IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 September 2023, accepted 6 October 2023, date of publication 7 November 2023, date of current version 10 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3330843

RESEARCH ARTICLE

Tunnel Lining Multi-Defect Detection Based on an Improved You Only Look Once Version 7 Algorithm

SONG JUAN, HE LONG-XI, AND LONG HUI-PING
School of Civil and Architectural Engineering, Shaoyang University, Shaoyang 422000, China
Corresponding author: Song Juan (SongJSYuniversity@163.com)

This work was supported by the General Project of Hunan Provincial Education Department: Research on Green Construction Methods Based on Low Carbon Emission Reduction in Southwest Hunan Province under Grant 20C1658.

ABSTRACT In the domain of tunnel lining defect detection, object detection algorithms have been widely employed. However, existing algorithms suffer from inadequate extraction of global information and low detection accuracy. To address these issues, a novel algorithm called Tunnel Defect Detection You Only Look Once (TDD-YOLO) is proposed, leveraging the YOLOv7 framework. The TDD-YOLO algorithm incorporates several enhancements to improve global and local information extraction capabilities, thereby enhancing defect detection accuracy. Firstly, MobileViT is utilized as the backbone feature extraction network, augmenting the network's ability to extract comprehensive information from both global and local contexts. Secondly, a Coordinate Attention (CA) module is introduced after the upsampling and downsampling stages of the feature pyramid network. This module highlights defect-related features while eliminating background interference. Lastly, a convolutional module called TP Block is devised to further enhance the network's feature extraction capability with reduced computational complexity. To validate the effectiveness of the proposed algorithm, a comparative analysis is conducted against five existing algorithms: SSD, Faster-RCNN, EfficientDet, YOLOv5, and YOLOv7. Experimental results demonstrate that the TDD-YOLO algorithm achieves superior performance with an F1 score of 77.43% and a mean Average Precision (mAP) of 77.52%. These results surpass those of the other five algorithms, establishing the TDD-YOLO algorithm as the most accurate and suitable solution for defect detection tasks in tunnels.

INDEX TERMS Tunnel defects, deep learning, defect recognition, object detection, neural network.

I. INTRODUCTION

As of the end of 2021, China operated 23,268 road tunnels with a total length of 24,698.9 km and 17,532 railroad tunnels with a total length of 21,055 km, showcasing China's emergence as a significant force in tunnel infrastructure [1].

Tunnel lining plays a crucial role in safeguarding the stability and safety of tunnel structures. However, over time, tunnel linings can develop various surface defects such as cracks, lining detachment, and water leakage due to geological, hydrological, and operational conditions [2]. The prompt and accurate identification of these surface defects is vital for tunnel maintenance and safety management [3]. Traditional methods for identifying tunnel lining defects rely heavily on manual experience and subjective judgment, resulting in low identification efficiency and susceptibility to human factors [4], [5], [6]. Moreover, minor defects may elude human eyes, further diminishing the accuracy and timeliness of defect identification [7].

With the rapid advancements in deep learning algorithms [8], intelligent recognition of tunnel lining surface defects based on deep learning has emerged as a promising research direction. Deep learning algorithms surpass traditional methods by overcoming the limitations of human-designed defect features and achieving more accurate and comprehensive defect recognition through learning implicit features in the

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

data. These algorithms exhibit strong generalization capabilities and can handle images at various scales, angles, and lighting conditions, thus enhancing recognition robustness, making them well-suited for tunnel surface defect detection [9], [10], [11], [12], [13].

In recent years, numerous scholars have utilized the powerful data mining capabilities of deep learning algorithms to achieve intelligent identification of structural defects. For instance, Savino and Tondolo [14] employed the GoogLeNet network to automate accurate classification of a wide range of defects in bridges, tunnels, and pavements. Dung and Anh [15] proposed a concrete crack detection method based on the fully convolutional network (FCN), which demonstrated excellent detection results on a self-constructed defect dataset. Han et al. [16] developed a pavement defect detection system using the You Only Look Once version 2 (YOLOv2) network to automate the identification of cracks in pavement images. Cheng and Wang [17] achieved high accuracy identification of cracks and water leakage in tunnels using the Faster Region Convolutional Neural Network (Faster RCNN) two-stage network.

While deep learning algorithms offer powerful feature learning capabilities, their effectiveness heavily relies on the quality of the dataset used for training. Models trained on simple datasets may exhibit lower accuracy when applied to images with complex backgrounds. Scholars have observed that directly applying existing deep learning algorithms to the task of tunnel defect identification is not entirely suitable, as further improvements are required to adapt to the complex environmental disturbances in tunnels and the multi-scale characteristics of tunnel defects [18], [19]. To enhance the accuracy and speed of tunnel surface defect identification in complex environments, researchers have explored improved network architectures. For instance, Zhu et al. [20] enhanced the inception module and network structure of the existing GoogLeNet network by introducing a new convolutional kernel. The resulting tunnel lining defect classification model achieved an accuracy of over 95%. Liu et al. [21] utilized a proposed image enhancement algorithm to improve defect images, subsequently applying the Faster RCNN network for defect detection. They found that the image enhancement algorithm effectively enhanced the network's defect recognition accuracy. Zhou et al. [22] improved the YOLOv4 network and introduced a new tunnel defect detection algorithm called YOLOv4-ED. They evaluated this algorithm on a self-built multi-defect dataset, demonstrating excellent detection results. Li et al. [23] proposed the integration of an adaptive spatial feature fusion module into the YOLOv5 framework to enhance the network's capability to detect tunnel defects. They further utilized network pruning and knowledge distillation techniques to balance detection performance and efficiency, resulting in high-accuracy detection of tunnel defects. Liao et al. [24] proposed a lightweight CNN called LinkCrack, in which the encoder uses the ResNet-34 network and the decoder uses up-sampling and convolution to achieve feature fusion of different scales of feature maps, realizing high-precision segmentation of tunnel lining cracks.

II. CONTRIBUTION

Despite the significant progress made in practical applications of tunnel lining surface defect identification using deep learning algorithms, there are still several shortcomings that need to be addressed, which can be summarized as follows:

(1) Lack of global information: Existing tunnel surface defect recognition algorithms based on convolutional neural networks often focus on extracting local information from the defect images, resulting in limited capability to extract global information. As a result, minor defects may be overlooked by these algorithms.

(2) Poor resistance to interference: Deep learning algorithms may struggle with robustness and generalization when faced with varying lighting conditions, scale changes, noise, and occlusion in the tunnel environment. This limitation reduces the effectiveness of defect identification under different conditions.

(3) Limitations in the number of tunnel defect images: Acquiring a large-scale dataset of tunnel lining surface defect images is challenging and requires substantial support from tunnel engineering projects. The scarcity of sufficient defect image data can hinder the detection performance of the models.

To address these shortcomings, this paper proposes a novel algorithm called TDD-YOLO for precise identification of tunnel lining surface defects. The main contributions of this study are as follows:

(1) Mobile-friendly Vision Transformer (MobileViT) network is utilized as the backbone feature extraction network in TDD-YOLO. This choice fully exploits the advantages of the network and enables the comprehensive extraction of both global and local information from the input image.

(2) The TP Block is introduced to facilitate fast and effective feature extraction from input images. Additionally, the Coordinate Attention (CA) mechanism is incorporated to enhance the network's resistance to interference in complex tunnel environments, improving its robustness and generalization ability.

(3) A high-quality image dataset comprising three types of surface defects (cracks, water leakage, and lining detachment) in tunnels is constructed. This dataset serves as the foundation for the algorithm experiments conducted in this paper.

(4) Among the tunnel defect datasets constructed in this study, the TDD-YOLO network demonstrates the best overall performance and achieves high accuracy in identifying tunnel surface defects.

By addressing the identified shortcomings and leveraging the proposed algorithm, the TDD-YOLO network presents a promising solution for accurate tunnel surface defect identification.

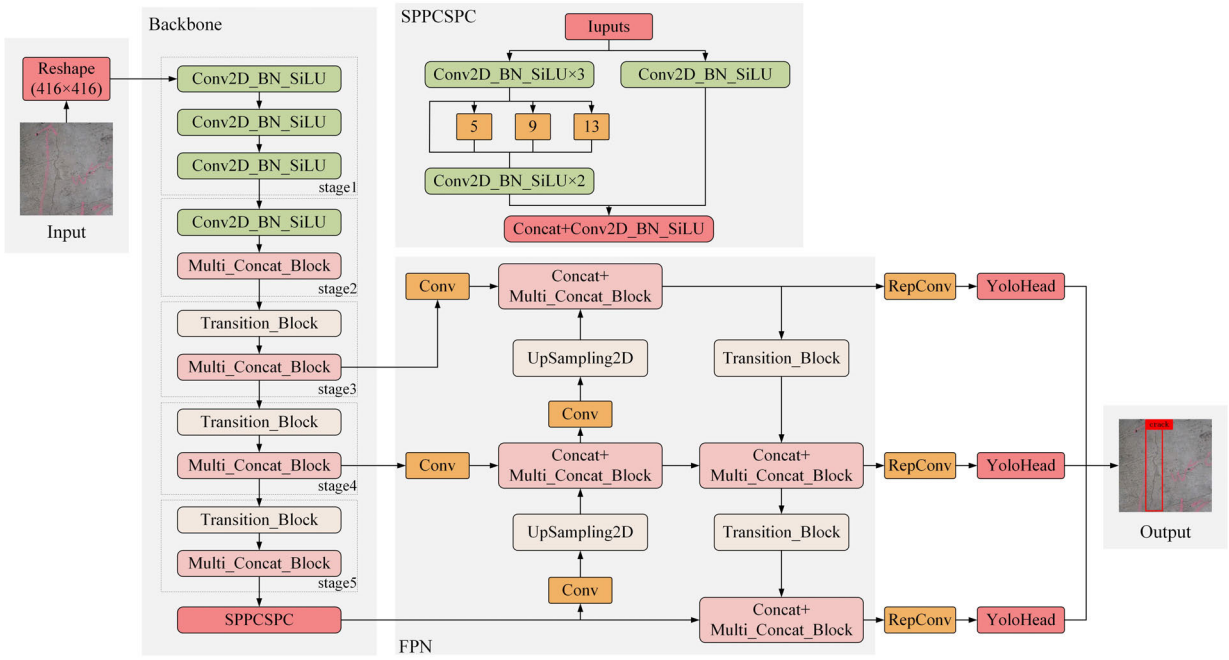


FIGURE 1. Structure of YOLOv7 network.

III. TUNNEL SURFACE DEFECT DETECTION ALGORITHM

A. YOLOV7 ALGORITHM

In YOLOv7 [25], the network architecture consists of three main components: the backbone feature extraction network, the feature pyramid network, and the output network. Figure 1 illustrates the overall structure of YOLOv7. Before inputting the images into the backbone feature extraction network, a pre-processing step is performed to resize the images uniformly. Commonly used image sizes for the YOLOv7 network are 416×416 and 640×640.

The primary purpose of the backbone feature extraction network is to extract relevant features from the input image. YOLOv7 incorporates a more complex backbone network architecture compared to previous versions such as YOLOv4 and YOLOv5, which enables improved feature extraction for images. When an image is fed into the backbone feature extraction network, it undergoes multiple convolution operations, normalization processes, and activation functions to extract meaningful features. In contrast to earlier YOLO iterations, YOLOv7 introduces multi-concat blocks and transition blocks, which further enhance the network's recognition accuracy. As a result, the backbone feature extraction network generates three feature layers with dimensions of 13×13, 26×26, and 52×52, respectively.

The feature pyramid network in YOLOv7 performs additional feature extraction on the three feature layers obtained from the backbone network and combines features of different scales to extract more effective features. The feature pyramid network outputs three feature layers with sizes of 13×13, 26×26, and 52×52, respectively, after performing the feature extraction process.

The output network takes these three feature layers from the feature pyramid network as input and predicts the type, confidence, and location of the target objects. For each feature layer, the network distinguishes the three anchors associated with it to determine whether they contain the target objects or not. It adjusts the anchors and removes redundant bounding boxes using a non-maximum suppression method to obtain the final prediction results [26], [27], [28].

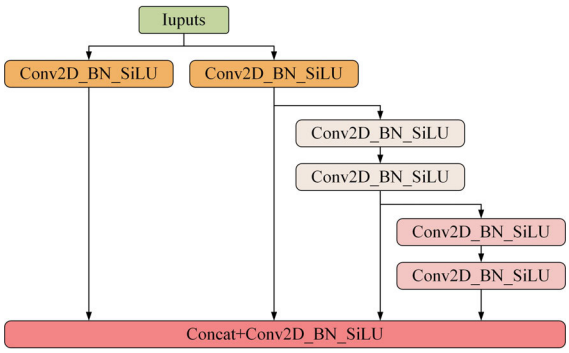


FIGURE 2. Multi-concat block.

The output network incorporates the RepConv structure, inspired by the RepVGG network, before generating the prediction results. During network training, a special residual structure is introduced to assist in training. However, during the prediction phase, the complex residual structure can be simplified to an ordinary 3×3 convolution. This simplification reduces the network complexity without sacrificing the prediction performance of the network.

Figure 2 illustrates the multi-concat block utilized in the YOLOv7 network. This block enables feature extraction through multiple branches and facilitates effective integration of features by stacking them together. This architecture increases the depth of the network, leading to improved prediction accuracy. Additionally, the utilization of skip connections helps mitigate the problem of gradient disappearance, which can occur with deep neural networks.

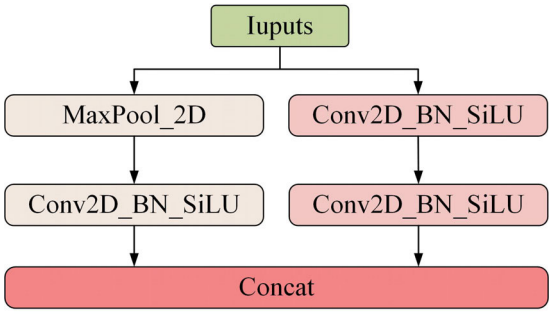


FIGURE 3. Transition block.

Figure 3 demonstrates the implementation of a transition block for downsampling within the YOLOv7 network. The left branch of the block consists of a 2×2 maximum pooling operation followed by a 1×1 convolution. On the other hand, the right branch comprises a 1×1 convolution followed by a 3×3 convolution with a stride size of 2×2. The outputs of these two branches are stacked together to achieve downsampling of the feature layer. These architectural components contribute to the overall effectiveness and performance of the YOLOv7 network [29], [30].

B. MOBILEVIT ALGORITHM

Convolutional neural networks (CNNs) excel at capturing local feature information within input data and offer advantages in terms of computational speed. However, their limitation lies in the inability to effectively extract global feature information [31]. On the other hand, transformer-based networks, such as the Transformer series, can capture global contextual information by leveraging the self-attention mechanism, enabling information aggregation from all positions in the input sequence. Nonetheless, transformer networks often require significant computational resources, have slower inference speeds, and can be more challenging to train. MobileViT [32] is a network that combines the strengths of both CNN and transformer architectures to extract both global and local feature information from images simultaneously. By integrating CNN, MobileViT accelerates network convergence and enhances the stability of the training process. Figure 4 provides an illustration of the MobileViT network architecture, wherein the MV2 structure represents the Inverted Residual block structure employed in the MobileNet network [33]. MobileViT's hybrid architecture offers a balance between efficient computation and capturing global contextual information, making it a suitable choice for various applications.

In MobileViT network, the input image goes through multiple MobileViT Block structures and MV2 structures for feature extraction. In MobileViT Block, a 3×3 convolution and a 1×1 convolution are used to achieve local feature extraction, then unfold, transformer and fold are used to complete global feature extraction, and finally the number of channels is adjusted by convolution to get the output feature layer. Among them, the unfolding structure is to divide the feature layer into multiple 2×2 sized pixel blocks and associate the pixels at corresponding positions in each block for self-attentive feature extraction in Transformer, and the folding structure is opposite to the unfolding structure. In the MV2 structure, feature extraction is performed through a series of convolution, normalization and activation functions.

C. COORDINATE ATTENTION

Existing attention mechanisms, such as CBAM and SE, typically employ global maximum pooling or global average pooling operations, which can result in the loss of spatial information. In contrast, the Coordinate Attention (CA) module [34] incorporates location information into channel attention, allowing for the consideration of both channel and location information. This integration effectively increases the emphasis on the target to be recognized within the image. The principle of the CA module is depicted in Figure 5. The CA module consists of two parallel stages. In the first stage, the input feature layers are globally averaged pooled along the height and width directions. These pooled features are then combined, transposed to match the same dimensions, and stacked. Subsequently, convolution, normalization, and activation functions are applied to obtain the intermediate feature representation. In the second stage, the intermediate feature representation is divided into two parallel stages. The number of channels is adjusted using a 1×1 convolution, and the Sigmoid function is applied to determine the attention distribution in both the height and width directions. Finally, the attention distribution is multiplied with the input feature layer to obtain the output feature layer. The CA module's architecture allows for the effective integration of channel and location information, enabling more precise attention and enhancing the network's performance in recognizing specific targets within the image.

D. TP BLOCK

To enhance the network's ability to rapidly and effectively extract defect image features, a novel convolutional block structure called TP Block is proposed, based on the concept of Partial Convolution (PConv) [35]. The TP Block consists of two components: a regular convolution block and a partial convolution block. The process begins with the input feature layer, which undergoes feature extraction using regular convolution, normalization, and activation functions. Next, fast feature extraction is performed using the PConv

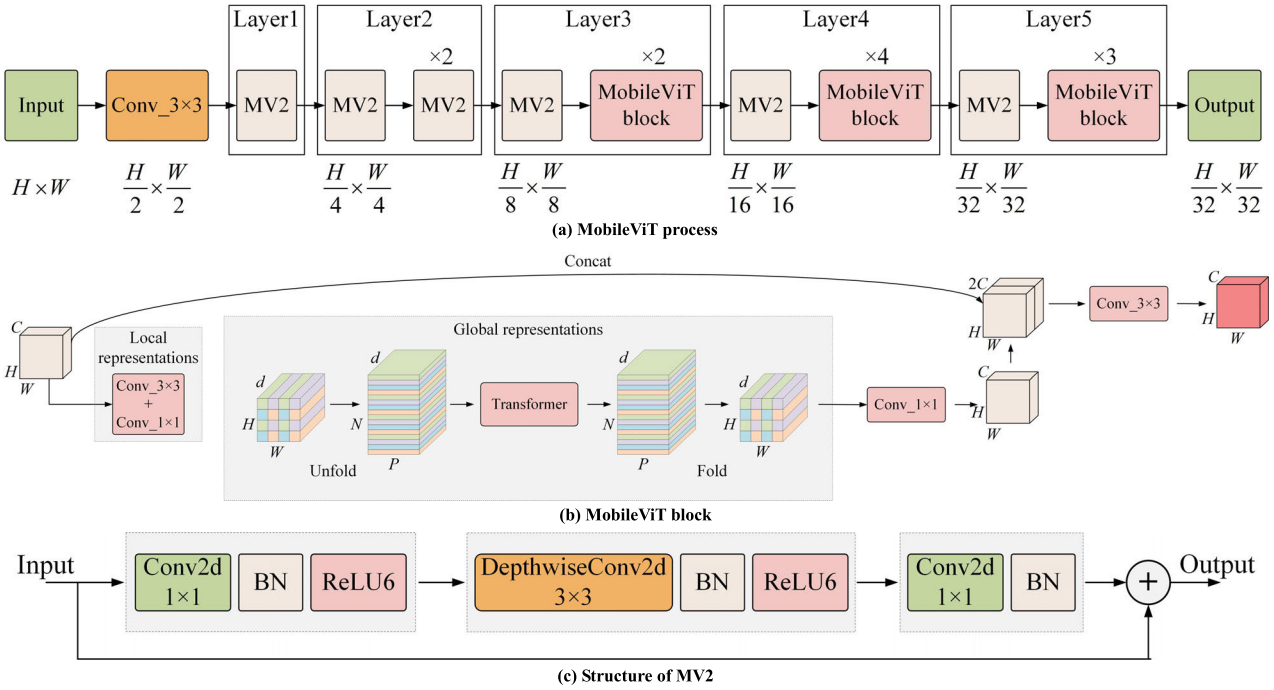


FIGURE 4. Structure of MobileViT network.

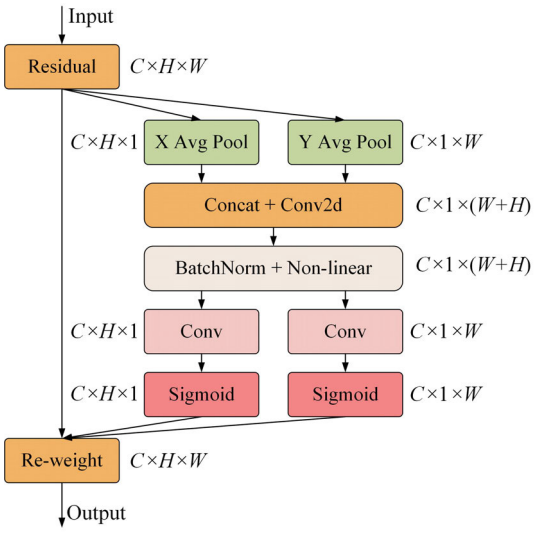


FIGURE 5. Coordinate attention module.

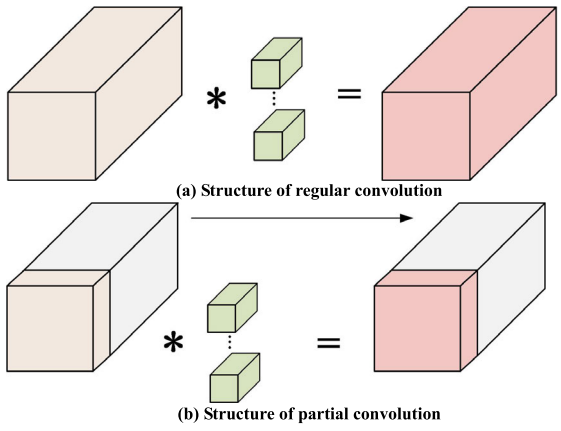


FIGURE 6. Structures of regular convolution and partial convolution.

operation, resulting in the output feature layer. The PConv operation conducts spatial feature extraction through regular convolution, but only on a portion of the input channels while keeping the remaining channels unchanged. This approach minimizes memory access and enables quick and efficient feature extraction. In comparison to regular convolution, the TP Block exhibits superior feature extraction capabilities and higher efficiency. The principles of regular convolution and partial convolution are depicted in Figure 6. The TP Block structure enhances the network's feature extraction capability, enabling it to efficiently capture relevant defect image features. It contributes to the overall performance improvement of the network in tunnel surface defect identification tasks.

E. TDD-YOLO ALGORITHM

The proposed TDD-YOLO tunnel defect detection algorithm is built upon the YOLOv7 network architecture, with several key modifications and additions. The algorithm utilizes the MobileViT network as the backbone network, incorporating the CA module after the upsampling and downsampling operations of the feature pyramid network, and introducing the new TP Block module. The algorithm's principle is illustrated in Figure 7. To enhance the dataset and reduce memory

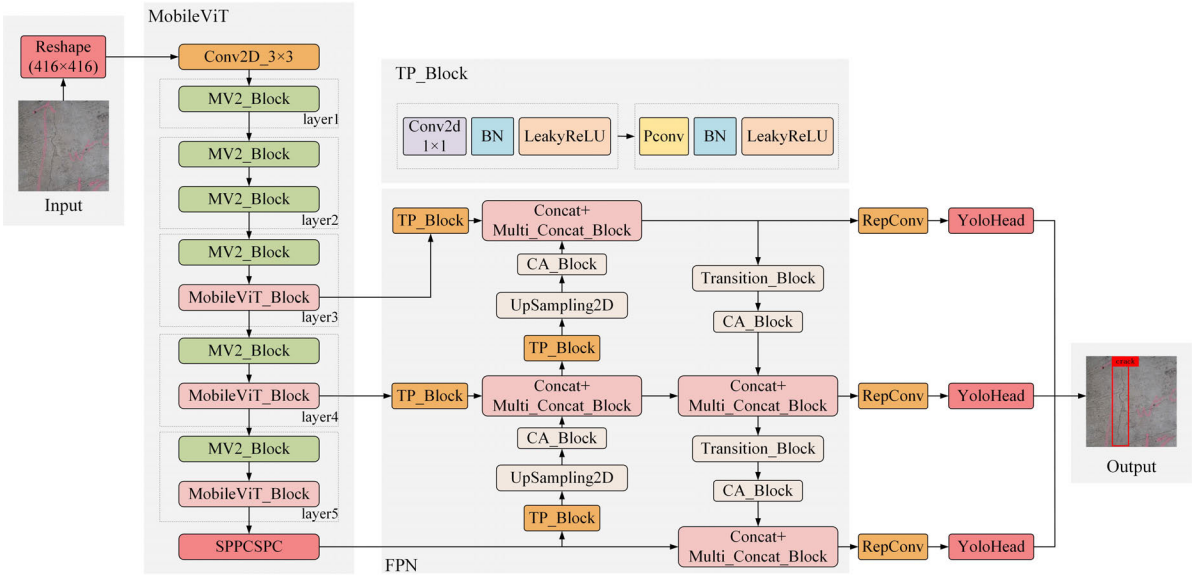


FIGURE 7. Structure of TDD-YOLO network.

requirements, two data enhancement techniques, Mosiac data enhancement [36] and mixup data enhancement [37], are applied prior to inputting the images into the backbone network. These techniques involve random cropping, stitching, and scaling operations to enrich the dataset. The input images are then size-normalized and uniformly resized to 416 × 416 pixels. It is important to note that direct resizing can distort the image, compromising recognition accuracy. To avoid distortion, the images are resized by scaling them equally and filling the remaining area with a solid color. The processed images are input into the MobileViT network for feature extraction, resulting in three feature layers: S1, S2, and S3, with dimensions of 52×52, 26×26, and 13×13, respectively. The S3 feature layer is further expanded using the SPCSPC module to enlarge the network's perceptual field, generating the P3 feature layer. Subsequently, feature layers S1, S2, and P3 are convolved, downsampled, and fused to obtain feature layers L1, L2, and L3. Next, a series of operations, including convolution, upsampling, downsampling, normalization, and activation functions, are applied for feature extraction and fusion, resulting in the final three feature layers: F1, F2, and F3. Finally, the RepConv module and YOLO head module decode these feature layers to produce the tunnel defect detection results for the input images. The proposed TDD-YOLO algorithm combines the strengths of MobileViT, CA module, and TP Block, offering improved feature extraction, robustness, and accuracy for tunnel defect detection tasks.

IV. TUNNEL SURFACE DEFECT DATASET

A. DATASET COMPOSITION

Collecting a dataset of more than 1400 images for tunnel defect detection is a significant accomplishment. The dataset includes three types of tunnel surface defects: cracking, water leakage, and lining falling off. Some of the collected images are depicted in Figure 8, providing a visual representation of the dataset. To enhance the robustness of the training model and improve its adaptability to the complex tunnel environment, various methods were employed to augment the dataset. These methods include rotation, cropping, noise injection, and fuzzing, as illustrated in Figure 9. Data augmentation techniques like these help increase the diversity of the dataset, allowing the model to learn from a wider range of scenarios and improve its generalization ability.

TABLE 1. Tunnel defects dataset.

Defect categories	Total amount	Training set	Validation set	Test set
crack	1102	742	180	180
water leakage	1004	688	158	158
lining falling off	698	470	114	114

To ensure proper evaluation and comparison of the model's performance, the dataset was divided into three sets: a training set, a validation set, and a test set, following a 6:2:2 ratio. This partitioning scheme allows for model training on the training set, hyperparameter tuning on the validation set, and unbiased evaluation on the test set.

Table 1 provides the distribution of defect images in each category, indicating the number of images available for training and testing for each type of tunnel surface defect.

FIGURE 8. Example of tunnel surface defect images.

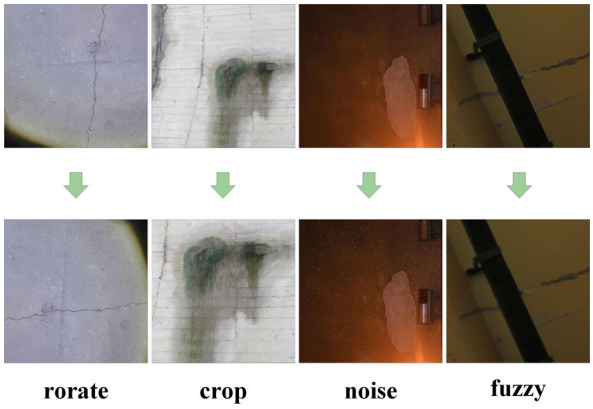


FIGURE 9. Defect image enhancement methods.

B. DATASET ANNOTATION

The usage of the LabelImg annotation software to annotate the tunnel defect images and convert them to the VOC (Visual Object Classes) format is a common practice in object detection tasks. The annotation process involves marking the bounding box coordinates and assigning the corresponding class label to each defect in the image. The labeling content includes two main components:

1.Label frame coordinates: These coordinates represent the position and size of the bounding box around each defect in the image. They provide the spatial information necessary for the model to locate and recognize the defects accurately.

2.Classification information: This information indicates the class of each defect, distinguishing between different types of tunnel surface defects. In Figure 10, the labels ''C'' indicate cracks, ''W'' indicates water leakage, and ''L'' indicates lining falling off.

Figure 10 presents a selection of annotated images, showcasing the labeled bounding boxes and corresponding class labels for the detected defects. These annotations serve as ground truth data for training and evaluating the network.

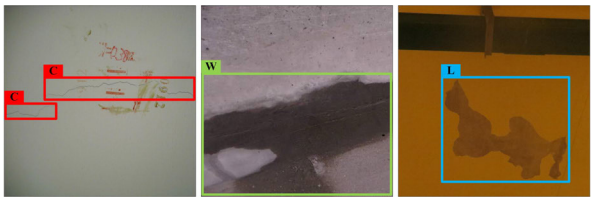


FIGURE 10. Labeling method of tunnel lining defects.

V. EXPRIMENTS ON TUNNEL SURFACE DEFECT DETECTION

The experimental setup in this paper includes four main parts:

1.Model Training: The networks used in the experiments are trained using the provided dataset. During training, the loss on both the training set and the validation set is monitored to ensure that the networks are not overfitting. The best-performing weights are selected for each network to be used in the subsequent experiments.

2.Backbone Network Comparison Experiments: Several classification networks with excellent performance are chosen as the backbone networks for YOLOv7. The performance of each network is evaluated using the tunnel defect test set constructed in this paper. This comparison helps assess the impact of different backbone networks on the tunnel defect detection task.

3.Ablation Experiments: The improved methods proposed in this paper are randomly combined, and ablation experiments are conducted to analyze the effectiveness of these methods. By selectively removing or modifying specific components, the researchers can assess the contribution of each improvement and determine their impact on the overall performance.

4.Model Comparison Experiments: The performance of different networks, including the proposed TDD-YOLO algorithm, is evaluated using the defect test set constructed in this paper. This evaluation involves a comprehensive analysis and comparison of the different models, considering factors such as accuracy, precision, recall, and other relevant metrics.

By conducting these experiments, the researchers aim to demonstrate the effectiveness of their proposed algorithm and compare it with other existing models in terms of tunnel defect detection accuracy and overall performance.

A. EXPERIMENT ENVIRONMENT

The experiments in this paper are conducted with Python 3.8 computer language and Pytorch 1.10.0 module. The system used is Windows 10 and the GPU device is NVIDIA GeForce RTX 3060.

B. EVALUATION INDICATORS

In order to evaluate the performance of the model for tunnel multi-defect detection, Average Precision (AP), Mean Average Precision (mAP), the summed average of precision and recall (F1 score), Frames per second (FPS), and Model size (MS) are chosen as the evaluation indicators in this paper. The

TABLE 2. Network training hyperparameters.

Networks	Learning rate		Batch size		Momentum	Label smoothing
	Max	Min	Frozen	Unfrozen		
SSD	6×10^{-4}	6×10^{-6}	16	8	0.93	0
Faster RCNN	1×10^{-4}	1×10^{-6}	16	8	0.90	0
EfficientDet	3×10^{-4}	3×10^{-6}	32	16	0.90	0
YOLOv5	1×10^{-3}	1×10^{-5}	16	8	0.94	0.005
YOLOv7	1×10^{-3}	1×10^{-5}	16	8	0.94	0.005
TDD-YOLO	1×10^{-3}	1×10^{-5}	16	8	0.94	0.005

calculation formulas are shown as follows:

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

$$f1 = \frac{2 \times P \times R}{P + R} \tag{3}$$

$$F1 = \frac{\sum f1}{n} \tag{4}$$

$$AP = \int_0^1 P(R)dR \tag{5}$$

$$mAP = \frac{\sum AP}{n} \tag{6}$$

$$FPS = \frac{N}{T} \tag{7}$$

where TP represents the number of correctly identified defect targets, FP denotes the number of incorrectly identified defect targets, FN signifies the number of undetected defect targets, P is the precision rate, R is the recall rate, n is the number of defect types, while $P(R)$ represents a plot where the horizontal coordinate corresponds to R, and the vertical coordinate corresponds to P. N is the number of detecting images and T is the total time used to detect all the images.

C. MODEL TRAINING PARAMETERS

In the paper, five object detection networks (YOLOv7, YOLOv5, EfficientDet, Faster RCNN, and SSD) are selected for comparison. All networks are trained on the same dataset, and the training performance of each network is evaluated. The training process is conducted for 200 epochs to ensure convergence, and the training and validation errors are monitored to prevent overfitting. To accelerate convergence and save computational resources, the backbone network is initialized with pre-training weights obtained from training on the COCO dataset using transfer learning. The first 50 epochs use freeze training, where only the non-backbone parts of the network are trained, while the backbone network weights remain fixed. The subsequent 150 epochs use unfrozen training, where all weights of the network, including the backbone network, are trained and updated.

To enhance network performance and prevent training oscillation, a large training batch size is selected. The learning rate of the network is decayed using the cosine annealing learning rate schedule, which adjusts the learning rate based on a cosine function during training to prevent the network from getting trapped in local optima.

Mosaic and mixup data augmentation techniques are randomly applied during training to improve the network's adaptability to complex and variable environments. After completing the training process, the weight with the smallest validation error, while ensuring the network does not overfit, is selected as the optimal model weight for further analysis. The hyperparameters used in the network training process are specified in Table 2 of the paper.

D. BACKBONE NETWORK COMPARISON EXPERIMENTS

In this paper, the performance of different backbone feature extraction networks is evaluated by selecting various popular classification networks, including both convolutional neural networks and transformer series networks. The experiments are conducted using consistent hyperparameters, and the trained models are tested to select the best performing models for each network. From the results presented in Table 3, it can be observed that when five classification networks are used as the backbone for YOLOv7, MobileViT achieves the highest accuracy for crack and water leakage identification. Additionally, MobileViT and ResNet demonstrate the highest recognition precision for lining falling off, with MobileViT outperforming ResNet in terms of overall performance.

Considering the comprehensive model performance, MobileViT shows significant improvements over other networks. It achieves a 1.84% improvement in F1 score and a 3.61% improvement in mAP compared to MobileNet. Similarly, it outperforms GhostNet, ResNet, and Swin Transformer with improvements of 2.46%, 1.95%, and 1.45% in F1 score, and 4.88%, 1.53%, and 1.35% in mAP, respectively. Based on the comprehensive analysis, MobileViT is selected as the best backbone feature extraction network for YOLOv7 due to its superior performance across various evaluation metrics.

E. ABLATION EXPERIMENTS

To test the effectiveness of the MobileViT backbone network, CA mechanism, and TP Block used in this paper, ablation experiments are performed on the constructed tunnel

TABLE 3. Backbone network evaluation indicators.

Backbone networks	crack		water leakage		lining falling off		F1(%)	mAP(%)
	f1(%)	AP(%)	f1(%)	AP(%)	f1(%)	AP(%)		
MobileNet	74.43	75.54	74.85	75.74	65.66	62.33	71.65	71.20
GhostNet	73.64	73.60	74.12	75.31	65.32	60.87	71.03	69.93
ResNet	74.56	77.52	74.48	75.76	66.84	65.30	71.96	72.86
Swin transformer	74.48	76.97	75.03	78.61	66.91	64.49	72.14	73.36
MobileViT	75.83	79.58	76.32	79.62	68.32	65.24	73.49	74.81

TABLE 4. Results of network ablation experiments.

Networks	MobileViT	CA	TP Block	F1(%)	mAP(%)	FPS(frame/s)	MS(M)
YOLOv7	×	×	×	71.29	71.13	50.81	142.3
YOLOv7+MobileViT	√	×	×	73.49	74.81	51.39	113.9
YOLOv7+CA	×	√	×	71.55	71.71	51.43	142.6
YOLOv7+TP Block	×	×	√	72.42	72.53	51.26	142.6
YOLOv7+MobileViT+CA	√	√	×	74.61	75.72	52.50	114.2
YOLOv7+MobileViT+ TP Block	√	×	√	75.16	75.91	52.79	114.2
YOLOv7+CA+ TP Block	×	√	√	74.05	73.60	52.44	142.9
YOLOv7+MobileViT+CA+ TP Block	√	√	√	77.43	77.52	53.86	114.5

multi-defect dataset. The outcomes of the ablation experiments are displayed in Table 4. To compare the impact of different improvement strategies more clearly, F1, mAP, FPS, and MS are used as evaluation indicators in this experiment.

1) IMPACT OF SINGLE IMPROVEMENTS

The results presented in Table 4 highlight the effectiveness of the three improvement strategies individually applied to the YOLOv7 network. The following observations can be made:

1.When the MobileViT backbone network is used in YOLOv7 (row 3), there is a noticeable improvement of 2.20% in F1 score, 3.68% in mAP and 0.58 frame/s in FPS compared to the YOLOv7 network without any improvement strategy (row 2). This confirms that the MobileViT network effectively captures both global and local features of tunnel defect targets, leading to improved performance.

2.The inclusion of the CA modules after the upsampling and downsampling operations in the feature pyramid network (row 4) results in a modest improvement of 0.26% in F1 score, 0.58% in mAP and 0.62 frame/s in FPS. This indicates that the CA module effectively increases the weights of the tunnel defect target regions, enhancing the recognition precision of the network.

3.When the TP Block is introduced by replacing the convolution module in the feature pyramid network (row 5), there is a significant improvement of 1.13% in F1 score. 1.40% in mAP and 0.45 frame/s in FPS. This demonstrates that the TP Block improvement strategy effectively enhances the feature extraction ability of the model for tunnel surface defects, without significantly increasing the model parameters.

Overall, each of the three improvement strategies applied individually contributes to the improvement of feature extraction ability, recognition precision and speed of the network to some degree. The MobileViT backbone network improves both global and local feature extraction, the CA module enhances the weights of defect regions, and the TP Block improves feature extraction ability specifically for surface defects. The results emphasize the effectiveness of these strategies and their positive impact on the overall performance of the model.

2) IMPACT OF JOINT IMPROVEMENTS

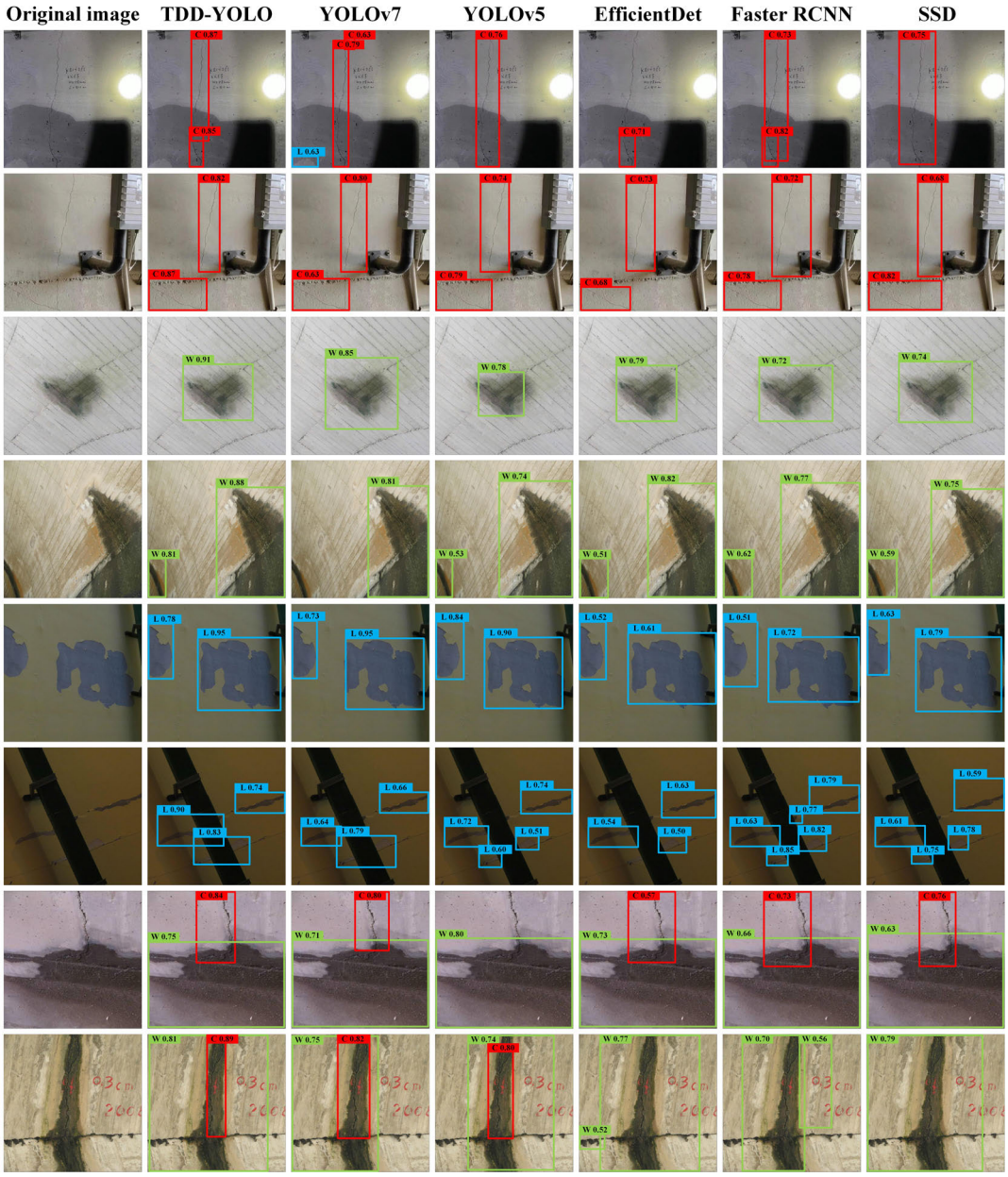
The joint impact of the three improvement strategies on the model performance is evaluated in rows 6-9 of Table 4. The results highlight the combined effect of incorporating the MobileViT network, CA module, and TP Block module in the YOLOv7 network. The following observations can be made:

1.When the MobileViT network and CA module are combined (row 6), there is a substantial improvement of 3.32% in F1 score, 4.59% in mAP and 1.69 frame/s in FPS compared to the YOLOv7 network without any improvement strategy (row 2). This demonstrates the complementary benefits of using both the MobileViT network for feature extraction and the CA module for enhancing defect region weights.

2.Similarly, when the MobileViT network and TP Block module are combined (row 7), there is a significant

TABLE 5. Model comparison experiment results.

Networks	crack		water leakage		lining falling off		F1(%)	mAP(%)
	f1(%)	AP(%)	f1(%)	AP(%)	f1(%)	AP(%)		
SSD	69.43	69.97	71.80	74.00	44.31	42.98	61.85	62.32
Faster RCNN	64.92	71.54	70.57	72.21	44.72	46.08	60.07	63.28
EfficientDet	64.42	70.78	74.93	76.02	56.37	57.43	65.24	68.08
YOLOv5	74.72	73.83	73.49	72.40	65.12	64.02	71.11	70.08
YOLOv7	74.97	76.53	73.55	74.71	65.35	62.15	71.29	71.13
TDD-YOLO	80.47	82.28	79.59	80.58	71.74	69.71	77.43	77.52



(a) Results under simple bright environment

FIGURE 11. Defect detection results.

(b) Results under complex dim environment

FIGURE 11. *(Continued.)* **Defect detection results.**

improvement of 3.87% in F1 score, 4.78% in mAP and 1.98 frame/s in FPS. This indicates that the MobileViT network, along with the TP Block module, effectively enhances the feature extraction ability of the model for tunnel surface defects.

3.Introducing the CA module and TP Block module together (row 8) results in a notable improvement of 2.76% in F1 score, 2.47% in mAP and 1.63 frame/s in FPS. This confirms that the joint utilization of the CA module and TP Block module enhances both defect region weights and feature extraction ability, leading to improved performance.

4.Combining all three improvement strategies simultaneously (row 9) yields the most significant improvement in model performance. The F1 score improves by 6.14%, the mAP improves by 6.39% and the FPS improves by 3.05 frame/s compared to the YOLOv7 network without any improvement strategy (row 2). This demonstrates the synergistic effect of integrating the MobileViT network, CA module, and TP Block module, resulting in substantial performance enhancements.

Overall, the joint improvement experiments confirm the effectiveness of the three improvement strategies used in this

paper. The combination of the MobileViT network, CA module, and TP Block module leads to significant improvements in feature extraction ability, recognition precision and speed, and overall model performance for tunnel multi-defect detection.

F. MODEL COMPARISON EXPERIMENTS

The comparison experiments in Table 5 evaluate the performance of the TDD-YOLO network in comparison to five other object detection models (SSD, Faster RCNN, EfficientDet, YOLOv5, and YOLOv7) on the test set constructed in this paper. The results highlight the superior performance of the TDD-YOLO model in terms of recognition accuracy for the three surface defects (cracking, water leakage, and lining falling off). The evaluation indicators, F1 score and mAP, show significant improvements compared to the other models.

The TDD-YOLO model achieves an F1 score of 77.43%, which is higher than the F1 scores of the other models by margins of 15.58% (SSD), 17.36% (Faster RCNN), 12.19% (EfficientDet), 6.32% (YOLOv5), and 6.14% (YOLOv7). Similarly, the mAP of the TDD-YOLO model is 77.52%, surpassing the mAP values of the other models by 15.20% (SSD), 14.24% (Faster RCNN), 9.44% (EfficientDet), 7.44% (YOLOv5), and 6.39% (YOLOv7).

These results clearly demonstrate the effectiveness of the three improvement strategies (MobileViT backbone network, CA module, and TP Block module) implemented in the TDD-YOLO model. These strategies enhance the feature extraction capabilities of the network for detecting tunnel defects, resulting in improved performance in terms of defect recognition accuracy. The TDD-YOLO model outperforms the other models in terms of both F1 score and mAP, indicating its superiority in tunnel defect detection tasks.

Figure 11 illustrates the detection results of the six models (TDD-YOLO, YOLOv7, YOLOv5, EfficientDet, Faster RCNN, and SSD) for tunnel lining surface defect images. The images are categorized into crack, water leakage, lining falling off, and mixed defect images. The detection results are shown in columns 2 to 7, with each column representing a different model.

In Figure 11(a), under simple bright environment conditions, all six models accurately locate individual defect images with minimal wrong detections and omissions. However, for the detection of mixed defect images, all five comparison models exhibit omissions. The TDD-YOLO model demonstrates the highest detection confidence and regional localization precision for defect targets among both single and mixed defect images. This indicates that the TDD-YOLO model achieves high accuracy detection of defect images by effectively distinguishing different types of tunnel surface defects in a simple environment.

Figure 11(b) presents the analysis of the detection results under complex dim environment conditions, where the detection precision of the five comparison models significantly decreases due to the interference of dim lights and clutter. Misdetections and omissions occur in both single and mixed defect images. For instance, in the seventh row of Figure 11(b), the contrast models struggle to accurately identify cracking and water leakage. The TDD-YOLO model, while experiencing a slight decrease in detection precision compared to the simple environment, still achieves precise location of the defect area. It effectively distinguishes different types of surface defects in mixed defect images, demonstrating its capability to resist the interference of complex tunnel environments and achieve precise detection of multiple surface defects.

In conclusion, the TDD-YOLO model proposed in this paper can accurately distinguish various types of tunnel surface defects and achieve high-accuracy detection in both simple bright and complex dim environments. It is suitable for detecting multiple lining surface defects in tunnels under complex environmental conditions.

VI. CONCLUSION

Based on the YOLOv7 network framework, this paper introduces several improvements to achieve accurate detection of multiple surface defects in tunnels, leading to the development of the TDD-YOLO tunnel lining surface defect detection algorithm. The main conclusions of the paper can be summarized as follows:

1. Backbone network selection: The experiment demonstrates that the MobileViT classification network, combining the advantages of convolutional networks and transformer networks, outperforms other popular networks such as MobileNet, GhostNet, ResNet, and Swin transformer in terms of evaluation indicators. Therefore, MobileViT is chosen as the backbone network for the TDD-YOLO model.
2. Ablation experiment analysis: The ablation experiments confirm the effectiveness of the MobileViT network, CA module, and TP Block module. Each of these improvements contributes to an improvement in the F1 score and mAP of the model, enhancing its defect detection performance to some degree.
3. Performance comparison: The TDD-YOLO model achieves an F1 score of 77.43% and an mAP of 77.52%, surpassing the performance of the five comparison models (SSD, Faster RCNN, EfficientDet, YOLOv5, and YOLOv7) in various evaluation indicators. This demonstrates that the TDD-YOLO model exhibits superior defect detection accuracy compared to existing detection models.
4. Robustness in complex environments: The TDD-YOLO model demonstrates its ability to resist interference in complex tunnel environments. It maintains high accuracy in detecting multiple surface defects even under challenging conditions of low light and environmental interference. Therefore, the TDD-YOLO model is well-suited for the task of detecting multiple defects in tunnels with complex environmental conditions.

In summary, the TDD-YOLO algorithm, incorporating the MobileViT backbone network, CA module, and TP Block module, enhances the defect detection performance and achieves accurate detection of multiple surface defects in tunnels, making it a valuable approach for real-world applications.

AUTHOR CONTRIBUTIONS

Song Juan: Methodology, software, data processing, writing, original draft. He Long-Xi: Conceptualization, methodology, supervision, project administration. Long Hui-Ping: Validation, formal analysis, writing-review & editing.

ACKNOWLEDGMENT

The authors thank Advanced Shaoyang University, for providing the experiment conditions. They also express special thanks to the editors and anonymous reviewers for their constructive comments.

CONFLICTS 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] M. Wei, Y. Song, X. Wang, and J. Peng, "Safety diagnosis of TBM for tunnel excavation and its effect on engineering," *Neural Comput. Appl.*, vol. 33, no. 3, pp. 997–1005, Feb. 2021, doi: 10.1007/s00521-020-05371-y.

[2] B.-G. He, R.-L. Zhen, T.-Y. Chen, and H.-P. Li, "Monitoring the behavior of subway tunnels during early operation in saturated soft strata," *Bull. Eng. Geol. Environ.*, vol. 81, no. 4, Apr. 2022, doi: 10.1007/s10064-022-02644-9.

[3] H.-W. Huang, Q.-T. Li, and D.-M. Zhang, "Deep learning based image recognition for crack and leakage defects of metro shield tunnel," *Tunnelling Underground Space Technol.*, vol. 77, pp. 166–176, Jul. 2018, doi: 10.1016/j.tust.2018.04.002.

[4] Y. Pan and L. Zhang, "Dual attention deep learning network for automatic steel surface defect segmentation," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 37, no. 11, pp. 1468–1487, Sep. 2022, doi: 10.1111/mice.12792.

[5] P. Chun, S. Izumi, and T. Yamane, "Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 1, pp. 61–72, Jan. 2021, doi: 10.1111/mice.12564.

[6] Y. Zhang, Z. Song, and W. Guo, "Multi-scale continuous gradient local binary pattern for leaky cable fixture detection in high-speed railway tunnel," *IEEE Access*, vol. 9, pp. 147102–147113, 2021, doi: 10.1109/ACCESS.2021.3124676.

[7] D. Ma, J. Liu, H. Fang, N. Wang, C. Zhang, Z. Li, and J. Dong, "A multi-defect detection system for sewer pipelines based on StyleGAN-SDM and fusion CNN," *Construct. Building Mater.*, vol. 312, Dec. 2021, Art. no. 125385, doi: 10.1016/j.conbuildmat.2021.125385.

[8] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010, doi: 10.1109/TPAMI.2008.275.

[9] Y. Xue, X. Cai, M. Shadabfar, H. Shao, and S. Zhang, "Deep learning-based automatic recognition of water leakage area in shield tunnel lining," *Tunnelling Underground Space Technol.*, vol. 104, Oct. 2020, Art. no. 103524, doi: 10.1016/j.tust.2020.103524.

[10] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, and X. Shen, "Image-based concrete crack detection in tunnels using deep fully convolutional networks," *Construct. Building Mater.*, vol. 234, Feb. 2020, Art. no. 117367, doi: 10.1016/j.conbuildmat.2019.117367.

[11] Y. Wang, T. Wang, X. Zhou, W. Cai, R. Liu, M. Huang, T. Jing, M. Lin, H. He, W. Wang, and Y. Zhu, "TransEffiDet: Aircraft detection and classification in aerial images based on EfficientDet and transformer," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, Apr. 2022, doi: 10.1155/2022/2262549.

[12] Z. Zhou, L. Yan, J. Zhang, and H. Yang, "Real-time tunnel lining crack detection based on an improved you only look once version X algorithm," *Georisk, Assessment Manage. Risk Engineered Syst. Geohazards*, vol. 17, no. 1, pp. 181–195, Jan. 2023, doi: 10.1080/17499518.2023.2172187.

[13] Z. Zhou, J. Zhang, and C. Gong, "Hybrid semantic segmentation for tunnel lining cracks based on Swin transformer and convolutional neural network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 38, no. 17, pp. 2491–2510, Nov. 2023.

[14] P. Savino and F. Tondolo, "Automated classification of civil structure defects based on convolutional neural network," *Frontiers Struct. Civil Eng.*, vol. 15, no. 2, pp. 305–317, Apr. 2021, doi: 10.1007/s11709-021-0725-9.

[15] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Autom. Construction*, vol. 99, pp. 52–58, Mar. 2019, doi: 10.1016/j.autcon.2018.11.028.

[16] Z. Han, H. Chen, Y. Liu, Y. Li, Y. Du, and H. Zhang, "Vision-based crack detection of asphalt pavement using deep convolutional neural network," *Iranian J. Sci. Technol., Trans. Civil Eng.*, vol. 45, no. 3, pp. 2047–2055, Sep. 2021, doi: 10.1007/s40996-021-00668-x.

[17] J. C. P. Cheng and M. Wang, "Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques," *Autom. Construct.*, vol. 95, pp. 155–171, Nov. 2018, doi: 10.1016/j.autcon.2018.08.006.

[18] Y. Pan and L. Zhang, "Mitigating tunnel-induced damages using deep neural networks," *Autom. Construction*, vol. 138, Jun. 2022, Art. no. 104219, doi: 10.1016/j.autcon.2022.104219.

[19] Y. Tan, R. Cai, J. Li, P. Chen, and M. Wang, "Automatic detection of sewer defects based on improved you only look once algorithm," *Autom. Construct.*, vol. 131, Nov. 2021, Art. no. 103912, doi: 10.1016/j.autcon.2021.103912.

[20] A. Zhu, S. Chen, F. Lu, C. Ma, and F. Zhang, "Recognition method of tunnel lining defects based on deep learning," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–12, Sep. 2021, doi: 10.1155/2021/9070182.

[21] J. Liu, Z. Zhao, C. Lv, Y. Ding, H. Chang, and Q. Xie, "An image enhancement algorithm to improve road tunnel crack transfer detection," *Construct. Building Mater.*, vol. 348, Sep. 2022, Art. no. 128583, doi: 10.1016/j.conbuildmat.2022.128583.

[22] Z. Zhou, J. Zhang, and C. Gong, "Automatic detection method of tunnel lining multi-defects via an enhanced you only look once network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 37, no. 6, pp. 762–780, May 2022, doi: 10.1111/mice.12836.

[23] Y. Li, T. Bao, T. Li, and R. Wang, "A robust real-time method for identifying hydraulic tunnel structural defects using deep learning and computer vision," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 38, no. 10, pp. 1381–1399, Jul. 2023, doi: 10.1111/mice.12949.

[24] J. Liao, Y. Yue, D. Zhang, W. Tu, R. Cao, Q. Zou, and Q. Li, "Automatic tunnel crack inspection using an efficient mobile imaging module and a lightweight CNN," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15190–15203, Sep. 2022, doi: 10.1109/TITS.2021.3138428.

[25] C. Ni, D. Wang, R. Vinson, M. Holmes, and Y. Tao, "Automatic inspection machine for maize kernels based on deep convolutional neural networks," *Biosyst. Eng.*, vol. 178, pp. 131–144, Feb. 2019, doi: 10.1016/j.biosystemseng.2018.11.010.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[27] L. Wen and K.-H. Jo, "Fast LiDAR R-CNN: Residual relation-aware region proposal networks for multiclass 3-D object detection," *IEEE Sensors J.*, vol. 22, no. 12, pp. 12323–12331, Jun. 2022, doi: 10.1109/JSEN.2022.3172446.

[28] C. Feng, H. Zhang, S. Wang, Y. Li, H. Wang, and F. Yan, "Structural damage detection using deep convolutional neural network and transfer learning," *KSCE J. Civil Eng.*, vol. 23, no. 10, pp. 4493–4502, Oct. 2019, doi: 10.1007/s12205-019-0437-z.

[29] Y. He, Z. Jin, J. Zhang, S. Teng, G. Chen, X. Sun, and F. Cui, "Pavement surface defect detection using mask region-based convolutional neural networks and transfer learning," *Appl. Sci.*, vol. 12, no. 15, p. 7364, Jul. 2022.

[30] Z. Wang and J. Zhang, "Incremental PID controller-based learning rate scheduler for stochastic gradient descent," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, Oct.2022, doi: 10.1109/TNNLS.2022.3213677.
[31] P. Dolata, M. Mrzyglód, and J. Reiner, "Double-stream convolutional neural networks for machine vision inspection of natural products," *Appl. Artif. Intell.*, vol. 31, nos. 7–8, pp. 643–659, Sep. 2017, doi: 10.1080/08839514.2018.1428491.
[32] G. Wen, Z. Gao, Q. Cai, Y. Wang, and S. Mei, "A novel method based on deep convolutional neural networks for wafer semiconductor surface defect inspection," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9668–9680, Dec. 2020, doi: 10.1109/TIM.2020.3007292.
[33] J. Cheng, L. Zhang, and Q. Chen, "Fast monocular visual-inertial initialization with an improved iterative strategy," *J. Sensors*, vol. 2021, pp. 1–18, May 2021, doi: 10.1155/2021/5565158.
[34] R. Mo, S. Lai, Y. Yan, Z. Chai, and X. Wei, "Dimension-aware attention for efficient mobile networks," *Pattern Recognit.*, vol. 131, Nov. 2022, Art. no. 108899, doi: 10.1016/j.patcog.2022.108899.
[35] T. L. McKinney and M. J. Euler, "Neural anticipatory mechanisms predict faster reaction times and higher fluid intelligence," *Psychophysiology*, vol. 56, no. 10, Oct. 2019, Art. no. e13426, doi: 10.1111/psyp.13426.
[36] Z. Meng, S. Xu, L. Wang, Y. Gong, X. Zhang, and Y. Zhao, "Defect object detection algorithm for electroluminescence image defects of photovoltaic modules based on deep learning," *Energy Sci. Eng.*, vol. 10, no. 3, pp. 800–813, Mar. 2022, doi: 10.1002/ese3.1056.
[37] Z. Xiong, T. Song, S. He, Z. Yao, and X. Wu, "A unified and costless approach for improving small and long-tail object detection in aerial images of traffic scenarios," *Int. J. Speech Technol.*, vol. 53, no. 11, pp. 14426–14447, Jun. 2023, doi: 10.1007/s10489-022-04108-9.

SONG JUAN was born in Loudi, Hunan, China, in 1977. She received the master's degree from Hohai University. She is currently an Associate Professor with the School of Civil and Architectural Engineering, Shaoyang University. Her research interests include tunnel defect identification, building energy efficiency, and green construction.

HE LONG-XI was born in Zhuzhou, Hunan, China, in 1979. He is currently a Senior Laboratory Technician and a Faculty Member with the School of Civil and Architectural Engineering, Shaoyang University. His research interests include geotechnical engineering and tunnel lining defect identification.

LONG HUI-PING was born in Shaoyang, Hunan, China, in 1982. He is currently an Assistant Engineer and a Faculty Member with the School of Civil and Architectural Engineering, Shaoyang University. His research interests include municipal engineering and green construction technology.