Journal of Real-Time Image Processing (2024) 21:26
https://doi.org/10.1007/s11554-023-01405-5

RESEARCH

MED-YOLOv8s: a new real-time road crack, pothole, and patch detection model

Minghu Zhao[1] · Yaoheng Su[1] · Jiuxin Wang[1] · Xinru Liu[2] · Kaihang Wang[1] · Zishen Liu[1] · Man Liu[1] · Zhou Guo[1]

Received: 14 October 2023 / Accepted: 19 December 2023 / Published online: 29 January 2024

Abstract
Real-time road damage detection and assessment is crucial to ensure road safety. Traditional road damage detection methods mostly rely on manual labor, which is not only inefficient, but it is also difficult to guarantee its reliability. In this study, a road damage detection model, MED-YOLOv8s, based on YOLOv8s is proposed. MobileNetv3 is adopted as the backbone of the detection algorithm, which reduces the number of parameters and the number of computations in the process of feature extraction, enabling the model to achieve a good balance between the detection speed and the detection accuracy. The introduction of the ultralightweight attention mechanism, ECA, adapts the optimization of the correlation of channels to improve the model generalization performance. In addition, replacing the standard convolution with the DW convolution in the 21st layer of the network not only eliminates part of the redundant feature maps but also better extracts the correlation information between the feature maps. In this study, we also discuss the influence of the mix-up data augmentation weight parameter on the detection effect of the model. The experimental results show that the mAP@0.5 of the MED-YOLOv8s model proposed in this study is 95.2%, which is 1.1% higher than that of the original model, and at the same time, the calculation amount of the model is reduced by 46.2%. This method not only improves the detection accuracy but also greatly reduces the model complexity, providing a reference for subsequent model migration.

Keywords Road damage · YOLOv8 · Lightweight model · Data augmentation · Deep learning

1 Introduction

Contemporary economic development is inseparable from convenient road transportation, and highways are one of the most important civilian facilities [1]. However, after long-term use and natural erosion of highways, a variety of damage will appear, which will seriously affect the service life of highways and the safety of vehicles [2–4]. Real-time pavement damage detection can understand the actual situation of the road, provide accurate data support for the designation and optimization of maintenance plans. Improve the service life of the road and reduce unnecessary maintenance work, thereby saving maintenance costs. And it can also help researchers understand the formation mechanism and development trend of road damage, and promote the development and innovation of road maintenance technology. Therefore, timely detection and early warning of road damage is of great significance to driving safety and economic development [5]. Traditional road damage detection relies on staff experience, mainly manual visual inspection or manual assessment after collecting information from vehicle-mounted sensors, which usually requires considerable wasted time and has questionable accuracy [6–8]. To avoid such problems, the development of new means with automated rapid detection of road damage can greatly improve the efficiency and quality of road maintenance work. With the development of computer technology, methods based on traditional image processing are widely used in road damage detection [9]. These methods are mainly divided into three categories: the first category is the threshold segmentation method, which converts the road image into two areas of black and white and separates the road damage from the normal road surface by setting an appropriate threshold [10–12], the second category is the edge detection method, which extracts the edge information of the road damage in the road

✉ Yaoheng Su
suyaoheng@xpu.edu.cn

[1] School of Science, Xi’an Polytechnic University, Xi’an 710048, China

[2] School of Electronic Information, Xi’an Polytechnic University, Xi’an 710048, China

image [13–15], and the third category is the image transform method, which extracts or represents the damage feature in the road image by transforming the image [16–18]. Although traditional image processing methods have made progress in road damage feature extraction, they do not have the function of damage classification. In addition, due to the large-scale area to be inspected and the influence of the complex environment, traditional image processing means do not meet the requirements of large-scale detection in terms of detection speed and accuracy [6]. Both manual visual inspection and traditional image processing methods are indispensable for road damage detection without human involvement. With the development of artificial intelligence technology, automated target detection and classification technology has been rapidly developed [19]. Target detection based on artificial intelligence techniques is mainly divided into two categories: one-stage detection algorithms and two-stage detection algorithms [20]. Two-stage representative algorithms are R-CNN [21], Fast-RCNN [22], Fast-RCNN [23], FPN [24] and MASK-RCNN [25]. The two-stage target detection algorithm is divided into two phases. The first phase generates a series of proposals, and the second phase detects the target in each candidate. The two-stage detection algorithm can deal with the problem of class imbalance better so that the positive and negative samples are more balanced. Although the accuracy of this method is relatively high, it is slow. One-stage representative algorithms are YOLO series algorithms [26–31]. The one-stage detection algorithms are able to predict the location and class of the output target directly on the image with high detection speed. Two-stage object detection algorithms and one-stage object detection algorithms have developed rapidly in recent years and are widely used in road damage detection [32]. Wang et al. [33] used an improved Faster R-CNN network to detect road damage, with an average F1 value of 63% for eight types of damage detection. Road damage is irregular and the background of the damage is also very complex. To better solve such problems, a densely connected convolutional network is used to replace the backbone of Mask R-CNN, and a feature pyramid network and a full convolutional neural network are used to identify road damage. This method not only realizes damage detection but also performs damage segmentation [34]. Hacıefendioğlu et al. [35] used Faster R-CNN to identify cracks on concrete surfaces and studied the effect of light on the algorithmic model. An improved R-CNN detection model is used to detect vertical cracks in asphalt road with an average error of only 2.33% [36]. And under the same data set (RDD2022), the researchers found that the two-stage algorithm has higher detection accuracy than the original one-stage algorithm [37–39].

However, road damage detection is a wide-ranging task with many targets to be measured. In addition, road damage detection needs to be carried out under the rapid movement of the detection equipment, so the detection speed of the algorithm is highly needed. The one-stage detection algorithm can directly predict the position and category of the output target on the image, with high detection speed, we refer to similar research, on the basis of the one-stage algorithm with high detection speed, by continuously optimizing the network architecture, can improve the detection accuracy [40–42], and the model size is more suitable for embedding in hardware devices. Alfarrarjeh et al. [43] developed a road damage detection algorithm that can embed mobile phones based on YOLOv1, with an average F1 value of 62%. To improve the detection efficiency, MobileNetv3 was used instead of the original backbone network of YOLOv5, which not only improved the accuracy but also reduced the model size by only 4.2 MB and the GFLOPs by a factor of 1.69 [44]. Inam et al. [45] proposed a YOLOv5-based bridge road detection method with a single image inference speed of 1.2 ms. To improve the detection accuracy of large-scale input images, the generalized feature pyramid network (generalized FPN) structure was introduced into YOLOv5 to achieve feature fusion in complex scenes, and the FPS reached 42 [46]. To achieve rapid detection of road damage, a light road crack detection model based on YOLOv4-Tiny is proposed. The final size of the model parameters is 6.33 MB, which is 4–5 times less than the conventional model parameters [47].

Although the latest one-stage detection algorithm performs well in the road damage detection domain, there are problems of low detection accuracy and poor anti-interference ability [32]. When the road has shadows or other light interference, it is easy to misjudge, when the damage is small or the features are not obvious, it is easy to miss detection, and when the image is blurred, the problem of low detection accuracy easily occurs. Based on the above problems, this paper chooses YOLOv8s as the benchmark model, which is the latest YOLO model and has excellent detection speed and accuracy. The main contributions of this paper are as follows:

1. The dataset is roughly expanded online using mosaic and mix-up data augmentation, and the complex environment of the road is simulated.
2. MobileNetv3 is used to replace the original backbone network of YOLOv8, and the ultralight quantization attention mechanism ECA is used to replace the original SE attention mechanism in the MobileNetv3 module, making the improved model more suitable for fast detection tasks.
3. The original convolution at level 21 in neck was replaced with a depthwise separable convolution (DWConv) to further reduce the weight of the model.
4. This paper also discusses the impact of data enhancement strategies on model performance.

2 Materials and methods

2.1 Image dataset

The data in this paper are derived from the open-source dataset RDD2022 [48]. The images are labeled by official professionals and have high resolution and they can be used for tasks such as road damage detection and road condition assessment. The dataset contains various types of road damage, such as cracks, potholes, and road breaks. In this paper, some of these data are selected as the dataset for training and testing, there are a total of 2477 images in the dataset, including 1977 labeled file images and 500 unlabeled images. This paper selects the labeled file test images as the training set and test set. Stochastic divides the 1977 images into 1383 images as the training set, 594 images are the validation set of the model during the training process and another 500 images were manually labeled as model test sets to verify the performance of the model. The total number of labels in the dataset is 4957, and the number of each type of data set is: repair (277), D20 (641), D00 (3774), D40 (235); the number of each type in the training set is: Repair (180), D20 (435), D00 (2649), D40 (168). The number of each type in the validation set is: repair (97), D20 (206), D00 (1125), D40 (67). The dataset of this paper includes four different types of road damage: transverse longitudinal cracks (D00), alligator cracks (D20), potholes (D40), and patch (repair), and an example of the damage types is shown in Fig. 1.

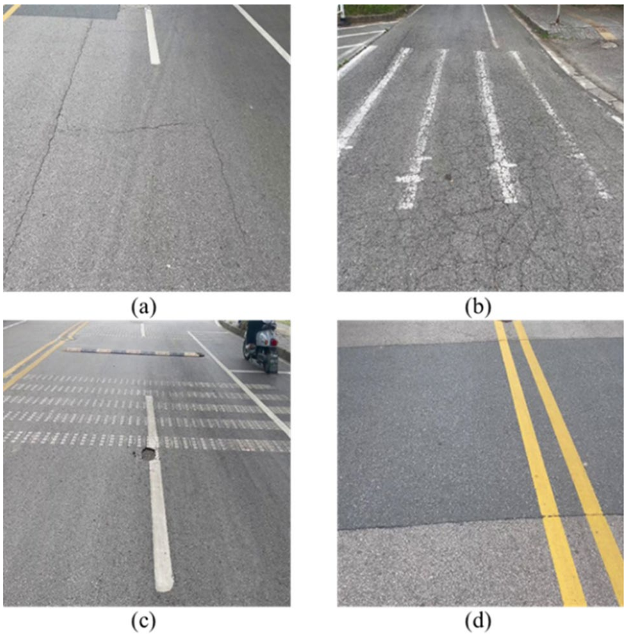
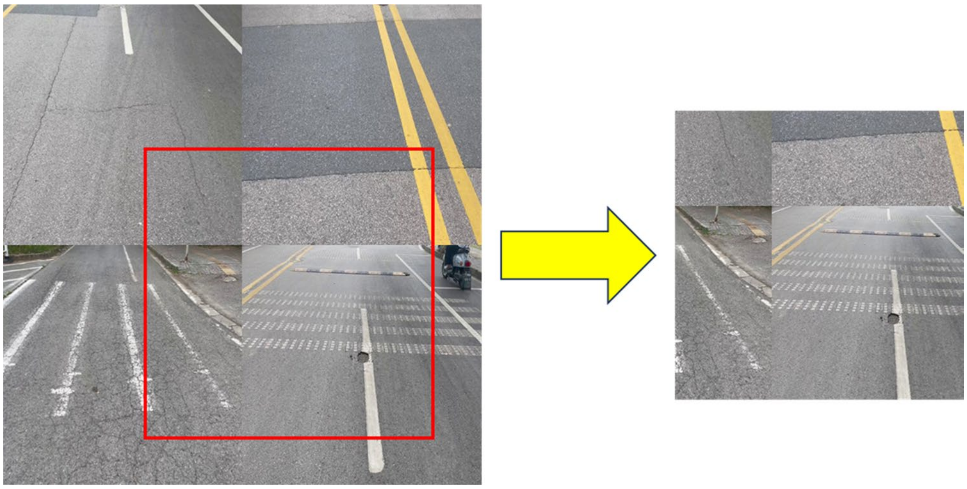


Fig. 1 Damage types: **a** transverse and longitudinal cracks; **b** alligator cracks; **c** potholes; **d** patch

YOLOv8s provides a variety of data enhancement strategies, including mosaic and mixup data enhancement. Bochkovskiy first proposed mosaic enhancement in YOLOv4 [29] to improve the training efficiency. Mosaic data enhancement randomly selects 4 images and stitches the 4 random images together into a single image by resizing. This mosaic data augmented image contains bounding boxes of the four images and is then randomly cropped to obtain a mosaic image. For the remaining images after cropping, if there is a bounding box, it will be used to generate the next mosaic image. Otherwise, it will be deleted. This kind of data enhancement enables the model to deal with multiple different scenes and objects at the same time during the training process, and it can reduce the batch size by a factor of four. Figure 2 shows an example of mosaic data augmentation.

Mixup [49] data enhancement is a common strategy to enhance the robustness and generalization ability of the model, and the effect of data enhancement is shown in Fig. 3 where two input images are randomly selected and

Fig. 2 Mosaic data enhancement example



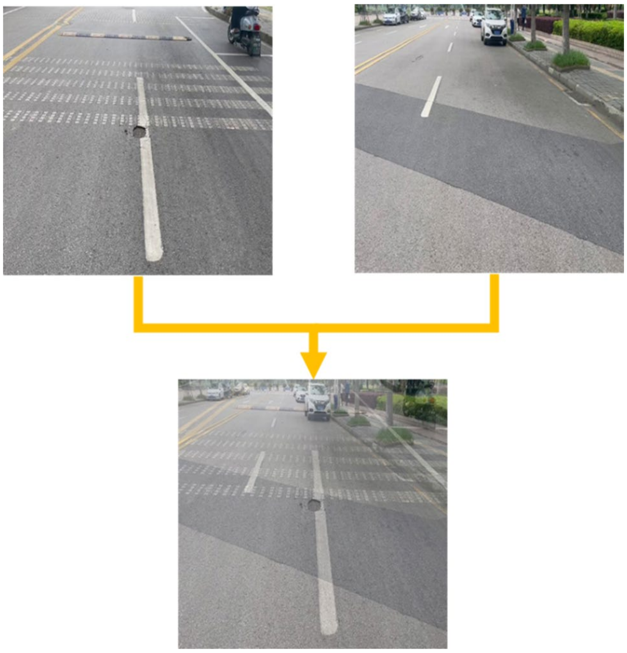


Fig. 3 Example of the mix-up data enhancement effect

a new training image is generated by linear interpolation to achieve the fusion of images and enrich the training samples. Assuming that the features and one-hot labels of the two input images are represented by and, respectively, and the new training samples generated by linear interpolation are represented by, whose features and labels are linear combinations of the original samples, the linear interpolation formula is shown in Eq. (1):

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \\ \tilde{y} = \lambda y_i + (1-\lambda)y_j, \tag{1}$$

where $\lambda \in (0,1)$ is a random variable satisfying the beta distribution. In this paper, we set λ to obey the Beta(α, β) distribution, where $\alpha = 0.4$.

2.2 The YOLOv8 model

YOLOv8 is a major update in the YOLO family of single-stage target detection algorithms, which includes five different versions: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The overall structure of YOLOv8 is similar to that of YOLOv5, which consists of backbone, neck, and head parts. However, it has a new architecture, an improved convolutional layer (backbone) and a more advanced detection head, which now supports not only the target detection task but also the image classification and instance segmentation tasks. The structure of YOLOv8 is shown in Fig. 4.

The input of YOLOv8 uses adaptive scaling technology to scale the input image to 640 × 640 size to better fit the needs of the target detection algorithm. YOLOv8 adaptively generates various sizes of anchor frames and obtains the most suitable anchor frame size through NMS filtering. YOLOv8 not only provides multiple data enhancement strategies but also supports selectively disabling the mosaic data enhancement strategy in the last YOLOv8 not only provides multiple data enhancement strategies but also supports selectively turning off the mosaic data enhancement strategy in the last 10 rounds of model training according to the detection task.

The backbone layer of YOLOv8 adopts a Darknet-53 backbone network, i.e., a "52-layer convolution" + output layer, which uses a convolution operation to extract features of various scales from RGB color images. In the backbone part, YOLOv8 first uses two 3 × 3 convolutions and proposes the C2f module with reference to the idea of the C3 module in YOLOv5 and the ELAN structure in YOLOv7. The information of the feature map is fully extracted by the split operation, and n bottleneck serial branches and more branches are used for cross-layer connections, which enriches the gradient flow of the model and, at the same time, can ensure the lightweight of the model. In addition, YOLOv8 retains the SPPF module in YOLOv5. Compared with the SPP module, the SPPF module changes the parallel max pooling to serial, which not only avoids the image distortion and size inconsistency caused by cropping and scaling operations on the image region but also reduces the computation amount of the model and increases the feeling field of the model.

The neck layer of YOLOv8s uses the PAN-FPN structure to merge the features extracted by the backbone network, which realizes the full fusion of multiscale information. The convolutional structure of the PAN-FPN upsampling stage is removed, and the feature outputs from the different stages of the backbone are directly fed into the upsampling operation. The C3 module is replaced by the C2f module, which further reduces the computational complexity of YOLOv8 while maintaining compatibility with architectures such as YOLOv5.

Although YOLOv8 is structurally similar to YOLOv5z, YOLOv8 adopts the current mainstream decoupled head instead of the coupled head used in YOLOv5 and changes from YOLOv5's anchor-based to anchor-free, which reduces the number of prediction frames and thus speeds up nonmaximal values. number of prediction frames, thus accelerating nonmaximal suppression (NMS). The decoupled head structure separates the classification task from the regression task, removes the object loss, and uses the CIoU and DFL loss functions as the bounding box loss (box loss). In addition, YOLOv8 uses the binary cross entropy (BCE) loss as the classification loss (Cls loss). Different branches and loss functions were used for computation to improve the accuracy of the model.

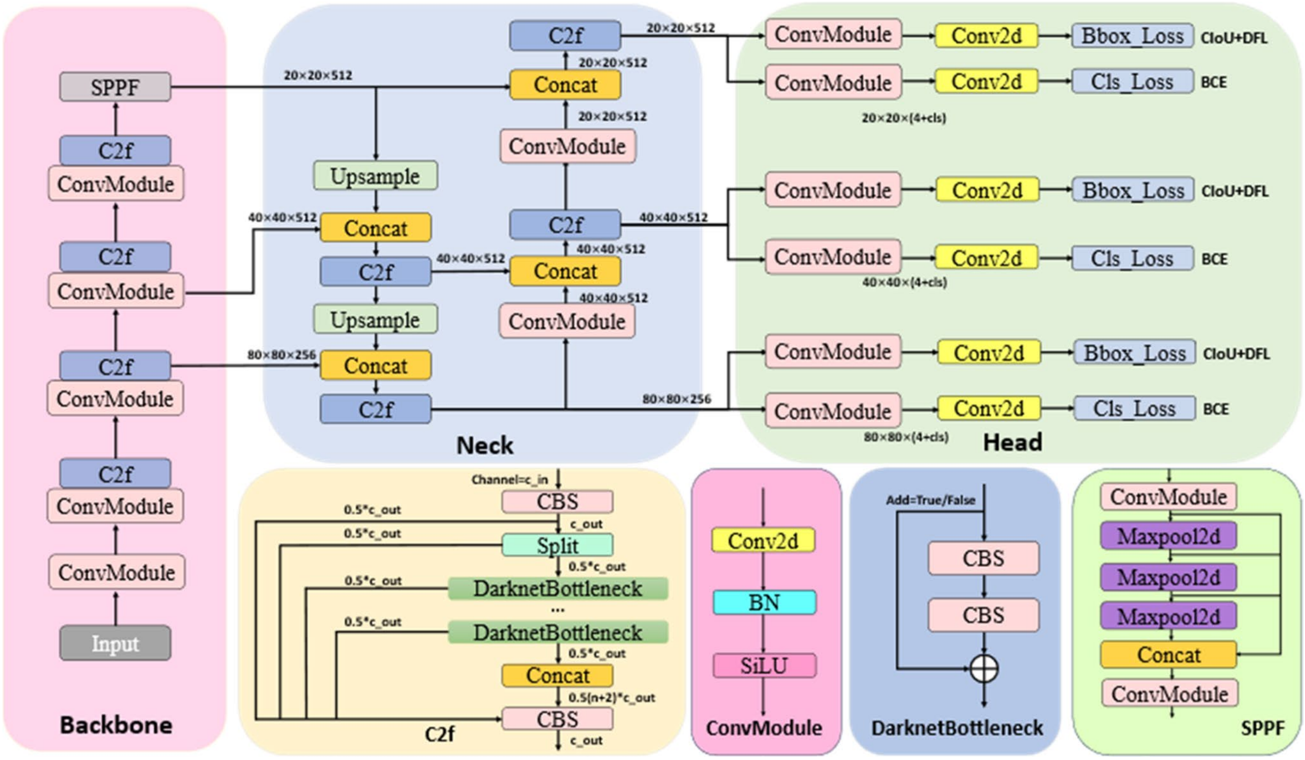


Fig. 4 YOLOv8 structure

3 Improvement of YOLOv8

The performance of the YOLOv8s model is already advanced in the field of target detection; however, road damage detection is a task that requires high detection speed and accuracy, and the original model needs to be further optimized to meet the requirements. In this study, relevant improvements are made on the basis of YOLOv8s, and a lightweight model named MED-YOLOv8s is proposed, which is smaller in size and can provide a basis for subsequent model migration. The structure of MED-YOLOv8s is shown in Fig. 5.

3.1 Improvement of the backbone module

YOLOv8 adopts the structure of DarkNet53, which contains many CBS and C2f modules and mainly consists of ordinary convolution and residual concatenation, with a high number of parameters and a large model size [50]. To reduce the model complexity and improve the portability of the model while guaranteeing the detection accuracy. In this section of experiments, we compare the detection effects of five light quantization networks, including Original-YOLOv8s (N0), YOLOv8s-EfficientNetv2 (N1), YOLOv8s-GhostNet (N2), YOLOv8s-shufflenetV2 (N3), Our (N4), and compare parameters including accuracy, F1, and model size. In addition, according to the experimental results, we further analyze the specific advantages and unique functions of MobileNetv3 [51], and prove that it is better than other comparison models. Four light quantization models and the improved model in this paper are used for comparison. The comparison results are shown in Fig. 6. EfficientNetV2 [52] goes further than EfficientNet [53], improving the training speed and parameter efficiency. This network is generated using a combination of scaling (width, depth, resolution) and neural search architecture. The main goal is to optimize the training speed and parameter efficiency. From the training data, the map and F1 values of N1 are not much different from other network models, but the weight size is the largest among the five light quantization networks. Although N2 has good MAP and F1 values, the weight size cannot meet the requirements of embedded devices. The weight size of N3 is closest to the improved model in this paper, but the accuracy loss due to light quantization reaches 4.1%. MobileNetv3 continues the deep separable convolution of MobileNetv1 and the linear bottleneck residual error structure of MobileNetv2. Reduce the amount of parameter calculation by light quantization network architecture, and reduce the model size while speeding up detection. The purpose of this study is to study a method that can achieve real-time pavement damage detection in low-cost devices with good detection accuracy. N4 can achieve a good balance between mAP and

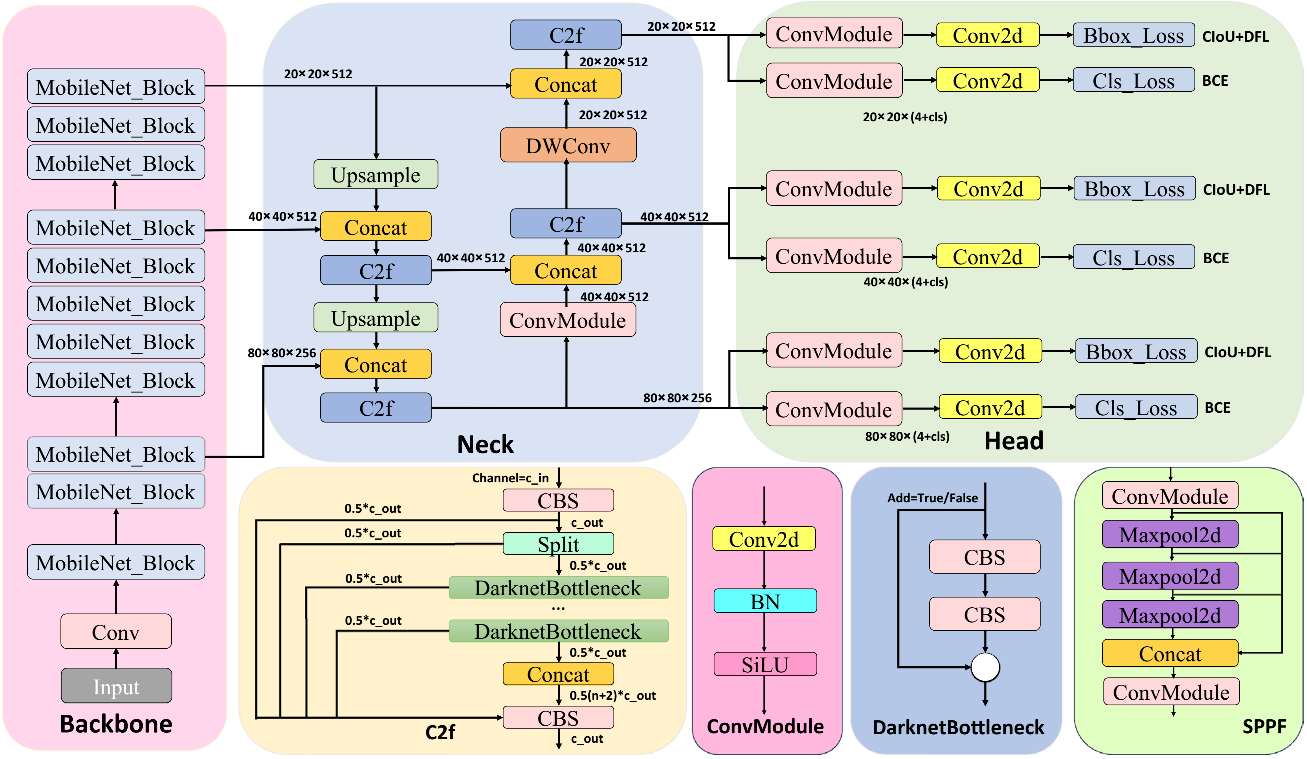
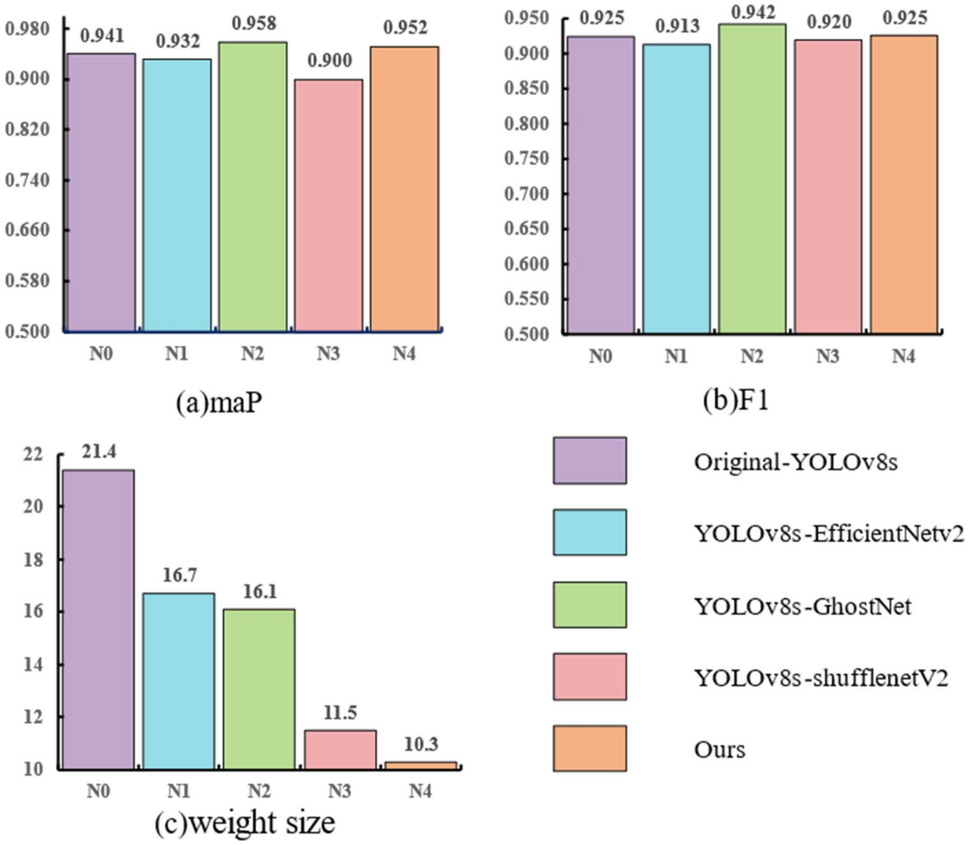


Fig. 5 Improvement of YOLOv8s structure

Fig. 6 Comparisons of lightweight networks



weight size. Therefore, this study uses MobileNetv3 as the backbone network of YOLOv8 for subsequent improvement.

MobileNetv3 carries over the deeply separable convolution of MobileNetv1 [54] and the linear bottleneck residual structure of MobileNetv2 [55]. Parameter computation is reduced by lightweighting the network structure to shrink the model size while speeding up detection. The DW convolution [56] of MobileNetv3 divides the standard convolution into two parts: depth convolution and point-by-point convolution. MobileNetv3 adds a squeeze-and-excite (SE) structure to the neck structure and replaces swish with h-swish. As shown in Eqs. (2) and (3):

$$\text{swish } x = x\sigma(x) \tag{2}$$

$$h - \text{swish } x = x * \frac{\text{ReLU6}(x+3)}{6} \tag{3}$$

MobileNetv3 is designed for efficient image classification and object detection tasks in computer vision. It can reduce the number of parameters and the amount of computation in the feature extraction process and achieve a good balance between speed and accuracy. To achieve a more lightweight effect, this study exchanges the SE [57] module in MobileNetv3 for an ultralightweight attention mechanism, ECA. SE uses global average pooling and two fully connected layers that introduce high computational volume, while ECA [58] uses a one-dimensional convolutional operation, which has a relatively small computational volume and a higher computational efficiency. The improved MobileNetv3 module is shown in Fig. 7.

3.2 Improvement of the neck module

Traditional feature extraction methods used for convolutional neural networks include using multiple convolutional kernels using convolutional mapping operations on all input channels. However, stacking multiple convolutional layers produces many duplicate feature maps, which generates more computation. Figure 8 shows the crack feature map of the dataset after ordinary convolution. It can be seen that after multiple ordinary convolutions, a number of redundant images are created that are very similar, and although these redundant feature maps are also useful for network training, creating them is computationally and time costly.

The YOLOv8 deep convolutional neural network can output feature maps at multiple scales to predict targets at different scales, and the structure of PAN-FPN is used to fuse different levels of feature maps at the neck layer, but the information contained in the high-level feature maps during the fusion process may have already appeared in the low-level feature maps, which will result in redundant computation and parametric counts after Concat. Deep separable convolution can be used as an independent module to replace the standard convolution in YOLOv8, reducing the computation and number of parameters. In addition, YOLOv8 processes images by meshing the input image, each mesh corresponds to a different feature map, and the overlapping regions of the feature maps may contain similar feature information, especially in the high-level feature maps Therefore, the standard convolution before the last layer of the Concat operation is replaced with the DWConv module in this paper's experiments to reduce redundancy and help the model better capture the details and features of the input feature maps.

Depthwise convolution divides the convolution operation into two steps: depthwise convolution and pointwise convolution. First, depthwise convolution performs the convolution operation only on each channel of the input feature map, uses a smaller convolution kernel to obtain the spatial information of the input feature map and can reduce the computation of the model. Second, a pointwise convolution operation is performed to convolve the output of depthwise convolution using a 1 × 1 convolution kernel. The role of pointwise convolution is to linearly combine the output feature maps of depthwise convolution to generate the final output feature maps, increase the depth of the feature maps, and improve the expressive power of the network.

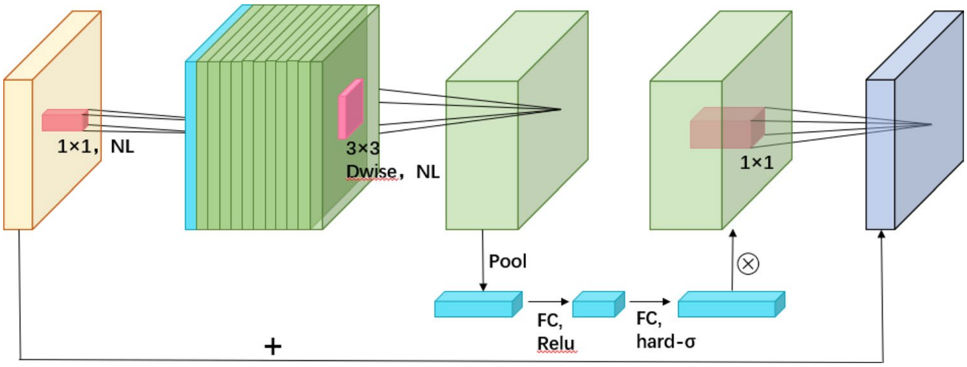


Fig. 7 MobileNetv3 network architecture

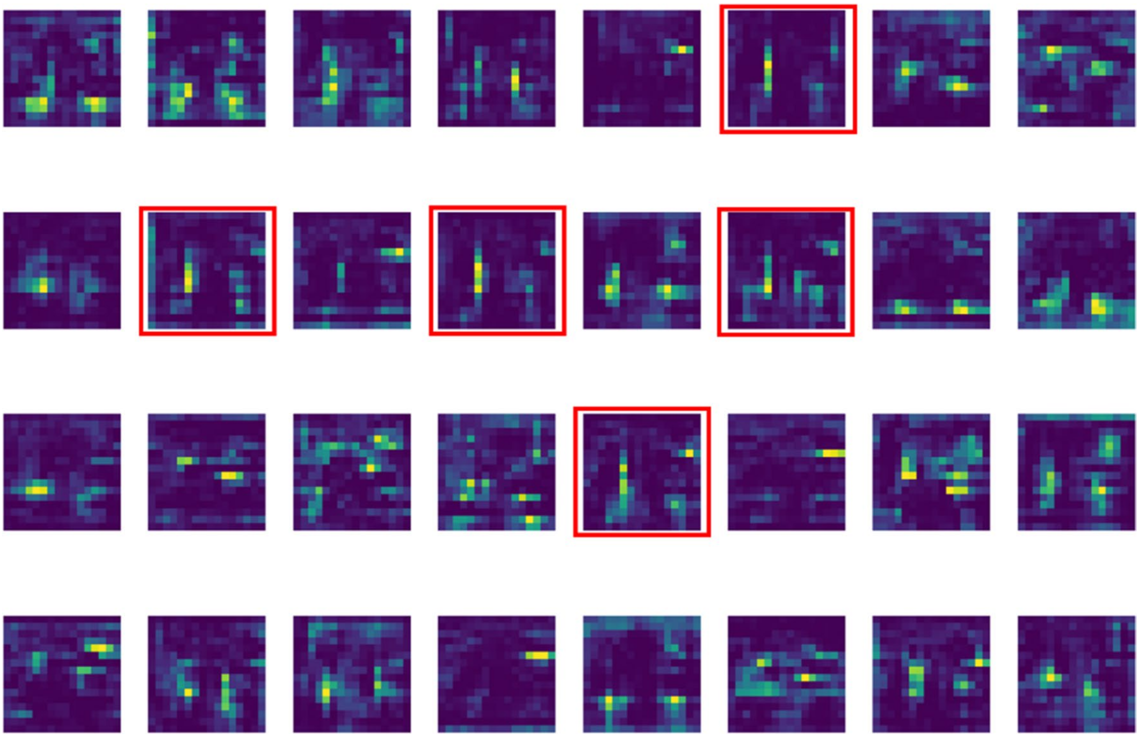


Fig. 8 Plain convolutional feature map

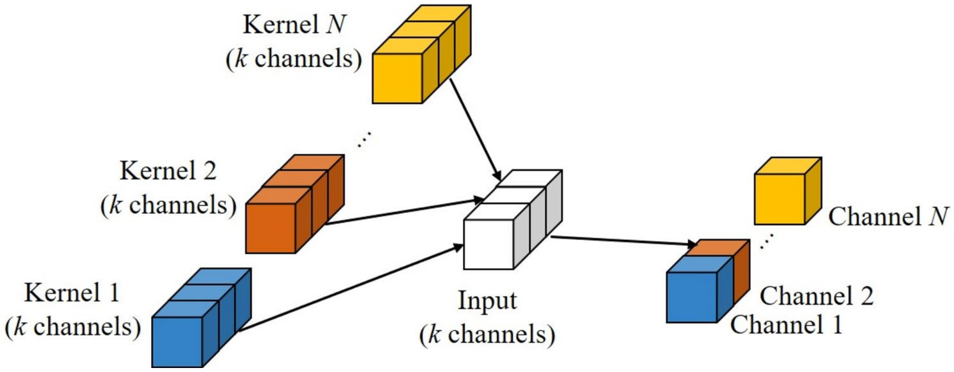
Let the size of the input feature map be $D_k \times D_k \times M$ if there are N convolution kernels of size $D_F \times D_F \times M$.

(1) For ordinary convolution (e.g., Fig. 9), assuming that each point in the feature map undergoes a convolution operation, the computational effort for the case of a single convolution kernel is $D_k \times D_k \times D_F \times D_F \times M$, and the computational effort for N convolution kernels can be expressed as Eq. (4).

$$D_k \times D_k \times D_F \times D_F \times M \times N \tag{4}$$

DWConv splits the convolution operation into two steps (as shown in Fig. 10). Then, the depthwise convolution operation uses a single convolution kernel for the convolution operation, which is a separate convolution of each channel of the image, and it is an operation in the two-dimensional plane, i.e., the amount of computation is $D_k \times D_k \times D_F \times D_F \times M$. The pointwise convolution operation is a pair that combines the channel-by-channel convolution process to obtain the pointwise convolution operation combines the independent feature maps obtained from the channel-by-channel convolution process and weights the combination in the depth direction to generate a new feature map, i.e., the computational amount is $M \times N \times D_k \times D_k$. The above two steps are independent of each other. The

Fig. 9 Conv structure



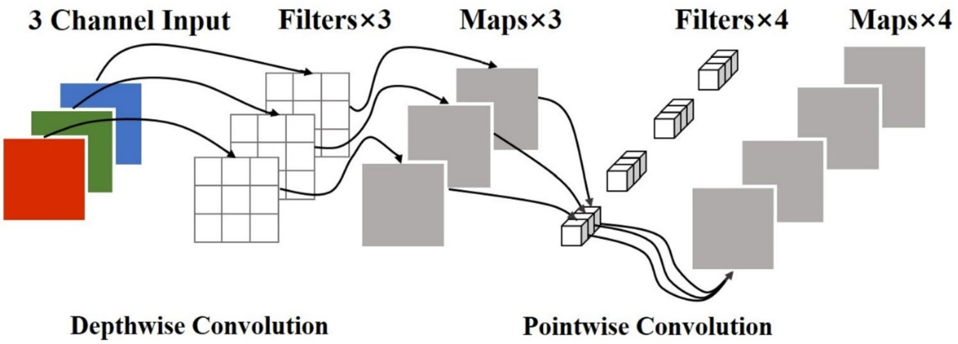


Fig. 10 DWConv structure

above two steps are independent of each other, so the computational amount of DWconv is the sum of the computational amount of the depthwise convolution operation and pointwise convolution operation:

$$D_k \times D_k \times D_F \times D_F \times M + M \times N \times D_k \times D_k \tag{5}$$

From Eqs. (4) and (5), the ratio of the computational effort of DWConv and ordinary convolution is

$$\frac{D_k \times D_k \times D_F \times D_F \times M + M \times N \times D_k \times D_k}{D_k \times D_k \times D_F \times D_F \times M \times N} = \frac{1}{N} + \frac{1}{D_F \times D_F} \tag{6}$$

Therefore, the use of DWConv not only improves the expressive power of the model but also has obvious advantages in terms of computational and parametric quantities.

4 Experiments and results

4.1 Experimental environment

The operating system used in this study was Windows 10 Professional, the CPU model was a 12th Gen Intel(R) Core(TM) i5-12400F, and the GPU was an NVIDIA GeForce RTX 3070; PyTorch was used as the framework for developing the deep learning model with version Model 2.0.1; the compiled language was Python 3.8.17; and the YOLOv8 version was Ultralytics 8.0.135. In the training phase, mosaic data enhancement of 1.0 was used, and the mixup value was set to 0.5. The input image was of size 640 × 640, the batch size was set to automatic, and the epochs were set to 600 rounds. YOLOv8 requires the input image size to be 640 × 640, the original image needs to be resized to the standard size input network, and the direct use of stretching may cause the target scale imbalance (distortion), so this paper maintains the original input data set image size 512 × 512, and YOLOv8 has the ability to image AdaGrad scaling, which can scaling the original image in equal proportion (when the width/height is up to 640, the remaining background filling is used).

4.2 Indicators for model evaluation

In this study, precision, recall, F1-score, FPS (frames per second), and mean average precision (mAP) were used to evaluate the recognition performance of road surface damage. Precision is used to reflect the accuracy of the prediction of road surface damage, expressed as the proportion of true positive samples among the positive samples determined by the classifier, and recall is used to reflect whether all road surface damage is detected, expressed as the proportion of correctly determined positive samples among the total positive samples. The formulas for precision and recall are shown in Eqs. (7) and (8):

$$P_{\text{Precision}} = \frac{T_{\text{P}}}{T_{\text{P}} + F_{\text{P}}} \times 100\% \tag{7}$$

$$P_{\text{Recall}} = \frac{T_{\text{P}}}{T_{\text{P}} + F_{\text{N}}} \times 100\% \tag{8}$$

where T_{P} denotes the number of correctly classified positive samples (true positives), F_{P} denotes the number of incorrectly classified positive samples (false-positives), F_{N} denotes the number of incorrectly classified negative samples (false-negatives), and T_{N} denotes the number of correctly classified negative samples (true negatives).

The mAP is a measure of detection accuracy in target detection. The mAP is obtained by calculating the average precision of each category through P–R (Precision–Recall) curve integration and then averaging, and the formula is shown in Eq. (9):

$$m_{\text{mAP}} = \int_0^1 P(R)\text{d}R \tag{9}$$

The F1-score can be used to balance the two metrics of precision and recall. This metric is also the evaluation metric specified in the IEEE Big Data 2022 [48] Road Damage Detection Challenge, and it is defined as shown in Eq. (10):

$$F_{1-\mathrm{Score}} = 2 \cdot \frac{P \cdot R}{P+R} \tag{10}$$

In addition to detection accuracy, operation speed is another important performance metric for target detection algorithms. A common metric used to evaluate the speed is FPS, which represents the number of images that can be processed per second and is calculated as shown in Eq. (11). In this study, FPS was tested on a single NVIDIA GeForce RTX 3070 graphics card.

$$\mathrm{FPS} = \frac{1000}{T_{\mathrm{preprocess}} + T_{\mathrm{inference}} + T_{\mathrm{postprocess}}} \tag{11}$$

4.3 Experimental results

During model training, a new model is generated for each round. The validation set can evaluate and adjust model performance during model training, help identify over-fitting phenomena, and also evaluate model performance under different hyperparameter combinations, and select the best hyperparameter combination to improve the generalization ability and overall performance of the model. After each training session, the network uses the 594 validation sets described earlier to evaluate the performance of the model. Throughout the training round, YOLOv8 automatically saves a model with the best performance. The specific results are shown in Table 1, where precision is 92.6%, recall is 92.5%, F1 is 92.5%, and mAP@0.5 is 95.2%. The MED-YOLOv8s model also has high detection performance for different types of road damage. In all kinds of damage, the feature of patch is more obvious, and the model can easily extract the feature of this type of damage, so the detection accuracy of patch is the highest, and its precision, recall and mAP@0.5 are 97.3, 97.9 and 97.9%, respectively. However, for the network crack (D20), because it is easily confused with the damaged traffic markings, the model learns some incorrect features, thus affecting the detection performance. However, the precision, recall, and mAP@0.5 of this type of damage are 90.6, 94.7, and 96.5%, respectively. Transverse and longitudinal cracks (D00), i.e., single cracks in different directions, which are characterized by large crack widths and are susceptible to interference from objects with similar characteristics on both sides of the road, are another challenging type of damage. For example, factors such as tree trunks or shadows cast by objects may cause the model to misjudge. Therefore, the detection accuracy of this kind of injury is lower than that of other injuries; the precision is 87.1%, recall is 82%, and mAP@0.5 is 88.9%. For pothole (D40) damage, when the model is trained, the potholes are specially marked to help the model learn features to improve the model's ability to detect potholes. Its precision, recall and mAP@0.5 are 95.5, 95.5 and 97.6%, respectively.

To verify the detection effect of the MED-YOLOv8s model proposed in this study on different types of road damage, we selected 500 manually labeled images and used the model for inference. The experimental results are shown in Table 1. The number of labels of each type: Repair (88), D20 (180), D00 (789), D40 (57). The number of labels successfully detected by the model: Repair (86), D20 (177), D00 (746), D40 (53). The model was applied to the detection of different road damage images in the test set. The detection results are shown in Fig. 11. Figure 11a shows the detection results of longitudinal cracks. Whether it is obvious cracks or slight cracks, the MED-YOLOv8s model proposed

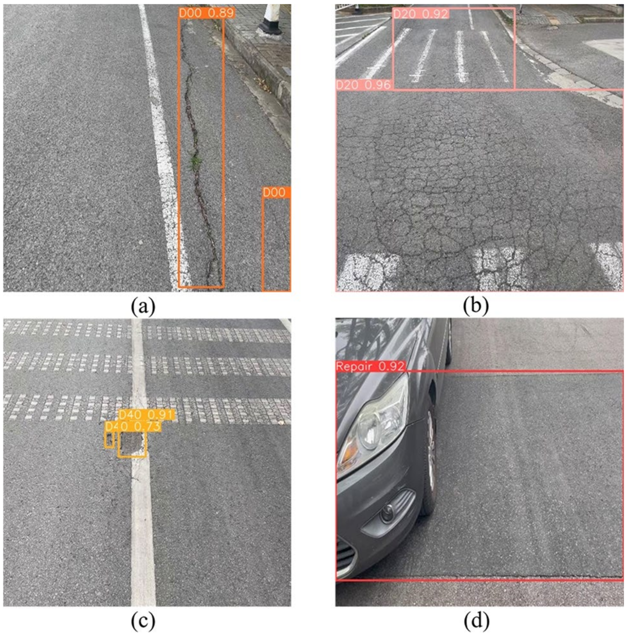


Fig. 11 MED-YOLOv8s model inspection results. **a** Transverse and longitudinal cracks; **b** alligator cracks; **c** potholes; **d** patch

Table 1 MED-YOLOv8s training results

form	Precision	Recall	F1-score	mAP@0.5	Correct number	Total number
All	92.6%	92.5%	92.5%	95.2%	1062	1114
Repair	97.3%	97.9%	97.6%	97.9%	86	88
D20	90.6%	94.7%	92.6%	96.5%	177	180
D00	87.1%	82.0%	84.4%	88.9%	746	789
D40	95.5%	95.5%	95.5%	97.6%	53	57

in this paper can detect them. This shows that the model has good recognition ability for longitudinal cracks, which can help workers detect such damage in time, thereby improving the safety and reliability degree of roads. Figure 11b shows the detection result of alligator cracks. Although alligator cracks are easily confused with damaged traffic markings, the MED-YOLOv8s model proposed in this paper can still identify such damage and detect it accurately, which shows that the MED-YOLOv8s model still has high robustness in the face of complex scenarios. Figure 11c is the detection result of potholes. For potholes with smaller targets, the MED-YOLOv8s model can also achieve better detection results, which is essential for maintaining road safety and comfort because small potholes may gradually expand and cause potential dangers to vehicles and pedestrians. Figure 11d is the detection result of patch. The model can accurately detect patch blocked by cars because our model fully learns the features of patch during the training process. This ability makes the application of the model in actual road scenarios more reliable and practical. The results show that the MED-YOLOv8s model can accurately identify different types of road damage and has a good detection effect on small target potholes and patch blocked by cars. The model has high precision and reliability in road damage detection. These results provide strong support for road maintenance and traffic safety and provide a research basis for further model embeddings.

4.4 Comparison of different algorithms

To verify the effectiveness of the MED-YOLOv8s model in this study, this section designs a comparison experiment of different algorithms on the same dataset under the same conditions. The comparison algorithms include the mainstream single-stage object detection algorithms YOLOv5s, YOLOv6s, YOLOv7s and YOLOv8s and the latest two-stage object detection algorithm Sparse R-CNN [59]. Under the same conditions, the above five object detection algorithms were used to train the road damage dataset. An example of the detection road damage results is shown in Fig. 12. The training results are shown in Table 2. The MED-YOLOv8s model proposed in this paper has significant advantages in mAP@0.5, GFLOPS, etc. The model size is 10.3 MB, which is only approximately half that of the original model. Compared with YOLOv5s, YOLOv6s, YOLOv7 and SparseR-CNN, it is reduced by 41.5, 67.0, 85.5 and 97.5%, respectively; the mAP@0.5 reaches 95.2%, which is 1.1% higher than the original model compared with YOLOv8s. Recall refers to the proportion of the number of correctly detected targets to all actual targets when using the network for object detection. The model proposed in this paper is an improved model after light quantization, while other models in the table are original models. In terms of network depth and parameter quantity, the improved model has a shallower depth and fewer parameters than other models, while YOLOv7 is the original model, with deep network, many parameters, large size, and strong learning features, so it has higher recall. However, this paper proposes a light quantization model suitable for fast and accurate identification of road diseases. The size of YOLOv7 reaches 71.3 MB, which is nearly 7 times that of the improved model, and the Gflops reaches 105.2. The required computing power is 6.8 times that of the improved model. Such a huge size and required computing power cannot meet the needs of real-time detection of road diseases.

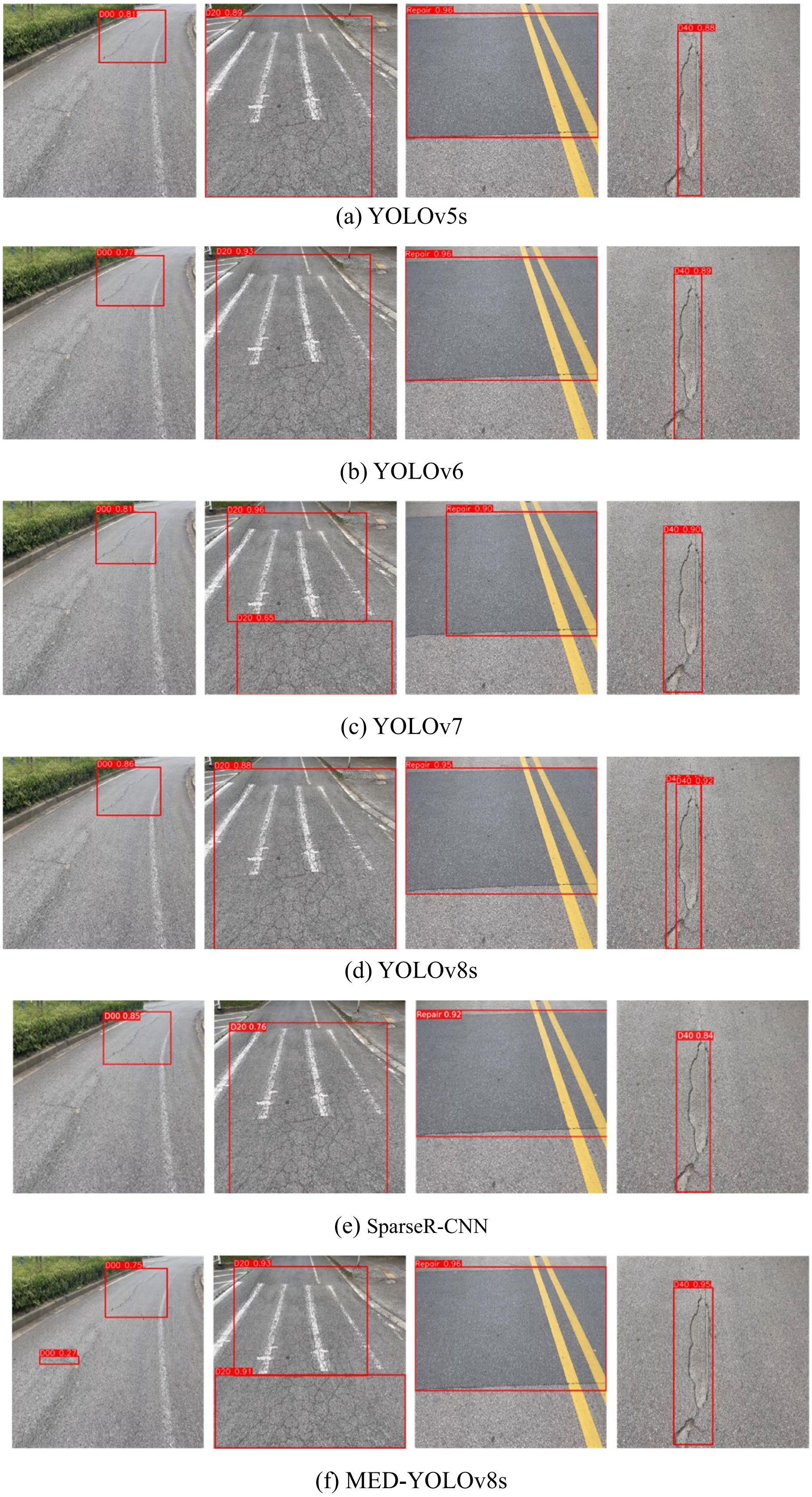
The experimental results show that the MED-YOLOv8s model proposed in this study has better detection performance in road damage detection. It is more suitable for road damage detection because it can ensure high detection accuracy while significantly reducing the computational burden of the model.

4.5 Ablation experiments

Selecting a new backbone feature extraction network MobileNetV3 to replace YOLO's basic network can reduce the number of parameters and GFLOPs of the model, and reduce the size of the model [44, 60]; introducing DWConv in YOLO can reduce the number of parameters and improve the inference speed of the network [61, 62]; compared with the original YOLO network model, the improved network model incorporating ECA-Net only needs to increase a small number of parameters, and realizes higher classification and detection accuracy without affecting the detection speed [63, 64]. Therefore, in order to verify the effectiveness of the improved algorithm in this paper, an ablation experiment is designed, and the results are shown in Table 3. Model 1 represents the original YOLOv8s model. Model 2 represents the model after replacing the backbone network of the original YOLOv8s with MobileNetv3. Model 3 represents the model that replaces the Conv in the 21st neck layer with the depth separable convolution (DWConv) based on Model 2. Model 4 represents the model that replaces the SE in MobileNetv3 with the attention mechanism ECA based on Model 3. Model 5 denotes the model after setting the mixup data enhancement weight to 0.5 on the basis of Model 4.

As shown in Table 4, the size of Model 2 is 12.3 MB which is 42.5% smaller than that of Model 1 indicating that the improvement of using MobileNetv3 to replace the backbone of YOLOv8 greatly reduces the size of the model and achieves the improvement purpose of model light quantization. The frame rate of Model 2 is 123.5 FPS, which is 14.6% lower than that of Model 1. This is due to the introduction of MobileNetv3, which reduces the number of parameters of the model and the detection ability, resulting in a longer target retrieval time. Furthermore, DW convolution can not only

Fig. 12 Comparison of detection results of different algorithms



(a) YOLOv5s

(b) YOLOv6

(c) YOLOv7

(d) YOLOv8s

(e) SparseR-CNN

(f) MED-YOLOv8s

Table 2 Comparison of training results of different algorithms

Model	Weight size	Precision	Recall	mAP@0.5	GFLOPs
YOLOv5s	17.6 MB	93.1%	90.5%	94.5%	24.2
YOLOv6s	31.2 MB	92.2%	91.2%	94.1%	44.9
YOLOv7	71.3 MB	93.1%	95.1%	95.9%	105.2
YOLOv8s	21.4 MB	92.6%	92.4%	94.1%	28.8
SparseR-CNN	413 MB	57.8%	68.4%	84.9%	246
MED-YOLOv8s	10.3 MB	92.6%	92.5%	95.2%	15.5

eliminate some redundancy but also better extract the correlation information between feature maps. Therefore, Model 3 further lightens the model to 11.2 MB on the basis of Model 2, and the mAP@0.5 of the model is improved by 0.7% compared with Model 2. Model 4 introduces the ultralight quantization attention mechanism ECA to replace the SE attention mechanism in MobileNetv3, and the frame rate of the model is 10.8 FPS higher than that of Model 3 and model size is further reduced to 10.3 MB. Finally, to compensate for the inevitable loss of precision caused by light quantization of the network, Model 5 uses a data augmentation strategy, setting the weight of mixup data augmentation to 0.5. Map@0.5 is 0.9% higher than the average precision of the Model 4 network. The purpose of this study is to develop a lightweight and efficient network architecture that can meet the requirements of real-time accuracy in road damage detection. We validate the feasibility of model improvement through ablation experiments. Model 4 is the model with the best network architecture. To further improve model performance, we design data augmentation weight experiments considering the possible impact of data augmentation strategies on model performance. Data augmentation can increase the diversity of data, thus helping the model to generalize better during training. It is undeniable that it may play a role in model enhancement. Although data augmentation has the potential to generate relevant data and may lead to result deviation, this does not mean that we should give up data augmentation completely. In fact, by rationally designing data augmentation strategies, it is possible to minimize correlation and deviation while preserving the advantages of data augmentation. Through the above ablation experiments, the improved method in this paper not only improves the average precision of the model but also reduces the model size by 51.9%, so the improved strategy in this paper is effective.

5 Discussion

5.1 Comparison and discussion of different improvement strategies

In order to compare the model effects of replacing DWConv at different locations, this section discusses three different improvement strategies through comparative experiments. M1–M3 are three different improvement strategies. M1 represents replacing all two standard convolutions in the Neck network of the network with DWConv, M2 replaces the first standard convolution in the Neck network with DWConv, and M3 replaces the second standard convolution in the Neck network with DWConv. It can be seen from the experimental data that when DWConv is placed in different positions, the performance of the model changes accordingly. Obviously, the various indicators of M3 are more suitable for the requirements of real-time and accuracy of road detection.

5.2 Influence of data augmentation on the object detection model

At present, there is a problem of a single sample and similar background in the open-source dataset of road damage

Table 3 Results of ablation experiments

Group	Precision	Recall	mAP@0.5	mAP@0.95	F1-Score	Weight size	FPS
Model 1	92.6%	92.4%	94.1%	67.8%	92.5%	21.4 MB	156.3
Model 2	93.8%	90.2%	93.6%	65.4%	92.0%	12.3 MB	123.5
Model 3	92.9%	90.9%	94.3%	65.8%	91.9%	11.2 MB	123.5
Model 4	92.4%	91.3%	94.3%	67.7%	91.8%	10.3 MB	133.3
Model 5	92.6%	92.5%	95.2%	68.1%	92.5%	10.3 MB	131.6

Table 4 Comparison of three improvement strategies

Model	D1	D2	Weight size	Precision	Recall	mAP@0.5	F1
M1	√	√	10.9 MB	89.8%	89.2%	92.5%	89.5%
M2	√		12.0 MB	89.9%	89.7%	93%	89.8%
M3		√	11.2 MB	92.9%	90.9%	94.3%	91.9%

detection, which will lead to poor generalization performance of the trained model. Data augmentation is a commonly used technique to expand the training dataset to improve the performance of object detection models. By performing a series of transformations on the original image, a new image sample is generated and added to the training set to enrich the training data. YOLOv8 provides a variety of data augmentation strategies, of which mosaic is set to 1.0 by default. The MED-YOLOv8s model uses mixup data augmentation on the basis of mosaic data augmentation to further improve the complexity of the dataset. To verify whether data augmentation affects the detection effect of road damage, this section trains the improved YOLOv8s model with datasets using mixup data augmentation and datasets without mixup data augmentation. The comparison experimental results are shown in Table 5. MED-YOLOv8s performs better on the dataset using mixup data augmentation, and the precision, recall, mAP@0.5 and F1 of the model are improved by 0.2, 1.2, 0.9 and 0.7%, respectively, compared with the original model. It can be seen that the dataset sample using data augmentation is more complex, which makes the overall detection performance of the model better. Therefore, the use of a reasonable data augmentation method can enhance the detection effect of the model on complex images.

Table 5 Model detection metrics before and after using mixup

Whether mixup was used	Precision (%)	Recall (%)	mAP@0.5 (%)	F1-score
No	92.4%	91.3%	94.3%	91.8%
Yes	92.6%	92.5%	95.2%	92.5%

5.3 Effect of data enhancement parameters on model performance

To increase the complexity of the dataset, mosaic data enhancement and mixup data enhancement strategies are used in this paper. These two data enhancement methods have been widely used in studies similar to this paper. In the official model released by YOLO, the mosaic parameter is set to 1.0 by default, and the mixup parameter is set to 0. Both data enhancement methods can take values ranging from 0 to 1. Mosaic has been widely used in previous studies, but there are fewer studies on the mixup data enhancement strategy. Therefore, this paper focuses on the effect of different weights of mixup settings on the model performance through experiments, and the experimental results are shown in Fig. 13. The mixup weights start from 0 and increase by 0.1 each time, and it can be seen from Fig. 13 that the mixup data enhancement parameter settings with different weights affect the detection performance of the model. With the change in mixup weights, map reaches a peak value of 95.2% at weights of 0.2 and 0.5, while at the same time, the F1 values are 92.4 and 92.5%, respectively. In a comprehensive analysis, the model reaches the best performance when the mixup weight parameter is equal to 0.5. Therefore, in this paper, the mixup parameter is set to 0.5. When using the data enhancement method, it is important to reasonably set the data enhancement parameter that is suitable for the model which is of great help to train the best model.

6 Conclusions

To achieve high-precision and fast road damage detection, a lightweight road damage detection method called MED-YOLOv8s is proposed. First, MobileNetv3 is used as the backbone network of YOLOv8s, and the SE attention mechanism in MobileNetv3 is replaced by the

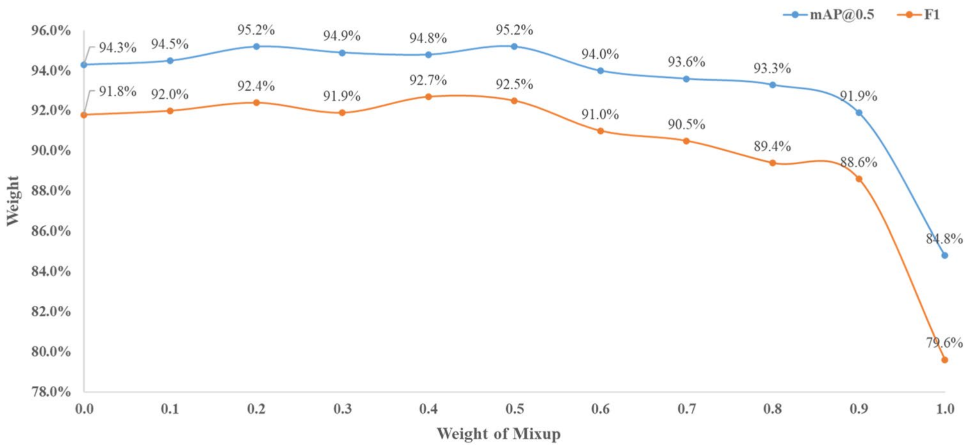


Fig. 13 Effect of the mix-up weighting parameters on the model

ultralightweight attention mechanism ECA, which makes the original YOLOv8s model more lightweight. Second, the standard convolution of the 21st layer in the neck layer is replaced by DW convolution, which not only reduces the model size but also eliminates the redundant feature map generated by the model in the detection process, which is helpful for improving the model lightweight and accuracy. Finally, this paper discusses the impact of the data enhancement strategy on the model detection performance and obtains the best data enhancement weight parameters through experiments to further improve the detection accuracy of the model. The MED-YOLOv8s model proposed in this study has improved the applicability of complex traffic environments and embedded hardware devices, which is better than all the comparative models, not only having the smallest model size (10.3 MB) and the smallest calculation amount (GFLOPs = 15.5) but also having a detection accuracy improvement of 1.1%. In future work, we will further explore the performance improvement of MED-YOLOv8s embedded in mobile devices to provide a reference method for early warning of road damage.

Acknowledgements This study was partly supported by the XIAN Youth Talent Support Program (Grant no. 959202313010). The authors are responsible for all views and opinions expressed in this paper

Author contributions MZ: Methodology, Software, Formal analysis, Investigation. YS: Supervision, Writing – original draft, Writing – review & editing, Funding acquisition. JW: Conceptualization, Methodology, Validation, Supervision. XL: Conceptualization, Methodology, Validation, Supervision. KW: Methodology, Validation, Supervision. ZL: Methodology, Validation, Supervision. ML: Validation, Resources, Supervision, Writing – review & editing. ZG: Validation, Resources, Supervision, Writing review & editing.

Data availability Data will be made available on request.

Declarations

Conflict of 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.

References

1. Hou, Y., Li, Q., Zhang, C., et al.: The state-of-the-art review on applications of intrusive sensing, image processing techniques, and machine learning methods in pavement monitoring and analysis. Engineering **7**(6), 845–856 (2021)
2. Pais, J.C., Amorim, S.I.R., Minhoto, M.J.C.: Impact of traffic overload on road pavement performance. J. Transp. Eng. **139**(9), 873–879 (2013)
3. Madli, R., Hebbar, S., Pattar, P., et al.: Automatic detection and notification of potholes and humps on roads to aid drivers. IEEE Sens. J. **15**(8), 4313–4318 (2015)
4. Gao, Y., Cao, H., CAI, W., et al.: Pixel-level road crack detection in UAV remote sensing images based on ARD-Unet. Measurement **219** (2023)
5. Rojo, M., Gonzalo-Orden, H., Linares, A., et al.: Impact of a lower conservation budget on road safety indices. J. Adv. Transp. **2018**, 1–9 (2018)
6. Pan, Y., Zhang, X., Tian, J., et al.: Mapping asphalt pavement aging and condition using multiple endmember spectral mixture analysis in Beijing, China. J. Appl. Remote Sens. **11**(1) (2017)
7. Zalama, E., Gómez-García-Bermejo, J., Medina, R., et al.: Road crack detection using visual features extracted by Gabor filters. Comput.-Aid. Civ. Infrastruct. Eng. **29**(5), 342–358 (2014)
8. Laurent, J., Hébert, J.F., Lefebvre, D., et al.: Using 3D laser profiling sensors for the automated measurement of road surface conditions. Rilem Bookser. **4**, 157–167 (2012)
9. Gopalakrishnan, K.: Deep learning in data-driven pavement image analysis and automated distress detection: a review. Data **3**(3) (2018)
10. Quan, Y., Sun, J., Zhang, Y. et al.: The method of the road surface crack detection by the improved Otsu threshold. In: Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), 2019
11. Dan, D., Dan, Q.: Automatic recognition of surface cracks in bridges based on 2D-APES and mobile machine vision. Measurement **168** (2021)
12. Wang, W., Li, L., Han, Y.: Crack detection in shadowed images on gray level deviations in a moving window and distance deviations between connected components. Constr. Build. Mater. **271** (2021)
13. Zhao, H., Qin, G., Wang, X.: Improvement of canny algorithm based on pavement edge detection. In: Proceedings of the 2010 3rd International Congress on Image and Signal Processing, 2010. IEEE
14. Hanzaei, S.H., Afshar, A., Barazandeh, F.: Automatic detection and classification of the ceramic tiles' surface defects. Pattern Recognit. **66**, 174–189 (2017)
15. Li, P., Xia, H., Zhou, B., et al.: A method to improve the accuracy of pavement crack identification by combining a semantic segmentation and edge detection model. Appl. Sci. **12**(9) (2022)
16. Prasad, A., Kumar, M., Choudhury, D.R.: Color image encoding using fractional Fourier transformation associated with wavelet transformation. Opt. Commun.Commun. **285**(6), 1005–1009 (2012)
17. Sharma, K.K., Sharma, M.: Image fusion based on image decomposition using self-fractional Fourier functions. SIViP **8**(7), 1335–1344 (2012)
18. Yae, S., Ikehara, M.: Inverted residual Fourier transformation for lightweight single image deblurring. IEEE Access. **11**, 29175–29182 (2023)
19. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
20. Zhao, Z.Q., Zheng, P., Xu, S.T., et al. Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. 3212–3232 (2019)
21. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Conf. Comput. Vis. Pattern Recognit. **2014**, 580–587 (2014)
22. Girshick, R.: Fast R-CNN. In: Proceedings of the International Conference on Computer Vision, 2015
23. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell.Intell. **39**(6), 1137–1149 (2017)
24. Lin, T.-Y., Dollar, P., Girshick, R., et al. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936–944 (2017)

25. He, K., Gkioxari, G., Dollár, P., et al.: Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. (2017)
26. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2016**, 779–788 (2016)
27. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2017**, 6517–6525 (2017)
28. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv e-prints (2018)
29. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. (2020)
30. Li, C., Li, L., Jiang, H., et al.: YOLOv6: a single-stage object detection framework for industrial applications. arXiv preprint http://arxiv.org/abs/220902976 (2022)
31. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023
32. Roy, A.M., Bhaduri, J.: DenseSPH-YOLOv5: an automated damage detection model based on DenseNet and Swin-Transformer prediction head-enabled YOLOv5 with attention mechanism. Adv. Eng. Inform. **56** (2023)
33. Wang, W., Wu, B., Yang, S., et al.: Road damage detection and classification with faster R-CNN. In: Proceedings of the 2018 IEEE International Conference on Big Data (Big data), 2018. IEEE
34. Chen, Q., Gan, X., Huang, W., et al.: Road damage detection and classification using mask R-CNN with DenseNet backbone. Comput. Mater. Continua **65**(3), 2201–2215 (2020)
35. Haciefendioğlu, K., Başağa, H.B.: Concrete road crack detection using deep learning-based faster R-CNN method. Iran. J. Sci. Technol. Trans. Civ. Eng. 1–13 (2022)
36. Liu, Z., Yeoh, J.K.W., Gu, X., et al.: Automatic pixel-level detection of vertical cracks in asphalt pavement based on GPR investigation and improved mask R-CNN. Autom. Constr. **146** (2023)
37. Shen, T., Nie, M.: Pavement damage detection based on cascade R-CNN. In: Proceedings of the Proceedings of the 4th International Conference on Computer Science and Application Engineering, 2020
38. Li, S., Huang, Y.: Damage detection algorithm based on faster-RCNN. In: Proceedings of the 2023 5th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT), 2023. IEEE
39. Ding, W., Zhao, X., Zhu, B., et al.: An ensemble of one-stage and two-stage detectors approach for road damage detection. In: Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), 2022 [C]. IEEE
40. Tran, T.S., Nguyen, S.D., Lee, H.J., et al.: Advanced crack detection and segmentation on bridge decks using deep learning. Constr. Build. Mater. **400**, 132839 (2023)
41. Sami, A.A., Sakib, S., Deb, K., et al.: Improved YOLOv5-based real-time road pavement damage detection in road infrastructure management. Algorithms **16**(9), 452 (2023)
42. Wang, X., Gao, H., Jia, Z., et al.: BL-YOLOv8: an improved road defect detection model based on YOLOv8. Sensors **23**(20), 8361 (2023)
43. Alfarrarjeh, A., Trivedi, D., Kim, S.H., et al.: A deep learning approach for road damage detection from smartphone images. In: Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), 2018. IEEE
44. Guo, G., Zhang, Z.: Road damage detection algorithm for improved YOLOv5. Sci. Rep. **12**(1), 15523 (2022)
45. Inam, H., Islam, N.U., Akram, M.U., et al.: Smart and automated infrastructure management: a deep learning approach for crack detection in bridge images. Sustainability **15**(3) (2023)
46. Ren, M., Zhang, X., Chen, X., et al.: YOLOv5s-M: a deep learning network model for road pavement damage detection from urban street-view imagery. Int. J. Appl. Earth Observ. Geoinf. **120** (2023)
47. Du, Y., Zhong, S., Fang, H., et al.: Modeling automatic pavement crack object detection and pixel-level segmentation. Autom. Constr. **150** (2023)
48. Arya, D., Maeda, H., Ghosh, S.K., et al.: Crowdsensing-based road damage detection challenge (CRDDC'2022). In: Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), 2022. IEEE
49. Zhang, H., Cisse, M., Dauphin, Y.N., et al.: mixup: beyond empirical risk minimization. arXiv preprint http://arxiv.org/abs/171009412 (2017)
50. Terven, J., Cordova-Esparza, D.: A comprehensive review of YOLO: from YOLOv1 and beyond. arXiv 2023. arXiv preprint http://arxiv.org/abs/230400501
51. Koonce, B., Koonce, B.: MobileNetV3. In: Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, pp. 125–44 (2021)
52. Tan, M., Le, Q.: Efficientnetv2: smaller models and faster training. In: Proceedings of the International Conference on Machine Learning, 2021. PMLR
53. Koonce, B., Koonce, B.: EfficientNet. In: Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, pp. 109–23 (2021)
54. Howard, A.G., Zhu, M., Chen, B., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint http://arxiv.org/abs/170404861 (2017)
55. Sandler, M., Howard, A., Zhu, M., et al.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018
56. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017
57. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018
58. Wang, Q., Wu, B., Zhu, P., et al.: ECA-Net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020
59. Sun, P., Zhang, R., Jiang, Y., et al.: Sparse R-CNN: end-to-end object detection with learnable proposals. In: Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021
60. Wang, G., Chen, Y., An, P., et al.: UAV-YOLOv8: a small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. Sensors **23**(16), 7190 (2023)
61. Zheng, X., Qian, S., Wei, S., et al.: The combination of transformer and you only look once for automatic concrete pavement crack detection. Appl. Sci. **13**(16), 9211 (2023)
62. Wu, Y., Han, Q., Jin, Q., et al.: LCA-YOLOv8-Seg: an improved lightweight YOLOv8-Seg for real-time pixel-level crack detection of dams and bridges. Appl. Sci. **13**(19), 10583 (2023)
63. Yang, L., Yan, J., Li, H., et al.: Real-time classification of invasive plant seeds based on improved YOLOv5 with attention mechanism. Diversity **14**(4), 254 (2022)

64. Huang, Y., He, J., Liu, G., et al.: YOLO-EP: a detection algorithm to detect eggs of *Pomacea canaliculata* in rice fields. Ecol. Inform. **77**, 102211 (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.