

Article

MFF-YOLO: An Accurate Model for Detecting Tunnel Defects Based on Multi-Scale Feature Fusion

Anfu Zhu * , Bin Wang, Jiaxiao Xie and Congxiao Ma

School of Electronic Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450045, China; z20211181076@stu.ncwu.edu.cn (B.W.); z202210181146@stu.ncwu.edu.cn (J.X.); z20201181029@stu.ncwu.edu.cn (C.M.)

* Correspondence: zhuanfu@ncwu.edu.cn

Abstract: Tunnel linings require routine inspection as they have a big impact on a tunnel's safety and longevity. In this study, the convolutional neural network was utilized to develop the MFF-YOLO model. To improve feature learning efficiency, a multi-scale feature fusion network was constructed within the neck network. Additionally, a reweighted screening method was devised at the prediction stage to address the problem of duplicate detection frames. Moreover, the loss function was adjusted to maximize the effectiveness of model training and improve its overall performance. The results show that the model has a recall and accuracy that are 7.1% and 6.0% greater than those of the YOLOv5 model, reaching 89.5% and 89.4%, respectively, as well as the ability to reliably identify targets that the previous model error detection and miss detection. The MFF-YOLO model improves tunnel lining detection performance generally.

Keywords: deep learning; target detection; multiscale; feature fusion



Citation: Zhu, A.; Wang, B.; Xie, J.; Ma, C. MFF-YOLO: An Accurate Model for Detecting Tunnel Defects Based on Multi-Scale Feature Fusion. *Sensors* **2023**, *23*, 6490. <https://doi.org/10.3390/s23146490>

Academic Editor: Oscar Reinoso Garcia

Received: 9 June 2023

Revised: 12 July 2023

Accepted: 13 July 2023

Published: 18 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With China's recent economic growth, the tunnel sector has also entered a golden age of development, and tunnel building has emerged as one of the crucial elements of China's infrastructure development. However, due to geological factors, poor construction practices, and natural calamities, tunnels may have concealed flaws including uncompacted, hollow, and water filling that gravely jeopardize their service life [1].

To identify and address issues with the tunnels as soon as they arise, regular inspection and maintenance are required. However, conventional methods for finding tunnel defects, such as visual inspection and acoustic inspection, have low detection efficiencies and high result error. Given the quick advancement of deep learning technologies in the detection of targets, such as damage detection and localization of bridge deck pavement [2], crack detection in concrete bridges [3,4], steel surface flaw detection [5,6], and wheel defect detection [7] across a variety of disciplines in society, science, and engineering, deep learning-based tunnel flaw detection has recently gained the attention of both domestic and international academics. For example, Sjölander et al. [8] summarized the research on the application of optical detection technology and autonomous evaluation methods based on machine learning technology in tunnel lining inspection, as well as the research on digital cameras, laser scanning, fiber optic sensors, and other methods; they also proposed issues with traditional tunnel inspection methods, such as their low efficiency. Deep learning technology was used by Maeda et al. [9] to address the issue of data acquisition and traditional data enhancement techniques like rotation, translation, and flip that may alter the semantic information of the image and cause other drawbacks, using methods like selective image cropping and stitching to address the issue of the insufficient data set; however, the model detection effect did not improve. With the primary objective of resolving the data signal-to-noise ratio and multi-path interference problems to offer assurances for data feature extraction, Lei et al. [10] suggested an air-coupled geo-radar detection

technique provided by F-K filtering and BP migration. A strong technical guarantee for single defect detection was provided by the GPR forward simulation model developed by Wu et al. [11], but since tunnel defects are frequently made up of hollow, water-filling, and non-compactness, the model's generalization ability is not ideal. Ali et al. [12], using Faster-RCNN and YOLOv3 networks as well as conventional detection techniques, examined the performance of concrete structures; the findings revealed that the convolutional neural network-based detection strategy had greater detection accuracy and localization precision. Before training the model with an enhanced U-Net network, Wang et al. [13] first performed picture preprocessing. Smaller and more subtle faults might be detected by the model, but the model's efficiency was low, and its real-time performance could not keep up with demand. Li et al. [14] used U-Net and alternating update convolutional neural networks for automatic tunnel defect detection: first, the image is segmented and predicted to extract defect features; next, alternating update convolutional neural networks are used for classification and localization; however, the model is more complex and cannot satisfy the tunnel defect detection engineering requirements.

Although several researchers have developed numerous techniques for finding tunnel defects, these techniques still have flaws and disadvantages, such as low detection accuracy and slow detection speeds. Therefore, we are working on the relevant research and attempting to address the current shortcomings to further enhance the effectiveness and accuracy of tunnel defect detection. To increase the accuracy and usefulness of tunnel defect detection, as well as to better support and guarantee tunnel construction and maintenance operations, we will keep researching new methods and approaches. Previous studies by our group have included improvements to SGD networks and residual modules [15], the introduction of adversarial networks for data expansion problems [16], and the use of neural network fusion techniques to improve the generalization performance of the model.

This study addresses the current issues with tunnel defect detection in two ways: first, in terms of data acquisition, the most recent radar detection equipment's working process is shown in Figure 1, which has the qualities of high accuracy and high resolution and can precisely identify the defects within the depth of 10 m underground through the on-site survey; second, in terms of data processing, a detection model based on multi-scale feature fusion technology is proposed in this study, which uses the MFF-YOLO method to detect defects.



Figure 1. Ground-Penetrating Radar Vehicle TGRI-GPR.

The following are this paper's key contributions:

(1) Designing the Weighted Cross Connections Feature Pyramid Network (WCFPN) to address the issue of missing feature map information.

(2) Designing the Re-weighted Non-Maximum Suppression (RWNMS) to address the issue of redundancy and miss detection of detection frames.

(3) Improving the loss function by integrating the aspect ratio and Euclidean distance factors to improve the model training efficiency and detection accuracy.

The remainder of the text is organized as follows: Section 2 discusses the work on multiscale feature fusion and convolutional neural networks; Section 3 describes the technological strategies for improvement; the data set and assessment metrics are introduced in Section 4; tests are carried out in Section 5 to confirm the improved model performance; and the work is reviewed and expected in Section 6.

2. Related Work

2.1. Convolutional Neural Network

CNN [17] is a deep learning model consisting of convolutional, pooling, and fully connected layers, etc., for implementing tasks such as image classification, target detection, and image segmentation. One of the target detection methods can be divided into two-stage and single-stage, and the difference between the two algorithms lies in the way the generation frame is combined with the candidate frame. In a two-stage algorithm, candidate regions are first generated and then the CNN applies to these candidate regions for classification. For example, the R-CNN [18–22] family of algorithms has high detection accuracy but is computationally intensive and inefficient. In contrast, single-stage algorithms such as the SSD [23–27] and YOLO [28–35] series can achieve significant detection speedups. Single-stage target detection algorithms have become preferred in industrial applications due to their ability to directly output information about the position and detection frame of the target to be detected. In conclusion, with the continuous development of deep learning techniques, target detection algorithms are also constantly being advanced and optimized, with different algorithms being suitable for different application scenarios and needs.

2.2. Multi-Scale Feature Fusion

Multi-scale feature fusion is a technique for combining feature maps at different scales to improve the performance of computer vision tasks. Common approaches include cascade structures [36–38], pyramid networks [39–42] and attention mechanisms [43–45]. Cascade structures link feature maps at different scales together to form cascade networks, which can be effective in improving performance; pyramid networks are a hierarchical approach to image processing that extracts features at different scales and combines them; and attention mechanisms can make the network focus more on important features by weighting feature maps at different scales. All these methods aim to improve the performance of computer vision tasks by using feature maps at different scales, thus improving the accuracy and robustness of the task.

3. Methods

The MFF-YOLO model improvement approach is described in this section, and Figure 2 illustrates the structural characteristics of the MFF-YOLO model. The model improvement consists of two main components: the WCFPN structure of the neck network, where the blue module represents the Weighted Cross Connections built by weighting and the red line portion reflects the network across connection idea; and the RWNMS mechanism at the predicted end, where the EIOU loss function is used in the screening process of the prediction frame and the weights are further provided by normalization.

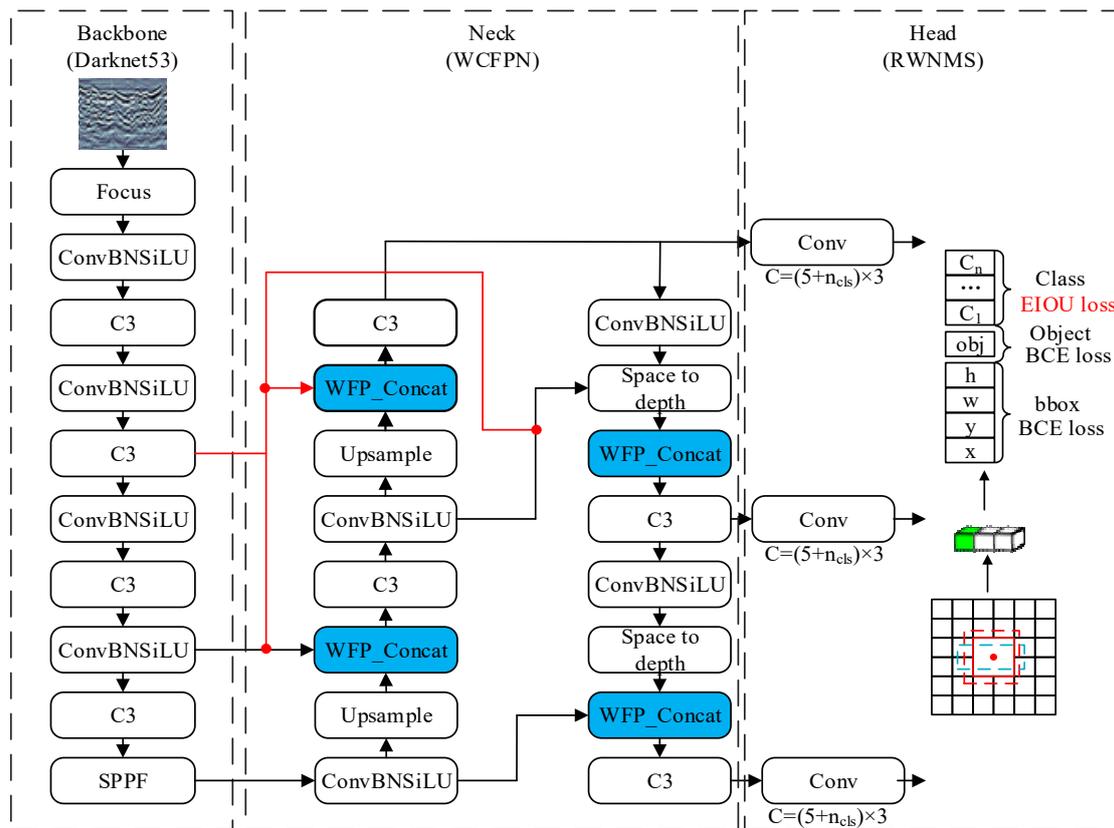


Figure 2. Structure of MFF-YOLO. Where the black solid lines indicate the input-output paths and the red solid lines are the additional input-output paths of the improved model, and the black solid lines are used to form a grid map in the detection head part, and the red and blue dashed boxes indicate the location of the prediction box.

3.1. WCFPN

The target detection model is prone to the problem that as the number of layers in the feature map increases, the resolution of the feature map decreases, resulting in poor information transfer in the feature map and hence the lack of detection capability of the model. The model may zoom in and out of the feature maps by adding operations like convolutional layers and pooling layers; however, this reduces the performance of the model since they commonly cause information loss or information redundancy. The study uses cross-scale linking to connect feature maps of multiple sizes, better using the data in those feature maps, and then suggests the concept of weighted fusion when zooming in and out on the feature maps to address this issue.

Using the effective feature fusion technique known as weighted fusion, data are combined in feature maps of various scales according to predetermined weights to create multi-resolution feature maps. Cross-scale linking, unlike weighted fusion, is a feature alignment technique that aligns feature maps at several sizes to create a multi-resolution feature map for enhanced feature fusion and model performance at many scales.

The fusion process is shown in Figure 3. Multiple convolution and pooling operations are performed on the input image to obtain feature maps of different sizes, and the last three layers are selected to construct the WCFPN, as shown in Equations (1) and (2).

$$\vec{P}^{\text{in}} = (p_{11}^{\text{in}}, p_{12}^{\text{in}}, \dots) \tag{1}$$

$$\vec{P}^{out} = f\left(\vec{P}^{in}\right) \tag{2}$$

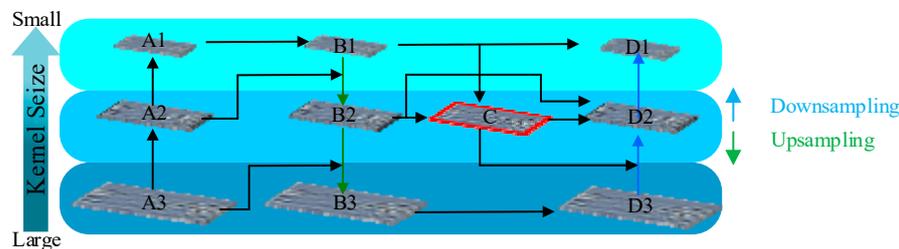


Figure 3. Structure of WCFPN. Where $A_1, A_2, \dots, D_1, D_2, D_3$ represent different upper and lower sampling layers respectively.

The multi-scale feature fusion in Equation (1) takes the feature maps from different resolutions and weights them, resulting in a richer and more comprehensive feature representation. Equation (2), on the other hand, cascades feature maps from different resolutions, thus combining feature information from different resolutions in the spatial dimension.

The problem of incomplete and inconsistent feature information is solved by the multi-scale feature fusion method, which can identify and locate targets more accurately. The research feature fusion method uses the fast normalization method, which can adjust the weights of different features by applying a normalization operation on each input feature to fuse feature maps at different scales more effectively, with the formula shown in Equation (3).

$$O = \sum_i (\omega_i \cdot X_i) / \left(\epsilon + \sum_j \omega_j \right) \tag{3}$$

where O denotes the output feature mapping; ω_i is the weighting factor; X_i denotes the input feature mapping, taken as 0.001.

Take D_2 layer output as an example. First, the A_1 layer is convolved 1×1 to obtain the feature mapping B_1 . Second, the B_1 layer is fused with the A_2 layer to obtain the B_2 layer, and then the B_1 layer and B_2 layer are fused. Then, fuse both layers to obtain the intermediate layer mapping C . Finally, this layer is fed to D_2 for cross-scale connection and weighting to obtain the D_2 layer. The layer output can be expressed as Equations (4) and (5).

$$D_2^{in} = \text{Conv} \left(\frac{\omega_1 \cdot B_2^{in} + \omega_2 \cdot \text{Res}(B_1^{in})}{W_1 + W_2 + \epsilon} \right) \tag{4}$$

$$D_2^{out} = \text{Conv} \left[\left(\omega'_1 \cdot B_2^{in} + \omega'_2 \cdot B_2^{td} + \omega'_3 \cdot \text{Res}(B_3^{out}) \right) / \omega'_1 + \omega'_2 + \omega'_3 + \epsilon \right] \tag{5}$$

where D_2^{in} is the middle feature of the top-down path layer 2, and D_2^{out} is the output feature of the bottom-up path layer 2. Res denotes the up-sampling or down-sampling adjustment size, which is used to resize the feature map for feature fusion. All other features are constructed in the same way.

3.2. RWNMS

In target detection, sliding window or region extraction methods are usually used to generate candidate predictor frames, but the same target may be covered by multiple predictor frames, resulting in duplicate and inaccurate target detection results. The role of NMS operation is to select the predictor frame with the highest confidence among these overlapping predictor frames and remove other low-confidence results, and the calculation process is shown in Equation (6).

$$D_2^{out} = \text{Conv} \left[\left(\omega'_1 \cdot B_2^{in} + \omega'_2 \cdot B_2^{td} + \omega'_3 \cdot \text{Res}(B_3^{out}) \right) / \omega'_1 + \omega'_2 + \omega'_3 + \epsilon \right] \tag{6}$$

Although NMS can remove overlapping prediction frames, there may be a large amount of overlap in the prediction frames generated on multiple grids or multiple scales of feature maps, which will directly lead to the inability of the NMS algorithm to remove all the redundant prediction frames. To avoid the problem of redundant prediction frames, the study introduces the EIOU loss function, which considers not only the overlap between detection frames but also the similarity and confidence between them, etc.; by giving various detection findings distinct weights, the weighting approach may be utilized to pick and optimize outcomes or instance; while utilizing NMS, some significant targets or detection findings with higher confidence might be assigned larger weights and are hence more likely to be chosen. Additionally, a smoothing strategy is used to further increase the accuracy and robustness of detection by reducing issues like noise and oscillation by weighting the detection results of adjacent frames and averaging them when processing adjacent prediction frames, leading to more accurate and stable detection results; this can successfully boost detection’s resilience and generalization capabilities, better enabling it to handle detection jobs in complicated scenarios. The process of the improved RWI mechanism is shown in Figure 4, and its confidence score is calculated as Equation (7).

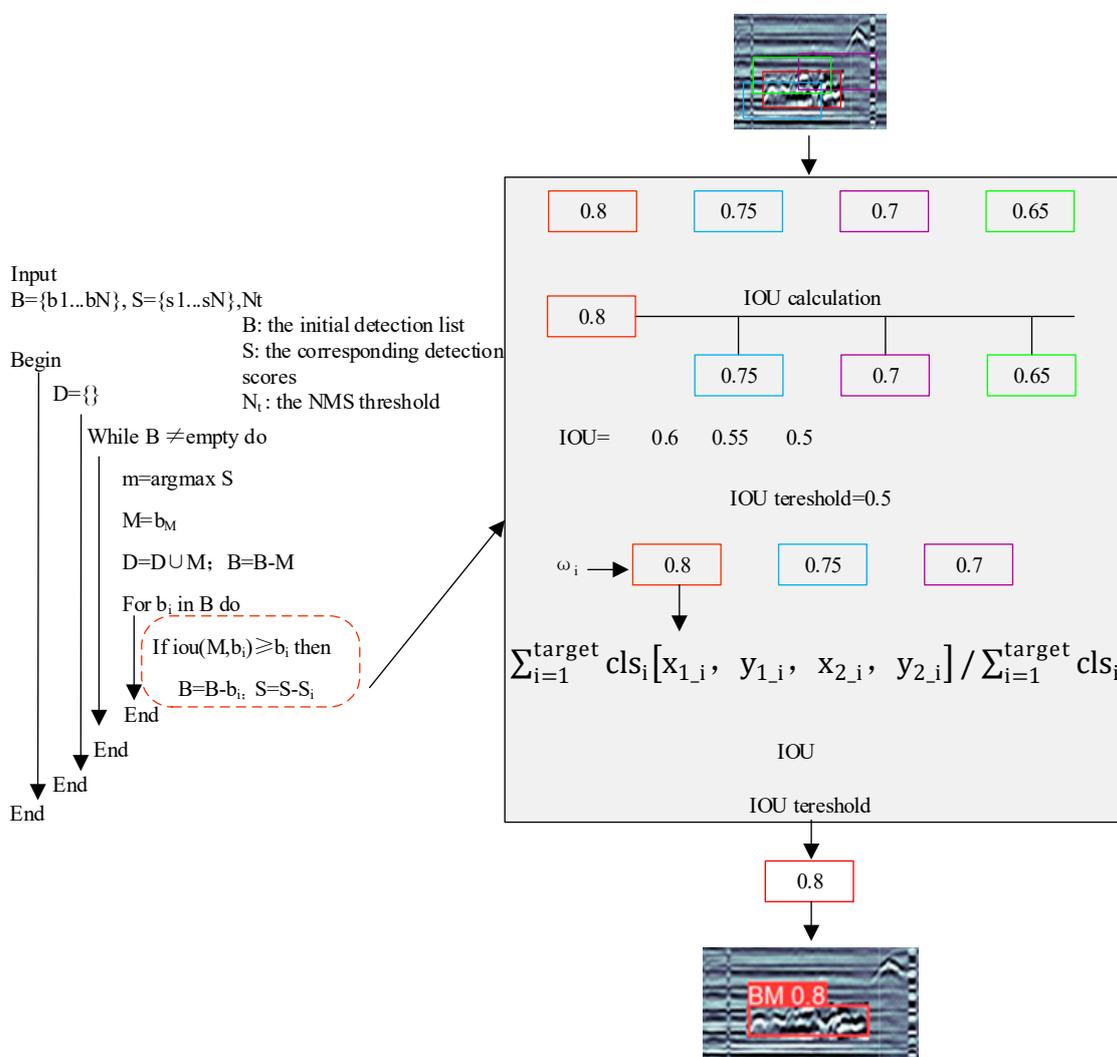


Figure 4. Improved RWI module. Where different colors represent different prediction frames.

$$c = p_0 \times L_{EIOU} \tag{7}$$

where L_{EIOU} denotes the intersection ratio of the prediction frame and the true frame obtained based on the EIOU loss function, which can be expressed by Equation (8); p_0 denotes the probability of the presence of the target in the prediction frame, and if it exists then $p_0 = 1$, and vice versa $p_0 = 0$.

$$L_{EIOU} = \sum_{i=0}^{s^2} \sum_{j=0}^B (L_{IOU} + L_{dis} + L_{asp}) \quad (8)$$

$$L_{dis} = \frac{\rho^2(b, b^{gt})}{(w^c)^2 + (h^c)^2}$$

$$L_{asp} = \frac{\rho^2(w, w^{gt})}{(w^c)^2} + \frac{\rho^2(h, h^{gt})}{(h^c)^2}$$

where L_{dis} denotes distance loss, L_{asp} denotes the phase loss, and $\rho^2(b, b^{gt})$ denotes the center point of the two frames b and b^{gt} , w^c and h^c that denote the length and width of the smallest frame containing A and B, respectively, and the specific parameters are shown in Figure 5.

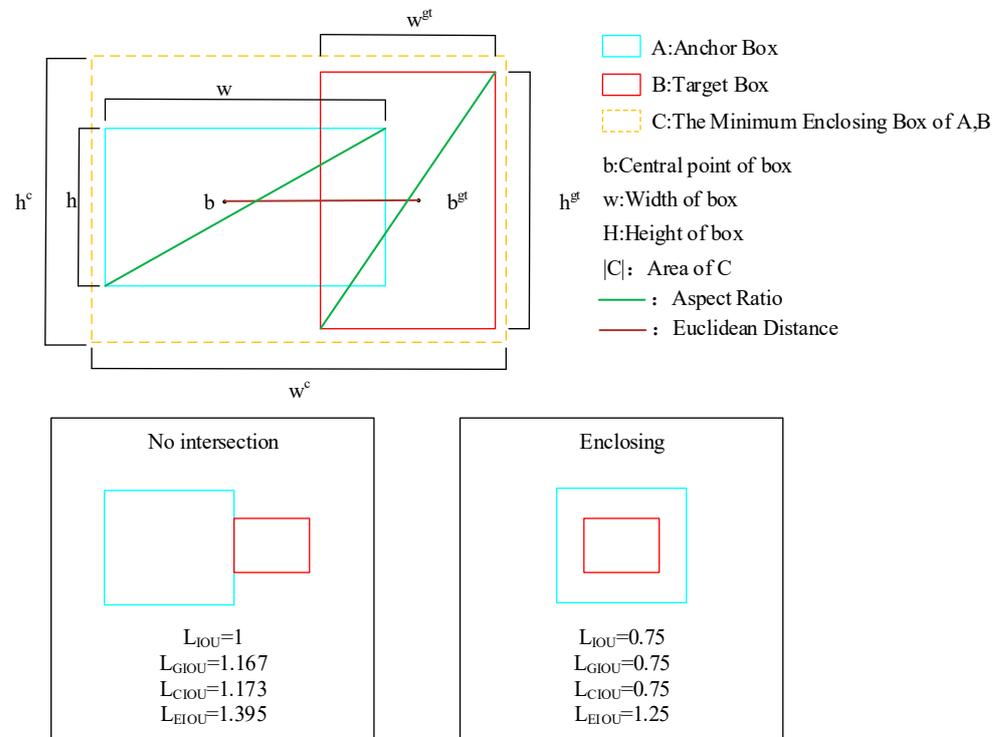


Figure 5. Some parameters of EIOU loss function.

When there are targets of different scales or different shapes in the target detection, the traditional IOU calculation method is somewhat inaccurate because it only takes into account the ratio of the intersection and the concatenation of the detection frame and true labeled frame, ignoring the shape or size of the frame. To address this issue, EIOU introduces the aspect ratio and the Euclidean distance factor, increasing the variability between IOU values so that larger IOU values are closer to 1 and smaller IOU values are closer to 0, improving the accuracy and robustness of target detection.

In the calculation process, all prediction frames are first arranged in descending order according to the confidence score. Then, starting from the frame with the highest confidence score, all prediction frames are given weights and traversed again, the width and height of

the current best frames are updated according to Equation (9), and the current best frame is finally obtained as the result after RWNMS.

$$\text{Box}_{i=\max}[x_1, y_1, x_2, y_2] = \frac{\sum_{i=1}^{\text{target}} \text{cls}_i [x_{1_i}, y_{1_i}, x_{2_i}, y_{2_i}]}{\sum_{i=1}^{\text{target}} \text{cls}_i} \quad (9)$$

where $\text{box}_{i=\max}[\]$ denotes the box with the highest confidence level, and $\sum_{i=1}^{\text{target}}$ denotes the prediction box from $i = 1$ to the target obtained by filtering. After that, the prediction box position is updated by Equation (9) to improve the stability of the prediction results.

4. Experimental Studies

4.1. Data Processing

To construct the data set required for this study, multiple segments of tunnel defective radar data were collected and post-processing operations such as image enhancement and image annotation were performed on these data. The final nearly 5700 images were obtained by Labelling software labeling to form the data set of this study, and the defect images were classified into five categories such as BM, TK, KD, CS, and YBM, which denote five types of defects, namely, uncompacted, emptying, hollow, water-filled and severely uncompacted, respectively. Table 1 shows the distribution of the five types of defective images in the data set and Figure 6 shows some typical examples of defects.

Table 1. Distribution of tunnel defect data sets.

Defect Type	Training Set	Validation Set	Test Set	Label
BM	1301	114	76	0
TK	1815	127	139	1
KD	904	78	254	2
CS	1136	80	197	3
YBM	1175	111	107	4

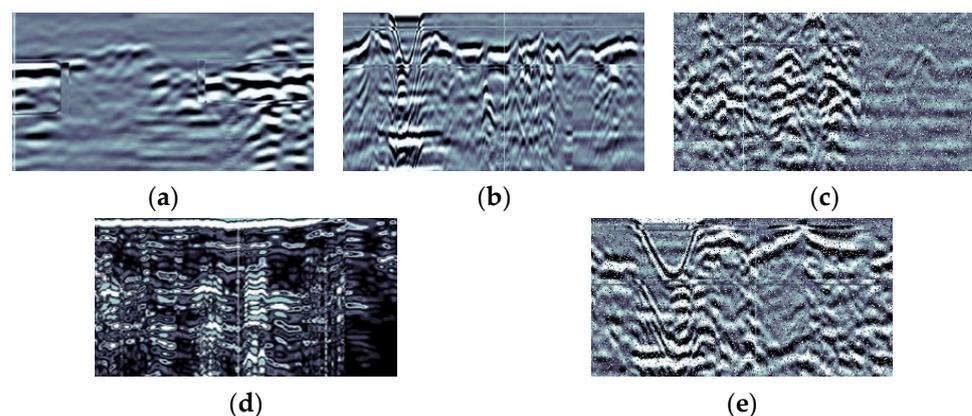


Figure 6. Defect sample example. (a) uncompacted, (b) emptying, (c) hollow, (d) water-filled, (e) severely uncompacted.

4.2. Experimental Procedure

4.2.1. Experimental Configuration

Radar detection equipment that emits radar waves and then receives the signals bounced back to obtain information about the parameters shown in Table 2.

Table 2. Radar rover-related parameters.

Name	Configure
Equipment Model	TGRI-GPR
Center Frequency	200 MHz
Operating Bandwidth	100–500 MHz
Depth of detection	3 m
Dynamic Range	40 dB

The deep learning simulation experiments were built on a Linux system, using Python and PyTorch to build the deep learning framework. The hardware setup shown in Table 3 includes components such as CPU, GPU, memory, and storage.

Table 3. Experimental hardware configuration.

Name	Configure
Operating System	Linux
Video Card	NVIDIA RTX3090
Video Memory	24 G
Processor	Intel® Core i3-8100
Programming Language	Python
Deep training framework	PyTorch
Programming Platforms	Pycharm

4.2.2. Evaluation Indicators

The experiments in this study use recall, mAP to evaluate the effectiveness of the model in detecting tunnel defects, which can be defined as Equation (10):

$$\begin{aligned}
 \text{Precision} &= \frac{TP}{TP+FP} \\
 \text{Recall} &= \frac{TP}{TP+FN} \\
 \text{AP} &= \int_0^1 P \epsilon dR \\
 \text{mAP} &= \frac{\sum_{i=1}^N AP_i}{N} \\
 \text{mAP}_{[@0.5,@0.95]} &= (\text{mAP}_{@0.5} + \dots + \text{mAP}_{@0.95})/10
 \end{aligned} \tag{10}$$

where TP represents the number of positive examples correctly classified, TN represents the number of negative examples correctly classified, FP represents the number of positive examples incorrectly classified, and FN represents the number of negative examples incorrectly classified.

5. Results

The study used the MFF-YOLO model for tunnel lining defect detection and successfully detected a wide range of defect types and sizes, with some of the results shown in Figure 7.

The loss function curve in Figure 8 was generated using a scatter plot to represent the training data. It is evident from observation that the MFF-YOLO model converges significantly more quickly than other models do, and in the 20th round, all loss functions begin to converge and subsequently begin to stabilize. While the convergence and stabilization of the loss function value suggest that the model has achieved the ideal state, the reduction in the loss function value shows that the model is gradually being optimized during the training phase.

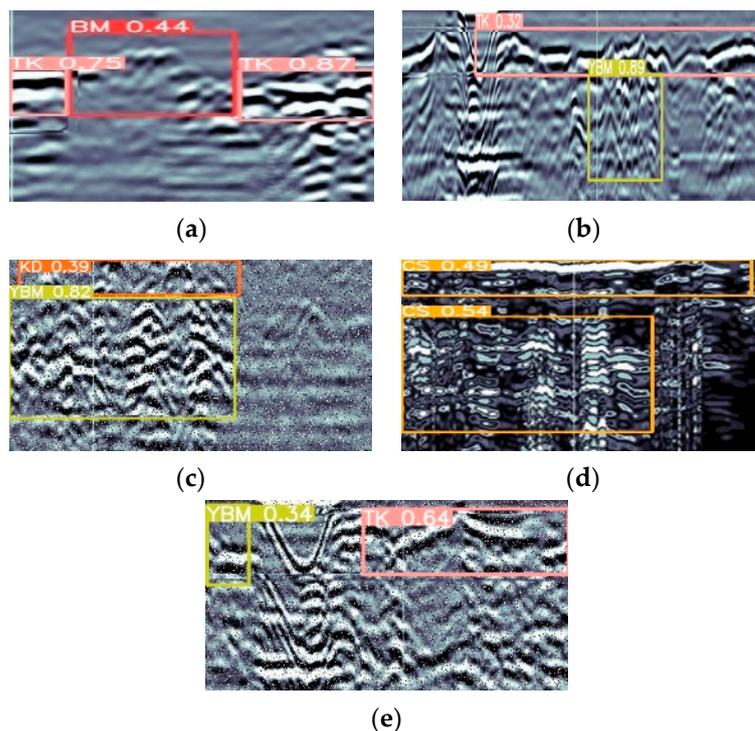


Figure 7. Detection of different categories of defects. (a) Effect of BM detection, (b) Effect of TK detection, (c) Effect of KD detection, (d) Effect of CS detection, (e) Effect of YBM detection.

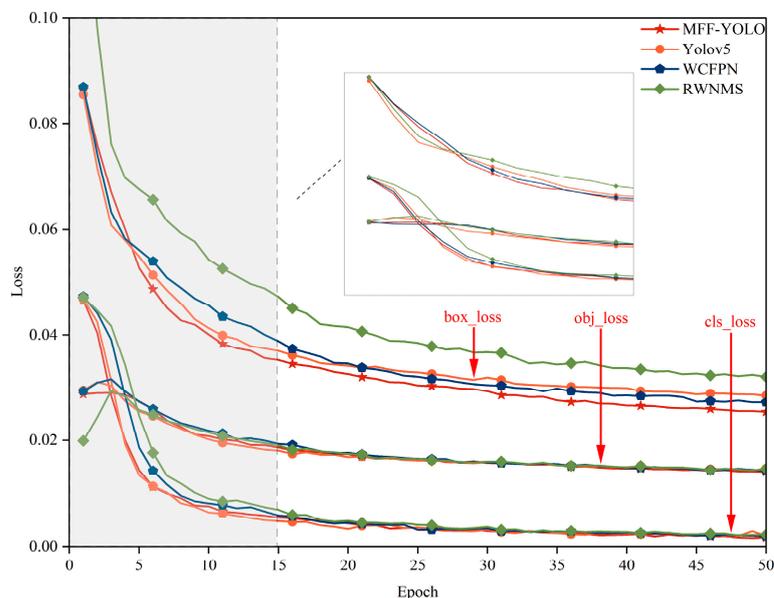


Figure 8. MFF-YOLO loss curve.

In order to test the effectiveness of each model for each type of defect detection to highlight the importance of each module, the study tallied the accuracy results for different types as shown in Figure 9 and Table 4. 5 sections in Figure 9 represent each of the five types of defects, with different modules added to each section from left to right, where the leftmost is the original model and the rightmost red is the improved MFF-YOLO model. The results show that the improved model proposed in this paper outperforms the original model for all tunnel defect types, which further validates the effectiveness of the model in tunnel defect detection.

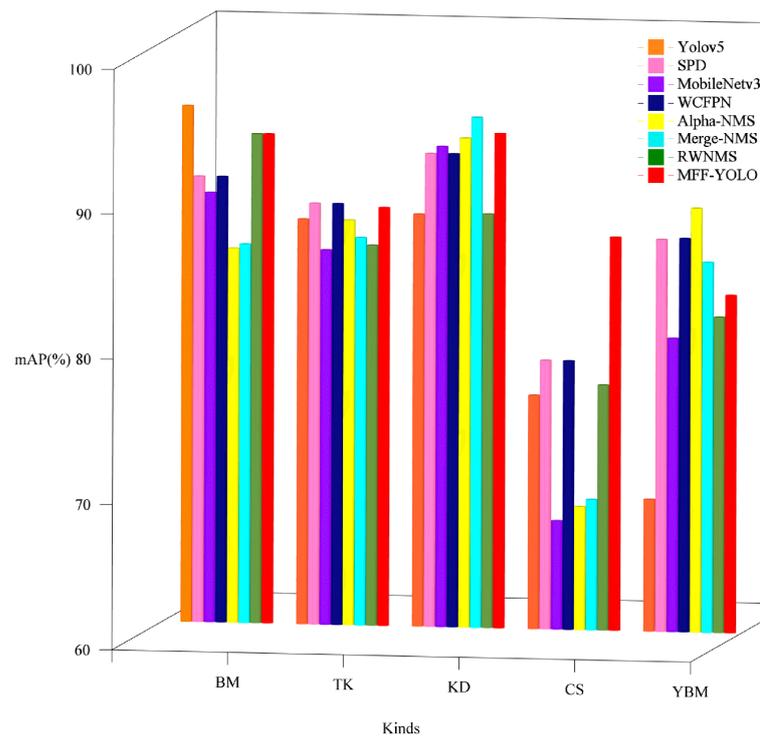


Figure 9. Comparison of detection effects of different models for different categories.

Table 4. Comparison of experimental results of each model in different defects.

Model	Type	Average Accuracy Rate (%)				
		BM	TK	KD	CS	YBM
Yolov5		95.6	87.9	88.4	76.1	69.1
SPD		90.7	89.0	92.6	78.5	87.0
MobileNetv3		89.6	85.8	93.1	67.5	80.2
WCFPN		90.7	89.0	92.6	78.5	87.1
Alpha-NMS		85.8	87.9	93.7	68.5	89.2
Merge-NMS		86.1	86.7	95.2	69.0	85.5
RWNMS		93.7	86.2	88.5	76.9	81.7
MFF-YOLO		93.7	88.8	94.1	87.1	83.2

To more accurately assess the performance of the MFF-YOLO model proposed in this paper, the study conducted ablation tests and obtained the experimental results shown in Figure 10 and Table 5. The results show that the model MFF-YOLO achieves an accuracy of 89.4%. In addition, through the ablation tests, we also verified that the model has a strong robustness and generalization capability for effective tunnel defect detection under different environments and conditions.

The research in Table 5 shows that after adding WCFPN, the mAP, Precision, Recall and Gflops of the model are significantly improved through deep fusion extraction of feature information. After adding RWNMS to reprocess the prediction box, although the accuracy improvement is not as obvious as WCFPN, it keeps the FPS stable. When the two modules are combined to construct the MFF-YOLO model in this study, the result is a synergistic impact that outperforms the effect of each module acting alone and raises the model's total performance ceiling.

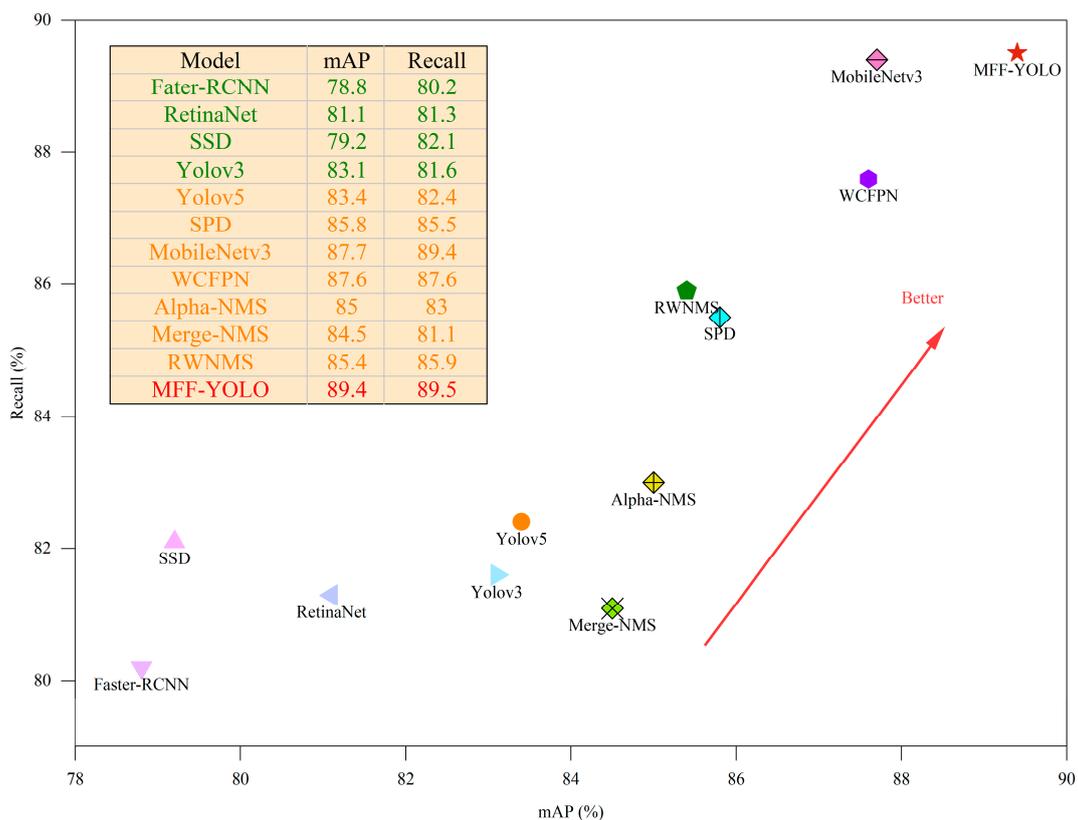


Figure 10. Compare of mAP and Recall.

Table 5. Results of ablation experiments.

Name	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Precision (%)	Recall (%)	Gflops	FPS
Fater-RCNN	78.8	48.8	78.6	80.2	88.2	18.2
RetinaNet	81.1	49.8	79.7	81.3	70.3	18.3
SSD	79.2	47.0	75.2	82.1	15.2	65.8
Yolov3	83.1	49.1	80.3	81.6	154.6	16.4
Yolov5	83.4	47.9	75.4	82.4	16.8	66.6
SPD	85.8	49.1	80.3	85.5	16.0	71.4
MobileNetv3	87.7	50.4	81.0	89.4	15.3	50.0
WCFPN	87.6	51.6	82.2	87.6	33.1	52.6
Alpha-NMS	85.0	51.3	78.6	83.0	15.8	71.4
Merge-NMS	84.5	50.5	78.3	81.1	15.8	71.4
RWNMS	85.4	49.7	80.8	85.9	16.8	66.6
MFF-YOLO	89.4	50.8	82.2	89.5	33.3	50.0

Figure 11 presents a visualization of the test results of this study on the data set, allowing for a more intuitive understanding of the performance and performance of the model, and thus better model improvement and optimization.

These results show that the improved algorithm proposed in this study has high performance and practicality in tunnel defect detection and can be better adapted to a variety of different types and sizes of defect detection tasks.

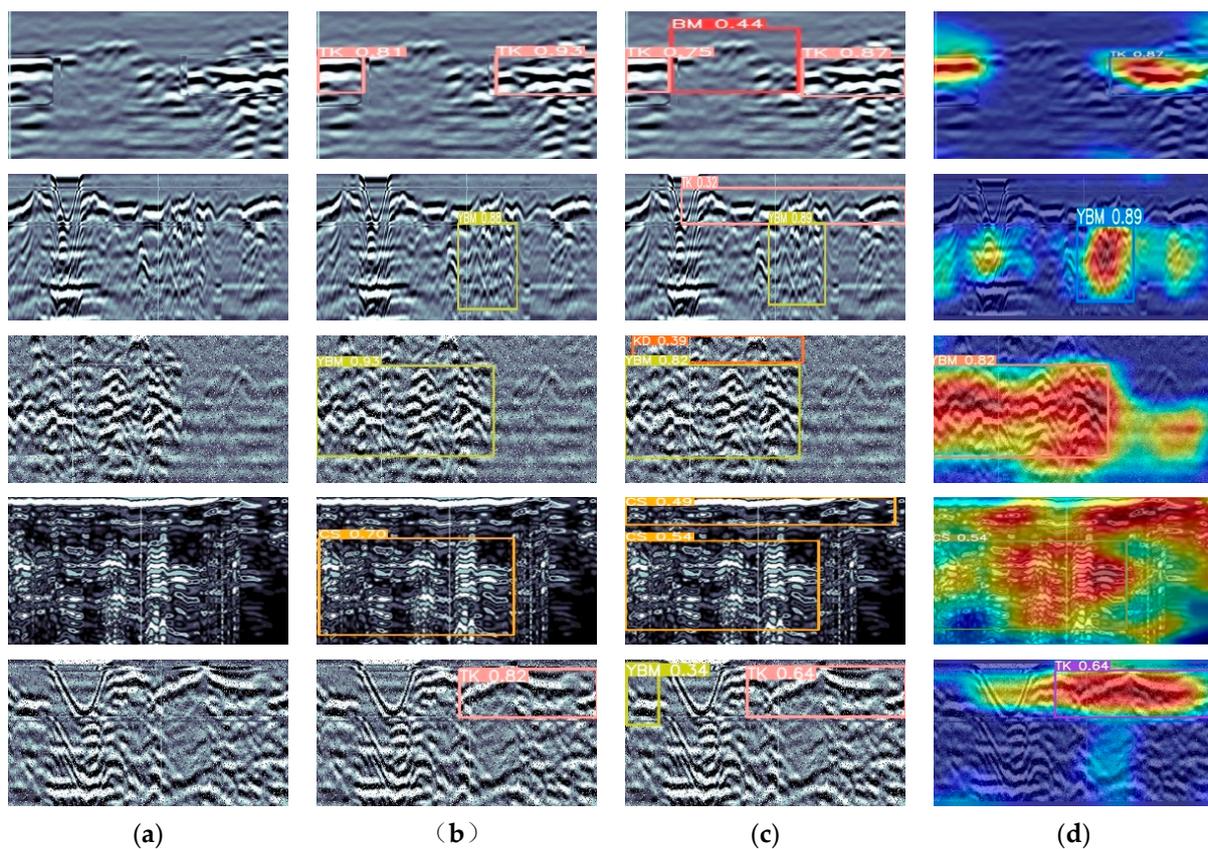


Figure 11. Detection result plots. Where (a) is the original defect maps, (b) is the original model detection result maps, (c) is the improved MFF-YOLO model detection result maps, and (d) is the result visualization maps.

6. Conclusions

In this study, we developed an MFF-YOLO model based on multi-scale feature fusion technology. In the neck network, WCFPN is designed based on a weighted fusion strategy to improve the ability to obtain feature information from multiple dimensions. At the predicted end, the prediction mechanism of RWNMS is designed in combination with a weighted smoothing strategy. Finally, the EIOU loss function is improved by considering the aspect ratio and Euclidean distance.

The feasibility of the strategy was evaluated by testing the tunnel defect data set. The result shows that the MFF-YOLO model's recall rate and accuracy rates were 7.1% and 6.0% higher than those of the YOLOv5 model, reaching 89.5% and 89.4%, respectively.

Compared to conventional image processing approaches, deep learning-based tunnel defect detection algorithms may learn higher-level feature representations from a huge quantity of data, improving the identification and classification of flaws. Additionally, automated and intelligent deep learning-based detection strategies can increase the effectiveness and dependability of detection. In the future, we will continue to explore ways to improve model defect detection performance, including reducing model complexity, improving detection accuracy, and optimizing data set quality.

Author Contributions: Conceptualization, A.Z.; Methodology, A.Z.; Software, B.W. and J.X.; Validation, B.W. and J.X.; Formal analysis, J.X.; Investigation, B.W. and C.M.; Resources, A.Z.; Data curation, C.M.; Writing—original draft, B.W.; Writing—review & editing, A.Z.; Supervision, A.Z.; Project administration, A.Z.; Funding acquisition, A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Science and Technology Project of Henan Province (222102210135).

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare that they have no competing interest.

References

1. Wang, J.; Xie, X.; Huang, H. A fuzzy comprehensive evaluation system of mountain tunnel lining based on the fast nondestructive inspection. In Proceedings of the 2011 International Conference on Remote Sensing, Environment and Transportation Engineering, Nanjing, China, 24–26 June 2011; pp. 2832–2834.
2. Ni, Y.; Mao, J.; Fu, Y.; Wang, H.; Zong, H.; Luo, K. Damage Detection and Localization of Bridge Deck Pavement Based on Deep Learning. *Sensors* **2023**, *23*, 5138. [[CrossRef](#)] [[PubMed](#)]
3. Kao, S.-P.; Chang, Y.-C.; Wang, F.-L. Combining the YOLOv4 Deep Learning Model with UAV Imagery Processing Technology in the Extraction and Quantization of Cracks in Bridges. *Sensors* **2023**, *23*, 2572. [[CrossRef](#)]
4. Santaniello, P.; Russo, P. Bridge Damage Identification Using Deep Neural Networks on Time–Frequency Signals Representation. *Sensors* **2023**, *23*, 6152. [[CrossRef](#)] [[PubMed](#)]
5. Yan, R.; Zhang, R.; Bai, J.; Hao, H.; Guo, W.; Gu, X.; Liu, Q. STMS-YOLOv5: A Lightweight Algorithm for Gear Surface Defect Detection. *Sensors* **2023**, *23*, 5992. [[CrossRef](#)] [[PubMed](#)]
6. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. [[CrossRef](#)] [[PubMed](#)]
7. Shaikh, K.; Hussain, I.; Chowdhry, B.S. Wheel Defect Detection Using a Hybrid Deep Learning Approach. *Sensors* **2023**, *23*, 6248. [[CrossRef](#)]
8. Sjölander, A.; Belloni, V.; Ansell, A.; Nordström, E. Towards automated inspections of tunnels: A review of optical inspections and autonomous assessment of concrete tunnel linings. *Sensors* **2023**, *23*, 3189. [[CrossRef](#)]
9. Maeda, K.; Takada, S.; Haruyama, T.; Togo, R.; Ogawa, T.; Haseyama, M. Distress Detection in Subway Tunnel Images via Data Augmentation Based on Selective Image Cropping and Patching. *Sensors* **2022**, *22*, 8932. [[CrossRef](#)]
10. Lei, Y.; Jiang, B.; Su, G.; Zou, Y.; Qi, F.; Li, B.; Jia, F.; Tian, T.; Qu, Q. Application of Air-Coupled Ground Penetrating Radar Based on FK Filtering and BP Migration in High-Speed Railway Tunnel Detection. *Sensors* **2023**, *23*, 4343. [[CrossRef](#)]
11. Wu, X.; Bao, X.; Shen, J.; Chen, X.; Cui, H. Evaluation of Void Defects behind Tunnel Lining through GPR forward Simulation. *Sensors* **2022**, *22*, 9702. [[CrossRef](#)]
12. Ali, L.; Alnajjar, F.; Jassmi, H.A.; Gocho, M.; Khan, W.; Serhani, M.A. Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors* **2021**, *21*, 1688. [[CrossRef](#)]
13. Wang, A.; Togo, R.; Ogawa, T.; Haseyama, M. Defect detection of subway tunnels using advanced U-Net network. *Sensors* **2022**, *22*, 2330. [[CrossRef](#)]
14. Li, G.; Ma, B.; He, S.; Ren, X.; Liu, Q. Automatic tunnel crack detection based on u-net and a convolutional neural network with alternately updated clique. *Sensors* **2020**, *20*, 717. [[CrossRef](#)]
15. Zhu, A.; Chen, S.; Lu, F.; Ma, C.; Zhang, F. Recognition Method of Tunnel Lining Defects Based on Deep Learning. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9070182. [[CrossRef](#)]
16. Zhu, A.; Ma, C.; Chen, S.; Wang, B.; Guo, H. Tunnel Lining Defect Identification Method Based on Small Sample Learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1096467. [[CrossRef](#)]
17. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
19. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, 1137–1149. [[CrossRef](#)]
21. Srinivasu, P.N.; SivaSai, J.G.; Ijaz, M.F.; Bhoi, A.K.; Kim, W.; Kang, J.J. Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors* **2021**, *21*, 2852. [[CrossRef](#)]
22. Xu, X.; Zhao, M.; Shi, P.; Ren, R.; He, X.; Wei, X.; Yang, H. Crack detection and comparison study based on faster R-CNN and mask R-CNN. *Sensors* **2022**, *22*, 1215. [[CrossRef](#)] [[PubMed](#)]
23. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
24. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
25. Li, Z.; Zhou, F. FSSD: Feature fusion single shot multibox detector. *arXiv* **2017**, arXiv:1712.00960.

26. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European conference on computer vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 385–400.
27. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
28. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
29. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
30. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
31. Bochkovskiy, A.; Wang, C.; Liao, H. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
32. Mathias, A.; Dhanalakshmi, S.; Kumar, R. Occlusion aware underwater object tracking using hybrid adaptive deep SORT-YOLOv3 approach. *Multimed. Tools Appl.* **2022**, *81*, 44109–44121. [[CrossRef](#)]
33. Lai, H.; Chen, L.; Liu, W.; Yan, Z.; Ye, S. STC-YOLO: Small Object Detection Network for Traffic Signs in Complex Environments. *Sensors* **2023**, *23*, 5307. [[CrossRef](#)]
34. Bao, C.; Cao, J.; Hao, Q.; Cheng, Y.; Ning, Y.; Zhao, T. Dual-YOLO Architecture from Infrared and Visible Images for Object Detection. *Sensors* **2023**, *23*, 2934. [[CrossRef](#)]
35. Xia, K.; Lv, Z.; Zhou, C.; Gu, G.; Zhao, Z.; Liu, K.; Li, Z. Mixed Receptive Fields Augmented YOLO with Multi-Path Spatial Pyramid Pooling for Steel Surface Defect Detection. *Sensors* **2023**, *23*, 5114. [[CrossRef](#)]
36. Ruan, Z.; Wang, H.; Cao, J.; Zhang, H. Cross-scale feature fusion connection for a YOLO detector. *IET Comput. Vis.* **2022**, *16*, 99–110. [[CrossRef](#)]
37. Huang, K.; Li, C.; Zhang, J.; Wang, B. Cascade and fusion: A deep learning approach for camouflaged object sensing. *Sensors* **2021**, *21*, 5455. [[CrossRef](#)]
38. Mo, L.; Zhu, Y.; Zeng, L. A Multi-Label Based Physical Activity Recognition via Cascade Classifier. *Sensors* **2023**, *23*, 2593. [[CrossRef](#)]
39. Huang, H.; Tang, X.; Wen, F.; Jin, X. Small object detection method with shallow feature fusion network for chip surface defect detection. *Sci. Rep.* **2022**, *12*, 3914. [[CrossRef](#)] [[PubMed](#)]
40. Xu, Z.; Yang, Y.; Gao, X.; Hu, M. DCFF-MTAD: A Multivariate Time-Series Anomaly Detection Model Based on Dual-Channel Feature Fusion. *Sensors* **2023**, *23*, 3910. [[CrossRef](#)] [[PubMed](#)]
41. Qian, X.; Wang, X.; Yang, S.; Lei, J. LFF-YOLO: A YOLO Algorithm with Lightweight Feature Fusion Network for Multi-Scale Defect Detection. *IEEE Access* **2022**, *10*, 130339–130349. [[CrossRef](#)]
42. Mao, K.; Jin, R.; Chen, K.; Mao, J.; Dai, G. Trinity-Yolo: High-precision logo detection in the real world. *IET Image Process.* **2023**, *17*, 2272–2283. [[CrossRef](#)]
43. Wang, J.; Dong, Y.; Zhao, S.; Zhang, Z. A High-Precision Vehicle Detection and Tracking Method Based on the Attention Mechanism. *Sensors* **2023**, *23*, 724. [[CrossRef](#)] [[PubMed](#)]
44. Hu, W.; Cao, L.; Ruan, Q.; Wu, Q. Research on Anomaly Network Detection Based on Self-Attention Mechanism. *Sensors* **2023**, *23*, 5059. [[CrossRef](#)]
45. Wang, D.; Xiang, S.; Zhou, Y.; Mu, J.; Zhou, H.; Irampaye, R. Multiple-Attention Mechanism Network for Semantic Segmentation. *Sensors* **2022**, *22*, 4477. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.