RailNet: A Lightweight Transfer Learning Model for Real-time Rail Component Detection and Defect Segmentation

Jiawei Guo[1], Boshi Chen[1], Yu Qian[2] and Yi Wang[1*]

[1]*Department of Mechanical Engineering, University of South Carolina, Columbia, SC, 29208*

jiaweig@email.sc.edu; boshi@email.sc.edu

** Corresponding Author: yiwang@cec.sc.edu*

[2]*Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC, 29208*

yuqian@sc.edu

ABSTRACT

Rapid railroad inspection is vital to ensuring operational safety, yet conventional methods remain inefficient and inadequate in scope. This paper introduces RailNet, a lightweight, modular transfer learning framework for real-time rail component detection and rail surface defect segmentation on edge devices. RailNet couples a frozen pre-trained detection backbone with a trainable segmentation head featuring two key innovative components: a Context Rebalancing Module (CRM) to mitigate pretrained bias, and Selective Channel Attention (SCA) to help select the relevant features. With only a 5 MB trainable component (0.96 GFLOPs), RailNet achieves 93.2% pixel accuracy and 92.6% recall for defect segmentation, while preserving high detection performance (mAP@0.5 of 98.7%). Evaluated on Nvidia's AGX Orin, RailNet outperforms benchmarks such as YOLOv12-n and MobileSAMv2 in both accuracy and inference speed. These results underscore RailNet's potential as an accurate, real-time, and energy-efficient solution for multi-task railway inspection.

1. INTRODUCTION

According to the Federal Railroad Administration (FRA) safety database, over 400 accidents in 2024 were caused by missing track components. These incidents resulted in losses exceeding $120 million. Therefore, rigorous inspection protocols are essential for detecting flaws in railroad infrastructure and ensuring the safe operation of trains. However, current inspection methods mainly rely on manual procedures, which depend heavily on the expertise of operators. These methods tend to be costly, time-consuming, and prone to human error. As a result, there is a critical need for an automated, real-time, and cost-effective computer vision-based system capable of performing accurate rail track inspections.

Recent advances in deep learning have highlighted the potential of large-scale multi-task frameworks and general-purpose vision models. Advanced architectures such as GPT-4o (Hurst et al., 2024), CLIP (Radford et al., 2021), and SAMv2 (Ravi et al., 2024) have demonstrated strong transfer learning abilities across diverse domains. This success largely arises from extensive pretraining on large and varied datasets. Such pretraining enables these models to generalize well with limited task-specific annotations. Despite their accuracy and versatility, these models require significant computational resources, making deployment in edge-based industrial environments challenging. In real-time railway infrastructure monitoring, where resource efficiency is crucial, it becomes a major obstacle.

Deep learning methods have also gained increasing attention in anomaly detection and structural health monitoring (SHM). For example, Song et al. (2023) proposed a semi-supervised GAN-based framework for auditing energy-consumption anomalies in robotic manipulators. Their model achieved 93% instant-wise detection accuracy by monitoring side-channel signals. Although this shows the potential of GANs under low-data conditions, the approach targets a specific industrial task and cannot be directly applied to visual rail inspection. In the specific domain of railway infrastructure monitoring, lightweight CNN and hybrid models have shown promising results. Ferdousi et al. (2024) proposed an ensemble CNN that combines MobileNetV3 (Howard et al., 2019), VGG-19 (Simonyan & Zisserman, 2014), and ResNet-50 (He et al., 2016) to improve robustness when data is limited. Guo et al. (2023) introduced a lightweight teacher–student model based

on NanoDet, using an adaptively weighted loss function. Their model has a size under 2 MB, requires only 1.52 GFLOPs, and runs inference in less than 14 ms. This model achieves an overall mAP@0.5 of 98.7% on component detection tasks. Similarly, Li et al. (2024) implemented CNN on an FPGA-based edge platform. It achieved 88.9% accuracy in real-time rail defect detection. Bai et al. (2024) further advanced visual inspection with a CNN–Transformer hybrid network. Their model performs pixel-wise segmentation of rail surface defects, achieving precision and recall between 84–87%, and mean intersection-over-union (mIoU) between 77–87%. Furthermore, Wu et al. (2023) proposed a hybrid deep-learning framework that combines classification and segmentation in a single pipeline. This method effectively handles multiple track component types. Building on semantic segmentation approaches, Min et al. (2023) developed an enhanced UPerNet architecture incorporating the Swin Transformer Tiny (Swin-T) as the backbone for semantic segmentation of rail surface defects. Their model achieved pixel accuracies of 91.39% and 93.35%, IoU scores of 83.69% and 87.58%, and Dice coefficients of 91.12% and 93.38% across two datasets. Additionally, Du et al. (2024) developed RSDNet, an improved YOLOv8n-based model with multiscale feature extraction and attention mechanisms, achieving a mAP of 95.4% on the RSDDs dataset. However, most existing models are designed for single-task objectives, focusing exclusively on detection, classification, or segmentation, and seldom integrate multiple tasks—such as component detection and defect segmentation—within a unified framework. Moreover, the majority of these models rely on large pretrained backbones, which limits their efficiency and practicality for deployment on edge devices.

Transfer learning has become a key technique across many fields because it allows the reuse of pretrained models for new tasks, thereby reducing the need for large-labeled datasets and heavy computational resources. It is particularly valuable in domains such as Natural Language Processing (NLP), Computer Vision (CV), and multimodal learning, where data annotation is costly and time-consuming. By leveraging generalized feature representations learned from source domains, transfer learning effectively addresses data scarcity in target domains. Moradi & Groth (2020) provide a detailed taxonomy of transfer learning methods, emphasizing their relevance when failure data is limited or hard to obtain. In the context of SHM, Furlong & Reichard (2023) introduced a hybrid approach that combines physics-based models and data-driven learning, improving generalization by embedding domain knowledge. Additionally, J. Han & Kwon (2024) showed how pretrained diagnostic models can be efficiently adapted across different power plants, even when operational data from new sites is scarce. These studies demonstrate transfer learning's flexibility and robustness in real-world SHM systems, enabling intelligent diagnostics and decision-making in uncertain environments. Beyond these areas, recent advances have shown the effectiveness of transfer learning in railway infrastructure inspection. This domain faces high costs and difficulties in collecting labeled defect data. For example, Ye et al. (2024) proposed a framework that uses pretrained CNNs on general image datasets and fine-tunes them on limited railway images. This significantly improves defect detection accuracy while reducing annotation requirements. Moreover, Zhao et al. (2024) proposed CBAM-SwinT-BL, a Swin Transformer enhanced with block-level attention modules trained via transfer learning. On small-scale rail-surface defect datasets, the model achieves an mAP@0.50 of 0.691 on the MUET dataset and 0.881 on the RIII dataset.

Conventional transfer learning approaches typically optimize for a single downstream task. However, in practical applications where both the original (upstream) and new (downstream) tasks are equally important, such a singular focus can lead to trade-offs that compromise performance on the upstream task.

To meet the dual demands of component detection and defect segmentation in railway monitoring, this paper proposes RailNet, a compact multi-task framework designed for edge deployment. RailNet builds on a frozen pretrained backbone for upstream component recognition. It also introduces a dedicated segmentation module for downstream surface defect identification. This design allows isolated learning, preserving upstream knowledge while enabling efficient adaptation. To improve feature representation, RailNet integrates a Context Refinement Module (CRM) and a Selective Channel Attention (SCA) mechanism. These modules enhance segmentation accuracy without increasing computational load. Furthermore, a novel Single-step Upsample Block speeds up decoding by combining pixel shuffle and transposed convolution. Thanks to this design, RailNet achieves real-time, low-latency inference with high accuracy. It offers a practical solution for resource-constrained railway inspection environments.

The subsequent sections of this paper are organized as follows: Section 2 will present an overview of transfer learning and discuss the existing issue of fine-tuning-based transfer learning. Section 3 will show the details of the proposed RailNet. The results and experimental setup will be shown in Section 4. And Section 5 will give the conclusion.

2. PRELIMINARIES

This section first provides an overview of transfer learning, the topic of the present study. Subsequently, it discusses the key challenges associated with fine-tuning-based transfer learning, particularly in the context of real-world deployment.

2.1. Transfer learning

Transfer learning is a powerful and efficient method, particularly effective when annotated data in the target domain is limited. Transfer learning methodologies can be broadly categorized into two main groups: zero-shot learning (Wang et al., 2019) and fine-tuning (Han et al., 2024).

Zero-shot learning enables pre-trained models to perform downstream tasks without any additional training or fine-tuning. It relies on the general knowledge acquired from large-scale pretraining on diverse datasets to enable inferences for unseen tasks or domains. This approach is particularly useful when the downstream task aligns well with the objectives and data distributions of the source task. However, zero-shot methods often suffer from degraded performance in the presence of significant domain shifts or when handling heterogeneous task types. Their effectiveness heavily depends on how well the pre-trained knowledge generalizes to the new context. As such, while zero-shot learning minimizes the need for labeled data, its applicability is constrained by the semantic gap between source and target domains.

Fine-tuning, by contrast, has emerged as the dominant approach in transfer learning. It is particularly advantageous when the target domain shares feature-level similarities with the source domain but lacks sufficient labeled data. In this method, a pre-trained model, typically trained on a large-scale dataset, is used to initialize the model weights. The model is then further trained on the target dataset with a smaller learning rate, allowing it to retain general feature representations while adapting to specific characteristics of the new task. In practice, lower layers of the network, which capture general features like edges and textures, are often frozen, while higher layers are fine-tuned to learn task-specific representations. In cases where the source and target tasks are significantly different, the entire network may be fine-tuned. Key factors affecting performance include the choice of layers to fine-tune, the learning rate schedule, and the size of the target dataset. Compared to zero-shot learning, fine-tuning offers improved adaptability and often yields superior performance, especially in domains such as fault diagnosis, medical imaging, autonomous navigation, and predictive maintenance. Nevertheless, it requires careful hyperparameter tuning and regularization to mitigate overfitting and avoid catastrophic forgetting, where previously learned knowledge from the source task is overwritten. Despite these challenges, fine-tuning remains a cornerstone of modern transfer learning workflows. It serves as a practical bridge between general-purpose pre-trained models and domain-specific applications, balancing accuracy, computational efficiency, and data requirements.

2.2. Issues with Fine-tuning Based Transfer Learning

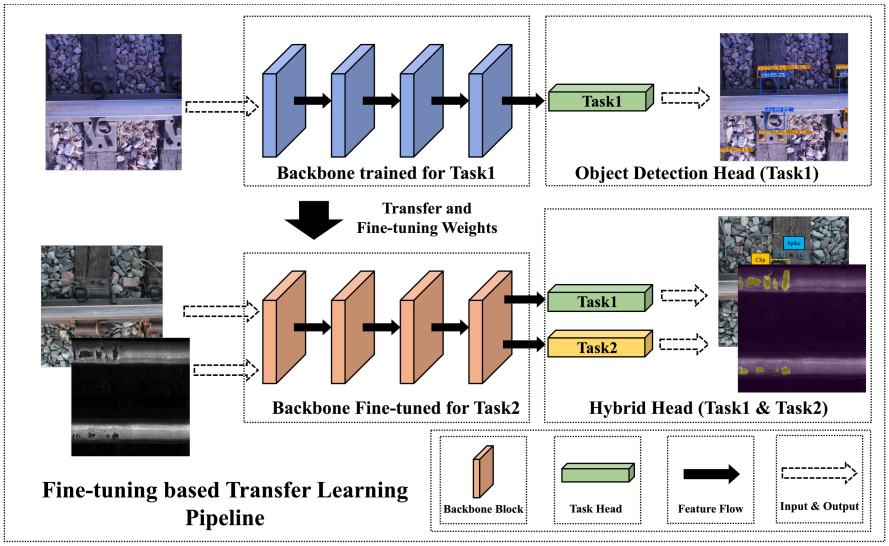

Figure 1. Fine-tuning Based Transfer Learning Pipeline

Fine-tuning-based transfer learning has become a widely used strategy for adapting pre-trained models to new tasks, particularly in data scarce environments. The general workflow is illustrated in Figure **1**. Initially, a base model is trained on a primary task, such as object detection, to recognize key railway components, including clips and spikes along the track. The trained model is then fine-tuned on a secondary task, such as instance segmentation, to identify and segment surface-level defects on the rail.

Despite its effectiveness, several critical challenges arise in fine-tuning-based transfer learning. A primary concern is

catastrophic forgetting, wherein the model's performance on the original task deteriorates significantly after fine-tuning for the new task. This issue becomes more significant in multi-task scenarios. For example, the model must simultaneously perform object detection and instance segmentation. In such cases, the model may struggle to learn both tasks well at the same time, leading to suboptimal performance on one or both tasks.

Another significant limitation is poor model efficiency in real-time or edge-computing environments. Fine-tuned models, especially those derived from large backbone architectures, often require considerable computational and memory resources. This constraint limits their deployment in resource-constrained systems, where lightweight and efficient architectures are desired to ensure low latency and computation-efficient inference.

3. PROPOSED RAILNET

To address the aforementioned limitations in railway inspection, we propose a novel lightweight architecture named RailNet. The design integrates two key components: a Context Rebalancing Module (CRM) to compensate spatial bias inherited from pretrained models, and a Selective Channel Attention (SCA) mechanism to emphasize the most informative feature channels during decoding.

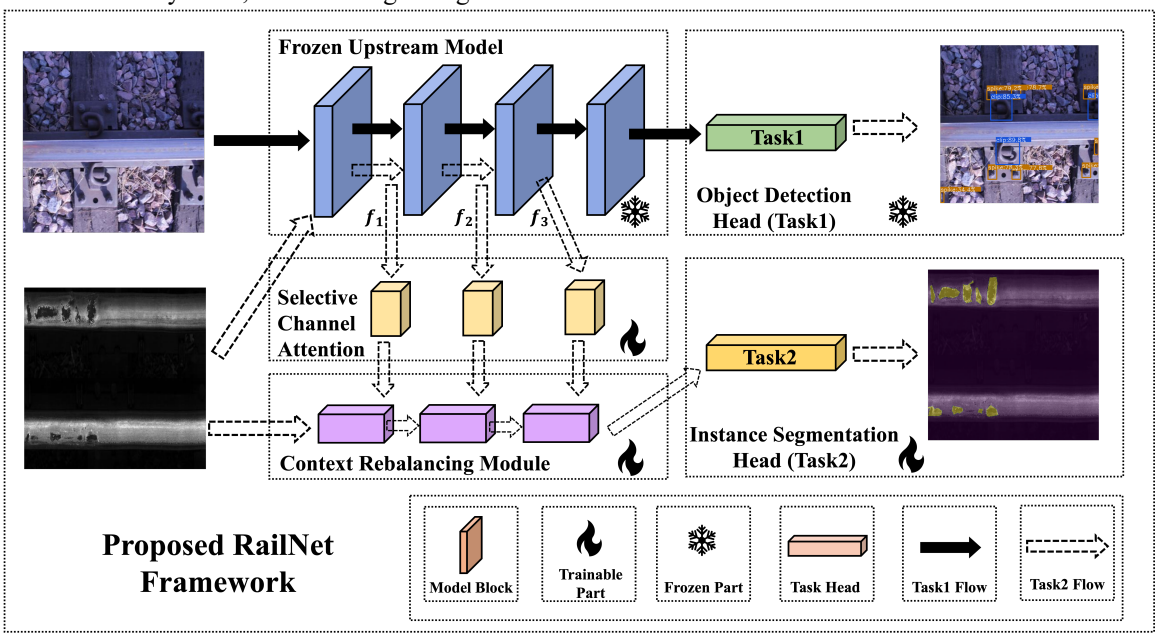

Figure 2. The proposed RailNet Framework

Figure 2 illustrates the proposed RailNet framework designed for multi-task rail inspection. A frozen backbone pretrained on Task 1 and fixed during Task 2 training is shared across both tasks to prevent interference with the original detection capability. Task 1's head remains unchanged, ensuring its performance is preserved. For Task 2, features are extracted from multiple stages of the frozen backbone. However, since these features originate from a task-specific pretraining process, they may not align well with the requirements of the new task.

To address this problem, two key components are introduced: the SCA module filters out irrelevant or less informative channels and retains the most effective features for Task 2. The CRM complements the frozen features by incorporating image-level context from the original input, mitigating potential bias introduced by the frozen backbone. Together, SCA and CRM enhance the quality and task-specificity of the representations used in Task 2.

Each block will be detailed in the following sections.

3.1. Frozen Upstream Model

The upstream model used in RailNet follows a YOLO-like architecture, which is widely recognized for its efficiency and strong multi-task capability in detection and segmentation. Its unified design allows for fast inference and compact feature representation, making it well-suited for real-time railroad inspection scenarios.

Given these advantages, we adopt the YOLOv11 (Rasheed & Zarkoosh, 2024) backbone as our frozen feature extractor. Specifically, we use the C3k2 block proposed in YOLOv11, an enhanced bottleneck module composed of two 3*3 convolutional layers. The C3k2 structure achieves a better

trade-off between accuracy and speed. It enhances feature reuse, supports efficient gradient flow, and improves representation quality across scales, all of which are critical for generating reliable feature maps for downstream defect segmentation. Formally, given an input image from Task 2, denoted as $img_{t2} \in \mathbb{R}^{H*W*C}$, the frozen backbone outputs three feature maps at different stages:

$$f^1, f^2, f^3 = f_{bb}(img_{t2}) \tag{1}$$

Here, $f_{bb}(\cdot)$ represents the frozen backbone, and the output $f^i \in \mathbb{R}^{H_i*W_i*C_i}$, are multi-scale features extracted from different depths. These stage-wise features are used as input to the downstream modules (SCA and CRM) for effective segmentation of rail surface anomalies.

3.2. Selective Channel Attention (SCA)

To identify and retain the most informative channels from the frozen backbone, we introduce a Selective Channel Attention (SCA) module. As shown in Figure 3, the module begins by embedding each intermediate feature map, $f^i \in \mathbb{R}^{H_i*W_i*C_i}$ into a vectorized format following a ViT-like (Dosovitskiy et al., 2020) embedding strategy:

$$f_e^i = ViT_{em}(f^i),\ f_e^i \in \mathbb{R}^{B*P_n^i*D^i} \tag{2}$$

Here, B is the batch size, P_n^i is the number of feature patches, and D^i is the embedding dimension. This embedding allows the module to process spatial context in a patch-wise manner, similar to ViT.

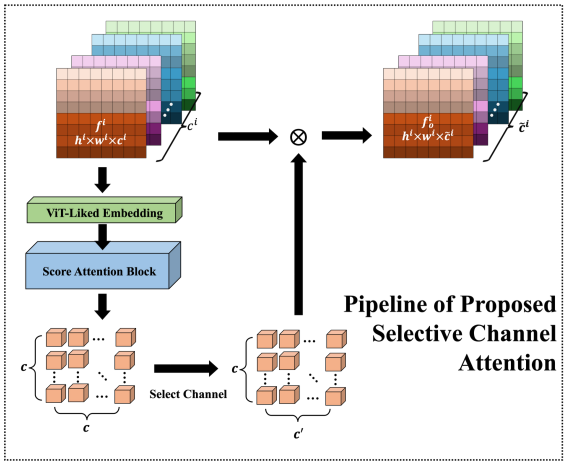

Figure 3. Framework of Selective Channel Attention

Next, the embedded features are passed through our proposed Score Attention mechanism, which determines the importance of each channel. Specifically, we compute key and query matrices as

$$K^i, Q^i = f_e^i * W^{K^i}, f_e^i * W^{Q^i} \tag{3}$$

where $W^{K^i}, W^{Q^i} \in \mathbb{R}^{D^i*D^i}$, are learnable weights, and $K^i, Q^i \in \mathbb{R}^{B*P_n^i*D^i}$ are the key and query matrices. To evaluate attention at the channel level, we reshape both K^i and Q^i into $\mathbb{R}^{B*P_n^i*16*16*C^i}$, where the spatial size $16 * 16$ follows the standard patch size used in ViT, allowing the attention mechanism to reason over channel-wise information within each spatial region. Then, global average pooling is applied across all patches (P_n^i) to obtain a global view of each channel:

$$\overline{K}^i, \overline{Q}^i \in \mathbb{R}^{B*256*C^i} \tag{4}$$

This pooling step effectively increases the receptive field and allows each channel to be evaluated in the context of the full image. We then compute channel-wise attention scores using a sigmoid activation:

$$S = \sigma\left(\overline{Q}^i \overline{K}^{i^T}\right), S \in \mathbb{R}^{C^i*C^i} \tag{5}$$

Unlike softmax, sigmoid is used here because the goal is not multi-class weighting but binary-like importance estimation, i.e., whether each channel is useful or not. From the resulting score matrix, the top $\tilde{C}^i$ channels with the highest aggregated scores are selected, forming a selective score matrix $SS \in \mathbb{R}^{C^i*\tilde{C}^i}$.

Finally, the output feature is reconstructed by projecting the SS matrix

$$f_o^i = SS * f^i, f_o^i \in \mathbb{R}^{h^i*w^i*\tilde{C}^i} \tag{6}$$

This operation ensures that only the most informative channels are retained for downstream processing, while irrelevant or noisy channels are suppressed. The SCA module thus enhances the signal-to-noise ratio of the frozen features and helps the decoder focus on the most relevant information.

3.3. Context Rebalancing Module (CRM)

The CRM is introduced to address potential misalignment between the frozen upstream backbone and the downstream instance segmentation task. Since the backbone is pretrained on a different task (Task 1) and kept frozen during Task 2 training, the extracted features may not fully reflect the semantics required for accurate defect segmentation. CRM serves to inject task-specific spatial cues and adapt the frozen features without modifying the upstream model.

The CRM consists of three sequential C3k2 blocks. The Task 2 input image is first processed by the initial C3k2 block to extract shallow visual features. These features are then concatenated with the first-stage output (f_o^1) from the SCA module. The combined features f_c^1 are passed through a second C3k2 block, followed by another concatenation with the second-stage SCA output f_o^2, yielding f_c^2. And then it is passed through the third C3k2 block, which integrates the final SCA output f_o^3, and generates the final output f_c of CRM.

Through this progressive fusion, CRM gradually rebalances the frozen features with fresh, task-specific information extracted from the original image. The final output f_c of

CRM is then fed directly into the segmentation head. Such enriched representation improves spatial awareness and context alignment, allowing the model to perform accurate instance segmentation while keeping the backbone intact.

3.4. Instance Segmentation Head

For Task 2, we adopt a decoder-based instance segmentation head to convert the refined feature map into a pixel-wise binary mask. The decoder consists of a series of deconvolution (Deconv) blocks that progressively upsample the feature resolution, enabling accurate reconstruction of spatial details.

Compared to native interpolation, the learnable Deconv layers enhance boundary sharpness and segmentation precision, especially in identifying fine-grained surface defects. This head operates on the fused feature map f_c from CRM. The final segmentation output is computed as

$$Mask = De(f_c), Mask \in \mathbb{R}^{H*W*1} \tag{7}$$

where $De(\cdot)$ denotes the deconvolution-based decoder that projects the latent features into the mask space.

4. EXPERIMENTAL SETUP AND RESULTS

All training procedures were executed using PyTorch 2.1.0 on a high-performance workstation equipped with an NVIDIA RTX A6000 GPU (10,752 CUDA cores, 48 GB GDDR6). The batch size was fixed at 16 throughout all experiments. During training, the loss function used was mean squared error (MSE) with the Adam optimizer, which directly measures the pixel-wise discrepancy between predicted and ground truth masks. The learning rate was set to 0.0015. For deployment evaluation, inference was performed on an NVIDIA Jetson AGX Orin module to simulate real-world edge scenarios, with all latency measurements (Task1 + Task2) reported under this hardware setting.

4.1. Dataset Collection

The dataset used in this study was collected using the Track Component Imaging System (TCIS) (*"TCIS," [Online]. Available: Https://www.Ensco.Com/Rail/Track-Component-Imaging-System-Tcis.*, n.d.), a platform designed for rapid railroad safety inspection. The TCIS camera was securely mounted beneath a geometry inspection vehicle to ensure stable and consistent image acquisition during operation. The downward-facing camera maintained a fixed height relative to the rail surface, allowing for uniform coverage of the track bed and component areas. All images were captured at a resolution of 512 × 512 pixels, providing sufficient spatial detail for defect detection and segmentation tasks. A total of 400 images were used in the experiments, with 280 images for training and 120 images set aside for testing, which is sufficient in our study since transfer learning was employed, and therefore, a large-scale dataset was not required.

4.2. Performance Metrics

To evaluate the segmentation performance of the proposed model, several representative performance metrics are employed, including Dice Coefficient, Pixel-wise Precision, Pixel-wise Recall, and Inference Time. These metrics provide a comprehensive view of the model's behavior from both accuracy and efficiency perspectives. The Dice Coefficient is utilized to measure the spatial agreement between the predicted segmentation mask and the ground truth. It is defined as

$$Dice\ Coefficient = \frac{2\left|Y \cap \hat{Y}\right|}{|Y| + \left|\hat{Y}\right|} \tag{8}$$

where $\left|Y \cap \hat{Y}\right|$ represents the number of true positive pixels, $|Y|$ is the number of positive pixels in the ground truth, and $\left|\hat{Y}\right|$ is the number of positive pixels in the predicted mask. This coefficient ranges from 0 to 1, where higher values indicate a better overlap and more accurate segmentation.

To further assess classification performance at the pixel level, Pixel-wise Precision and Pixel-wise Recall are also computed. They are given by Eq. (9) and Eq. (10):

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Here, TP denotes the number of true positive pixels, FP is the number of false positives, and FN is the number of false negatives. Pixel-wise Precision reflects how accurate the model's positive predictions are, while Pixel-wise Recall evaluates its ability to capture all actual positive regions.

In addition to accuracy-related metrics, we report Inference Time to evaluate the model's operational efficiency. Specifically, the total inference time is measured as the sum of durations for both Task 1 and Task 2 on the Jetson AGX Orin platform. This metric reflects the model's suitability for deployment in real-time or embedded systems where computational resources are limited.

4.3. Result

This section presents the experimental results of the proposed RailNet model. We first compare its performance with several state-of-the-art (SOTA) baselines to assess its effectiveness in rail component segmentation. An ablation study is also conducted to analyze the individual contributions of key modules within the RailNet architecture.

4.3.1. Comparison with SOTA

We compare RailNet against several recent segmentation baselines, including YOLOv12-n(Tian et al., 2025), MobileSAMv2(Zhang et al., 2023), DINOv2-S(Oquab et al., 2023), UNet(Ronneberger et al., 2015), and SegFormer(Xie

et al., 2021). These methods represent diverse segmentation paradigms, ranging from lightweight real-time detectors to large-scale transformer-based models. The performance is measured using Dice Coefficient (DC), Precision (P), Recall (R), and Inference Time (IT) (the IT includes both detection and segmentation stages: Task1 + Task2), the Task1 performances for all models is consistent (mAP@ 0.5 of 98.7%).

As shown in Table 1, RailNet achieves the highest DC (0.78), P (66.8%), and R (92.6%), while maintaining a low inference latency of 6.9 ms. These results demonstrate RailNet's superior ability to preserve segmentation shape integrity, accurately localize defects, and operate efficiently in real-time scenarios.

Table 1. Comparison Result

Model	*Dice Coefficient*	*Precision (%)*↑	*Recall (%)*↑	*Inference Time (ms)*↓
RailNet (Proposed)	**0.78**	**66.8**	**92.6**	**6.9**
YOLOv12-n	0.77	66.2	90.2	7.9
MobileSAMv2	0.73	64.3	87.2	322.6
DINOv2-S	0.69	56.8	86.2	121.9
UNet	0.61	52.7	74.3	9.4
SegFormer	0.62	57.4	76.0	11.4

Although YOLOv12-n performs competitively in shape preservation (DC: 0.77) and R (90.2%), its slightly lower P (66.2%) indicates a higher false-positive rate. This performance gap can be attributed to its backbone being pretrained on general datasets, which lack the spatial structures and defect patterns specific to railway imagery.

Transformer-based models such as MobileSAMv2 (DC: 0.73,P: 64.3%) and DINOv2-S (Dice: 0.69, P: 56.8%) show further degradation in segmentation quality while incurring significant inference overhead (322.6 ms and 121.9 ms, respectively). These models, although effective for generic vision tasks, fail to generalize well to structural irregularities commonly seen on rail surfaces due to their lack of domain-specific adaptation.

Traditional encoder–decoder baselines like UNet and SegFormer score lowest across all metrics, further confirming the limitations of solely convolutional designs in modeling complex rail textures and shapes.

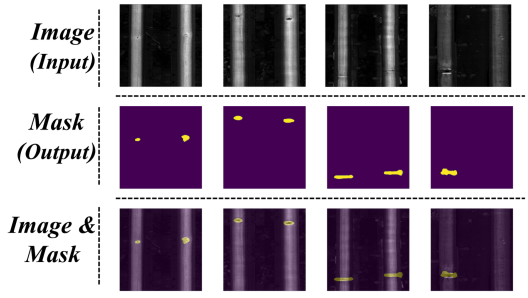

Figure 4. Result Example

Figure 4 presents the visual results of RailNet's segmentation performance on rail surface. The first row shows input images containing various defects on the rail surface. Some of these defects are visually subtle and difficult to distinguish by eye, including small scratch or corrosion spots. The second row displays the corresponding predicted binary masks generated by RailNet. Yellow regions indicate the model's prediction of defect areas. The results show that RailNet is highly sensitive to defects and capable of detecting even very faint or narrow patterns. The third row overlays the predicted masks onto the input images, providing a more intuitive visualization of the model's detection effectiveness. This composite view highlights RailNet's ability to localize surface anomalies while maintaining alignment with the rail geometry.

4.3.2. Ablation Study

To better understand the contribution of each core component in RailNet, we conduct an ablation study by selectively disabling the SCA and CRM. Table 2 reports the performance under four different conditions.

When both SCA and CRM are active, RailNet achieves the best results, with a DC of 0.78, P of 66.8%, and R of 92.6%. Removing CRM leads to a noticeable drop in accuracy, i.e., DC of 0.64, as the model can no longer rebalance frozen features with spatial bias. In this case, the SCA module must merge upstream features through simple interpolation, which limits its effectiveness. Eliminating SCA results in significant performance degradation, since the model receives no explicit guidance from upstream model. Without SCA, the frozen Task 1 backbone cannot meaningfully transfer useful knowledge to Task 2.

When both modules are removed, the model degenerates into a standalone segmentation head trained from scratch, yielding the weakest results (DC of 0.58 and P of 50.4%). This confirms that the feature refinement and task transfer enabled by CRM and SCA are both essential for accurate and robust rail defect segmentation.

Table 2. Ablation Study

Component		*Dice Coefficient (%)*↑	*Precision (%)*↑	*Recall (%)*↑
SCA	CRM			
✓	✓	**0.78**	**66.8**	**92.6**
✓	✗	0.64	59.2	86.3
✗	✓	0.75	63.9	89.7
✗	✗	0.58	50.4	74.8

5. CONCLUSION

This work presents RailNet, a lightweight and modular framework for real-time rail component detection and defect segmentation on edge devices. By combining a frozen detection backbone with a compact segmentation head incorporating CRM and SCA, RailNet achieves accurate multi-task performance with minimal computational overhead. Experimental results on an edge-computing platform demonstrate high accuracy and low-latency inference, validating the model's suitability for on-device railway inspection. Future work will focus on improving robustness under varying operational conditions, extending to additional railway detection tasks.

ACKNOWLEDGEMENT

This research is funded by the Federal Railroad Administration (FRA), Contract No. 693JJ621C000011. Mr. Abe Meddah and Mr. Cameron Stuart from FRA have provided essential guidance and insight during the system development. The opinions expressed in this article are solely those of the authors and do not represent the opinions of the funding agency.

REFERENCES

Bai, S., Yang, L., & Liu, Y. (2024). A vision-based nondestructive detection network for rail surface defects. *Neural Computing and Applications*, *36*(21), 12845–12864.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*.

Du, J., Zhang, R., Gao, R., Nan, L., & Bao, Y. (2024). RSDNet: a new multiscale rail surface defect detection model. *Sensors*, *24*(11), 3579.

Federal Railroad Administration. (2024). *Railroad Equipment Accident/Incident Source Data (Form 54)*.

Ferdousi, R., Laamarti, F., Yang, C., & Saddik, A. El. (2024). A reusable AI-enabled defect detection system for railway using ensembled CNN. *Applied Intelligence*, *54*(20), 9723–9740.

Furlong, T., & Reichard, K. (2023). A Physics-informed, Transfer Learning Approach to Structural Health Monitoring. *Annual Conference of the PHM Society*, *15*(1).

Guo, J., Zhang, S., Qian, Y., & Wang, Y. (2023). A NanoDet Model with Adaptively Weighted Loss for Real-time Railroad Inspection. *Annual Conference of the PHM Society*, *15*(1).

Han, J., & Kwon, D. (2024). Transfer Learning-based Adaptive Diagnosis for Power Plants under Varying Operating Conditions. *PHM Society European Conference*, *8*(1), 6.

Han, Z., Gao, C., Liu, J., Zhang, J., & Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *ArXiv Preprint ArXiv:2403.14608*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., & Vasudevan, V. (2019). Searching for mobilenetv3. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1314–1324.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A. J., Welihinda, A., Hayes, A., & Radford, A. (2024). Gpt-4o system card. *ArXiv Preprint ArXiv:2410.21276*.

Li, J., Fu, Y., Yan, D., Ma, S. L., & Sham, C.-W. (2024). An Edge AI System Based on FPGA Platform for Railway Fault Detection. *2024 IEEE 13th Global Conference on Consumer Electronics (GCCE)*, 1387–1389.

Min, Y., Li, J., & Li, Y. (2023). Rail Surface Defect Detection Based on Improved UPerNet and Connected Component Analysis. *Computers, Materials & Continua*, *77*(1).

Moradi, R., & Groth, K. M. (2020). On the application of transfer learning in prognostics and health management. *ArXiv Preprint ArXiv:2007.01965*.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., & El-Nouby, A. (2023). Dinov2: Learning robust visual features without supervision. *ArXiv Preprint ArXiv:2304.07193*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., &

Clark, J. (2021). Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 8748–8763.

Rasheed, A. F., & Zarkoosh, M. (2024). YOLOv11 Optimization for Efficient Resource Utilization. *ArXiv Preprint ArXiv:2412.14790*.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., & Gustafson, L. (2024). Sam 2: Segment anything in images and videos. *ArXiv Preprint ArXiv:2408.00714*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention--MICCAI*, 234–241.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Song, G., Hong, S. H., Kyzer, T., & Wang, Y. (2023). An energy consumption auditing anomaly detection system of robotic manipulators based on a generative adversarial network. *Annual Conference of the PHM Society*, *15*(1).

"TCIS," [Online]. Available: https://www.ensco.com/rail/track-component-imaging-system-tcis. (n.d.).

Tian, Y., Ye, Q., & Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. *ArXiv Preprint ArXiv:2502.12524*.

Wang, W., Zheng, V. W., Yu, H., & Miao, C. (2019). A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(2), 1–37.

Wu, Y., Chen, P., Qin, Y., Qian, Y., Xu, F., & Jia, L. (2023). Automatic railroad track components inspection using hybrid deep learning framework. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–15.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, *34*, 12077–12090.

Ye, W., Ren, J., Li, C., Liu, W., Zhang, Z., & Lu, C. (2024). Intelligent Detection of Surface Defects in High-Speed Railway Ballastless Track Based on Self-Attention and Transfer Learning. *Structural Control and Health Monitoring*, *2024*(1), 2967927.

Zhang, C., Han, D., Zheng, S., Choi, J., Kim, T.-H., & Hong, C. S. (2023). Mobilesamv2: Faster segment anything to everything. *ArXiv Preprint ArXiv:2312.09579*.

Zhao, J., Yeung, A. W., Ali, M., Lai, S., & Ng, V. T.-Y. (2024). CBAM-SwinT-BL: Small Rail Surface Defect Detection Method Based on Swin Transformer with Block Level CBAM Enhancement. *IEEE Access*.

BIOGRAPHIES

Jiawei Guo earned his B.S. from Tianjin University of Technology and Education (2019), Tianjin, China, and his M.S. from the University of Southern California, CA, USA. He is now pursuing his Ph.D. of mechanical engineering with the University of South Carolina. His research interests are in computer vision and machine learning for engineering applications.

Boshi Chen received his B.S. degree in Biomedical Engineering from Hefei University of Technology, Hefei, China, in 2023. He is currently pursuing a Ph.D. in Mechanical Engineering at the University of South Carolina. His research interests include non-destructive evaluation, robotics, and autonomous systems.

Yu Qian Yu Qian holds dual B.S. degrees from Huazhong University of Science and Technology and Wuhan University (2008), an M.S. from University of Kansas (2009), another M.S. and a Ph.D. from the University of Illinois at Urbana-Champaign (2013, 2014). He is currently an Associate Professor at the University of South Carolina. His research focuses on heavy haul and transit track structures, artificial intelligence, and infrastructure 4.0.

Yi Wang earned his B.S. and M.S. from Shanghai Jiao Tong University (1998, 2000), and his Ph.D. from Carnegie Mellon University (2005). Currently, he is a Professor at the University of South Carolina. His research focuses on computational and data-enabled science and engineering, multi-fidelity surrogate modeling, machine learning, computer vision, and autonomous systems.