCK refers to the percentage of key points that are correctly predicted in the entire data set and is an indicator of the accuracy of a pose estimation model [43]. According to the definition of the human pose estimation problem, if the normalized distance between a predicted position and the ground truth position is within a given threshold α , where $\alpha \in [0, 1]$, the candidate key point is assumed to be correctly located. For human posture estimation, 0.05, 1, and 0.2 are usually used as the upper limits of α . The key points of bolts are easier to locate than are those of human postures; therefore, the upper limit used to calculate the PCK curve of each model was set as $\alpha = 0.025$ in the present study for model performance to be

accurately evaluated. The PCK is calculated using the following equation:

$$\text{PCK}(\alpha) = \frac{1}{Mk} \sum_{m=1}^M \frac{\sum_{k=1}^K (\{(d_{m,k}/S_m) \cdot \delta\} < \alpha)}{\sum_{k=1}^K \delta}, \alpha \in [0, 0.025], \quad (12)$$

where α is the threshold for calculating and drawing the PCK curve and the definitions of the other parameters are the same as those in equation (11).

3. Training and Key Point Assessment

3.1. Experimental Hardware and Software. All experiments in this study were performed using the open-source stacked hourglass networks [13] Win 10, CUDA, and PyTorch 1.7 on a desktop computer equipped with 12 CPUs (Intel Core i7-9800X CPU @3.80 GHz) with 32 GB of DDR4 RAM and a 10 GB GeForce RTX 3080 graphics card.

3.2. Training. Training was conducted using the PyTorch framework. For training, the network was fed 512×512 -pixel, three-channel color images, and the batch size was 144 and initial learning rate was 1×10^{-4} . The learning rate decreased by 10% for every 2000 steps in the training process. A total of 70%, 10%, and 20% of the data set (1313, 187, and 375 images, respectively) were used for training, validation, and testing, respectively. Root Mean Square Propagation (RMSprop) [44] was selected as the network optimizer for the first 1000 epochs, after which Adam [45] was used as the network optimizer. This approach was adopted because RMSprop has high convergence speed but low convergence accuracy, whereas Adam has low convergence speed but high convergence accuracy; thus, the aforementioned method increased the training speed and reduced the training time. Early-stopping technology [46] was used to monitor the loss on the validation set, so as to stop training when the model starts to overfit and improve the generalization ability of the model. The criteria for Early-stopping technique can be as follows: the loss on the train set does not decrease after multiple iterations of the model. 2. The loss on the validation set starts to rise. As shown in Figure 5, the final number of training iterations was 9,800, and the model loss decreased with an increase in the number of iterations until the model loss became stable.

Table 1 presents all the training parameters. At this time, the model was stable, and its recognition accuracy was high. Finally, the trained model was applied to the test data set to evaluate its performance for new images.

3.3. Model Evaluation. The performance of the hourglass network in identifying the key points of bolts under different annotation methods was examined through experiments. The detection time and training time were the same in the experiments. The detection time for each graph was 71.55 ms, and the training time was 5.32 h.

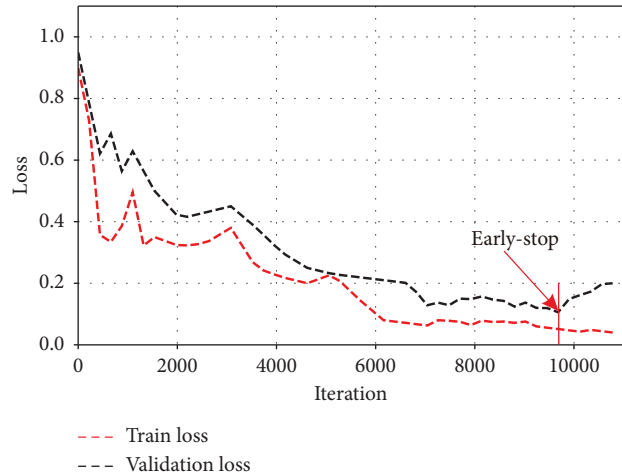


FIGURE 5: Train and validation loss.

TABLE 1: Training parameters.

Parameters	Value
Image dimension	512×512
Learning rate	0.0001
Weight decay	0.0001
Momentum	0.9
Total epoch	784
Batch_size	144
Sigma	0.5

The NE value was used to evaluate the model accuracy (Table 2). As presented in Table 2, automatic labeling considerably improved the accuracy of the hourglass model, with the NE values of all key points being reduced by 9.28% on average. The NE values of the model trained through manual and automatic annotation differed between key points. For both annotation methods, the lowest NE was achieved for P_1 possibly because this point was closest to the letter mark on the bolt. The NE values for P_1 under manual and automatic annotation were 7.98×10^{-3} and 7.70×10^{-3} , respectively. The highest NE was achieved for P_3 under both annotation methods. The NE values for this point under manual and automatic annotation were 10.94×10^{-3} and 10.22×10^{-3} , respectively. The performance improvement (i.e., percentage reduction in NE) resulting from automatic annotation compared with manual annotation was different for each key point. Specifically, the largest NE reductions when automatic annotation was used were observed for P_6 (18.1%) and P_4 (13.04%), whereas the smallest NE reduction was observed for P_5 (3.03%).

The PCK curves of the hourglass model were calculated using different threshold values ($\alpha \in [0, 0.025]$) under manual annotation and automatic annotation (Figure 6). In general, the PCK increased with the threshold. For each key point, the PCK increased considerably with the threshold value when $\alpha \in [0, 0.015]$, and the magnitude of increase decreased when α exceeded 0.015. The average PCK value of different key points plateaued at 98.1% and 99.3% under manual annotation and automatic annotation, respectively. In general, under the two annotation methods, the PCK

TABLE 2: NEs obtained under manual and automatic labeling.

Keypoint	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	Average
Manual label (10 ⁻³)	7.98	9.69	10.94	8.91	9.69	10.46	9.61
Auto label (10 ⁻³)	7.70	8.71	10.22	7.88	9.41	8.85	8.79
NEreduction (%)	3.64	11.27	7.06	13.04	3.03	18.10	9.28

values for P₁ and P₄ were higher than the average PCK value, whereas the PCK values for P₃ and P₆ were lower than the average PCK value; thus, the network recognized the key points located closer to the letters on the bolt more accurately. The PCK value of the adopted hourglass model was higher under automatic annotation than under manual annotation for the different key points. Thus, the automatic annotation method proposed in this paper can improve the identification accuracy of key points, which can enhance the detection accuracy of the bolt loosening angle.

4. Calculation of Bolt Loosening Angle

After undergoing training and testing, the developed stack hourglass network determined the corner coordinates of bolts in the data set images. These coordinates were then used to obtain the rotation angle, and the degree of bolt loosening and the bolt loosening damage was determined by detecting the change in the rotation angle.

4.1. Recognition Accuracy for Different Rotation Angles. Table 3 presents the absolute values and errors of the bolt loosening angles obtained using the proposed method and the method of Zhao [10] under the same rotation angle. Zhao used a Single-Shot MultiBox Detector network to locate rectangular frames for a bolt and nut; however, the center position deviation of the recognized rectangular frames was large, and only these two positioning points existed, which resulted in a large error for the bolt loosening angle. The average error of the bolt loosening angle obtained using Zhao's method is 5% for angles of up to 360°. By contrast, other bolt looseness detection methods can only recognize bolt loosening angles of up to 60°. Table 3 indicates that under the same rotation angle, the proposed method provides a more accurate bolt loosening angle than does the method of Zhao. The result was primarily because the hourglass network used in the proposed method achieves high-accuracy bolt looseness detection by overlaying feature layers and incorporating six key points in the last feature map.

4.1.1. Recognition Accuracy for Small Rotation Angles. To determine the minimum rotation angle that is recognizable with the proposed method, an original image captured at a shooting distance of 30 cm under mild light conditions was rotated every 0.5° between 0° and 30° by using Python to obtain 60 rotated images.

Figure 7 depicts the identification of bolt images with rotation angles of 0.5°, 1°, 2°, and 4°. The results indicated that the average angular difference was approximately 0.4°. For a rotation angle of 0.5° and 1°, the error rate reached 36% and 39%, the error rate was too large to use as minimum rotation angle. For a rotation angle of 2°, the error rate reached 19%. Although this error rate was still large, the absolute angular difference was only approximately 0.38°, and the average error rate was approximately 3%. The absolute and percentage errors in the bolt loosening angle are presented in Table 4. The error rate decreased with an increase in the rotation angle. On the basis of the obtained results, the minimum bolt loosening angle that can be identified with the proposed method was determined to be 2°.

4.1.2. Recognition Accuracy for Large Rotation Angles. To determine the maximum rotation angle that is recognizable with the proposed method, an original image captured at a shooting distance of 30 cm under mild light conditions was rotated every 5° between 0° and 360° by using Python to obtain 72 rotated images. The developed model was then tested on these images. The rotation angles of the bolt were calculated using the proposed method and compared with the actual rotation angles. Figure 8 displays the recognition results for rotation angles of 90°, 180°, 270°, and 350°. The obtained results indicate that the proposed method can achieve 360° detection of the bolt loosening angle.

4.2. Recognition Accuracy under Different Conditions

4.2.1. Recognition Accuracy under Different Shooting Distances. The variation in the recognition accuracy of the proposed method with shooting distance was examined. Each image in the data set was rotated every 10° from 0° to 360° in Python, yielding 36 images for every original image. The recognition results obtained with the proposed method under different shooting distances are displayed in Figure 8. The absolute and percentage errors in the bolt loosening angles obtained with the proposed method under different shooting distances are presented in Table 5, in which the numbers in parentheses indicate the rotation angle. The results indicated that the detection accuracy of the proposed method was affected by the shooting distance. When the shooting distance was 20 cm, the average error in the bolt loosening angle was 0.27° (average error rate of 0.29%); however, when the shooting distance was 50 cm, the average error in this angle increased to 1.07° (average error rate of 1.10%). Figure 9 indicates that as the shooting distance increased, the absolute and percentage errors increased considerably; this was because the number of pixels occupied by the bolt and the image definition was lower at higher shooting distances. Moreover, the letters on the bolt became blurred, which was inconducive for positioning and recognition by the hourglass network. Therefore, as the shooting distance increased, the detection accuracy for the bolt loosening angle decreased. The recommended shooting distance for the proposed method is approximately 30 cm.

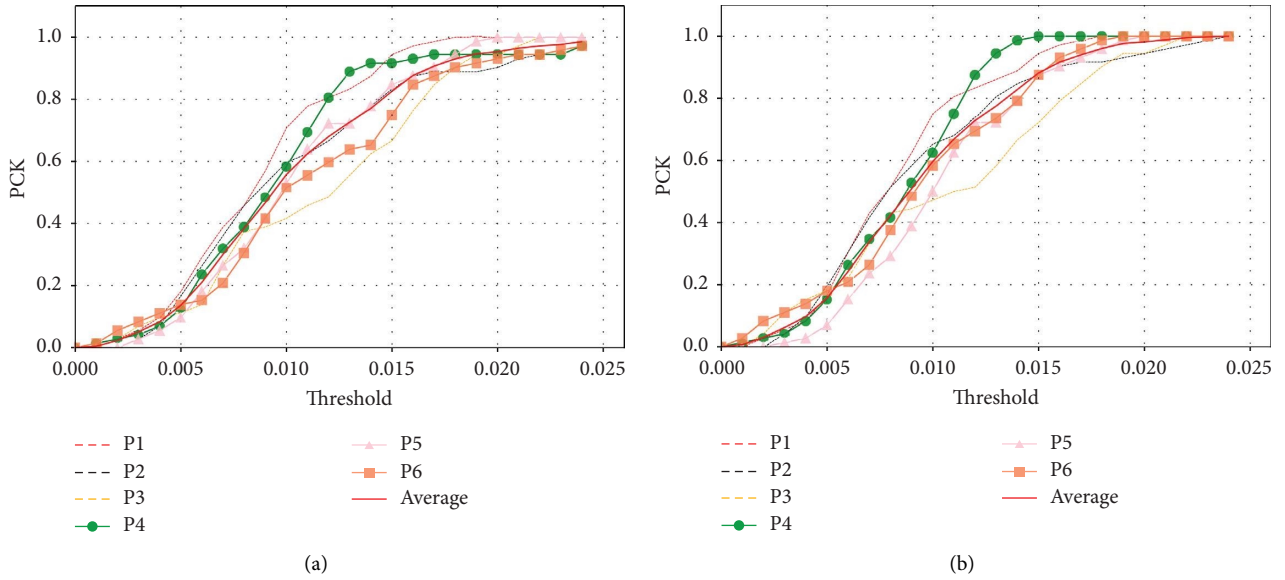


FIGURE 6: PCK curves of the stacked hourglass network for each key point of a bolt. (a) Manual label. (b) Auto label.

TABLE 3: Absolute values and errors of the bolt loosening angles obtained using the proposed method and the method of Zhao [10] under the same rotation angle.

Test sample	Calculated value (°)	Actual value (°)	Difference value (°)	Error (%)
Zhao [10] a	6.33	10	3.67	36.7
Zhao [10] b	16.2	20	3.8	19
Zhao [10] c	23.05	30	6.95	23.2
a	9.54	10	0.46	4.6
b	19.72	20	0.28	1.4
c	29.9	30	0.1	0.33

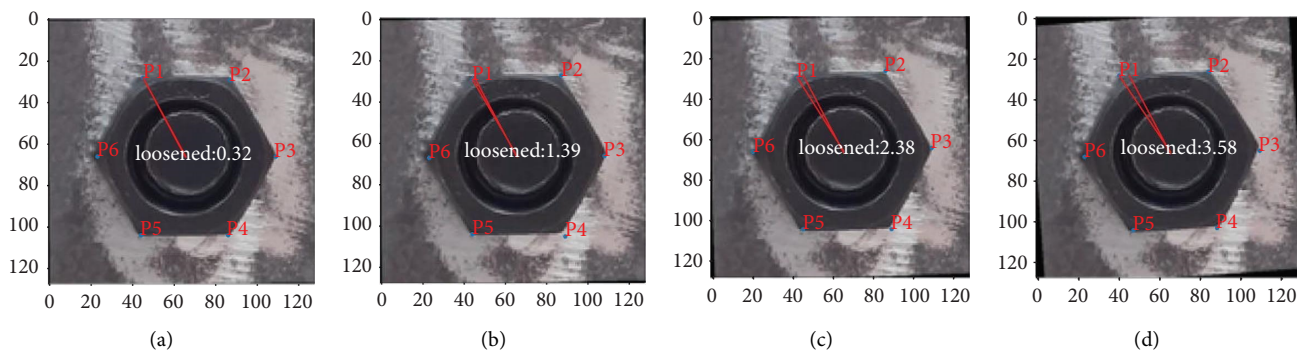


FIGURE 7: Recognition results for small rotation angles. (a) Rotation angle of 0.5°. (b) Rotation angle of 1°. (c) Rotation angle of 2°. (d) Rotation angle of 4°.

TABLE 4: Absolute and percentage errors in the bolt loosening angle when using the proposed method.

Calculated value (°)	Actual value (°)	Difference value (°)	Error rate (%)
0.32	0.5	0.18	36
1.39	1	0.39	39
2.38	2	0.38	19
3.58	4	0.42	10.5

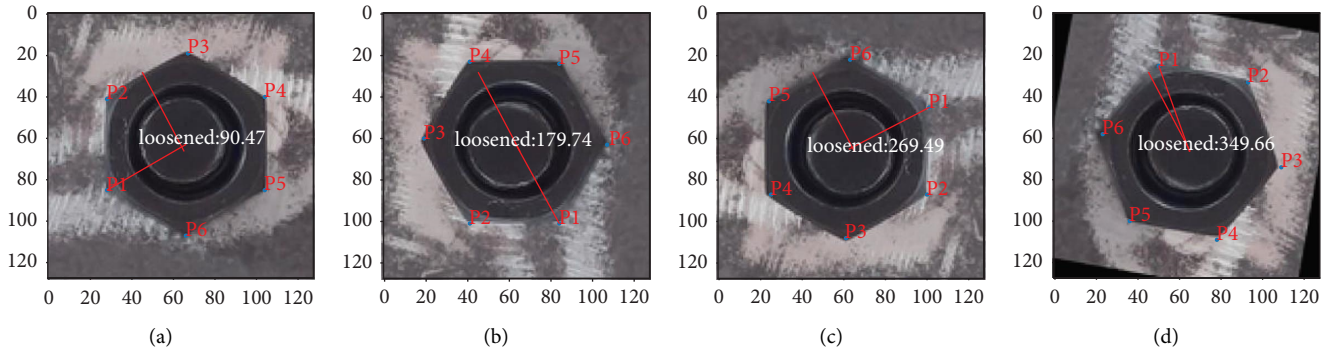


FIGURE 8: Recognition results for large rotation angles. (a) Rotation angle of 90°. (b) Rotation angle of 180°. (c) Rotation angle of 270°. (d) Rotation angle of 350°.

TABLE 5: Test results obtained under different shooting distances.

Test sample (cm)	Maximum error (°)	Maximum error rate (%)	Average error (°)	Average error rate (%)
20	0.79 (250)	2.5 (10)	0.27	0.29
30	0.97 (230)	3.7 (10)	0.42	0.44
40	1.63 (150)	6.6 (10)	0.79	0.82
50	2.24 (260)	8.5 (10)	1.07	1.10

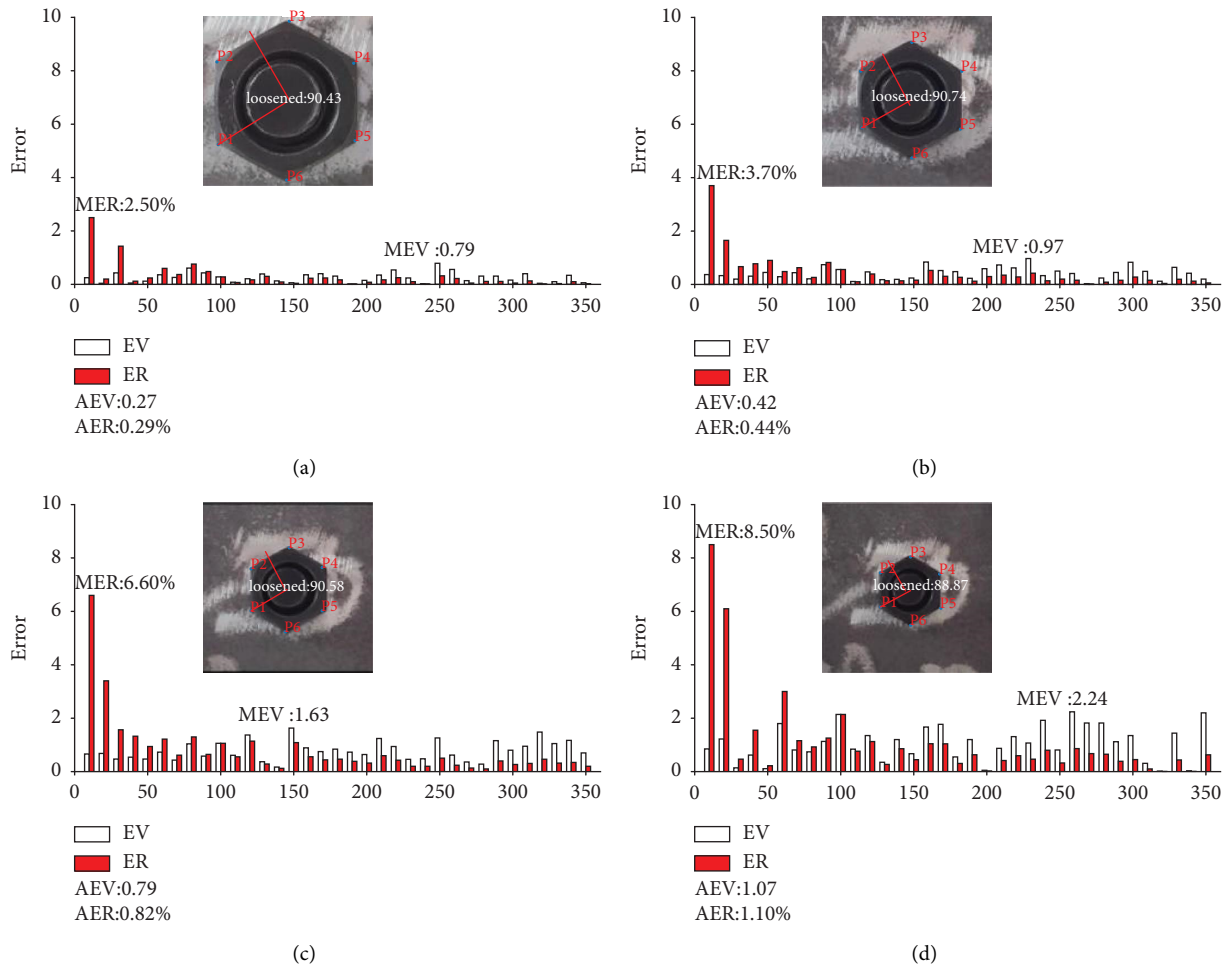


FIGURE 9: Recognition accuracy and error observed under different shooting distances. (a) Loosening angle with a distance of 20 cm. (b) Loosening angle with a distance of 30 cm. (c) Loosening angle with a distance of 40 cm. (d) Loosening angle with a distance of 50 cm.

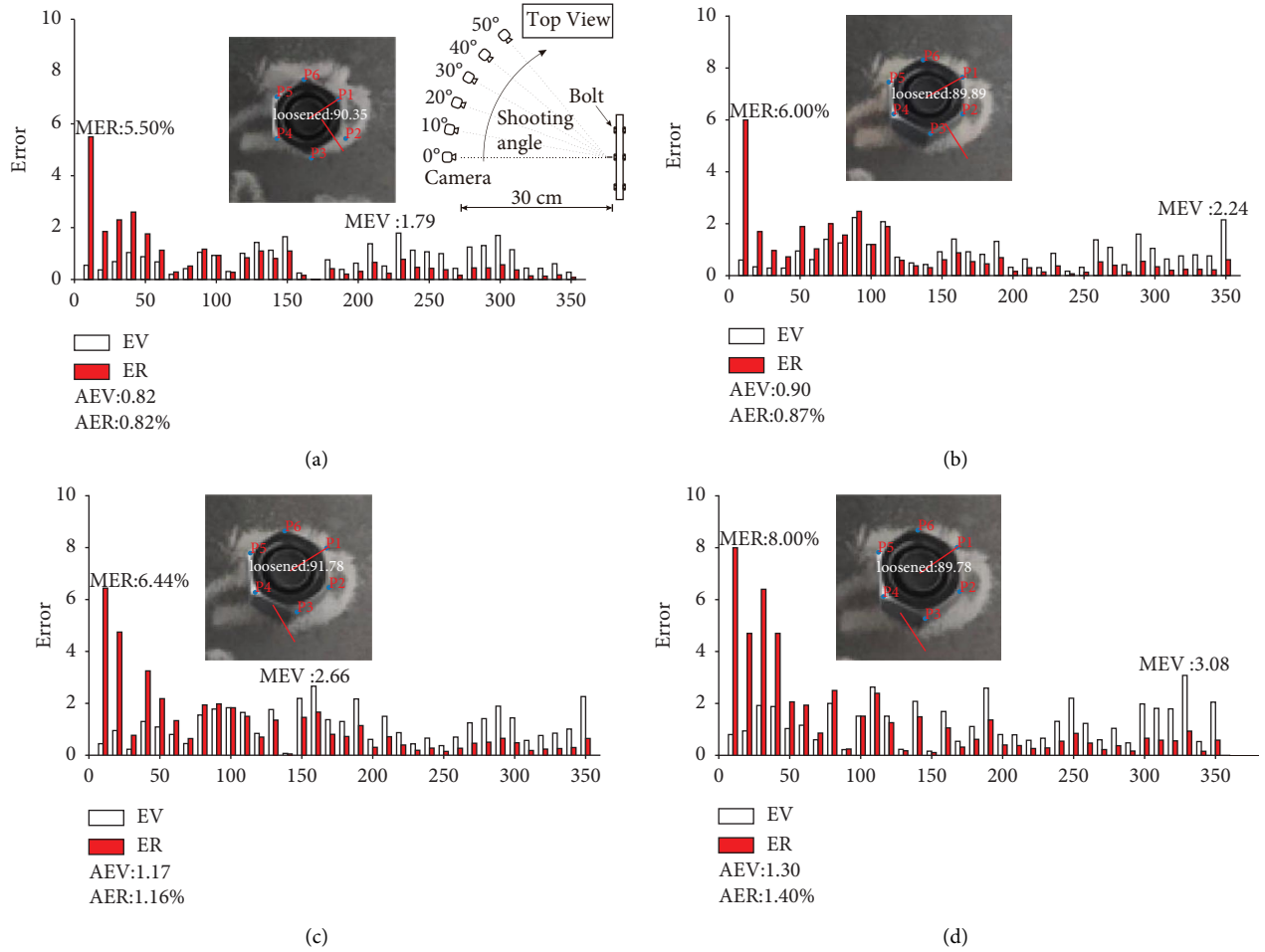


FIGURE 10: Recognition accuracy and error observed under different shooting angles. (a–d) Loosening angle.

4.2.2. Recognition Accuracy under Different Shooting Angles.

The effect of the shooting angle on the recognition accuracy of the proposed method was examined. An original image was rotated every 10° between 0° and 360° by using Python to obtain 36 rotated images. The recognition results obtained with the proposed method under different shooting angles are displayed in Figure 10, which depicts schematics of the shooting angles. Table 6 presents the absolute and percentage errors in the bolt loosening angles obtained with the proposed method under different shooting angles. The results indicated that the detection accuracy of the proposed method was affected by the shooting angle. When the shooting angle was 10° , the average error in the loosening angle was 0.82° (average error rate of 0.82%); however, when the shooting angle was 30° , the average error in the loosening angle was 1.30° (average error rate of 1.40%). The aforementioned results were obtained because when the shooting angle increased, the bolt was no longer a complete regular hexagon and the distortion increased. Although the hourglass network could still accurately locate and identify the bolt, the irregularity of the bolt caused errors in the calculated bolt loosening angle. Therefore, as the shooting angle increased, the accuracy of bolt loosening detection

decreased. Vertical shooting should be ensured as far as possible when using the proposed method.

4.2.3. Recognition Accuracy under Different Luminance Levels.

To study the influence of the luminance level on the recognition accuracy of the proposed method, images with different luminance values (70, 50, 30, and 10) were produced from an original image with a luminance value of 80 that was captured vertically under moderate light at a shooting distance of 30 cm. Python was used to rotate the original image every 10° between 0° and 360° to obtain 36 rotated images. The recognition results obtained with the proposed method under different luminance levels are displayed in Figure 11. The absolute and percentage errors in the bolt loosening angles obtained with the proposed method under different luminance levels are presented in Table 7. The results indicated that the detection accuracy of the proposed method was affected by the image luminance. When the luminance was 70, the average error in the bolt loosening angle was 0.41° (average error rate of 0.49%); however, when the luminance was 10, the average error in the loosening angle was 0.85° (average error rate of 0.84%). As the luminance level decreased, the bolt became increasingly similar to the

TABLE 6: Test results obtained under different shooting angles.

Test sample	Maximum error (°)	Maximum error rate (%)	Average error (°)	Average error rate (%)
0° of tilt	1.79 (230)	5.50 (10)	0.82	0.82
10° of tilt	2.34 (350)	6.00 (10)	0.90	0.87
20° of tilt	2.66 (160)	6.44 (10)	1.17	1.16
30° of tilt	3.08 (340)	8.00 (10)	1.30	1.40

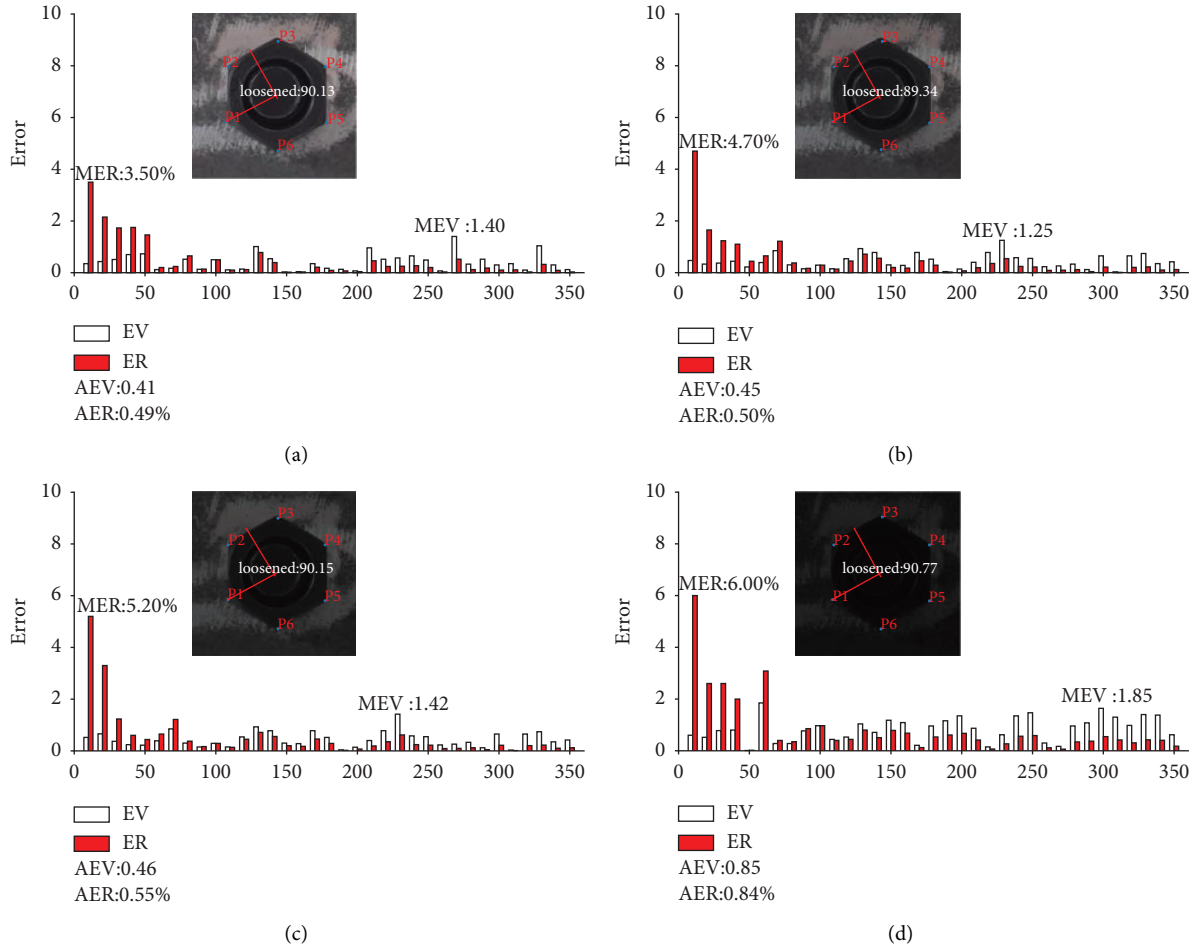


FIGURE 11: Recognition accuracy and error under different luminance levels. (a) Loosening angle with a luminance of 70. (b) Loosening angle with a luminance of 50. (c) Loosening angle with a luminance of 30. (d) Loosening angle with a luminance of 10.

TABLE 7: Test results obtained under different luminance levels.

Test sample	Maximum error (°)	Maximum error rate (%)	Average error (°)	Average error rate (%)
Luminance70	1.40 (270)	3.50 (10)	0.41	0.49
Luminance50	1.25 (230)	4.70 (10)	0.45	0.50
Luminance30	1.42 (230)	5.20 (10)	0.46	0.55
Luminance10	1.85 (300)	6.00 (10)	0.85	0.84

background, which was inconducive for location recognition by the hourglass network. Therefore, a decrease in the luminance level caused a reduction in the accuracy of

bolt loosening detection. A moderate light luminance value of 80 should be selected when capturing images for use in the proposed method.

5. Conclusion

In this study, a method based on computer vision and deep learning technology was designed for automatically estimating the bolt loosening angle by identifying the key points of a bolt in an image of the bolt. In contrast to bolt loosening detection methods based on manual visual inspection or sensors and traditional computer vision technology, the proposed method does not require the installation of sensing equipment on steel frame bridges or the manually design of feature extraction programs. Compared with the aforementioned manual or sensor-based methods, the proposed method based on an hourglass network is faster, less labor-intensive, more accurate, and more efficient. The conclusions of this study are as follows:

- (1) The proposed method exhibited an average NE of 8.79×10^{-3} . Moreover, when the threshold α was set as 0.025, the PCK value of the proposed method was 99.3%, and its detection speed was 71.55 ms per image.
- (2) The average error rate of the bolt loosening angle obtained with the proposed method was less than 1.5% under different shooting distances, shooting angles, and luminance levels.

The improvement of the present study in the future is discussed as follows: first, although augmented images are automatically labeled in the proposed method, considerable manual annotation is performed for labeling the images. Second, for images that have a large deviation angle or are captured under a long shooting distance, the proposed method might perform poorly. Third, the proposed method is designed for hexagon bolt, and it also has potential use for other bolt types. For round bolt type, the proposed method requires adding symbols to the bolt head to detect the looseness angle. To ensure the stability of the developed deep learning model, additional images with different observation viewpoints and shooting distances will be collected to expand the size of the bolt angle data set.

Data Availability

All data, codes, tables or figures during the study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Ying Yu conceptualized the study, proposed the methodology, performed formal analysis, investigated the study, reviewed and edited the manuscript, and supervised the study. Shichuan Wei performed software validation and data curation and wrote the original draft. Wei Zhao conceptualized the study, was responsible for resources, reviewed and edited the manuscript, and supervised the study.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the Natural Science Foundation of China (grant no. 52238001) and Guangdong Provincial Science and Technology Project of China (grant no. STKJ2021190).

References

- [1] J. Fisher and S. Roy, "Fatigue of steel bridge infrastructure," *Structure and Infrastructure Engineering*, vol. 7, no. 7-8, pp. 457-475, 2011.
- [2] H. Gong, J. Liu, and X. Ding, "Study on local slippage accumulation between thread contact surfaces and novel anti-loosening thread designs under transversal vibration," *Tribology International*, vol. 153, Article ID 106558, 2021.
- [3] Y. Jiang, M. Zhang, and C.-H. Lee, "A study of early stage self-loosening of bolted joints," *Journal of Mechanical Design*, vol. 125, no. 3, pp. 518-526, 2003.
- [4] R. Miao, R. Shen, S. Zhang, and S. Xue, "A review of bolt tightening force measurement and loosening detection," *Sensors*, vol. 20, no. 11, p. 3165, 2020.
- [5] B. A. Graybeal, B. M. Phares, D. D. Rolander, M. Moore, and G. Washer, "Visual inspection of highway bridges," *Journal of Nondestructive Evaluation*, vol. 21, no. 3, pp. 67-83, 2002.
- [6] F. Wang, L. Huo, and G. Song, "A piezoelectric active sensing method for quantitative monitoring of bolt loosening using energy dissipation caused by tangential damping based on the fractal contact theory," *Smart Materials and Structures*, vol. 27, no. 1, Article ID 15023, 2017.
- [7] F. Wang, Z. Chen, and G. Song, "Monitoring of multi-bolt connection looseness using entropy-based active sensing and genetic algorithm-based least square support vector machine," *Mechanical Systems and Signal Processing*, vol. 136, Article ID 106507, 2020.
- [8] Q. Sun, B. Yuan, X. Mu, and W. Sun, "Bolt preload measurement based on the acoustoelastic effect using smart piezoelectric bolt," *Smart Materials and Structures*, vol. 28, no. 5, Article ID 55005, 2019.
- [9] K. K. Maurya, A. Rawat, and G. Jha, "Smart materials and electro-mechanical impedance technique: a review," *Materials Today: Proceedings*, vol. 33, pp. 4993-5000, 2020.
- [10] F. Wang, S. C. M. Ho, L. Huo, and G. Song, "A novel fractal contact-electromechanical impedance model for quantitative monitoring of bolted joint looseness," *IEEE Access*, vol. 6, pp. 40212-40220, 2018.
- [11] S. Fan, W. Li, Q. Kong, Q. Feng, and G. Song, "Monitoring of pin connection loosening using electromechanical impedance: numerical simulation with experimental verification," *Journal of Intelligent Material Systems and Structures*, vol. 29, no. 9, pp. 1964-1973, 2018.
- [12] L. Zhou, S.-X. Chen, Y.-Q. Ni, and A. W.-H. Choy, "EMI-GCN: a hybrid model for real-time monitoring of multiple bolt looseness using electromechanical impedance and graph convolutional networks," *Smart Materials and Structures*, vol. 30, no. 3, Article ID 35032, 2021.
- [13] J. Moll, C. Kexel, S. Pötzsch, M. Rennoch, and A. S. Herrmann, "Temperature affected guided wave propagation in a composite plate complementing the Open Guided Waves Platform," *Scientific Data*, vol. 6, no. 1, pp. 191-199, 2019.
- [14] J.-H. Park, T.-C. Huynh, S.-H. Choi, and J.-T. Kim, "Vision-based technique for bolt-loosening detection in wind turbine tower," *Wind and Structures*, vol. 21, no. 6, pp. 709-726, 2015.

- [15] P. Mukhopadhyay and B. B. Chaudhuri, "A survey of Hough transform," *Pattern Recognition*, vol. 48, no. 3, pp. 993–1010, 2015.
- [16] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognition*, vol. 34, no. 3, pp. 721–725, 2001.
- [17] Y.-J. Cha, K. You, and W. Choi, "Vision-based detection of loosened bolts using the Hough transform and support vector machines," *Automation in Construction*, vol. 71, pp. 181–188, 2016.
- [18] Y. Zhang, X. Sun, K. J. Loh, W. Su, Z. Xue, and X. Zhao, "Autonomous bolt loosening detection using deep learning," *Structural Health Monitoring*, vol. 19, no. 1, pp. 105–122, 2020.
- [19] C. Yuan, W. Chen, H. Hao, and Q. Kong, "Near real-time bolt-loosening detection using mask and region-based convolutional neural network," *Structural Control and Health Monitoring*, vol. 28, no. 7, Article ID e2741, 2021.
- [20] H. Gong, X. Deng, J. Liu, and J. Huang, "Quantitative loosening detection of threaded fasteners using vision-based deep learning and geometric imaging theory," *Automation in Construction*, vol. 133, Article ID 104009, 2022.
- [21] X. Yang, Y. Gao, C. Fang, Y. Zheng, and W. Wang, "Deep learning-based bolt loosening detection for wind turbine towers," *Structural Control and Health Monitoring*, vol. 29, no. 6, Article ID e2943, 2022.
- [22] L. Ge, D. Dan, and H. Li, "An accurate and robust monitoring method of full-bridge traffic load distribution based on YOLO-v3 machine vision," *Structural Control and Health Monitoring*, vol. 27, no. 12, Article ID e2636, 2020.
- [23] J. Wu, X. Li, and Y. Zhou, "An infrared image detection of power equipment based on super-resolution reconstruction and YOLOv4," *Journal of Engineering*, vol. 2022, no. 10, pp. 1006–1016, 2022.
- [24] X. Deng, J. Liu, and H. Gong, "A novel vision-based method for loosening detection of marked T-junction pipe fittings integrating GAN-based segmentation and SVM-based classification algorithms," *Journal of Intelligent Manufacturing*, 2022.
- [25] X. Deng, J. Liu, H. Gong, and J. Huang, "Detection of loosening angle for mark bolted joints with computer vision and geometric imaging," *Automation in Construction*, vol. 142, Article ID 104517, 2022.
- [26] T.-C. Huynh, J.-H. Park, H.-J. Jung, and J.-T. Kim, "Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing," *Automation in Construction*, vol. 105, Article ID 102844, 2019.
- [27] T.-C. Huynh, "Vision-based autonomous bolt-looseness detection method for splice connections: design, lab-scale evaluation, and field application," *Automation in Construction*, vol. 124, Article ID 103591, 2021.
- [28] X. Zhao, Y. Zhang, and N. Wang, "Bolt loosening angle detection technology using deep learning," *Structural Control and Health Monitoring*, vol. 26, no. 1, Article ID e, 2019.
- [29] Y. Yu, Y. Liu, J. Chen, D. Jiang, Z. Zhuang, and X. Wu, "Detection method for bolted connection looseness at small angles of timber structures based on deep learning," *Sensors*, vol. 21, no. 9, p. 3106, 2021.
- [30] A. Newell, K. Yang, and J. Deng, *Stacked Hourglass Networks for Human Pose Estimation*, *European Conference on Computer Vision*, Springer, Berlin, Germany, 2016.
- [31] Y. Zhang, G. Zhang, B. Guan, and J. Yao, "Detection of human lower limb mechanical axis key points and its application on patella misalignment detection," *Applied Intelligence*, vol. 52, no. 5, pp. 5385–5399, 2021.
- [32] A. Simoni, A. D'Eusano, S. Pini, G. Borghi, and R. Vezzani, "Improving car model classification through vehicle keypoint localization," in *Proceedings of the 16th International Conference on Computer Vision Theory and Applications*, Vienna, Austria, September 2021.
- [33] C. Liang, K. Lundeen, W. McGee, C. Menassa, S. Lee, and V. Kamat, "Stacked hourglass networks for markerless pose estimation of articulated construction robots," in *Proceedings of the 35th International Symposium on Automation and Robotics in Construction*, Berlin, Germany, July 2018.
- [34] H. Luo, M. Wang, P. K.-Y. Wong, and J. C. Cheng, "Full body pose estimation of construction equipment using computer vision and deep learning techniques," *Automation in Construction*, vol. 110, Article ID 103016, 2020.
- [35] T.-C. Nguyen, T.-C. Huynh, J.-Y. Ryu, J.-H. Park, and J.-T. Kim, "Bolt-loosening identification of bolt connections by vision image-based technique," *Nondestructive Characterization and Monitoring of Advanced Materials Aerospace, and Civil Infrastructure*, vol. 9804, pp. 227–243, 2016.
- [36] H. C. Pham, Q.-B. Ta, J.-T. Kim, D.-D. Ho, X.-L. Tran, and T.-C. Huynh, "Bolt-loosening monitoring framework using an image-based deep learning and graphical model," *Sensors*, vol. 20, no. 12, p. 3382, 2020.
- [37] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*, Machine Learning Mastery, San Juan, Puerto Rico, 2019.
- [38] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008.
- [39] D. Crockford, "The application/json media type for javascript object notation (json)," p. 4627, 2006, <https://www.rfc-editor.org/rfc/rfc4627.html>.
- [40] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago, Chile, December 2015.
- [41] A. Gupta, K. Thakkar, V. Gandhi, and P. Narayanan, "Nose, eyes and ears: head pose estimation by locating facial keypoints," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1977–1981, Brighton, UK, May 2019.
- [42] J. Yang, Q. Liu, and K. Zhang, "Stacked hourglass network for robust facial landmark localisation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 79–87, Honolulu, HI, USA, July 2017.
- [43] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: new benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3686–3693, Washington, DC, USA, June 2014.
- [44] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude," *Coursera: Neural networks for machine learning*, vol. 4, pp. 26–31, 2017.
- [45] Z. Zhang, "Improved adam optimizer for deep neural networks," in *Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQOS)*, pp. 1-2, Banff, Canada, June 2018.
- [46] Y. Bai, E. Yang, B. Han et al., "Understanding and improving early stopping for learning with noisy labels," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24392–24403, 2021.