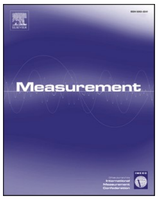
Measurement 240 (2025) 115603

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement



Enhancing autonomous pavement crack detection: Optimizing YOLOv5s algorithm with advanced deep learning techniques

Shuangxi Zhou [a], Dan Yang [b], Ziyu Zhang [a], Jinwen Zhang [a], Fulin Qu [c,*], Piyush Punetha [d], Wengui Li [c], Ning Li [e]

[a] *School of Civil and Engineering Management, Guangzhou Maritime University, Guangzhou, 510725, China*
[b] *School of Civil Engineering and Architecture, East China Jiaotong University, Nanchang, 330013, China*
[c] *Centre for Infrastructure Engineering and Safety, School of Civil and Environmental Engineering, University of New South Wales, NSW 2052, Australia*
[d] *School of Civil and Environmental Engineering, University of Technology Sydney, NSW 2007, Australia*
[e] *Department of Solids and Structures, University of Manchester, Manchester M13 9PL, United Kingdom*

ARTICLE INFO

Keywords:
Road maintenance
Crack detection
YOLO algorithm
AI integration
Advanced image processing

ABSTRACT

To enhance the safety and comfort of vehicle travel, detecting pavement cracks is a critical task in road management. This article introduces an advanced single-stage target detection method utilizing the YOLOv5s algorithm to enhance real-time performance and accuracy. Initially, Squeeze-and-Excitation Networks are integrated into the model to facilitate self-learning for improved crack characterization. Subsequently, anchors computed through the K-means clustering algorithm are closely aligned with the fracture dataset, achieving an adaptation rate of 99.9 % and enhancing the recall rate of the model. Furthermore, the inclusion of the SimSPPF module from YOLOv6 diminishes memory usage and expedites detection speed. By replacing the original nearest up-sampling method with transposed convolution, optimization of up-sampling for crack datasets is achieved. Performance assessments reveal that the refined YOLOv5s algorithm attains an F1 score of 91 %, a mean Average Precision (mAP) of 93.6 %, and a 1.54 % increase in frames per second (fps) for pavement crack detection. This enhancement in detection technology signifies a substantial advancement in the maintenance and longevity of road infrastructure.

1. Introduction

Highway infrastructure is continually advancing; however, numerous roads still exhibit significant cracking due to inadequate load-bearing capacities [1,2]. Minor fissures compromise vehicular stability and diminish aesthetic appeal, while larger breaches present substantial safety risks. Consequently, the meticulous detection and subsequent repair of pavement cracks are of paramount importance.

Conventional detection methods, such as manual visual inspections, are not only laborious and time-intensive but also incur substantial costs [3–8]. The integration of automated pavement crack detection techniques, underpinned by advanced deep learning algorithms, not only mitigates these expenses but also markedly augments operational efficiency [9–14]. Utilizing sophisticated object detection methodologies such as deep learning target detection, instance segmentation, and semantic segmentation (which have been successfully applied in diverse sectors including agriculture, transportation, and healthcare), deep learning facilitates a more effective and precise identification of pavement anomalies. These techniques are categorized into two primary types: (a) two-stage target detection, which initially isolates the object region candidate frame employing a Convolutional Neural Network (CNN), subsequently undertaking CNN-based classification and recognition as exemplified by the Region-based Convolutional Neural Network (RCNN) series and Spatial Pyramid Pooling Network (SPPNet) [15,16]; and (b) single-stage target detection, which seamlessly extracts both the category and location of the target object using robust backbone extraction networks like the YOLO series and Single Shot MultiBox Detector (SSD) without the necessity for region candidate frames [17–19]. Embracing deep learning methodologies for pavement crack detection not only elevates the precision and speed of the detection processes, but also underscores the transformative potential of integrating these advanced technologies into traditional road maintenance regimes.

In the domain of two-stage target detection, Faster-RCNN has been

[*] Corresponding author.
E-mail address: Fulin.qu@unsw.edu.au (F. Qu).

https://doi.org/10.1016/j.measurement.2024.115603
Received 5 May 2024; Received in revised form 5 August 2024; Accepted 25 August 2024
Available online 28 August 2024

widely adopted for pavement crack detection [20,21]. For instance, Sekar and Perumal [22] proposed replacing the backbone feature extraction networks in Faster-RCNN with VGG16, MobileNet-V2, and ResNet50 to better address the challenges of distinguishing cracks on complex asphalt backgrounds. An attention mechanism was also integrated into the ResNet50 network, enhancing detection accuracy to 85.64 % and effectively highlighting subtle discrepancies in asphalt surfaces. Similarly, Hao et al. [23] focused on runway cracks, employing MobileNet-V2 as the backbone in a modified Faster-RCNN, which yielded a 6.4 % increase in precision.

Conversely, the SSD has proven effective in a one-stage detection framework. Yan et al. [24] utilized SSD enhanced with variational convolution on the VGG16 backbone to detect highway pavement cracks, achieving a 3.1 % improvement in average precision mean on the Pascal VOC2007 dataset. Furthermore, Feng et al. [25] developed a fusion model combining SSD with U-Net for more efficient crack detection, exceeding an 85 % precision rate. Moreover, Han et al. [26] and Ha et al. [27] integrated MobileNet into SSD, enhancing detection capabilities and showing increases in precision rates and average mean precision, respectively.

Despite these advancements, both one-stage and two-stage detection methods mentioned above exhibit limitations when applied to pavement crack detection, necessitating further research to optimize accuracy and efficiency in diverse environments. This continued innovation is critical for improving the maintenance and longevity of transportation infrastructure globally.

Nevertheless, the YOLOv3 and YOLOv5 series have garnered significant interest within the transportation sector for their robust performance in detecting pavement and infrastructure defects [28–34]. Wang et al. [35] advanced YOLOv3 by integrating data augmentation and altering the network structure to accurately assess pavement cracks. This adaptation included replacing the original DarkNet-53 backbone with ResNet101 and shifting from Distance Intersection over Union (DIoU) to Complete Intersection over Union (CIoU), achieving an average precision of 89.3 % and an F1 score of 86.5 % in crack detection. Similarly, Zhang et al. [36] utilized YOLOv3 to detect surface defects on concrete bridges, enhancing detection by incorporating SENet and SPP modules, which improved the average precision by 5.5 %. Further explorations into YOLOv5 variants - YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x - revealed that while YOLOv5s offered the fastest detection speeds, YOLOv5l provided the highest precision at 88.1 %, indicating nuanced performance differences within the series. Subsequent enhancements, as developed by Hu et al. [37] and Yu et al. [38], which included integrating transformer technologies and optimizing module configurations, further demonstrated the adaptability and effectiveness of the YOLOv5 models in crack detection. Notably, Guo et al. [39] and Roy et al. [40] achieved substantial gains in precision and efficiency by tweaking YOLOv5s's backbone and integrating advanced algorithms like DenseNet and Swin Transformer, underscoring the potential of these modifications in real-world applications. The utilization of the YOLOv5s algorithm, particularly its adaptations and enhancements, underscores the necessity for high precision and rapid processing in the challenging domain of pavement crack detection [41–44]. As demonstrated, the continual evolution and refinement of the YOLO series, especially YOLOv5s, cater effectively to the diverse and demanding requirements of modern road maintenance.

Renowned as the fastest training algorithm in the YOLO series, the YOLOv5s algorithm has garnered attention from several researchers [45–48] for its efficiency. This article explores the automation of pavement crack identification using an enhanced version of the YOLOv5s algorithm, emphasizing practical applications and technological advancements. The article is structured as follows: first, the technological foundations relevant to this study, such as the YOLOv5s algorithm, attention mechanisms, K-means clustering, SimSPPF, and transposed convolution are discussed. Subsequently, the preparatory steps of the experiments are outlined, followed by the presentation of evaluation metrics, and discussion on the experimental outcomes. Finally, the concluding section synthesizes the findings, underscoring the contributions of this research. According to the structure of the above article, it can be seen that the improved YOLOv5 model has added an attention mechanism to the feature extraction part, improved the sampling method in the feature enhancement structure, enhanced the feature extraction ability, and adopted SimSPPF, which not only improves the accuracy of crack feature extraction but also reduces the training accuracy of the model. Indeed, the innovation of this study lies in its methodological enhancements and the integration of advanced processing techniques, which significantly improve the detection accuracy and operational efficiency of pavement crack identification systems. By refining the YOLOv5s framework and incorporating novel computational methods, this paper not only advances the technological landscape but also has the potential to influence future developments in infrastructure maintenance.

2. Principle of experimental method

2.1. YOLOv5 networks

The YOLOv5 series, including YOLOv5m, YOLOv5l, and YOLOv5x, comprises three structural components: Backbone, Neck, and YOLOHead. The distinctions between these models are primarily defined by two parameters in the yolov5.yaml file: **depth_multiple** and **width_multiple**, which control the number of submodules and convolutional kernels, respectively. In the YOLOv5 version 6.0, a 6 × 6 convolutional layer replaces the Focus network in the main structure, offering equivalent functionality but with enhanced efficiency on current GPU devices. Among these variants, YOLOv5x demonstrates the highest mean average precision (mAP) when trained on public datasets, whereas YOLOv5s is noted for its rapid training capability. To balance accuracy with training efficiency, we have opted for the YOLOv5s-v6.0 network.

2.1.1. Skeleton feature extraction structure CSPDarknet53

The backbone network of YOLOv5s-v6.0 consists of five Conv_BN_SiLU layers, four CSPLayer modules, and one SPPF. Each Conv_BN_SiLU layer incorporates three key processes: convolution, batch normalization, and activation function processing. The activation function used, SiLU (Sigmoid Linear Unit), represents an enhancement over traditional Sigmoid and ReLU functions. It operates by constraining the scale of the predicted offset between -0.5 and 1.5, which ensures more accurate confinement of these values within the 0 to 1 range.

Fig. 1 illustrates the CSPLayer (C3) structure, which splits the original residual block into left and right segments. One segment allows the continuation of stacking original residual blocks, while the other connects directly to the end with minimal processing. This configuration enhances the layering of image features. SPPF, an acronym for Spatial Pyramid Pooling-Fast, accelerates the traditional Spatial Pyramid Pooling (SPP) method. SPP simultaneously applies three differently sized max-pooling layers to the input and merges the outcomes. This technique mitigates issues associated with multi-scale targets to some degree. Conversely, SPPF sequentially applies three 5 × 5 max-pooling layers and integrates the results successively. This procedure is depicted in Fig. 2.

2.1.2. Neck structure

In the YOLOv5s network, the Neck structure incorporates a Feature Pyramid Network (FPN) to augment feature extraction capabilities [49]. This system processes images of pavement cracks through the backbone feature extraction network, known as CSPDarknet53. After initial feature extraction, three distinct feature layers are derived from specific levels within the network: the middle, middle-lower, and final layers. For an image with dimensions 640 × 640 × 3, the corresponding shapes of these feature layers are designated as feat1= (80,80,256), feat2= (40,40,512), and feat3=(20,20,1024).

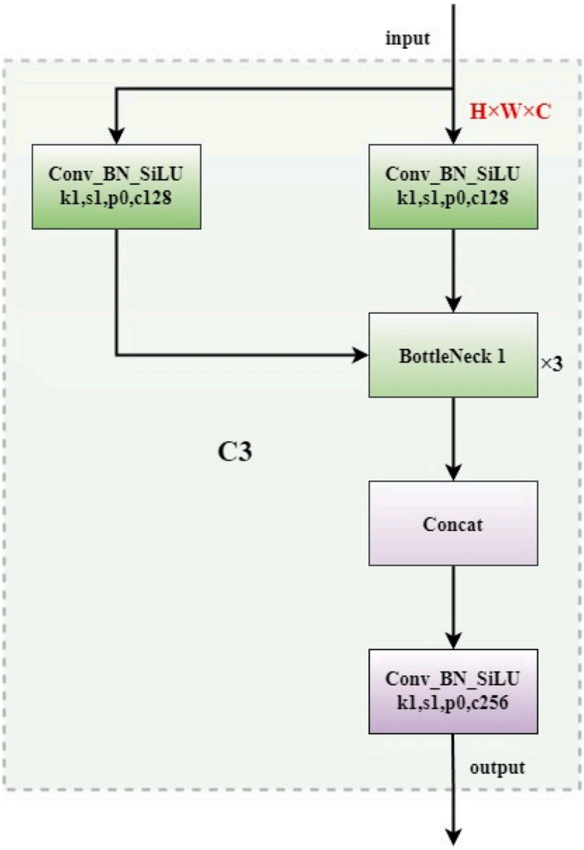


Fig. 1. CSPlayer structure.

According to the Feature Pyramid Network design, the output from the last layer is refined through a 1 × 1 convolution for channel adjustment, producing P5. This layer, P5, is then upsampled and merged with the feat2 layer (40,40,512) for further processing. The combined features undergo feature extraction in the CSPLayer network, yielding P5_Upsampling=(40,40,512). This upsampled layer, P5_Upsampling, is subjected to another convolution for channel adjustment, resulting in P4. Subsequently, P4 is upsampled and integrated with the middle layer, feat1=(80,80,256). After additional processing through the CSPLayer network, the final output, P3_out=(80,80,256), is produced.

The feature layer P3_out is processed with a 3 × 3 convolution and then downsampled. Following this, it is combined with P4, and the merged layers undergo feature extraction in the CSPLayer network, resulting in P4_out=(40,40,512). Similarly, the feature layer P4_out is treated with a 3 × 3 convolution, downsampled, and then merged with P5. This combination is further processed through the CSPLayer network, producing P5_out=(20,20,1024). The layers P3_out, P4_out, and P5_out, derived from the Neck structure, represent feature layers that are fused across multiple iterations, enhancing the detection capabilities for a wider array of pavement crack features.

2.1.3. *The YOLOHead structure*

The YOLOHead structure functions by processing three feature layers - namely P3_out, P4_out, and P5_out - that are outputted from the Neck structure. This structure primarily consists of a 3 × 3 convolution followed by a 1 × 1 convolution. The 3 × 3 convolution serves to integrate features from the output feature layers of the Neck structure, whereas the 1 × 1 convolution is utilized to refine the channel count of the input feature layers. After being processed through the YOLOHead structure, these feature layers collectively exhibit a channel count of 27.

$$27 = (4 + 4 + 4) \times 3 \tag{1}$$

The value 27 results from a configuration where each grid cell contains three anchor boxes for identification purposes. Each anchor box comprises four adjustment parameters, and one of these adjustments is specifically dedicated to identifying pavement cracks, covering four distinct types as outlined in the RDD2022 Pavement Cracks Dataset. Consequently, the dimensions of the prediction outputs are (80,80,27), (40,40,27), and (20,20,27). The detailed architecture of the YOLOv5s network is depicted in Fig. 3.

2.1.4. *Decoding layer*

The YOLOv5s network encodes and outputs predictions for three feature layers, characterized by the shapes (N,80,80,27), (N,40,40,27), and (N,20,20,27). These encoded shapes, however, do not directly correspond to the final positions of the predicted boxes on the images. Thus, a decoding process is required to accurately determine the ultimate positions of these predicted boxes. After decoding, the shapes are transformed to (N,80,80,3,9), (N,40,40,3,9), and (N,20,20,3,9). The numbers 80, 40, and 20 represent the division of the pavement cracks image into grids of *80 × 80*, *40 × 40*, and *20 × 20* feature points, respectively. At this resolution, if a specific feature point aligns with a target object's corresponding box, it is utilized to predict the target pavement crack.

2.2. *Improving the YOLOv5s network*

This article introduces an advanced YOLOv5s network tailored for automated pavement crack detection. Enhancements to the network are detailed as follows:

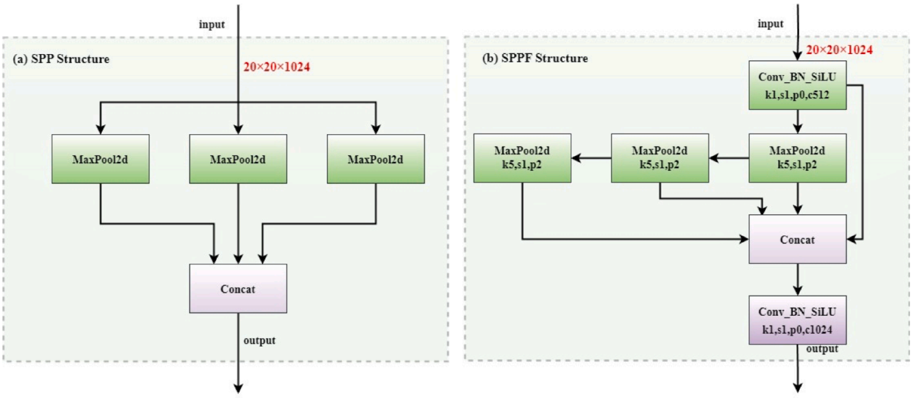


Fig. 2. SPP and SPPF structure.

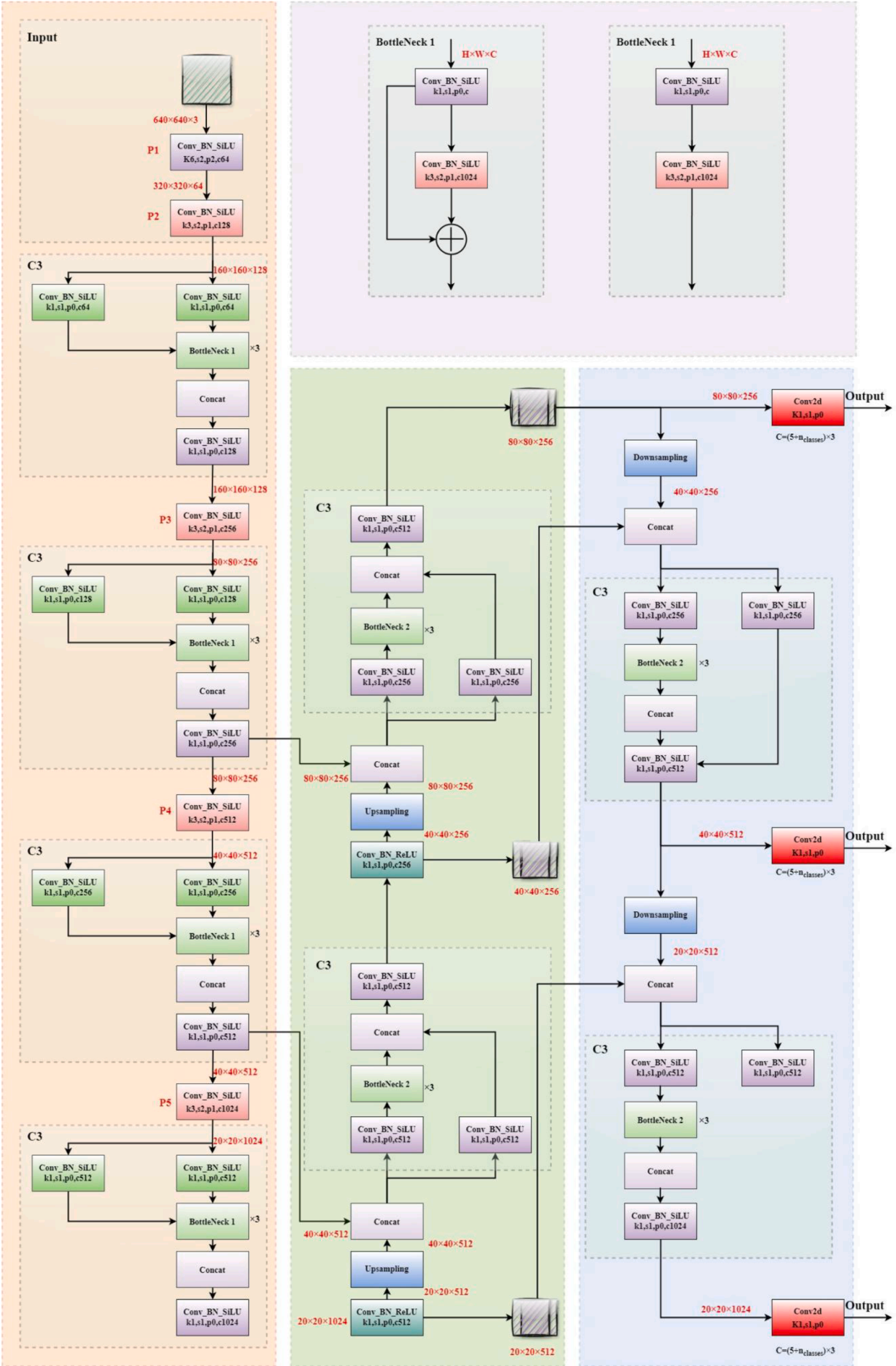


Fig. 3. YOLOv5s-6.0 network structure.

1) **Anchor optimization**: Initially, the YOLOv5s network employs a predefined anchor configuration, which achieves a 99.3 % fit for targeted cracks, potentially limiting the recall rate. To address this, the network recalculates anchors to ensure a 99.9 % fit, thus boosting the recall rate. The recalibrated anchors are [27, 48, 170, 31, 38, 157], [120, 90, 82, 219, 53, 581], and [465, 73, 146, 459, 392, 311].
2) **Integration of SENet modules**: The network integrates SENet channel attention mechanisms following each CSPLayer and SPPF

layer within the backbone, enhancing the capacity of the network to autonomously learn crack features at the channel level. This addition increases the total layer count in the YOLOv5s architecture from 157 to 191.

3) **Efficiency in layer processing**: To counterbalance the potential slowdown from increased layer count, the original SPPF layer is replaced with a SimSPPF layer, which aids in preserving training speed.
4) **Neck structure modification**: The Neck structure of the network is modified by replacing the nearest-neighbor interpolation method with transposed convolution for up-sampling. This change facilitates more effective learning of the optimal up-sampling technique for detecting target crack features.

These strategic enhancements are summarized in Table 1, outlining the specific improvements made to the YOLOv5s network's structure.

2.3. Method and principle of improving YOLOv5s network

2.3.1. Channel attention mechanism

Squeeze-and-Excitation Networks (SENet) constitute a type of channel attention mechanism [50]. This method facilitates autonomous learning of target features within the feature channels, emphasizing essential features associated with the target to improve image recognition accuracy. The functional principles of the SENet block are depicted in Fig. 4.

Here, (H), (W), and (C) denote the height, width, and number of channels of the input, respectively.

The computational formulas for each symbol in the SENet block are as follows:

$$F_{tr} : X \rightarrow U \tag{2}$$

where $\mathrm{F_{tr}}$ is a standard convolution operator and U is calculated by inputting X. besides, there is $X \in R^{w' \times H' \times C'}$ and $U \in R^{w \times H \times C}$ as well as some subsequent calculation steps as follows:

$$U = [u_1, u_2, \cdots, u_C] \tag{3}$$

$$v_C = [v_C^1, v_C^2, \cdots, v_C^C] \tag{4}$$

$$V = [v_1, v_2, \cdots, v_C] \tag{5}$$

$$u_C = v_C \times X = \sum_{s=1}^{C'} v_C^s \times x^s \tag{6}$$

where Equation (5) is the set of filters, and the filter is a three-dimensional matrix composed of multiple convolution kernels v_C^s. Moreover, the extra dimension is channel C, and * in Equation (6) represents convolution.

In Fig. 4, the SENet calculation block consists of Global Average Pooling (GAP) as well as activation function F_{ex} and F_{scale}.

$$z_C = F_{sq}(u_C) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_C(i,j) \tag{7}$$

$$sigmoid = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \sigma(W_1 z)) \tag{8}$$

$$\widetilde{X_C} = F_{scale}(u_C, s_C) = s_C u_C \tag{9}$$

$$\widetilde{X} = [\widetilde{x_1}, \widetilde{x_2}, \cdots, \widetilde{x_C}] \tag{10}$$

where σ is ReLU function calculation and $W_1 \in R^{\frac{C}{r} \times c}$, $W_1 \in R^{C \times \frac{C}{r}}$.

The operational mechanism of SENet, based on the aforementioned computational formulations, is delineated as follows: Initially, two-dimensional features $H \times W$ from each channel's output are

Table 1
The structure of improved YOLOv5s network.

No.	From[1]	n[2]	Params[3]	Module[4]	Arguments[5]
0	−1	1	3520	models.common. Conv	[3,32,6,2,2]
1	−1	1	18560	models.common. Conv	[32, 64, 3, 2]
2	−1	1	18816	models.common. C3	[64, 64, 1]
3	−1	1	512	models.common. SE	[64, 64]
4	−1	1	73984	models.common. Conv	[64, 128, 3, 2]
5	−1	2	115712	models.common. C3	[128, 128, 2]
6	−1	1	2048	models.common. SE	[128, 128]
7	−1	1	295424	models.common. Conv	[128, 256, 3, 2]
8	−1	3	625152	models.common. C3	[256, 256, 3]
9	−1	1	8192	models.common. SE	[256, 256]
10	−1	1	1180672	models.common. Conv	[256, 512, 3, 2]
11	−1	1	1182720	models.common. C3	[512, 512, 1]
12	−1	1	32768	models.common. SE	[512, 512]
13	−1	1	656896	models.common. SimSPPF	[512, 512, 5]
14	−1	1	32768	models.common. SE	[512, 512]
15	−1	1	131584	models.common. Conv	[512, 256, 1, 1]
16	−1	1	4352	torch.nn.modules. conv. ConvTranspose2d	[256, 256, 4, 2, 1, 0, 256]
17	[-1,8]	1	0	models.common. Concat	[1]
18	−1	1	361984	models.common. C3	[512, 256, 1, False]
19	−1	1	33024	models.common. Conv	[256, 128, 1, 1]
20	−1	1	2176	torch.nn.modules. conv. ConvTranspose2d	[128, 128, 4, 2, 1, 0, 128]
21	[-1,6]	1	0	models.common. Concat	[1]
22	−1	1	90880	models.common. C3	[256, 128, 1, False]
23	−1	1	147712	models.common. Conv	[128, 128, 3, 2]
24	[-1,19]	1	0	models.common. Concat	[1]
25	−1	1	296448	models.common. C3	[256, 256, 1, False]
26	−1	1	590336	models.common. Conv	[256, 256, 3, 2]
27	[-1,15]	1	0	models.common. Concat	[1]
28	−1	1	1182720	models.common. C3	[512, 512, 1, False]
29	[22,25,28]	1	24273	models.yolo. Detect	[4, [[27, 48, 170, 31, 38, 157], [120, 90, 82, 219, 53, 581], [465, 73, 146, 459, 392, 311]], [128, 256, 512]]

Note: From[1] denotes the origin layer and −1 indicates the preceding layer. n[2] corresponds to the repetition count of the module, and typically set to 1. Params[3] represents the quantity of parameters, while Module[4] denotes the module's name. Arguments[5] indicates the input and output channel quantity, convolutional kernel, and stride.

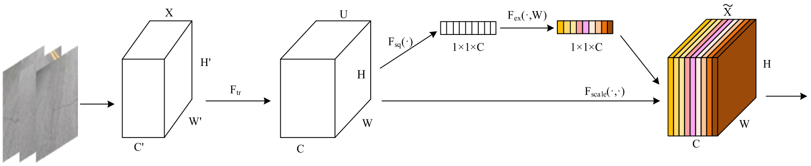


Fig. 4. SENet Block.

compressed into a real number through the application of algorithm F_{sq}, resulting in the compression of the feature map from $H \times W \times C$ to $1 \times 1 \times C$. In the second step, a sequence of operations, including Fully Connected layers (FC), ReLU activation functions, and another set of FC layers, is systematically employed to acquire a set of weights. These weights are subsequently allocated to individual channels, and the resulting values, constrained within the range of 0 to 1 by a sigmoid function, contribute to the algorithmically derived output $1 \times 1 \times C$, which is further processed through operation F_{ex} to yield $1 \times 1 \times C$. The final step involves adding the normalized weights $1 \times 1 \times C$ obtained from the previous two steps to the feature maps of each channel $H \times W \times C$. The algorithm employed for this addition is matrix multiplication, denoted as $[H,W,C]*[1,1,C]$, resulting in the output $\widetilde{X} = [H,W,C]$.

2.3.2. K-means clustering algorithm

In the YOLOv5s network, the computation of anchors is achieved through the implementation of the K-means clustering algorithm, which falls within the domain of unsupervised learning. Tailored to the specific input target dataset, an initial set of K cluster centers is randomly selected. Utilizing the Euclidean distance, the distances from these initial cluster centers C_i ($i \leq 1 \leq K$) to other data objects C_i are computed. The cluster center C_i closest to a given data object is identified, and the data object is subsequently assigned to the corresponding cluster (where similar objects are grouped together). Following this, the average of the data objects within each cluster is computed and designated as the new cluster center. This iterative process continues until the calculated cluster centers cease to change or reach the maximum iteration limit. The Euclidean distance is computed using the following formula:

$$d(X, C_i) = \sqrt{\sum_{j=1}^{m} (X_j - C_{ij})^2} \tag{11}$$

where X represents the data object, C_i is denoting the i-th cluster center and m is the dimension of the data object.

2.3.3. SimSPPF

SimSPPF, utilized in YOLOv6, represents a Simplified Spatial Pyramid Pooling – Fast adaptation of the traditional SPPF. This streamlined version integrates into the core feature extraction network and simplifies the original structure by replacing the SiLU activation function with ReLU. This substitution enhances the processing speed, making Conv_BN_ReLU operations faster than their Conv_BN_SiLU counterparts. The architecture of SimSPPF is depicted in Fig. 5.

2.3.4. Transposed convolution

The Neck structure employs up-sampling to restore the image to its original size, facilitating the transition from lower to higher resolutions and subsequently extracting features, as illustrated in Fig. 6. Common up-sampling techniques include nearest neighbor interpolation, bilinear interpolation, and transposed convolution. In this study, the original nearest neighbor interpolation method in the YOLOv5s network is replaced with transposed convolution. This form of convolution, distinct from predetermined up-sampling methods, enables the network to autonomously learn the characteristics of data objects and select the most effective up-sampling technique. Transposed convolution essentially inverts the relationship between input and output. In ordinary convolution, if an input feature layer is $H \times W = 4 \times 4$ with a 3×3 convolution kernel, no padding, and a stride of 1, the resulting output feature layer will be $H \times W = 2 \times 2$. Conversely, with transposed convolution - using the same 3×3 kernel, zero padding, and a stride of 1 - but applied to an input feature layer of $H \times W = 2 \times 2$, the output feature layer expands to $H \times W = 4 \times 4$.

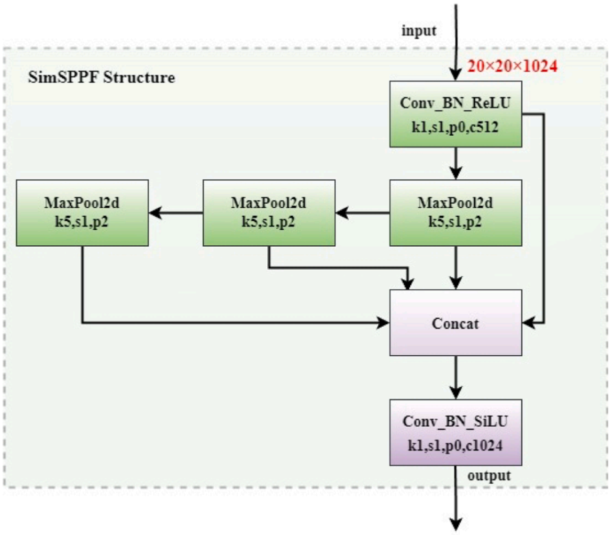


Fig. 5. SimSPPF structure.

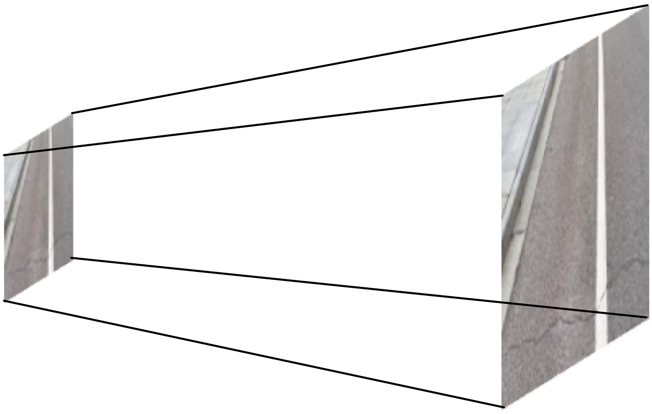


Fig. 6. Sample Up-sampling.

3. Research on road crack detection based on improved YOLOv5s

In this paper, the enhanced YOLOv5s network is utilized for automatic road crack detection. The operational steps of the network are as follows: First, the prepared crack dataset is fed into the CSPDarknet53 structure, where features of crack images are extracted using convolution, CSPLayer, the SENet channel attention mechanism, and SimSPPF, resulting in three feature layers: feat1, feat2, and feat3. The second step involves processing these layers within the Neck structure, which employs the Feature Pyramid Network concept. Here, features are augmented by up-sampling usingthe CSPLayer from bottom to top, then refined by down-sampling from top to bottom to produce three output layers: P3_out, P4_out, and P5_out. These layers are subsequently processed in the YOLOHead structure, where they undergo a *3* × *3* convolution and a *1* × *1* convolution to finalize the detection process. The detailed workflow of the improved YOLOv5s network is depicted in Fig. 7.

4. Experimental preparations

4.1. Data acquisition

Target detection requires a substantial dataset of road crack images. This study utilizes the public RDD2022 dataset [51], which comprises over 40,000 images depicting road conditions from China, the Czech Republic, India, Japan, Norway, and the United States. Images are initially categorized into eight types using labelImg: longitudinal cracks (D00), transverse cracks (D10), alligator cracks (D20), potholes cracks (D40), and four additional unspecified types. Given that the prevalent road issues are longitudinal cracks, transverse cracks, alligator cracks and potholes, the dataset is refined by removing the labels for the lesser-relevant categories, retaining only D00, D10, D20, and D40. Additionally, the data format is converted from XML to TXT for processing efficiency. For the purposes of this research, the dataset from China is specifically selected for training. The dataset of road cracks within the China region was compiled by mounting a camera on a motorcycle traveling at a speed of 30 km/h, capturing a total of approximately 2,500 images, each with a resolution of 512 × 512. For the purposes of this study, the dataset is divided into subsets: 500 images constitute the test set, and over 1,900 images are allocated between the training and validation sets, maintaining a training-to-validation ratio of 8:2, as detailed in Table 2. Some samples of the data set are shown in Fig. 8.

4.2. Equipment and software preparation

The specifications of computer hardware equipment used to realize the functions of building a preliminary improved YOLOv5s network algorithm, data set processing algorithm, training model, evaluation model and prediction results are shown in Table 3.

The experiment uses Python language, the deep learning framework is Pytorch, the environment needed for training is configured in Anaconda, Python programs are written with VSCode editor, and the network is trained in Anaconda software. The software versions and languages used this time are shown in Table 4.

4.3. Experimental parameter setting

For this experiment, due to the use of animation frames when labeling cracks, the network is trained to identify the most suitable label frame for cracks during prediction. Before commencing the experiment, it is essential to specify several parameters: the input image size (**imgsz**), the number of crack images per training batch (**batch_size**), and the number of epochs for a single training cycle. Additionally, the Intersection over Union (IoU) training threshold (**IoU_t**) for Non-Maximum Suppression (NMS) must be set. The deep learning framework employs the SGD optimizer, for which the initial learning rate (**lr0**), momentum, and the optimizer's weight decay (**weight_decay**) are specified. Proper calibration of these hyperparameters is crucial to align the training outcomes closely with the actual labels. Table 5 details the specific values for these hyperparameters.

5. Evaluation indicators, results, and discussion

5.1. Evaluation indicators

To demonstrate the robustness of the enhanced YOLOv5s network in pavement crack detection, three key performance metrics are employed: the comprehensive F1 score, mAP, and fps. These indicators are used to assess and validate the effectiveness of the proposed method.

5.1.1. Comprehensive evaluation index F1

The comprehensive F1 score is utilized to balance precision and recall, addressing the challenges of a binary classification model. Additionally, the Confusion Matrix is employed to further define these metrics. 'Positive/Negative' refers to the predicted outcome being either positive or negative, while 'True/False' indicates whether the prediction was accurate or not. Thus, 'TP' denotes a true positive case, 'FP' represents a false positive, and so forth.

Precision is calculated as the ratio of TP to the total number of samples classified as positive by the classifier. This total includes both TP and FP. Thus, the formula for precision is:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

Recall is calculated as the ratio of TP to the total number of actual positive samples. This total includes both TP and FN. Therefore, the formula for recall is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

F value is the harmonic average value between accuracy rate and recall rate, which is expressed by $F_{\beta\text{ - score}}$, and the formula is:

$$F_{\beta\text{ - score}} = \frac{(1+\beta^2)*P*R}{\beta^2*P+R} \tag{14}$$

where β is used to balance the weights of Precision and Recall in the calculation of $F_{\beta\text{ - score}}$, and there are three values as follows:

(1) When $\beta < 1$, the Precision is more important in the training results,
(2) When $\beta > 1$, the Recall is more important in the training results,
(3) When $\beta = 1$, both Precision and Recall should be paid attention to in the training results, and $F_{1\text{ - score}}$ is the common comprehensive evaluation index F1, calculated as follows:

$$F_{1\text{ - score}} = \frac{2*P*R}{P+R} \tag{15}$$

5.1.2. Mean average precision (mAP)

The mAP is defined as the mean value of Average Precision (AP) scores across different categories. AP quantifies the area beneath the Precision-Recall curve, representing the relationship between precision and recall for various thresholds. Consequently, mAP is the average of these AP values, encompassing all Precision-Recall curves. A higher mAP value indicates superior training results. Typically, mAP is evaluated by comparing AP scores at an Intersection over Union (IoU) threshold of 0.5 during training. In Fig. 9, IoU is calculated as the ratio of the intersection of the actual and predicted boundaries of the target to the union of these boundaries.

5.1.3. Frames per second (fps)

The fps measures the refresh rate of images during model training, indicating the number of frames processed each second. This metric is critical for evaluating the model's efficiency during live detection scenarios, ensuring that the system can operate in real-time applications without lag. It is calculated as follows:

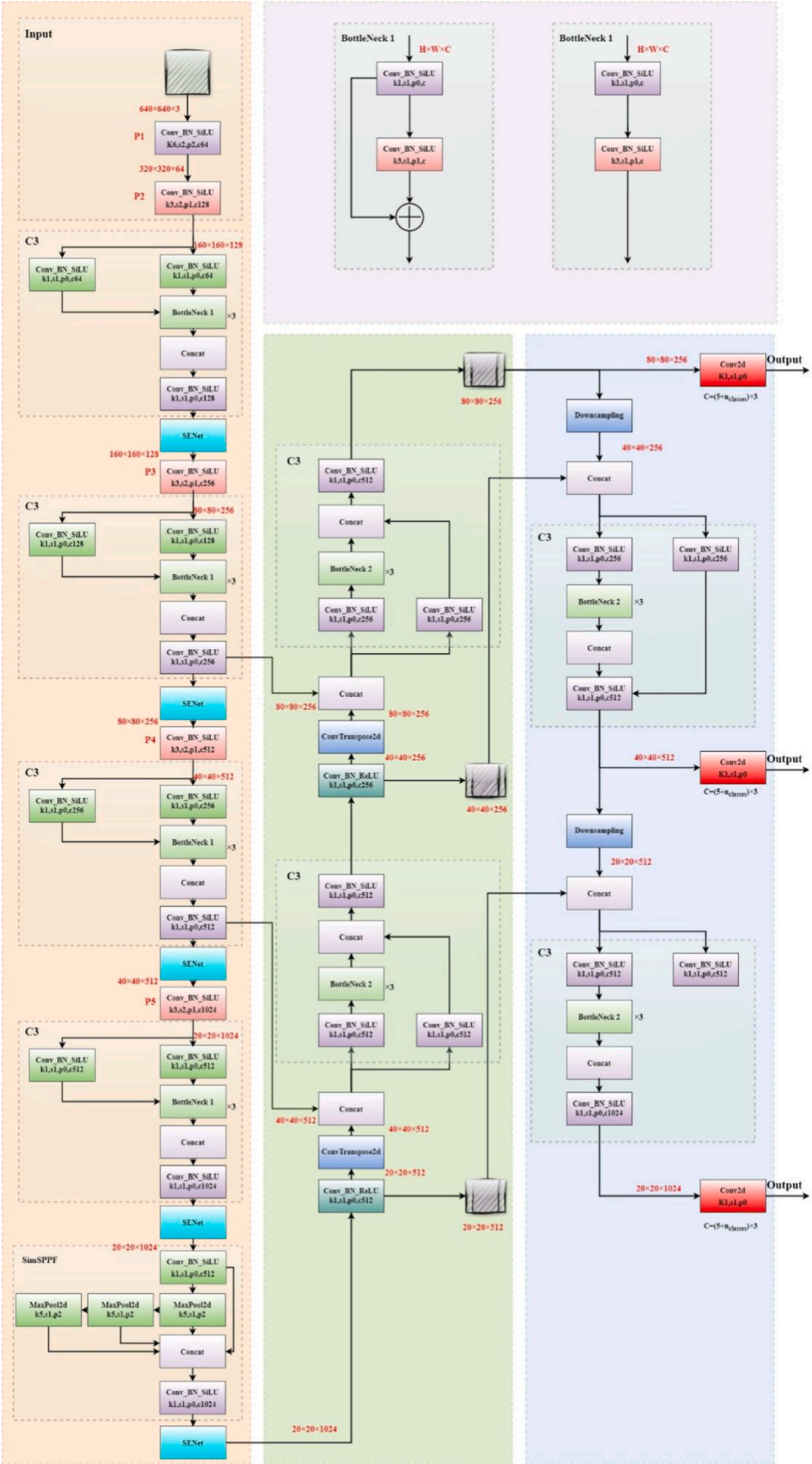


Fig. 7. Improved YOLOv5s network flowchart.

Table 2
Dataset division.

Dataset	Quantity
Training	1547
Validation	387
Testing	500

$$fps = \frac{1000ms}{(pre_{process} + \text{inference} + NMS)ms} \tag{16}$$

5.2. Results and discussion

The enhanced YOLOv5s network, alongside other YOLO variants and models like YOLOv3, YOLOv7, and YOLOv8, underwent rigorous comparative testing to determine the most effective model for fracture training. These models were evaluated for their ability to detect various types of pavement cracks under different conditions.

5.2.1. Results of model train

Initially, the original YOLOv5s model was employed to train the fracture dataset. Subsequent to the initial training, optimal hyperparameters were selected based on the results, enabling the model to achieve an approximate recognition accuracy of 90 %. Table 6 illustrates the training outcomes for the original YOLOv5s network. Analysis of these results indicates that the detection of linear cracks is less effective compared to other crack types and potholes. Given the challenges in identifying minute features in linear cracks and various orientations of cracks, repeated training is essential to refine the network's ability to recognize these features accurately. To enhance the detection of linear cracks specifically, four enhanced versions of the YOLOv5s network have been developed, building upon the original network structure.

The modification to the YOLOv5s_1 model involves incorporating the Convolutional Block Attention Module (CBAM) into the C3 structure of the original YOLOv5s backbone feature extraction network and adding the channel attention mechanism SENet between the ninth and tenth layers. This integration facilitates autonomous learning of target features in both channel and spatial dimensions. However, the mere addition of convolution and channel attention mechanisms has not enhanced

Table 3
The specification of the computer.

Indicator	Value
GPU	NVIDIA GeForce RTX 2080 Ti
CPU	Intel(R) Xeon(R) Platinum 8260
CUDA	12.0
Cudnn	8.2.2

Table 4
Software and language settings.

Indicator	Value
Anacond	2019.10
VSCode	2017
Python	3.8.5
Torch	1.8.0

Table 5
Hyperparameter value settings.

Hyperparameter	Value
Imgsz	640 × 640
Batch_size	16
IoU_t	0.20
Optimizer	SGD
lr0	0.01
Momentum	0.937
Weight_decay	0.0005

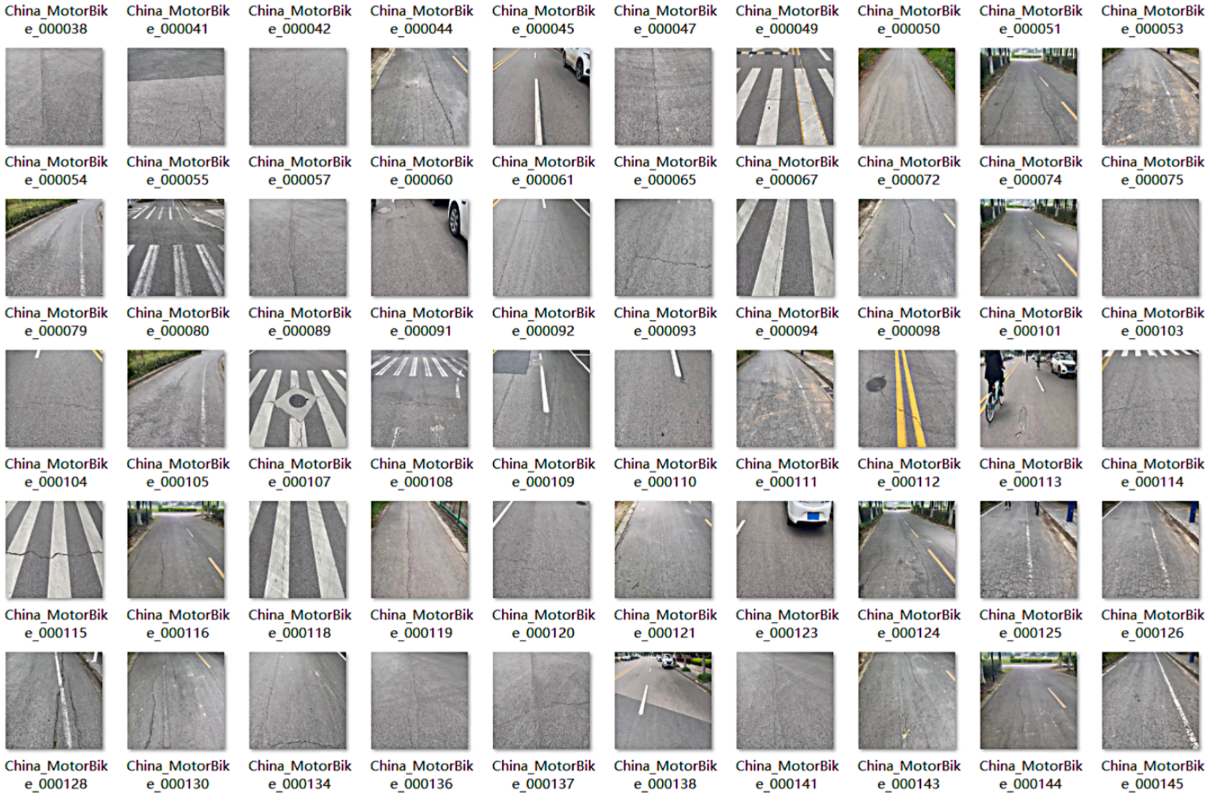


Fig. 8. Partial RDD2022 Dataset Samples.

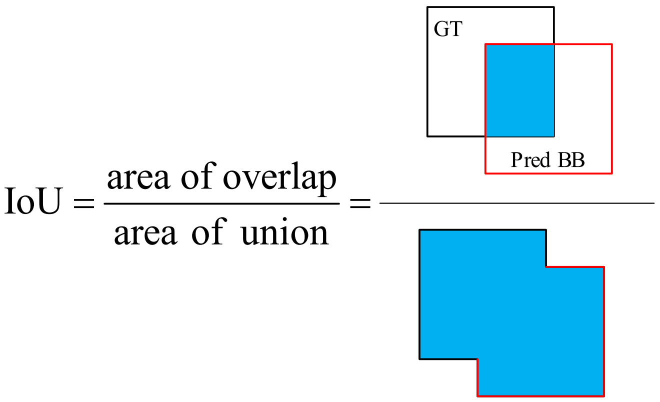


Fig. 9. Schematic diagram of IoU calculation.

Table 6
The training results for YOLOv5s model.

Evaluation	Precision	Recall	F1	mAP (0.5)
all	0.909	0.888	0.898	0.932
D00	0.886	0.824	0.867	0.887
D10	0.882	0.847	0.864	0.919
D20	0.917	0.902	0.909	0.936
D40	0.952	0.980	0.966	0.986

the recognition of linear cracks. The training outcomes for the YOLOv5s_1 model are provided in Table 7.

On the basis of YOLOv5s_1 model, YOLOv5s_2 model replaces the up-sampling mode in Neck structure with transposed convolution to realize feature enhancement extraction. The training results for the YOLOv5s_2 model are listed in Table 8.

The F1 value of YOLOv5s_2 model in identifying all cracks has increased, but it is still lacking in identifying linear cracks. Therefore, based on YOLOv5s_2 network structure, the channel attention mechanism SENet is added, and the SPPF layer is modified to SimSPPF, which is YOLOv5s_3. After the channel attention mechanism was added to each layer of the modified C3CBAM, the network training time became longer due to the addition of four layers of SENet modules, and finally the SPPF layer was modified to SimSPPF. The training results are shown in Table 9.

After evaluating the training outcomes of the four models, modifications were made to the original YOLOv5s network. Initially, the channel attention mechanism SENet was integrated exclusively following the C3 structure within each layer of the backbone feature extraction network. Subsequently, the C3 structure itself remained unaltered, while the SPPF and Neck structures were replaced with SimSPPF and transposed convolution, respectively. Post-training evaluations revealed that the overall recognition performance of this revised model surpasses that of other models in the YOLO series, particularly in detecting linear fractures. Consequently, this model was ultimately chosen for training fracture datasets. The results are presented in Table 10.

Table 7
The training results for YOLOv5s_1 model.

Evaluation	Precision	Recall	F1	mAP (0.5)
all	0.905	0.879	0.892	0.921
D00	0.878	0.814	0.845	0.878
D10	0.876	0.823	0.849	0.882
D20	0.907	0.898	0.902	0.933
D40	0.959	0.980	0.970	0.990

Table 8
The training results for YOLOv5s_2 model.

Evaluation	Precision	Recall	F1	mAP (0.5)
all	0.903	0.909	0.906	0.932
D00	0.851	0.856	0.853	0.879
D10	0.887	0.879	0.883	0.925
D20	0.895	0.918	0.906	0.941
D40	0.980	0.981	0.980	0.985

Table 9
The training results for YOLOv5s_3 model.

Evaluation	Precision	Recall	F1	mAP (0.5)
all	0.912	0.899	0.905	0.930
D00	0.856	0.833	0.844	0.880
D10	0.918	0.884	0.901	0.926
D20	0.921	0.918	0.919	0.946
D40	0.954	0.961	0.984	0.969

Table 10
The training results for the proposed model.

Evaluation	Precision	Recall	F1	mAP (0.5)
all	0.905	0.916	0.910	0.936
D00	0.864	0.856	0.860	0.885
D10	0.887	0.902	0.894	0.931
D20	0.919	0.946	0.932	0.946
D40	0.951	0.961	0.956	0.981

5.2.2. Model comparison

The YOLO series was utilized to train models for crack detection, comparing the performance of YOLOv3, YOLOv5s, four enhanced versions of YOLOv5s, YOLOv7, and YOLOv8. Fig. 10 illustrates the performance comparison results across these models. Although the YOLOv8 network achieves the highest mean Average Precision (mAP) of 0.5, its combination of low precision and high recall leads to a relatively high number of false positives. The model ultimately selected offers higher F1 scores and mAP values than the other models. Despite YOLOv8′s marginally superior mAP values, its precision, recall, and F1 scores are comparatively lower. Therefore, the enhanced YOLOv5s model exhibits the best overall performance.

The model is not only compared from the recognition effect, but also evaluated from the training speed. YOLOv3 and YOLOv7 networks is not discussed due to their poor and the training speed. The fps calculation results of various models are shown in Fig. 11. The improved YOLOv5s network has the largest fps value, that is, the fastest training speed such

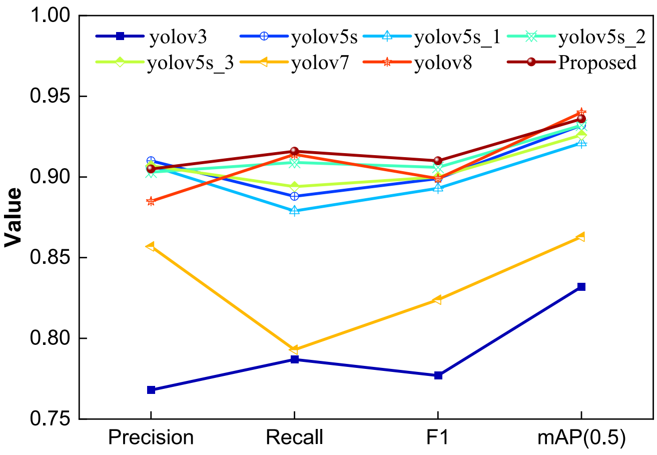


Fig. 10. Models evaluation comparison.

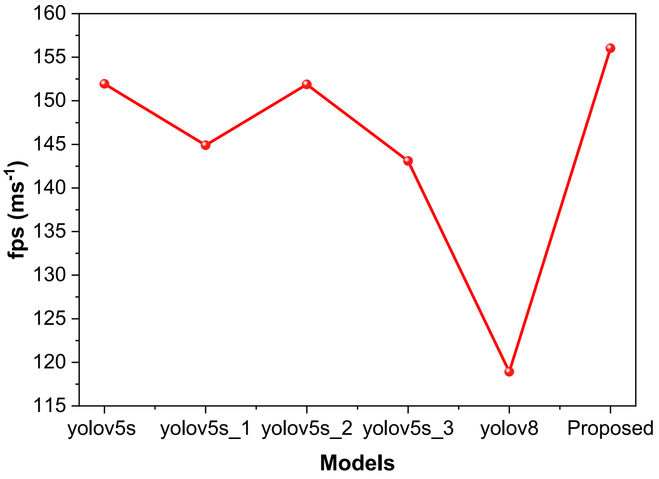


Fig. 11. Time comparison.

as it takes about two and a half hours to train 500epoch.

5.2.3. Ablation experiment

To demonstrate the benefits of the newly added optimization module in the YOLOv5s algorithm, seven ablation experiments were performed, as detailed in Table 11. The mean mAP, GFLOPs, and parameters of the original YOLOv5s algorithm are 93.2 %, 15.8, and 7,020,913, respectively. In the enhanced model, the mAP increased from 93.2 % to 93.6 %; likewise, GFLOPs and the number of layers were also augmented, although the parameters remained relatively consistent with the original model. This enhancement in the model has improved both accuracy and training speed, aligning well with the operational demands of daily pavement crack detection.

5.2.4. Prediction results of pavement cracks

The enhanced model has been rigorously tested in a variety of environmental conditions to detect pavement cracks, consistently proving its robustness and reliability. The results from these extensive field tests, vividly illustrated in Fig. 12, validate the model's exceptional ability to accurately identify and classify different types of pavement cracks. Notably, the figure demonstrates the model's precision in distinguishing longitudinal cracks (D00), transverse cracks (D10), and alligator cracks (D20), each marked with high confidence scores that attest to the model's accuracy. Moreover, the effectiveness of the model is further highlighted by its performance under varying lighting conditions and on different pavement materials, showcasing its adaptability and precision. The detailed annotations in the figure capture the nuanced differences between the crack types, with clear demarcations that facilitate easy identification and assessment. This capability is crucial for the timely and effective maintenance of road infrastructure, potentially reducing repair costs and increasing road safety.

Furthermore, the model not only identifies the type of crack but also estimates the severity and dimensions, which are critical for maintenance prioritization and planning. Such comprehensive analysis capabilities of the model ensure that road maintenance professionals can efficiently plan interventions and allocate resources effectively. This combination of high accuracy, reliability, and detailed analytical output makes the enhanced model an indispensable tool in the domain of pavement maintenance, underscoring its importance in contemporary road condition assessment strategies.

6. Conclusions

To enhance pavement crack detection, a novel neural network algorithm is presented in this article to automate crack identification. In the proposed technique, the Squeeze-and-Excitation Networks are integrated behind the C3 structure of the trunk feature extraction network, the SPPF layer is replaced with SimSPPF, and the up-sampling method in the Neck structure is upgraded to transposed convolution. The performance of the proposed model is compared with other models, including YOLOv3-tiny, YOLOv5s, YOLOv5s_1, YOLOv5s_2, YOLOv5s_3, and YOLOv7, in terms of speed and accuracy in crack identification, using comprehensive evaluation metrics such as F1 and mAP (0.5). The following conclusions can be drawn from this study.

1) The integration of an attention mechanism into the YOLOv5s model has markedly enhanced training efficiency. Previously, achieving 500 epochs with the original model required more time, but with the revised attention-enhanced model, this can now be accomplished in just 2.4 h. This improvement underscores the significant increase in processing speed, which is crucial for practical applications requiring rapid model training and deployment.
2) The proposed model has demonstrated high accuracy in detecting pavement cracks, achieving a detection accuracy of 90.5 %. Moreover, the model exhibits a recall rate of 91.6 %, an F1 score of 91 %, and a mAP of 93.6 %. These metrics are indicative of the model's reliability and precision in identifying and classifying various crack types in pavement, which are essential for effective maintenance and repair operations.
3) When compared to the original YOLOv5s network, the proposed model shows a noticeable improvement across several performance metrics. There is an increase of 0.7 % in the F1 score, 0.2 % in mAP, and 1.54 % in fps. These enhancements reflect the model's refined ability to process images more quickly and accurately, which is vital for real-time applications.
4) Looking ahead, the focus will be on further enhancing the YOLO series network. The goals include reducing memory usage and model weight, which are crucial for improving computational efficiency. Additionally, efforts will be made to accelerate detection speeds and ensure the model can be easily installed on small mobile devices. This development is aimed at enabling real-time detection capabilities, which are increasingly demanded in modern technological applications. In the future, a detection window that can automatically detect and identify road cracks will be designed, which can be embedded into mobile devices such as drones, mobile phones, and inspection vehicles at any time. This can solve the current problem of timeliness in managing and repairing road cracks.

Overall, this study provides an in-depth analysis of the experimental

Table 11
Ablation experiment.

Groups	Preprocess	K-means	C3CBAM	SENet	SimSPPF	TC	GFLOPs	Parameters	mAP(0.5)
1(yolov3-tiny)	×	×	×	×	×	×	12.9	8673622	0.832
2(yolov5s)	√	×	×	×	×	×	15.8	7020913	0.932
3(yolov5s_1)	√	√	√	√	×	×	15.8	7069855	0.921
4(yolov5s_2)	√	√	√	×	×	√	15.9	7076383	0.932
5(yolov5s_3)	√	√	√	√	√	√	15.9	7120671	0.926
6(yolov7)	√	√	×	×	×	×	—	—	0.863
7(Proposed)	√	√	×	√	√	√	15.9	7104497	0.936

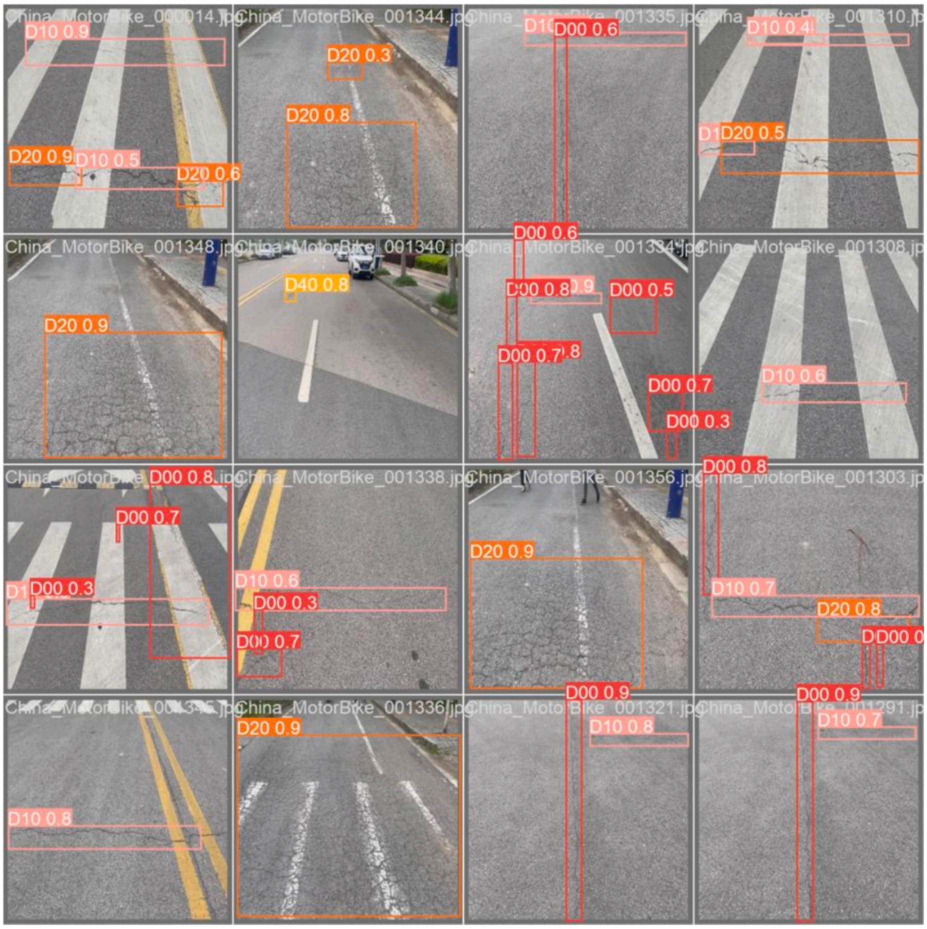

China_MotorBike_000014.jpg
D10 0.9
D20 0.9
D10 0.5
D20 0.6
China_MotorBike_001344.jpg
D20 0.3
D20 0.8
China_MotorBike_001335.jpg
D10
D00 0.6
China_MotorBike_001310.jpg
D10 0.4
D1
D20 0.5
China_MotorBike_001348.jpg
D20 0.9
China_MotorBike_001340.jpg
D40 0.8
China_MotorBike_001334.jpg
D00 0.6
D00 0.8
0.9
D00 0.5
D00 0.7
0.8
D00 0.7
D00 0.3
China_MotorBike_001308.jpg
D10 0.6
D00 0.8
D00 0.7
D1
D00 0.3
China_MotorBike_001338.jpg
D10 0.6
D00 0.3
D00 0.7
China_MotorBike_001356.jpg
D20 0.9
China_MotorBike_001303.jpg
D00 0.8
D10 0.7
D20 0.8
DD00 0
D00 0.9
D10 0.8
China_MotorBike_001336.jpg
D20 0.9
China_MotorBike_001321.jpg
D00 0.9
D10 0.8
China_MotorBike_001291.jpg
D00 0.9
D10 0.7

Fig. 12. The prediction results of pavement cracks.

outcomes and highlights significant advancements made to the YOLOv5s model. These improvements have notably enhanced the model's accuracy and processing speed, met the critical requirements of contemporary pavement maintenance practices and contributing to safer driving conditions. The progress documented here not only sets the stage for future innovations but also demonstrates the potential for further advancements in automated pavement monitoring systems.

CRediT authorship contribution statement

Shuangxi Zhou: Writing – original draft, Validation, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **Dan Yang:** Writing – original draft, Visualization, Investigation, Formal analysis, Data curation. **Ziyu Zhang:** Writing – review & editing, Visualization, Investigation. **Jinwen Zhang:** Writing – review & editing, Visualization, Supervision, Resources, Project administration, Methodology. **Fulin Qu:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Piyush Punetha:** Writing – review & editing, Visualization, Investigation. **Wengui Li:** Writing – review & editing, Supervision, Resources, Methodology, Conceptualization. **Ning Li:** Writing – review & editing, Visualization, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the Innovative Research Groups of the National Natural Science Foundation of China (51968022); the Academic and Technical Leaders of Major Disciplines in Jiangxi Province (20213BCJL22039); the Scientific Research Ability Improvement Project of Key Construction Disciplines in Guangdong Province (2022ZDJS095).

References

[1] S. Katsigiannis, S. Seyedzadeh, A. Agapiou, N. Ramzan, Deep learning for crack detection on masonry façades using limited data and transfer learning, J. Building Eng. 76 (2023) 107105.

[2] S. Navaratnam, K. Selvaranjan, D. Jayasooriya, P. Rajeev, J. Sanjayan, Applications of natural and synthetic fiber reinforced polymer in infrastructure: a suitability assessment, J. Building Eng. 66 (2023) 105835.

[3] R. Fan, M.J. Bocus, Y. Zhu, J. Jiao, L. Wang, F. Ma, S. Cheng, M. Liu, Road crack detection using deep convolutional neural network and adaptive thresholding, 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE (2019) 474–479.

[4] K. Chen, G. Reichard, X. Xu, A. Akanmu, Automated crack segmentation in close-range building façade inspection images using deep learning techniques, J. Building Eng. 43 (2021) 102913.

[5] Q. Han, S. Yan, L. Wang, K.i. Kawaguchi, Ceiling damage detection and safety assessment in large public buildings using semantic segmentation, J. Building Eng. 80 (2023) 107961.

[6] Z. Zheng, C. Yi, J. Lin, Y. Hu, A novel deep learning architecture and its application in dynamic load monitoring of the vehicle system, Measurement (2024) 114336.

[7] Z. Al-Huda, B. Peng, R.N.A. Algburi, M.A. Al-antari, A.-J. Rabea, O. Al-maqtari, D. Zhai, Asymmetric dual-decoder-U-Net for pavement crack semantic segmentation, Autom. Constr. 156 (2023) 105138.

[8] Z. Al-Huda, B. Peng, R.N.A. Algburi, M.A. Al-antari, A.-J. Rabea, D. Zhai, A hybrid deep learning pavement crack semantic segmentation, Eng. Appl. Artif. Intel. 122 (2023) 106142.

[9] I.H. Sarker, Machine learning: algorithms, real-world applications and research directions, SN Comp. Sci. 2 (3) (2021) 160.

[10] L. Yang, H. Huang, S. Kong, Y. Liu, A deep segmentation network for crack detection with progressive and hierarchical context fusion, J. Building Eng. 75 (2023) 106886.

[11] D. Zhu, A. Tang, C. Wan, Y. Zeng, Z. Wang, Investigation on the flexural toughness evaluation method and surface cracks fractal characteristics of polypropylene fiber reinforced cement-based composites, J. Building Eng. 43 (2021) 103045.

[12] K. Zhang, W. Wang, Y. Cui, Z. Lv, Y. Fan, X. Zhao, Deep learning-based estimation of ash content in coal: unveiling the contributions of color and texture features, Measurement 233 (2024) 114632.

[13] J. Zhu, J. Zhong, T. Ma, X. Huang, W. Zhang, Y. Zhou, Pavement distress detection using convolutional neural networks with images captured via UAV, Autom. Constr. 133 (2022) 103991.

[14] Z. Tong, T. Ma, W. Zhang, J. Huyan, Evidential transformer for pavement distress segmentation, Comput. Aided Civ. Inf. Eng. 38 (16) (2023) 2317–2338.

[15] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, T. Huang. 2018 Revisiting rcnn: On awakening the classification power of faster rcnn, Proceedings of the European conference on computer vision (ECCV). pp. 453-468.

[16] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.

[17] T. Wang, X. Zhu, J. Pang, D. Lin. 2021 Fcos3d: Fully convolutional one-stage monocular 3d object detection, Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 913-922.

[18] P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A review of yolo algorithm developments, Procedia Comput. Sci. 199 (2022) 1066–1073.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, computer vision–ECCV 2016: 14th european conference, amsterdam, The Netherlands, October 11–14, 2016, proceedings, Part I 14, Springer (2016) 21–37.

[20] R. Girshick, Fast r-cnn, Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440-1448.

[21] J. Moon, Y. Noh, S. Jung, J. Lee, E. Hwang, Anomaly detection using a model-agnostic meta-learning-based variational auto-encoder for facility management, J. Building Eng. 68 (2023) 106099.

[22] A. Sekar, V. Perumal, Automatic road crack detection and classification using multi-tasking faster RCNN, J. Intell. Fuzzy Syst. 41 (6) (2021) 6615–6628.

[23] S. Hao, L. Shao, S. Wang, A faster RCNN airport pavement crack detection method based on attention mechanism, Academic J. Sci. and Technol. 4 (2) (2022) 129–132.

[24] K. Yan, Z. Zhang, Automated asphalt highway pavement crack detection based on deformable single shot multi-box detector under a complex environment, IEEE Access 9 (2021) 150925–150938.

[25] X. Feng, L. Xiao, W. Li, L. Pei, Z. Sun, Z. Ma, H. Shen, H. Ju, Pavement crack detection and segmentation method based on improved deep learning fusion model, Math. Probl. Eng. (2020) 1–22.

[26] Z. Han, H. Chen, Y. Liu, Y. Li, Y. Du, H. Zhang, Vision-based crack detection of asphalt pavement using deep convolutional neural network, Iranian J. Sci. Technol, Transactions of Civil Eng. 45 (2021) 2047–2055.

[27] J. Ha, K. Park, M. Kim, A development of road crack detection system using deep learning-based segmentation and object detection, J. Society for e-Business Studies 26 (1) (2021) 93–106.

[28] J. Terven, D. Cordova-Esparza. 2023 A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS, arXiv preprint arXiv:2304.00501 (2023).

[29] Y. Huang, J. Zheng, S. Sun, C. Yang, J. Liu, Optimized YOLOv3 algorithm and its application in traffic flow detections, Appl. Sci. 10 (9) (2020) 3079.

[30] D. Snegireva, A. Perkova, Traffic sign recognition application using yolov5 architecture, 2021 Int. Russian Automation Conference (RusAutoCon), IEEE (2021) 1002–1007.

[31] Z. Liu, X. Gao, Y. Wan, J. Wang, H. Lyu, An improved YOLOv5 method for small object detection in UAV capture scenes, IEEE Access 11 (2023) 14365–14374.

[32] L. Xu, X. Xu, Q. Xia, Y. Yao, Z. Jiang, A light-weight defect detection model for capacitor appearance based on the Yolov5, Measurement (2024) 114717.

[33] H. Hu, Z. Li, Z. He, L. Wang, S. Cao, W. Du, Road surface crack detection method based on improved YOLOv5 and vehicle-mounted images, Measurement 229 (2024) 114443.

[34] C. Li, H. Yan, X. Qian, S. Zhu, P. Zhu, C. Liao, H. Tian, X. Li, X. Wang, X. Li, A domain adaptation YOLOv5 model for industrial defect inspection, Measurement 213 (2023) 112725.

[35] D. Wang, Z. Liu, X. Gu, W. Wu, Y. Chen, L. Wang, Automatic detection of pothole distress in asphalt pavement using improved convolutional neural networks, Remote Sens. (Basel) 14 (16) (2022) 3892.

[36] R. Zhang, Y. Shi, X. Yu. 2021 Pavement crack detection based on deep learning, 2021 33rd Chinese Control and Decision Conference (CCDC), IEEE. pp. 7367-7372.

[37] N. Hu, J. Yang, X. Jin, X. Fan, Few-shot crack detection based on image processing and improved YOLOv5, J. Civ. Struct. Heal. Monit. 13 (1) (2023) 165–180.

[38] G. Yu, X. Zhou, An improved YOLOv5 crack detection method combined with a bottleneck transformer, Mathematics 11 (10) (2023) 2377.

[39] G. Guo, Z. Zhang, Road damage detection algorithm for improved YOLOv5, Sci. Rep. 12 (1) (2022) 15523.

[40] A.M. Roy, J. Bhaduri, DenseSPH-YOLOv5: an automated damage detection model based on densenet and swin-transformer prediction head-enabled YOLOv5 with attention mechanism, Adv. Eng. Inf. 56 (2023) 102007.

[41] Y. Xu, F. Sun, L. Wang, YOLOv5-PD: a model for common asphalt pavement defects detection, J. Sensors (2022).

[42] V. Pham, D. Nguyen, C. Donan, Road damage detection and classification with yolov7, in: 2022 IEEE International Conference on Big Data (big Data), IEEE, 2022, pp. 6416–6423.

[43] G. Ye, J. Qu, J. Tao, W. Dai, Y. Mao, Q. Jin, Autonomous surface crack identification of concrete structures based on the YOLOv7 algorithm, J. Building Eng. 73 (2023) 106688.

[44] A. Ashraf, A. Sophian, A.A. Shafie, T.S. Gunawan, N.N. Ismail, A.A. Bawono, Efficient pavement crack detection and classification using custom YOLOv7 model, Indonesian J. Electrical Eng. Informatics (IJEEI) 11 (1) (2023) 119–132.

[45] Y. Liu, M. Duan, G. Ding, H. Ding, P. Hu, H. Zhao, HE-YOLOv5s: efficient road defect detection network, Entropy 25 (9) (2023) 1280.

[46] M. Ahmed, R. Seraj, S.M.S. Islam, The k-means algorithm: a comprehensive survey and performance evaluation, Electronics 9 (8) (2020) 1295.

[47] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie. 2022 YOLOv6: A single-stage object detection framework for industrial applications, arXiv preprint arXiv:2209.02976 (2022).

[48] H. Gao, H. Yuan, Z. Wang, S. Ji, Pixel transposed convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (5) (2019) 1218–1227.

[49] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie. 2017 Feature pyramid networks for object detection, Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117-2125.

[50] J. Hu, L. Shen, G. Sun. 2018 Squeeze-and-excitation networks, Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132-7141.

[51] D. Arya, H. Maeda, S.K. Ghosh, D. Toshniwal, Y. Sekimoto, Rdd2022: A multi-national image dataset for automatic road damage detection, arXiv preprint arXiv:2209.08538 (2022).