World Electric Vehicle Journal

MDPI

Article

BI-TST_YOLOv5: Ground Defect Recognition Algorithm Based on Improved YOLOv5 Model

Jiahao Qin, Xiaofeng Yang *, Tianyi Zhang and Shuilan Bi

School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China; 2222104064@stmail.ujs.edu.cn (J.Q.); 2112304072@stmail.ujs.edu.cn (T.Z.); 2212104034@stmall.ujs.edu.cn (S.B.)
* Correspondence: yangxf18@ujs.edu.cn

Abstract: Pavement defect detection technology stands as a pivotal component within intelligent driving systems, demanding heightened precision and rapid detection rates. Addressing the complexities arising from diverse defect types and intricate backgrounds in visual sensing, this study introduces an enhanced approach to augment the network structure and activation function within the foundational YOLOv5 algorithm. Initially, modifications to the YOLOv5′s architecture incorporate an adjustment to the Leaky ReLU activation function, thereby enhancing regression stability and accuracy. Subsequently, the integration of bi-level routing attention into the network's head layer optimizes the attention mechanism, notably improving overall efficiency. Additionally, the replacement of the YOLOv5 backbone layer's C3 module with the C3-TST module enhances initial convergence efficiency in target detection. Comparative analysis against the original YOLOv5s network reveals a 2% enhancement in map50 and a 1.8% improvement in F1, signifying an overall advancement in network performance. The initial convergence rate of the algorithm has been improved, and the accuracy and operational efficiency have also been greatly improved, especially on models with small-scale training sets.

Keywords: YOLOv5; attention mechanism; BiFormer; C3-TST; activation function

Citation: Qin, J.; Yang, X.; Zhang, T.; Bi, S. BI-TST_YOLOv5: Ground Defect Recognition Algorithm Based on Improved YOLOv5 Model. *World Electr. Veh. J.* **2024**, *15*, 102. https://doi.org/10.3390/wevj15030102

Academic Editor: Joeri Van Mierlo

Received: 14 December 2023
Revised: 14 February 2024
Accepted: 4 March 2024
Published: 7 March 2024

1. Introduction

Automatic driving technology is also following the rapid development of information technology, computer technology, automation technology, and artificial intelligence technology. In the development process of automatic driving technology, to meet the requirements for vehicle suspension adjustment—aiming to provide a better ride experience and extend the mechanical life—the demand for accurate and fast intelligent pavement defect detection is also growing rapidly. Pavement defect detection is characterized by strong timeliness, diverse detection environments, rich and effective information, and complex environmental factors. Analyzing vehicle suspension video data (obtained through cameras or sensors installed to monitor the vehicle suspension system in real-time during driving) is crucial for acquiring real-time road information, and this research holds immense significance for advancing autonomous driving technology.

Roads are an important part of the modern transportation system, which is of great importance to social and economic development. However, road surface defects, such as potholes, cracks, road wear, and damage (see Figure 1 for complex road defects), not only endanger driving safety but also lead to vehicle damage and traffic congestion [1]. The vehicle-borne pavement defect detection technology will help road maintenance departments plan and implement maintenance work more effectively. Through real-time monitoring of road conditions, managers can accurately determine which road sections need maintenance and what maintenance measures should be taken. This helps to reduce resource waste, improve work efficiency, and extend the service life of roads. The vehicle suspension's mechanical system can also be dynamically adjusted to enhance the driving experience and extend its mechanical life. Therefore, the real-time monitoring and

evaluation of the condition of road surfaces are very important for road maintenance and driving safety. Detection technology based on computer vision is highly valuable due to its efficiency and accuracy [2].

Figure 1. Complex types of road surface defects. 0 is the speed bump, 1 is the manhole cover, 2 is the pothole, and 3 is the facture.

Pavement defect detection technology utilizes sensors and vehicle-based systems to non-invasively identify pavement issues. These sensors encompass a range of technologies, including cameras, laser scanners, radars, ultrasonic sensors, and inertial navigation systems. They capture road images, point cloud data, or vibration information, transmitting them to the vehicle's computer system for analysis. Real-time processing of sensor-generated data is essential to extract details about pavement defects, such as their location, size, type, and severity. Leveraging computer vision algorithms centered around target detection, due to their accuracy and stability, holds significant application value in related fields.

Research on target detection technology can be traced back to the 1980s. Early methods mainly relied on hand-designed features and traditional machine learning algorithms, such as edge detection, color features, and template matching. These methods perform well in specific scenes, but their detection ability for complex backgrounds and multi-scale objects is limited. To deal with the problem of multi-scale objects, researchers introduced sliding window technology, which involves sliding a window across the image and using a classifier to determine whether the window contains the target. A representative of this method is the Viola-Jones [3] detector, which has achieved success in face detection using Haar features and the AdaBoost classifier. However, these methods are still limited by the selection of features and the demand for computing resources.

In 2012, AlexNet [4] was successfully applied to the ImageNet large-scale visual recognition challenge, marking a breakthrough in convolutional neural networks (CNNs) in the task of image recognition. The method of deep learning has been gradually introduced into the field of target detection. Driven by deep learning, more rapid and accurate target detection methods have emerged. In 2016, YOLO (you only look once) proposed a real-time target detection method, which modeled the target detection task as a regression problem and simultaneously generated predictions for the location and category of targets. The current main two-stage target aggregation algorithms include Fast R-CNN [5], Faster R-CNN [6], Mask R-CNN [7], SPP Net [8], etc. Mainstream one-stage target aggregation networks include SSD [9], EfficientDet [10], and YOLO [11–15].

This preserves the advantages of swift detection. Among the target aggregation algorithms, the YOLO series algorithm currently stands as the most extensively utilized

one. Having undergone multiple iterations, it has gradually enhanced its accuracy while also increasing the number of parameters. As the preferred lightweight network, YOLOv5s stands out as the most easily deployable network for embedded devices. In this paper, YOLOv5 has undergone modifications to better suit the detection of road defects and address the inefficiency in recognizing road defects within complex environments.

In the research of road defect detection technology, past researchers have conducted studies to varying extents on traditional methods, deep learning methods, and integrated methods [16–20]. Among these, deep learning methods have shown great potential due to their outstanding feature learning capabilities, the ability to automatically extract features related to road defects, the capacity to handle different types of road defects, and the potential for achieving strong performance and generalization through end-to-end training.

Currently, road defect detection algorithms face various challenges. Firstly, due to the diversity and complexity of road defects, algorithms encounter difficulties in accurately identifying various defect situations, including but not limited to cracks, potholes, and pavement damage. Secondly, the robustness of algorithms is crucial, given the variability in lighting, weather conditions, and traffic situations in real road environments, as these factors can impact the surface features and visibility of defects. Additionally, the acquisition and annotation of large-scale datasets pose challenges, as obtaining road defect data requires substantial time and resources, and ensuring accurate labeling of data is essential. Furthermore, real-time performance and efficiency are critical concerns, especially in scenarios such as highways where algorithms need to process large amounts of data rapidly and make accurate judgments. Consequently, the development of road defect detection algorithms necessitates a comprehensive consideration of factors such as accuracy, robustness, data availability, and real-time capabilities.

In previous research on deep learning methods, researchers often directly adopted generic models such as YOLOV5S and YOLOV5N, introducing relevant data related to road defects for training and making adjustments mainly at the parameter level. This has left significant room for optimization in terms of accuracy, recognition efficiency, and lightweight modeling. To address this, we took an approach focused on the network structure. We attempted to improve the shortcomings of previous research by adding, modifying, and optimizing network modules. Thus, a breakthrough in algorithmic efficiency and lightweight modeling was achieved.

To validate the effectiveness of the modules used in our algorithm, we took YOLOV5S, a widely used baseline model, and conducted simulated experiments on the same dataset and environment. This was done to verify the enhancements provided by our algorithm. Additionally, we conducted ablation experiments on each module separately to confirm the effectiveness of each individual component. Simultaneously, we determined a recommended range of parameters through numerous comparative experiments. The aim was to assist users in optimizing the utilization of our model in their experiments. Furthermore, we conducted stress tests on the model, offering insights into its relative effectiveness when operating under limited computational resources.

In order to meet the experimental requisites and uphold the precision and objectivity of the study, we employed databases previously utilized in relevant research and collected additional datasets within mainland China for experimentation. Detailed information on the data sources and other pertinent databases is presented in the Section 4 of this paper. This approach seeks to concurrently validate the model's effectiveness in comparison with prior models and account for potential idiosyncrasies in mainland China's road conditions.

Firstly, the C3 module in the head of the backbone layer in the YOLOv5 network is replaced by the C3-TST module, and the defect features are captured and enhanced by the C3-TST module. Secondly, the attention mechanism established by BiFormer is introduced into the head layer of the network structure to further improve the efficiency and detection accuracy. Thirdly, the activation function of Leaky ReLU is added to the network structure of YOLOv5, which further improves the stability and accuracy of regression. Finally, the performance of various improvement algorithms is tested on publicly available and self-

constructed datasets, the result data are plotted, and the data are analyzed in conjunction with the images.

2. Materials and Methods

Introduction to the YOLOv5 Algorithm

The YOLO series is a typical representative of a single-stage structure detection algorithm, which was iteratively developed for the fifth generation of YOLOv5. It can change the model structure by adjusting the width and depth of the network to take into account the detection accuracy and speed. The network structure of the original YOLOv5 includes an input layer, backbone layer, neck layer, and head layer, as shown in Figure 2.

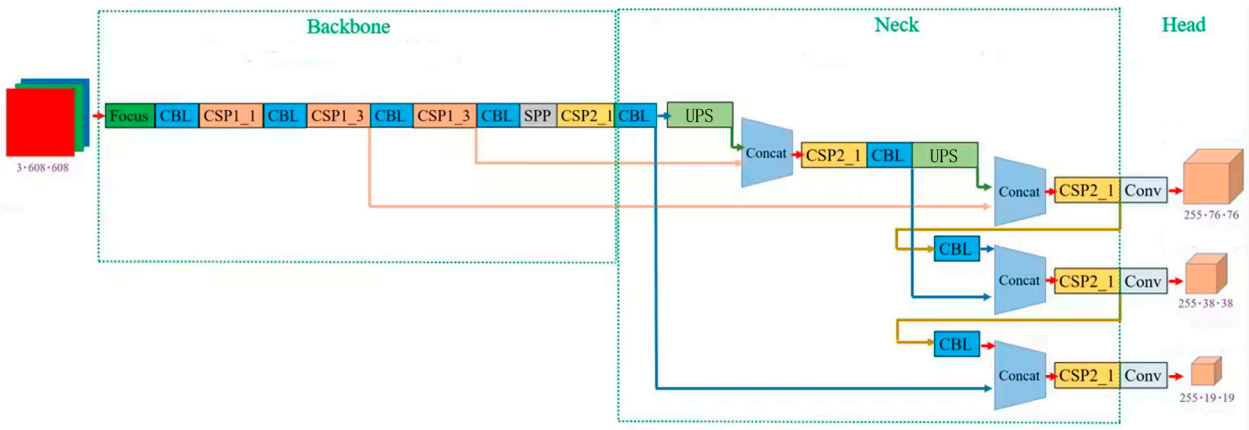

Figure 2. The network structure of the original YOLOv5.

The backbone layer is responsible for extracting feature representation from the input layer and usually uses lightweight architectures such as cspparknet53 or cspparknetlite. The head layer is responsible for target detection at different scales. YOLOv5 realizes multi-scale detection by performing detection on feature maps with different depths. This is achieved by introducing anchor boxes of different sizes and feature pyramids into the network. This design enables the model to detect targets of different sizes and proportions at the same time. For each anchor box, the detection head will return the position information of the target bounding box, usually including the central coordinates, width, and height of the bounding box. At the same time, it also predicts the target category probability score of each bounding box. Since the output of each detection head has different resolutions when detecting on multiple scales, it is necessary to associate them with anchor boxes and map their output to the same size as the input image to obtain the final detection result [1].

YOLOv5 uses multiple loss terms to balance the regression accuracy of the target position, the classification accuracy of the target category, and the attention to difficult samples. It consists of three main parts—CLS, obj, and box. Box (bounding box loss): the purpose is to correct the predicted border position so that it is as close to the real value as possible. YOLOv5 uses the mean square error (MSE) to calculate the loss of this part. Obj (objectness loss): the goal of this loss function is to better distinguish which areas contain detection targets and which areas do not. This is calculated by the predicted object confidence. For regions containing targets, the model will be forced to predict a high confidence level; for regions without targets, the model will predict a lower confidence. Cla (class loss): this part of the loss function aims to correct the predicted category to make it conform to the real category as much as possible. Cross entropy loss is usually used for calculation [21].

3. Implemented YOLOv5 Algorithm

3.1. Adjust Model Activation Function

In vehicle sensor videos, YOLOv5 performs poorly in detecting road defects. The main reason is the complexity and diversity of the road environment and the wide variety of road defects, which are often difficult to discern manually from images. Therefore, the effect of target aggregation is optimized by adjusting the YOLOv5 activation function.

In the original YOLOv5 backbone network cspparknet53, the Mish activation function is used. The Mish function is a self-regularized activation function. Its mathematical expression is as follows:

$$f(x) = \mathrm{x}\cdot\tanh(\mathrm{softplus}(\mathrm{x})) \quad (1)$$

The advantage of the Mish function is that it can improve the performance of neural network models, especially for deeper network structures, which can yield better performance than ReLU and some of its variants. In the last part of the network, YOLOv5 uses the Leaky ReLU activation function as the default activation function. The mathematical expression of Leaky ReLU is as follows:

$$f(x) = \max(0.01\mathrm{x}, \mathrm{x}) \quad (2)$$

when x < 0, the slope of the activation function is 0.01, which improves the stability of the model.

Leaky ReLU (Leaky rectified linear unit) [22,23] is a variant of the traditional ReLU (rectified linear unit) activation function. The value of ReLU is 0 in the interval where its activation output is negative, which may cause neurons to die in this interval; that is, the neurons may become unable to transfer and learn the gradient. This problem is called "ReLU death". For the deep learning model with a high learning rate, this may lead to more neuron death and ultimately affect the performance of the model. Figure 3 shows their respective function graphs.

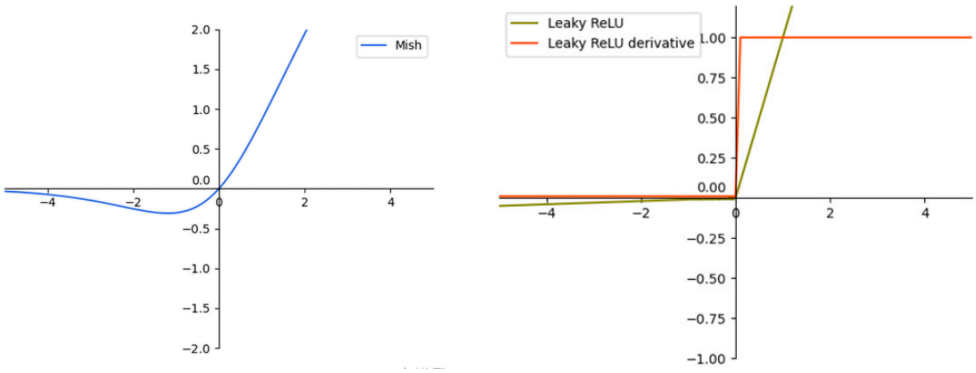

Figure 3. Image of Mish function (**left**) and Leaky ReLU function (**right**).

The solution proposed to address this issue is the Leaky ReLU activation function, expressed mathematically as $f(x)$ = max (0.01x, x). When x < 0, this function introduces a small non-zero slope (e.g., 0.01), allowing negative activations to propagate forward, effectively preventing the problem of neuron 'death' observed in traditional ReLU. Compared to ReLU, Leaky ReLU enhances the model's expressive capacity by providing a non-zero output for negative intervals. In practical applications, it exhibits remarkable robustness.

We use Leaky ReLU to replace the original Mish activation function in the improved network backbone layer. This makes our model more efficient and accurate in dealing with the problem of road defect detection, and the overall system can be more lightweight.

3.2. C3-TST Block

Almost all existing models are based on CNNs and use convolution or pooling operations in the backbone layer. Due to the local connectivity of convolution, this kind of

backbone is often more suitable for local information and ignores global information [24,25]. This makes this kind of model often perform poorly on surface defect images in practice.

Inspired by Vit, we use self-attention at the end of the trunk to capture and extract richer context information and image features. We use Swin Transformer blocks [26] to create the C3-TST block. Instead of directly applying global self-attention to high-resolution images, we employ an attention mechanism based on shift windows. Direct global self-attention requires a significant number of parameters, resulting in substantial computational and memory overhead. This makes it challenging for the model to learn, train, and deploy, especially on edge devices. To solve this problem, we refer to the implementation idea of C3STR and use C3-TST to divide the image into non-overlapping windows of the same size, performing local self-attention within each window. At the same time, a hierarchical attention strategy is adopted to obtain more effective key information from a global perspective. Through the segmentation strategy of region and level, we are able to achieve effective attention to global information while reducing parameters.

A twin Swin Transformer block (TST) is composed of the following parts: two-layer normalization layers (Norm), one shift-window attention layer (SWA), one bi-level-attention layer (bi-attention), and one multi-layer perceptron (MLP) layer [27,28]. The normalization layers are used to normalize the input data, making the data stable and easy to process. The shift window attention layer is designed to calculate the shift window attention. The MLP layer is used to integrate attention information, extract features, and change image dimensions.

It is worth noting that, inspired by st-ca_YOLOv5, we use paired Swin Transformer blocks to build C3-TST modules [29], as shown in Figure 4.

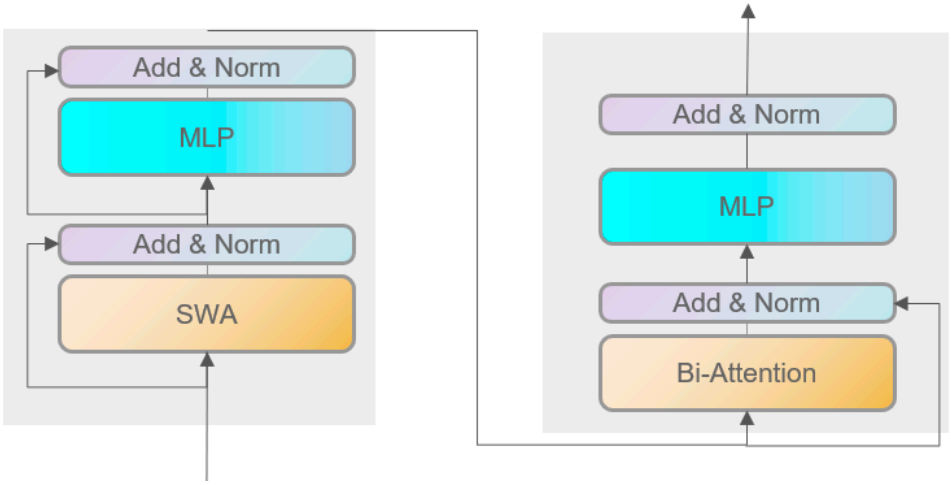

Figure 4. Framework diagram of two successive Swin Transformer blocks.

By combining SWA and bi-attention, the model is able to integrate different attention patterns and capture global features during this process. This solves the problem of the original model's difficulty in paying attention to global information.

What is more, by designing C3-TST in this way, we can obtain a model with better performance, fewer parameters, and a more stable convergence process, and we can prevent model degradation as the network depth increases. C3-TST consists of three convolutional blocks and n $\times$ twin Swin Transformer blocks (TSTs), as shown in Figure 5. And the original C3 block structure is shown in Figure 6.

$$SiLU(x) = x \cdot \sigma(x)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

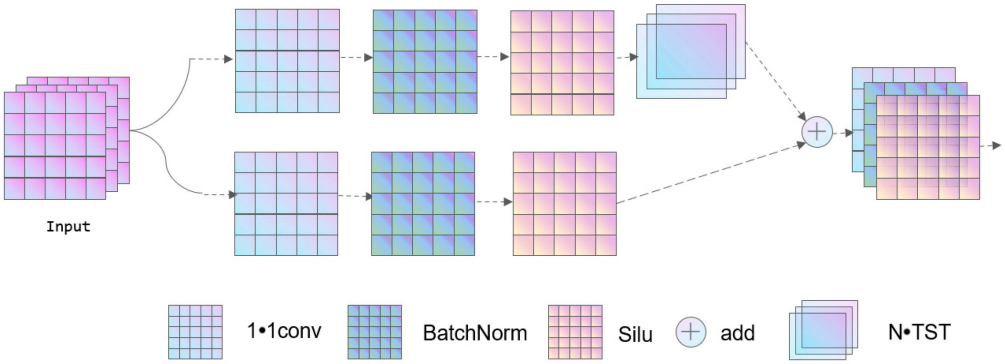

Figure 5. The structure of the C3-TST module.

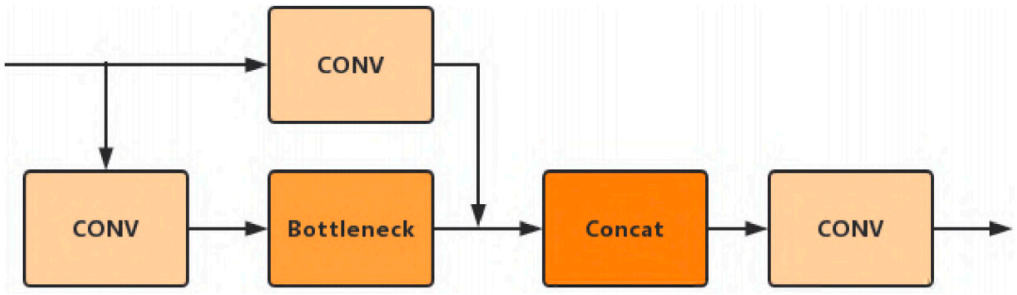

Figure 6. Original C3 block structure.

Each convolutional block consists of a 1 × 1 convolutional layer, a batch normalization layer, and a Silu activation function. The input feature is divided into two parts: the first part consists of a convolutional block and n × TST processing, and the second part is processed by a sub-convolutional block. After adding the two parts, another convolutional block is used to restore the original number of channels. This enables the C3-TST block to adapt to channels and become more versatile [30,31].

3.3. Bi-Level-Routing Attention Head

For actual natural images, road defects may be quite diverse. It is difficult to extract the direction and position information in the image based on the CNN model. The existing attention mechanism in the application of surface defect detection is not flexible and portable. Therefore, it is difficult for the model to detect small defects accurately and quickly in practice.

To alleviate the scalability problem of MHSA (multi-head self-attention), some previous methods proposed different sparse attention mechanisms [32,33], in which each query only focused on a small number of key-value pairs, rather than all. However, these methods have two common problems:

1. They use manual static mode (unable to adapt);
2. They share the sample set of key-value pairs in all queries (it is impossible for them not to interfere with each other).

Therefore, we use BiFormer to construct the attention mechanism to capture the perceptual information about the channel, direction, and position.

BiFormer uses overlapping block embedding in the first stage and a block merging module in the second to fourth stages to reduce the input spatial resolution and increase the number of channels. Then, it uses continuous BiFormer blocks for feature transformation. It should be noted that at the beginning of each block, the depth convolution is used to implicitly encode the relative position information. Then, the bra module and the multi-layer perceptron (MLP) module with the expansion rate are applied in turn for cross-location relationship modeling and each location embedding, respectively, as shown in Figure 7.

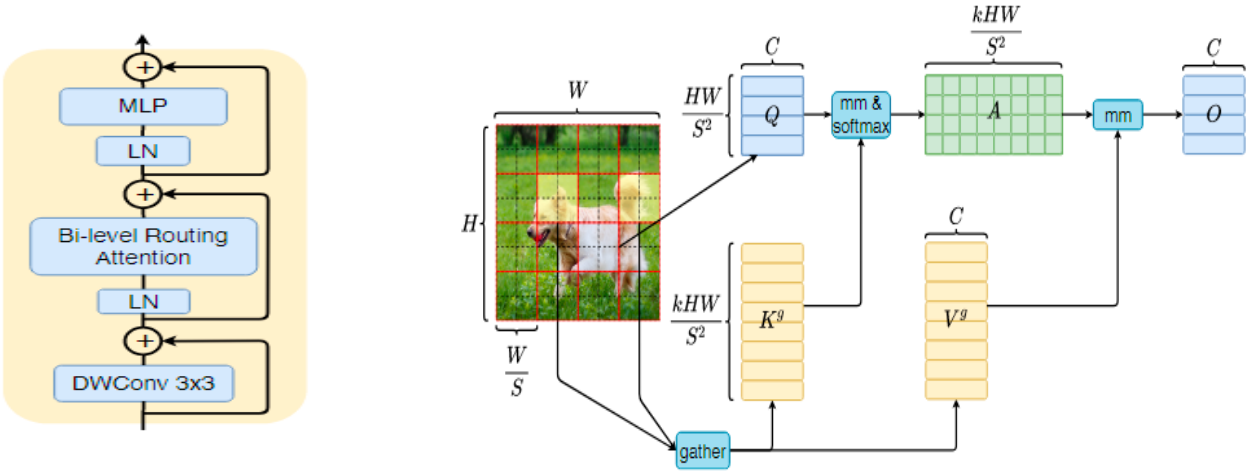

Figure 7. Structure of bi-level routing attention.

After paying attention to the coordinates of these features, the head can better separate the features along different spatial directions and retain accurate position information. Therefore, the representation of objects of interest in the feature graph is enhanced. Thus, the bi-level-routing attention head can monitor objects quickly and accurately.

4. Results

4.1. Dataset Preparation

During the experiment, we utilized a publicly available dataset [34]. This dataset consists of several hundred thousand images captured within Japan using in-vehicle mobile devices, specifically smartphones. Due to limitations in both length and relevance, for more detailed information regarding the dataset, please refer to the provided URL link associated with the dataset.

And in order to meet the experimental requirements, we also conducted on-site data collection based on Chinese road conditions. The data collection process was as follows:

A. For the acquisition of the dataset, we used real-time driving videos of the road in a certain region of China, used FFmpeg v6.1.1 software to cut the video screen into pictures, and selected the effective part of the total 1.45 g data for use.
B. We used Labellmg v1.8.1 software to complete the labeling task and carried out preliminary data cleaning.

The experimental environment can be found in the appendices (Figure A2 and Table A1).

It is important to note that, within the current research landscape, the optimal hyperparameters for the model cannot be precisely determined through exact mathematical formulas. We have iteratively conducted experiments to obtain relatively favorable parameters and recommend that readers use parameters within a similar range. However, it should be acknowledged that the possibility of achieving even more optimal parameters cannot be excluded. For a more nuanced parameter optimization, optimization algorithms such as genetic algorithms can be employed.

4.2. Evaluation Index

For the sake of comparison with the widely employed original YOLO model, this experiment opted for the evaluation criteria outlined in the initial YOLO model paper [35]: mAP (mean average precision). This metric was used as the benchmark for comparison against the baseline.

"mAP" is a commonly utilized metric for evaluating the performance of object detection models. It serves as a comprehensive measure assessing the accuracy of object identification and localization in object detection tasks.

mAP amalgamates information from precision–recall curves, providing a comprehensive evaluation of detection results across different object classes. Its computation involves the following steps:

A. Calculation of Precision and Recall: For each class, the model computes precision and recall at varying confidence thresholds. Precision signifies the proportion of correctly identified positive samples out of all predicted positives, while recall denotes the fraction of true positives detected by the model.
B. Precision–Recall Curve: precision–recall curves are constructed based on the computed precision and recall values across different confidence thresholds.
C. AP Computation (Average Precision): The area under the precision–recall curve is calculated for each class, representing the average precision (AP) value for that class.
D. mAP Calculation: mAP is obtained by averaging the AP values across all classes, offering an overall assessment of the model's performance across the entire dataset.
E. mAP serves as a holistic performance metric, providing a unified evaluation of detection outcomes for diverse object classes, thereby offering a comprehensive assessment. In the context of training and optimizing object detection models, achieving a high mAP typically indicates superior performance in detecting objects across multiple categories.

In the pursuit of training object detection models, optimizing the model to enhance mAP is a common objective. This involves adjusting the model architecture, refining loss functions, and employing data augmentation techniques, among other strategies aimed at improving the model's detection capabilities.

The other evaluation parameters and their specific meanings used in this experiment are as follows:

Precision: indicates how many of the samples predicted as positive categories in the model are positive categories. It is used to evaluate the classification accuracy of the model, especially when the cost of false positive examples is high.

Recall: measures how many of the real positive category samples of the model are successfully predicted as positive categories. The recall rate is used to evaluate whether the model can capture all positive samples, especially when the cost of false negative cases is high.

F1 score: the harmonic average of accuracy and recall rate, which provides an indicator for a comprehensive evaluation of model performance. The F1 score is suitable for handling unbalanced datasets and considering the balance of accuracy and recall rate.

$$Percision = \frac{TP}{TP + FP} \cdot 100\% \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \cdot 100\% \tag{4}$$

$$F1 = \frac{2 \cdot (P \cdot R)}{P + R} \cdot 100\% \tag{5}$$

TP (True Positives): True positives indicate that the model correctly predicts samples of the positive class as positive.

FP (False Positives): False positives represent cases where the model incorrectly predicts samples of the negative class as positive.

FN (False Negatives): False negatives occur when the model incorrectly predicts samples of the positive class as negative.

4.3. Ablation Study

To study the effectiveness of the proposed model components, we conducted ablation studies on datasets in a large number of experiments. We used YOLOv5s [25] as the baseline and set up two comparative experiments.

During the training process, we divided the dataset to be processed into multiple batches, each containing 320 images, for the purpose of grouping training and testing.

In our ablation experiment, we trained our model alongside the YOLOv5s and YOLOv5n models, pushing them to their limits using the same dataset and rounds of training. Specifically for our model, we individually trained it based on the original model while adding each new component (as depicted in the figure; for example, 'YOLOv5 + Leaky ReLU' signifies a new model incorporating a modified Leaky ReLU function, and the same principle applies to other models). To assess the effectiveness and applicability of each component, we also conducted separate training for every combination thereof.

The specific process of the ablation experiment is as follows:

a. Feature ablation:

Eliminate one or more features one by one or in batches, and retrain the model. This means that in each experiment, a feature or combination of features will be removed.

b. Model evaluation:

For each ablation experiment, evaluate the model performance using the test set. Compare the performance of the model before and after ablation, such as accuracy, precision, recall, F1 score, and other indicators.

Table 1 and the figure for quantitative comparison are as follows (Table 1):

Table 1. Evaluation parameters of different models.

	P/%	R/%	Map-50/%	F1/%
YOLOv5n	90.0	90.3	91.7	38.1
YOLOv5s	90.2	90.9	94.4	44.2
YOLOv5 + Leaky ReLU	93.7	94.2	95.5	43.5
YOLOv5 + Leaky ReLU + C3-TST	91.2	95.3	95.4	39.6
YOLOv5 + Leaky ReLU + C3-TST + bi-head	89.8	96.3	96.4	46.0

Effectiveness of Leaky ReLU: after improving the activation function, the presented data significantly improved all indicators; P and R increased by 3.5% and 3.3%, respectively, compared with the baseline; map and F1 increased by 2.7% and 6.1%. The contribution of the optimized activation function is confirmed.

Effectiveness of C3-TST: adding C3-TST to various indicators did not produce a significant improvement in accuracy, but the overall judgment stability can be significantly improved by observing the data. See Figure A1 for details.

Effectiveness of bi-head: after comparing (5), (3), and (1), it can be seen that bi-head has significantly improved the indicators of the model, and the use of C3-TST ensures the lightness of the model.

The figures below shows the results of the final model YOLOv5s and YOLOv5n after 100 epochs (Figures 8–10). Appendix A contains additional training results data and corresponding diagrams (Figure A1). These data demonstrate that the model exhibits strong stability.

Examples of training, testing, and results can be found in the appendices (Table A1 and Figure A2).

Building upon this foundation, we conducted stress tests on the model to address potential unexpected events in real-world scenarios, such as insufficient voltage or reduced computing power due to hardware aging. By operating the model with the application memory limited to below thirty percent (as well as under reduced voltage conditions), the experimental results still ensured accuracy in the majority of situations. In cases of insufficient computing power, the model prioritizes attention on more severe road damage to minimize the impact of similar events (Figure A3). However, comprehensive resolution of such issues may still require optimization in the mechanical structure or the addition of auxiliary equipment.

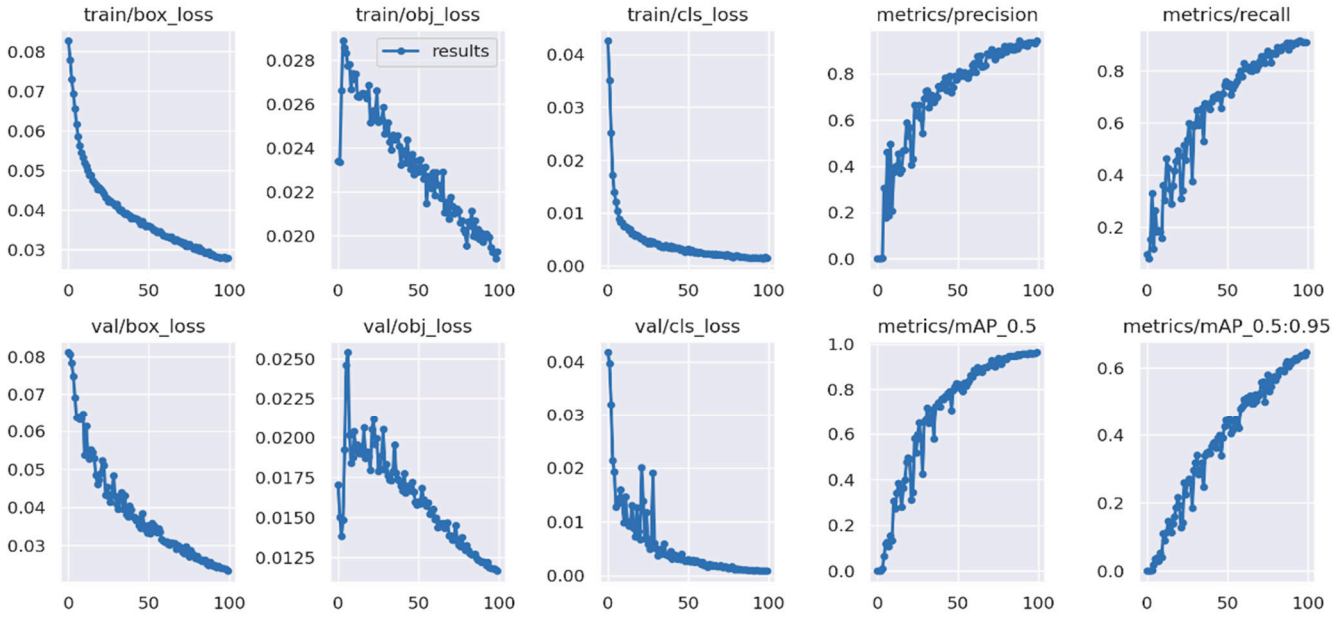

Figure 8. Evaluation parameters of the final model.

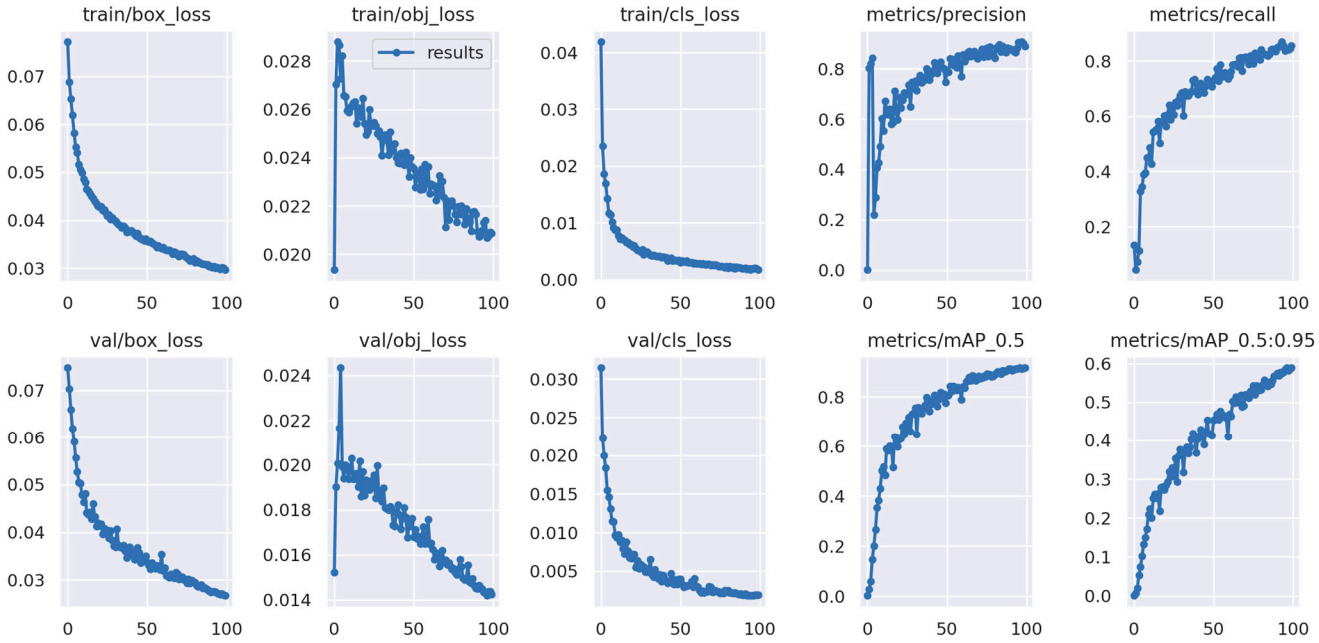

Figure 9. Evaluation parameters of YOLOv5n.

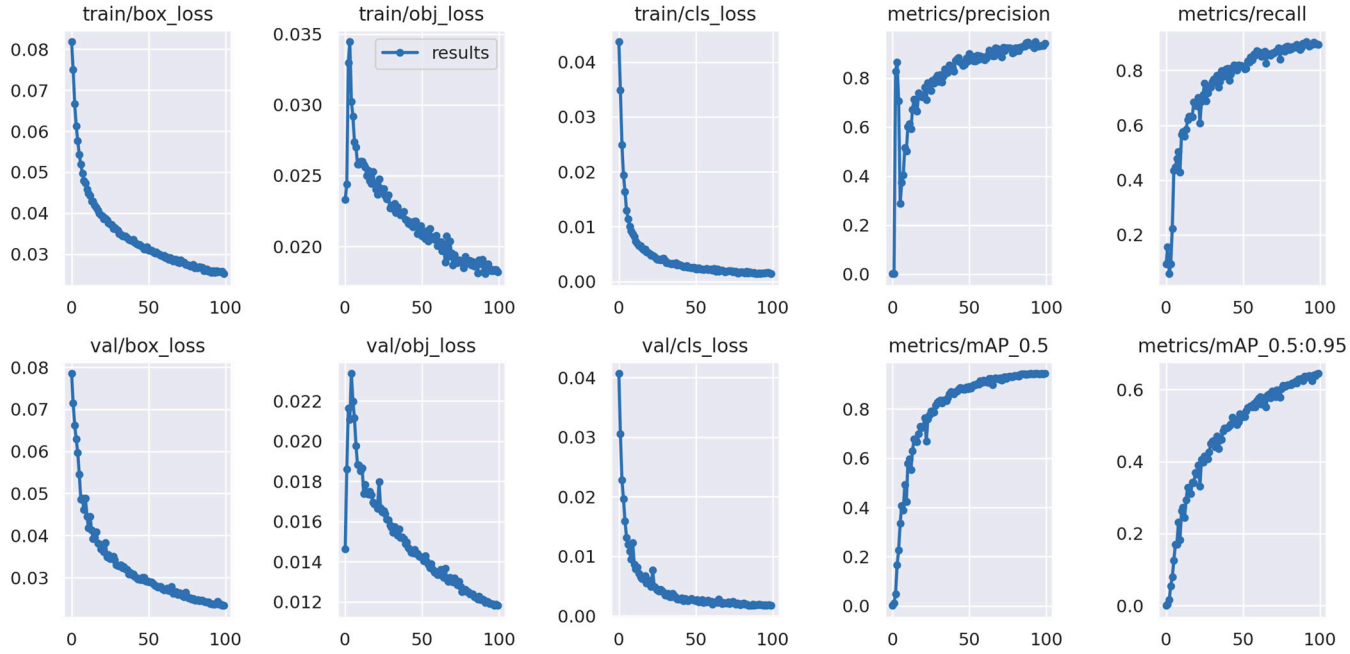

Figure 10. Evaluation parameters of YOLOv5s.

5. Discussion

In previous research on autonomous vehicle decision-making, researchers paid more attention to the dynamic interaction with surrounding vehicles, such as the prediction of vehicle intention based on MSNE. The related research fields are relatively mature [21,36,37]. Pavement defect detection is more frequently used in the field of road maintenance [38,39]. The related algorithms are mostly medium- and high-level deep learning networks and common recognition modules. There are also self-vehicle trajectory inference methods similar to RSC [40]. Compared with previous research, The algorithm used in this paper has specificity for the problem of road defect detection, and the design is lightweight. It can maintain recognition speed without sacrificing accuracy and is suitable for use in vehicle suspension preview systems. A large number of experimental results show that our method is effective and superior to the existing methods. However, due to the relatively small sample size and short training time, the model in this paper still has some room for improvement. At the same time, the problem of video capture devices being covered by stains in bad weather means that it still needs to be integrated with mechanical design to achieve stable effects in real-world environments. An important future research direction is computer vision recognition of non-visible light imaging, which can be used as a supplement in special cases. Similarly, the network structure of this model still has room for further optimization, and future work will focus on further research into attention mechanisms and the development of lightweight parameters.

6. Conclusions

In this paper, we propose an improved YOLOv5 model based on the Swin Transformer block and bi-head and focus on the field of road defect detection. The model is named TST-BI YOLOv5 and has excellent performance in real-time road monitoring. By using the Swin Transformer block to design the C3-TST module, we embedded the C3-TST module into the end of the trunk to enhance the ability to extract non-local feature information. To improve the detection accuracy and facilitate the detection of multiple obstacles, we introduced bi-level-routing attention to form a bi-head structure in the detection head. Specifically, we use lightweight modules such as Swin Transformer block and bi-level attention to improve the detection accuracy and minimize the change in the number of model parameters.

From the conclusive results, it is evident that our research still has certain shortcomings and areas open for optimization. These include the algorithm's diminished performance after extensive training and the necessity for better equilibrium between recognition accuracy and efficiency. To address these challenges, a viable solution lies in dynamically selecting and invoking algorithms based on the system's available resources. During the model construction, specific classifications were assigned to certain special road defects (such as speed bumps, barriers, etc.), enabling the model to possess identification capabilities for these particular obstacles. However, for obstacles not utilized during training, the model only possesses basic recognition abilities, and its generalization capability is relatively weak. This issue can be addressed by augmenting the model's training dataset. Similarly, we will continue our research to further enhance the algorithm.

Author Contributions: Conceptualization, X.Y. and J.Q.; methodology, X.Y. and J.Q.; software, X.Y. and J.Q.; validation, J.Q.; formal analysis, J.Q.; investigation, J.Q, T.Z. and S.B.; resources, J.Q.; data curation, X.Y.; writing—original draft preparation, J.Q.; writing—review and editing, X.Y. and J.Q.; visualization, J.Q.; supervision, X.Y.; project administration, X.Y.; funding acquisition, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The work was funded by the National Natural Science Foundation Project, grant number 52072157.

Data Availability Statement: The dataset was collected independently by institutions and is not publicly available due to privacy restrictions.

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

Appendix A

Table A1. Experimental equipment.

OS	CPU	GPU	Memory
WINDOWS 11	I7 13,700 KF 3.4 GHZ	RTX3060TI-8G	DDR5 32 G 5600 MHZ

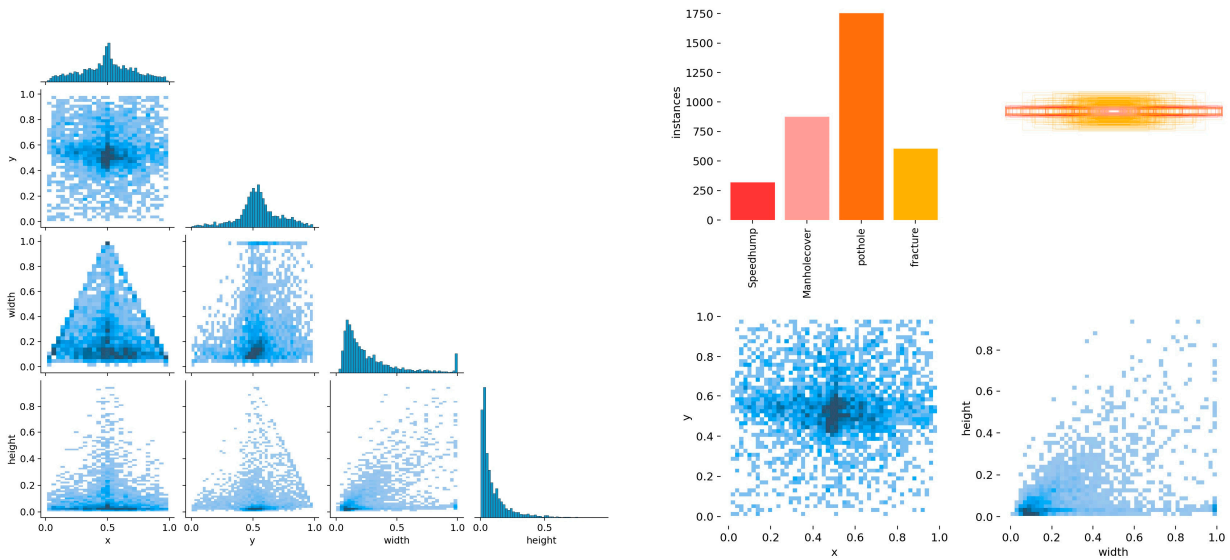

Figure A1. Labels_correlogram of the Leaky ReLU + C3-TST model.

```
momentum: 0.74832
weight_decay: 0.00025
warmup_epochs: 3.3835
warmup_momentum: 0.59462
warmup_bias_lr: 0.18657
```

Figure A2. Partial parameters.

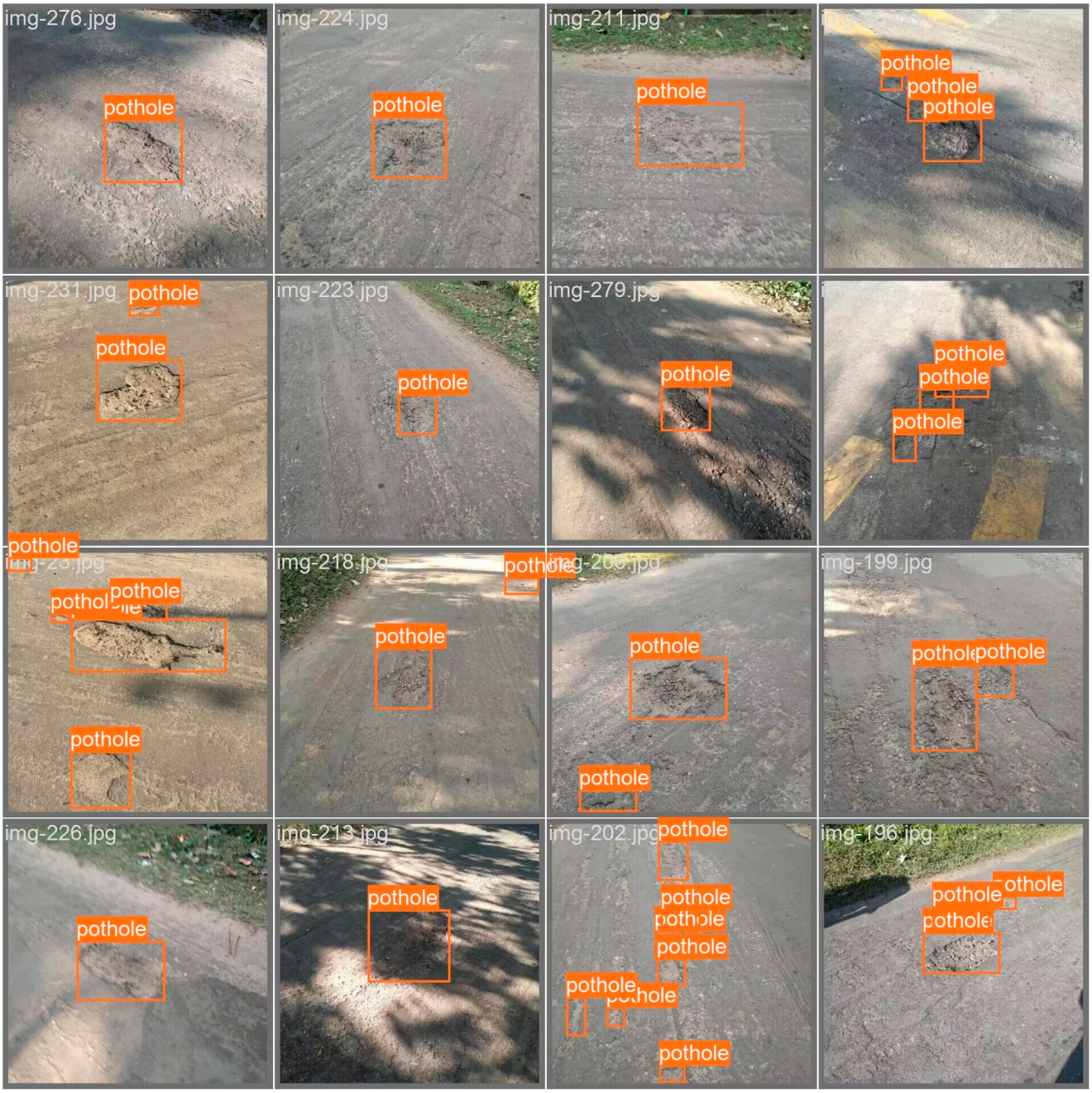

Figure A3. Testing examples.

References

1. 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, Seattle, VA, USA, 17–21 June 2016.
2. Demasi, F.; Loprencipe, G.; Moretti, L. Road safety analysis of urban roads: Case study of an Italian municipality. *Safety* **2018**, *4*, 58. [CrossRef]
3. Viola, P.; Jones, M. Robust real-time face detection. In Proceedings of the Procedures Eight IEEE International Conference on Computer Vision ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; p. 747. [CrossRef]
4. Smirnov, E.A.; Timoshenko, D.M.; Andrianov, S.N. Comparison of Regularization Methods for Imagenet Classification with Deep Revolutionary Neural Networks. *AASRI Procedia* **2014**, *6*, 89–94. [CrossRef]
5. Girshock, r. Fast R-CNN. In Proceedings of the Procedures of IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 1440–1448.
6. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Fast R-CNN: Towards real time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. He, K.; Gkioxari, G.; Dollar, P.; Girsick, R. Mask R-CNN. *arXiv* **2018**, arXiv:1703.06870.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in DCEP revolutionary networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
9. 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 Procedures of European Conference on Computer Vision, Berlin, Germany, 11–14 October 2016; pp. 21–37.
10. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and effective object detection. In Proceedings of the Procedures of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 13–19 June 2020; pp. 10781–10790.
11. Jocher, G.; Stoken, A.; Borovec, J.; Chaurasia, A.; Xie, T.; Liu, C.; Abhiram; Laughing; tkianai; yxNONG; et al. ultralytics/YOLOv5: v5.0—YOLOv5-p6 1280 Models, AWS, su-visit.ly and Youtube Integrations. Zenodo. 2021; Available online: https://www.semanticscholar.org/paper/ultralytics-yolov5:-v5.0-YOLOv5-P6-1280-models,-and-Jocher-Stoken/fd550b29c0efee17be5eb1447fddc3c8ce66e838 (accessed on 23 November 2023).
12. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y. YOLO4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934v1.
13. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.027672018.
14. 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. 6517–6525.
15. Li, P.; Shan, S.; Zeng, P.; Wei, H. Improved YOLOv5 algorithm for surface defect detection of solar cell. In Proceedings of the 35th China Control and Decision Making Conference, Yichang, China, 20–22 May 2023; pp. 379–383.
16. Zhang, L.; Satta, R.; Merialdo, B. Road damage detection and classification in smartphone images. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016.
17. Guo, X.; Hu, B.; Hu, L.; Yang, Z.; Huang, L.; Li, P. Pavement Crack Detection Method Based on Deep Learning Models. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 13. [CrossRef]
18. Singh, J.; Shekhar, S. Road Damage Detection and Classification in Smartphone Captured Images Using Mask R-CNN. *arxiv* **2018**, arXiv:1811.04535.
19. Verma, A.; Jain, A. Road damage detection and classification using convolutional neural networks. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2018.
20. Zhou, S.; Yuan, Y.; Guo, C.; Li, J.; Lei, Z. A road crack detection method based on deep learning. *IEEE Access* **2019**, *7*, 31560–31569.
21. Bochkovskiy, A.; Chien, Y.W.; Hong, Y.; Liao, M. YOLOv5: End to end real time object detection with YOLO. *arXiv* **2021**, arXiv:2103.06317.
22. Sadeghi, F.; Balog, M.; Popovic, M.; Gross, M. Gated activation functions. In Proceedings of the Advances in Neural Information Processing Systems (NEurIPS), Vancouver, BC, Canada, 8–14 December 2019.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surmounting human level performance on Imagenet classification. In Proceedings of the Procedures of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chille, 7–13 December 2015.
24. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
25. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D. Training data effective image transformers & disintegration through attention. *arXiv* **2020**, arXiv:2012.12877.
26. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *Computer Vision—ECCV 2014*; Lecture Notes in Computer Science; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; Volume 8689. [CrossRef]
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
28. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
29. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swing transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.

30. Arya, D.; Maeda, H.; Kumar Ghosh, S.; Toshniwal, D.; Omata, H.; Kashiyama, T.; Sekimoto, Y. Crowdsensing-Based Road Damage Detection Challenge (CRDDC'2022). In Proceedings of the 2022 IEEE International Conference on Big Data (IEEE Big Data), Osaka, Japan, 17–20 December 2022; pp. 6378–6386. Available online: https://github.com/sekilab/RoadDamageDetector/ (accessed on 23 November 2023).
31. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arxiv* **2015**, arXiv:1502.03167.
32. Yang, W.; Wu, H.; Tang, C.; Lv, J. YOLOv5: Improved YOLOv5 based on swing transformer and coordinated attention for surface defect detection. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, QLD, Australia, 18–23 June 2023; pp. 1–8. [CrossRef]
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, I.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
34. Prahar, M.B.; Malhan, R.; Rajendran, P.; Shah, B.; Thakar, S.; Yoon, Y.J.; Gupta, S.K. Image-based surface defect detection using deep learning: A review. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 040801.
35. Tian, C.; Leng, B.; Hou, X.; Huang, Y.; Zhao, W.; Jin, D.; Xiong, L.; Zhao, J. Robust Identification of Road Surface Condition Based on Ego-Vehicle Trajectory Reckoning. *Automot. Innov.* **2022**, *5*, 376–387. [CrossRef]
36. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R. Biformer: Vision transformer with bi level routing attention. *arXiv* **2023**, arXiv:2303.08810.
37. Deng, H.; Zhao, Y.; Wang, Q.; Nguyen, A.-T. Deep Reinforcement Learning Based Decision-Making strategy of Autonomous Vehicle in Highway Uncertain Driving Environments. *Automot. Innov.* **2023**, *6*, 438–452. [CrossRef]
38. Lucente, G.; Dariani, R.; Schindler, J.; Ortgiese, M. A Bayesian Approach with Prior Mixed strategy Nash Equilibrium for Vehicle Intention Prediction. *Automot. Innov.* **2023**, *6*, 425–437. [CrossRef]
39. Nguyen, S.D.; Tran, T.S.; Tran, V.P.; Lee, H.J.; Piran, M.J.; Le, V.P. Deep learning-based crack detection: A survey. *Int. J. Pavement Res. Technol.* **2023**, *16*, 943–967. [CrossRef]
40. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]