

Research Article

Complex Texture Contour Feature Extraction of Cracks in Timber Structures of Ancient Architecture Based on YOLO Algorithm

Jian Ma , Weidong Yan, Guoqi Liu, Shiyu Xing, Siqi Niu , and Tong Wei

School of Civil Engineering, Shenyang Jianzhu University, Shenyang 110168, China

Correspondence should be addressed to Siqi Niu; nsq0920@126.com

Received 29 April 2022; Revised 5 July 2022; Accepted 7 July 2022; Published 23 August 2022

Academic Editor: Piotr Smarzewski

Copyright © 2022 Jian Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep learning has achieved good results in the crack detection of roads and bridges. However, the timber structures of ancient architecture have strong orthotropic anisotropy and complex microscopic structures, and the law of cracks development is extremely complex. The image data has a large proportion of pixels, which is obviously different from the background gray value, and there is timber grain noise, thus the existing methods cannot accurately extract the complex texture contour feature of cracks. In previous studies, we have verified that YOLO v5s is effective in crack detection in timber structures of ancient architecture. However, there are many different versions of YOLO series models. In order to find a better algorithm, this paper mainly adopts three models including YOLO v3, YOLO v4s-mish, and YOLO v5s to detect cracks in the timber structures of ancient architecture, and compares and analyzes the advantages and disadvantages of the three models. In the comparing process, we mainly have discussed the index performance of the three models in terms of training time, loss function, recall rate, and mAP value. We have summarized and analyzed the advantages and disadvantages of the three models in cracks detection of the timber structures of ancient architecture, and concluded the comparing results of the three models in cracks detection based on experiments. We published the first picture data set of cracks in timber structures of ancient architecture, and applied YOLO model in the intelligent identification field of cracks in timber structures of ancient architecture for the first time, which opened up a new idea for the intelligent operation and maintenance of the timber structures of ancient architecture.

1. Introduction

There are 5,058 key national preservation units of cultural relics in China. 2,162 of them are ancient architecture and historical memorial architecture, which account for 42.74% (see Table 1 for details). Type one means revolutionary sites and revolutionary memorial architecture, type two means important historical sites and representative architecture in neoteric and modern times, type three means cave temples, stone carvings, and others, type four means ancient architecture and historical monuments, type five means ancient sites, and type six means ancient burials. At the same time, timber structures account for a large proportion of existing ancient architecture in our country. Taking Beijing as an example, there are more than 1,700 cultural relics at the national, municipal, and district levels, among which there is more than 1,200 ancient timber architecture, accounting for

about 71% [1]. The timber structures of ancient Chinese architecture have been constructed for a long time and have extremely high historical and cultural value. During its long service cycle, various forms of damage have appeared due to the influence of the natural environment, climate temperature, and human factors. Through our research and investigation, we have found that cracks are the most common form of damage due to the characteristics of biological materials in ancient timber architecture. Almost all structural components such as columns, beams, purlins, and fang will crack to a certain extent [2] (see Figure 1 for details).

The timber structures of ancient Chinese architecture adopt mortise and tenon joints, and the cracks are characterized by large numbers, complicated causes, and great harm. The strong orthotropic anisotropy and complex microstructure of timber lead to extremely complicated development laws of cracks. General cracks will not affect

TABLE 1: Key national preservation units of cultural relics.

| Date | Batch number | Type one | Type two | Type three | Type four | Type five | Type six | Total |
|------------|--------------|----------|----------|------------|-----------|-----------|----------|-------|
| 1961.03.04 | One | 33 | - | 25 | 77 | 26 | 19 | 180 |
| 1982.02.23 | Two | 10 | - | 7 | 28 | 10 | 7 | 62 |
| 1988.01.13 | Three | 41 | - | 28 | 111 | 49 | 29 | 258 |
| 1996.11.20 | Four | - | 50 | 12 | 110 | 56 | 22 | 250 |
| 2001.06.25 | Five | - | 40 | 36 | 248 | 144 | 50 | 518 |
| 2006.05.25 | Six | - | 206 | 64 | 513 | 220 | 77 | 1080 |
| 2013.05.03 | Seven | - | 329 | 117 | 795 | 516 | 186 | 1943 |
| 2019.10.16 | Eight | - | 234 | 51 | 280 | 167 | 30 | 762 |
| Sum | | 84 | 859 | 340 | 2162 | 1188 | 420 | 5058 |

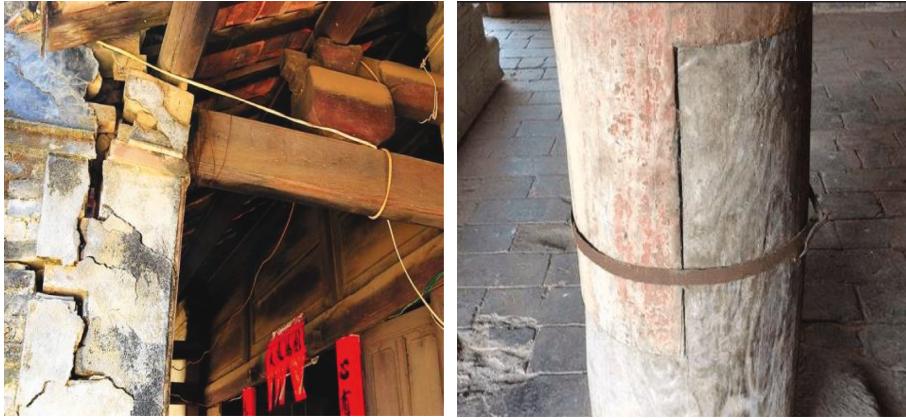


FIGURE 1: Cracks in the timber structures of ancient architecture [1].

the safety of the structure, but under the effect of load and environment, after the general cracks further expand into dangerous cracks, they will cause local structural fractures, resulting in the continuous collapse of the overall structure [2]. According to “Technical code for maintenance and strengthening of ancient timber architecture” and “Code for construction and acceptance of traditional Chinese ancient architecture”, the criteria for identifying dangerous cracks are shown in Table 2. Therefore, how to use new theories and new methods to accurately detect and dynamically monitor cracks in timber structures of ancient architecture that is about to reach dangerous crack indicators is a new problem and challenge in the field of timber structure protection of ancient architecture.

Most ancient timber structures in China only use manual inspection to identify and record cracks, and a small number of ancient timber structures have introduced nondestructive testing instruments such as ultrasonic, stress wave, and Provis. The above-mentioned crack detection methods have the following disadvantages: the first is the huge consumption of manpower, material, and financial resources; the second is that there are blind areas that cannot be reached in the detection process; the third is that the error of the detection result is large and it is closely related to the professional level of the inspectors; the fourth is the long detection cycle, which makes it difficult to feed back the structural emergencies, and the detection lacks timeliness. Therefore, seeking a fast, convenient, and easy-to-operate method for detecting cracks in timber structures of ancient

architecture, can reduce the risk of collapse of timber structures of ancient architecture and improve the level of the intellectualization, specialization, informatization, and refinement of preventive protection of timber structures of ancient architecture has very important scientific, cultural and social significance.

At present, there are few studies on the detection of cracks in ancient architecture, especially the studies on the detection of cracks in timber structures of ancient architecture. Perumal and Venkatachalam [3] proposed a method based on a multi-layer post feedback LSD (Linear Scale-Space Differentiation) model, to extract crack defects and suppress the noise of timber fiber ditching to the greatest extent. However, the accuracy of crack detection needs to be improved. Enrique García-Macías et al. [4] put forward a new seismic damage identification method for historical masonry structures, which was suitable for the use of computationally efficient metamodels for real-time system identification. Wang [5] proposed a new automatic damage detection technology that used the Faster R-CNN model based on the ResNet101 framework to detect two types of damage (weathering and spalling) to historic masonry structures. Dohyung Kwon [6] put forward a system that used deep learning technology to automatically detect and classify damaged cultural relics. M.Cabaleiro [7] proposed an automatic detection algorithm that used LiDAR to collect cracks in timber beams, which could identify, analyze and monitor cracks and their geometric characteristics. These methods are difficult to directly apply to the detection of

TABLE 2: Evaluation index of dangerous cracks [2].

| Crack type | Length index | Width index | Schematic diagram |
|--------------|---|-------------------|-------------------|
| Beam crack 1 | Greater than or equal to 1/3 of the beam length | Greater than 10mm | |
| Beam crack 2 | Greater than or equal to 1/3 of beam height | Greater than 10mm | |
| Column crack | Greater than or equal to 1/3 the column length | Greater than 10mm | |

timber structure cracks in ancient architecture. The main reason is that the existing research objects are mostly nontimber materials such as concrete, masonry, and steel. The pixels of these structural crack images are relatively small and overlap with the background gray value, but it is different from the background and the shape of the noise. The cracks in the timber structure of the ancient architecture have a large percentage of pixels, the gray value of the background is obviously different, and the timber grain noise exists. Therefore, these methods cannot be directly used in the crack detection of ancient timber architecture.

In recent years, deep learning theory, as the latest research result in the field of pattern recognition and machine learning, with its powerful modeling and representation capabilities, is gradually sweeping across the entire graphics research field, especially in tunnels, bridges, pavements, and concrete structures in our country. Phased results have been achieved in crack detection. Armi [8] elaborated on texture image analysis and texture classification, laying a foundation for this study. Pourkaramdel [9] have achieved good results in visual defect detection by using completed local quartet patterns and a majority decision algorithm.

In terms of tunnel and bridge crack detection: Aslam Y [10] proposed a new crack extraction algorithm, which used multi-layer features extracted from a full convolutional network and a naive Bayesian data fusion (NB-FCN) model to automatically split cracks and noise. Belloni et al. [11] combined advanced deep learning technology and innovative photogrammetric algorithms to develop a monitoring system that could effectively detect cracks in invisible images. Based on concrete crack image processing technology, Wang [12] put forward an image preprocessing scheme combining multiple adaptive filtering and contrast enhancement, which could improve the effect of removing background noise and obtaining information on the characteristics of veins and microcracks.

In terms of pavement crack detection: Ji [13] proposed an integrated method for crack detection based on Convolutional Neural Network DeepLabv3+, which was more effective and accurate in crack detection and quantification. Peng [14] proposed a three-threshold pavement crack detection method using random structure forest, which used channel features and paired difference features to enrich the

patch information that constituted the crack image. Song [15] put forward a crack detection network, which could effectively detect crack information in complex environments. Based on a deep learning method, I A Kanaeva [16] used the generated training data to segment the cracks in the driver's view image. Zhang [17,18] optimized and improved the CrackNet crack monitoring software, and the results showed that the accuracy, recall rate, and F-measure were superior to traditional detection methods.

In the aspect of concrete structure crack detection: Wang et al. [19] proposed a method for quantitative classification of cracks of different severity based on deep convolutional neural networks, which used the orthogonal projection method to pre-process the training data, which had good robustness and adaptability to noise and light intensity. Zhao [20] put forward a detection system to investigate possible crack development problems under different construction conditions, which could evaluate the crack behavior in large-scale concrete infrastructure. Qiao [21] proposed a concrete structure crack identification method based on an improved U-Net convolutional neural network to improve the accuracy of crack identification. Song Ee Park [22] used deep learning technology and structured light technology composed of vision and two laser sensors to detect and quantify cracks on the surface of concrete structures. Diana Andrushia [23] proposed a method to detect thermal cracks using ripple transformation, and the main components were noise removal, image enhancement, crack detection and crack geometric parameter detection. Zheng [24] established a fully convolutional network crack detection model, which provided strong theoretical support and practical value for the detection and research of concrete surface cracks. Wu [25] was based on Rayleigh's distributed optical fiber sensing technology to measure the evolution of the strain field related to the initiation and propagation of cracks in concrete structures. Gökhan Bayar [26] used the machine learning algorithm of Tyson polygons to study the crack pattern and propagation of random concrete surfaces. F. Panella [27] combined deep learning and traditional image processing to establish a tool that could detect, locate and measure structural defects. Hyunjun Kim [28] proposed a crack recognition strategy that combined hybrid image processing with UAV technology. Yu [29] proposed a novel method

based on deep convolutional neural networks to identify and localize damages of building structures equipped with smart control devices, this method is capable of automatically extracting high-level features from raw signals or low-level features and optimally selecting the combination of extracted features via a multi-layer fusion to satisfy any damage identification objective. After that, Yu et al. [30] achieved good results in the crack detection of concrete structures by using deep convolutional neural networks optimized by an enhanced chicken swarm algorithm.

It can be inferred from this that the exploration and application of deep learning in the crack detection in timber structures of ancient architecture is an important development trend in solving the difficulties. YOLO (You Only Look Once) is the first single-stage target detection algorithm that has achieved good results in detection accuracy and detection speed. It has been successfully applied in agriculture [31,32], geology [33], remote sensing [34,35] and medicine [36], and other fields. In addition, it is also widely used in the field of transportation, such as traffic sign detection [37], traffic flow detection [38], pavement pit detection [39], and visual crack detection [40]. At the same time, in our published paper “Research on Crack Detection Method of Wooden Ancient Building Based on YOLO v5” [41], we verified that YOLO v5s was feasible for crack detection in ancient architecture.

Currently, YOLO v3 is one of the most popular single-stage detection methods, achieving a huge balance between detection speed and detection accuracy. Recently, the YOLO series has been updated, including three new versions, namely YOLO v4, YOLO v5, and YOLO X. Among them, YOLO v3, YOLO v4, and YOLO v5 are widely used and have good effects in various fields. Therefore, this article will use these three models to study the crack detection effect of ancient architecture timber structures, and from the loss function (Box), loss average (Obj), precision (P), recall rate (R), F1 score, average precision (MAP), frames per second (FPS), inference time (Inference time) and weight (Weight) and other quantitative indicators for performance comparison and analysis. The conclusion of this paper will provide a reference for the selection of cracks detection methods for timber structures of ancient architecture.

2. Materials and Methods

2.1. Data Collection and Processing

2.1.1. Data Collection. Since there is currently no publicly published dataset of cracks in timber structures of ancient architecture in domestic research, to achieve the goal of the research, this paper has taken the Bawang Academy (founded in 1415) on the campus of Shen-yang Jianzhu University as the research object, collected 474 pictures, and established a dataset of cracks in timber structures of ancient architecture. We have published this dataset publicly at <https://github.com/WangMissYou/MaDataSet>, called MaDataSet, which contains both original and annotated images. Some selected samples are shown in Figure 2.

2.1.2. Subsection. The selection, labeling, and data information generation of the pictures have continued to follow the author’s thoughts and practices in the previous academic paper “Research on Crack Detection Method of Wooden Ancient Building Based on YOLO v5” [38] and selected the same batch of pictures of ancient architecture timber structure cracks. To show the forms of being of different types of cracks, a labeling information table of the same crack is generated.

(1) Picture Selection and Crack Marking. In this section, we select data with certain characteristics to illustrate the characteristics of cracks. This part has been published in the paper “Research on Crack Detection Method of Wooden Ancient Building Based on YOLO v5” [38]. On account of the same pictures being used, we have quoted the description of the picture in that paper accordingly. It can be seen from Figure 3(a) that the surface of the ancient architecture’s timber structure is rough, there is bark interference, and the cracks are not obvious. Such cracks need to show more obvious characteristics when marking, and the cracks interfered with by bark are not marked; Figure 3(b) shows that the surface of the timber structure of ancient architecture is relatively smooth, and there is local weathering, and the characteristics of cracks are obvious. At the same time, the spacing of some cracks is small. When such cracks are marked, the cracks with smaller spacing can be combined and marked separately. Separate cracks are marked separately; as can be seen from Figure 3(c), the surface of the timber structure of ancient architecture with paint is relatively smooth, with obvious crack features, a large number of cracks, and a short length. Such visible cracks need to be marked clearly all one by one; as can be seen from Figure 3(d), the surface of the timber structure of the ancient architecture is relatively smooth, and the characteristics of the cracks are obvious. After manual identification, it is found that there is wood grain, only cracks are marked, and wood grain is not marked; as can be seen from Figure 3(e), the surface of the timber structure of ancient architecture is relatively rough, with peeling and different colors, and there are also small cracks in the peeling part. After manual identification, it is sure that it is not a timber structure crack. When such cracks are marked, you can select the representative cracks with obvious characteristics from the clear and fuzzy cracks to mark, and the cracks caused by the peeling are not marked; from Figure 3(f), it can be seen that the paint on the surface of the ancient architecture’s timber structure has peeled off, and the cracks are clearly distributed. The length of the cracks is different, all such cracks can be marked when marking; from Figure 3(g), it can be seen that the surface of the ancient architecture timber structure is smooth, the cracks contrast sharply, and there are particularly obvious and regular wide and narrow cracks. Similar cracks can be marked directly when marking; as can be seen from Figure 3(h), the surface of the ancient architecture timber structure is smooth, the number of cracks is small, and the characteristics of the length of the cracks are



FIGURE 2: The crack examples in timber structure of ancient architecture. (a) Door crack1 (b) door crack2 (c) window crack1, (d) window crack2 (e) pillar crack1 and (f) pillar crack2.

obvious. Such cracks can be marked with one long crack and one short crack respectively. In the case of a large number of cracks, you can mark a few more as appropriate. As can be seen from Figure 3(i), the surface of the timber structure of the ancient architecture is rough, the timber structure is weathered and corroded, the log body is clearly visible, and the number of cracks is large. The spacing is small and such cracks can be combined and marked according to the concentrated area of the crack density when marking, and the timber grain of the log itself is not processed.

(2) *Generate Data Information.* File After the crack pictures are marked, save it, and use the label software LabelImg system to generate a txt file of the crack pictures marking information, including the number and location of the crack marks, and name it “crack”, as shown in Table 3.

3. YOLO Model Training

3.1. Model Validation. YOLO is a new target detection method, which is characterized by fast detection and high accuracy. Base YOLO model processes images in real-time at 45 frames per second. The YOLO networks have 24 convolutional layers followed by 2 fully connected layers. Alternating 1x1 convolutional layers reduce the feature space

from preceding layers. YOLO can pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection. The basic network structure is shown in Figure 4 [42].

Many YOLO versions have been produced after years of research. YOLO v3 algorithm includes three models: YOLO v3, YOLO v3-tiny, and YOLO v3-SPP; YOLO v4 algorithm includes four models: YOLO v4s-mish, YOLO v4m-mish, YOLO v4l-mish, and YOLO v4x-mish; YOLO v5 algorithm includes four models: YOLO v5n, YOLO v5s, YOLO v5l, and YOLO v5x. This paper selects the YOLO v3, YOLO v4s-mish, and YOLO v5s models for research. The main reasons are: first, the three models can generate test results graphs with the same indicators, which is convenient for comparison and analysis; second, the three models are lightweight, suitable for target detection in small scenes and small and medium data sets; third, the three models are based on the PyTorch deep learning framework developed by Facebook, and have good results in simple target detection.

3.2. Figures, Tables, and Schemes. This test uses a Dell laptop, the model is G15 5511-R1866B2021, the system configuration is GeForce RTX3060, the CPU model is 17-11800H, and



FIGURE 3: Different types of timber structure cracks in ancient architecture. (a) Type1 (b) type2 (c) type3, (d) type4 (e) type5 (f) type6, (g) type7 (h) type8 and (i) type9.

the internal memory is 16G, the operating system is Windows10, the development platform uses pycharm, and the programming language is Python. Based on the PyTorch

deep learning framework developed by Facebook, a series of models of the YOLO algorithm are trained using the established COCO data set.

3.3. Training Program. This test has selected 6608 cracks from 474 pictures of “Wang Ba Academy” as training samples and uses YOLO v3, YOLO v4s-mish, and YOLO v5s models for training to detect the cracks in the pictures, the total number of training rounds for each model is 450 rounds. Hyperparameters setting refers to the research results of Al-qudah and Suen [43], this work involves four steps: training, testing, model selection, and deployment. For the first step, the training phase is paired with a “models generator” component that can generate n-different models by optimizing the network hyperparameters based on various factors such as available processing power, batch size, and available images in the dataset. Therefore, this step will yield n-trained models instead of just one trained model. To test the robustness of each trained model, the testing phase is associated with a “testing configurator” that generates all possible testing configurations based on the optimized hyperparameters from the training phase. Each trained model is tested against all possible configurations. The model selector then selects the best model based on the recall ratio. Finally, the system is deployed for usage in real applications. For the deployment of the real application, the system can employ an estimation component that can match the system with the best model that best fits the system’s requirements which might not be the one with the highest recall ratio. According to the actual needs of the test, this paper defines the data set file as a “crack.yaml” and uses a single GPU to accelerate training.

4. Results and Discussion

4.1. Model Evaluation Indicators

4.1.1. Precision and Recall. In target detection, accuracy and recall are typical and important evaluation indicators. Accuracy represents the ratio of the number of pairs found to all the numbers found, and it measures the probability of a positive class classified by a classifier. The recall rate represents the ratio of the positive class that should be found to all the positive classes that should have been found, and it measures the ability of a class to find all the positive classes. The calculation formulas of precision and recall rate are shown in formula 1 and formula 2:

$$P = \frac{TP}{FP + TP}, \quad (1)$$

$$R = \frac{TP}{FN + TP}. \quad (2)$$

In the formula: P is the precision; R is the recall rate; TP is the number of positive samples that are identified as containing cracks; FP is the number of negative samples that are identified without cracks; FN is the number of positive samples that are not identified as containing cracks number of samples.

4.1.2. AP Value and F1 Score. When the detection target is identified, the index of inspection accuracy is represented by mAP, which represents the average value of multiple

categories of AP. Because the test sample has only one type of cracks in the timber structure of ancient architecture, AP can be directly used as an evaluation index for crack detection. AP represents the pros and cons of the model’s detection effect. The larger the value is, the better the detection result is. According to recall and precision to make a curve, the area under the curve is AP, and the AP value is the integration of the area. The specific formula is shown in the following formula:

$$AP = \int_0^1 P(R)dR. \quad (3)$$

F1 score is generally used to evaluate the comprehensive performance of the model, and its calculation formula is shown in the following formula:

$$F1 = \frac{2P \cdot R}{P + R}. \quad (4)$$

4.1.3. Box, Objectness, and Classification. Box represents the loss of using GIoU Loss as the bounding box. The box is inferred to be the mean value of the GIoU loss function. The smaller the value is, the more accurate the detected box is. Val Box represents the bounding box loss of the validation set. Object-ness means that it is presumed to be the average value of target detection loss. The smaller the value is, the more accurate the target detection is. Val Objectness represents the average value of the target detection loss in the validation set. Classification indicates that it is estimated to be the average value of the classification loss. The smaller the value is, the more accurate the classification is. Val Classification represents the mean value of the classification loss of the validation set.

4.1.4. FPS and Training Time. FPS is a related concept in the image field, which represents the number of image frames transmitted per second during the training process. When the FPS value of the model exceeds 30, it is proved that the model can realize real-time image processing. The training time is the total time used to complete the picture training. Both of these indicators are evaluation indicators that consider the speed of model training images.

4.2. Model Evaluation Indicators

4.2.1. Model Training Results. In this paper, three models including YOLO v3, YOLO v4s-mish, and YOLO v5s are used to train the cracks in the timber structure of ancient architecture, and the numerical results of related indicators are obtained. Among them, the loss function of the YOLO v3 model is 0.026, the recall rate is 91.64%, and the mAP value is 0.955, which is better than the YOLO v4s-mish and YOLO v5s models. The minimum accuracy of the YOLO v4s-mish model is 60%, the maximum loss function is 0.042, and the minimum F1 value is 0.717, indicating that its comprehensive performance is relatively poor. The loss function of the YOLO v5s model is 0.037, the F1 value is 0.914, which is between YOLO v3 and

TABLE 3: Crack marking information table.

| Category code | Coordinate 1 | Coordinate 2 | Coordinate 3 | Coordinate 4 |
|---------------|--------------|--------------|--------------|--------------|
| 0 | 0.215278 | 0.025546 | 0.023148 | 0.045635 |
| 0 | 0.202546 | 0.092262 | 0.014881 | 0.074485 |
| 0 | 0.221726 | 0.098526 | 0.012235 | 0.065476 |
| 0 | 0.260086 | 0.060744 | 0.022817 | 0.075893 |
| 0 | 0.268747 | 0.150546 | 0.026786 | 0.086318 |

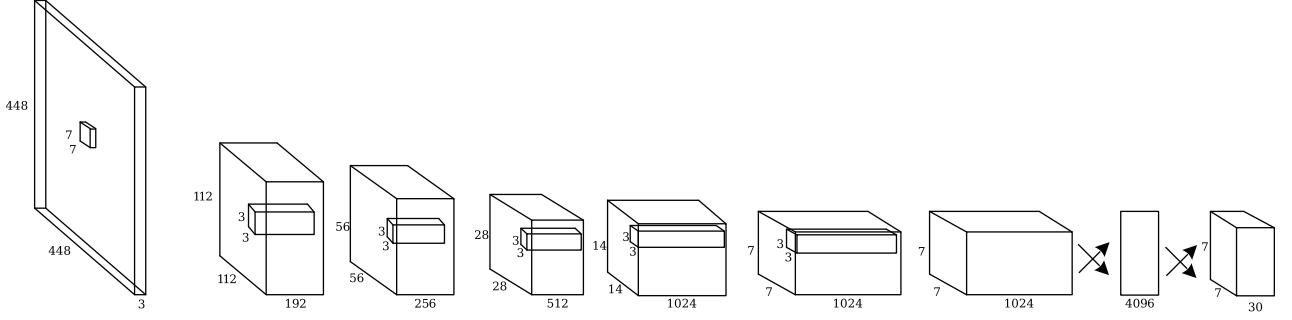


FIGURE 4: The network structure diagram of YOLO.

TABLE 4: YOLO v3 training results.

| Epoch | Gpu_men | Box | Obj | Clss | Total | Labels | Img_size | Precision | Recall | mAP@0.5 |
|---------|---------|--------|--------|------|--------|--------|----------|-----------|--------|---------|
| 0/449 | 4.06 G | 0.1105 | 0.0411 | 0 | 0.1516 | 82 | 640 | 0.0071 | 0.0318 | 0.0006 |
| 1/449 | 2.92 G | 0.1005 | 0.0489 | 0 | 0.1495 | 27 | 640 | 0.0230 | 0.0260 | 0.0019 |
| 2/449 | 2.92 G | 0.0926 | 0.0472 | 0 | 0.1397 | 78 | 640 | 0.0740 | 0.0679 | 0.0156 |
| 3/449 | 2.93 G | 0.0861 | 0.0484 | 0 | 0.1345 | 79 | 640 | 0.1690 | 0.1811 | 0.0769 |
| 4/449 | 2.93 G | 0.0845 | 0.0464 | 0 | 0.1309 | 76 | 640 | 0.1969 | 0.2161 | 0.1016 |
| : | : | : | : | : | : | : | : | : | : | : |
| 445/449 | 2.93 G | 0.0269 | 0.0218 | 0 | 0.0487 | 106 | 640 | 0.9198 | 0.9198 | 0.9559 |
| 446/449 | 2.93 G | 0.0261 | 0.0209 | 0 | 0.0470 | 33 | 640 | 0.9224 | 0.9156 | 0.9549 |
| 447/449 | 2.93 G | 0.0261 | 0.0206 | 0 | 0.0468 | 74 | 640 | 0.9145 | 0.9239 | 0.9557 |
| 448/449 | 2.93 G | 0.0267 | 0.0224 | 0 | 0.0491 | 75 | 640 | 0.9218 | 0.9223 | 0.9565 |
| 449/449 | 2.93 G | 0.0261 | 0.0213 | 0 | 0.0474 | 84 | 640 | 0.9253 | 0.9164 | 0.9554 |

TABLE 5: YOLO v4 training results.

| Epoch | Gpu_men | Box | Obj | Clss | Total | Labels | Img_size | Precision | Recall | mAP@0.5 |
|---------|---------|--------|--------|------|--------|--------|----------|-----------|--------|---------|
| 0/449 | 1.32 G | 0.1295 | 0.0752 | 0 | 0.2048 | 24 | 640 | 0 | 0 | 0.0008 |
| 1/449 | 1.35 G | 0.1162 | 0.0814 | 0 | 0.1977 | 22 | 640 | 0 | 0 | 0.0052 |
| 2/449 | 1.35 G | 0.2085 | 0.0862 | 0 | 0.1927 | 13 | 640 | 0.1217 | 0.0418 | 0.0175 |
| 3/449 | 1.35 G | 0.1026 | 0.0893 | 0 | 0.1920 | 62 | 640 | 0.0893 | 0.2780 | 0.0786 |
| 4/449 | 1.35 G | 0.0973 | 0.0923 | 0 | 0.1896 | 3 | 640 | 0.0984 | 0.2868 | 0.0769 |
| : | : | : | : | : | : | : | : | : | : | : |
| 445/449 | 1.36 G | 0.0417 | 0.0417 | 0 | 0.0997 | 11 | 640 | 0.5950 | 0.8913 | 0.8812 |
| 446/449 | 1.36 G | 0.0426 | 0.0426 | 0 | 0.1052 | 15 | 640 | 0.5813 | 0.8941 | 0.8815 |
| 447/449 | 1.36 G | 0.0420 | 0.0420 | 0 | 0.1022 | 56 | 640 | 0.5933 | 0.8909 | 0.8823 |
| 448/449 | 1.36 G | 0.0434 | 0.0434 | 0 | 0.1043 | 106 | 640 | 0.6014 | 0.8809 | 0.8817 |
| 449/449 | 1.36 G | 0.0423 | 0.0423 | 0 | 0.1032 | 29 | 640 | 0.5910 | 0.8921 | 0.8823 |

YOLO v4s-mish, the accuracy is up to 92.91%, and the mAP value is 0.929, showing a good overall performance. The training results of the three models are shown in Table 4–6, and Figure 5–7.

4.2.2. Analysis of Training Results. The above chart information is refined and summarized, and evaluation indexes such as FPS, inference time, total runtime, and weight are added at the same time to form the evaluation index table

TABLE 6: YOLO v5 training results.

| Epoch | Gpu_men | Box | Obj | Cls | Total | Labels | Img_size | Precision | Recall | mAP@0.5 |
|---------|---------|--------|--------|-----|--------|--------|----------|-----------|--------|---------|
| 0/449 | 1.81 G | 0.1235 | 0.0733 | 0 | 0.1968 | 106 | 640 | 0.0037 | 0.0118 | 0.0002 |
| 1/449 | 1.86 G | 0.1187 | 0.0811 | 0 | 0.1999 | 143 | 640 | 0.0048 | 0.0240 | 0.0003 |
| 2/449 | 1.88 G | 0.1157 | 0.0848 | 0 | 0.2005 | 90 | 640 | 0.0087 | 0.0253 | 0.0007 |
| 3/449 | 1.88 G | 0.1125 | 0.0855 | 0 | 0.1980 | 80 | 640 | 0.0225 | 0.0318 | 0.0035 |
| 4/449 | 1.88 G | 0.1096 | 0.0911 | 0 | 0.2012 | 104 | 640 | 0.0427 | 0.0572 | 0.0095 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 445/449 | 1.88 G | 0.0377 | 0.0613 | 0 | 0.0989 | 174 | 640 | 0.9345 | 0.8854 | 0.9330 |
| 446/449 | 1.88 G | 0.0372 | 0.0627 | 0 | 0.0998 | 121 | 640 | 0.9326 | 0.8873 | 0.9341 |
| 447/449 | 1.88 G | 0.0379 | 0.0614 | 0 | 0.0994 | 87 | 640 | 0.9220 | 0.8910 | 0.9310 |
| 448/449 | 1.88 G | 0.0382 | 0.0606 | 0 | 0.0988 | 72 | 640 | 0.9401 | 0.8738 | 0.9309 |
| 449/449 | 1.88 G | 0.0376 | 0.0599 | 0 | 0.0975 | 52 | 640 | 0.9273 | 0.8882 | 0.9291 |

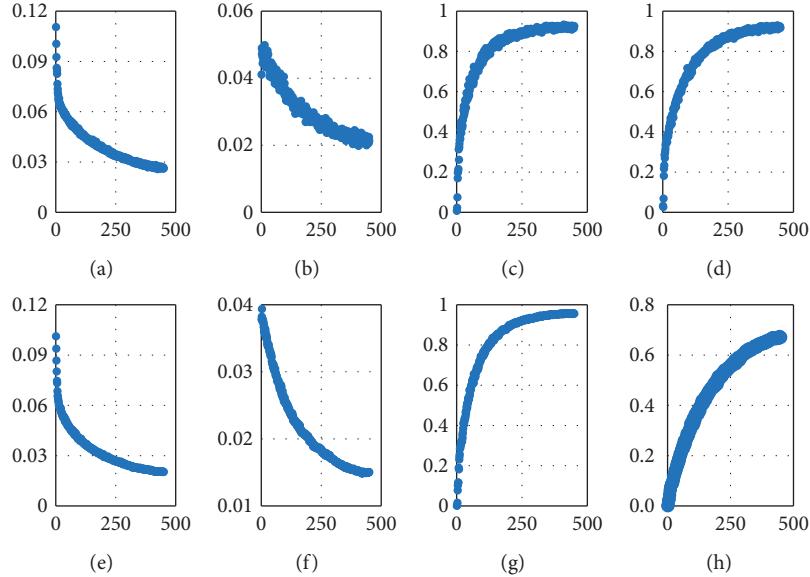


FIGURE 5: YOLO v3 training result graph. (a) Box (b) objectness, (c) precision, (d) recall, (e) Val Box, (f) Val Objectness, (g) mPA@0.5, and (h) mPA@0.5:0.95.

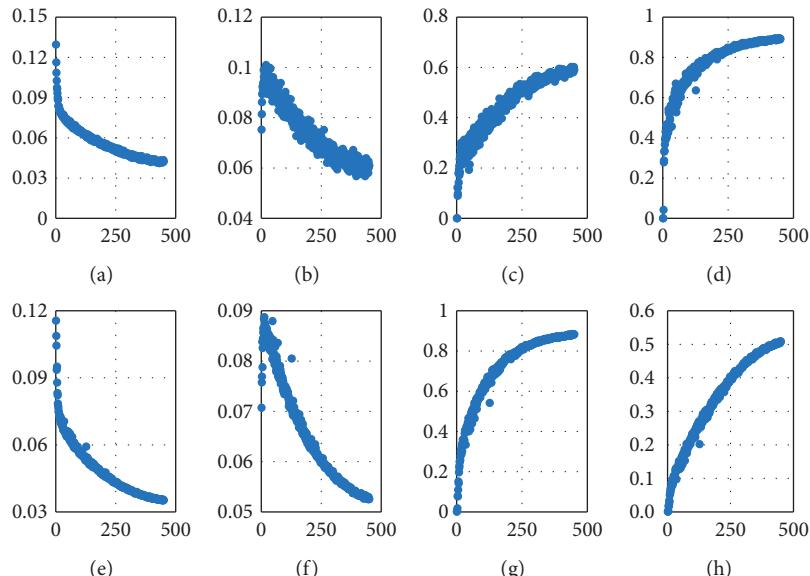


FIGURE 6: YOLO v4 training result graph. (a) GloU, (b) objectness, (c) precision, (d) recall, (e) Val GloU, (f) Val Objectness, (g) mPA@0.5, and (h) mPA@0.5:0.95.

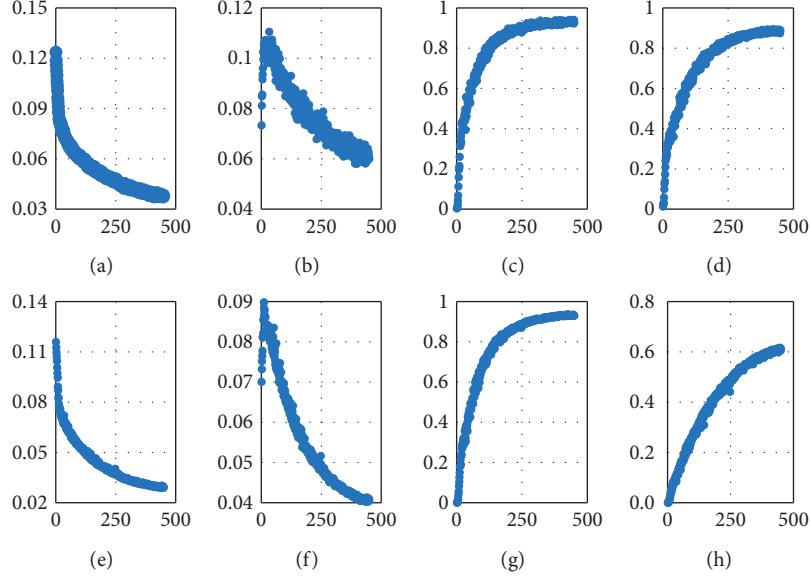


FIGURE 7: YOLO v4 training result graph. (a) Box, (b) objectness, (c) precision, (d) recall, (e) Val GloU, (f) Val Objectness, (g) mPA@0.5, and (h) mPA@0.5:0.95.

TABLE 7: Evaluation indicators of three models.

| Model | Box | Obj | P(%) | R(%) | F1 | mAP | FPS | Inference time (ms) | Total runtime (h) | Weight (MB) |
|---------------|-------|-------|-------|-------|-------|-------|--------|---------------------|-------------------|-------------|
| YOLO V3 | 0.026 | 0.021 | 92.53 | 91.64 | 0.922 | 0.955 | 71.43 | 21 | 20.79 | 118 |
| YOLO V4s-mish | 0.042 | 0.060 | 60.00 | 89.21 | 0.717 | 0.882 | 100.00 | 11 | 13.25 | 18 |
| YOLO V5s | 0.037 | 0.059 | 92.91 | 88.82 | 0.914 | 0.929 | 166.67 | 8.1 | 12.86 | 14 |

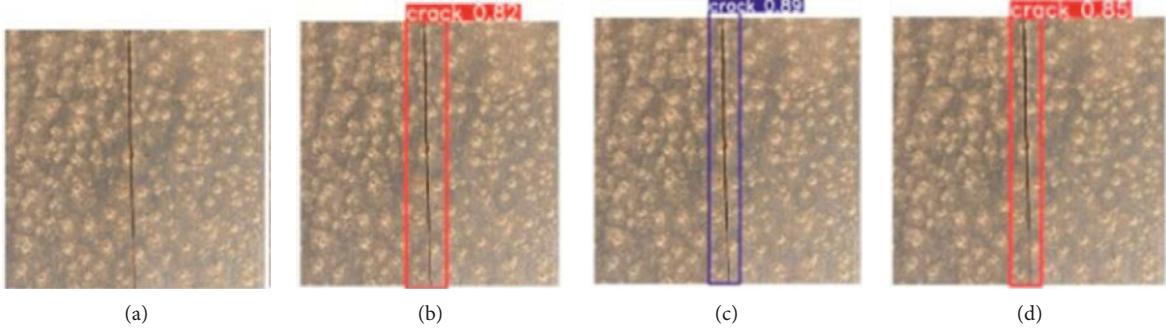


FIGURE 8: Model test effect pictures, (a) Test chart, (b) YOLO v3, (c) YOLO v4s-mish, and (d) YOLO v5s.

required for the performance comparison of the models in this paper, as shown in Table 7.

Table 7 summarizes the training results of the three models including YOLO v3, YOLO v4s-mish, and YOLO v5s. It can be seen that most of the evaluation indicators of the YOLO v3 model are effective, but the maximum weight is 118MB, the minimum FPS value is 71.43, and the longest inference time is 21ms, indicating that the greater the weight of the model is, the better the effect of the related evaluation indicators is. However, due to the complex network structure, the inference time will be prolonged. The YOLO v4s-mish model is improved based on the original YOLO v4

model. Its weight is 18MB and the network structure is relatively simple. But from the evaluation index value, the overall performance is poor and there is no significant advantage in training speed. The minimum weight of the YOLO v5s model is only 14MB, which is the simplest network structure among the three models. The maximum FPS value is 166.67, and the minimum inference time is 8.1ms. This shows that the training speed of this model is the fastest compared to the other two models. It can be seen from the above analysis that the YOLO v5s model has a better overall performance considering the accuracy, training speed, and network structure complexity.

4.2.3. Analysis of Test Results. Figure 8 shows a part of the test results of the three models YOLO v3, YOLO v4s-mish, and YOLO v5s. It can be seen that the confidence of the YOLO v3 model is 0.82, the confidence of the YOLO v4s-mish model is 0.89, and the confidence of the YOLO v5s model is 0.85. It shows that the test results of the three models are relatively close, and good test results have been achieved.

5. Conclusions

5.1. Research Conclusion. Based on the author's previous research, this paper continues to explore the application performance of the YOLO series models in the detection of cracks in ancient architecture timber structures and compares the training and testing effects of YOLO v3, YOLO v4s-mish, and YOLO v5s. The results are obtained as follows:

- (1) The loss function of the YOLO v3 model is 0.026, the recall rate is 91.64%, and the mAP value is 0.955, which are better than the YOLO v4s-mish and YOLO v5s models, but the maximum weight is 118MB, the minimum FPS value is 71.43, and the maximum inference time is 21ms, indicating that the model has good overall performance, but the training speed is slow.
- (2) The YOLO v4s-mish model is improved based on the original YOLO v4 model, its weight is 18MB, and the network structure is relatively simple. However, the minimum accuracy of this model is 60%, the maximum loss function is 0.042, and the minimum F1 value is 0.717, indicating that its comprehensive performance is relatively poor. At the same time, there is no obvious advantage in detection speed.
- (3) The loss function of the YOLO v5s model is 0.037, the F1 value is 0.914, which are both between YOLO v3 and YOLO v4s-mish, the accuracy is up to 92.91%, and the mAP value is 0.929, showing a good comprehensive performance. At the same time, the minimum weight of this model is only 14MB, which is the simplest network structure among the three models. The maximum FPS value is 166.67, and the minimum inference time is 8.1ms, which shows that the training speed of this model is the fastest compared to the other two models.
- (4) The confidence of the YOLO v3 model is 0.82, the confidence of the YOLO v4s-mish model is 0.89, and the confidence of the YOLO v5s model is 0.85. The confidence of the three models is above 0.8, indicating that the three models have better test results.

5.2. Future Research Directions. The author's research on the detection of cracks in the timber structures of ancient architecture is in its infancy, and the next step of the research work will continue to strengthen in the following aspects:

- (1) Use advanced technology such as drone tilt photogrammetry to obtain images of cracks in timber structures of ancient architecture, continuously

expand data sets, strengthen image screening and processing, obtain more effective data sets, and improve crack detection accuracy.

- (2) Establish a deep learning-based detection, positioning, measurement, and analysis integrated ancient architecture timber structure crack monitoring system to realize dynamic monitoring of crack development trends.
- (3) U-Net, Mask R-CNN, and other models have also achieved good results in the field of image recognition. Comparing the training results of the YOLO series model with these two models will help to find out more suitable methods for the crack detection of ancient architecture timber structures.

Data Availability

The data presented in this study are available on request from the corresponding author.

Disclosure

The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

We thank Professor Yan and Professor Liu for their guidance on this paper. This research was funded by The National Natural Science Foundation of China, grant number 51908379.

References

- [1] X. Yuan, Y. P. Chen, and W. J. Guo, "Features and prevention of common damages in ancient timber structures," *Chinese Journal of Timber Science and Technology*, vol. 35, no. 05, pp. 55–59, 2021.
- [2] N. Yang, C. Zhang, and T. H. Li, "Design of crack monitoring system for Chinese ancient timber architecture based on UAV and CV," *Engineering Mechanics*, vol. 38, no. 3, pp. 27–39, 2021.
- [3] R. Perumal and S. B. Venkatachalam, "Non-invasive detection of moss and crack in monuments using image processing techniques," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 5277–5285, 2021.
- [4] E. García-Macías, I. Venanzi, and F. Ubertini, "Metamodel-based pattern recognition approach for real-time identification of earthquake-induced damage in historic masonry structures," *Automation in Construction*, vol. 120, Article ID 103389, 2020.
- [5] N. Wang, X. Zhao, P. Zhao, Y. Zhang, Z. Zou, and J. Ou, "Automatic damage detection of historic masonry buildings based on mobile deep learning," *Automation in Construction*, vol. 103, pp. 53–66, 2019.

- [6] D. Kwon and J. Yu, "Automatic damage detection of stone cultural property based on deep learning algorithm," in *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W15, pp. 639–643, 2019.
- [7] M. Cabaleiro, R. Lindenbergh, W. Gard, P. Arias, and J van de Kuilen, "Algorithm for automatic detection and analysis of cracks in timber beams from LiDAR data," *Construction and Building Materials*, vol. 130, pp. 41–53, 2017.
- [8] L. Armi and F.-E. Shervan, "Texture image analysis and texture classification methods-A review," 2019, <http://arxiv.org/abs/1904>, Article ID 06554.
- [9] Z. Pourkaramdel, S. Fekri-Ershad, and L. Nanni, "Fabric defect detection based on completed local quartet patterns and majority decision algorithm," *Expert Systems with Applications*, vol. 198, Article ID 116827, 2022.
- [10] Y. Aslam, N. Santhi, N. Ramasamy, and K. Ramar, "Localization and segmentation of metal cracks using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 4205–4213, 2021.
- [11] V. Belloni, A. Sjölander, R. Ravanelli, M. Crespi, and A. Nascetti, "Tack project: tunnel and bridge automatic crack monitoring using deep learning and photogrammetry," in *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B4-2020, pp. 741–745, 2020.
- [12] Y. Wang, J. Y. Zhang, J. X. Liu et al., "Research on crack detection algorithm of the concrete bridge based on image processing," *Procedia Computer Science*, vol. 154, pp. 610–616, 2019.
- [13] A. Ji, X. Xue, Y. Wang, X. Luo, and W. Xue, "An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement," *Automation in Construction*, vol. 114, Article ID 103176, 2020.
- [14] C. Peng, M. Yang, Q. Zheng et al., "A triple-thresholds pavement crack detection method leveraging random structured forest," *Construction and Building Materials*, vol. 263, Article ID 120080, 2020.
- [15] W. D. Song, G. H. Jia, D. Jia, and H. Zhu, "Automatic pavement crack detection and classification using multiscale feature attention network," *IEEE Access*, vol. 7, pp. 171001–171012, 2019.
- [16] I. A. Kanaeva and J. A. Ivanova, "Road pavement crack detection using deep learning with synthetic data," *IOP Conference Series: Materials Science and Engineering*, vol. 1019, no. 1, Article ID 012036, 2021.
- [17] A. Zhang, K. C. P. Wang, Y. Fei et al., "Deep learning-based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet," *Journal of Computing in Civil Engineering*, vol. 32, no. 5, Article ID 04018041, 2018.
- [18] A. Zhang, K. C. P. Wang, B. X. Li et al., "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.
- [19] W. Wang, W. Hu, W. Wang et al., "Automated crack severity level detection and classification for ballastless track slab using deep convolutional neural network," *Automation in Construction*, vol. 124, Article ID 103484, 2021.
- [20] Z. Zhao, B. Chen, Z. Wu, and S. Zhang, "Multi-sensing investigation of crack problems for concrete dams based on detection and monitoring data: a case study," *Measurement*, vol. 175, Article ID 109137, 2021.
- [21] W. Qiao, H. Zhang, F. Zhu, and Q. Wu, "A crack identification method for concrete structures using improved U-net convolutional neural networks," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–16, Article ID 6654996, 2021.
- [22] S. E. Park, S. H. Eem, and H. Jeon, "Concrete crack detection and quantification using deep learning and structured light," *Construction and Building Materials*, vol. 252, Article ID 119096, 2020.
- [23] A. D. Andrushia, N. Anand, and G. Prince Arulraj, "Evaluation of thermal cracks on fire exposed concrete structures using Ripple transform," *Mathematics and Computers in Simulation*, vol. 180, pp. 93–113, 2021.
- [24] M. Zheng, Z. Lei, and K. Zhang, "Intelligent detection of building cracks based on deep learning," *Image and Vision Computing*, vol. 103, pp. 103987–103996, 2020.
- [25] J. Wu, H. Liu, P. Yang, B. Tang, and G. Wei, "Quantitative strain measurement and crack opening estimate in concrete structures based on OFDR technology," *Optical Fiber Technology*, vol. 60, Article ID 102354, 2020.
- [26] G. Bayar and T. Bilir, "A novel study for the estimation of crack propagation in concrete using machine learning algorithms," *Construction and Building Materials*, vol. 215, pp. 670–685, 2019.
- [27] F. Panella, J. Boehm, Y. Loo, A. Kaushik, and D. Gonzalez, "Deep learning and image processing for automated crack detection and defect measurement in underground structures," in *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2, pp. 829–835, 2018.
- [28] H. Kim, J. Lee, E. Ahn, S. Cho, M. Shin, and S. H. Sim, "Concrete crack identification using a UAV incorporating hybrid image processing," *Sensors*, vol. 17, no. 9, pp. 2052–2065, 2017.
- [29] Y. Yu, C. Wang, X. Gu, and J. Li, "A novel deep learning-based method for damage identification of smart building structures," *Structural Health Monitoring*, vol. 18, no. 1, pp. 143–163, 2019.
- [30] Y. Yu, M. Rashidi, B. Samali, M. Mohammadi, T. N. Nguyen, and X. Zhou, "Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm," *Structural Health Monitoring*, Article ID 147592172110535, 2022.
- [31] G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim, "YOLO-Tomato: a robust algorithm for tomato detection based on YOLOv3," *Sensors*, vol. 20, no. 7, pp. 2145–2164, 2020.
- [32] J. Liu and X. Wang, "Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model," *Plant Methods*, vol. 16, no. 1, pp. 83–6256, 2020.
- [33] H. Ma, Y. Liu, Y. Ren, and J. Yu, "Detection of collapsed buildings in post-earthquake remote sensing images based on the improved YOLOv3," *Remote Sensing*, vol. 12, no. 1, pp. 44–62, 2019.
- [34] J. Zhou, Y. Tian, C. Yuan, K. Yin, G. Yang, and M. Wen, "Improved UAV opium poppy detection using an updated YOLOv3 model," *Sensors*, vol. 19, no. 22, pp. 4851–4873, 2019.
- [35] L. Pang, H. Liu, Y. Chen, and J. Miao, "Real-time concealed object detection from passive millimeter wave images based on the YOLOv3 algorithm," *Sensors*, vol. 20, no. 6, pp. 1678–1692, 2020.
- [36] S. Yao, Y. Chen, X. Tian, R. Jiang, and S. Ma, "An improved algorithm for detecting pneumonia based on YOLOv3," *Applied Sciences*, vol. 10, no. 5, pp. 1818–1833, 2020.
- [37] H. Zhang, L. Qin, J. Li et al., "Real-time detection method for small traffic signs based on Yolov3," *IEEE Access*, vol. 8, pp. 64145–64156, 2020.

- [38] Y. Q. Huang, J. C. Zheng, S. D. Sun, C. F. Yang, and J. Liu, “Optimized YOLOv3 algorithm and its application in traffic flow detections,” *Applied Sciences*, vol. 10, no. 9, pp. 3079–3093, 2020.
- [39] E. N. Ukhwah, E. M. Yuniarso, and Y. K. Suprapto, “Asphalt pavement pothole detection using deep learning method based on YOLO neural network,” in *Proceedings of the 2019 International Seminar on Intelligent Technology and Its Applications*, pp. 35–40, IEEE, Surabaya Indonesia, August 2019.
- [40] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, “Pavement distress detection and classification based on YOLO network,” *International Journal of Pavement Engineering*, vol. 22, no. 13, pp. 1659–1672, 2020.
- [41] J. Ma, W. D. Yan, and G. Q. Liu, “Research on crack detection method of timber ancient architecture based on YOLO v5,” *Journal of Shenyang Jianzhu University (Natural Science)*, vol. 37, no. 05, pp. 927–934, 2021.
- [42] J. Redmon, D. Santosh, G. Ross, and F. Ali, “You only look once: unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, IEEE, Las Vegas, NV, USA, 27 June 2016.
- [43] R. Al-qudah and C. Y. Suen, “Enhancing YOLO deep networks for the detection of license plates in complex scenes,” in *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, Petra, Jordan, December 2019.