Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

Enhancing Rail Surface Defect Detection Using a Hybrid YOLOv11–Vision Transformer Framework

Baburao Markapudi[1], Lahari Gundiga[2], Venkata Ramesh Babu Kondeti[3], Sravanthi Kanumuri[4], Yamini Maddala[5]

[1]Professor and Head, [2,3,4,5]Student, Department of Computer Science and Engineering, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India.

E-mail: [1]baburaompd@gmail.com, [2]laharigundiga@gmail.com, [3]rameshkondeti11@gmail.com, [4]sravanthiknmr@gmail.com, [5]maddalayamini748@gmail.com

Abstract

Railway track surface defects cause severe challenges to the safety of humans, trains, and transported goods. Modern YOLO-based detectors efficiently detect rail surface defects in real time. However, they mainly focus on local features and also confuse defects that look similar. On the other hand, Vision Transformers are good at modeling global context using self-attention, but they do not localize objects properly when used alone. To overcome these issues, this study introduces a hybrid YOLOv11 and Vision Transformer (YOLOv11a and ViT) framework for enhanced rail surface defect detection. The approach integrates a lightweight Vision Transformer module into the YOLOv11 backbone so that the model learns both detailed local features and captures global dependencies. Experiments are conducted on a merged public railway surface defect computer vision dataset comprising 8,177 labeled images of five defect categories. The results show that the new method achieves an mAP@0.5 of 0.951 with a precision of 0.899 and recall of 0.921, outperforming the baseline YOLO model. The most significant advancements have been made in identifying elongated defects such as cracks and light bands.The framework also maintains real-time performance, making it practical for use in railway safety inspections.

Keywords: Rail Surface Defect Detection; YOLOv11; Vision Transformer; Hybrid Detection Framework; Global Context Modelling.

1. Introduction

Railways are one of the most reliable and cost-effective ways of transporting people and goods across the globe. However, the reliability of a railway system depends on the state of the tracks. Surface defects can weaken the tracks and may cause derailments and costly repairs. Cracks, breaks, scars, and light bands are some of the problems that may develop over a long period and may not be easy to identify at an early stage. Thus, it is very important to identify these defects at an early.

In general, traditional rail inspection techniques mostly rely on rule-based image processing techniques or human vision. Manual inspection is a time-consuming and tedious task, and it is also associated with human error. Therefore, it is not applicable for the continuous

DOI: https://doi.org/10.36548/jtcsst.2026.2.005
Received: 23.02.2026, received in revised form: 26.03.2026, accepted: 12.04.2026, published: 24.04.2026

inspection of railway networks. Similarly, traditional image processing methods mainly depend on manually designed features and fixed threshold values. Therefore, they are not applicable in many situations, as various factors can affect the performance of the image processing techniques, such as changes in illumination, surface conditions, noise, and complex backgrounds. Therefore, data-driven computer vision techniques for automated rail surface defect detection are becoming highly popular.

Recent progress in deep learning has significantly improved the efficiency of automated inspection across various industries. Object detection models based on convolutional neural networks, particularly single-stage detectors like YOLO, are commonly adopted, as they can perform detection and classification of objects simultaneously. Variants of the model, including YOLOv8 and YOLOv11, are popular choices, as they are both efficient in terms of speed and accurate in results, making them suitable for real-time applications. The model's performance is further enhanced in YOLOv11, making it efficient in detecting rail surface defects.

Although such detectors are effective, they are based mainly on local features. This implies a limitation in the model's ability to detect certain defects such as cracks and light bands. In such situations, depending only on local features can lead to fragmented detections or lower sensitivity to slight structural patterns. This, in turn, affects the complete performance of the system.

Recently, Vision Transformers (ViTs) have emerged as a newer approach designed for learning visual representation by employing attention-based mechanisms to capture global contextual relationships across an image. By directly capturing long-range dependencies, the transformer provides enhanced global reasoning capability, which is better suited for the identification of extended and structurally continuous patterns. However, they are inadequate for providing precise object location in dense detection tasks when used alone.

To address with the corresponding shortcomings of CNNs and transformers, hybrid approaches that combine CNN and transformer-based architectures are being increasingly explored in recent years. Inspired by this pathway, this paper proposes a hybrid YOLOv11–Vision Transformer (YOLOv11 and ViT) framework for enhanced rail surface defect detection. The experimental findings demonstrate that the suggested approach is superior to the benchmark, namely the YOLOv11, when considering elongated and minor visual defects on the rail surface.

2. Literature Survey

Detection of defects on rail surfaces plays an important role in enhancing the security and quality of the railway transport system. The manual defect detection process suffers from human errors due to its manual nature, which made researchers to develop automated systems for inspection. Automated inspection methods offer more accurate results compared to traditional techniques [1], [2].

The initial works in this field concentrated on the use of manually crafted descriptors to represent the image content. The descriptors used included color and structural characteristics of the image content. Content-based image retrieval methods made use of dominant color, co-occurrence matrix, and structural features in classification applications. However, the performance of these methods was not satisfactory in complex environments [1], [2].

Due to the development of deep learning, CNNs have been widely adopted for analyzing images because they can capture hierarchical representations of features. The deep residual learning approach was also developed, allowing for deeper networks to be trained, making it possible to extract more efficient features [3]. Moreover, the introduction of the FPN technique allowed for the use of multi-scale feature representation, leading to improvements in object detection for differently sized objects [4].

The invention of YOLO (You Only Look Once) models was a major step forward in object detection, as it made localization and classification one-step processes for more efficient object detection [5]. The evolution of YOLO model algorithms, including YOLOv5 models, has shown that further improvement in their performance is possible [6].

Several recent works have used the YOLO algorithm for detecting rail surface defects. Enhanced YOLOv8 algorithms provide a more advanced feature extraction process and fusion method, which increases the detection efficiency of rail surface defects in complicated railway scenarios [7]. Additionally, light-weight YOLOv8n-RSDD models have been designed to optimize computation while ensuring accurate detection [8]. Precision-oriented systems have also been constructed for identifying small rail surface defects under difficult railway conditions [9].

Besides object detection algorithms, many research works have also proposed intelligent learning schemes that include optimization based neural networks, reinforcement learning, and fused classification techniques, which exhibit good robustness and classification accuracy [10]–[12].

Despite the above-mentioned advancements, CNNs can only make use of local features because of their small receptive field size. Consequently, they cannot handle long-term dependencies, which are critical for defect detection, especially crack detection and other continuous surface defects.

By utilizing self-attention models, vision transformers (ViTs) overcome this drawback by identifying global context dependencies among images, thereby promoting interactions among distant image regions [13]. The Swin Transformer takes advantage of this method by implementing hierarchical representation learning alongside window shifting and attention techniques, making the process computationally more efficient [14].

In addition, the accessibility of annotated data allows for the creation of accurate detection models. Open datasets, such as those provided by Roboflow, offer a labeled set of images used in the training and testing process for rail surface defect detection [15].

In light of these advancements, hybrid models that leverage the CNN architecture for feature extraction and transformer networks for context modeling have emerged. These models utilize both local and global feature representations, thus achieving better defect detection results. With regard to this research, we incorporate a transformer encoder layer in the YOLOv11 model architecture.

3. Proposed Work

This section describes the proposed method for detecting rail surface defects using a hybrid YOLOv11-Vision Transformer model. This method incorporates real-time detection of objects along with global context. This helps in detecting both small and long defects. The

steps in the proposed method involve preparing the dataset, feature extraction using the YOLOv11 model, global context detection using the transformer module, feature combination, and finally, detection.

3.1 Dataset Description and Preprocessing

To check the efficiency of the proposed framework, a rail surface defect dataset is used. This dataset is obtained by combining different versions of a publicly available dataset from Roboflow [15]. Since the versions of the dataset were from the same stage of data collection, they are merged without any inconsistency. This also helps in diversifying the dataset.

The dataset includes five types of defects: cracks, rails, scars, breaks, and lightbands, covering both small localized defects and longer structural irregularities.

These defect classes are determined based on their unique visual features as well as structural features present in the rail image. Cracks can be identified as thin lines on the rail surface, whereas rail features can be identified as the rail head with uneven wear or deformation. Scars can be identified as unique marks on the rail surface due to friction over a long period of time, while breaks can be identified as damaged regions on the rail surface. Lightbands can be identified as unique long regions present on the rail image due to the presence of the rail surface or lighting conditions. The YOLOv11 detection head can be trained to identify the unique features of the defects present on the rail image as bounding box regions, whereas the unique features of the defects can be identified with the help of the added Vision Transformer.

These images can be captured under real-world conditions with varying lighting, texture of the rail surface, viewing angle, and complexity of the background image.

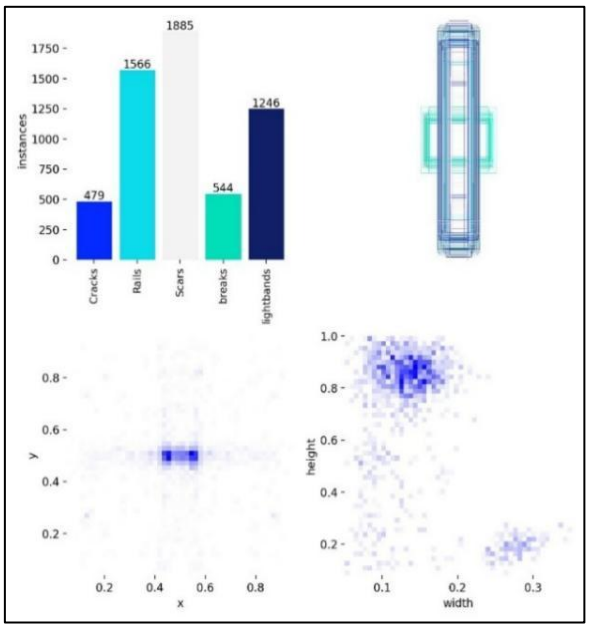

Figure 1. Statistical Visualization of the Merged Rail Surface Defect Dataset

The distribution of the defect classes in the training, validation, and test sets is depicted in Table 1. Random shuffling of the dataset is carried out, dividing it into the training, validation, and test sets in the ratio of 70:20:10 for the purpose of fair evaluation of the model's performance. A statistical summary of the merged dataset is shown in Figure 1.

Prior to the training process, the images are resized to 640 x 640 pixels. Normalization of the images is performed in the range of 0 to 1. Minimal data augmentation is applied so that the effect of the design of the model on the detection capability can be observed.

Table 1. Class-wise Distribution of Rail Surface Defect Instances

Defect Class	Train	Validation	Test	Total
Cracks	479	158	55	692
Rails	1566	449	263	2278
Scars	1885	529	222	2636
Breaks	544	164	76	784
Lightbands	1246	361	180	1787
Total	5720	1661	796	8177

3.2 Overall Framework Architecture

The proposed rail surface defect detection model is based on the YOLOv11 model and the lightweight Vision Transformer. The proposed model aims to improve overall context understanding and defect detection performance. In the proposed model, the YOLOv11 model is used to extract local features and detect rail surface defects. The overall proposed process for rail surface defect detection consists of four steps: input preprocessing, feature extraction, global context modeling, and multi-scale feature fusion for final defect detection. The overall proposed framework for rail surface defect detection is presented in Figure 2.

3.3 YOLOv11 Backbone for Feature Extraction

YOLOv11 is used as a backbone network for its high inference speed and accuracy in object detection. The earlier convolutional layers focus on basic visual features such as edges, textures, and surfaces, while deeper layers focus on higher-level features such as abstract representations related to defect class information.

The backbone structure adopts stacked convolutional layers and C3k2 residual blocks to reuse features and ensure stable gradient propagation during training. A Spatial Pyramid Pooling Fast (SPPF) module is included at the end of the backbone to combine the learning of contextual cues provided by different scales of the image. This is especially critical for the application of rail surface inspection, where different defect patterns vary in size, shape, and orientation.

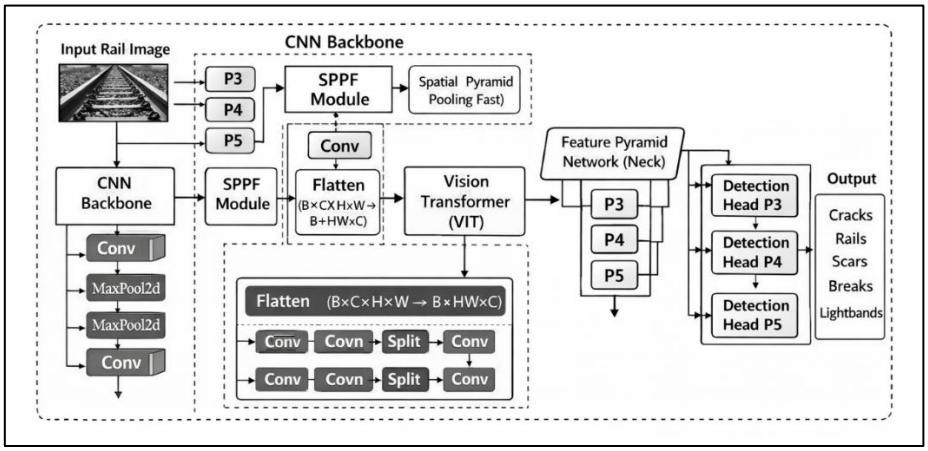

Figure 2. Overall Architecture of the Proposed YOLOv11–Vision Transformer Rail Surface Defect Detection Framework

3.4 Vision Transformer Integration for Global Context Modeling

The Vision Transformer module consists of a lightweight single encoder block inserted after the SPPF layer of the YOLOv11 backbone. Instead of using a full Vision Transformer architecture with patch embedding, the proposed block directly operates on convolutional feature maps.

Given an input feature map of size B×C×H×W, the spatial dimensions are flattened to form a sequence of HW tokens, each with an embedding dimension C. Multi-head self-attention is implemented using PyTorch's nn.MultiheadAttention module. The number of attention heads is dynamically adjusted according to the channel dimension C, defined as:

$$\text{Number of Heads} = \min(8, \max(1, C/64))$$

This ensures computational efficiency and stability across different feature dimensions.

The output from the attention mechanism is fused with the input via a residual connection. This is followed by a feedforward MLP with an expansion ratio of 4.0 and GELU activation. A residual connection is used again after this process. A dropout rate of 0.1 is applied within the attention mechanism for better generalization.

Due to the nature of how the transformer works with CNN feature maps, no additional positional encoding is required. Positional information can be considered implicit through convolutional encoding as well as tensor reshaping operations.

The placement of the transformer block occurs post-SPPF block to facilitate global interactions among enriched multiscale feature representations while reducing computational overhead. Multiple scales of transformer block placement will lead to high computational complexity and memory requirements.

The embedding dimension is directly derived from the backbone feature channel size. Empirical observations indicate that maintaining the native channel dimension preserves representational consistency and avoids additional projection overhead.

The structural organization of the lightweight transformer encoder block integrated into the YOLOv11 backbone is illustrated in Figure 3.

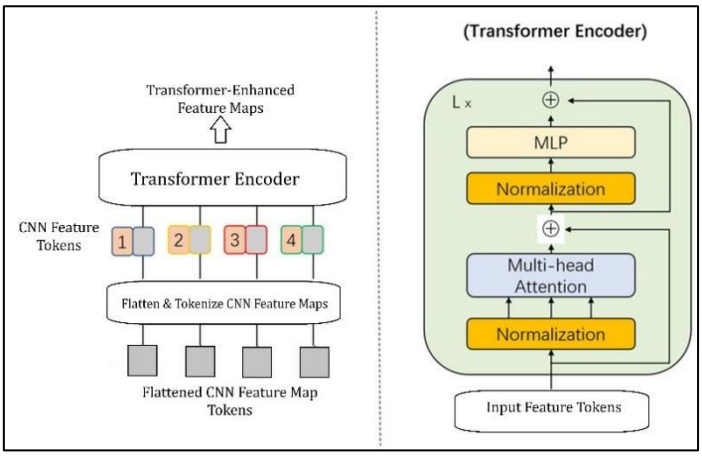

Figure 3. Lightweight Transformer Encoder Block Integrated after the SPPF Module in the YOLOv11 Backbone

3.5 Feature Fusion and Detection Head

Finally, after passing through a transformer-based enhancement network, the feature maps are fed into the YOLOv11 neck for multi-scale feature fusion using upsampling and concatenation operations. This helps to integrate deep semantic features with high-resolution spatial features for the accurate localization of defects across various scales.

The YOLOv11 detection head predicts bounding box coordinates, object confidence scores, and class probabilities for each defect category in a single forward pass. Thus, the detection approach remains the same, enabling the proposed framework to ensure efficiency and real-time performance.

The model distinguishes between different defect classes through the classification branch of the YOLOv11 detection head. After feature extraction and fusion, each predicted bounding box contains rich feature information that represents the local appearance of the defect. These features capture important patterns such as shape, texture, and structural variations present on the rail surface.

This information is further processed by the detection head with convolution operations to give each of these bounding boxes its respective class probability scores. With this learning from the features, the model is now capable of identifying the different classes such as Cracks, Rails, Scars, Breaks, and Lightbands that look similar at times.

Moreover, the addition of the Vision Transformer helps the model differentiate more accurately between classes of defects by learning long-range contextual information from the image. This enables the model to better comprehend elongated defects, reducing confusion between similar classes.

3.6 Training Strategy and Experimental Environment

The proposed YOLOv11–Vision Transformer model is trained using annotated rail surface images. The architecture for the model is defined using a custom YOLOv11 configuration file, and the model is not initialized with any weights for training purposes. All the parameters are trained and optimized together for integrated learning. purpose.

The model is optimized using the AdamW optimizer along with a cosine learning rate schedule. Training starts with an initial learning rate of 3×10^{-4}, and weight decay is applied to improve the model's generalization ability.

To verify sensitivity, some preliminary experiments on different learning rates, such as 1×10^{-4} and 5×10^{-4} have been carried out. It was observed that though there are some differences in the rates of convergence, the value of 3×10^{-4} provides the best optimization stability and detection accuracy.

The standard loss function used is the Ultralytics YOLOv11 loss function. In the standard loss function, bounding box regression, classification, and Distribution Focal Loss (DFL) are used. In the experiment, the default loss weights provided by the Ultralytics framework have been used.

The use of Automatic Mixed Precision (AMP) is necessary to avoid computational overhead while ensuring numerical stability. The training parameters along with the experiment settings can be viewed in Table 2.

Although no aggressive data augmentation strategies were applied, overfitting was mitigated through weight decay regularization, cosine learning rate scheduling, dropout within the transformer block, and validation-based best weight selection. The best-performing weights were selected based on validation metrics, and no significant divergence between training and validation performance was observed.

4. Results and Discussion

This section presents the experimental results obtained using the proposed YOLOv11-Vision Transformer detection framework. Additionally, the experimental results obtained using the conventional YOLOv11 model are compared in this section. The experiments are conducted using the Ultralytics library written in Python on a computer system supporting the NVIDIA CUDA architecture. The details of the experimental setup are provided in Table 2.

Table 2. Training Configuration for YOLOv11 and YOLOv11+ViT Models

Parameter	Value
Framework	Ultralytics YOLOv11
Programming Language	Python
Input Image Size	640 × 640
Optimizer	AdamW
Initial Learning Rate	0.0003
Learning Rate Scheduler	Cosine Annealing
Batch Size	8
Number of Epochs	200
Data Augmentation	None
Automatic Mixed Precision (AMP)	Enabled
Loss Functions	Box Loss, Classification Loss, Distribution Focal Loss
Hardware	NVIDIA Tesla T4 (15 GB)

4.1 Evaluation Metrics and Performance Visualization

The performance of the proposed rail surface defect detection mechanism using a rail surface defect detection framework is measured using commonly adopted object detection metrics.

Precision (P) indicates how accurately the model predicts defects by measuring how many predicted outputs are accurate. It is defined using Eq. (1).

$$Precision = \frac{TP}{(TP + FP)} \tag{1}$$

In this formula, TP refers to the number of correctly identified positive detections, while FP indicates the number of incorrectly classified positive detections.

Recall (R) represents the ability of the model to correctly detect all defect instances present in an image and is computed using Eq. (2).

$$Recall = \frac{TP}{(TP + FN)} \quad (2)$$

In this context, FN denotes the number of false negative detections.

For evaluating both detection accuracy and localization effectiveness, Average Precision (AP) is derived from the precision–recall curve for each defect category. The overall detection performance of the model is summarized using mean Average Precision (mAP), which is computed as shown in Eq. (3).

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (3)$$

where APi denotes the average precision associated with the ith defect class and N represents the total number of defect categories.

Precision and recall are reported at an Intersection-over-Union (IoU) threshold of 0.5. The mAP@0.5–0.95 metric follows the COCO evaluation protocol, averaging performance across IoU thresholds from 0.5 to 0.95 with a step size of 0.05. All evaluation metrics are computed using the default Ultralytics YOLOv11 evaluation pipeline.

The comparative detection outcomes of the YOLOv11 baseline and the proposed YOLOv11+ViT approach are visually presented in Figure 4.

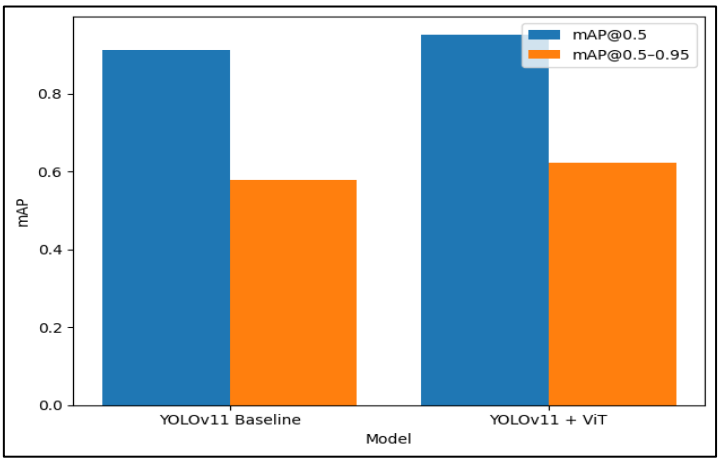

Figure 4. Performance Comparison Between the YOLOv11 Baseline and the Proposed YOLOv11+ViT Model in terms of mAP@0.5 and mAP@0.5–0.95

4.2 Quantitative and Class-wise Results

Quantitative evaluation is conducted on the test dataset by comparing the proposed YOLOv11+ViT model with the YOLOv11 baseline.

The performance of both models is summarized in Table 3.

Table 3. Overall Performance Comparison Between YOLOv11 Baseline and Proposed Method Using the Test Data

Model	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5–0.95
YOLOv11 (Baseline)	0.808	0.907	0.913	0.579
YOLOv11 and ViT (Proposed)	0.899	0.921	0.951	0.622

The proposed YOLOv11+ViT framework achieves an mAP@0.5 of 0.951 and an mAP@0.5–0.95 of 0.622, indicating a clear improvement over the YOLOv11 baseline, which

records an mAP@0.5 of 0.913 and an mAP@0.5–0.95 of 0.579. The improvement in mAP@0.5–0.95 highlights enhanced localization accuracy under stricter Intersection-over-Union thresholds, confirming that transformer-based global context modeling is effective.

This improvement is further supported by the additional evaluation parameters, as the proposed model has achieved a higher Precision value of 0.899, and Recall value of 0.921, while the baseline model has achieved a precision value of 0.808, and Recall value of 0.907. This indicates that the proposed model has eliminated false detections and achieved very good coverage for the defects.

The class-wise performance comparison in Table 4 shows the improvement in the performance of the proposed model over the existing model for all defect classes. The improvement is substantial for the elongated defect classes like Cracks and Lightbands, as these defect classes cover a large area of the image. In contrast, for the defect classes like Breaks and Scars, which cover a smaller area of the image, the proposed model shows strong performance compared to the existing model.

Table 4. Class-wise Detection Performance Comparison of YOLOv11 Baseline and Proposed Method

Defect Class	YOLOv11 Baseline mAP@0.5	Proposed model mAP@0.5
Cracks	0.889	0.941
Rails	0.924	0.958
Scars	0.936	0.964
Breaks	0.901	0.928
Lightbands	0.882	0.936

To assess the stability of the reported improvements, the training process was repeated with different random initializations under the same experimental configuration. Across repeated runs, the proposed model consistently outperformed the baseline model, with only minor fluctuations in mAP values. This confirms stable convergence behavior and reproducibility of the performance gains reported in Table 3.

The efficiency of the proposed method in terms of execution time and the computational needs of the model is summarized in Table 5. Despite the computational overhead added by the Vision Transformer component in the model, the overall computation is still within the real- time threshold.

Although full transformer-based detectors such as DETR and Swin-based detection frameworks demonstrate strong global modeling capability, they typically involve higher computational overhead. The proposed lightweight integration strategy aims to balance contextual reasoning and real-time efficiency within the YOLOv11 detection framework.

Table 5. Computational Cost and Runtime Analysis of the Detection Models

Model Architecture	Number of Parameters (Million)	Computational Cost (GFLOPs)	Inference Time (ms/image)
YOLOv11 Baseline	20.06	68.2	24.2
Proposedmodel	55.12	108.4	37.1

4.3 Qualitative Results and Discussion

Qualitative results from the detection task using the proposed method architecture on test images are depicted in Figure 5. The model is able to detect multiple types of rail surface defects effectively in real-life scenarios. The proposed model produces bounding boxes that

match the location of the defects in reality, suggesting better contextual awareness over the entire rail surface.

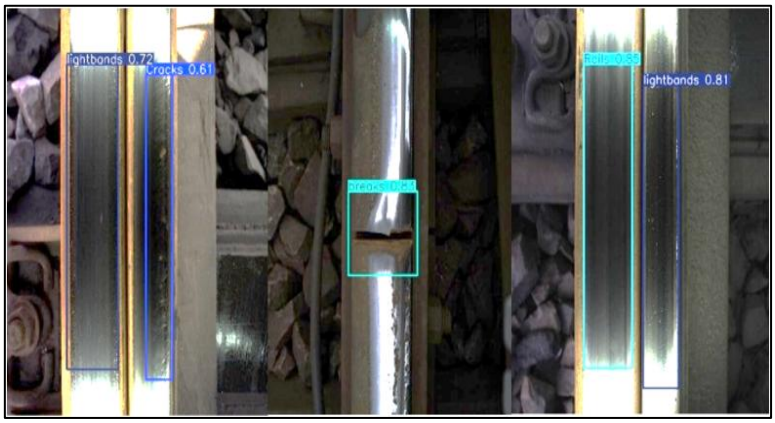

Figure 5. Qualitative Detection Outcomes from the Proposed YOLOv11-Vision Transformer Under Real World Inspection Scenarios

4.4 Confusion Matrix Analysis

Figure 6 displays the confusion matrix for the proposed model. Most values are concentrated along the main diagonal, implying that the model is correctly classifying most defects.

For the cracks class, there are 51 correct classifications. There are some misclassifications, mostly with the lightbands class, because the defects have similar shapes. The rails class has a high accuracy rate with 240 correct classifications and a few misclassifications. The scars class also has a high accuracy rate with 205 correct classifications.

For the breaks class, there are accurate classifications, as the defects have been identified correctly, with 76 correct classifications.

The lightbands class has 160 correct classifications; however, there are some misclassifications due to the defects having similar shapes.

Most misclassifications occur between the cracks and lightbands classes because the defects have similar appearances. From the confusion matrix, it is evident that the addition of the transformer improves class separability and reduces false positives.

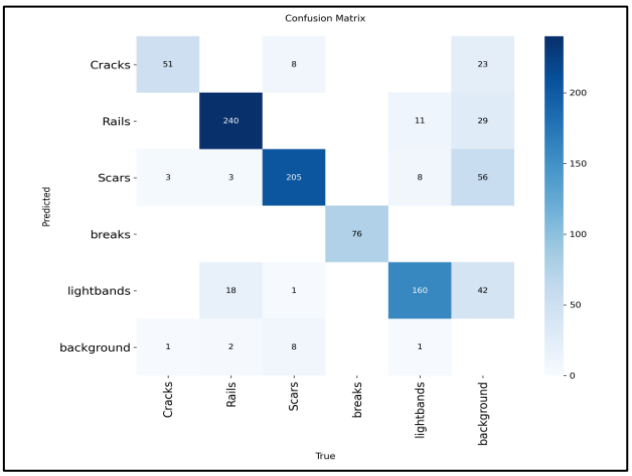

Figure 6. Confusion Matrix of the Proposed Model Evaluated on the Rail Surface Defect Test Dataset

4.5 Failure Case Analysis

Despite its impressive performance, there are certain limitations in the proposed model as well. Small cracks with poor contrast may occasionally be ignored due to the difficulty in distinguishing them at the pixel level. Additionally, there are instances where bright light bands with high reflections cause minor errors in the bounding box.

Although the transformer aids in identifying the global context, it is not easy for the model to detect extremely subtle or similar defect patterns.

5. Conclusion

The proposed model integrates the YOLOv11 algorithm and Vision Transformer architecture to form a hybrid YOLOv11-Vision Transformer architecture for detecting rail surface defects. The model outperforms its counterpart, the YOLOv11 algorithm, with metrics of precision at 0.899, recall at 0.921, and mAP@0.5 at 0.951, representing improvements of 9.1%, 1.4%, and 3.8% compared to the baseline, respectively. Moreover, the value of mAP@0.5–0.95 increased from 0.579 to 0.622, reflecting enhanced localization performance. There is a marked improvement in class-wise results for elongated defects, such as cracks and light bands, with Precision values increasing to 0.941 and 0.936, respectively. Despite the higher computational cost and inference time, the model remains feasible for real-time implementations, making it a suitable choice for rail inspections.

References

[1] M. Babu Rao, C. Kavitha, B. Prabhakara Rao and A. Govardhan, "A New Feature Set for Content based Image Retrieval," 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 2013, pp. 84-89.

[2] M. Babu Rao, Kavitha, C., Rao, B.P., Govardhan, A. “Content Based Image Retrieval Based on Dominant Color, Scan Pattern Co-occurrence Matrix of a Motif and Shape” In: Das, V.V., Thankachan, N. (eds) Computational Intelligence and Information Technology. CIIT. Communications in Computer and Information Science, vol 250. Springer, Berlin, Heidelberg, 2011, 353-357.

[3] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 770-778.

[4] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature Pyramid Networks for Object Detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, 2117-2125.

[5] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 779-788.

[6] G. Jocher et al., “Ultralytics YOLOv5,” 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[7] Wang, Yan, Kehua Zhang, Ling Wang, and Lintong Wu. "An Improved YOLOv8 Algorithm for Rail Surface Defect Detection." IEEE Access 12 (2024): 44984-44997.

[8] Fang, Zhanao, Liming Li, Lele Peng, Shubin Zheng, Qianwen Zhong, and Ting Zhu. "Yolov8n-rsdd: A High-Performance Low-Complexity Rail Surface Defect Detection Network." IEEE Access 12 (2024): 196249-196265.

[9] Cao, Yuan, Long Ma, Yongkui Sun, Feng Wang, and Shuai Su. "Improved YOLOv8 for High-Precision Detection of Rail Surface Defects on Heavy-Haul Railways." Chinese Journal of Electronics 34, no. 3 (2025): 802-815.

[10] Allada, Apparna, Rajaram Bhavani, Kavitha Chaduvula, and Rajaram Priya. "Alzheimer's Disease Classification Using Competitive Swarm Multi-Verse Optimizer-Based Deep Neuro-Fuzzy Network." Concurrency and Computation: Practice and Experience 35, no. 21 (2023): e7696.

[11] Kantapalli, Bhaskar, and Babu Rao Markapudi. "SSPO-DQN Spark: Shuffled Student Psychology Optimization Based Deep Q Network with Spark Architecture for Big Data Classification." Wireless Networks 29, no. 1 (2023): 369-385.

[12] Edupuganti, Mounika, V. Rathikarani, and Kavitha Chaduvula. "Classification of Heart Diseases Using Fusion Based Learning Approach." International Journal of Intelligent Systems and Applications in Engineering 12, no. 8s (2024): 570-580.

[13] A. Dosovitskiy et al., “An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale,” New York City, 23-26 June 2021, 45-67.

[14] Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows." In Proceedings of the IEEE/CVF international conference on computer vision, 2021, 10012-10022.

[15] EngDes2, “Rail Surface Defects Computer Vision Dataset,” Roboflow Universe. [Online]. Available: https://universe.roboflow.com/engdes2/rail-surface-defects-flrty-omt9o