Deep Learning Framework for Total Stress Detection of Steel Components

Wei Wang, Ph.D[1]; Peng Shi[2]; Honghu Chu[3]; Lu Deng, M.ASCE[4]; and Banfu Yan, Ph.D[5]

Abstract: Accurately obtaining the stress of steel components is of great importance for the condition assessment of civil structures. Currently, methods for structure stress detection have some drawbacks such as the capability of obtaining structural stress increment rather than the total stress, causing structural damage, and high cost. To overcome these drawbacks, a deep learning framework for total stress detection of steel components is proposed and its feasibility is illustrated with an example. First, the adopted deep neural network is briefly introduced, followed by the introduction of the dataset preparation. In order to maximize the stress detection accuracy, parameter analysis was conducted and the mean average precision achieved by the well-trained model for detection of the stresses under consideration is 89.67%. The robustness of the trained model was further examined and the procedures for application of the proposed approach were summarized. The presented method provides a new idea to detect the total stress of structure components that is difficult to obtain with a traditional sensor-based method.
DOI: 10.1061/(ASCE)BE.1943-5592.0001655. *© 2020 American Society of Civil Engineers*.

Author keywords: Total stress detection; Deep learning; Steel components; Parameter optimization.

Introduction

Accurately obtaining the stress of steel components is of great importance for the condition assessment of a structure. Existing methods for structure stress detection can be generally sorted into two classes, including the local destructive testing (DT) methods and the nondestructive testing (NDT) methods. The local DT methods, including the blind-hole drilling method (Weng and Chen 1993), the ring-core method (Zuccarello 1999), and the hole drilling method (Sánchez-Beitia and Roca 2014), are stress-relief techniques. Typical NDT methods include electrical methods [based on electric resistance strain gauges (Li et al. 2005) and/or vibrating wire strain gauges (Han et al. 2017)], magnetic methods [including the Barkhausen noise method (Franco Grijalba and Padovese 2018), the magneto acoustic emission method (Stupakov et al. 2017), the metal magnetic memory method (Venkatachalapathi et al. 2018)], the ultrasonic method (Kurashkin et al. 2019), and the X-ray diffraction method (Brar and Singh 2014; Turan et al. 2019). However, both the DT and the NDT methods have some drawbacks. The DT methods cause structural damage and usually require a complex operation. On the other hand, the NDT methods, for instance, the electrical methods, are usually only capable of obtaining the incremental stress rather than total stress of structures. The magnetic methods are only applicable for stress detection of ferromagnetic materials and are significantly affected by the magnetic characteristics of ferromagnetic materials. In addition, the detection accuracy of the ultrasonic method is usually low while the cost of the X-ray diffraction method is quite expensive due to its high demand on equipment and operation environment.

During the past few years, deep learning methods have been widely applied in many aspects of civil structures. One of the most important applications is related to crack detection, with focus on crack detection and classification (Cha et al. 2017; Li et al. 2020), obtaining crack width and length (Ni et al. 2019; Yang et al. 2018), and developing techniques for autonomous crack detection (Dung and Anh 2019; Kang and Cha 2018). Some other researchers focused on identifying the type of damages in civil structures based on deep learning-based approaches (Cha et al. 2018; Wang et al. 2019). Furthermore, some scholars presented a deep learning framework for assessing the remaining fatigue life of bridge steel components (Fathalla et al. 2018; Yan et al. 2019), while others developed deep convolutional neural network (CNN)-based methods for revealing the unknown correlation between structural damage and measured structure accelerations (Abdeljaber et al. 2017) and/or strains (Gulgec et al. 2017; Weinstein et al. 2018). In addition, Lee et al. (2018) utilized a deep learning algorithm to optimize the design of a ten-bar truss and demonstrated the advantages of such an algorithm over traditional neural networks. Recently, Dick et al. (2019) established a deep learning-based intelligent machine-vision platform for detecting and forecasting threats that are likely to cause structure failures. However, to the authors' best knowledge, there has been no application of deep learning approaches for the stress detection of civil structures.

The purpose of this paper is to present a deep learning framework for total stress detection of steel components. This paper is organized as follows. In Part 2, the development of the adopted deep learning approach, that is, the Faster region-based CNN (Faster R-CNN), is briefly described. In Part 3, the details on dataset preparation as well as the implementation of Faster R-CNN are

[1]Assistant Professor, Key Laboratory for Damage Diagnosis of Engineering Structures of Hunan Province, Hunan Univ., Changsha 410082, China; College of Civil Engineering, Hunan Univ., Changsha, Hunan 410082, China. Email: wang_wei@hnu.edu.cn

[2]Graduate Student, College of Civil Engineering, Hunan Univ., Changsha, Hunan 410082, China. Email: shipeng@hnu.edu.cn

[3]Graduate Student, College of Civil Engineering, Hunan Univ., Changsha, Hunan 410082, China. Email: chuhonghu@hnu.edu.cn

[4]Professor, Key Laboratory for Damage Diagnosis of Engineering Structures of Hunan Province, Hunan Univ., Changsha 410082, China; College of Civil Engineering, Hunan Univ., Changsha, Hunan 410082, China (corresponding author). ORCID: https://orcid.org/0000-0002-4113-4895. Email: denglu@hnu.edu.cn

[5]Associate Professor, College of Civil Engineering, Hunan Univ., Changsha, Hunan 410082, China. Email: yanbanfu@hnu.edu.cn

Note. This manuscript was submitted on December 28, 2019; approved on August 13, 2020; published online on November 9, 2020. Discussion period open until April 9, 2021; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Bridge Engineering***, © ASCE, ISSN 1084-0702.

introduced; the indexes for performance evaluation of the trained model are also described. In Part 4, the effects of three important parameters, including the anchor scale, the mini-batch size as well as the learning rate, on stress detection accuracy are investigated; an optimal combination of these parameters for training the Faster R-CNN was obtained. The robustness of the trained model was further examined using new images. In Part 5, strategies on improving the efficiency of the proposed approach are discussed and procedures for application of the proposed approach are summarized. In the last part, the main conclusions from the study are drawn and future research directions are discussed.

Methodology

In the present study, the Faster R-CNN, which is composed of two networks including the region proposal network (RPN) and Fast region-based CNN (Fast R-CNN) that share the same CNN for image feature extraction, was used to detect the stress of structures. Fig. 1 illustrates the architecture of the Faster R-CNN. The function of the RPN is to provide object proposals (OPs) based on these input images, and the function of the Fast R-CNN is to localize these OPs provided by the RPN and to detect stresses. The development of the Faster R-CNN is introduced briefly in this part.

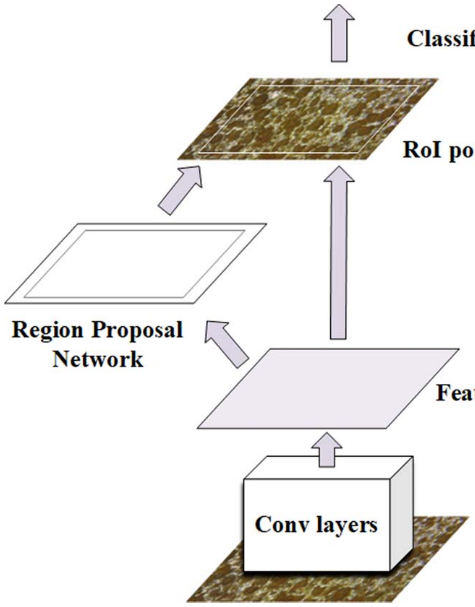

Fig. 1. Overall architecture of the Faster R-CNN.

CNNs

Usually, a CNN consists of some convolutional (CONV) layers, max-pooling layers (MPLs), fully connected (FC) layers, and soft-max layers. The function of CONV layers is to abstract features from input RGB images with a series of three-dimensional kernels (filters) consisting of learnable weights. The function of the MPL is to cut down the input size by performing downsampling and the function of an FC layer is to connect all neurons obtained from its previous layers. The function of a soft-max layer is to make prediction of the class of its input according to the probabilities of the input belonging to each individual class. These probabilities are calculated through a soft-max function based on the features abstracted from an FC layer. The input is sorted into the class with the highest probability. More information about the convolution process can be found from Cha et al. (2017).

RPNs

The function of the RPN, whose overall architecture is illustrated in Fig. 2, is to propose OPs. As shown in Fig. 2, a series of images under consideration are taken as the input of the RPN whose outputs are hundreds and thousands of OPs, each with a probability of being an object. These region proposals are generated by sliding a small network (namely, a 3×3 window) on the feature map output by the last layer of the CNNs (Ren et al. 2017). According to the study of Ren et al. (2017), when a window slides to each location of the feature map, nine region proposals are generated corresponding to nine rectangular boxes that are called anchors. The center of these anchors is the same as that of the sliding window and can be parameterized with eight constants [namely, center of each sliding window: $(x_a,\ y_b)$, three different widths and heights: $(w_a^k,\ h_a^l)$, where $k,\ l = 1,\ 2,\ 3$]. The concept of Intersection-over-Union (IoU) is put forward for calculating the overlap between a candidate box (e.g., an anchor) and a ground-truth box (GTB), as shown in Fig. 3, and the IoU is calculated with

$$\mathrm{IoU} = \frac{\mathrm{Area}(A_C \cap A_{GT})}{\mathrm{Area}(A_C \cup A_{GT})} \tag{1}$$

where $A_C \cap A_{GT}$ = the intersection of the candidate box and the GTB; and $A_A \cup A_{GT}$ = their union. Fig. 4 shows an example of the region proposal of specimen surface under the stress level of 50 MPa. An anchor is designated with a positive label under two different conditions: (1) its IoU overlapping with a GTB achieves the highest value; (2) its IoU overlapping with every GTB is larger than 0.7 (Ren et al. 2017). On the other hand, a nonpositive anchor is designated with a negative (background) label under the condition that its IoU

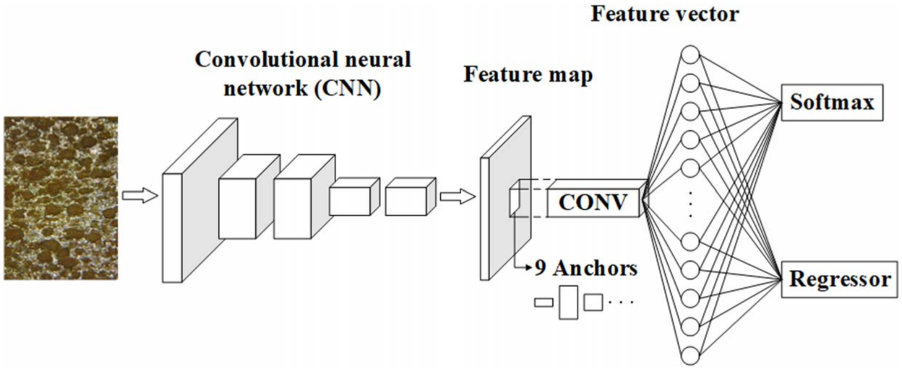

Fig. 2. Overall architecture of the RPN.

overlapping with any GTB is less than 0.3. Anchors that are neither positive nor nonpositive are discarded during the training process.

All sliding windows are mapped to a feature vector through the rectified linear unit (ReLU) activation function (Nair and Hinton 2010). The obtained feature vector is then taken as the input of two fully connected layers that are usually called the box-classification layer and box-regression layer, respectively. After the training process, the box-classification layer computes the probability of being an object in each of the nine generated boxes at the location of every sliding window (Cha et al. 2017), and the box-regression layer calculates the center coordinates, and the width as well as the height of a predicted bounding box that achieves a best match with a GTB (Girshick et al. 2014). The RPN can be trained end-to-end taking advantages of these techniques of stochastic gradient descent (SGD) min-batch as well as backpropagation. Readers can find more details about the RPN from Ren et al. (2017).

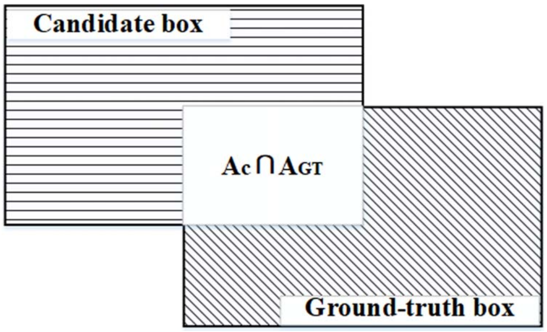

Fig. 3. Illustration of the overlap of the representative bounding box.

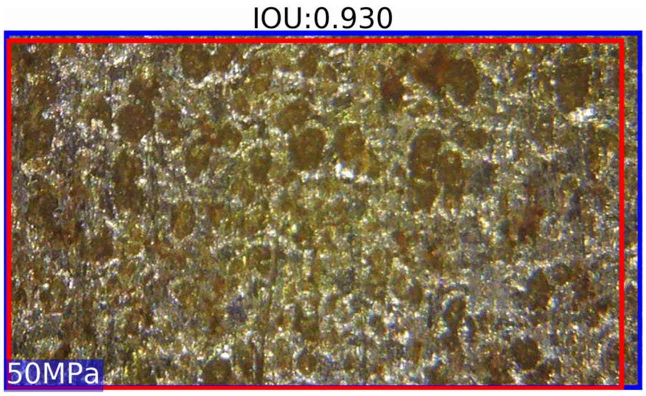

Fig. 4. Example of the region proposal of specimen surface under the stress level of 50 MPa.

Fast R-CNN

The function of the Fast R-CNN, whose overall architecture is illustrated in Fig. 5, is to localize and classify objects in images. As can be observed from Fig. 5, the Fast R-CNN uses CNNs for the abstraction of the feature map of an input image and adopts the OPs provided by the RPN. Corresponding to each OP, a fixed-size feature vector will be abstracted from the feature map by taking advantage of the region of interest (RoI) pooling layer that performs the max pooling operation. Each feature vector is then taken as the input of a series of FC layers followed by two correlated layers, namely, a soft-max layer and a regression layer. With regard to each RoI, the soft-max layer outputs the probability of being each of $p+1$ categories (p training classes + 1 background class) and, in the meantime, the regression layer calculates the outcome of four location parameters including two center coordinates (T_x^u, T_y^u), the height (T_h^u), and the width (T_w^u) of object bounding boxes.

The Fast R-CNN can be trained end-to-end as well by taking advantage of the techniques of SGD min-batch as well as backpropagation. For each labeled RoI, a multiclass loss function given in the following equation was adopted for the joint training of the classification and bounding-box regression:

$$L(p, u, T^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] \sum_{i \in \{x,y,w,h\}} L_{\text{reg}}(T_i^u, v_i) \qquad (2)$$

where L_{cls} = the log loss function; L_{reg} = the regression loss function, both of which can be found in the study of Ren et al. (2017); u and v = the label and location parameters (i.e., center coordinates, width as well as height) of each ground-truth bounding box, respectively. For each RoI, when its IoU overlapping with the bounding box of one ground truth is larger than 0.5, its label is positive ($u = 1$), and it is labeled as background ($u = 0$) when the maximal value of its IoU overlapping with the bounding boxes of all ground truth is within the range of (0.1, 0.5) (Ren et al. 2017). The Iverson bracket ($u \geq 1$) is defined to be 1 if $u \geq 1$ and 0 otherwise. To balance the two loss functions, the value of hyper-parameter λ was first determined to be 1 (Ren et al. 2017) and the rationality of the adopted value was verified with parameter analysis.

VGG16-Net Architecture of the CNNs

To improve computation efficiency, RPN and Fast R-CNN are thus expected when sharing the same architecture of CNN. There are a number of well-known architectures for CNN, including VGG16-net, ZF-Net, GoogleNet, and Microsoft ResNet-152 (Cha et al. 2018). In the present study, the VGG16-net was adopted as it can achieve a good balance between computation efficiency as well as detection accuracy. The VGG16-net is composed of 13 weighted CONV layers,

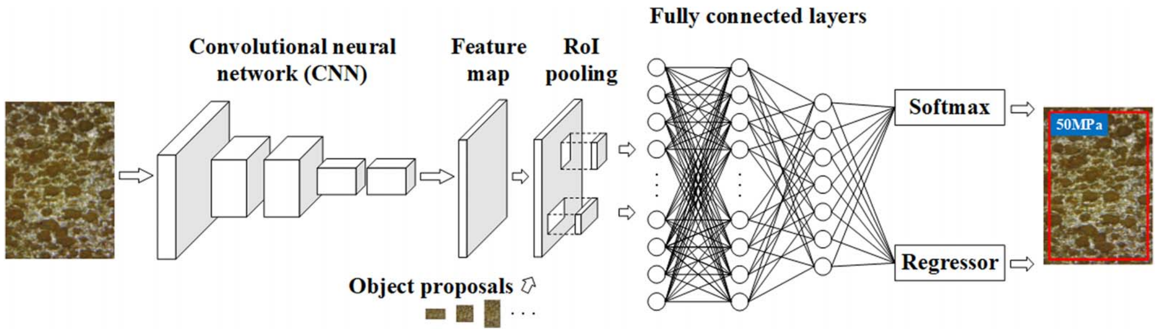

Fig. 5. Overall architecture of the Fast R-CNN.

five MPLs, three weighted FC layers and a soft-max layer. More details about the original VGG16-net can be found in Cha et al. (2018).

To put forward a Faster R-CNN-based approach for detecting different stresses, the original architecture of VGG16-net was modified to coordinate with that of the RPN as well as that of the Fast R-CNN. For the modified RPN illustrated in Fig. 6, a sliding CONV, followed by an FC layer (feature vector) with a depth of 512-dimension, is adopted to substitute the final MPL as well as the three FC layers of the primal VGG16-net, and soft-max and regression layers are adopted to substitute the soft-max layer of the primal VGG16-net. Table 1 lists the detailed information on the VGG16-net-based RPN.

For the modified Fast R-CNN shown in Fig. 7, a RoI pooling layer was used for substituting the final MPL of the primal VGG16-net. To avoid overfitting, dropout layers, whose threshold value was adopted to be 0.5, were inserted between each of the three FC layers of the primal VGG16-net. To guarantee the compatibility with the number of classifications under consideration, the depth of the final FC layer was changed into seven for six stress levels and background. The last soft-max layer was substituted with soft-max and regression layers.

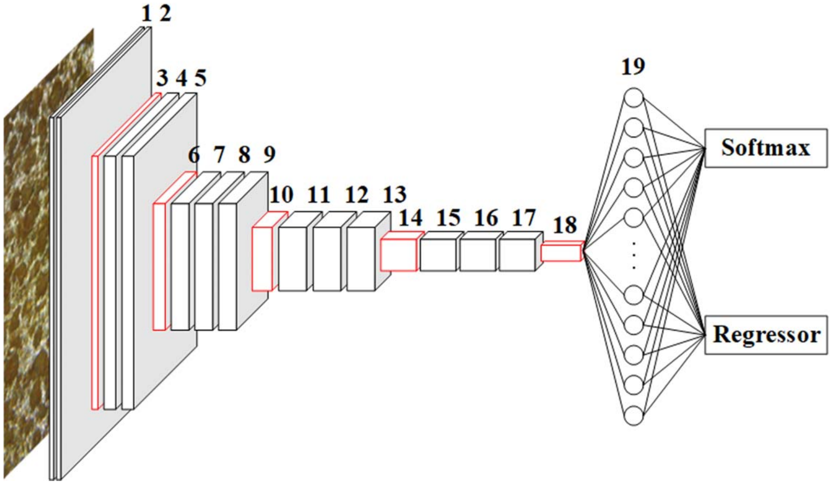

Fig. 6. Modified architectures of VGG16-net for RPN.

Table 1. Detailed information on the VGG16-net-based RPN's layer

Layer number	Type	Depth	Filter size	Stride	Layer number	Type	Depth	Filter size	Stride
1	CR	64	3 × 3	1	11	CR	512	3 × 3	1
2	CR	64	3 × 3	1	12	CR	512	3 × 3	1
3	MP	64	2 × 2	2	13	CR	512	3 × 3	1
4	CR	128	3 × 3	1	14	MP	512	2 × 2	2
5	CR	128	3 × 3	1	15	CR	512	3 × 3	1
6	MP	128	2 × 2	2	16	CR	512	3 × 3	1
7	CR	256	3 × 3	1	17	CR	512	3 × 3	1
8	CR	256	3 × 3	1	18	Sliding CR	512	—	—
9	CR	256	3 × 3	1	19	FC	512	—	—
10	MP	256	2 × 2	2	20	Soft-max & Regressor	—	—	—

Note: CR = CONV + ReLU; and MP = max pooling.

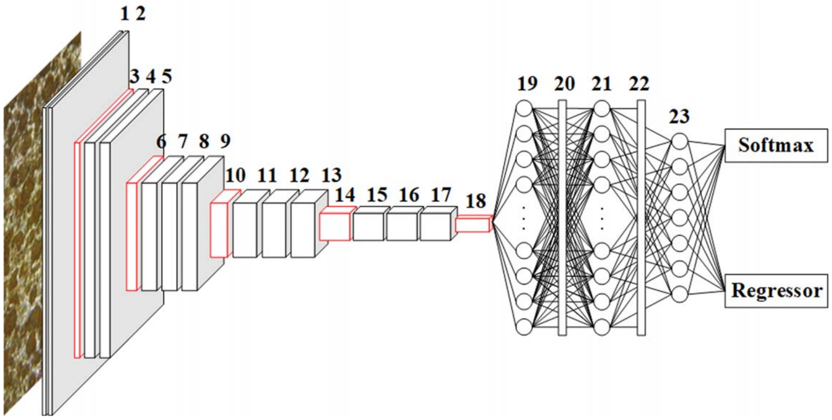

Fig. 7. Modified architectures of VGG16-net for Fast R-CNN.

Table 2 lists the detailed information on the VGG16-net-based Fast R-CNN.

Faster R-CNN Combined from the RPN and Fast R-CNN

Fig. 8 illustrates the overall architecture of the Faster R-CNN. As observed from Fig. 8, the Faster R-CNN is actually a combination of the RPN and Fast R-CNN that share the same nine-layer CNNs for the extraction of image features. Faster R-CNN can be trained efficiently through a four-step alternating training technique (Ren et al. 2017). More details about the Faster R-CNN training process can be found from Ren et al. (2017).

Dataset Preparation and Implementation Details

Dataset Preparation

To detect the stress of structures using deep learning methods, datasets need to be prepared in advance for the purpose of training, validating, and testing the deep learning model. The datasets were created through a three-step process as demonstrated in Fig. 9.

The first step was to prepare steel specimens. Twenty polished steel specimens with dimensions of 150 × 10 × 10 mm were fabricated, as shown on the left of Fig. 9. The type of steel is Q345q with a yield strength of 345 MPa, which is commonly used in civil structures.

The second step was to capture the images of steel specimens under different stresses. As shown in the middle of Fig. 9, a uniaxial compression experiment was conducted for each steel specimen with an electromechanical universal testing machine that has a capacity of 30 kN. The machine can be held on at any specific load within its capacity for a certain time and can give real-time display of the applied force. A loading scheme was programmed, in which the load was applied from 0 to 25 kN with an interval of 5 kN and at each load step the machine was held on for a certain time to capture images. As the cross-sectional area of these specimens is known, the stresses in each specimen can be calculated under the applied forces. At each of the six stress level (namely, from 0 to 250 MPa with an interval of 50 MPa), images of the microstructure

Table 2. Detailed information on the VGG16-net-based Fast R-CNN's layer

Layer number	Type	Depth	Filter size	Stride	Layer number	Type	Depth	Filter size	Stride
1	CR	64	3 × 3	1	13	CR	512	3 × 3	1
2	CR	64	3 × 3	1	14	MP	512	2 × 2	2
3	MP	64	2 × 2	2	15	CR	512	3 × 3	1
4	CR	128	3 × 3	1	16	CR	512	3 × 3	1
5	CR	128	3 × 3	1	17	CR	512	3 × 3	1
6	MP	128	2 × 2	2	18	RoI pooling	512	—	—
7	CR	256	3 × 3	1	19	FR	4,096	—	—
8	CR	256	3 × 3	1	20	Dropout	—	—	—
9	CR	256	3 × 3	1	21	FR	4,096	—	—
10	MP	256	2 × 2	2	22	Dropout	—	—	—
11	CR	512	3 × 3	1	23	FR	7	—	—
12	CR	512	3 × 3	1	24	Soft-max & Regressor	—	—	—

Note: CR = CONV + ReLU; MP = max pooling; and FR = FC + ReLU.

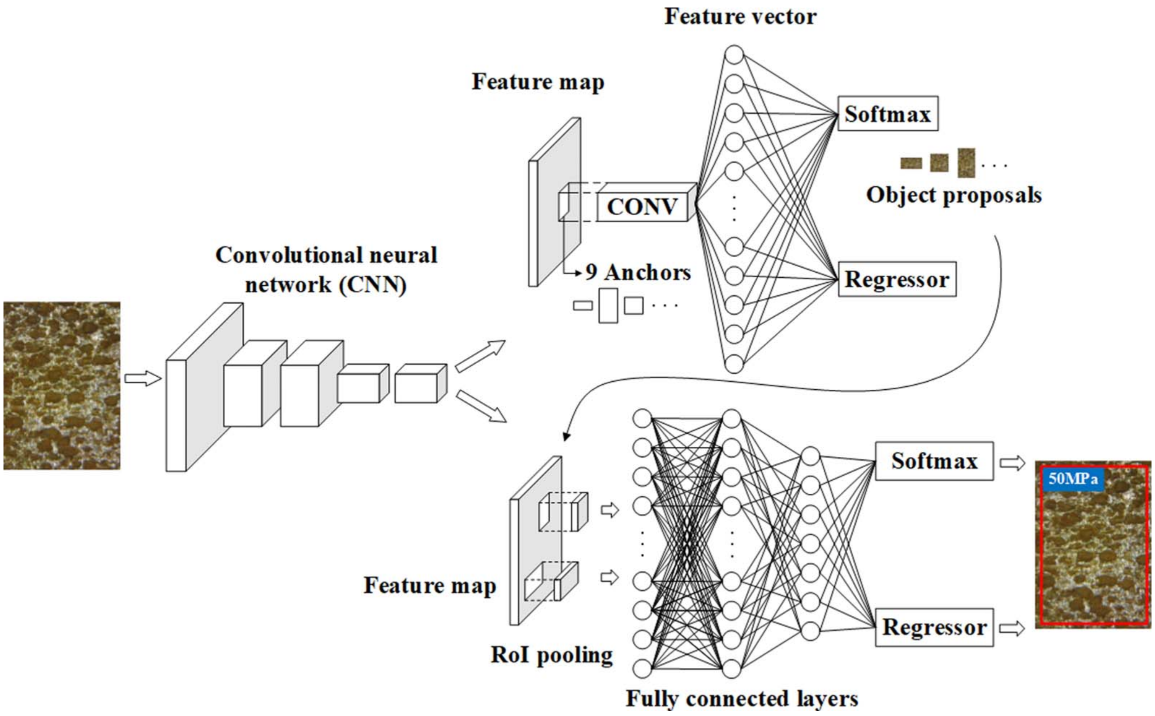

Fig. 8. Overall architecture of the Faster R-CNN.

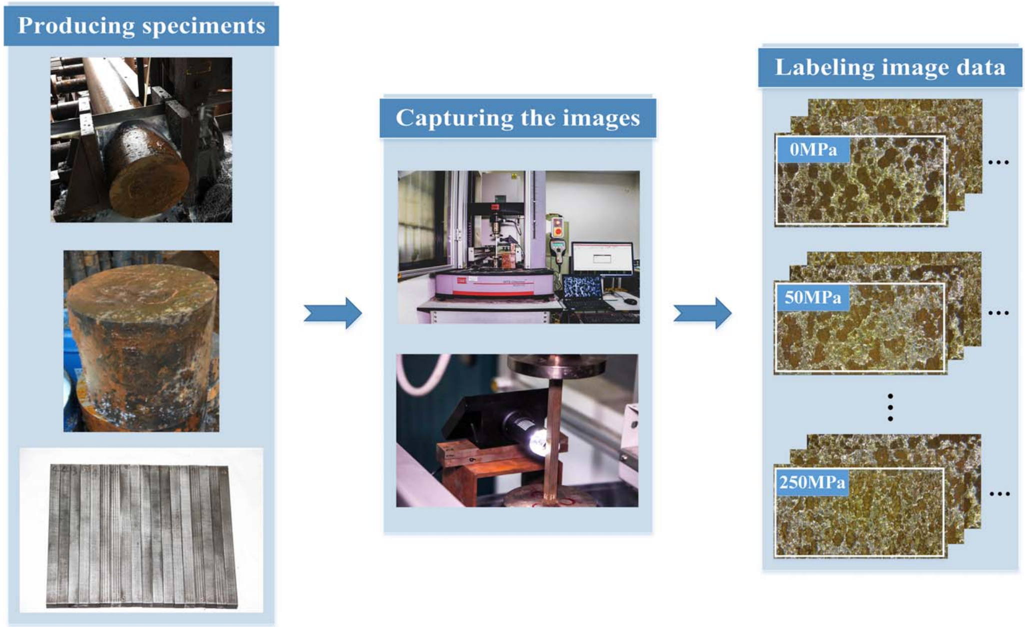

Fig. 9. Process to prepare dataset.

features of steel specimens were captured with 2-megapixel (1,920 × 1,080) portable digital microscopes. As only a 2.3- × 1.3-mm region can be photographed by the portable digital microscope, the captured images have a resolution of about 21,000 dots per in. (dpi). To improve the robustness of detection, different lighting conditions were adopted and images were taken by different photographers with different portable digital microscopes with the same specification. Under each stress level, 221 original images were obtained from all specimens. After dataset augmentation through horizontal mirror, vertical mirror, and 180-degree rotation, the total number of images increases to 5,304 (namely, 221 × 6 × 4). It should be noted that based on the Euler's formula the critical buckling stress of the specimens was calculated to be 752 MPa, which means that buckling will not occur during the compression experiment.

The final step was to establish datasets using images obtained from the previous step. Since a supervised deep learning model was used, labeled information of images was required for training. Among the 5,304 images obtained from the previous step, 60 images (10 for each stress) were randomly picked for examining the robustness of the proposed approach while the remaining 5,244 images were labeled with stresses under which they were taken, as illustrated on the right of Fig. 9. Specifically, if an image was taken for a specimen under a stress of 50 MPa, then it was labeled as 50 MPa. To generate a testing dataset, 20% of images were randomly chosen from all these labeled images and, in the meantime, the remaining 80% of labeled images were utilized for the generation of the training and validation datasets.

Implementation Details

The open-source Faster R-CNN package was used to implement all experiments with a computer [Central Processing Unit (CPU): Intel i7-8700k @3.20 GHz, 16 GB DDR4 memory; and Graphic Processing Units (GPU): 11 GB memory ZOTAC X-GMING GeForce RTX 2080Ti] through the GPU mode based on a Caffe framework. The software configuration is: Ubuntu 16.04.6, python 2.7, CUDA10.0, and CUDNN 7.5. For the Faster R-CNN, to achieve a good balance between calculation efficiency and accuracy in the present study, images utilized to train and test the RPN and Fast R-CNN are rescaled such that the maximum sizes of its shorter and longer sides are less than 600 and 1,000 pixels, respectively, and the rationality of the adopted image size was verified with parameter analysis. The initial weights of the CNN layers as well as the FC layers are adopted from zero-mean Gaussian distribution with standard deviations of 0.001 and 0.01, respectively. The values of the learning rate, the momentum, the weight decay as well as the mini-batch size are adopted to be 0.001, 0.9, 0.0005, and 128, respectively, for training both the RPN and the Fast R-CNN. Nine anchor scales are determined through different combinations of three scales $\{128^2, 256^2, 512^2\}$ and three aspect ratios {1:1, 1:2, 2:1}. Anchors that cross image boundaries were discarded during the training process as cross-boundary anchors will result in no convergence. In addition, a value of 0.7 was adopted for the non-maximum suppression to decrease the number of OPs highly overlapping with each other. More details on parameter initialization of the Faster R-CNN can be found from Ren et al. (2017).

Average precision (AP) is usually employed for performance evaluation of an object detector (Girshick 2015; Girshick et al. 2014), and is obtained by calculating the area under the precision–recall curve (Everingham et al. 2010). For a specific class, precision is theoretically computed to be the percentage of correct detections to the total detections returned by the algorithm while recall is theoretically computed to be the percentage of correct detections to the total ground truth instances under consideration. The term *mean AP* (mAP), as the name implies, is the mean of computed APs for all classes under consideration. Readers can find more information on the precision–recall curve and AP from Everingham et al. (2010).

Experiments

Training, Validation, and Testing Results

The Faster R-CNN was first trained with initial parameters following the four-step training algorithm introduced previously and its performance was evaluated with the testing dataset. In the GPU mode, the time required to train the network for 210,000 iterations and to evaluate a 1,920 × 1,080-pixel image is about 12.56 h and 0.08 s, respectively. The variation of training loss with the number of iterations is shown in Fig. 10, from which it can be seen that the training loss decreases with the increase of iterations and tends to be stable when iterations reach around 200,000. With the model trained for 200,000 iterations, the precision–recall curve for the testing dataset was obtained, as plotted in Fig. 11, based on which the APs and mAP were calculated. From Fig. 11, it can be seen that the APs for the detection of the six stresses (namely, from 0 to 250 MPa with an interval of 50 MPa) are 0.812, 0.872, 0.758, 0.898, 0.902, and 0.907, respectively, and the corresponding mAP is 0.858. Similarly, the APs and mAPs calculated from models trained for different iterations were plotted against the number of iterations in Fig. 12. As expected, the APs and mAPs both increase as the number of iterations increases and also tend to stabilize after 200,000 iterations.

Parameter Optimization

As Ren et al. (2017) acknowledged that the initial parameters of Faster R-CNN selected in their study may not be appropriate for a specific dataset, parameter optimization for Faster R-CNN was conducted to achieve higher accuracy of stress detection. Three important parameters, including the anchor scale, the mini-batch size as well as the learning rate, were found to have a significant impact on the performance of Faster R-CNN (Ren et al. 2017). Therefore, the effects of these parameters on stress detection accuracy were investigated in this study. A total of 45 combinations of three groups of anchor scales, five mini-batch sizes, and three learning rates were considered. The APs and mAPs for the test dataset were obtained and summarized, as shown in Fig. 13 and Table 3. It should be noted that Ren et al. (2017) labeled a RoI as the calculated class if the probability calculated by the soft-max layer for the RoI was 0.6 or higher.

From Fig. 13 and Table 3, it can be seen that the anchor scale, mini-batch size, and learning rate have a coupling effect on the APs and mAPs. The highest APs for stresses of 0, 50, 100, 150, 200, and 250 MPa are 88.20% for Case 26, 88.15% for Case 41, 97.10% for Case 42, 94.88% for Case 42, 90.58% for Case 45, and 96.86% for Case 38, respectively. To guarantee a rational balance among APs for different stresses, Case 41 was chosen, which achieved the second highest mAP of 89.67%, only slightly lower than the highest mAP of 89.83%. In Case 41, the APs for the six stresses, namely, from 0 to 250 MPa with an interval of 50 MPa, are 87.46%, 88.15%, 89.53%, 91.94%, 90.06%, and 90.91%, respectively. The anchor sizes as well as anchor ratios in Case 41 are $\{256^2, 512^2, 1{,}024^2\}$ and {1:1, 1:2, 2:1}, respectively. The mini-batch size and learning rate in Case 41 are 128 and 0.0005, respectively. To better illustrate the learning process in Case 41, the variation of training loss with the number of iterations is shown in Fig. 14, and with the model trained for 200,000 iterations, the precision–recall curve for the testing dataset is shown in Fig. 15. Fig. 14 demonstrates that the training of the Faster R-CNN converges well under Case 41 and Fig. 15 indicates that the well-trained model has a good performance in testing dataset. It should be noted that the mAP of 89.67% obtained in the present study is an acceptable level of accuracy, which can be comparable with the crack detection accuracy of 89.3% and 90.0% obtained in the study by Dung and Anh (2019) and Gopalakrishnan et al. (2017), respectively.

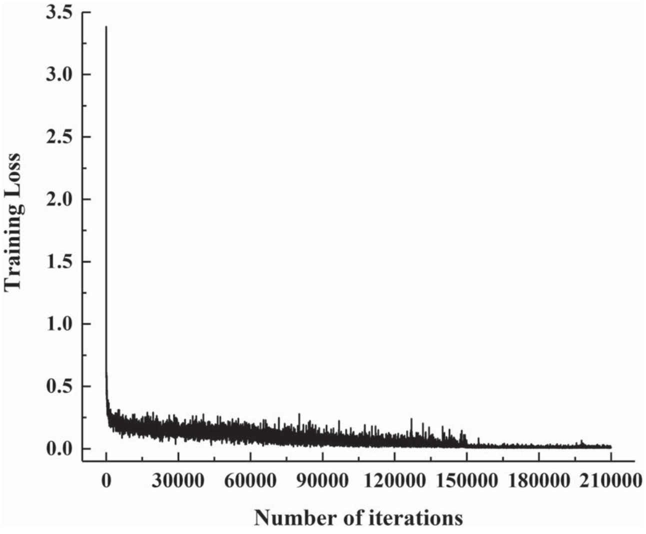

Fig. 10. Variation of training loss with number of iterations under initial combination of parameters.

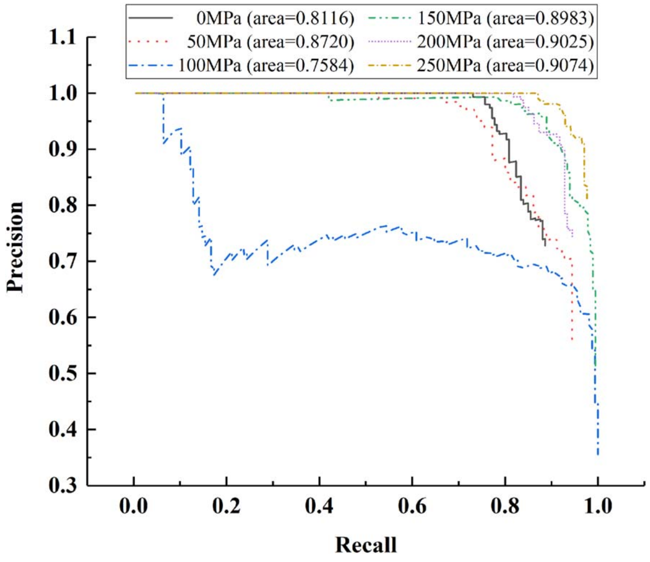

Fig. 11. Precision–recall curve for the testing dataset with the model trained for 200,000 iterations under initial combination of parameters.

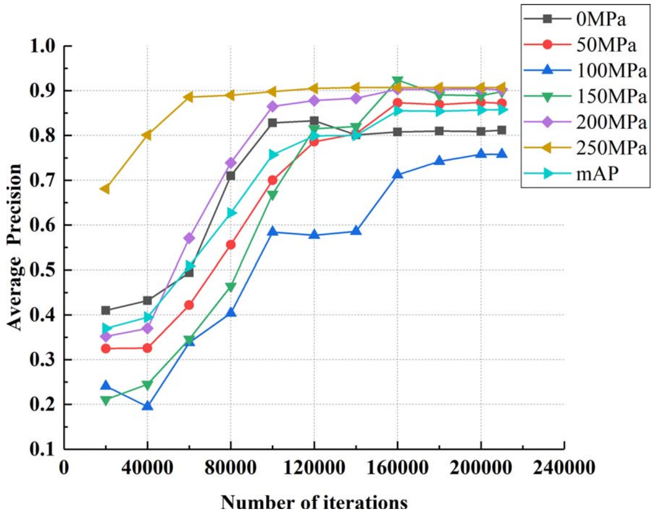

Fig. 12. Variation of APs and mAPs with number of iterations.

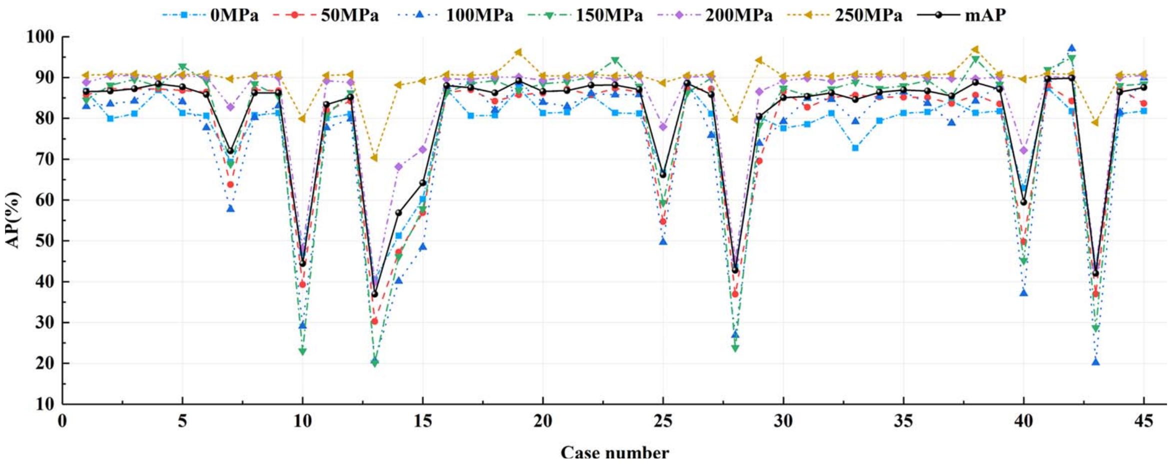

Fig. 13. APs for test dataset of 45 cases.

Table 3. Parameters under consideration and corresponding results for test dataset

				APs (%) for different stresses (MPa)						
Cases	Anchor scales/Aspect ratio	Mini-batch sizes	Learning rate	0	50	100	150	200	250	mAP (%)
1	$\{64^2, 128^2, 256^2\}$ {1:1, 1:2, 2:1}	16	0.0001	86.70	85.73	82.93	84.50	88.81	90.57	86.54
2			0.0005	79.93	87.26	83.56	88.11	90.43	90.79	86.68
3			0.001	81.16	87.19	84.26	89.54	90.53	90.85	87.26
4		32	0.0001	86.92	87.19	88.65	87.82	89.83	90.15	88.43
5			0.0005	81.32	86.86	84.09	92.81	90.44	90.74	87.71
6			0.001	80.64	86.42	77.72	89.14	90.14	90.85	85.82
7		64	0.0001	69.30	63.76	57.71	68.81	82.77	89.65	72.00
8			0.0005	80.78	86.98	80.22	88.46	90.38	90.46	86.21
9			0.001	81.32	86.73	83.06	85.39	90.06	90.73	86.21
10		128	0.0001	47.09	39.28	29.14	23.02	48.15	79.92	44.43
11			0.0005	80.09	82.00	77.74	80.71	89.21	90.50	83.38
12			0.001	81.06	84.20	79.85	86.25	88.82	90.73	85.15
13		256	0.0001	40.52	30.26	20.63	20.13	39.81	70.35	36.95
14			0.0005	51.25	47.22	40.15	46.16	68.15	88.16	56.85
15			0.001	60.23	56.92	48.46	57.89	72.36	89.20	64.18
16	$\{128^2, 256^2, 512^2\}$ {1:1, 1:2, 2:1}	16	0.0001	87.03	86.40	87.80	86.41	89.67	90.73	88.01
17			0.0005	80.65	86.98	88.94	88.53	89.58	90.52	87.53
18			0.001	80.73	84.22	82.03	89.29	90.14	90.85	86.21
19		32	0.0001	87.42	85.76	88.77	86.87	90.10	96.17	89.18
20			0.0005	81.27	86.19	83.93	88.46	89.32	90.46	86.60
21			0.001	81.48	87.14	82.93	88.96	90.06	90.38	86.82
22		64	0.0001	85.70	85.80	86.08	90.15	90.24	90.73	88.12
23			0.0005	81.37	87.32	85.78	94.41	89.63	90.42	88.16
24			0.001	81.17	86.48	85.85	87.87	90.44	90.55	87.06
25		128	0.0001	66.72	54.70	49.64	59.35	77.91	88.69	66.17
26			0.0005	88.20	87.12	89.07	86.30	90.14	90.50	88.56
27			0.001	81.16	87.20	75.84	89.83	90.25	90.74	85.84
28		256	0.0001	43.65	36.95	26.90	23.83	45.40	79.80	42.80
29			0.0005	80.30	69.58	73.92	78.34	86.56	94.22	80.49
30			0.001	77.60	86.60	79.27	87.42	89.20	90.30	85.06
31	$\{256^2, 512^2, 1{,}024^2\}$ {1:1, 1:2, 2:1}	16	0.0001	78.55	82.71	84.90	85.47	89.83	90.68	85.36
32			0.0005	81.22	85.15	84.65	87.18	89.12	90.27	86.26
33			0.001	72.73	85.70	79.22	88.86	90.24	90.79	84.59
34		32	0.0001	79.41	85.23	85.35	87.29	90.14	90.85	86.38
35			0.0005	81.32	85.19	86.49	87.96	90.34	90.41	86.95
36			0.001	81.53	85.09	83.70	89.31	90.01	90.62	86.71
37		64	0.0001	84.20	83.65	78.87	85.33	89.75	90.91	85.45
38			0.0005	81.37	85.68	84.27	94.65	89.71	96.86	88.76
39			0.001	81.76	83.56	88.11	88.30	90.10	90.79	87.10
40		128	0.0001	62.90	49.79	37.09	45.20	72.16	89.60	59.46
41			0.0005	87.46	88.15	89.53	91.94	90.06	90.91	89.67
42			0.001	81.76	84.26	97.10	94.88	90.15	90.85	89.83
43		256	0.0001	43.86	36.98	20.18	28.71	43.52	78.99	42.04
44			0.0005	81.12	87.19	81.52	88.03	90.06	90.63	86.42
45			0.001	81.82	83.63	89.92	88.42	90.58	90.85	87.54

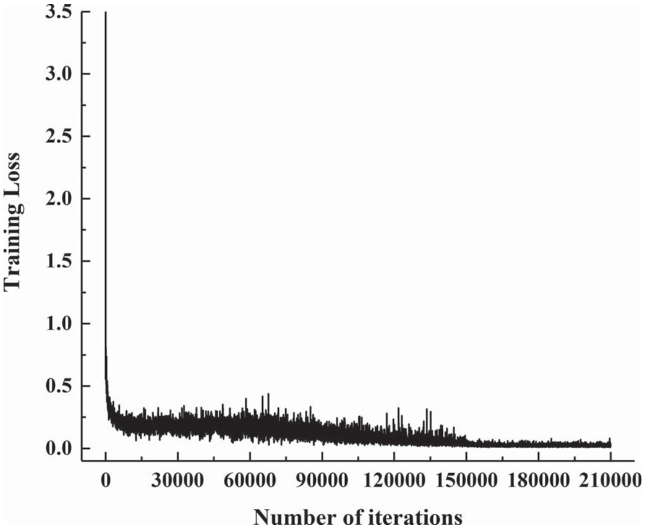

Fig. 14. Variation of training loss with number of iterations under optimal combination of parameters.

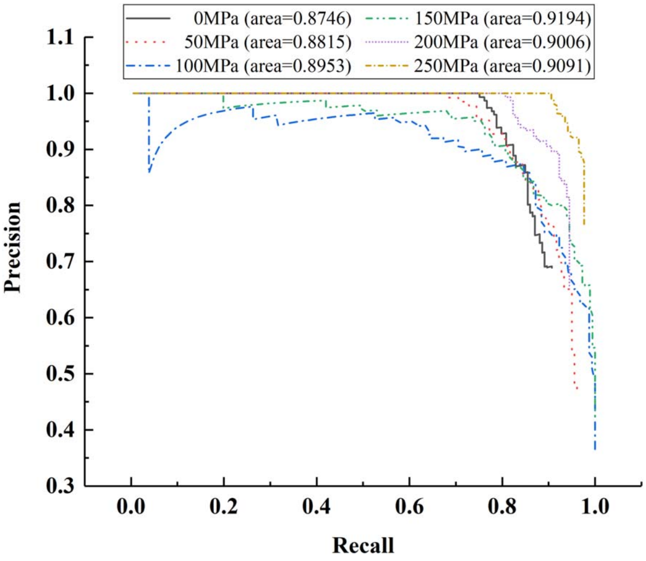

Fig. 15. Precision–recall curve for the testing dataset with the model trained for 200,000 iterations under optimal combination of parameters.

Testing New Images

To prove the robustness of this proposed method, the model trained in Case 41 was adopted to detect the stress of 60 unlabeled images (ten for each stress level) reserved in the dataset preparation stage. The testing results for new images were summarized in Table 4, from which it can be observed that the number of correct detections for the six stresses, namely, from 0 to 250 MPa with an interval of 50 MPa, are 8, 8, 10, 10, 9, and 8, respectively. The corresponding APs for the stresses of 0, 50, 100, 150, 200, and 250 MPa are 80%, 80%, 100%, 100%, 90%, and 80%, respectively, and the mAP is 88.33%. This indicates that the precision rates achieved for new images are comparable with that produced in Case 41. Fig. 16 illustrates the details on the stress detection of 200 MPa, from which it can be observed that one picture taken at the stress of 200 MPa was wrongly detected as 150 MPa with an accuracy of 0.99. In other words, though the trained model detected the stress of this picture to be 150 MPa with an accuracy of 0.99 based on the picture features, this detection is wrong. To overcome this situation and achieve reliable detection results in practice, it is suggested that if more pictures are captured around the point of interest and detected with the trained model, and the actual stress of the point is most likely to be the most frequently detected stress. It should be noted that these 60 unlabeled images were taken under different conditions as mentioned in the section on dataset preparation, indicating that the precision rate is rarely affected by these factors.

Table 4. Summary of testing results for new images.

Stresses (MPa)	Number of images	Number of correct detection	Aps (%)	mAP (%)
0	10	8	80	88.33
50	10	8	80	
100	10	10	100	
150	10	10	100	
200	10	9	90	
250	10	8	80	

Discussion

It is well known that the traditional sensor-based method is not able to detect the absolute stress of structure components due to some unknown stresses having already been produced before the installment of sensors. Therefore, only the stress increment can be detected with traditional sensors. The proposed deep learning method provides a new idea to detect the absolute stress, which establishes the relationship between the microstructure features of structures and the absolute stress experienced by the structure components through the model training based on the prepared dataset. The well-trained model can then be used to detect the absolute stress of other structure components made of the same material.

The results presented in the previous section have proven that the Faster R-CNN-based deep learning method has the capacity to learn and detect microstructure feature differences of steel specimens under six different stress levels. In reality, the stress to be detected may not be exactly the same as one of those used to train the model. Therefore, the system will likely identify the training stress value that is closest to the actual stress as the true stress value. To narrow the gap between the actual stress and the closest training stress value, one effective way is to augment the number of samples to train the model and therefore reduce the identification error. In addition, images should be taken at an angle that is perpendicular to the direction of the normal stress to be detected for good quality. Further, the surface of the structure component to be detected needs to be cleaned in case of rust and corrosion. Covers, if any, need to be removed. It should be noted that the microstructure features of different materials are not the same and the actual stress experienced by the structure components are also different from the stresses considered in the study. Therefore, new models need to be trained for new materials. The procedures to implement such a deep learning framework for stress detection are summarized as follows:

1. Prepare enough specimens under different stress levels and make sure that the specimens can cover the range of stresses to be detected and can ensure the target accuracy.
2. Obtain good-quality images of the specimens and label these images.
3. Choose an appropriate deep learning network and train it with the images obtained from the previous step; validate the efficiency of the trained model with new images not used in training.

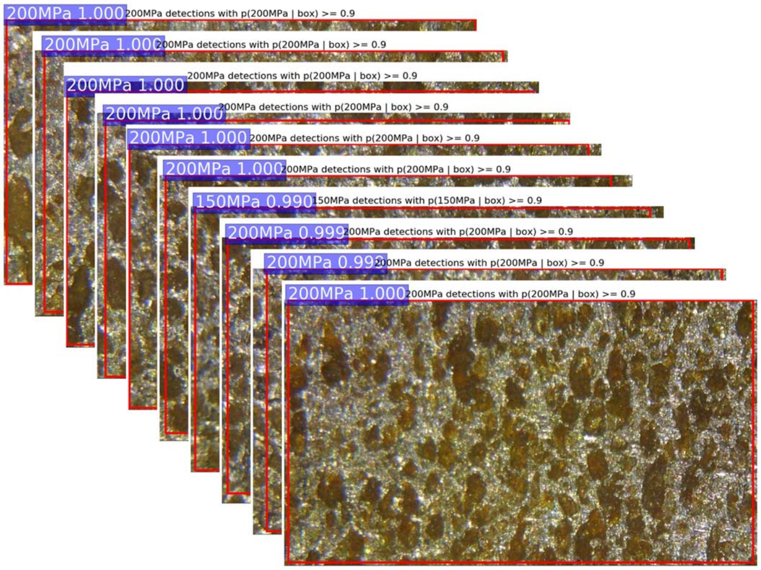
200MPa 1.000
200MPa detections with p(200MPa | box) >= 0.9
200MPa 1.000
200MPa detections with p(200MPa | box) >= 0.9
200MPa 1.000
200MPa detections with p(200MPa | box) >= 0.9
200MPa 1.000
200MPa detections with p(200MPa | box) >= 0.9
200MPa 1.000
200MPa detections with p(200MPa | box) >= 0.9
200MPa 1.000
200MPa detections with p(200MPa | box) >= 0.9
150MPa 0.990
150MPa detections with p(150MPa | box) >= 0.9
200MPa 0.999
200MPa detections with p(200MPa | box) >= 0.9
200MPa 0.99
200MPa detections with p(200MPa | box) >= 0.9
200MPa 1.000
200MPa detections with p(200MPa | box) >= 0.9

Fig. 16. Illustration of the details on the stress detection of 200 MPa.

4. Obtain the images of the structure component to be detected and use the trained model to detect the stress.

It should be noted that improving the quality of images and increasing the number of images utilized to train and validate the model will enhance the performance of the model.

Conclusions

Traditional methods for stress detection suffer from drawbacks such as only being capable of obtaining stress increments instead of absolute stress, and causing structural damage. A deep learning framework for total stress detection of steel components is proposed according to the Faster R-CNN, and the robustness of this proposed method is demonstrated with an example. Uniaxial compression experiments of steel specimens were conducted and images of microstructure features of the specimens under six different stress levels (namely, from 0 to 250 MPa with an interval of 50 MPa) were captured under different conditions with portable digital microscopes. The captured images have a resolution of about 21,000 dpi. Among the 5,304 images used in this study, 5,244 images were used for training (40%), validating (40%), and testing (20%) of the Faster R-CNN, and the remaining 60 images (10 for each stress level) were used for examining the robustness of the model as new images.

The effects of three important parameters, including the anchor scale, batch size, and learning rate, on the accuracy of stress detection were investigated and an optimal combination of these parameters for training the Faster R-CNN was obtained. Under an optimal combination of these parameters, the maximum APs achieved by the trained Faster R-CNN for stress detection of 0, 50, 100, 150, 200, and 250 MPa are 87.46%, 88.15%, 89.53%, 91.94%, 90.06%, and 90.91%, respectively, and the corresponding mAP reaches 89.67%. The proposed method provides a new idea to obtain the absolute stress of structure components that is difficult to obtain with the conventional sensor-based method. The results from the study also indicate that the Faster R-CNN-based deep learning method is capable of detecting the difference not only in macroscopic objects but also in microstructure features.

It should be noted that the microstructure features of different materials are usually different even under the same stress level. This means that specific models will be needed for different materials. In future studies, attempts will be made on structural components made of other materials, such as concrete, that are commonly used in civil structures to examine the applicability and accuracy of this presented approach. In addition, it should also be noted that from the perspective of civil engineering it is desirable to detect an arbitrary stress rather than detect it to be the closest labeled stress. To achieve this goal, using the Euclidean loss function to replace the soft-max function at the last layer of the Faster R-CNN may be a possible solution, but more efforts are needed in future research work.

Data Availability Statement

Some data, models, or code generated or used during the study are available from the corresponding author by request, including the datasets and the modified Faster R-CNN.

Acknowledgments

The authors would like to acknowledge the financial support provided by the National Natural Science Foundation of China (Grant Nos. 51808209, 51778222, and 51478176), Natural Science Foundation of Hunan Province (2019JJ50065), and the Fundamental Research Funds for the Central Universities (Grant No. 531118010081).

References

Abdeljaber, O., O. Avci, S. Kiranyaz, M. Gabbouj, and D. J. Inman. 2017. "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks." *J. Sound Vib.* 388: 154–170. https://doi.org/10.1016/j.jsv.2016.10.043.

Brar, G. S., and C. S. Singh. 2014. "FEA of residual stress in cruciform welded joint of hollow sectional tubes." *J. Constr. Steel Res.* 102: 44–58. https://doi.org/10.1016/j.jcsr.2014.06.011.

Cha, Y. J., W. Choi, and O. Büyüköztürk. 2017. "Deep learning-based crack damage detection using convolutional neural networks." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (5): 361–378. https://doi.org/10.1111/mice.12263.

Cha, Y.-J., W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk. 2018. “Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types.” *Comput.-Aided Civ. Infrastruct. Eng.* 33 (9): 731–747. https://doi.org/10.1111/mice.12334.

Dick, K., L. Russell, Y. S. Dosso, F. Kwamena, and J. R. Green. 2019. “Deep learning for critical infrastructure resilience.” *J. Infrastruct. Syst.* 25 (2): 05019003. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000477.

Dung, C. V., and L. D. Anh. 2019. “Autonomous concrete crack detection using deep fully convolutional neural network.” *Autom. Constr.* 99: 52–58. https://doi.org/10.1016/j.autcon.2018.11.028.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. “The pascal visual object classes (VOC) challenge.” *Int. J. Comput. Vision* 88 (2): 303–338. https://doi.org/10.1007/s11263-009-0275-4.

Fathalla, E., Y. Tanaka, and K. Maekawa. 2018. “Remaining fatigue life assessment of in-service road bridge decks based upon artificial neural networks.” *Eng. Struct.* 171: 602–616. https://doi.org/10.1016/j.engstruct.2018.05.122.

Franco Grijalba, F. A., and L. R. Padovese. 2018. “Non-destructive scanning for applied stress by the continuous magnetic Barkhausen noise method.” *J. Magn. Magn. Mater.* 446: 231–238. https://doi.org/10.1016/j.jmmm.2017.09.036.

Girshick, R. 2015. “Fast R-CNN.” In *IEEE Int. Conf. on Computer Vision*, 1440–1448. Piscataway, NJ: IEEE.

Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. “Rich feature hierarchies for accurate object detection and semantic segmentation.” In *IEEE Conf. on Computer Vision & Pattern Recognition*, 580–587. Piscataway, NJ: IEEE.

Gopalakrishnan, K., S. K. Khaitan, A. Choudhary, and A. Agrawal. 2017. “Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection.” *Constr. Build. Mater.* 157: 322–330. https://doi.org/10.1016/j.conbuildmat.2017.09.110.

Gulgec, N. S., M. Takáč, and S. N. Pakzad. 2017. *Structural damage detection using convolutional neural networks*, 331–337. Cham, Switzerland: Springer International Publishing.

Han, F., M. Prezzi, R. Salgado, and M. Zaheer. 2017. “Axial resistance of closed-ended steel-pipe piles driven in multilayered soil.” *J. Geotech. Geoenviron. Eng.* 143 (3): 04016102. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001589.

Kang, D., and Y.-J. Cha. 2018. “Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging.” *Comput.-Aided Civ. Infrastruct. Eng.* 33 (10): 885–902. https://doi.org/10.1111/mice.12375.

Kurashkin, K., V. Mishakin, and A. Rudenko. 2019. “Ultrasonic evaluation of residual stresses in welded joints of hydroelectric unit rotor frame.” *Mater. Today:. Proc.* 11: 163–168. https://doi.org/10.1016/j.matpr.2018.12.125.

Lee, S., J. Ha, M. Zokhirova, H. Moon, and J. Lee. 2018. “Background information of deep learning for structural engineering.” *Arch. Comput. Methods Eng.* 25 (1): 121–129. https://doi.org/10.1007/s11831-017-9237-0.

Li, B., K. C. P. Wang, A. Zhang, E. Yang, and G. Wang. 2020. “Automatic classification of pavement crack using deep convolutional neural network.” *Int. J. Pavement Eng.* 21 (4): 457–463. https://doi.org/10.1080/10298436.2018.1485917.

Li, H., C.-X. Mao, and J.-P. Ou. 2005. “Strain self-sensing property and strain rate dependent constitutive model of austenitic shape memory alloy: Experiment and theory.” *J. Mater. Civ. Eng.* 17 (6): 676–685. https://doi.org/10.1061/(ASCE)0899-1561(2005)17:6(676).

Nair, V., and G. E. Hinton. 2010. “Rectified linear units improve restricted boltzmann machines.” In *Proc., 27th Int. Conf. on Machine Learning*, 807–814. Madison, WI: Omnipress.

Ni, F. T., J. Zhang, and Z. Q. Chen. 2019. “Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning.” *Comput.-Aided Civ. Infrastruct. Eng.* 34 (5): 367–384. https://doi.org/10.1111/mice.12421.

Ren, S., K. He, R. Girshick, and J. Sun. 2017. “Faster R-CNN: Towards real-time object detection with region proposal networks.” *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (6): 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.

Sánchez-Beitia, S., and P. Roca. 2014. “Hole drilling technique for on-site stress measurements of masonry piers: Analysis of the Santa Maria del Mar Cathedral in Barcelona, Spain.” *J. Archit. Eng.* 20 (3): 05014002. https://doi.org/10.1061/(ASCE)AE.1943-5568.0000149.

Stupakov, A., O. Perevertov, and M. Landa. 2017. “Dynamic behaviour of magneto-acoustic emission in a grain-oriented steel.” *J. Magn. Magn. Mater.* 426: 685–690. https://doi.org/10.1016/j.jmmm.2016.10.142.

Turan, M. E., F. Aydin, Y. Sun, and M. Cetin. 2019. “Residual stress measurement by strain gauge and x-ray diffraction method in different shaped rails.” *Eng. Fail. Anal.* 96: 525–529. https://doi.org/10.1016/j.engfailanal.2018.10.016.

Venkatachalapathi, N., S. M. Jameel basha, G. Janardhan Raju, and P. Raghavulu. 2018. “Characterization of fatigued steel states with metal magnetic memory method.” *Mater. Today:. Proc.* 5 (2): 8645–8654. https://doi.org/10.1016/j.matpr.2018.04.002.

Wang, N., X. Zhao, L. Wang, and Z. Zou. 2019. “Novel system for rapid investigation and damage detection in cultural heritage conservation based on deep learning.” *J. Infrastruct. Syst.* 25 (3): 04019020. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000499.

Weinstein, J. C., M. Sanayei, and B. R. Brenner. 2018. “Bridge damage identification using artificial neural networks.” *J. Bridge Eng.* 23 (11): 04018084. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001302.

Weng, C. C., and J. J. Chen. 1993. “Study on residual stress relief of welded structural steel joints.” *J. Mater. Civ. Eng.* 5 (2): 265–279. https://doi.org/10.1061/(ASCE)0899-1561(1993)5:2(265).

Yan, W., L. Deng, F. Zhang, T. Li, and S. Li. 2019. “Probabilistic machine learning approach to bridge fatigue failure analysis due to vehicular overloading.” *Eng. Struct.* 193: 91–99. https://doi.org/10.1016/j.engstruct.2019.05.028.

Yang, X., H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang. 2018. “Automatic pixel-level crack detection and measurement using fully convolutional network.” *Comput.-Aided Civ. Infrastruct. Eng.* 33 (12): 1090–1109. https://doi.org/10.1111/mice.12412.

Zuccarello, B. 1999. “Optimal calculation steps for the evaluation of residual stress by the incremental hole-drilling method.” *Exp. Mech.* 39 (2): 117–124. https://doi.org/10.1007/BF02331114.