Hindawi
Structural Control and Health Monitoring
Volume 2023, Article ID 8817058, 17 pages
https://doi.org/10.1155/2023/8817058

Research Article

Computer Vision-Based Autonomous Method for Quantitative Detection of Loose Bolts in Bolted Connections of Steel Structures

Wulve Lao,[1] **Chuang Cui**,[1] **Dengke Zhang**,[1] **Qinghua Zhang**,[1] **and Yi Bao**[2]

[1]*School of Civil Engineering, Southwest Jiaotong University, Chengdu 610031, China*
[2]*Department of Civil, Environmental and Ocean Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA*

Correspondence should be addressed to Chuang Cui; ccui@swjtu.edu.cn

Received 6 February 2023; Revised 22 April 2023; Accepted 15 May 2023; Published 25 May 2023

Academic Editor: Lin Chen

In this study, an autonomous computer vision-based method is presented to quantitatively detect loose bolts. The method integrates keypoint detection via YOLOv5 and PIPNet, distortion correction via perspective transformation, and rotation angles quantification via geometric imaging. Distortion correction is incorporated to address skewed angles and improve the accuracy of rotation angles. A representative experiment on bolted connection of steel structures is conducted to evaluate the presented approach. The effects of the focal distance, skewed angle, and lighting conditions on the detection and quantification performance are evaluated by varying the imaging conditions. The results demonstrate that the presented approach automatically detects all bolts and their corners, irrespective of the imaging conditions. No false detection occurs, and the quantification errors are lower than 1°. The proposed method can be deployed for automatic detection of loose bolts and quantification of rotation angles in bolted connections under different imaging conditions.

1. Introduction

Bolted connections are widely used in engineering structures due to the advantages of easy construction [1]. However, the bolts are prone to loosening under operation and environmental effects such as dynamic loads and extreme temperature. Loosening of bolts compromises the load capacity of the connections and may lead to catastrophic consequences such as structural collapse. The traditional approaches to detect loosening use torque wrench and percussion, which however are costly and labor-intensive. The approaches also involve delayed detection and have limited reliability as the detection results are dependent on the experience of the inspector. Moreover, the safety of inspectors is another concern in the cases of bridges and transmission towers. It is significant to develop more effective and efficient approaches to assess the condition of bolted connections and detect loosening timely.

Various sensing technologies have been proposed to detect loosening of bolts. The previous efforts mainly focused on the development of advanced sensors and data analysis techniques such as mechanistic-data fusion [2–4]. Various types of sensors, such as piezoelectric sensors [5], strain gauges [6], and fiber optic sensors [7], and acoustic collector [8] were deployed on structures to assess bolting forces [9]. The sensor data often involve substantial nonlinearity and uncertainties, making it difficult to accurately assess loose bolts. Advanced data analysis approaches were proposed to handle the complex sensor data [10–12]. The advances in sensors and data analysis showed successes in prior research, but they require sophisticated sensors and sensing instruments to collect unique sensor data. The deployment of sensors can be inconvenient in many scenarios.

Machine learning-based computer vision methods are attracting increasing attention [13–33]. Some early research classified loose and tight bolts using support vector machines combined with the Hough transform and the Canny edge detection [13] or the Viola–Jones detection [14]. However, it was difficult to use those approaches to detect bolts from images with complex backgrounds. Improved detection performance was achieved by convolutional neural networks (CNNs) such as region-based CNN (RCNN) [15],

fast-RCNN [16], faster-RCNN [17], mask RCNN [18], you only look once (YOLO) [19], and single shot multibox detector (SSD) [20]. Those CNNs were trained using large datasets to classify bolts into loose and tight for qualitative assessment of bolts [21–23].

To enable quantitative assessment, the rotation angles of loose bolts were calculated [24, 26]. Perspective transformation was applied to correct the geometric distortion of the images of bolt connections [26]. The rotation angles of loose bolts were determined by quantifying the rotation angles of the edge lines of the bolts, which were detected by using the Hough transform line detection algorithm [27, 28]. Images of bolts were synthesized to enlarge the dataset size by supplementing the real photos of bolts [29]. The efficiency of these approaches was improved by using efficient algorithms [30]. However, it was difficult to completely detect the edge lines of bolts. To tackle this problem, the YOLO and an optical flow tracking algorithm were integrated to detect the feature points of bolts and quantify the rotation angles of loose bolts [31], but this approach requires a fixed image acquisition device such as a fixed camera. In recent papers [32, 33], a cascaded pyramid network and a keypoint-RCNN were separately used to detect keypoints on exposed screws without having to fix the camera, but these papers focused on a single bolt and the results were subjected to errors due to distortion of photos.

The goal of this research is to develop an approach to detect loose bolts and quantify their rotation angles for connections with many bolts without having to use a fixed image acquisition device. To this end, this study presents a comprehensive framework that integrates the detection of corner points as keypoints for all bolts in a bolted connection, the correction of distortion, and the calculation of rotation angles of loose bolts. There are three main challenges of detecting the corner points of many bolts: (1) some corners are blur in images because the geometry of bolts involves varying focal distances, (2) it is difficult to accurately capture all corners of bolts from connections with many bolts because some corners are partially hidden, and (3) the images involve many variables such as the varying background condition of images. To address these challenges, this research adopted a top-down keypoint detection approach integrating YOLOv5 [34] to detect and locate bolts and a pixel-in-pixel net (PIPNet) [35] to detect the corner points of bolts. The reason for using PIPNet is its high detection accuracy and fast detection speed compared to HRNet and HourglassNet and other classical networks [35].

This research has three novelties: (1) the proposed approach does not require fixed cameras and is applicable to multiple bolts in connections rather than single bolts, (2) a top-down keypoint detection approach is incorporated into detection of the corners of bolts to achieve high accuracy, and (3) distortion correction is incorporated into the presented framework to improve the accuracy and enhance the robustness of the skewed angles of images. This research significantly advances the capability of automatically detecting loose bolts and quantifying the rotation angles of loose bolts for civil engineering structures.

2. Methods

2.1. Overview. Figure 1 shows the proposed framework that consists of three main steps which are as follows:

(1) Detection of the corners through integration of YOLOv5 and PIPNet. The center of the ellipse fitted by the corner points is used as the center of each bolt.

(2) Correction of distortion through the homography matrix of an arbitrary view. This is performed for both the initial and the loose states, and the corner points in the initial state are set as baseline to evaluate the rotation angles in the loose state.

(3) Quantification of the rotation angles of loose bolts.

2.2. Detection of Corner Points. A top-down approach was adopted to detect the corner points of bolts in two steps. First, YOLOv5 is utilized to detect bolts from the original images. Then, PIPNet is utilized to detect the corner points of the bolts with high precision. These two steps are executed sequentially to avoid missing any bolts while achieving high precision in locating the corner points.

2.2.1. Detection of Bolts Using YOLOv5. Figure 2 shows the architecture of the bolt detector based on YOLOv5 [36]. The backbone is responsible for extracting contextual information from the input image, followed by the neck that aggregates as much information as possible, and finally the head that outputs the detection result.

In the backbone, the DarkNet53 [37] uses CSP to preserve as much information as possible as it moves through the network. A focus is used before the DarkNet53 to improve efficiency by reducing computation. A spatial pyramid pooling (SPP) [38] is used after the DarkNet53 to improve the diversity of information scales. The neck is a path aggregation network [39] implemented by upsampling (UP) and concatenating (Concat) to prevent small-object information from being lost when transferred to higher levels of abstraction.

In the head, dense prediction is performed by taking in aggregated feature maps from the neck. The dimensions are $3 \times 80 \times 80 \times 6$, $3 \times 40 \times 40 \times 6$, and $3 \times 20 \times 20 \times 6$. Each input image has 640×640 pixels and is divided into 80×80, 40×40, and 20×20 grid cells. Each grid cell has three anchor boxes, and each anchor box has 6 parameters to be predicted, which are the horizontal and the vertical coordinates, the height of the box, the width of the box, the confidence score, and the category distribution. Finally, nonmaximum suppression (NMS) [40] is performed to filter poorly predicted boxes and only retain the boxes with high response.

Typically, the loss function of YOLOv5 (L_{all}) consists of three components, which are the BBox (bounding box) loss (L_{CIoU}), the object confidence loss (L_{obj}), and the classification loss (L_{cls}). In this study, there is only one class, so L_{cls} is set at zero. Then, L_{all} is equal to L_{CIoU} plus L_{obj}. L_{CIoU} considers the overlap area, the central point distance, and the aspect ratio of BBoxes [41]. The focal loss is adopted for L_{obj} to deal with the unbalanced sample classification [42].

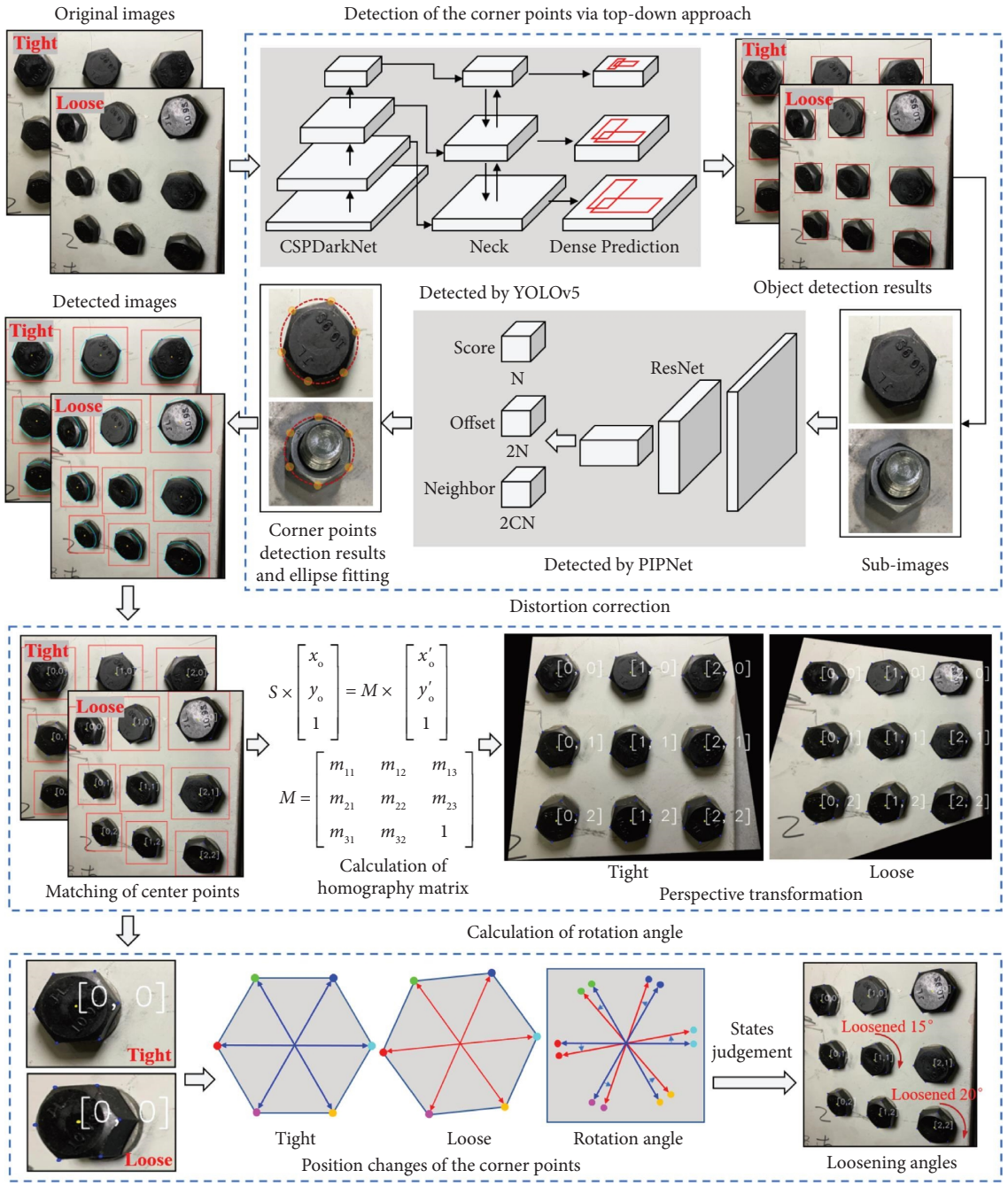


Figure 1: Overview of the multibolt loosening detection method.

Four YOLOv5, namely YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, were investigated to optimize the detection performance [34]. In the four YOLOv5 models, the depth of each CSP is progressively increased, thus increasing the feature extraction and feature fusion capabilities. However, because there is only one class to be detected, the appropriate depth is determined to achieve both high accuracy and high efficiency.

The performance metrics adopted to evaluate the YOLOv5 models were precision (p), recall (r), and dice coefficient (F_1), which are defined as follows [43]:

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{1a}$$

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{1b}$$

$$F_1 = \frac{2\mathrm{pr}}{p + r} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{1c}$$

where TP, FP, and FN are the true positive, false positive, false negative, respectively. T (true) or F (false) means the

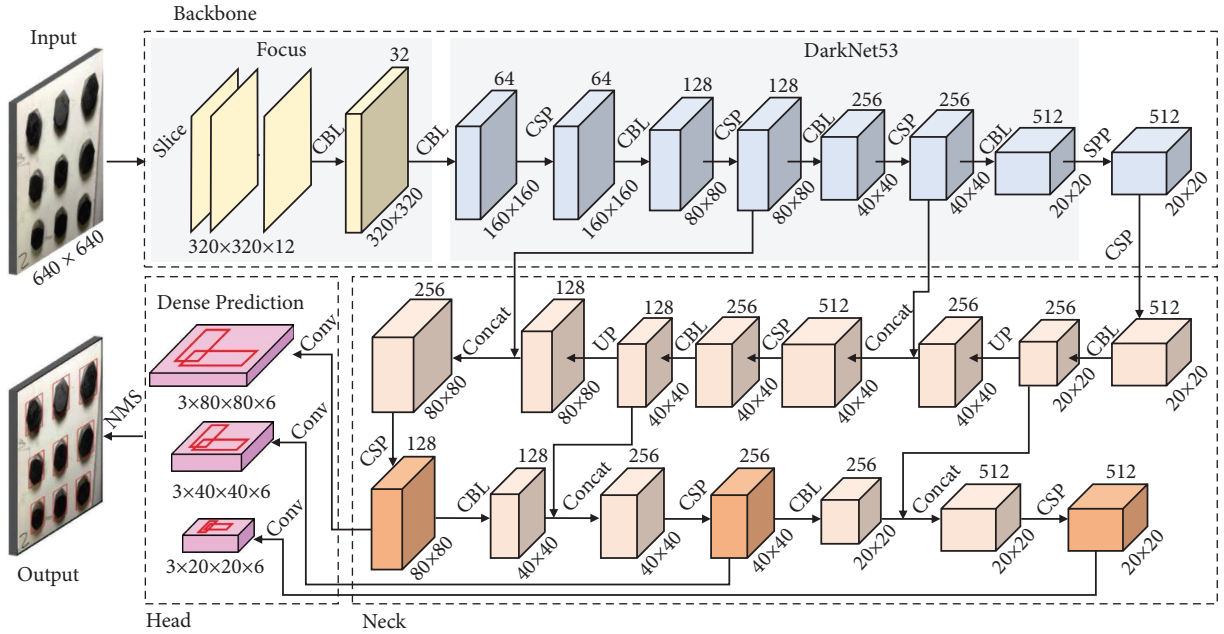


FIGURE 2: Illustration of the architecture of the YOLOv5 for detection of bolts. "Conv" represents convolutional layers; "CBL" represents Conv, batch normalization, and leaky ReLU; and "CSP" represents cross stage partial connections with CBL, Bottleneck, and Conv.

YOLOv5 gives a correct or incorrect detection. P (positive) or N (negative) means the actual result is or is not a bolt. Therefore, p means the proportion of correctly detected bolts to the total bolts, and r means the proportion of correctly detected bolts to the total detected bolts. F_1 is the summed average of the p and r.

2.2.2. Detection of Corners Using PIPNet. Figure 3 shows the architecture of the PIPNet. In the backbone, ResNet [43] pretrained on ImageNet is used to extract information from a single bolt. The default input of the ResNet is 256×256 pixels, so all images of single bolts are resized to 256×256 pixels. The default stride of the ResNet is 32. To get PIPNets with different strides, deconvolutional and convolutional layers are added at the end of the ResNet [44]. Four strides (S), which are 16, 32, 64, and 128, are applied to generate different dimensions (G) of the feature map. In the head, convolutional layers are used to predict the corner points, which are the most likely grids on the heatmap for each point, offset within each grid on the x-axis and y-axis relative to the top-left corner of the grid. The offsets of C number of neighbor corners are also predicted. The outputs are a score map $(6 \times G \times G)$, an offset map $(2 \times 6 \times G \times G)$, and a neighbor map $(2 \times 6 \times C \times G \times G)$, where "6" is the number of corner points of a hexagonal bolt and C is the number of neighbor corners. Each corner point has 5 neighbor corners which have different distances. As the distance varies, C can be equal to 2, 4, or 5. The score, offset, and neighbor maps are independent and computed in parallel to improve efficiency.

Figure 4 shows an example of the score, offset, and neighbor maps of an image ($S = 64$). Only one neighbor is shown. The red dot is the current corner point to be located, and the blue dot is the neighbor corner. The gird with the red dot is denoted as the positive grid. In the score map, the positive grid is 1 and the other grids are 0. In the offset map, the offset distances of the red dot along the x-axis and y-axis are 20% and 40% of the grid side length, respectively, so the positive grid is 0.2 on the x-offset map and 0.4 on the y-offset map. The neighbor maps are determined in a similar way. The offset distances of the blue dot along the x-axis and y-axis are 180% and 40%, respectively, so the positive grid is 1.8 on the x-offset map and 0.4 on the y-offset map. Intuitively, a large S yields higher accuracy for the score maps but also lead to more difficulty for the prediction of the offset and neighbor map. Therefore, a moderate S gives the best overall performance.

The loss function of the PIPNet model is formulated as follows:

$$L = L_S + \lambda L_O + \lambda L_N, \tag{2}$$

where L_S, L_O, and L_N are the losses for the score maps, offset maps, and neighbor maps, respectively, and λ is a balancing coefficient and set to 0.1. The losses for the three types of maps are defined as follows:

$$L_S = \frac{1}{6 \times G \times G} \sum_{i=1}^{6} \sum_{j=1}^{G} \sum_{k}^{G} \left(s_{\text{ijk}}^{*} - s_{\text{ijk}}^{'} \right)^2, s_{ijk}^{*} \in \{0, 1\},$$

$$L_O = \frac{1}{2 \times 6} \sum_{s_{\text{ijk}}^{*}=1} \sum_{l=1}^{2} \left| o_{\text{ijkl}}^{*} - o_{\text{ijkl}}^{'} \right|, o_{\text{ijkl}}^{*} \in [0, 1],$$

$$L_N = \frac{1}{2 \times 6 \times C} \sum_{s_{\text{ijk}}^{*}=1} \sum_{l=1}^{2} \sum_{m=1}^{C} \left| n_{\text{ijklm}}^{*} - n_{\text{ijklm}}^{'} \right|, n_{\text{ijklm}}^{*} \in [0, 1], \tag{3}$$

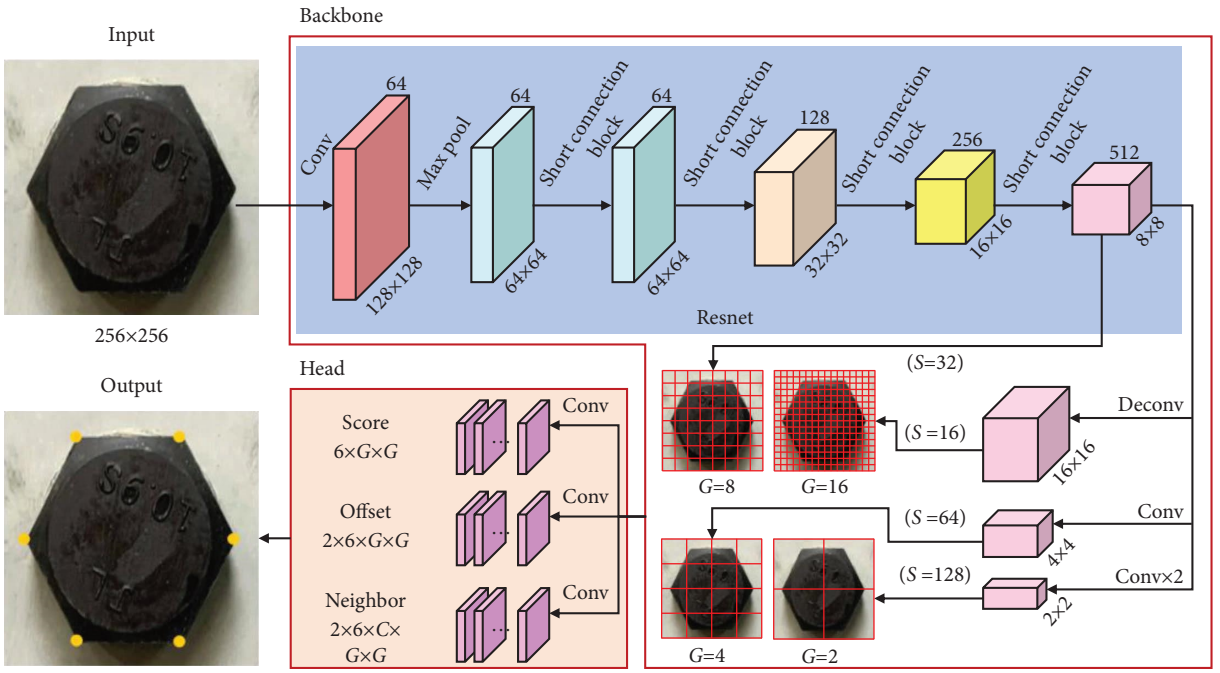


Figure 3: Illustration of the architecture of the PIPNet for detection of the corner points of a bolt.

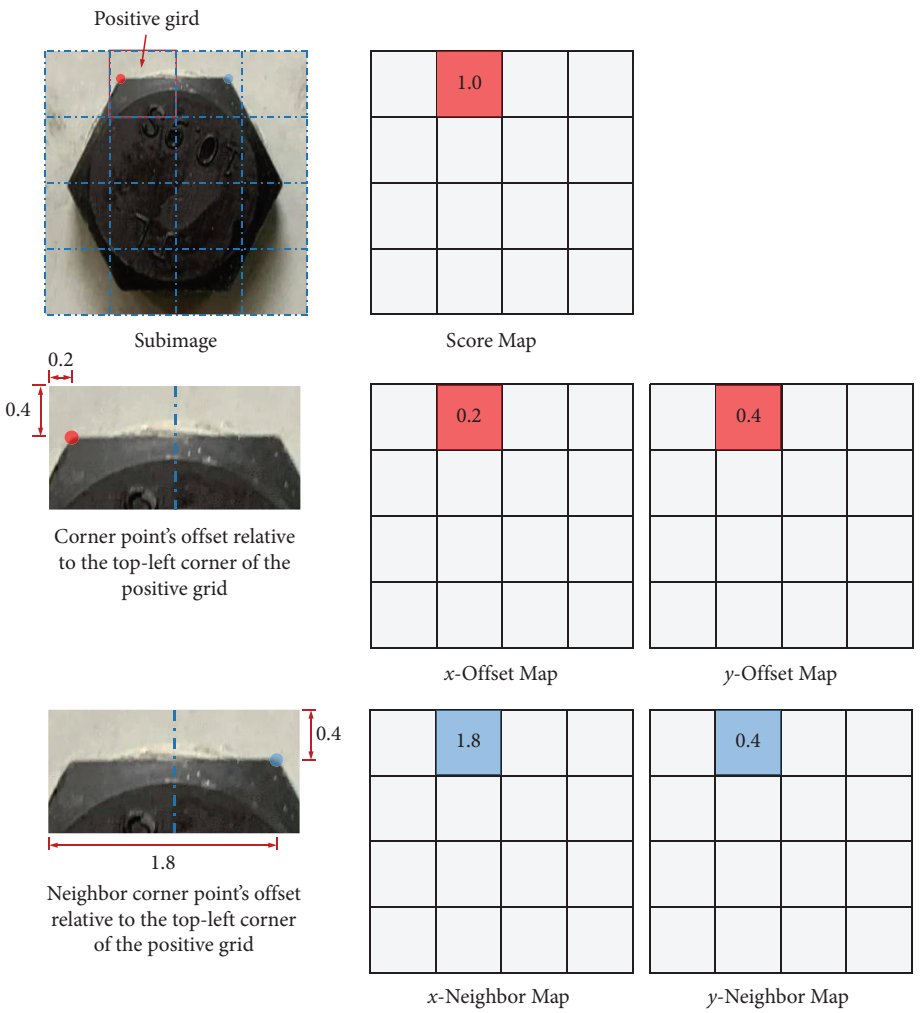


Figure 4: Mapping from a ground-truth landmark to heatmap labels.

where s^*_{ijk}, o^*_{ijkl}, and n^*_{ijklm} denote the ground truth generated from the location of the corner points and $s_{ijkl}{}'$, $o_{ijkl}{}'$, and $n_{ijklm}{}'$ denote the values predicted by the PIPNet.

The PIPNet has two important hyperparameters which are the stride S and the number of neighbors C. To tune the hyperparameters, ResNet-18 was taken as the backbone to train the PIPNets with different S and C. ResNet-50 and ResNet-101 were used to improve the results. MobileNets were also adopted as the backbone since they were efficient and easy to deploy [45]. The performance of different PIPNets was evaluated by the normalized mean error (NME):

$$e_i = \frac{\sqrt{\left(x_i^* - x_i'\right)^2 + \left(y_i^* - y_i'\right)^2}}{(1/6)\sum_{i=1}^{6} d_i}, \tag{4a}$$

$$\text{NME} = \frac{1}{6}\sum_{i=1}^{6} e_i, \tag{4b}$$

where (x_i^*, y_i^*) and (x_i', y_i') are the ground-truth and predicted coordinates of the corner points, respectively, d_i is the distance between adjacent corner points, and e_i is the normalized error of the i-th corner point.

2.3. Distortion Correction. Projective transformation is performed to correct the distortion caused by the skewed angle of images in two steps, as shown in Figure 5. First, the relative positions of all center points are matched. Second, the coordinates of the matched points are used to calculate the homography matrix. The object plane, perspective plane, and viewpoint constitute the initial perspective imaging model. Distortion correction is performed to re-establish the perspective imaging model and project the image onto a reperspective plane parallel to the object plane.

2.3.1. Position Matching of Center Points. Figure 6(a) shows the center points of all bolts. The center points are numbered from O_1 to O_{16}, and their arrangement indicates that the bolts are grouped into two layers, from the outside to the inside of the array of bolts. The set of points closest to O_i forms a polygonal region, V_i, called the Voronoi region of O_i, as shown in Figure 6(b), where the red lines are the boundary of these regions. Such a diagram is called a Voronoi diagram [46], which divides the infinite plane into many regions, denoted as $\{V_1, V_2, ..., V_{16}\}$. When V_i is unbounded, the corresponding O_i is judged as the boundary. In this case, the first layer of center points (O_1 to O_{12}) is classified as the first boundary and they form a quadrilateral when they are connected in a sequence, as indicated by the blue line in Figure 6(b). Similarly, the Voronoi diagram of the second layer of center points (O_{13} to O_{16}) is classified as the second boundary, as shown in Figure 6(c). This process is repeated until all V_i of the Voronoi diagram are unbounded. It should be noted that this classification method is applicable to bolts deployed in rectangular rows, which is common in civil engineering. The last boundary layer may be a point or multiple points along a line.

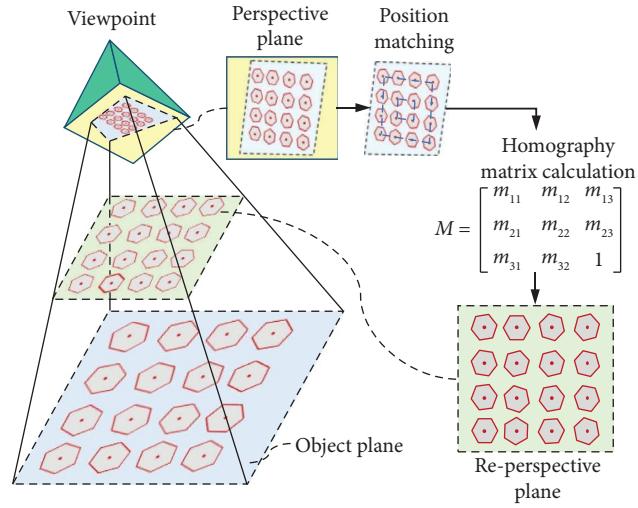


Figure 5: Illustration of the distortion correction.

After the classification, the vertices of the quadrilateral boundaries are determined by the magnitude of the horizontal and the vertical coordinates of the points. For example, given (x_o, y_o) as the pixel coordinates of a point in an image, when the value (x_o+y_o) is the minimum or the maximum among all the points, the point is located at the top left or the bottom right vertex; when the value of (x_o-y_o) is the minimum or the maximum, the point is located at the top right or the bottom left vertex. Therefore, O_1, O_4, O_7, and O_{10} are classified as the four vertices of the first boundary, as depicted in Figure 6(b). Then, the rows and the columns of the rectangular array of the bolts, as well as the position of each bolt can be determined.

2.3.2. Homography Matrix. Perspective transformation is a coordinate system transformation [47], which is expressed by equation (5), where **M** is the homography matrix, (x_o, y_o) and (x_o', y_o') are the pixel coordinates of the center point before and after transformation, respectively, and s is the scale factor. There are eight independent elements in **M**, which relate (x_o, y_o) to (x_o', y_o'), as shown in equation (6). At least four pairs of pixel coordinates before and after the transformation are required to solve the eight elements in **M**. This research uses the center points of all bolts to solve **M**. The optimal solutions of the eight elements are obtained using the least square approach.

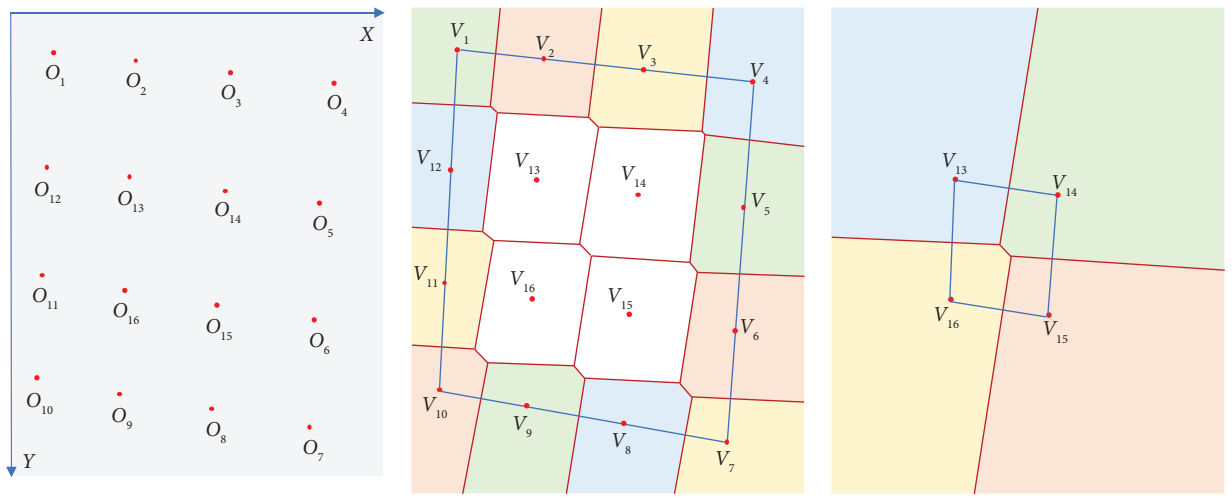


FIGURE 6: Process of position matching: (a) the center points, (b) classification of the first layer, and (c) classification of the second layer.

$$s \times \begin{bmatrix} x_o' \\ y_o' \\ 1 \end{bmatrix} = \mathbf{M} \times \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}, \mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix}, \tag{5}$$

$$\begin{bmatrix} x_{o,1} & y_{o,1} & 1 & 0 & 0 & 0 & -x_{o,1}'x_{o,1} & -x_{o,1}'y_{o,1} \\ 0 & 0 & 0 & x_{o,1} & y_{o,1} & 1 & -y_{o,1}'x_{o,1} & -y_{o,1}'y_{o,1} \\ x_{o,2} & y_{o,2} & 1 & 0 & 0 & 0 & -x_{o,2}'x_{o,2} & -x_{o,2}'y_{o,2} \\ 0 & 0 & 0 & x_{o,2} & y_{o,2} & 1 & -y_{o,2}'x_{o,2} & -y_{o,2}'y_{o,2} \\ & & & & \vdots & & & \\ & & & & \vdots & & & \\ x_{o,n} & y_{o,n} & 1 & 0 & 0 & 0 & -x_{o,n}'x_{o,n} & -x_{o,n}'y_{o,n} \\ 0 & 0 & 0 & x_{o,n} & y_{o,n} & 1 & -y_{o,n}'x_{o,n} & -y_{o,n}'y_{o,n} \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \\ m_{12} \\ m_{22} \\ m_{32} \\ m_{13} \\ m_{23} \end{bmatrix} = \begin{bmatrix} x_{o,1}' \\ y_{o,1}' \\ x_{o,2}' \\ y_{o,2}' \\ \vdots \\ \vdots \\ x_{o,n}' \\ y_{o,n}' \end{bmatrix}. \tag{6}$$

2.4. Rotation Angle of Loose Bolts. Figure 7 illustrates the principle of determining the rotation angle of loose bolts. The corner points of bolts in the tight state are the baseline to evaluate the rotation angles in the loose state. The rotation angle ($\Delta\overline{\theta}$) is calculated as follows:

$$\Delta\overline{\theta} = \frac{1}{6}\sum_{i=1}^{6}\Delta\theta_i, \tag{7}$$

where $\Delta\theta_i\,(i = 1 \sim 6)$ is the rotation angle of each corner point of a bolt. Due to symmetry of the corner points, the discriminable range of rotation angles is 0° to 60°. When the rotation angle is greater than or equal to 60°, the method fails. It is still feasible to use this method to detect and track rotation angles in small steps, since the bolt loosening is slow.

It should be noted that the congruent relationship of the corner points between tight and loose is essential to the rotation angle calculation. The corner points have similar characteristics, but each point is distributed in a different orientation. Therefore, the corner points can be clearly classified according to their relative position. As shown in Figure 8, $OP_0 = [1]$, the vector with the smallest counterclockwise angle to OP_0 is noted as OP_1 and then the other vectors are noted as OP_2 to OP_6 along the counterclockwise direction. When detecting the corner points, the categories of the corner points are determined by their relative positions. Then, after considering the direction of rotation of the loosened bolt, the matching of the congruent relationship before and after loosening can be completed.

There are errors in the detection of the corner points, which affect the accuracy of the rotation angle $\Delta\theta_i\,(i = 1 \sim 6)$. It is necessary to analyze the errors in the angles and determine the threshold (T) of loose bolts. Considering that the corner points should form a positive hexagon, $\theta_i\,(i = 1 \sim 6)$ should be 60°. The value of $\overline{\theta}$ is used to evaluate the angle accuracy as follows:

$$\overline{\theta} = \frac{1}{6}\sum_{i=1}^{6}\left|\theta_i - 60\right|. \tag{8}$$

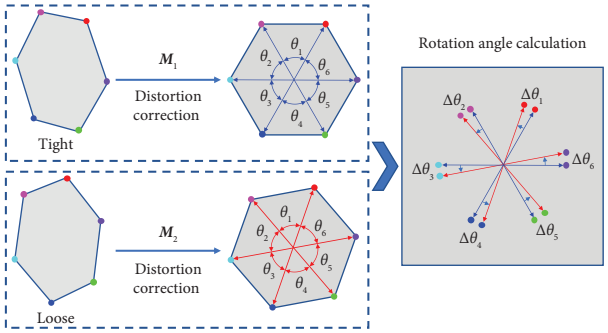


Figure 7: Illustration of the rotation angles of loose bolts.

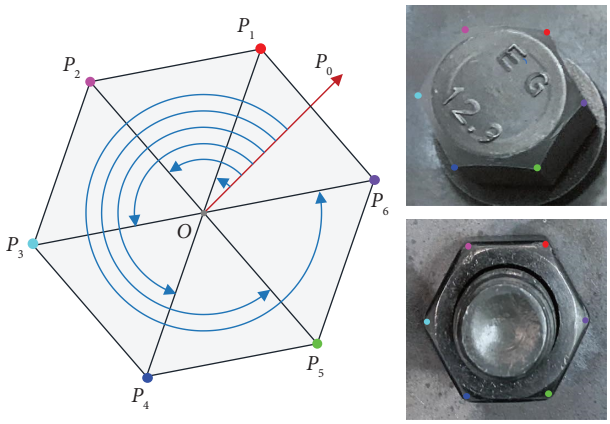


Figure 8: Classification of the corner points.

To simplify the procedure, T is defined by the value of $\overline{\theta}$, as elaborated in the subsequent experimental studies.

3. Experiments and Results

3.1. Preparation and Augmentation of Datasets. Two different datasets were created to train the YOLOv5 and the PIPNet models, separately. A smartphone camera of iPhone 13 was used to acquire images of bolted connections. The shooting angle, distance, and lighting conditions were varied to obtain diverse photos. A total of 1,340 images were acquired. The bolts in the raw photos were manually labelled by the red boxes for training the YOLOv5 model, as shown in Figure 9(a). The red boxes were enlarged by 20% and were subsequently used to extract the subimages, as shown in Figure 9(b). The corner points of those subimages were labelled by yellow dots for training the PIPNet model. The subimages with incomplete corner points were eliminated. Finally, 13,000 subimages were obtained.

To enlarge the datasets, augmentation techniques such as horizontal flipping, rotation, and blurring were applied. In addition, mosaic (random mix of 4 images) and occlusion (random occlusion by rectangular blocks) were applied to the datasets for the YOLOv5 and PIPNet, respectively, as shown in Figure 10.

Mosaic allows multiple images to be passed at the same time to increase the variety of the image background. Occlusion strengthens coherence across neighboring corner points, which can improve the consistency of the neighbor maps. The generated data were combined with the raw data to form the total datasets for the YOLOv5 and the PIPNet models. Each of the two datasets was randomly divided into a training set (80%) and a testing set (20%).

3.2. Training of the YOLOv5. The YOLOv5 models were trained through transfer learning. The model parameters obtained from pretraining by MS COCO (test-dev 2017) dataset were incorporated to make the YOLOv5 converge quickly. The related training parameters included the epochs, batch size, learning rate, and momentum, which were set at 100, 16, 0.01, and 0.937, respectively. The Cosine annealing scheduler was used to adjust the learning rate during the training process. The weight decay and the warmup epochs were set at 0.0005 and 3, respectively. The training process were performed on a desktop computer (GPU: RTX3090 and 24G; CPU: Intel Core i9-10850K, 3.7 GHz, RAM 32G).

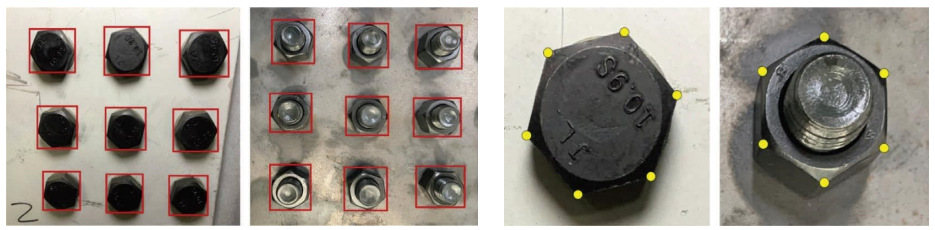
Figure 11 shows the evolution of the loss function of different YOLOv5 models, indicating that the model converged after 80 epochs. Table 1 shows the parameter size, speed, p, r, and F_1 of different YOLOv5 models based on the testing set. As the parameter size increases, the model slows down but F_1 do not show a significant difference. To achieve good results in terms of both accuracy and speed, the YOLOv5l model was selected for the bolt detection.

3.3. Training and Optimization of the PIPNet. The number of training epochs was 80. The learning rate was 0.0001. The equipment used for training PIPNet is the same as that used for training YOLOv5. Figure 12(a) shows the evolution of the loss function of some PIPNets ($S = 16$, 32, 64, and 128; $C = 5$), indicating that these models converged after 40 epochs. These models were evaluated by the NME on the testing set. Figure 12(b) shows the NME of different PIPNets, which indicate that the NME achieves a minimum of 3.8% when $S = 32$ and $C = 4$.

The combination of the two hyperparameters ($S = 32$ and $C = 4$) was used in the subsequent PIPNets. Table 2 shows the parameter size, speed, and NME of different PIPNets. As the backbone grows larger, the trend of decreasing NME is less significant, but the speed decreases rapidly. To achieve good results in terms of both accuracy and speed, the PIPNet with ResNet-50 as backbone was selected for the corner point detection.

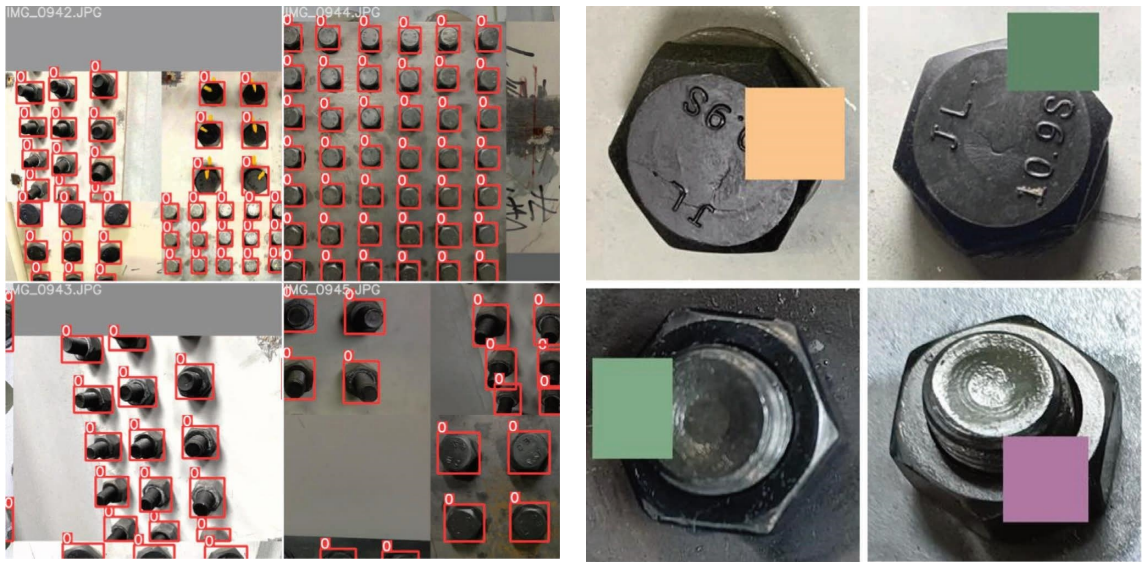
Figure 13 shows representative results of the original and occluded images, where the yellow dots represent the ground truth and the red dots are predicted results. Table 3 shows the specific error data for each bolt. The random occlusion has adverse effects on the predicted results, but the results are acceptable. Overall, the PIPNet has high adequate accuracy and robustness.

To obtain the optimal estimate of the center point position, random sample consensus algorithm (RANSAC) [48] is used to find the possible outliers in six corner points. Ideally, the six corner points are on an ellipse, but this is



(a) (b)

Figure 9: Illustration of the labelled data for (a) YOLOv5 and (b) PIPNet.



(a) (b)

Figure 10: Data augmentation: (a) mosaic for the YOLOv5 and (b) occlusion for the PIPNet.

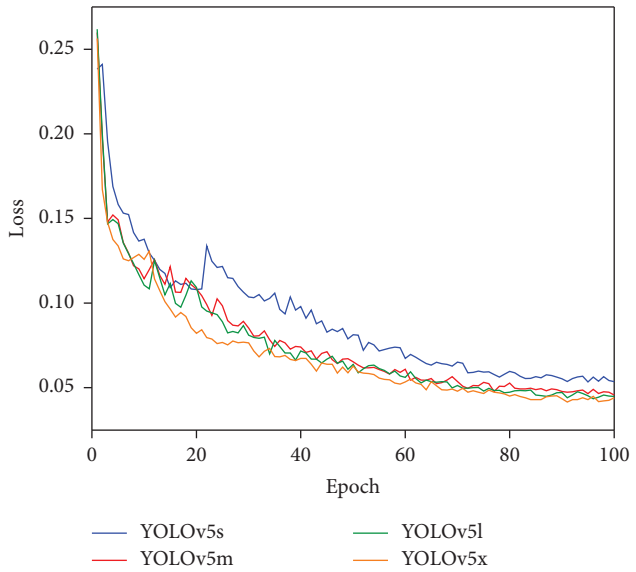


Figure 11: Loss functions of different YOLOv5 models.

Table 1: Parameter size, FPS, p, r, and F_1 of different YOLOv5 models.

Model	#Param. (M)	FPS (GPU)	p	r	F_1
YOLOv5s	7.5	476	0.989	0.991	0.990
YOLOv5m	21.8	333	0.993	0.992	0.993
YOLOv5l	47.8	256	0.996	0.993	0.994
YOLOv5x	89.0	164	0.993	0.998	0.995

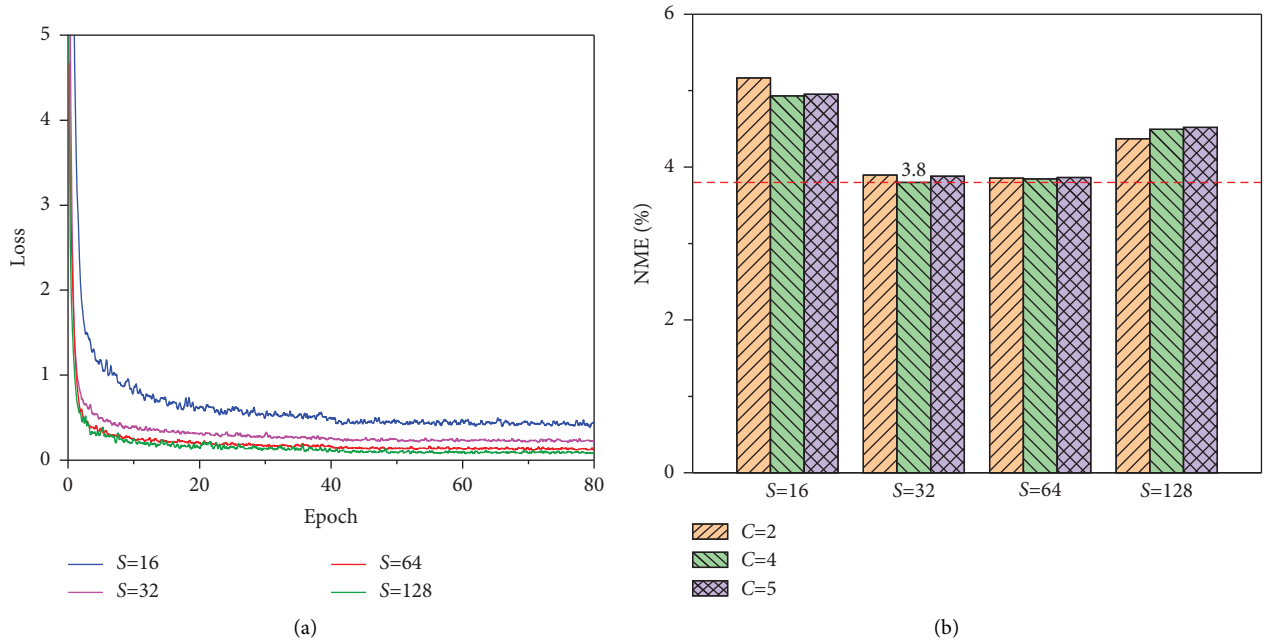


Figure 12: Training of the PIPNets: (a) the loss function and (b) the NME.

Table 2: Parameter size, GFLOPs, and FPS of PIPNets.

Backbone	#Param. (M)	FPS (GPU)	NME (%)
MobileNetV2	4.2	121	4.1
MobileNetV3	4.5	80	3.9
ResNet-18	12.0	200	3.8
ResNet-50	26.7	99	3.4
ResNet-101	45.7	56	3.3

usually not satisfied due to corner point detection errors. RANSAC can take a random sample of five points to solve the elliptic equation based on algebraic relations. The distance between another point and that ellipse can then be calculated. The distances of each corner point from the ellipse solved by the other corner points can be given by repeating the process. The threshold for determining a corner point as an outlier is simply computed as three standard deviation of the six distances. When there are no outliers, the center of the ellipse with six corner points fitted by least squares is the center point. When there is one outlier, the center of the ellipse solved by the other five corner points is the center point. Actually, for the testing set of 2600 subimages, only 75 subimages had two outliers and no subimages had more than two. Therefore, when the number of outliers is greater than one, it is simple and feasible to fit the ellipse using the six corner points. Figure 14 shows representative results with multiple bolts whose corner points are detected and whose center point are fitted by those corner points.

3.4. Results of Distortion Correction. Figure 15(a) shows the results of position matching for the center points of bolts. The row and column numbers of the center points are denoted by (x, y), which indicate the coordinates of the center points. Figure 15(b) shows the images after perspective transportation. Figure 15(c) shows the angle errors $(\overline{\theta})$ of all bolts in the connection.

In general, most of the $\overline{\theta}$ were within 1°, except for some bolts exceeding 1°, and the maximum value was only 1.6°. The abovementioned value indicated high accuracy of the corner point detection and perspective transformation. T was defined as the maximum value of $\overline{\theta}$ and 1°.

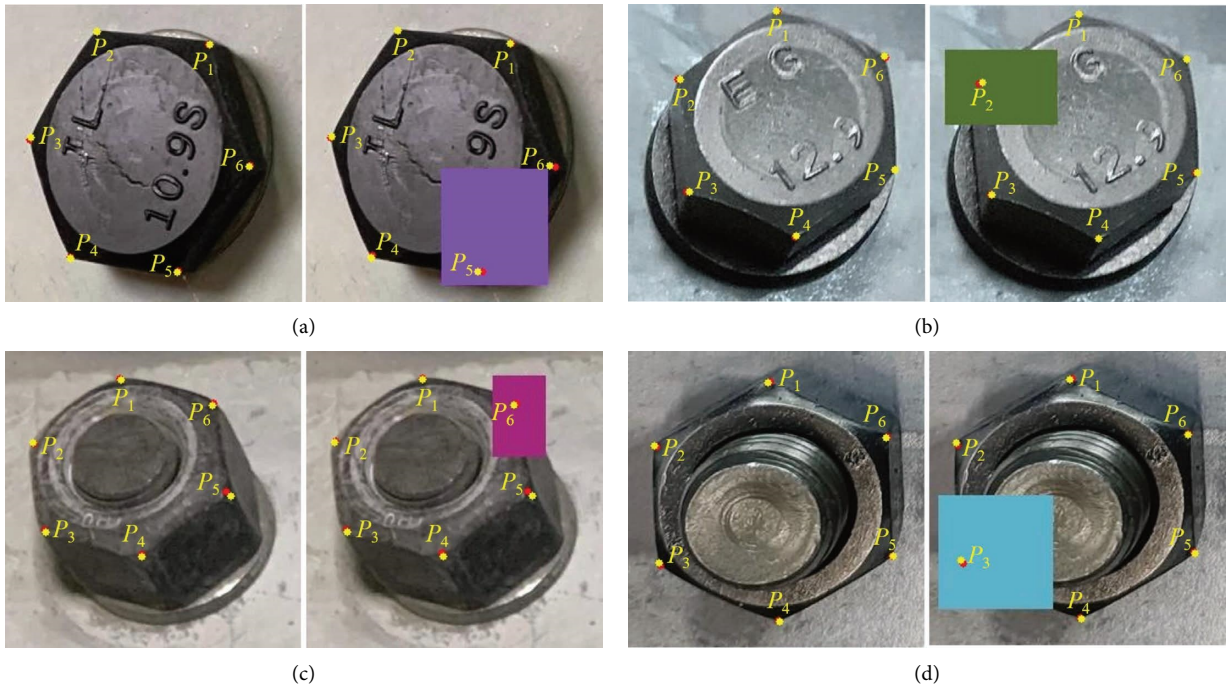


Figure 13: Prediction results of the original and occluded images.

Table 3: Corner point detection error of Figure 13.

Points number	Original images				Occluded images			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
e_1	1.05	1.06	1.22	1.85	0.87	0.80	1.47	2.21
e_2	0.80	1.68	1.42	1.03	0.39	2.74	2.00	1.87
e_3	1.84	1.28	3.85	1.15	1.20	1.16	2.83	3.19
e_4	1.12	1.28	3.00	0.87	1.88	0.78	2.78	1.06
e_5	0.39	0.39	6.92	0.59	3.82	0.73	7.57	0.97
e_6	1.66	0.68	1.75	1.33	4.42	0.18	2.03	1.53
NME	1.14	1.06	3.03	1.14	2.10	1.06	3.11	1.80

Unit: %.

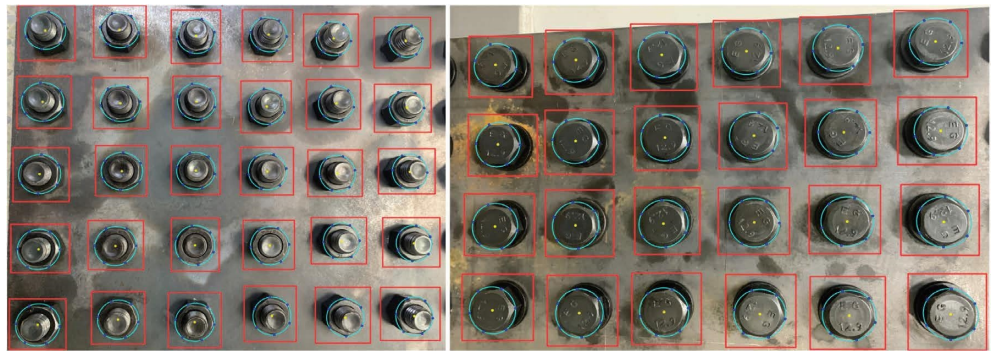


Figure 14: Results of corner point detection and center point fitting for all bolts in the connections.

In addition, the homography matrix calculation method requires the center points to be on the same plane. If the bolt is loose, the plane where the corner points are located theoretically is not on the same plane as before. To analyze this effect, images before and after bolt loosening under the same shooting conditions were used to calculate the homography matrix. Figure 16 shows the results of the corner point detection and homography matrix solution.

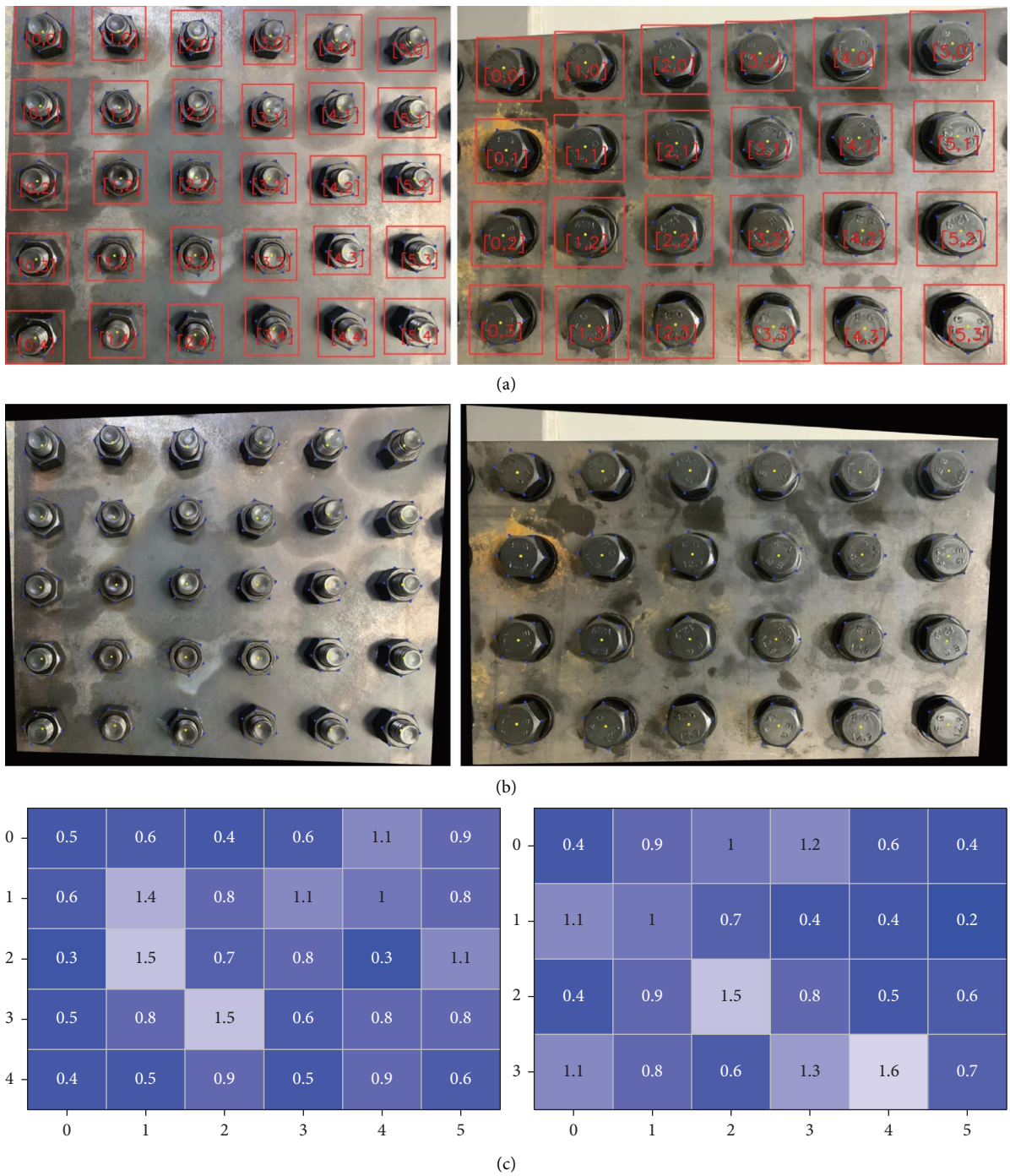
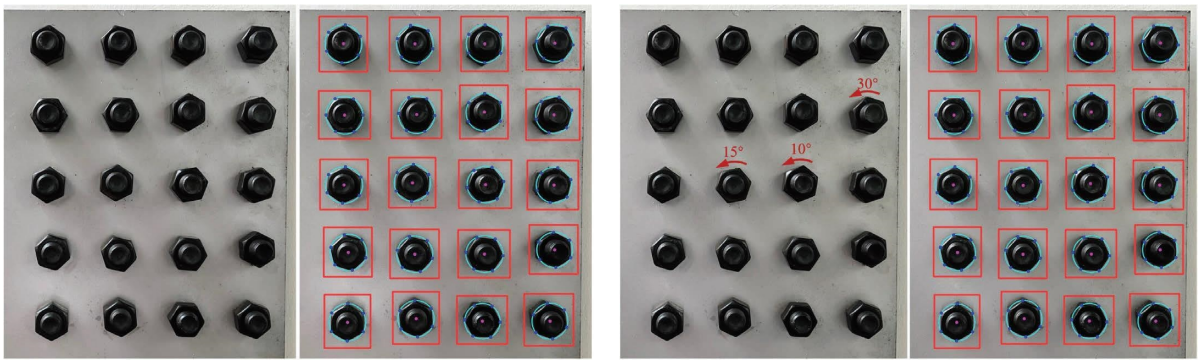


Figure 15: Automatic distortion correction: (a) position matching, (b) perspective transformation, and (c) $\overline{\theta}$ of the bolts (unit: degree).

The bolt loosening has minimal effect on the homography matrix calculation. Therefore, the effect of this factor on the angle calculation will not be considered.

3.5. Validation. Detection of loose bolts was performed for a bolted connection with many bolts under different conditions. A bolted connection with rectangular arrangement of bolts was used, as shown in Figure 17. First, all the bolts were tight, and an image of the connection was captured using a camera. The tight state is denoted by I_0. Then, three bolts were loosened using a torque wrench. The rotation angles were 10°, 25°, and 35°, respectively. Four images of the connection were captured using the same camera under different conditions, denoted by I_1 and I_4. Specifically, I_1 represents the loose state, and the other conditions were the same as those of I_0; I_2 had a distance of 50 cm, which is twice that of I_0; I_3 had a skewed angle of 30° in relation to the angle of zero in I_0 to I_2; and I_4 had dark light while the other conditions were the same as those of I_0.



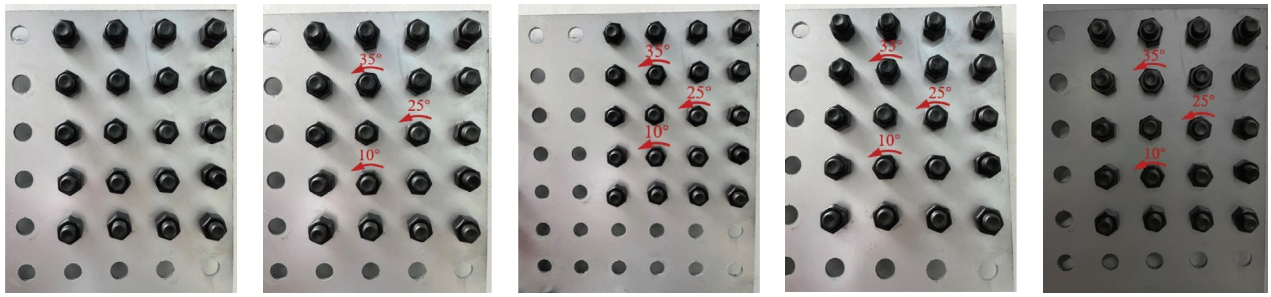
$$M = \begin{bmatrix} 1.31 & -0.04 & 832.38 \\ -1.68 \times 10^{-4} & 1.27 & 880.07 \\ -2.08 \times 10^{-6} & -1.87 \times 10^{-5} & 1 \end{bmatrix}$$

(a)

$$M = \begin{bmatrix} 1.31 & -0.04 & 832.32 \\ -1.55 \times 10^{-4} & 1.27 & 880.22 \\ -2.11 \times 10^{-6} & -1.98 \times 10^{-5} & 1 \end{bmatrix}$$

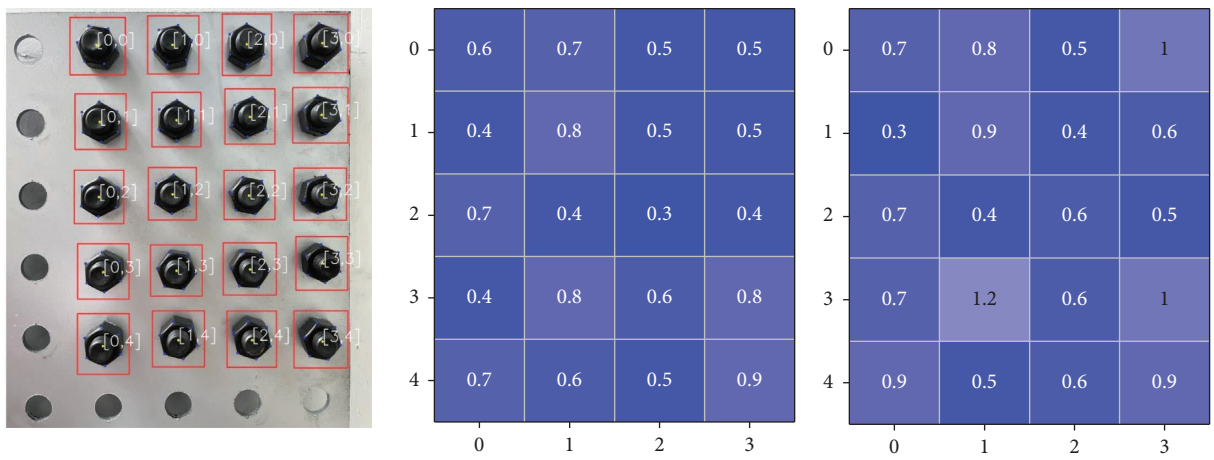
(b)

Figure 16: Corner point detection and homography matrix solving under the same image shooting conditions for (a) tight and (b) loose.



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

Figure 17: Images acquisition under different shooting conditions: (a) I_0: tight (distance = 25 cm); (b) I_1: loose (distance = 25 cm); (c) I_2: loose (distance = 50 cm); (d) I_3: loose (distance = 25 cm and angle = 30 degree); (e) I_4: loose (distance = 25 cm, dark).



(a) (b) (c)

Figure 18: Continued.

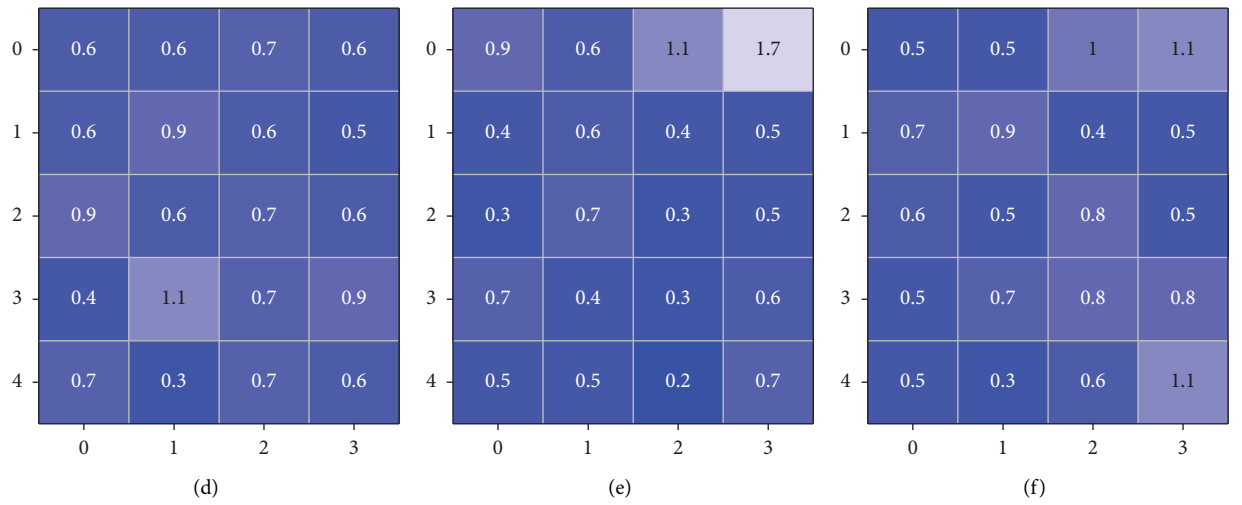


Figure 18: Detection results (unit of angle errors in degree): (a) detection of bolts and corner points, (b) $\overline{\theta}$ in I_0, (c) $\overline{\theta}$ in I_1, (d) $\overline{\theta}$ in I_2, (e) $\overline{\theta}$ in I_3, and (f) $\overline{\theta}$ in I_4.

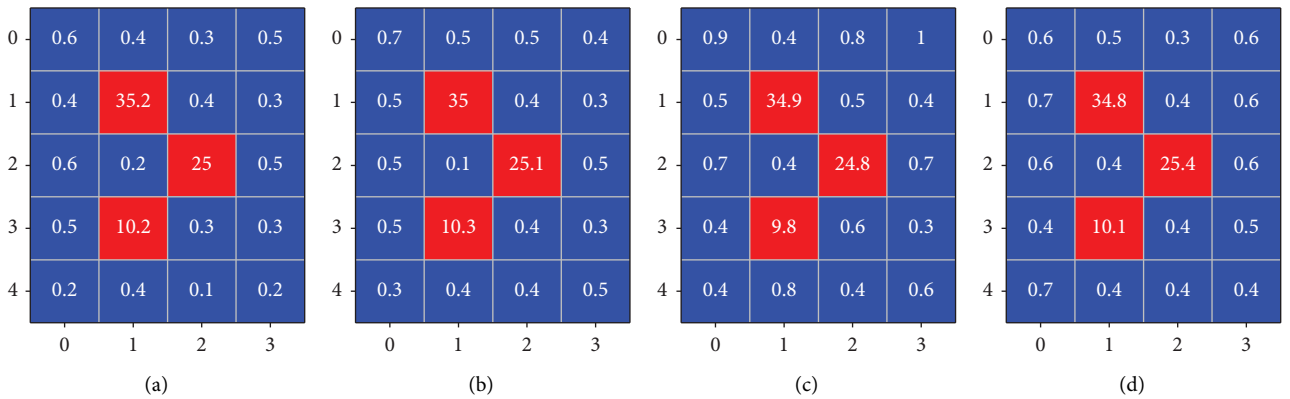


Figure 19: Detection results of rotation angles of bolts: (a) I_1, (b) I_2, (c) I_3, and (d) I_4.

Table 4: Result of rotation angle quantification.

Bolt number	I_1			I_2			I_3			I_4		
	$\Delta\overline{\theta}$	T	Error	$\Delta\overline{\theta}$	T	Error	$\Delta\overline{\theta}$	T	Error	$\Delta\overline{\theta}$	T	Error
[0, 0]	0.6	1.0		0.7	1.0		0.9	1.0		0.6	1.0	
[0, 1]	0.4	1.0		0.5	1.0		0.5	1.0		0.7	1.0	
[0, 2]	0.6	1.0		0.5	1.0		0.7	1.0		0.6	1.0	
[0, 3]	0.5	1.0		0.5	1.0		0.4	1.0		0.4	1.0	
[0, 4]	0.2	1.0		0.3	1.0		0.4	1.0		0.7	1.0	
[1, 0]	0.4	1.0		0.5	1.0		0.4	1.0		0.5	1.0	
[1, 1]	35.2	1.0	0.2	35.0	1.0	0.0	34.9	1.0	−0.1	34.8	1.0	−0.2
[1, 2]	0.2	1.0		0.1	1.0		0.4	1.0		0.4	1.0	
[1, 3]	10.2	1.2	0.2	10.3	1.1	0.3	9.8	1.0	−0.2	10.1	1.0	0.1
[1, 4]	0.4	1.0		0.4	1.0		0.8	1.0		0.4	1.0	
[2, 0]	0.3	1.0		0.5	1.0		0.8	1.1		0.3	1.0	
[2, 1]	0.4	1.0		0.4	1.0		0.5	1.0		0.4	1.0	
[2, 2]	25.0	1.0	0.0	25.1	1.0	0.1	24.8	1.0	−0.2	25.4	1.0	0.4
[2, 3]	0.3	1.0		0.4	1.0		0.6	1.0		0.4	1.0	
[2, 4]	0.1	1.0		0.4	1.0		0.4	1.0		0.4	1.0	
[3, 0]	0.5	1.0		0.4	1.0		1.0	1.7		0.6	1.1	
[3, 1]	0.3	1.0		0.3	1.0		0.4	1.0		0.6	1.0	
[3, 2]	0.5	1.0		0.5	1.0		0.7	1.0		0.6	1.0	
[3, 3]	0.3	1.0		0.3	1.0		0.3	1.0		0.5	1.0	
[3, 4]	0.2	1.0		0.5	1.0		0.6	1.0		0.4	1.1	

Figure 18 shows the detection results for $I_0 - I_4$. The corner points of each bolt under different conditions were detected by the YOLOv5 and the PIPNet, and the angles $\overline{\theta}$ of the bolts were determined after distortion correction. $\overline{\theta}$ in I_3 was 1.7° due to the skewed angle which caused occlusion of corner points.

Figure 19 shows the rotation angles ($\Delta\overline{\theta}$), which are compared with T to judge whether a bolt is tight or loose. Table 4 shows the comparison of $\Delta\overline{\theta}$ and T for all bolts. Most T were 1°, with very few exceeding 1° and a maximum value of 1.7°, indicating a lower limit of 1° for the detectable rotation angle. Most $\Delta\overline{\theta}$ of the tight bolts were less than 1°, except for one bolt in I_3 reached 1.0°. They were all smaller than T, which means there is no false detection. For the loose bolts, whose rotation angles were 10°, 25°, and 35°, respectively, the errors of the $\Delta\overline{\theta}$ were lower than 1°. Overall, the proposed method can detect multiple loose bolts in bolted connections under different imaging conditions.

4. Conclusions

This study presents an approach to detect loose bolts and quantitatively determine the rotation angles of the loose bolts based on advanced computer vision techniques. The presented approach integrates the capabilities of keypoint detection, distortion correction, and rotation angle quantification. Based on the abovementioned investigations, the following conclusions are drawn:

(i) The presented approach is able to detect multiple loose bolts from the connections with many bolts. The YOLOv5 model is able to detect bolts, and the PIPNet model is able to detect the corner points of the bolts under different imaging conditions with varying skewed angles and lighting conditions.

(ii) Four YOLOv5 versions and different PIPNets were investigated to get better detection performance. The YOLOv5l model and the PIPNet with ResNet-50 as backbone were selected for the keypoint detection. Their detection speeds can reach 256 FPS and 99 FPS, respectively, while the dice coefficient of the YOLOv5l is 0.994, and the normalized mean error of the PIPNet is 3.4%.

(iii) The presented approach is able to quantify the rotation angles of loose bolts. In most cases, the lower limit of the detectable rotation angles is up to 1°. There is no false detection under different imaging conditions, and the errors of the rotation angles were lower than 1°.

Based on the established capabilities of loose bolts detection and rotation angles quantification, there are three potential directions for further research in the future as follows:

(i) In the process of creating the PIPNet model dataset, the corner points were manually labelled. However, the accuracy of the labelling process is difficult to guarantee. In the future, advanced digital image processing algorithms will be used to assist the annotation process to reduce human annotation errors.

(ii) The accuracy of corner point detection can be affected by changes in image shooting conditions. To improve the accuracy, it is necessary to add subimages with different shooting conditions to the dataset. Generating subimages using generative adversarial networks [49, 50] will be considered to expand the dataset diversity.

(iii) With the innovation and development of computer vision, the advanced keypoint detection method can be applied to improve the accuracy of corner point detection. This makes it possible to reduce the detectable rotation angle to less than 1°.

Data Availability

The image datasets used to support the findings of this study were produced by the authors, and they are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (grant nos. 2022YFB3706405 and 2022YFB3706404), National Natural Science Foundation of China (grant nos. 52278318, 52108176, and 51978579), and Sichuan Science and Technology Program (grant no. 2021YJ0037).

References

[1] J. Huang, J. Liu, H. Gong, and X. Deng, "A comprehensive review of loosening detection methods for threaded fasteners," *Mechanical Systems and Signal Processing*, vol. 168, Article ID 108652, 2022.

[2] 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.

[3] P. Becht, E. Deckers, C. Claeys, B. Pluymers, and W. Desmet, "Loose bolt detection in a complex assembly using a vibro-acoustic sensor array," *Mechanical Systems and Signal Processing*, vol. 130, pp. 433–451, 2019.

[4] Y. Zhou, S. Wang, M. Zhou, H. Chen, C. Yuan, and Q. Kong, "Percussion-based bolt looseness identification using vibration-guided sound reconstruction," *Structural Control and Health Monitoring*, vol. 29, no. 2, p. 2876, 2022.

[5] C. Hei, M. Luo, P. Gong, and G. Song, "Quantitative evaluation of bolt connection using a single piezoceramic transducer and ultrasonic coda wave energy with the consideration of the piezoceramic aging effect," *Smart Materials and Structures*, vol. 29, no. 2, Article ID 27001, 2020.

[6] T. Wang, B. Tan, G. Lu, B. Liu, and D. Yang, "Bolt pretightening force measurement based on strain distribution of bolt head surface," *Journal of Aerospace Engineering*, vol. 33, no. 4, Article ID 4020034, 2020.

[7] Z. Wang, M. Liu, Z. Zhu et al., "Clamp looseness detection using modal strain estimated from FBG based operational modal analysis," *Measurement*, vol. 137, pp. 82–97, 2019.

[8] C. Yuan, S. Wang, Y. Qi, and Q. Kong, "Automated structural bolt looseness detection using deep learning-based prediction model," *Structural Control and Health Monitoring*, vol. 29, no. 3, p. 2899, 2022.

[9] X. Quan, H. Lv, C. Liu et al., "An investigation on bolt stress ultrasonic measurement based on acoustic time difference algorithm with adaptive hybrid extended Kalman filter," *Measurement*, vol. 186, Article ID 110223, 2021.

[10] J. Jiang, Y. Chen, J. Dai, and Y. Liang, "Multi-bolt looseness state monitoring using the recursive analytic based active sensing technique," *Measurement*, vol. 191, Article ID 110779, 2022.

[11] J. Pal, S. Sikdar, and S. Banerjee, "A deep-learning approach for health monitoring of a steel frame structure with bolted connections," *Structural Control and Health Monitoring*, vol. 29, no. 2, p. 2873, 2022.

[12] W. S. Na, "Bolt loosening detection using impedance based non-destructive method and probabilistic neural network technique with minimal training data," *Engineering Structures*, vol. 226, Article ID 111228, 2021.

[13] 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.

[14] L. Ramana, W. Choi, and Y. J. Cha, "Fully automated vision-based loosened bolt detection using the Viola–Jones algorithm," *Structural Health Monitoring*, vol. 18, no. 2, pp. 422–434, 2019.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[16] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, March 2017.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[20] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Heidelberg, Germany, September 2016.

[21] 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.

[22] 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, p. 2741, 2021.

[23] 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, p. 2943, 2022.

[24] X. Zhao, Y. Zhang, and N. Wang, "Bolt loosening angle detection technology using deep learning," *Structural Control and Health Monitoring*, vol. 26, no. 1, p. 2292, 2019.

[25] 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.

[26] C. Wang, N. Wang, S. C. Ho, X. Chen, and G. Song, "Design of a new vision-based method for the bolts looseness detection in flange connections," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1366–1375, 2020.

[27] 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.

[28] 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.

[29] 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.

[30] Q. B. Ta and J. T. Kim, "Monitoring of corroded and loosened bolts in steel structures via deep learning and Hough transforms," *Sensors*, vol. 20, no. 23, p. 6888, 2020.

[31] X. Pan and T. Y. Yang, "Image-based monitoring of bolt loosening through deep-learning-based integrated detection and tracking," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 10, pp. 1207–1222, 2022.

[32] 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.

[33] 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.

[34] P. Guo, X. Meng, W. Meng, and Y. Bao, "Monitoring and automatic characterization of cracks in strain-hardening cementitious composite (SHCC) through intelligent interpretation of photos," *Composites Part B: Engineering*, vol. 242, Article ID 110096, 2022.

[35] H. Jin, S. Liao, and L. Shao, "Pixel-in-pixel net: towards efficient facial landmark detection in the wild," *International Journal of Computer Vision*, vol. 129, no. 12, pp. 3174–3194, 2021.

[36] G. Jocher, A. Stoken, J. Borovec et al., "ultralytics/yolov5: v5. 0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," 2021, https://ui.adsabs.harvard.edu/abs/2021zndo...4679653J/abstract.

[37] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: a new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 390-391, Seattle, WA, USA, July 2020.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[39] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768, College Park, MD, USA, July 2018.

[40] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS: improving object detection with one line of code," in *Proceedings of the European Conference on Computer Vision*, pp. 5561–5569, College Park, MD, USA, June 2017.

[41] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12993–13000, 2020.

[42] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 2980–2988, 2017.

[43] P. Guo, W. Meng, and Y. Bao, "Automatic identification and quantification of dense microcracks in high-performance fiber-reinforced cementitious composites through deep learning-based computer vision," *Cement and Concrete Research*, vol. 148, Article ID 106532, 2021.

[44] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2018–2025, IEEE, Barcelona, Spain, November 2011.

[45] A. Howard, A. Zhmoginov, L. C. Chen, M. Sandler, and M. Zhu, "Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation," 2018, https://arxiv.org/abs/1801.04381.

[46] F. Aurenhammer, "Voronoi diagrams-a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[47] S. J. Yang, C. C. Ho, J. Y. Chen, and C. Y. Chang, "Practical homography-based perspective correction method for license plate recognition," in *Proceedings of the 2012 International Conference on Information Security and Intelligent Control*, pp. 198–201, IEEE, Yunlin, Taiwan, August, 2012.

[48] B. Xu and C. Liu, "A 3D reconstruction method for buildings based on monocular vision," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 3, pp. 354–369, 2022.

[49] B. Xu and C. Liu, "Pavement crack detection algorithm based on generative adversarial network and convolutional neural network under small samples," *Measurement*, vol. 196, Article ID 111219, 2022.

[50] C. Liu and B. Xu, "A night pavement crack detection method based on image-to-image translation," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 13, pp. 1737–1753, 2022.