Deep Learning Approaches for Railroad Infrastructure Monitoring: Comparing YOLO and Vision Transformers for Defect Detection

Advay Chandramouli*, Hwapyeong Song†, Mingyan Liu‡, Aayush Damai†, Husnu S. Narman†, Ammar Alzarrad†
*The University of Texas at Dallas, advay.chandramouli@utdallas.edu
† Marshall University, {damai2, song24, narman, alzarrad}@marshall.edu
‡ Smith College, iliu32@smith.edu

***Abstract*—Railroad crossties are paramount to ensuring structural integrity and passenger safety. Traditional inspection methods have been a manually intensive process, subject to fatigue, bias, and human error, prompting an inquiry into more consistent and expedited inspection modalities. Automated defect detection leveraging deep learning and computer vision can drastically reduce diagnostic time and ensure prompt maintenance intervention. This paper evaluates two contrasting state-of-the-art object detection models, You Only Look Once (YOLOv11, the latest release from Ultralytics) and Real-Time Detection Transformer (RT-DETR), for automating detection of defective ties, marking the first competitive benchmarking within this domain. This study uses real, field-test footage collected on stretches of railroads that was standardized and preprocessed for normalization to form our dataset, consisting of 500 annotated frames across three distinct classes: wood checks, decay, and ties. Both models were trained using 5-fold cross-validation and evaluated based on the F1 score, Precision, Recall, and Mean Average Precision (mAP) at varying IoU thresholds. YOLOv11 outperformed RT-DETR in all metrics except Recall (0.9104 vs. 0.9119), achieving an F1 score of 0.9400, mAP50 of 0.9530, and mAP50-95 of 0.9014. These results suggest that YOLOv11 is effective not only in defect recognition but also in spatial localization at various precision thresholds, fostering higher safety standards in the process. This makes it a strong candidate for deployment on lightweight mobile or embedded platforms, representative of real-world use cases. Overall, this research demonstrates the robust utility of deep learning for enhanced infrastructure monitoring, reducing inspection time and labor costs while heightening passenger safety and mitigating derailment risks, with future work needed to assess performance under varied lighting, weather, and tie conditions.**

***Index Terms*—Railroad Safety, Infrastructure Monitoring, Computer Vision, Deep Learning, Object Detection, Vision Transformers**

I. Introduction

Modern railroads are pivotal modes of daily public transportation, bridging cities, states, economies, and cultures. According to the U.S. Census Bureau, in a 2019 survey, more than 11.8% of public transit commuters in metropolitan complexes relied on railroad systems to commute [1]. Seemingly minor defects can exacerbate under continuous pressure and stress, leading to train derailments [2]. Given the dependence of regional transit and freight logistics on railway systems, issues such as aging infrastructure, high passenger volume, and transit speeds have heightened the need for effective railroad Structural Health Monitoring (SHM).

Traditional Non-Destructive Testing (NDT) methods have been favored in diagnosing structural defects. Wang et al. [3] inform that years of high speed and loads make the top surfaces of rails vulnerable to defects. While effective at finding medium to large faults, Magnetic Flux Testing (MFL) is hindered on a few fronts. First, its efficacy is maximized only at low speeds up to 20km/hr [3]. Second, as Gong et al. [4] suggest, even at low speeds, MFL testing captures fluctuations in flux signals due to surface irregularities (coatings, debris). Third, the fallback option is using manual diagnostic tools, undercutting MFL's accuracy and is time and labor-intensive [5]. On the contrary, other NDT methods, such as Ultrasonic Testing (UT), detect internal structural faults at speeds up to 80km/hr. However, high-frequency ultrasonic waves create disturbances or echoes, rendering surface detection through previous methods futile in extreme cases [3]. Moreover, the vast speed discrepancies between these methods make parallel assessment of surface and internal faults challenging.

Consequently, this limitation has prompted research into automated, real-time defect detection using machine learning. Convolutional Neural Network (CNN)-based object detectors, particularly the You Only Look Once (YOLO) lineage, have seen widespread adoption in this space. In contrast, transformer-based architectures remain underexplored despite promising results in other computer vision domains. Additionally, few studies focus on crosstie (also referred to as ties' or sleepers') defect detection, which is imperative for track-level structural integrity. This paper addresses these gaps by benchmarking YOLOv11 against the Real-Time Detection Transformer (RT-DETR) to identify railroad tie defects under diverse visual conditions.

The remainder of this paper presents the following: Section II outlines existing scholarly discussions. Section III delves into the machine learning models used. Section IV introduces our methodology, spanning our dataset and training protocol. Section V reveals the quantitative findings of our study, while discussing these results in a real-world context. Section VI finally concludes the paper with directions for future work.

II. Related Works

A. Early Machine Learning Approaches to Rail Defect Detection

Rapid advancements in imaging technology and computational resources have catalyzed exploring Machine Learning (ML) applications for rail surface defect detection, beginning in static image classification and trending toward real-time detection [6]. Morteza Mirzaei et al. [7] used ML to identify rail components, such as ballasts and ties, from LiDAR and inductive proximity sensor data. Using both decision trees and logistic regression led to a classification accuracy of 95%, highlighting the efficacy of ML in this domain, even if not in real-time deployment contexts [7]. Similarly, Sresakoolchai and Kaewunruen [8] integrated a dataset of defect logs and track profiles (shape and alignment metrics) collected from Track Geometry Cars (TGCs), testing a combination of Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and Gradient-Boosted (GB) models. This led to 94% accuracy, while also uncovering correlations between track curvature and defect likelihood, underscoring inferential possibilities with ML models without auxiliary inputs [8]. Collectively, these studies demonstrate how primitive ML models effectively capture the complexities of heterogeneous data streams, even in the absence of vision-based features.

B. Deep Learning Methods and CNN-Based Models

Xu et al.'s study [9] build upon previous studies, implementing Deep Learning (DL) to address subgrade defects impacting foundational soil and ground mixtures using pre-trained models and CNNs trained on Ground Penetrating Radar (GPR) waveforms. With an F-1 score of 83.6%, CNNs outperformed simpler methods such as SVMs and Histogram-Oriented Gradients (HOGs), confirming that they can capture insights from dense datasets, even in subterranean, non-visual contexts [9]. CNNs are equally robust at visual feature extraction, forming the basis for modern object detection. Even in a spartan binary classification of defective image samples (positive/negative), CNNs implemented using TensorFlow achieved an overall accuracy of 92.21% [10]. Compared to classical techniques, CNNs facilitate the spatial localization of surface defects, capturing abstract visual patterns across a myriad of track settings, making them effective in exhaustive search tasks [6], [9], [11].

These conclusions spurred the development of CNN-based detection algorithms such as You Only Look Once (YOLO). For instance, Damai et al. [11] benchmarked four different versions of the YOLO model (v5, v8, v9, and v10) to find a configuration best suited to detect missing track bolts, concluding that YOLOv5 was performed best with an F-1 score of 86.45%. However, D'Arms et al.'s investigation [12] disputed this finding by demonstrating that after training eight different models (YOLO variants and ResNet) to identify cracks and gaps in railroad infrastructure, YOLOv8 performed best with an accuracy of over 92%. Given that YOLOv5 is intended for lightweight deployment contexts, it is noteworthy that it outperformed YOLOv8 in Damai et al.'s study. This disparity may arise from applying these models to different subproblems, leading to exploring how changing the model architecture impacts performance.

Both Chen et al.'s [13] and Wang et al.'s [14] study aim to fill this void, advancing beyond more incremental enhancements to YOLO and reflecting a shift towards architectural optimality balancing performance with real-world deployability. Minimizing the model parameters and computation load without sacrificing detection accuracy is paramount. Chen et al. [13] implemented Depthwise Separable Convolutions (DSP) and Ghost modules, reducing the computational workload of YOLOv5s by up to 15%. Unlike the former study, Wang et al. [14] retain the base model size for YOLOv8 to show that architectural tuning, using feature recalibration and modified convolutional structures, can still yield improved performance while minimizing resource demands. These modifications align with a real-world need to make real-time infrastructure inspection compatible with mobile and field devices. Both studies use enhanced attention mechanisms for improved feature representation (PSCA and EMA) to dynamically recalibrate features across multiple layers. This result proves valuable in complex faults, where standard YOLO feature maps may underperform. Another commonality between both studies is modifying YOLO's default loss function, substituting Alpha-IoU and Focal-SIoU, respectively, to improve robustness against hard-to-classify samples [13], [14]. In the case of Wang et al. [14], these modifications to YOLOv8n yielded enhanced accuracy of 94.1%.

C. Mixed Modality Frameworks and Data Fusion Techniques

Mixed-modality approaches in research have gained traction, combining object detection methods with complementary inputs or reasoning frameworks. Unimodal approaches stand vulnerable to real-world conditions due to insufficient, representative training data, leading to poor generalization and over-reliance on hyperparameter tuning. To address this issue, researchers have explored two approaches, including integrating new input frameworks and addressing dataset limitations to heighten decision-making accuracy.

Unimodal DL methodologies, despite their promise, face limitations in complex deployment environments due to variations such as noisy, real-world data, small targets, and shadows [15]. Recent work in mixed modality frameworks hopes to overcome these deficiencies. For example, Rivero et al.'s paper [15] integrated a modified UberNet and YOLO DL model with belief theory frameworks (Markov Random Fields) to simulate human-like reasoning when detecting structural faults such as surface damage, broken rails, and missing fasteners. Accompanying this research direction is Wen et al.'s study [6], in which a framework called MSCM-Net is introduced, fusing RGB image inputs with geometric depth maps through the use of ResNet34 and multi-stream fusion modules such as MSDF, CCAM, and TSDF for an enriched spatial representation of rail defects. Concurrently, an affiliated decoder structure

localizes and reconstructs defect boundaries under visually ambiguous conditions. MSCM-Net outperformed unimodal approaches across all standard evaluation metrics, particularly showing promise in generalizing defect characteristics even in the presence of noisy features. Despite enhanced reasoning frameworks and architectures, Rivero et al. [15] and Wen et al. [6] acknowledge that model performance is predominantly contingent on data quality, with complications such as inconsistent lighting conditions, boundary regions, and target sizes ultimately undermining prediction quality and generalizability.

Scarcity in robust, model-worthy railroad datasets has prompted interest in synthetic data generation, particularly using Generative Adversarial Networks (GANs). GANs are conflicting DNNs, consisting of two components, a generator that trains on authentic images and reproduces synthetic samples and a discriminator that repeatedly verifies whether a sample is real or synthetic, thereby creating high-fidelity samples [16], [17]. Xia et al. [5] report that it augments training data and enables research groups to control defect characteristics and appearance. However, GANs in this domain are limited in generalizability, especially for underrepresented subproblems. Labeling is another hindrance with visual datasets, and current literature acknowledges it as a significant bottleneck in dataset construction for rail defect detection. Moreover, label fatigue due to this manual process directly impacts model performance. Lester et al.'s paper [18] introduces a semi-automated labeling pipeline using reinforcement learning. They use a pre-trained YOLO model from a small subset of their dataset to label the remaining samples, attaching class labels and bounding boxes. Their work shows massive promise, reducing label time by up to 50%: implementing this in future work can minimize bias, fatigue, and time costs, particularly in large-scale ML workflows.

D. Emergence of Transformer-Based Architectures

Recent developments in Vision Transformers (ViTs) have challenged the contention that transformer architectures are not suitable for real-time detection applications. Zhao et al. [2] contend that their Real-Time Detection Transformer (RT-DETR) model outperforms analogous YOLOv5 on the COCO (Common Objects in Context) dataset, particularly at high-speed image thresholds (108 fps). This study challenges the conventional notion that transformers are excessively slow or impractical for real-time deployment. Other features such as limiting Non-Maximum Suppression (NMS), hybrid encoder architecture, and adjustable decoder layers, allow ViTs to support more robust object detection pipelines tuned for their respective deployment contexts. Phaphuangwittayakul et al. [19] adapted a Dual-Attention Vision Transformer (DaVIT) in parallel for railroad defect detection. Its dual attention encoder architecture allows for the capture of a global context and granular features, resulting in a high degree of precision in object detections. Trained across a multitude of classes and features (rail ballasts, fishplates, fasteners, and surface defects), they found that ViTs adapt well to unseen features, particularly in scenarios where inadequate data is available for benchmarking. Most notably, this study does not compare the defect detection performance of DaVIT with a comparable YOLO despite the latter's widespread adoption in this domain. This omission underscores a gap in the literature, prompting exploration of how the latest transformer architectures compare with the preeminent CNN-based detector, YOLO.

On the one hand, YOLO has been extensively benchmarked in railroad defect detection scenarios, but few studies focus on tie-level faults. On the other hand, ViTs represent an emerging technology, yet their full potential remains relatively untapped in this domain, particularly in comparison with YOLO-based methodologies. This paper bridges this gap by comparing the performance of a traditional CNN-based detector (YOLOv11) against a transformer architecture-based detector (RT-DETR) to detect tie defects under diverse visual conditions and to gain insight into the scope for future field deployment.

III. Machine Learning Models

The existing scholarly conversation enabled a comparison of traditional CNN-based detectors with emerging transformer-based detectors. However, selecting an appropriate model was non-trivial, given the bevy of algorithms available. To facilitate our selection, we set a three-fold criterion: (1) real-time inspection potency to relieve manual assessment and expedite maintenance, (2) effective balance of speed and accuracy, particularly in object localization, and (3) multi-class detection capabilities for defective ties. Accordingly, we finalized YOLOv11 and RT-DETR for their alignment with this criterion.

A. You Only Look Once (YOLO)

Object detection tasks encompass both object recognition and localization, utilizing bounding boxes. Early algorithms divided input images into multiple regions, classified each partition, and derived high probability scores, indicating accurate detection.

However, this repetitive process was both time and resource-intensive. The YOLO lineage of object detection algorithms addressed this limitation by using a one-stage CNN to analyze all regions of an image, enabling real-time detection prospects. Developed by Darknet in 2015, YOLOv1 overlays a grid on an input image and generates a series of bounding boxes around the grid cells [20]. Each bounding box is associated with a confidence level that reflects the probability of the box containing a particular class. Lastly, Non-Maximum Suppression (NMS) is implemented during post-processing to systematically review all proposed bounding boxes. After sorting bounding boxes by their respective class probabilities, NMS iteratively considers the highest probability bounding box and suppresses any overlapping bounding boxes, based on a predefined threshold.

In the following decade, YOLO has undergone rapid evolution. YOLOv2 introduced batch normalization, anchor boxes, and k-means clustering to improve localization accuracy and recall, particularly for small objects. YOLOv3 advanced the

architecture by adopting Darknet-53, integrating residual connections and multi-scale prediction while replacing softmax with independent logistic classifiers to support multi-label classification. YOLOv4 and YOLOv5 leveraged a CSPDarknet53 backbone, Spatial Pyramid Pooling (SPP), and PANet for bottom-up path aggregation, mosaic data augmentation, and Self-Adversarial Training pipelines. Furthermore, YOLOv5 was the first of its kind to support multiple model sizes (s, m, l, or x) to accommodate varying deployment scenarios. YOLOv7 introduced E-ELAN (Extended Efficient Layer Aggregation Network), enhancing feature extraction without compromising existing gradient paths. Later variants like YOLOv8 and YOLOX presented decoupled heads, anchor-free detection, and modular task support-firmly establishing a shift toward flexible, accurate, and lightweight object detectors for diverse use cases. YOLOv9 took a significant stride by introducing Programmable Gradient Information (PGI) and a Generalized Efficient Layer Aggregation Network (GELAN), heightening network gradient flow and feature reuse during training, setting a new benchmark on MS COCO, especially in real-time precision [20].

In late 2024, Ultralytics unveiled YOLOv11, building upon a decade-long pursuit of enhancements tailored to modern computer vision challenges. It features an upgraded backbone for enhanced feature extraction in complex search tasks. Moreover, with a revised neck architecture and an optimized training and evaluation pipeline, YOLOv11 outperformed predecessors like YOLOv8 in mean Average Precision (mAP) on the COCO dataset despite housing 22% fewer model parameters. These improvements support broader vision tasks, including image segmentation, classification, pose estimation, and oriented object detection [20].

Its lightweight architecture and enhanced integration for multi-device deployment contexts make it a compelling choice for tie-grade defect detection in rail infrastructure, where real-time precision and hardware constraints are key operational considerations.

B. Real-Time Detection Transformer (RT-DETR)

Transformer architectures were predominantly used for Natural Language Processing (NLP) tasks. However, in recent years, they have been adapted for vision tasks [21]. ViTs, such as the Detection Transformer (DETR), reframe object detection as a set prediction problem using an encoder-decoder architecture. A convolutional backbone first extracts a feature map from the input image. The encoder then divides this map into fixed-size patches, each flattened into an embedding and paired with a positional tag to preserve the spatial structure. Using self-attention, the encoder models relationships across all patches simultaneously. The decoder receives learned object queries and interacts with the encoded features to predict object bounding boxes and class labels directly. These facets enable ViTs to model global spatial dependencies within an image, which is particularly valuable when detecting diverse defect patterns and tie types [19].

RT-DETR builds on these milestones with a series of key architectural changes. First, a hybrid encoder architecture enables multi-scale feature extraction while reducing computational resource demands. Second, object queries employ Intersection over Union (IoU) thresholds to prioritize attention to the most relevant objects in an image, enhancing detection accuracy. Third, RT-DETR's decoder layers can be adjusted, impacting its inference speeds: this scope for modification facilitates practical application. Finally, RT-DETR simplifies the detection pipeline by eliminating the need for NMS post-processing and anchored bounding boxes compared to equivalent YOLO versions, heightening model efficiency and generalization [22]. When benchmarked on the COCO dataset, RT-DETR outperformed state-of-the-art YOLO variants on mean Average Precision (mAP), solidifying its viability for real-time, high-accuracy detection tasks.

IV. Methodology

A. Dataset

Railroad ties are horizontal beams that span the rail ballast, providing structural stability and integrity to the railroad infrastructure. Ties are typically made of wood or reinforced concrete, with recent investments in degradable, composite materials. Each material's structural properties are prone to different types of visual defects. However, due to the lack of reliable images for both concrete and composite ties, this study focused solely on visual defects for wood ties.

To construct the dataset, overhead railroad footage was recorded using a custom-built rig, producing 5-minute video streams. To ensure each frame included a distinct set of ties, averting redundancy during training, a sampling interval of 15 frames was applied, extracting every 15th frame from the MP4 video, ensuring variability and reducing overfitting.

TABLE I
Distribution of Annotated Classes in The Dataset

Class Label	Distribution Count
Wood Check	716
Wood Decay	1,329
Wood Ties	1,779

B. Data Annotation Strategy and Preprocessing

Following the initial frame extraction, 573 images were obtained from the field test footage. Each image was manually inspected for visual clarity and relevance. Frames with high or low exposure, found at the beginning or end of these recordings, were discarded. Other frames with blur or visual obstructions from surrounding vegetation were also excluded. All valid images were then padded with black bars on top and bottom to fit the square dimension requisites for DL models without altering the original aspect ratio of 4:3.

This final dataset, comprising 500 images, is suitable for two primary use cases. The first focused on identifying segments of railroad tracks with missing ties. During the initial manual inspection, 20 images were found to be missing ties. This inclusion, though limited, helps the model learn features of

tracks without ties. For this task, bounding boxes were drawn around each tie, using a predetermined threshold to detect frames with missing ties. The second task detects defective ties from railroad track segments. Annotations were overlaid to capture two predominant defect types in wood ties: decay and checking. Decay results in surface discoloration and undulation, while checking is characterized by surface splits along or against the grain of the wood. Defect annotations were nested inside their corresponding tie bounding boxes, retaining spatial context. Table I summarizes the defect distributions from the 480 frames with ties.

For additional preprocessing, contrast stretching was applied through Roboflow's dashboard, dynamically enhancing contrast and exposure ratios to improve feature visibility. Lastly, the dataset was augmented from 500 to 1,000 samples by applying the following augmentations: horizontal flips, -10 to +10 degree adjustments of image hues, -10 to 10% adjustment of saturation, -5% to 5% adjustment in brightness, and "salt-and-pepper" noise injection in up to 0.1% of pixels in the dataset.

C. Training and Evaluation Protocol

To train both models, first, we selected an appropriate size and version. For both RT-DETR and YOLOv11, we used the Large model configuration, which has comparable parameter counts (32.9 million and 25.3 million, respectively). Conversely, these versions strike a balance between performance and compute resource demands, making them suitable for mobile or embedded device deployment, unlike their XL counterparts.

From here, both models were trained using 5-fold cross-validation, with an 80-20 train-test split. This entails training each model on five distinct partitions of the dataset, ensuring statistical fidelity and reducing variance from any single dataset split.

Training took place on Google Compute Engine (GCE) virtual machines hosted on NVIDIA A100-SXM4 GPUs equipped with 432 tensor cores and 40GB of virtual memory, optimal for processing memory-intensive image datasets. For consistency and experimental fairness, both models used fixed hyperparameter configurations for epoch count, batch size, and optimizer type. To evaluate the efficacy of both YOLOv11 and RT-DETR, we averaged the performance of each fold based on F1 Score, Precision, Recall, and Mean Average Precision (mAP).

V. Results and Discussion

Following training and validation, the box loss, classification (CLS) loss, and distribution focal (DFL) loss were plotted over epochs for both models. Across both sets of graphs, a consistent downward trend indicates that the models fit the training data while generalizing effectively to validation patterns, with strong convergence and minimal overfitting.

Table II demonstrates the comparative analysis between YOLOv11 and RT-DETR for railroad tie defect detection, based on the average F1 score, Precision, Recall, mAP-50, and mAP50-95 following 5-fold cross-validation. YOLOv11 achieved an F1 score of 0.9400, compared to 0.9300 for RT-DETR, indicating it has a more balanced predictive threshold between Precision and Recall for all three classes. Accordingly, we observe that YOLOv11 also outperforms RT-DETR, achieving a Precision score of 0.9696 compared to 0.9498 for the latter. These two metrics indicate that YOLOv11 can more effectively localize tie defects without setting false alarms, enabling a more trustworthy set of predictions. In our proposed deployment context, with presumably limited field resources, a higher Precision score allows maintenance crews to focus on the most critical infrastructure faults rather than responding to false alerts.

TABLE II
Average Performance Metrics of YOLOv11-L and RT-DETR-L Across 5-Fold Cross Validation

Performance Metric	YOLOv11-L	RT-DETR-L
F1 Score (All Classes)	0.9400 ± 0.0089	0.9300 ± 0.0114
Precision	0.9696 ± 0.0077	0.9498 ± 0.0088
Recall	0.9104 ± 0.0147	0.9119 ± 0.0152
mAP50	0.9530 ± 0.0106	0.9321 ± 0.0094
mAP50-95	0.9014 ± 0.0134	0.7898 ± 0.0131

Notably, RT-DETR outperformed YOLOv11 in Recall, with a score of 0.9119 compared to 0.9104 for the latter. This discrepancy, albeit marginal, suggests RT-DETR is more effective at identifying all true positives during inferencing. Although having a higher recall score is indicative of fewer missed detections, this discrepancy alone is insufficient grounds to favor RT-DETR over YOLOv11.

TABLE III
Average Per-Class Detection Performance Across 5-Fold Cross Validation

Class Label	YOLOv11	RT-DETR
Wood Check	0.90	0.92
Wood Decay	0.85	0.876
Wood Ties	0.99	0.992

To further investigate these trade-offs, we compared the confusion matrices of the optimal YOLOv11 and RT-DETR models, as shown in Figures 1 and 2, respectively, to assess per-class performance. Our initial observation was that in both matrices, the diagonals indicating the percentage of correct predictions are identical, with 95% for Wood Check, 89% for Wood Decay, and 99% for Wood Sleeper/Ties. Seeing this, we averaged the performance per class across all five folds to better gauge model performance. These results are shown in Table III.

Despite similarities in core classifications, we notice observable differences in defect misclassifications. In Figure 1, the final column grid illustrates that backgrounds were misclassified as Wood Checks (30%), Decays (59%), and Ties (11%). Conversely, in Figure 2, backgrounds were misclassified as Checks (53%), Decays (44%), and Ties (2%). This indicates that it is better at distinguishing ties from the background than

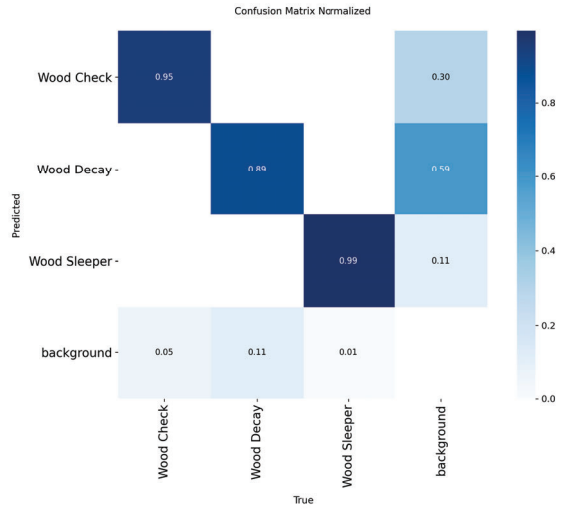

Fig. 1. Normalized confusion matrices for YOLOv11-L on the best-performing folds, showing per-class prediction accuracy across Wood Check, Wood Decay, Wood Tie/Sleeper, and background classes.

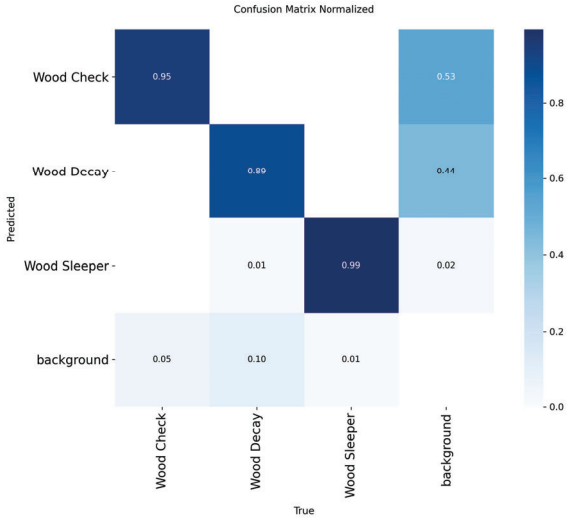

Fig. 2. Normalized confusion matrices for RT-DETR-L on the best-performing folds, showing per-class prediction accuracy across Wood Check, Wood Decay, Wood Tie/Sleeper, and background classes.

YOLOv11, but at the same time, it is worse at discerning checks from the background. It could also explain RT-DETR's Recall score as it identifies more backgrounds as defects, which inflates true positives, as exemplified in Table III, at the expense of recording false positive classifications. On the other hand, YOLOv11 appears to have a more even distribution of false positives, with fewer extremities in its confusion matrices, which aligns with its higher Precision and F1 scores. To complement these findings, Figures 3 and 4 present qualitative predictions from both models across annotated defect classes.

This assessment is further validated when considering how both models performed in terms of Mean Average Precision

Fig. 3. Sample predictions generated by YOLOv11-L across annotated classes (Wood Check, Wood Decay and Wood Tie/Sleeper).

Fig. 4. Sample predictions generated by RT-DETR-L across annotated classes (Wood Check, Wood Decay and Wood Tie/Sleeper).

(mAP) at a standard IoU threshold of 0.5, as well as at varying thresholds, as shown in Table II. At an IoU threshold of 0.5, YOLOv11 achieved an mAP score of 0.9530, compared to 0.9321 for RT-DETR. Similarly, when the IoU thresholds increased from 0.5 to 0.95, YOLOv11 significantly outperformed RT-DETR with an mAP50-95 of 0.9014 compared to 0.7898. This not only indicates YOLOv11's superior defect localization, with a more accurate set of bounding boxes at standard IoU thresholds, but also shows the consistent

quality of predictions even with varying degrees of precision. From these evaluations, YOLOv11's superior performance strengthens its case as the ideal object detection backbone for deployment when configured to run on lightweight mobile platforms or field-level embedded devices.

VI. Conclusion and Future Directions

This research outlined powerful implications of leveraging DL for railroad infrastructure monitoring. First, object detection models effectively identify discrete visual defects and capture spatial relationships between object subclasses, as outlined in the co-occurrences of checks and decays within wood ties. Second, our findings underscore the importance of quality data for reliable predictive power. Initial research directions attempted to integrate more than 10 subclasses (including various tie materials and defects), without consistent imaging data. Pivoting to a subset of these defects significantly narrowed the scope, while also enabling a more comprehensive performance assessment of two contrasting models. Third, although transformer architectures offer real-time object detection capabilities and interpretability advantages, traditional CNN-based detectors, like YOLOv11, remain the benchmark for real-time applications that fulfill speed, accuracy, and ease of deployment constraints.

One limitation is that our field test footage was collected from abandoned rail segments in one geographic region. While this was a practical testbed, it was more defective, and the lack of variability in environmental conditions, such as lighting and weather, limits the generalizability of our results to more frequently serviced railroads.

Future work can extend this methodology to concrete and composite ties and source more representative datasets for defective ties. Other directions include integrating segmentation algorithms as part of two-step methodologies for finer localization or developing a defect severity rating system with railroad governing bodies to generate accessible and actionable, data-driven insights from infrastructure inspections.

Acknowledgment

This research was supported by Marshall University's NSF REU program for Data Analytics. We thank the NSF and the U.S. Army Engineer Research and Development Center for facilitating this work in intelligent transportation systems, as well as program chairs Dr. Haroon Malik and Dr. Yousef Fazea for their guidance and feedback.

References

[1] M. Burrows, C. Burd, and B. Mckenzie, "Commuting by public transportation in the united states: 2019 american community survey reports," 04 2021. [Online]. Available: https://www.census.gov/content/dam/Census/library/publications/2021/acs/acs-48.pdf

[2] Y. Zhao, Z. Liu, D. Yi, X. Yu, X. Sha, L. Li, H. Sun, Z. Zhan, and W. J. Li, "A review on rail defect detection systems based on wireless sensors," *Sensors*, vol. 22, p. 6409, 01 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/17/6409

[3] Y. Wang, Y. Wang, P. Wang, K. Ji, J. Wang, J. Yang, and Y. Shu, "Rail magnetic flux leakage detection and data analysis based on double-track flaw detection vehicle," *Processes*, vol. 11, p. 1024, 03 2023.

[4] W. Gong, M. F. Akbar, G. N. Jawad, M. F. P. Mohamed, and M. N. A. Wahab, "Nondestructive testing technologies for rail inspection: A review," *Coatings*, vol. 12, p. 1790, 11 2022.

[5] Y. Xia, S. W. Han, and H. J. Kwon, "Image generation and recognition for railway surface defect detection," *Sensors*, vol. 23, pp. 4793–4793, 05 2023.

[6] X. Wen, X. Zheng, and Y. He, "Mscm-net: Rail surface defect detection based on a multi-scale cross-modal network," *Computers, Materials & Continua*, vol. 82, pp. 4371–4388, 2025.

[7] S. M. Mirzaei, A. Radmehr, C. Holton, and M. Ahmadian, "In-motion, non-contact detection of ties and ballasts on railroad tracks," *Applied Sciences*, vol. 14, pp. 8804–8804, 09 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/19/8804

[8] J. Sresakoolchai and S. Kaewunruen, "Railway defect detection based on track geometry using supervised and unsupervised machine learning," *Structural Health Monitoring*, vol. 21, p. 147592172110444, 01 2022.

[9] X. Xu, Y. Lei, and F. Yang, "Railway subgrade defect automatic recognition method based on improved faster r-cnn," *Scientific Programming*, vol. 2018, pp. 1–12, 06 2018.

[10] A. Şener, B. Ergen, and M. Toğaçar, "Fault detection from images of railroad lines using the deep learning model built with the tensorflow library," *Turkish Journal of Science and Technology*, 02 2022.

[11] A. Damai, H. Song, H. S. Narman, A. Lambert, and A. Alzarrad, "Enhancing railway safety: A machine learning approach for automated detection of missing track bolts," in *Proceedings of the ASCE International Conference on Computing in Civil Engineering (i3CE)*, New Orleans, LA, May 11–14 2025.

[12] A. D'Arms, H. Song, H. S. Narman, N. C. Yurtcu, P. Zhu, and A. Alzarrad, "Automated railway crack detection using machine learning: Analysis of deep learning approaches," in *Proceedings of the 2024 IEEE 15th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, University of California, Berkeley, CA, Oct. 24–26 2024.

[13] L. Chen, Q. Sun, Z. Han, and F. Zhai, "Dp-yolo: A lightweight real-time detection algorithm for rail fastener defects," *Sensors*, vol. 25, p. 2139, 03 2025.

[14] Y. Wang, K. Zhang, L. Wang, and L. Wu, "An improved yolov8 algorithm for rail surface defect detection," *IEEE Access*, vol. 12, pp. 44 984–44 997, 01 2024.

[15] A. Rivero, S. Radosavljevic, and P. Vanheeghe, "Application of belief theories for railway track defect detection," *International Journal of Automation, Artificial Intelligence and Machine Learning*, vol. 4, pp. 10–35, 06 2024.

[16] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv (Cornell University)*, vol. 1, 06 2014.

[17] Y. Kataoka, T. Matsubara, and K. Uehara, "Image generation using generative adversarial networks and attention mechanism," *Annual ACIS International Conference on Computer and Information Science*, pp. 1–6, 06 2016.

[18] D. Lester, J. Gao, S. Sutphin, P. Zhu, and H. S. Narman, "A yolo-based semi-automated labeling approach to improve fault detection efficiency in railroad videos," in *Proceedings of the 2025 ASEE North Central Section Conference (ASEE NCS)*, Huntington, WV, Mar. 28–29 2025.

[19] A. Phaphuangwittayakul, N. Harnpornchai, F. Ying, and J. Zhang, "Railtrack-davit: A vision transformer-based approach for automated railway track defect detection," *Journal of Imaging*, vol. 10, pp. 192–192, 08 2024.

[20] M. L. Ali and Z. Zhang, "The yolo framework: A comprehensive review of evolution, applications, and benchmarks in object detection," *Computers*, vol. 13, no. 12, 2024. [Online]. Available: https://www.mdpi.com/2073-431X/13/12/336

[21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 213–229.

[22] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," 2024. [Online]. Available: https://arxiv.org/abs/2304.08069