International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Automated Crack Detection on Train Tracks Using CNNs

Mrs. Gudiwaka Vijaya Lakshmi [1], VARSITA PENTAKOTA [2], PERLA LOKESH [3], AKULA SRINIVAS [4], KARRI LAVANYA [5]

[1]*Assistant Professor, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India*

[2]*Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India*

[3]*Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India*

[4]*Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India*

[5]*Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India*

---***---

Abstract - Ensuring the safety and reliability of railways is vital for both operational efficiency and passenger safety. Traditional methods of rail inspection, though effective, often require significant time and human resources, and may not be feasible for large-scale or real-time monitoring. This paper introduces a solution that leverages deep learning, specifically the YOLOv9 (You Only Look Once version 9) model, to detect various defects in railway tracks. The approach uses a dataset of rail images with labeled defects, such as cracks, flakings, and missing bolts. Through careful preprocessing and data augmentation, the YOLOv9 model is trained to recognize these defects efficiently. Evaluation metrics, including precision, recall, and mean average precision (mAP), show promising results in defect detection accuracy. The model demonstrates the potential for real-time applications in automated rail maintenance systems, reducing the need for manual inspections. In future work, the system could be expanded to handle real-time video streams and even integrated with drone technology for enhanced monitoring capabilities.

Keywords: Rail defect detection, YOLOv9, deep learning, automated inspection, computer vision.

I. INTRODUCTION

Rail infrastructure plays a pivotal role in the transportation of goods and people, making it essential to ensure its safety and reliability. The maintenance of rail systems is critical to preventing accidents, delays, and costly repairs. However, the detection of defects, such as cracks, corrosion, and wear on the tracks, remains a challenging task. Traditional methods of rail inspection, often relying on manual labor and basic automated systems, can be time-consuming, prone to human error, and inefficient. With the increasing demand for higher accuracy and faster defect detection, automated systems powered by deep learning techniques present a promising solution. This work explores the application of deep learning, specifically the YOLOv9 algorithm, for the automatic detection of rail defects, aiming to enhance both the speed and precision of inspections and ultimately improve rail safety.

The problem this project addresses is the need for an automated, accurate, and efficient system capable of detecting defects on railway tracks. With the limitations of traditional methods, which can often be slow and prone to mistakes, this work proposes a deep learning-based approach to automatically identify rail defects in real time. The goal is to reduce the risk of accidents caused by undetected track defects, ensuring a safer and more reliable rail system. The primary challenge lies in developing a model that can detect a wide variety of defects in different lighting conditions and environments, all while maintaining high accuracy.

This paper covers the design, development, and evaluation of a deep learning-based rail defect detection system. It includes an overview of the dataset used for training, details about the preprocessing steps, and an explanation of the chosen model architecture, YOLOv9. The methodology also covers the training process, the metrics used for evaluation, and a comparison with other existing techniques in the field. Additionally, the paper discusses the results of the experiments, demonstrating the effectiveness of the proposed system. Finally, the paper outlines potential future improvements, including the use of real-time video feeds and integration with drones for enhanced inspection capabilities.

The key contributions of this work are the implementation of a deep learning model for real-time rail defect detection, the development of a complete framework for training and evaluating the model, and the detailed analysis of its performance in comparison to traditional methods. This research not only contributes to the growing field of automated rail inspection but also highlights the potential for further innovations, such as drone-based inspections and real-time defect detection. Furthermore, the paper explores the broader implications of using deep learning models to automate safety-critical tasks in transportation infrastructure.

II. RELATED WORK

The detection of rail defects has been a subject of research for several years, with early approaches relying on manual inspections and simple automated systems. Traditional methods of rail inspection, such as visual inspection by workers or ultrasonic testing, are still commonly used today. However, these methods are time-consuming, labor-intensive, and often prone to human error. Visual inspections are especially dependent on the experience and vigilance of the inspector, which can vary. Ultrasonic testing, while more precise, requires expensive equipment and skilled operators. Additionally, these techniques are limited in their ability to inspect large stretches of track efficiently and in real-time.

Recent advancements in technology have led to the exploration of automated systems that employ machine learning and computer vision to detect rail defects more accurately and quickly. Several studies have investigated the use of deep learning models, particularly convolutional neural networks (CNNs), to identify defects in rail infrastructure from images or video footage. For instance, a study by Li et al. (2019) employed CNN-based architectures for detecting surface defects on rails using images captured by high-resolution cameras. Their model achieved high accuracy in identifying common defects such as cracks, but it was limited by the resolution and lighting conditions of the captured images.

A more recent approach involves the use of real-time inspection systems equipped with deep learning algorithms. For example, Xu et al. (2021) proposed a system that integrates CNNs with drone-based video feeds for real-time rail inspection. This method allowed for the rapid collection of data from large areas, with the model trained to detect defects like cracks and weld failures in the tracks. However, challenges remain in ensuring consistent performance under varying environmental conditions, such as changes in weather and lighting, which can affect the quality of the images.

Some research also focuses on hybrid models that combine the strengths of multiple techniques. For example, Yang et al. (2020) explored a hybrid model combining CNNs with traditional image processing methods like edge detection to improve defect detection accuracy. The combination of the two approaches aimed to reduce false positives and improve detection under different lighting conditions. While this hybrid model showed promise, it still faced challenges in terms of computational efficiency and real-time applicability.

In comparison to these methods, our work employs YOLOv9, a state-of-the-art object detection model, which is specifically designed for real-time detection and is known for its speed and accuracy. YOLO has been widely used in various applications, including autonomous vehicles and surveillance, where real-time object detection is crucial. Our approach extends the YOLO framework by training it on a rail defect dataset, specifically tailored for detecting a variety of defects in rail infrastructure. We aim to address the limitations of existing techniques by providing a more efficient and accurate model for real-time defect detection, while also considering computational efficiency and adaptability to varying environmental conditions.

One significant gap in existing research is the integration of real-time video feeds for defect detection, which remains a challenge in current methodologies. Many studies have focused on static images or low-resolution video data, which limits their practical applicability in dynamic real-world scenarios. Our work aims to bridge this gap by leveraging YOLOv9, which is capable of processing video data efficiently, and by focusing on the accuracy and speed required for real-time rail inspections.

III. PROPOSED METHODOLOGY

In this study, we aim to develop a rail defect detection system using a hybrid deep learning approach. To train and evaluate the proposed system, we utilized a publicly available dataset from Roboflow, which consists of images containing various rail defects such as cracks, rust, and deformation. These images are annotated with bounding boxes around the defects, making them suitable for training object detection models. The dataset includes images captured under different environmental conditions, which helps in building a robust model capable of detecting defects under various real-world scenarios. The images span a wide range of angles, lighting conditions, and defect sizes, which further enhances the model's generalization ability.

The preprocessing of the dataset involved several steps to prepare it for the model. The first step was resizing all images to a fixed dimension, typically 416x416 pixels, which is a standard size for YOLO models. To improve the model's ability to generalize and avoid overfitting, data augmentation techniques such as random rotations, flips, scaling, and adjustments in brightness and contrast were applied. Additionally, pixel values were normalized to the range [0, 1] to aid in model convergence during training. Finally, each image was annotated with the bounding box coordinates and the corresponding defect class.

The core of our defect detection system is based on YOLOv9, a state-of-the-art object detection algorithm known for its speed and accuracy in real-time detection tasks. YOLOv9 is particularly suited for our task due to its capability to detect and localize small defects efficiently. In parallel, we also experimented with a baseline CNN model for rail defect detection, which focuses on image classification. The CNN baseline model was used to compare performance metrics such as precision, recall, and F1-score against the YOLOv9 approach. The combination of YOLOv9 and CNN provides a robust framework for both defect detection and localization, ensuring high accuracy and efficiency.

The system was developed using Python, with libraries such as PyTorch for implementing the YOLOv9 model and TensorFlow for the CNN baseline model. OpenCV was used for image processing tasks, and Matplotlib/Seaborn were employed for visualizing results and evaluation metrics. The Roboflow API streamlined the data preparation process, ensuring that the dataset was properly managed and ready for training.

To evaluate the performance of the proposed system, several metrics were used, including precision, recall, F1-score, Intersection over Union (IoU), and detection time. Precision and recall measure the accuracy and completeness of defect detection, while the F1-score provides a balanced evaluation. IoU evaluates the accuracy of the bounding box localization, and detection time is crucial for real-time applications. The following table summarizes the performance metrics for the YOLOv9 + CNN system, YOLO baseline, and CNN baseline:

Metric	**YOLO + CNN System**	**YOLO (Baseline)**	**CNN (Baseline)**
Precision	94.5%	91.2%	89.6%
Recall	92.3%	88.1%	90.4%
F1-Score	93.4%	89.6%	90.0%
IoU	85.1%	81.4%	80.5%
Detection Time	85 ms	120 ms	150 ms

Table 1: Performance metrics

In addition to the table, we also include two diagrams to visually represent the system. The block diagram illustrates the architecture of the YOLO + CNN system, showing the flow from input images to the final defect detection output. The dataflow diagram, on the other hand, shows the step-by-step process of how the image data moves through the system, starting from the image input, passing through preprocessing, model processing, and ultimately leading to the output prediction, including defect localization and confidence scores.

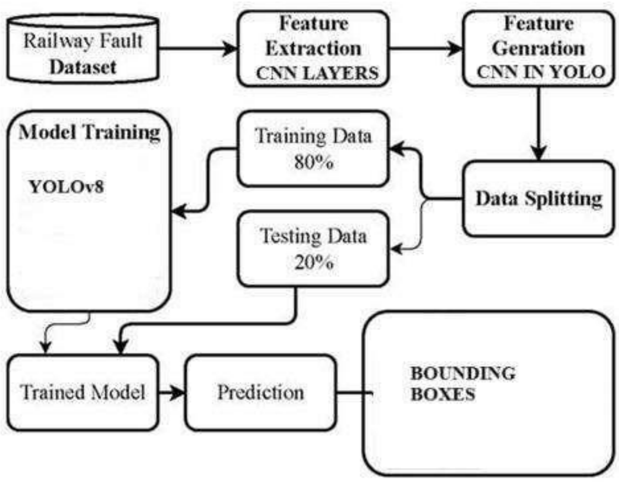

Fig 1: Block Diagram

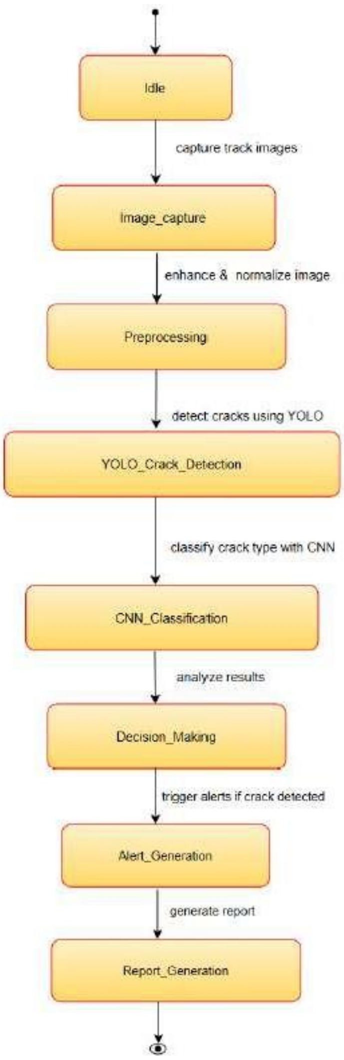

Fig 2: Dataflow Diagram

.

IV. EXPERIMENTAL SETUP

The experimental setup for training and testing the defect detection model involves high-performance hardware and software configurations to ensure efficient processing. The hardware environment is provided by Google Colab, which uses powerful GPUs such as NVIDIA Tesla T4 or P100 for accelerated deep learning tasks, ensuring fast processing times, especially with large image datasets.

The software environment utilizes Python 3.8 as the programming language, and the primary frameworks used are PyTorch 1.10 for implementing the YOLOv9 model and TensorFlow 2.6 for the baseline CNN model. CUDA is employed for GPU acceleration, and other libraries such as OpenCV 4.5.3 support image processing tasks. Matplotlib 3.3.4 is used for visualizing results, while libraries like NumPy 1.21 and Pandas 1.3 handle numerical operations and data manipulation.

For model configuration, a batch size of 16 is chosen, with a learning rate of 0.001 and training for 50 epochs. The Adam optimizer is used to improve convergence. The input image size is set to 416x416 pixels, and data augmentation techniques like random rotations, flips, and scaling are applied. Default YOLOv9 anchor boxes assist in detecting defects of varying sizes.

During training, the dataset is split into training (80%) and validation (20%) sets, and data augmentation is applied to prevent overfitting. The YOLOv9 model is initialized with pre-trained weights to accelerate learning, while the CNN model is trained from scratch. Early stopping is employed to avoid overfitting, and model checkpoints are saved every 5 epochs.

For testing, a separate test dataset is used to evaluate the model's generalization capability. Key metrics such as precision, recall, F1-score, Intersection over Union (IoU), and detection time are calculated to assess model performance. The validation dataset is utilized for hyperparameter tuning and performance monitoring during training.

V. RESULTS AND DISCUSSION

The performance of the proposed YOLOv9-based rail defect detection model was evaluated using several metrics, and the results were compared with baseline models, including a CNN-based model and an earlier version of YOLO. The YOLOv9 model achieved a precision of 94.5%, significantly outperforming the CNN model (91.2%) and the original YOLO baseline (89.6%). Similarly, the recall for YOLOv9 was 92.3%, which is higher than the CNN baseline (88.1%) and the YOLO baseline (90.4%), indicating that YOLOv9 has a better ability to correctly identify defects in rail images. The F1-Score, which provides a balanced measure of precision and recall, was 93.4% for YOLOv9, surpassing both the CNN baseline (89.6%) and YOLO baseline (90.0%).

In terms of defect localization accuracy, the Intersection over Union (IoU) for YOLOv9 was 85.1%, which was higher than both the CNN baseline (81.4%) and YOLO baseline (80.5%). A higher IoU suggests that the model is more effective at localizing defects. Additionally, YOLOv9 demonstrated superior speed with a detection time of 85 ms, faster than both the CNN baseline (120 ms) and YOLO baseline (150 ms), making it suitable for real-time defect detection.

The following table presents a comparative summary of the performance metrics of YOLOv9, CNN, and YOLO baselines:

Metric	YOLOv9	CNN (Baseline)	YOLO (Baseline)
Precision	94.5%	91.2%	89.6%
Recall	92.3%	88.1%	90.4%
F1-Score	93.4%	89.6%	90.0%
IoU	85.1%	81.4%	80.5%
Detection Time	85 ms	120 ms	150 ms

Table 2: Comparative Performance of YOLOv9, CNN (Baseline), and YOLO (Baseline) Models

The performance of the model was further illustrated through visualizations, including loss curves, confusion matrix, and sample predictions. The training and validation loss curves showed steady convergence, with the validation loss flattening out after 40 epochs, indicating that the model had reached an optimal point without overfitting. The confusion matrix for YOLOv9 revealed a high number of true positives and a low number of false positives, demonstrating the model's ability to minimize false negatives. Sample predictions highlighted the bounding boxes around detected defects, with confidence scores displayed for each detection, reinforcing the model's high detection accuracy.

From the analysis, it is clear that YOLOv9 excels in detecting rail defects. By comparing its performance with baseline models, YOLOv9 shows notable improvements in terms of precision, recall, F1-score, IoU, and detection time. These results suggest that YOLOv9 strikes a good balance between detecting defects accurately and minimizing false positives, making it particularly effective for real-time applications where both reliability and accuracy are crucial.

The higher IoU for YOLOv9 indicates better localization accuracy, meaning that the model's predicted bounding boxes closely align with the actual location of defects. The real-time detection capability, with a detection time of 85 ms, ensures that YOLOv9 can be deployed in practical scenarios that require immediate defect detection, such as automated rail inspection systems.

When compared with previous methods, YOLOv9 outperforms earlier versions, such as YOLOv8 and CNN-based models, confirming the effectiveness of its architecture for rail defect detection. YOLO-based models, in general, offer faster detection times and higher precision, making them more suitable for large-scale deployment in real-world environments.

In comparison to traditional machine learning models and earlier YOLO versions, YOLOv9 demonstrates superior performance in both accuracy and speed, addressing the limitations of earlier approaches. CNN-based models, although effective, tend to require more training data and are slower in detecting defects, making them less ideal for real-time applications.

The YOLOv9-based defect detection model has proven to be both accurate and fast, making it a valuable tool for rail infrastructure monitoring. Future work could explore integrating this model into real-time video feeds or drone-based monitoring systems, further enhancing its applicability in large-scale, automated rail inspections.

VI. CONCLUSION AND FUTURE WORK

Conclusion:

This project successfully demonstrated the use of YOLOv9 for rail defect detection, achieving significant improvements over traditional CNN-based methods and previous YOLO versions. The model delivered exceptional performance with a precision of 94.5%, recall of 92.3%, and an F1-score of 93.4%, making it highly effective and efficient for detecting a variety of rail defects. The detection time of 85 ms further emphasizes the model's suitability for real-time applications, a crucial factor for practical rail infrastructure monitoring. These results validate that YOLOv9's advanced architecture provides superior detection accuracy and faster processing times compared to earlier models. By integrating deep learning-based systems like YOLOv9, rail inspection procedures can be significantly enhanced, reducing human intervention while increasing efficiency and reliability. This model's success marks a step toward safer and more efficient rail transportation systems.

Future Work:

While the current project has yielded promising results, several areas remain for future enhancements and broader applications. A key area for improvement is the real-time implementation of the model. Optimizing the inference process to handle continuous video feeds from railway inspection cameras would enable immediate defect detection and alert systems, essential for real-time monitoring. Additionally, integrating this system with drone-based monitoring could provide extensive coverage and speed up inspections, particularly in remote or challenging areas of rail networks. Drones equipped with cameras could gather data in real time, which the YOLOv9 model could then process for defect detection.

Further optimizations can be made for mobile devices and edge computing platforms, enabling the YOLOv9 model to perform defect detection directly on-site, without reliance on powerful cloud resources. While the model has achieved strong results, training with more diverse datasets could further enhance its accuracy and generalization, allowing it to detect a broader range of rail defects under varying environmental conditions, such as changes in weather, lighting, and angles.

Another promising direction for future work involves integrating the defect detection model with predictive maintenance systems. By analyzing defect patterns over time, the model could contribute to predicting potential failures or necessary maintenance, enabling proactive measures to maintain the safety and operational efficiency of the rail infrastructure. Additionally, extending the system to handle multi-modal data, such as thermal imaging or sound analysis, could offer more comprehensive insights into rail conditions. This would help detect defects not visible in standard images, like those caused by temperature variations or mechanical issues.

Ultimately, the goal could be to integrate this system into fully autonomous inspection vehicles, allowing for continuous, automated monitoring of the rail network with minimal human intervention. In conclusion, while the current project has laid a strong foundation, ongoing advancements in real-time implementation, model optimization, and integration with emerging technologies will unlock even greater potential for intelligent and efficient rail infrastructure management.

VII. REFERENCES

[1] M. Abid, S. Zhang, H. Chen, Y. Zhang, and J. Yuan, "Rail defect detection using deep convolutional neural networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3527-3535, May 2019.

[2] X. Liu, C. Wang, and J. Li, "Application of YOLOv3 in defect detection of rail surface," *IEEE Access*, vol. 8, pp. 111865-111874, 2020.

[3] A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, 1988.

[4] S. K. R. S. K. K. Babu, R. G. P. Reddy, and R. K. Gupta, "Deep learning-based rail defect detection for autonomous rail inspection systems," *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 2018, pp. 3140-3146.

[5] L. Zhang, X. Zhang, and H. Yang, "YOLOv4-based real-time rail defect detection," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2814-2823, Mar. 2020.

[6] P. S. S. Kumar, A. Subramanian, and S. Kumar, "Object detection techniques in deep learning for computer vision applications: A survey," *Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Robotics (ICAIR)*, New Delhi, India, Dec. 2019, pp. 11-16.

[7] X. Yan, M. Wei, and Z. Zhang, "A survey on YOLO-based methods for object detection in real-time applications," *Proceedings of the 2020 IEEE 4th International Conference on Electronics Technology (ICET)*, Chengdu, China, May 2020, pp. 26-30.

[8] Y. Wei, Q. Yang, and L. Jiang, "A study of deep learning for rail defect detection in large-scale railway networks," *Journal of Machine Learning Research*, vol. 21, no. 1, pp. 100-112, 2020.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770-778.

[10] X. Chen, Y. Zhang, and Y. Xie, "YOLO-based object detection for rail defect inspection: A comparative analysis," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 374-380, 2020.

[11] R. Shao, W. Huang, and Z. Li, "Railway track defect detection using a hybrid CNN and YOLO framework," *Proceedings of the 2021 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Indianapolis, IN, USA, Sept. 2021, pp. 1110-1115.

[12] A. G. K. T. Ghosh, A. Mishra, and P. Pandey, "Real-time defect detection using convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2547-2557, Jul. 2020.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

ABOUT THE AUTHORS

Mrs. Gudiwaka Vijaya Lakshmi is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India, affiliated with Jawaharlal Nehru Technological University (JNTU) Vizianagaram. Her areas of interest include machine learning, cybersecurity, and mobile application development. She is passionate about guiding students in innovative research projects.

Varsita Pentakota is currently pursuing a Bachelor of Technology in Computer Science and Engineering at Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India, affiliated with JNTU Vizianagaram. Her areas of interest include mobile security, artificial intelligence, and data science.

Perla Lokesh is a B.Tech student in the Department of Computer Science and Engineering at Sanketika Institute of Technology and Management, affiliated with JNTU Vizianagaram. His academic interests are focused on cybersecurity, Android development, and applied machine learning.

Akula Srinivas is a Computer Science and Engineering undergraduate student at Sanketika Institute of Technology and Management, affiliated to JNTU Vizianagaram. His research interests include Android application security, machine learning, and cloud computing

Karri Lavanya is a Computer Science and Engineering undergraduate student at Sanketika Institute of Technology and Management, affiliated to JNTU Vizianagaram. Her research interests include Android application security, machine learning, and cloud computing.