ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20240422001485
Original scientific paper

Detection and Classification of Railway Track Surface Erosion-caused Holes and Scratches Defects Based on YOLOv5s

Congbo YANG, Kaharudin DIMYATI*, Mahazani MOHAMAD

Abstract: Railway tracks are exposed to various environmental conditions, which can lead to defects that affect the safe operation of trains and potentially cause accidents. This paper presents an approach for detecting and classifying two common types of railway track surface defects, namely erosion-caused holes and scratches, using the YOLOv5s algorithm based on convolutional neural networks (CNN) and computer vision. The proposed method is compared with the YOLOv3-Tiny algorithm to demonstrate its superiority in terms of precision, recall, and mean average precision (mAP). A custom dataset containing labeled images of both defect types was collected and used for training and evaluation. The experimental results show that YOLOv5s outperforms YOLOv3-Tiny. The improved performance of YOLOv5s can be attributed to its optimized architecture and better balance between speed and accuracy. The limitations of the study include the relatively small dataset size and the need for further validation in real-world scenarios. Future work will focus on expanding the dataset, investigating advanced data augmentation techniques, and integrating the model with other sensing modalities for robust and real-time defect detection in railway maintenance systems. The proposed approach has the potential to significantly improve the efficiency and effectiveness of railway track inspection and maintenance processes.

Keywords: CNN (convolutional neural network); railway surface defect detection; target detection; YOLOv5

1 INTRODUCTION

Railway transportation plays a crucial role in modern society, offering efficient and cost-effective means of passenger and freight transport. However, ensuring the safety and reliability of railway infrastructure, particularly the tracks, remains a significant challenge. Railway tracks are subjected to various environmental conditions and loads, which can lead to the development of defects such as erosion-caused holes and scratches [1]. These defects, if not detected and addressed in a timely manner, can compromise the structural integrity of the tracks and increase the risk of accidents [2].

Traditional manual inspection methods for railway track defects are time-consuming, labor-intensive, and prone to human error [3]. In recent years, computer vision and deep learning techniques have shown great potential in automating the defect detection process, improving efficiency and reliability [4]. Various algorithms, such as convolutional neural networks (CNNs), have been applied to detect and classify defects in different domains, including railway infrastructure [5]. However, in certain instances, the class of defects present may not be homogeneous, thereby rendering the inspection process ambiguous and more challenging [6]. There is a need for a robust and efficient method that can simultaneously detect and classify multiple types of track surface defects with high accuracy.

To address this research gap, we propose an approach based on the YOLOv5s algorithm, a state-of-the-art object detection model that leverages CNNs and computer vision techniques. YOLOv5s is known for its superior performance in terms of speed and accuracy compared to its predecessors, such as YOLOv3-Tiny [7]. By adapting and fine-tuning the YOLOv5s architecture for railway track defect detection, we aim to develop a robust and efficient system that can identify and classify erosion-caused holes and scratches with high precision and recall.

The main contributions of this study are as follows:

We present an application of the YOLOv5s algorithm for detecting and classifying two common types of railway track surface defects, demonstrating its superiority over the YOLOv3-Tiny model.

We collect and annotate a custom dataset containing labeled images of erosion-caused holes and scratches, which can serve as a valuable resource for future research in this domain.

We conduct extensive experiments to evaluate the performance of the proposed method, providing insights into its strengths, limitations, and potential for real-world deployment in railway maintenance systems.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on railway track defect detection and the application of deep learning techniques. Section 3 describes the research methodology, including the dataset collection, YOLOv5s architecture, and experimental setup. Section 4 presents the results and analysis of the experiments, comparing the performance of YOLOv5s with YOLOv3-Tiny. Finally, Section 5 concludes the paper and discusses future research directions.

2 LITERATURE REVIEW

2.1 Railway Track Defect Detection

The detection of railway track defects has been a subject of extensive research due to its critical importance in ensuring the safety and reliability of rail transportation. Various methods have been proposed to identify and classify different types of defects, such as cracks, corrosion, and deformation [8]. Traditional approaches rely on manual inspection by trained personnel, which is time-consuming, labor-intensive, and subject to human error [3]. To overcome these limitations, automated defect detection techniques using sensors and computer vision have gained significant attention in recent years [9].

One common approach for track defect detection is the use of machine vision systems, which employ cameras and image processing algorithms to identify anomalies in the track surface [10]. These systems can be mounted on inspection vehicles or installed along the trackside to capture images of the passing trains and track components [11]. The acquired images are then analyzed using various image processing techniques, such as edge detection, texture analysis, and pattern recognition, to detect and localize defects [12, 13]. However, traditional machine vision methods often struggle with the variability and complexity of real-world track conditions, leading to high false alarm rates and missed detections [14].

2.2 Deep Learning for Defect Detection

Deep learning techniques, particularly convolutional neural networks (CNNs), have revolutionized the field of computer vision and have shown remarkable performance in various object detection and classification tasks [15]. CNNs are capable of automatically learning hierarchical features from raw image data, making them well-suited for detecting and classifying defects in complex and variable environments [16].

Several studies have applied CNNs for railway track defect detection, demonstrating their potential for improving detection accuracy and efficiency. Faghih-Roohi et al. [17] proposed a deep CNN architecture for detecting and classifying multiple types of track surface defects, achieving an accuracy of 92% on a large dataset of track images. Shang et al. [18] developed a multi-scale CNN model for detecting fastener defects, which outperformed traditional machine vision methods in terms of precision and recall. Chen et al. [19] combined CNNs with transfer learning to detect rail surface defects, achieving high accuracy while reducing the need for large annotated datasets.

2.3 YOLO Algorithms for Object Detection

Among the various CNN-based object detection algorithms, the You Only Look Once (YOLO) family of models has gained significant popularity due to their real-time performance and high accuracy [20]. YOLO models treat object detection as a regression problem, directly predicting bounding boxes and class probabilities from the input image in a single forward pass [21]. This approach enables fast and efficient detection, making YOLO models suitable for real-time applications.

The original YOLO model has undergone several improvements and iterations, including YOLOv2 [22], YOLOv3 [23], and YOLOv4 [24], each introducing architectural enhancements and training techniques to boost performance. Li et al. [25] used YOLOv3 to detect surface defects on steel sheets, achieving high accuracy and real-time performance. In the study of erosion and spalling of concrete, Cui et al. proposed an algorithm based on an improved version of YOLOv3 to detect the disease damage of concrete and the results demonstrated that the improved YOLOv3 is more effective in this aspect of detection [26].

YOLOv5, the relatively new members' addition to the family, further optimizes the model architecture and training process, achieving relatively accurate and well-balanced results on various object detection benchmarks [27]. Furthermore, it exhibits a high level of maturity and performance, rendering it suitable for use in a multitude of industrial applications [28]. YOLOv5 offers several variants, such as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, catering to different speed and accuracy requirements. As demonstrated in the literature [29, 30], it is necessary to detect tomatoes and tomato leaves in complex environments with external disturbances, and to identify and classify the detected objects to determine if they are the desired targets. Several studies have applied YOLO models for defect detection in various domains, demonstrating their effectiveness and versatility. Ren et al. [31] employed YOLOv5 for detecting and classifying steel surface defects, outperforming traditional machine vision methods.

Despite the promising results of YOLO models in various defect detection tasks, their application to railway track defect detection remains limited. Most existing studies focus on detecting a single type of defect or employ complex architectures that may not be suitable for real-time deployment [32]. There is a need for a comprehensive evaluation of the performance and feasibility of YOLO models, particularly YOLOv5, for detecting and classifying multiple types of track surface defects in real-world scenarios.

In this study, we aim to address this gap by adapting and fine-tuning the YOLOv5s architecture for detecting and classifying erosion-caused holes and scratches on railway track surfaces. We compare the performance of YOLOv5s with YOLOv3-Tiny, a lightweight variant of the YOLOv3 model, to assess the trade-offs between speed and accuracy. Our work contributes to the growing body of literature on automated railway track defect detection and provides insights into the potential of YOLO models for real-world deployment in this domain.

3 RESEARCH METHODOLOGY

3.1 Experimental Equipment, Dataset Collection and Preprocessing

The dataset used in this study consists of 1,211 images of railway track surfaces, capturing two common types of defects: erosion-caused holes and scratches. The images were collected from various online sources and have different resolutions, ranging from 55 × 1250 to 224 × 224 pixels. To ensure the quality and consistency of the dataset, several preprocessing steps were applied.

The main piece of hardware required for the pilot was a GPU. Two PCs were used: one was a mainframe computer equipped with an RTX3070 discrete GPU, installed with Ubuntu 20.04, which served as the remote training terminal. The other PC was a regular laptop with Windows 11, which was used to remotely control the Ubuntu endpoint through the SSH (Secure Shell Protocol) protocol in MobaXterm to perform the high-load training process. The images used for training were labelled with detection targets through LabelImg software.

First, the images were manually inspected to verify the presence of the target defects and to remove any irrelevant or low-quality samples. Next, data augmentation techniques, such as random cropping, flipping, and rotation, were employed to increase the diversity and size of the dataset, ultimately improving the robustness and generalization ability of the trained models.

The dataset was then annotated using the LabelImg tool, with each defect instance labeled as either "defect_1" (erosion-caused holes) or "defect_2" (scratches). There are 408 images with "defect_2" annotation and 859 images with "defect_1" annotation, including 56 images with both "defect_1" and "defect_2" annotation. The annotated images were split into training and validation sets, with a ratio of 80:20, ensuring a balanced representation of both defect types in each set. Fig. 1 below displays some of the selected image datasets.

Figure 1 Partial image dataset

3.2 YOLOv5s Architecture and Hyperparameters

YOLOv5s, the smallest and fastest variant of the YOLOv5 family, was chosen as the primary model for this study due to its excellent balance between speed and accuracy. The architecture of YOLOv5s consists of a backbone network for feature extraction, a neck network for feature aggregation, and a head network for predicting bounding boxes and class probabilities.

The backbone network is based on the Cross Stage Partial Network (CSP) architecture, which efficiently integrates feature maps from different scales to capture both local and global information. The neck network employs a Path Aggregation Network (PANet) to further enhance the multi-scale feature representation. The head network uses anchor boxes and predicts the objectness score, class probabilities, and bounding box coordinates for each detected object. Fig. 2 below shows the model structure diagram of YOLOv5s.

The key hyperparameters of the YOLOv5s model were carefully tuned to optimize its performance on the railway track defect dataset. The input image size was set to 640 × 640 pixels, ensuring a good balance between resolution and computational efficiency. The model was trained for 300 epochs with a batch size of 16, using the Adam optimizer with an initial learning rate of 0.01 and a weight decay of 0.0005. Data augmentation techniques, such as mosaic and mixup, were applied during training to improve the model's robustness and generalization ability.

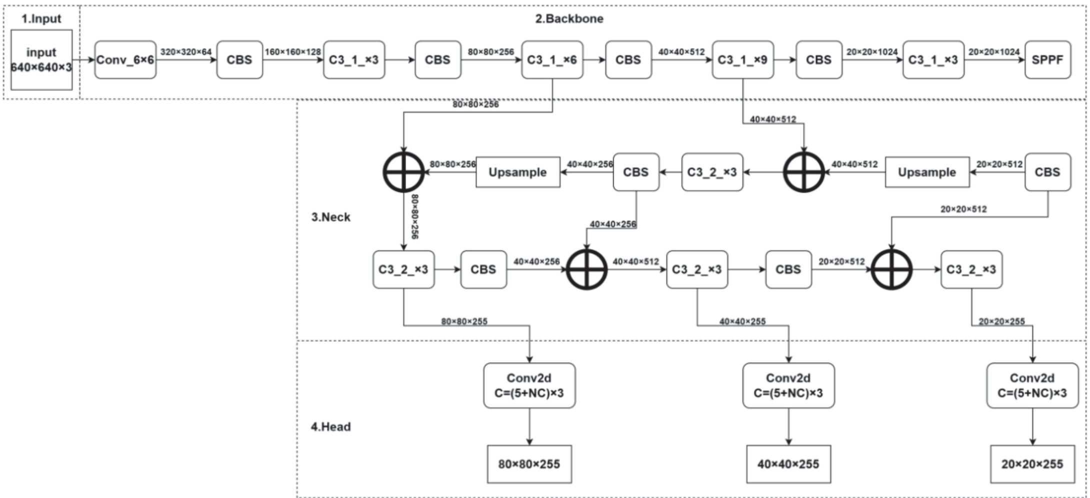


Figure 2 Structure of the model of YOLOv5s

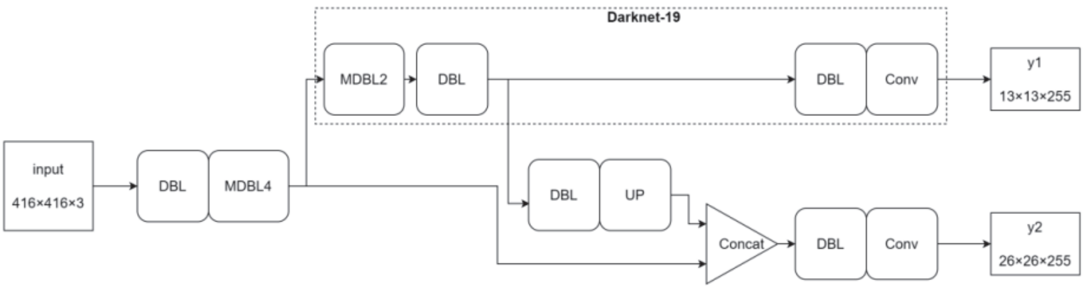


Figure 3 Structure of the model of YOLOv3-Tiny

3.3 YOLOv3-Tiny Architecture and Hyperparameters

To provide a baseline comparison, the YOLOv3-Tiny model was also trained and evaluated on the same dataset. YOLOv3-Tiny is a lightweight variant of the YOLOv3 architecture, designed for real-time object detection on resource-constrained devices. It consists of a smaller backbone network and fewer convolutional layers compared to the full YOLOv3 model, trading off some accuracy for improved speed. Fig. 3 below shows its model structure.

The YOLOv3-Tiny model was trained using similar hyperparameters as YOLOv5s, with an input image size of 416 × 416 pixels, a batch size of 32, and a total of 200 epochs. The initial learning rate was set to 0.001 and decreased by a factor of 10 at epochs 100 and 150. Data augmentation techniques were also applied during training to enhance the model's performance.

3.4 Evaluation Metrics

The performance of the trained YOLOv5s and YOLOv3-Tiny models was evaluated using several standard object detection metrics, including P(Precision), R(Recall) and mAP (Mean Average Precision). P(Precision) measures the percentage of true positive predictions among all positive predictions, while R(Recall) measures the percentage of true positive predictions among all actual positive instances. The mAP is a comprehensive metric that calculates the average precision over all classes and IoU (Intersection over Union) thresholds, providing a single value to compare the overall performance of different models. The other metrics included in the subsequent equations are TP (True Positive, the prediction is a positive sample, the target is measured and this target does exist.), FP (False Positive, the prediction is a positive sample, the target is measured but in reality this target does not exist.) and FN (False Negative, the prediction is a negative sample, no target is detected but the target is in fact present) while AP is Average Precision, whose value is the area covered by the P-R plot. The formula shows that C represents the number of detected classes (Classes) in the project, which is taken as 2 since it contains two types of defects:

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

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

$$mAP = \frac{\sum_{c}^{C} AP(c)}{C} \tag{3}$$

In addition to these metrics, the inference speed and model size were also considered to assess the practicality and efficiency of the models for real-world deployment in railway track inspection systems. The models were tested on the PC with NVIDIA RTX 3070 GPU to measure their inference time per image and to evaluate their potential for real-time detection.

3.5 Testing Process

The process is illustrated in Figure 4. We extracted representative images and classified them into three types: those with only defect_1, those with only defect_2, and those with both defect_1 and defect_2. These images were exported and saved for later comparison and verification.

During the training of the YOLOv3-Tiny-based model, the image dataset and the original darknet project file are transferred to the terminal via SSH. The dataset is then converted and sliced according to the type and number of detected targets using the voc2Yolo.py utility file. The resulting YOLO dataset is stored in the folder labels file, which contains the object_train.txt training set and the object_val.txt validation set. Configure the parameters based on the dataset's location. Generate the data file train_anchors.txt for anchor clustering and run kmeans.py to generate the data file Yolo_anchors.txt for anchors. Obtain the anchors value and modify the convolution number and anchors value in the configuration file Yolov3-tiny-rail. Modify the convolution number and anchors value in the configuration file Yolov3-tiny-rail.cfg based on the number of detected targets and anchors value. Finally, the model is trained using the GPU in accordance with the instructions of the darknet framework for track surface defect detection model training. The trained model file Yolov3-tiny-rail_final Weights is obtained, along with the relevant performance parameters of the model.

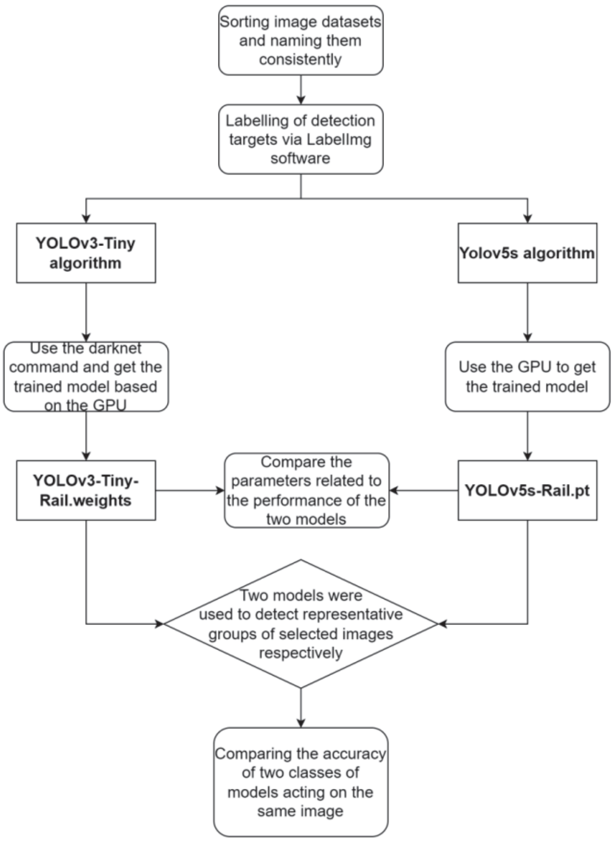


Figure 4 General flowchart of the testing process

During YOLOv5s-based model training, the initial step is identical to the training process of YOLOv3-Tiny. The dataset of images and original project files are transferred to the terminal via SSH. The dataset is then transformed and sliced using the voc2Yolo tool. This generates the YOLO dataset in the current folder, which includes label files, object_train.txt training set, and object_val.txt validation set. Compared to YOLOv3-Tiny, the parameter configuration process for YOLOv5s is much simpler. It only needs to be modified based on the number of detection targets. The conda environment is then activated, and batch size and epochs are adjusted according to memory and other conditions. Finally, train.py is run for training, resulting in the trained model YOLOv5s-rail.pt .

4 RESULTS AND DISCUSSION

4.1 Comparison of YOLOv5s and YOLOv3-Tiny

The trained YOLOv5s and YOLOv3-Tiny models were evaluated on the validation set to compare their performance in detecting and classifying erosion-caused holes (defect_1) and scratches (defect_2) on railway track surfaces. Tab. 1 presents the overall performance metrics for both models.

Table 1 Comparison of results of models trained by YOLOv3-Tiny and YOLOv5s

Model	Precision	Recall	mAP@ 0.50	AverageInference Time / ms	Model Size / MB
YOLOv5s	0.929	0.843	0.886	4.4	14.5
YOLOv3-Tiny	0.794	0.678	0.749	1.8	33.1

The results demonstrate that YOLOv5s outperforms YOLOv3-Tiny across all evaluation metrics. YOLOv5s achieves a higher precision (0.929), recall (0.843), and mAP (0.886) compared to YOLOv3-Tiny, indicating its superior ability to accurately detect and classify both types of defects. The improved performance of YOLOv5s can be attributed to its more advanced architecture, which incorporates features such as the CSP backbone and PANet neck, enabling better feature extraction and multi-scale representation.

However, YOLOv3-Tiny exhibits a faster average inference time (1.8 ms per image) compared to YOLOv5s (4.4ms per image). This highlights the trade-off between accuracy and efficiency, with YOLOv3-Tiny being more suitable for resource-constrained environments where real-time performance is critical.

4.2 Qualitative Analysis

To gain further insights into the models' performance, we visualized the detection results on a sample of images from the validation set. Fig. 5 shows a comparison of the detection outputs from YOLOv5s and YOLOv3-Tiny on six representative images containing both defect types.

Figure 5 Comparison of the detection outputs from YOLOv5s and YOLOv3-Tiny on both defect types

The qualitative analysis reveals that YOLOv5s produces more accurate and precise bounding boxes compared to YOLOv3-Tiny. YOLOv5s successfully detects most instances of both defect types, with minimal false positives and false negatives. In contrast, YOLOv3-Tiny struggles with smaller and less prominent defects, resulting in more missed detections and misclassifications.

The analysis reveals that both models perform better in detecting and classifying scratches (defect_2) compared to erosion-caused holes (defect_1). This can be attributed to the more distinct and consistent appearance of scratches in the dataset, making them easier to learn and detect.

However, both models face challenges in detecting erosion-caused holes (defect_1) when they are small, irregular, or located in poorly illuminated regions of the image. This highlights the need for further improvement in the robustness of the models to handle diverse and challenging real-world scenarios.

4.3 The Analysis of the Final Training Results of the YOLOv5s Model

This chapter focuses on the analysis of the performance of the model obtained based on the YOLOv5s algorithm, which can further demonstrate the rationality and reliability of using this model in this field. When the number of epochs is set to 300 during the training process, Precision(P) , Recall(R), mAP@0.5:0.95 (the average mAP value for IoU thresholds from 0.5 to 0.95 in steps of 0.05) and mAP@0.5 (the average mAP value for thresholds greater than 0.5) are finally obtained in the output result file. mAP), the dotted line graph of the output results is shown in Fig. 6 below. The figure illustrates that while there are significant fluctuations in precision and recall values for small epoch values, they stabilize and increase as the epoch value reaches 200.

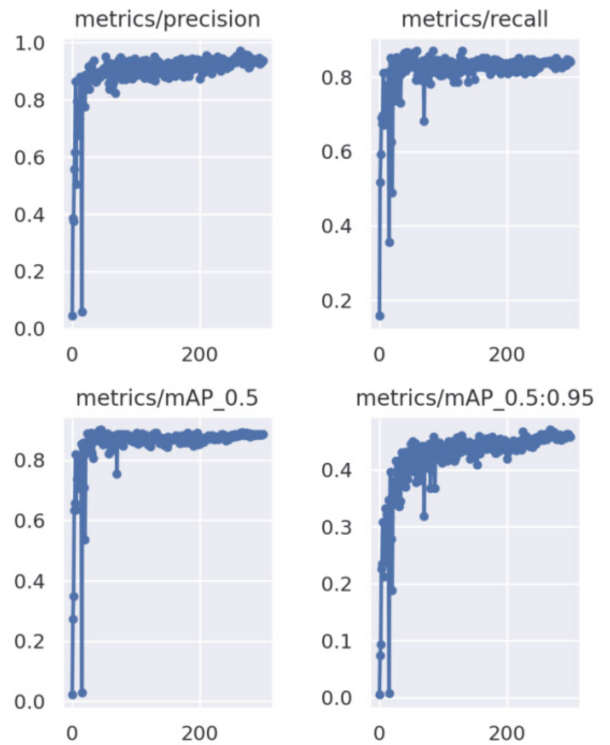


Figure 6 Plot of performance parameters of the training output of YOLOv5s model

Other plots showing the model performance are also included in the output results. For instance, Fig. 7 shows the P-R graph, which describes the relationship between precision and recall. In this category, higher precision typically results in lower recall, but it is crucial to ensure that the curve achieves the highest possible recall while maintaining precision. The AP value mentioned above represents the percentage of the coverage area under the curve in the total area. The closer the AP value is to 1, the better the effect. In this project, the features of defect_2 are more uniform, resulting in an overall AP value closer to 1. However, the overall mAP value can reach 0.886, indicating that the model's overall detection performance is relatively good.

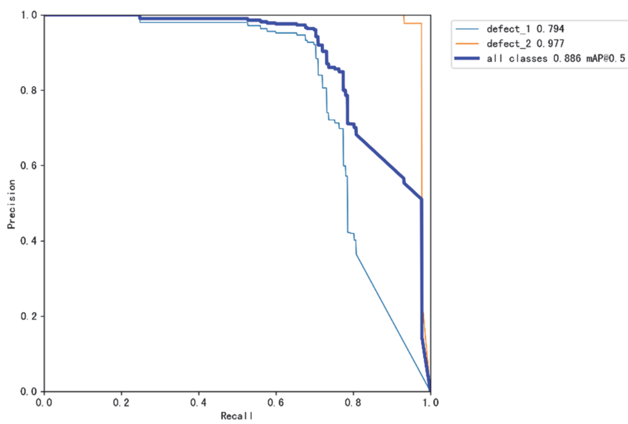


Figure 7 Precision-recall plot of training output of YOLOv5s model

Fig. 8 illustrates the relationship between precision and confidence level. It shows the precision of each detection target category at a determined confidence value of 0.953. Both categories of defects that need to be detected reach 100% precision at this threshold. The value of 0.953 indicates that the model operation results are normal and reliable when both precision and confidence are high. This suggests that the model's detection precision is more dependable when the detection data source is reliable and has a high level of confidence [7].

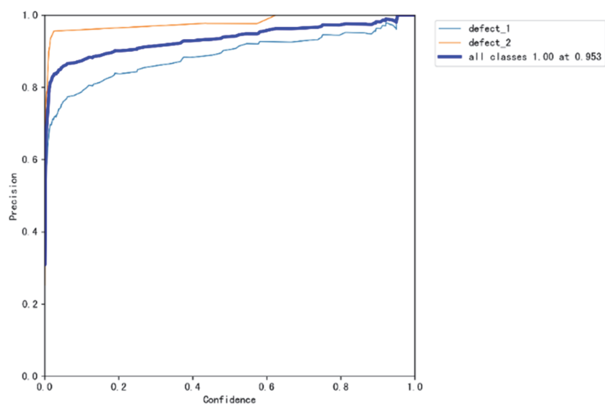


Figure 8 Plot of precision and confidence relationship of YOLOv5s model training outputs

Fig. 9 below illustrates the relationship between recall and confidence, which is then used to measure the precision performance of the model at different recall rates. The figure illustrates that the overall recall for detecting both types of defects reaches 0.89 at a confidence threshold of 0. It is evident that the proportion of True Positive (TP) among the model's detections can reach 89% when the confidence threshold is 0. Recall-confidence curves are useful for evaluating the precision of the model's detections when the certainty of the detection target varies. They also help to balance precision and recall to reflect different levels of confidence [7].

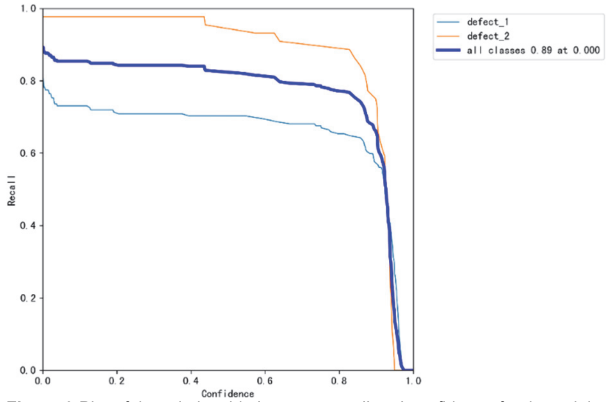


Figure 9 Plot of the relationship between recall and confidence for the training output of the YOLOv5s model

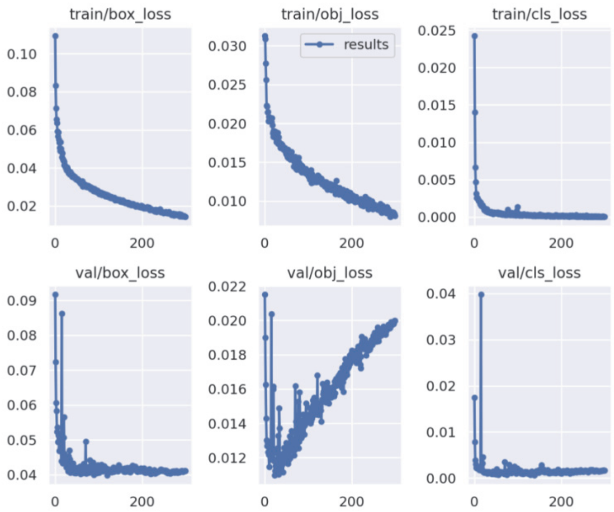
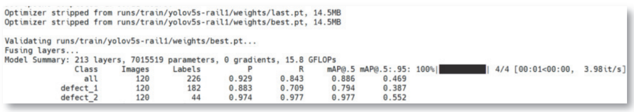


Figure 10 Dotted line plots of YOLOv5s model training outputs for each type of loss



```
Optimizer stripped from runs/train/yolov5s-rail1/weights/last.pt, 14.5MB
Optimizer stripped from runs/train/yolov5s-rail1/weights/best.pt, 14.5MB

Validating runs/train/yolov5s-rail1/weights/best.pt...
Fusing layers...
Model Summary: 213 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100%|██████████| 4/4 [00:01<00:00,  3.98it/s]
                 all        120        226      0.929      0.843      0.886      0.469
            defect_1        120        182      0.883      0.709      0.794      0.387
            defect_2        120         44      0.974      0.977      0.977      0.552
```

Figure 11 Final result plot of YOLOv5s model training outputs

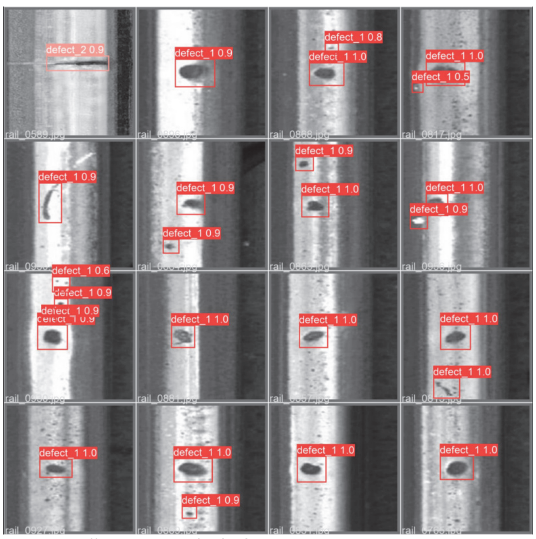


Figure 12 Effectiveness of YOLOv5s model tested on sampled images

As the number of epochs increases, the different types of losses converge to a stable and small value. This indicates that the final training effect is more reliable, as shown in Fig. 10. This indicates that the model is reliable for detecting the two specific types of defects. The final values for each category at 300 epochs are displayed in Fig. 11. In the output folder, Fig. 12 displays the model's ability to detect the sampled images.

5 CONCLUSION AND FUTURE WORK

In this study, we proposed an approach for detecting and classifying two common types of railway track surface defects, namely erosion-caused holes and scratches, using the YOLOv5s algorithm. The YOLOv5s model, based on convolutional neural networks and computer vision techniques, demonstrated superior performance compared to the YOLOv3-Tiny model in terms of precision, recall, and mean average precision (mAP). The experimental results, conducted on a custom dataset of 1211 annotated track images, validated the effectiveness and efficiency of the proposed approach.

The main contributions of this study lie in the successful adaptation and fine-tuning of the YOLOv5s architecture for the specific task of railway track defect detection. By leveraging the optimized architecture and advanced features of YOLOv5s, such as the CSP backbone and PANet neck, the proposed approach achieved a precision of 0.929, a recall of 0.843, and the mAP of 0.886, surpassing the performance of the lighter YOLOv3-Tiny model. Furthermore, the creation of a comprehensive dataset, containing labeled instances of erosion-caused holes and scratches, provides a valuable resource for future research and benchmarking in this domain.

The qualitative analysis of the detection results revealed the strengths and limitations of the YOLOv5s model in handling various defect types and scenarios. While the model demonstrated robust performance in detecting and classifying scratches, it faced challenges in accurately localizing small and irregular erosion-caused holes, particularly in poorly illuminated regions. This highlights the need for further improvements in the model's ability to handle diverse and complex real-world track conditions.

The practical implications of this study extend to the development of efficient and reliable railway track inspection systems. By automating the process of defect detection and classification, the proposed approach has the potential to significantly reduce the time and labor costs associated with manual inspection methods. The real-time inference capabilities of YOLOv5s, with an average processing time of 4.4ms per image, make it suitable for integration into existing track monitoring frameworks. However, the deployment of such systems in real-world scenarios requires careful consideration of factors such as model size, computational resources, and data privacy.

To address the limitations of the current study and further enhance the performance and applicability of the proposed approach, several directions for future research are identified. First, expanding the dataset to include a wider range of defect types, such as cracks, deformations, and corrosion, would enable the development of a more comprehensive and robust defect detection system. Second, incorporating additional data augmentation techniques, such as generative adversarial networks (GANs) or style transfer, could help improve the model's ability to handle variations in lighting, weather, and track conditions. Third, exploring the integration of multiple sensing modalities, such as 3D laser scanning or ultrasonic testing, could provide complementary information and enhance the overall defect detection performance.

Furthermore, future work could focus on optimizing the model architecture and training process to reduce computational requirements and improve real-time performance. Techniques such as model compression, pruning, and quantization could be investigated to develop lightweight and efficient variants of the YOLOv5s model, suitable for deployment on resource-constrained devices. Additionally, the incorporation of domain-specific knowledge, such as the spatial and temporal relationships between different defect types, could help refine the detection and classification results.

In conclusion, this study demonstrates the potential of the YOLOv5s algorithm for automated railway track surface defect detection and classification. The proposed approach, validated through extensive experiments and analysis, offers a promising solution for enhancing the efficiency and reliability of track inspection processes. By addressing the limitations and exploring the identified future research directions, this work paves the way for the development of intelligent and robust track monitoring systems, ultimately contributing to the safety and sustainability of railway transportation networks.

6 REFERENCES

[1] Yue, G., Cui, X., Zhang, K., Wang, Z., & An, D. (2020). Guided Wave Propagation for Monitoring the Rail Base. *Mathematical Problems in Engineering*, *2020*, 4756574. https://doi.org/10.1155/2020/4756574

[2] Ye, J., Stewart, E., Zhang, D., Chen, Q., & Roberts, C. (2020). Method for automatic railway track surface defect classification and evaluation using a laser-based 3D model. *Iet Image Processing*, *14*(12), 2701-2710. https://doi.org/10.1049/iet-ipr.2019.1616

[3] Ren, Y., Lu, P., Ai, C., Gao, L., Qiu, S., & Tolliver, D. (2021). Review of emerging technologies and issues in rail and track inspection for local lines in the United States. *Journal of Transportation Engineering, Part A: Systems*, *147*(10), 04021062. https://doi.org/10.1061/JTEPBS.0000567

[4] Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020). Computer vision technology in agricultural automation - A review. *Information Processing in Agriculture*, *7*(1), 1-19. https://doi.org/10.1016/j.inpa.2019.09.006

[5] Zhang, H., Song, Y., Chen, Y., Zhong, H., Liu, L., Wang, Y., ... & Wu, Q. J. (2021). MRSDI-CNN: Multi-model rail surface defect inspection system based on convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 11162-11177. https://doi.org/10.1109/TITS.2021.3101053

[6] Kumar, A. (2008). Computer-vision-based fabric defect detection: A survey. *IEEE transactions on industrial electronics*, *55*(1), 348-363. https://doi.org/10.1109/TIE.1930.896476

[7] Dang, F., Chen, D., Lu, Y., & Li, Z. (2023). YOLO Weeds: a novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems. *Computers and Electronics in Agriculture*, *205*, 107655. https://doi.org/10.1016/j.compag.2023.107655

[8] Sresakoolchai, J. & Kaewunruen, S. (2022). Railway defect detection based on track geometry using supervised and unsupervised machine learning. *Structural Health Monitoring*, *21*(4), 1757-1767. https://doi.org/10.1177/14759217211044492

[9] Bao, Y., Tang, Z., Li, H., & Zhang, Y. (2019). Computer vision and deep learning-based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, *18*(2), 401-421. https://doi.org/10.1177/1475921718757405

[10] Chen, Z., Wang, Q., He, Q., Yu, T., Zhang, M., & Wang, P. (2022). CUFuse: Camera and ultrasound data fusion for rail defect detection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(11), 21971-21983. https://doi.org/10.1109/TITS.2022.3189677

[11] Ni, Y. Q. & Zhang, Q. H. (2021). A Bayesian machine learning approach for online detection of railway wheel defects using track-side monitoring. *Structural Health Monitoring*, *20*(4), 1536-1550. https://doi.org/10.1177/1475921720921772

[12] Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., & Li, Y. (2019). Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Engineering Applications of Artificial Intelligence*, *80*, 66-81. https://doi.org/10.1016/j.engappai.2019.01.008

[13] Ni, X., Liu, H., Ma, Z., Wang, C., & Liu, J. (2021). Detection for rail surface defects via partitioned edge feature. *IEEE Transactions on Intelligent Transportation Systems*, *23*(6), 5806-5822. https://doi.org/10.1109/TITS.2021.3058635

[14] Pincott, J., Tien, P. W., Wei, S., & Calautit, J. K. (2022). Indoor fire detection utilizing computer vision-based strategies. *Journal of Building Engineering*, *61*, 105154. https://doi.org/10.1016/j.jobe.2022.105154

[15] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, *77*, 354-377. https://doi.org/10.1016/j.patcog.2017.10.013

[16] Cantero-Chinchilla, S., Wilcox, P. D., & Croxford, A. J. (2022). Deep learning in automated ultrasonic NDE-developments, axioms and opportunities. *NDT & E International*, *131*, 102703. https://doi.org/10.1016/j.ndteint.2022.102703

[17] Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R., & De Schutter, B. (2016, July). Deep convolutional neural networks for detection of rail surface defects. *2016 International joint conference on neural networks (IJCNN)*, 2584-2589. https://doi.org/10.1109/IJCNN.2016.7727522

[18] Shang, H., Sun, C., Liu, J., Chen, X., & Yan, R. (2022). Deep learning-based borescope image processing for aero-engine blade in-situ damage detection. *Aerospace Science and Technology*, *123*, 107473. https://doi.org/10.1016/j.ast.2022.107473

[19] Chen, S. X., Zhou, L., Ni, Y. Q., & Liu, X. Z. (2021). An acoustic-homologous transfer learning approach for acoustic emission-based rail condition evaluation. *Structural Health Monitoring*, *20*(4), 2161-2181. https://doi.org/10.1177/1475921720976941

[20] Saidani, T. (2023). Deep Learning Approach: YOLOv5-based Custom Object Detection. *Engineering, Technology & Applied Science Research*, *13*(6), 12158-12163. https://doi.org/10.48084/etasr.6397

[21] Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, *396*, 39-64. https://doi.org/10.1016/j.neucom.2020.01.085

[22] Lee, J. & Hwang, K. I. (2022). YOLO with adaptive frame control for real-time object detection applications. *Multimedia Tools and Applications*, *81*(25), 36375-36396. https://doi.org/10.1007/s11042-021-11480-0

[23] Redmon, J. & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804*.02767.

https://doi.org/10.48550/arXiv.1804.02767

[24] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.*10934. https://doi.org/10.48550/arXiv.2004.10934

[25] Zhou, S., Zeng, Y., Li, S., Zhu, H., Liu, X., & Zhang, X. (2022). Surface defect detection of rolled steel based on lightweight model. *Applied Sciences*, *12*(17), 8905. https://doi.org/10.3390/app12178905

[26] Cui, X., Wang, Q., Dai, J., Zhang, R., & Li, S. (2021). Intelligent recognition of erosion damage to concrete based on improved YOLO-v3. *Materials Letters*, *302*, 130363. https://doi.org/10.1016/j.matlet.2021.130363

[27] Cui, J., Qin, Y., Wu, Y., Shao, C., & Yang, H. (2023). Skip Connection YOLO Architecture for Noise Barrier Defect Detection Using UAV-Based Images in High-Speed Railway. *IEEE Transactions on Intelligent Transportation Systems*, *24*(11), 12180-12195. https://doi.org/10.1109/TITS.2023.3292934

[28] Ragab, M. G., Abdulkader, S. J., Muneer, A., Alqushaibi, A., Sumiea, E. H., Qureshi, R., ... &Alhussian, H. (2024). A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023). *IEEE Access*, *12*, 57815-57836. https://doi.org/10.1109/ACCESS.2024.3386826

[29] Rajamohanan, R. & Latha, B. C. (2023). An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset. *Engineering, Technology & Applied Science Research*, *13*(6), 12033-12038. https://doi.org/10.48084/etasr.6377

[30] Gao, G., Wang, S., Shuai, C., Zhang, Z., Zhang, S., & Feng, Y. (2022). Recognition and Detection of Greenhouse Tomatoes in Complex Environment. *Traitement du Signal, 39*(1), 291-298. https://doi.org/10.18280/ts.390130

[31] Ren, F., Fei, J., Li, H., & Doma, B. T. (2024). Steel Surface Defect Detection Using Improved Deep Learning Algorithm: ECA-SimSPPF-SIoU-Yolov5. *IEEE Access*, *12*, 32545-32553. https://doi.org/10.1109/ACCESS.2024.3371584

[32] Tulbure, A. A., Tulbure, A. A., & Dulf, E. H. (2022). A review on modern defect detection models using DCNNs–Deep convolutional neural networks. *Journal of Advanced Research*, *35*, 33-48. https://doi.org/10.1016/j.jare.2021.03.015

Contact information:

Congbo YANG
Centre of Advanced Communication Research and Innovation (ACRI),
Department of Electrical Engineering, Faculty of Engineering,
University of Malaya,
Kuala Lumpur 50603, Malaysia
E-mail: s2191786@siswa.um.edu.my

Kaharudin DIMYATI, PhD, Professor
(Corresponding author)
Centre of Advanced Communication Research and Innovation (ACRI),
Department of Electrical Engineering, Faculty of Engineering,
University of Malaya,
Kuala Lumpur 50603, Malaysia
E-mail: kaharudin@um.edu.my

Mahazani MOHAMAD, MEng
Centre of Advanced Communication Research and Innovation (ACRI),
Department of Electrical Engineering, Faculty of Engineering,
University of Malaya,
Kuala Lumpur 50603, Malaysia
E-mail: mahazani@um.edu.my