

Article

A Direction-Aware Lightweight Network for Camera-Based Underground Mine Track Region Segmentation

Haijun Li, Baolong Ma, Jianjun Gong, Dengyin Jiang, Jie Yang, Kuangang Fan  and Zhichao Chen *

School of Electrical and Energy Engineering, Shanghai Dianji University, Shanghai 201306, China; 25600001010303@st.sdju.edu.cn (H.L.); 25600001010819@st.sdju.edu.cn (B.M.); 241001700214@st.sdju.edu.cn (J.G.); jiangdy@sdju.edu.cn (D.J.); 32239@sdju.edu.cn (J.Y.); kuangangfriend@sdju.edu.cn (K.F.)

* Correspondence: zhichaochen@sdju.edu.cn

Abstract

Accurate localization of the visible track region is essential for perception using front-mounted cameras on underground rail-guided mine vehicles. The task is difficult because the track foreground occupies only a small image area, and its boundary appearance changes with illumination, water, dust, and scene clutter. This study formulates local perception of the track corridor as binary semantic segmentation of the surface bounded by the two visible rails. RailDLA is a lightweight encoder–decoder network. It combines track context preconditioning, RDLA directional strip propagation, context-guided feature fusion, and track axis proxy decoding. On a self-constructed dataset of underground mine vehicle imagery, RailDLA achieves 96.50% mIoU, 92.10% track IoU, 97.50% track accuracy, and 99.70% pixel accuracy. On the working split, its track IoU exceeds those of FastSCNN, PIDNet-S, DDRNet-23-slim, and SegNeXt-S by absolute margins of 6.06, 1.70, 1.56, and 0.37 percentage points, respectively. Under the unified runtime protocol, RailDLA reaches 120.00 FPS on an NVIDIA GeForce RTX 3070 Laptop GPU. These results demonstrate accurate, real-time inference for underground mine vehicle perception.

Keywords: underground mine; rail-guided vehicle; track region segmentation; local track perception; lightweight neural network; directional strip context; semantic segmentation

1. Introduction

Underground railways are widely used in mines for material transport, equipment movement, and inspection patrols. Reliable perception of the visible track region ahead of a rail-guided vehicle is therefore important for navigation assistance, inspection, and local operational awareness based on front-camera imagery in confined mine tunnels. Recent studies on mine automation, underground sensing, and robotic inspection have emphasized that perception reliability is a key requirement for reducing operational risk in poorly illuminated and communication-limited mining environments [1,2]. Research on industrial intelligent diagnosis also shows that incorporating physical constraints and multimodal information can improve model robustness under imbalanced or under-sampled operating data [3]. Recent video surveillance studies have also used transfer learning and spatiotemporal visual cues for abnormal activity detection, further supporting the need for reliable visual monitoring in constrained scenes [4]. This study focuses on image-based segmentation of the local track region. Given an image from a front camera, the model predicts a binary mask of the visible track region bounded by the rails.



Academic Editors: Wolfgang Kainz and Dev Raj Paudyal

Received: 9 May 2026

Revised: 27 July 2026

Accepted: 2 August 2026

Published: 4 August 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the International Society for

Photogrammetry and Remote Sensing.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Perception of underground track regions from camera images differs from general object parsing. The rail-bounded track region occupies a small fraction of each image and forms a narrow, elongated perspective structure, while dust, exposed sleeper ends outside the rails, cables, water stains, equipment occlusion, specular reflections, and nonuniform illumination can disturb the local visual evidence. In the target application, a front-mounted camera observes the track area ahead during routine transport and inspection. As illustrated in Figure 1, this perception setting requires reliable local estimation of track position and extent from weak visual evidence while remaining efficient enough for real-time GPU inference.

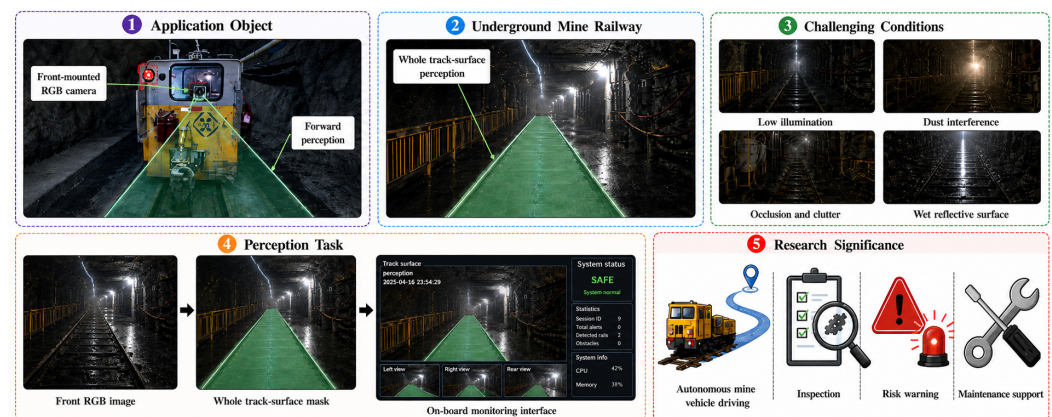


Figure 1. Application background for underground mine track region perception from a front camera under confined tunnel imaging conditions.

Semantic segmentation provides a direct formulation for this perception task: given an underground image, the model predicts a pixel-level mask of the visible track surface bounded by the rails. The foreground track region is defined as the surface between the outer edges of the left and right visible rails. It includes the rail bodies and the interior surface between them, while exposed sleepers, ballast, mud, and tunnel floor regions outside the rails are treated as background. The predicted mask represents a local perception output for the front camera view.

General dense prediction networks, including fully convolutional and encoder–decoder models, provide the basic technical foundation for this formulation [5–7]. Domain-specific visual learning has also been applied to UAV-assisted monitoring, multi-task object detection, and ecological detection from acoustic video [8–10]. Above-ground railway track region segmentation has shown that lightweight encoder–decoder networks can support real-time railway safety applications [11]. Directly transferring these generic or above-ground designs to underground mine images is still difficult because the operational target is a sparse foreground structure whose local integrity matters more than accuracy on the dominant background class. A fragmented track mask may preserve a high pixel accuracy score, but it can degrade subsequent track axis extraction or inspection modules.

The mine track region is strongly directional in the image plane. Under perspective projection, the visible rail-bounded track surface forms an elongated region delimited by the rail outer edges, while exposed sleeper ends and surrounding floor regions remain background. This geometry motivates the use of directional context on the image feature grid. Purely local convolution can preserve edges but may miss long-range evidence along the track region. Unrestricted self-attention can model global interaction, but it introduces the quadratic affinity cost of dense pairwise comparison [12,13]. One effective way to introduce this prior is to use low-cost horizontal and vertical strip context rather than a dense pairwise attention map.

To address this problem, RailDLA is proposed as a lightweight encoder–decoder for the track task. Its RDLA block aggregates context through horizontal and vertical strip propagation on the image feature grid. The term “direction” refers to image plane strip directions derived from the elongated geometry of the visible track region. The two strip responses are introduced through gated residual injection to provide structured context at low cost. The complete network contains compact spatial–semantic feature extraction, track context preconditioning, RDLA, context-guided feature fusion, and a track axis proxy decoder. Under the working validation protocol, the resulting model achieves 96.50% mIoU, 92.10% track IoU, and 120.00 FPS in the runtime test on the same GPU. On the working split, RailDLA exceeds SegNeXt-S, the strongest generic reference in terms of validation accuracy, by 0.80 percentage points in best mIoU and 0.37 percentage points in track IoU.

The main contributions of this work are summarized as follows:

- An underground mine track region segmentation task is formulated for front-camera imagery from rail-guided mine vehicles, with the foreground defined as the visible rail-bounded track surface in the local camera view.
- RailDLA is developed as a lightweight encoder–decoder for this task, combining a track context preconditioning block, an RDLA block, context-guided feature fusion, and a track axis proxy decoding branch for sparse, elongated foreground segmentation.
- Comparative accuracy and efficiency experiments are conducted on a self-constructed dataset of underground mine vehicle imagery; track region metrics and controlled runtime measurements are reported using the same GPU.

2. Related Work

The related work is organized around computer vision and underground perception studies most directly connected to the proposed image-based segmentation task. The task is related to underground infrastructure monitoring, while the benchmark used in this study contains images from a front camera and pixel-level masks of the track region without georeferenced coordinates, topological corridor annotations, or spatial database metadata. Accordingly, RailDLA is positioned as a local visual perception model that may provide inputs for broader mine mapping or decision support systems.

2.1. Railway and Underground Track Perception

Track perception has been studied in the broader context of railway safety, infrastructure monitoring, and automated inspection. Recent work has extended this line to catenary support component detection with graph-guided and multireceptive-field Mamba models, prompt-based component anomaly detection, and machine-vision-driven physics-informed state estimation for the pantograph–catenary system [14–17], and meta-learning-based dynamic modeling of high-speed rail subsystems [18]. However, these studies mostly focus on above-ground railway inspection, component-level defect recognition, or image classification rather than pixel-level segmentation of the rail-bounded track surface from a front-mounted vehicle camera. Recent research on illumination-aware multispectral detection shows that adverse illumination can degrade RGB perception and that explicit illumination modeling with cross-modal alignment can improve visual robustness [19]. This evidence supports the need to consider poor illumination in underground visual perception, including settings that use a monocular front camera. Recent railway environment perception studies have further extended visual learning to sparse-to-dense fusion for three-dimensional object detection [20] and to lightweight LiDAR-based 3-D object detection for railway transportation [21]. Visual–inertial SLAM based on line features further shows that structured visual lines can provide useful cues for real-time navigation using cameras [22].

Underground mine perception introduces a different set of constraints. Mine automation, underground IoT, and mobile inspection robot studies identify perception reliability as a key requirement for safe operation in confined, poorly illuminated, and communication-limited environments [1,2]. For mine vehicles with a front camera, the perception target is the visible rail-bounded track surface ahead of the vehicle. The interior surface between the rails may be contaminated by mud or water stains, while exposed sleeper ends, cables, dust, equipment, and shadows around the rails create difficult background clutter. Accordingly, the perception problem is formulated as binary segmentation of the rail-bounded track surface, distinct from rail line extraction, segmentation of sleepers or ballast outside the rails, and defect classification.

2.2. Track Region Segmentation and Dense Prediction

Semantic segmentation provides a direct way to obtain a pixel-level mask of the track region. Classical dense prediction networks such as FCN and U-Net established end-to-end pixel prediction and skip-based spatial recovery [5,6]. DeepLabv3+ further showed that an encoder–decoder structure can combine contextual aggregation with sharper object boundaries [7]. These principles are useful for track region segmentation, where the model must recover both foreground structure and local boundaries. Segmentation systems designed for specific domains, such as mammogram analysis based on adaptive Trans-ResUnet++, also illustrate how encoder–decoder designs can be adapted to domain-dependent boundary recovery [23].

Railway-specific segmentation has also been investigated. Chen et al. [11] proposed an efficient railway track region segmentation network for real-time railway safety applications, combining a lightweight encoder with a cross-fusion decoder and additional training strategies. Their work demonstrates that semantic segmentation can support track region understanding while balancing accuracy and speed. The present task differs in both environment and foreground geometry. Underground mine rail-bounded track surfaces occupy a small and elongated area, are affected by low illumination and contamination, and require a coherent response across the surface between the two rail outer edges, including both the rail bodies and the interior surface. Therefore, RailDLA introduces track axis proxy modulation, local geometry supervision, and directional context to address fragmented foreground masks.

2.3. Real-Time Semantic Segmentation

Real-time semantic segmentation aims to maintain dense prediction accuracy under strict inference constraints. FastSCNN [24] is a representative lightweight model that separates rapid spatial downsampling from semantic extraction at low resolution. The structure is attractive for edge or onboard platforms because it uses few parameters and a low FLOP count, while the original architecture remains generic and lacks explicit geometry modeling for the track region. Lightweight perception architectures have also been investigated for railway catenary foreign object detection, where real-time sensing is important for railway safety monitoring [25]. Video surveillance recognition and tracking studies similarly emphasize compact feature extraction and temporal association for efficient scene understanding [26]. Efficient vehicle detection in traffic scenes has also been explored with improved single-stage detectors, where lightweight backbones, attention-based feature weighting, and cross-level feature fusion improve the speed–accuracy trade-off [27].

Another line of efficient segmentation models preserves fine spatial detail through multi-resolution or multi-branch representations. BiSeNet V2 [28] separates detail and semantic branches and fuses them through guided aggregation. DDRNet [29] maintains dual-resolution streams for real-time segmentation, while PIDNet [30] introduces parsing,

detail, and boundary branches inspired by PID control. These studies indicate that efficient segmentation requires reduced computation together with preservation of boundary and thin-region information. The proposed model follows this principle and adapts it to underground track region segmentation by keeping spatial detail in a shallow branch while placing context modules for the track task in the semantic pathway at low resolution. Dedicated railway perception datasets have also supported visual recognition research in related safety scenarios, such as foreign object detection on railroad transmission lines [31]. These dataset construction efforts provide useful references for railway visual perception and data organization.

2.4. Convolutional Attention, Transformers, and Linear Context

Convolutional attention and Transformer-style attention provide different routes for enlarging the receptive field. Channel and spatial attention mechanisms, including squeeze-and-excitation and CBAM, recalibrate feature responses with modest overhead [32,33]. Cross-modal attention has also been used in infrared-visible pedestrian detection to combine complementary thermal and texture cues [34]. Multimodal attention has also been explored in medical image fusion with modified Swin cross-attention [35]. Large-kernel and criss-cross context designs further show that structured long-range interaction can improve semantic segmentation without requiring unrestricted global attention [36,37]. SegNeXt [38] follows this direction and shows that carefully designed convolutional attention can achieve high segmentation accuracy. In the full benchmark of this study, SegNeXt-S is the strongest generic reference in terms of validation accuracy, reaching 95.70% mIoU. Because this study targets a compact model for the track task, SegNeXt-S is interpreted as a larger generic reference rather than a direct lightweight competitor.

Transformer models have also become important segmentation references. Self-attention was originally introduced as a general sequence modeling mechanism [12]; ViT extended Transformer tokenization to images [39]. Dense prediction models such as SETR, Segmenter, and SegFormer then adapted Transformer representations to semantic segmentation [40–42]. Although these mechanisms are expressive, dense self-attention constructs a global interaction matrix whose memory and computation costs can be excessive for high-resolution prediction. Linear attention variants reduce this cost by rearranging the computation or using kernelized feature maps [43]. The strip attention module in RailDLA is aligned with the motivation for efficient context modeling but introduces a stronger task constraint: context is propagated through horizontal and vertical strips on the image feature grid rather than through arbitrary global interaction. The design uses the elongated structure of the target class as an inductive bias and avoids the cost of dense pairwise affinity.

The above studies provide useful components for visual track understanding, dense prediction, efficient inference, and contextual feature modeling. In broader industrial sensing, heterogeneous multimodal fusion has also been used in rotating-machinery fault diagnosis to improve interpretability and robustness under complex industrial signals [44]. Few existing studies jointly address rail-bounded track region segmentation, underground mine visual degradation, and lightweight inference from images captured by front cameras.

3. Method

3.1. Problem Formulation and Track Structural Prior

Let $I \in \mathbb{R}^{3 \times H \times W}$ denote an underground mine image and $Y \in \{0, 1\}^{H \times W}$ denote the corresponding binary mask of the rail-bounded track region. The foreground is the surface between the outer edges of the two visible steel rails; exposed sleepers and ground regions outside the rails are treated as background. The proposed model f_θ produces two-class

segmentation logits $Z_s = f_\theta(I) \in \mathbb{R}^{2 \times H \times W}$, and the foreground probability is the softmax probability of class 1, denoted by P . With τ denoting the decision threshold, the supervised learning objective is formulated as

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(I,Y) \sim \mathcal{D}_{\text{mine}}} [\mathcal{L}(f_\theta(I), Y)], \quad P_{ij} = \text{softmax}(Z_s)_{1ij}, \quad \hat{Y}_{ij} = \mathbf{1}(P_{ij} \geq \tau). \quad (1)$$

The validation protocol uses $\tau = 0.5$ unless otherwise specified. The task has two dominant properties: the positive class is sparse, and its visible region is geometrically elongated. As a lightweight structural prior, the context support around a spatial location $p = (i, j)$ is restricted to horizontal and vertical strip neighborhoods on the image feature grid:

$$\begin{aligned} \Omega_r(p) &= \{p + t\mathbf{e}_h : |t| \leq r\} \cup \{p + t\mathbf{e}_v : |t| \leq r\}, \\ \mathbf{e}_h &= (0, 1), \quad \mathbf{e}_v = (1, 0), \quad k = 2r + 1. \end{aligned} \quad (2)$$

The strip support on the image grid provides an intermediate form of context between dense global interaction and purely local convolution: it propagates evidence along two long axes while avoiding dense pairwise affinity. This design follows structured context modules in dense prediction, where long-range evidence is introduced beyond small convolutional kernels [36,37]. The support is restricted to two orthogonal directional spans that often connect evidence inside the elongated rail-bounded track surface while keeping the operator inexpensive.

3.2. Network Overview

The overall architecture follows a lightweight encoder–decoder design, which is commonly used to recover spatial detail after semantic feature extraction [6,7]. It comprises a shallow spatial branch, a semantic branch, a track context preconditioning block, the proposed RDLA block, a context-guided feature fusion module, and a track axis proxy decoding branch. RDLA models direction-aware context through horizontal and vertical strip propagation on the image feature grid. The strip support aggregates context for elongated track foregrounds with linear spatial cost. Figure 2 gives the overall data flow, showing how shallow spatial detail, lower-resolution directional context, and track axis proxy decoding are connected for final segmentation. The forward computation is summarized as

$$\begin{aligned} F_s &= \mathcal{D}(I), \quad F_g = \mathcal{G}(F_s), \quad F_r = \mathcal{B}_{\text{track}}(F_g), \quad \tilde{F}_g = \mathcal{R}_{\text{rdla}}(F_r), \\ F_f &= \phi_s(F_s) + \phi_g(\text{up}(\tilde{F}_g)) + \phi_{cg}(F_s, \text{up}(\tilde{F}_g)). \end{aligned} \quad (3)$$

Here, $\mathcal{D}$ denotes the learning-to-downsample branch, $\mathcal{G}$ is the semantic extractor, $\mathcal{B}_{\text{track}}$ prepares track-oriented context, $\mathcal{R}_{\text{rdla}}$ denotes the RDLA directional strip attention operation, and F_f is the fused decoder feature. Following real-time segmentation networks that separate detail preservation from semantic abstraction [24,28,30], the operations specific to the track task are placed in the semantic pathway at low resolution so that longer context can be introduced without dominating inference cost, while the shallow branch preserves spatial detail for boundary localization. The figure also introduces the two prediction heads used throughout the method: the rail-bounded segmentation mask Z_s and the auxiliary track axis proxy cue Z_c .

Table 1 gives the network structure in a compact notation similar to MobileNetV2 [45]. The symbols t , c , n , and s denote the expansion ratio, output channels, number of repetitions, and stride, respectively; for upsampling, s denotes the scale factor. For modules specific to the task, the table reports the module name and the number of channels in its main output.

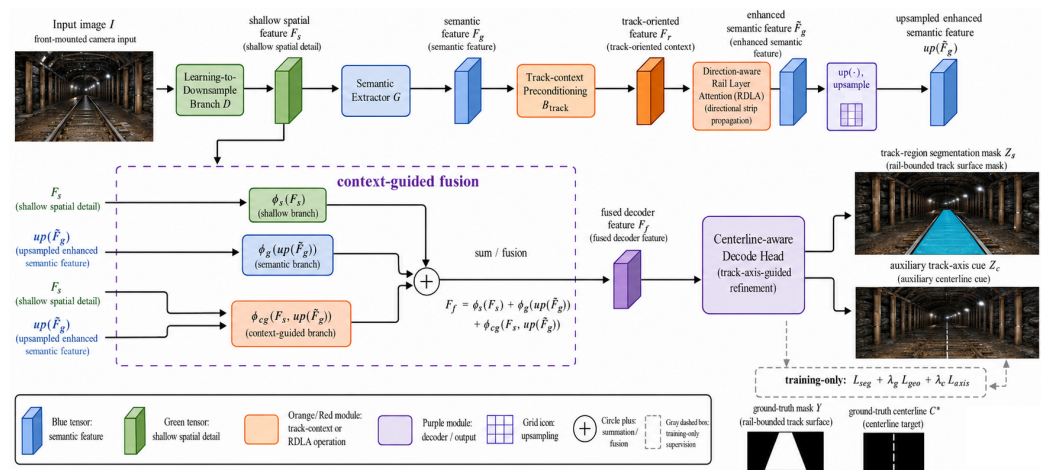


Figure 2. Overview of the RailDLA lightweight encoder–decoder for underground mine track region segmentation. C^* denotes the ground-truth centerline target used for the auxiliary track-axis supervision.

Table 1. Network layer configuration of RailDLA. DSConv denotes depthwise separable convolution, and IRB denotes a MobileNetV2 inverted-residual block [45].

Input Size	Operator	t	c	n	s
$512^2 \times 3$	Conv 3×3	—	32	1	2
$256^2 \times 32$	DSConv 3×3	—	48	1	2
$128^2 \times 48$	DSConv 3×3	—	64	1	2
$64^2 \times 64$	IRB	6	64	3	2
$32^2 \times 64$	IRB	6	96	3	2
$16^2 \times 96$	IRB	6	128	3	1
$16^2 \times 128$	Conv 3×3	—	128	1	1
$16^2 \times 128$	Track context preconditioning block	—	128	1	1
$16^2 \times 128$	RDLA block	—	128	1	1
$16^2 \times 128$	Bilinear upsampling	—	128	1	$\times 4$
$64^2 \times 128/64^2 \times 64$	Context-guided feature fusion	—	128	1	1
$64^2 \times 128$	Decoder DSConv 3×3	—	128	2	1
$64^2 \times 128$	Track axis proxy decoding branch	—	64	1	1
$64^2 \times 128$	Final segmentation head	—	2	1	1

3.3. Track Context Preconditioning

The track context preconditioning block is inserted before the strip attention block. It enriches the semantic feature with local, directional, and mildly dilated cues from the rail-bounded surface. Figure 3 details this preconditioning stage. For $X \in \mathbb{R}^{C \times h \times w}$, the track context preconditioner in Figure 3 is

$$\mathcal{B}_{track}(X) = X + \Gamma_b(U) \odot U, \quad \Gamma_b(U) = \sigma(W_b \text{ GAP}(U)), \quad (4)$$

$$U = \phi_b([D_3(X); S_h^{kr}(X); S_v^{kr}(X); D_{3,d}(X)]).$$

Here, D_3 is a depthwise 3×3 operator, S_h^{kr} and S_v^{kr} are horizontal and vertical strip operators, $D_{3,d}$ is a dilated depthwise 3×3 operator, and ϕ_b is a pointwise projection. The gate Γ_b makes the residual update channel-adaptive, following the general principle of channel recalibration in attention modules [32,33]. The gate prevents the directional prior from being injected uniformly into all semantic channels. In the proposed architecture, $k_r = 11$ and $d = 2$. The gate is learned end-to-end through the segmentation losses. It is computed from global average pooled responses and a learned pointwise channel projection

followed by a sigmoid activation. Therefore, channels that respond to elongated rail-edge or track-interior evidence can receive a larger residual update from the preconditioned feature U , whereas channels dominated by local texture clutter, water stains, or equipment shadows can receive a smaller update. No manually specified channel weights are used.

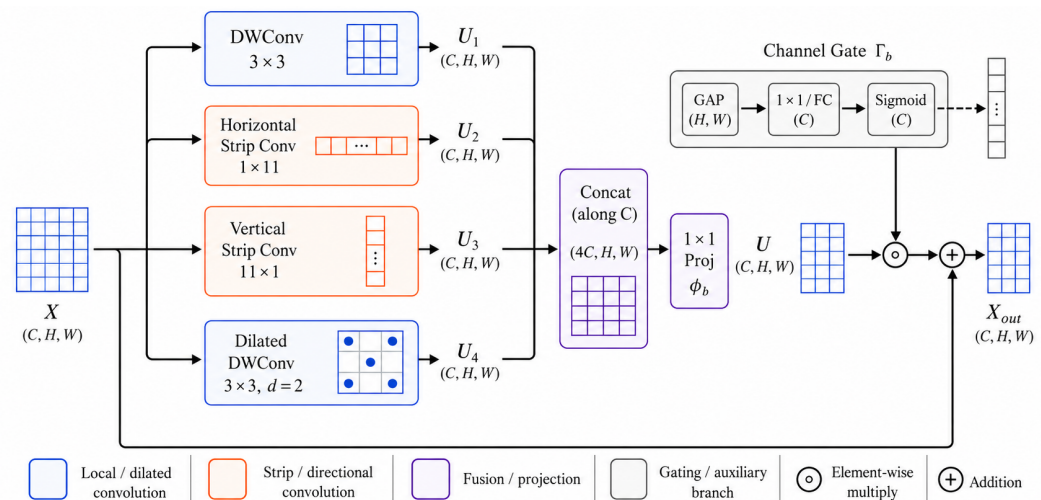


Figure 3. Track context preconditioning block with local, strip, and dilated depthwise responses followed by gated residual injection.

3.4. Direction-Aware Rail Linear Attention

After preconditioning, RDLA provides low-cost directional context for track region features without constructing a dense pairwise attention matrix. As shown in Figure 4, RDLA is inspired by attention, but it performs structured propagation. Its direction-aware context is implemented by two strip axes on the image feature grid. The horizontal response collects evidence along image rows, and the vertical response collects evidence along image columns. These two directions reflect the elongated perspective geometry of the rail-bounded track region. Compared with dense self-attention, RDLA restricts the interaction support from data-dependent pairwise query–key affinity to learned horizontal and vertical strip propagation. Linear attention variants such as Performer and Linear Transformer reduce the cost of global attention through kernel feature maps or recurrent computation. RDLA follows a different design: it fixes the spatial support to two orthogonal directions on the image grid and learns strip weights for each channel and relative offset. The two directional strip responses are generated, projected, and injected through a gated residual path. For each channel c , RDLA first applies two learnable strip operators:

$$\begin{aligned} [T_h(X)]_{c,i,j} &= \sum_{t=-r}^r a_{c,t}^{(h)} X_{c,i,j+t}, \\ [T_v(X)]_{c,i,j} &= \sum_{t=-r}^r a_{c,t}^{(v)} X_{c,i+t,j}. \end{aligned} \quad (5)$$

These operators can be viewed as channel-wise banded Toeplitz maps over the flattened spatial grid. Their nonzero entries are restricted to the support defined in Equation (2), which gives RDLA a linear spatial cost with respect to image area. The formulation is consistent with efficient attention research, which reduces the quadratic cost of dense self-attention by changing the interaction structure or computation order [43]. In implementation, $a_{c,t}^{(h)}$ and $a_{c,t}^{(v)}$ are the learned weights of depthwise horizontal and vertical strip kernels. They are optimized by back-propagation together with the remaining network parameters. The weights are specific to each channel and relative offset. They are shared

over spatial locations, allowing RDLA to produce adaptive directional responses while preserving a fixed strip-shaped structural bias.

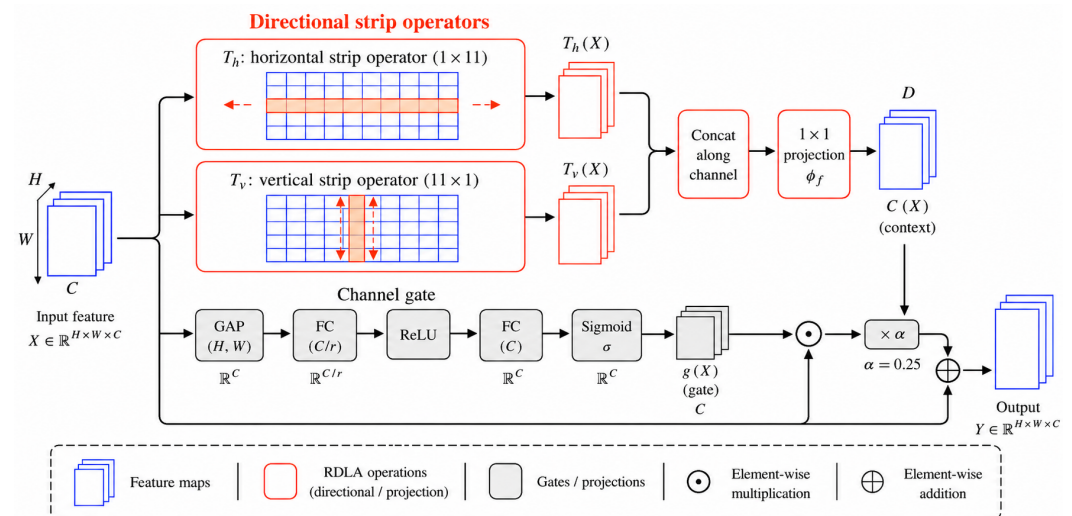


Figure 4. RDLA block with horizontal and vertical strip operators and gated residual injection for directional context aggregation. Arrows indicate the data flow through the strip operators, channel gate, and gated residual injection.

The two directional responses are then mixed and injected through a gated residual update:

$$\begin{aligned} C(X) &= \phi_f([T_h(X); T_v(X)]), \\ g(X) &= \sigma(W_2 \delta(W_1 \text{GAP}(X))), \quad \mathcal{R}_{rdla}(X) = X + \alpha g(X) \odot C(X). \end{aligned} \quad (6)$$

The projection ϕ_f performs dense channel mixing after directional spatial aggregation, while $g(X) \in (0, 1)^C$ controls which channels should receive the directional residual. For channel c , the injected directional update is $\alpha g_c(X) C_c(X)$. A larger $g_c(X)$ increases the contribution of the horizontal–vertical strip response for that channel, whereas a smaller $g_c(X)$ suppresses the directional residual when the channel response is less consistent with the learned global context. Thus, the gate provides a channel-wise mechanism for injecting the directional prior to different degrees without manually assigning channel weights. The final RDLA block uses a strip kernel size of $k = 11$ and a residual scale of $\alpha = 0.25$.

3.5. Structured Affinity and Stability

To clarify why this design differs from local convolution and dense attention, Figure 5 gives a conceptual comparison of receptive support. The proposed operator can be interpreted as a learned structured directional operator with strip support on the image grid. Standard self-attention and non-local operations permit all pairwise spatial interactions over $N = hw$ positions [12,13]. RDLA instead parameterizes only the horizontal and vertical strip interactions in $\Omega_r(p)$:

$$\begin{aligned} A^{dense} &= \text{softmax}(QK^\top / \sqrt{d_a}), \\ A_{pq,c}^{strip} &= \sum_{t=-r}^r a_{c,t}^{(h)} \mathbf{1}(q = p + t\mathbf{e}_h) + \sum_{t=-r}^r a_{c,t}^{(v)} \mathbf{1}(q = p + t\mathbf{e}_v). \end{aligned} \quad (7)$$

Equation (7) provides a conceptual comparison rather than an implementation step: neither A^{dense} nor A^{strip} is explicitly materialized during inference. The comparison clarifies the design constraint. RDLA uses learnable strip weights for relative offsets in place of unrestricted, data-dependent spatial affinity, yielding a structured propagation mechanism

for efficient track region representation. The distinction from generic linear attention mechanisms lies in the interaction structure. Performer-type attention reduces the cost of global attention by approximating the softmax kernel, and Linear Transformer variants rearrange the attention computation to avoid materializing a quadratic affinity matrix. RDLA instead uses fixed horizontal and vertical strip support with learned weights specific to each channel and relative offset, yielding task-specific structured propagation for the rail-bounded foreground.

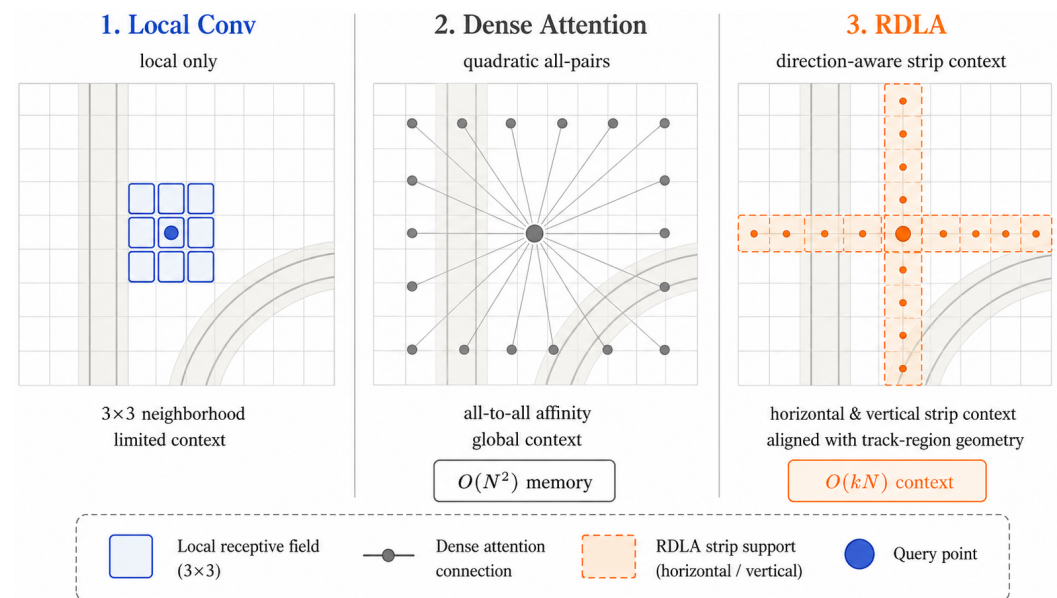


Figure 5. Conceptual comparison of local convolution, dense pairwise attention, and strip attention for elongated track context modeling.

The residual formulation also provides an interpretive bound for the perturbation introduced by RDLA. If the spectral norms of T_h , T_v , and ϕ_f are bounded by ρ_h , ρ_v , and ρ_f , respectively, then $0 < g_c(X) < 1$ gives

$$\begin{aligned} \|\mathcal{R}_{rdla}(X) - X\|_F &\leq \alpha \rho_f (\rho_h + \rho_v) \|X\|_F, \\ \|\mathcal{R}_{rdla}(X)\|_F &\leq \left[1 + \alpha \rho_f (\rho_h + \rho_v)\right] \|X\|_F. \end{aligned} \quad (8)$$

The bound indicates that RDLA modifies the semantic representation through a controlled residual perturbation under bounded operator norms. It is an interpretive property of the residual form rather than an explicit spectral regularization constraint. Such control is important in cluttered scenes, where the directional prior should support local appearance evidence rather than override it.

The corresponding computational and memory complexities are contrasted with dense attention as follows:

$$\begin{aligned} \mathcal{C}_{rdla} &= \mathcal{O}(CkN + C^2N + C^2/r_g), & \mathcal{M}_{rdla} &= \mathcal{O}(CN), \\ \mathcal{C}_{dense} &= \mathcal{O}(N^2d_a + Nd_aC), & \mathcal{M}_{dense} &= \mathcal{O}(N^2). \end{aligned} \quad (9)$$

Here, CkN corresponds to the two depthwise strip operators up to a constant factor, C^2N to the pointwise projection used for channel mixing, and C^2/r_g to the lightweight channel gate with reduction ratio r_g . Exact constants can vary with kernel implementation and operator fusion. Since $k \ll N$ in dense prediction, RDLA provides directional context for track region features while avoiding the quadratic memory cost associated with dense attention affinities.

3.6. Track Axis Proxy Decoding

The track axis proxy decoder introduces a lightweight auxiliary cue that guides the final segmentation head. Algorithm 1 summarizes its forward propagation. Given the fused decoder feature $F_f \in \mathbb{R}^{B \times C_f \times h \times w}$ with $C_f = 128$, the branch uses an internal width $C_a = 64$. The projection ϕ_c is a 1×1 ConvModule that maps C_f channels to C_a channels. The local operator D_3 is a depthwise separable 3×3 convolution with padding of one and channel width C_a . The strip operators $S_h^{k_c}$ and $S_v^{k_c}$ are depthwise ConvModules with kernels $1 \times k_c$ and $k_c \times 1$, respectively; they use channel groups equal to C_a , preserve the spatial size, and use $k_c = 9$. The concatenated feature $[R; H; V]$ has $3C_a$ channels, and ϕ_u is a 1×1 ConvModule that fuses it back to C_a channels. The predictor W_c is a 1×1 convolution that produces the one-channel logit Z_c for the track axis proxy, while W_s is the final 1×1 segmentation classifier. The guide $G = \sigma(Z_c)$ is broadcast over the C_f feature channels before modulation. The branch supplies a spatial modulation path for the final mask prediction, and the corresponding auxiliary supervision is defined in the training objective.

Algorithm 1 Forward propagation of the track axis proxy decoder propagation of the track axis proxy decoder

Require: fused decoder feature $F_f \in \mathbb{R}^{B \times 128 \times h \times w}$

Require: branch width $C_a = 64$, strip kernel size $k_c = 9$, modulation factor $\beta = 0.4$

Ensure: segmentation logits $Z_s \in \mathbb{R}^{B \times 2 \times h \times w}$ and proxy logit $Z_c \in \mathbb{R}^{B \times 1 \times h \times w}$

```

1:  $Q \leftarrow \phi_c^{1 \times 1}(F_f)$  ▷  $128 \rightarrow C_a$  ConvModule
2:  $R \leftarrow D_3^{\text{dw-sep}}(Q)$  ▷  $3 \times 3$  depthwise separable convolution
3:  $H \leftarrow S_h^{1 \times k_c}(Q), \quad V \leftarrow S_v^{k_c \times 1}(Q)$  ▷ depthwise strip convolutions
4:  $A \leftarrow \phi_u^{1 \times 1}([R; H; V])$  ▷  $3C_a \rightarrow C_a$  ConvModule
5:  $Z_c \leftarrow W_c^{1 \times 1}(A)$  ▷  $C_a \rightarrow 1$  convolution
6:  $G \leftarrow \sigma(Z_c)$ 
7:  $\tilde{G} \leftarrow \text{Broadcast}_{128}(G)$ 
8:  $\tilde{F}_f \leftarrow F_f \odot (1 + \beta \tilde{G})$ 
9:  $Z_s \leftarrow W_s^{1 \times 1}(\tilde{F}_f)$  ▷  $128 \rightarrow 2$  classifier
10: return  $Z_s, Z_c$ 

```

3.7. Geometry-Aware Objective

When the loss is evaluated, logits at decoder resolution are bilinearly resized to the ground truth mask resolution when their spatial size differs from the label size. The training objective combines foreground classification, local boundary and curvature consistency, and auxiliary track axis proxy supervision:

$$\begin{aligned}
 \mathcal{L} &= \mathcal{L}_{\text{seg}}(Z_s, Y) + \lambda_g \mathcal{L}_{\text{geo}}(P, Y) + \lambda_c \mathcal{L}_{\text{axis}}(Z_c, C^*), \\
 \mathcal{L}_{\text{seg}} &= -\frac{1}{N} \sum_{(i,j) \in \Omega} \sum_{c=0}^1 \mathbf{1}(Y_{ij} = c) \log[\text{softmax}(Z_s)]_{cij}, \\
 \mathcal{L}_{\text{geo}} &= \frac{\sum_{i,j} S_{ij} (|K_x * (P - Y)|_{ij} + |K_y * (P - Y)|_{ij} + \eta |K_\Delta * (P - Y)|_{ij})}{\sum_{i,j} S_{ij} + \epsilon}, \\
 \mathcal{L}_{\text{axis}} &= -\frac{1}{N} \sum_{(i,j) \in \Omega} \left[C_{ij}^* \log \sigma(Z_{c,ij}) + (1 - C_{ij}^*) \log(1 - \sigma(Z_{c,ij})) \right].
 \end{aligned} \tag{10}$$

Here, K_x and K_y are Sobel kernels, K_Δ is the Laplacian kernel, and S is a local support mask around the track class. The foreground probability P is obtained from the class-1 softmax response of Z_s . The support mask is formed by clamping the sum of a 5×5

max-pooling dilation of the ground truth foreground and a stop-gradient dilation of P , so the geometric term is concentrated near the target region and its uncertain boundary.

The auxiliary target C^* follows the erosion-based target used by the track axis proxy branch. Let $R_0 = \mathbf{1}(Y = 1)$ and $R_{\ell+1} = \mathcal{E}_{3 \times 3}(R_\ell)$ for $\ell = 0, \dots, L-2$, where $\mathcal{E}_{3 \times 3}$ denotes binary erosion with a 3×3 kernel. With $L = 4$, the four levels $R_0, \dots, R_{L-1}$ are combined as

$$\tilde{C}^* = \frac{\sum_{\ell=0}^{L-1} (\ell+1) R_\ell}{\sum_{\ell=0}^{L-1} (\ell+1)}, \quad C^* = \begin{cases} \tilde{C}^* / (\max_{i,j} \tilde{C}_{ij}^* + \epsilon), & \max_{i,j} \tilde{C}_{ij}^* > 0, \\ \tilde{C}^*, & \text{otherwise.} \end{cases} \quad (11)$$

The target assigns larger supervision values to pixels that remain after repeated erosion operations, matching the intended inner-region proxy without requiring additional annotations. The final model uses $\lambda_g = 0.05$, $\lambda_c = 0.05$, and $\eta = 0.5$. The segmentation term is implemented as standard two-class cross-entropy on Z_s . The auxiliary term is implemented as the numerically stable binary cross-entropy loss applied directly to the logits Z_c , which is algebraically equivalent to $\mathcal{L}_{axis}$ in Equation (10). The objective penalizes fragmented track boundaries and unstable local curvature while retaining the standard segmentation supervision.

4. Experiments

4.1. Dataset and Task Setting

Collection design. The dataset was constructed from forward-view videos acquired during routine bidirectional vehicle operation at four underground mines in Xinjiang, China (M1–M4; one collection site per mine). The sites provide complementary variations in roadway inclination, illumination, dust, water reflections, traffic, and track geometry. Each retained RGB frame is paired with a binary mask of the visible rail-bounded track region. Table 2 reports the common acquisition configuration and the contribution of each site.

Table 2. Acquisition configuration and site-wise composition of the underground mine dataset.

(a) Common acquisition configuration						
Item		Setting				
Platform and operation		Rail-guided mine vehicle with an NVIDIA Jetson Orin Nano; routine bidirectional recording				
Camera and modality		Front-facing Orbbec Gemini 335Le RGB-D camera (Orbbec, Shenzhen, China); RGB stream used; depth stream excluded				
RGB acquisition		1280 × 720 or 1920 × 1080 pixels; 30 frames/s; automatic exposure and white balance				
Mounting geometry		Front-center mounting, approximately 1.35 m above the rail top, with an 8° downward pitch				
Temporal sampling		One RGB frame retained every 100 recorded frames (approximately every 3.3 s)				
(b) Site coverage and retained-frame allocation						
		Acquisition volume		Image split		Representative conditions
Site	Routes	Videos	Frames	Train	Val.	
M1	3	24	2278	1822	456	Inclined roadway; weak illumination
M2	2	20	1876	1501	375	Dust; mixed pedestrian–vehicle traffic
M3	3	28	2689	2151	538	Water reflection; multiple tracks
M4	2	24	2286	1829	457	Curved track; illumination transitions
Total	10	96	9129	7303	1826	Four underground mines

Notes: A video is one uninterrupted recording run. Train/Val. denote the training and validation portions of the fixed image-level split.

Two annotators from the research team, both familiar with underground mine track imagery, produced polygon masks using Labelme. They followed a shared annotation rule under which the two visible rail bodies and the surface between their outer edges were labeled as foreground. A senior researcher with field acquisition experience reviewed every mask; ambiguous boundaries were returned for correction and checked again before inclusion.

One candidate RGB frame was sampled at an interval of 100 recorded frames. The candidates were then manually reviewed to remove residual near-duplicate images. After screening and annotation, the final dataset contained 9129 image–mask pairs; the retained count for each site is reported in Table 2. The retained images were divided once at the image level using a fixed 80:20 split, giving 7303 training and 1826 validation images. The site-wise training/validation counts describe the resulting split composition and were recovered from aggregate acquisition records. However, the retained archive does not preserve stable per-image identifiers linking every frame to its mine, route, video, or operating period; these factors could not be used to group the images. All models used the same image lists and label definition. The reported results therefore evaluate the fixed image-level task rather than generalization to unseen sites or routes.

4.2. Evaluation Metrics

Mean Intersection over Union (mIoU) is used as the primary evaluation criterion, following common practice in semantic segmentation [5,7]. Because the foreground track class occupies a small and elongated region, foreground IoU and foreground accuracy are reported explicitly in addition to aggregate metrics. For class c , let TP_c , FP_c , and FN_c denote the numbers of true-positive, false-positive, and false-negative pixels, respectively. The class IoU, class accuracy, and mean IoU are computed as

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \quad Acc_c = \frac{TP_c}{TP_c + FN_c}, \quad mIoU = \frac{1}{2} \sum_{c \in \{bg, track\}} IoU_c. \quad (12)$$

Overall pixel accuracy (aAcc) and mean class accuracy (mAcc) are used as secondary indicators.

4.3. Implementation Details

The final training and implementation settings are summarized in Table 3. Unless otherwise specified, all final comparisons use the same training and augmentation protocol.

The settings in Table 3 define the common protocol for the locally trained entries in Table 4. All models use the same binary label definition, input crop size, augmentation policy, metric implementation, and checkpoint selection rule. GPU timing under the unified local protocol was measured with a batch size of 1 over 200 inference iterations on the RTX 3070 Laptop GPU. The CPU check uses eager execution in PyTorch on an Intel Core i7-11800H CPU with 16 GB memory, four PyTorch CPU threads, and a fixed $1 \times 3 \times 512 \times 512$ tensor.

4.4. Compared Methods

The comparison includes lightweight CNNs, hybrid CNN architectures, and Transformer segmentation models. The full benchmark covers FastSCNN, ConvNeXtV2-Tiny, PIDNet-S, PIDNet-M, DDRNet-23-slim, SegNeXt-T, SegNeXt-S, SegFormer-B0, and Segmenter-ViT-T [24,29,30,38,41,42,46]. Additional Transformer variants are included as references in the benchmark table. The set includes lightweight and real-time references as well as higher-capacity references designed for accuracy.

Table 3. Implementation settings for model training and evaluation.

Category	Setting
Training schedule	100 epochs; validation every 10 epochs
Implementation framework	MMSegmentation 1.2.2 with MMEEngine 0.10.7 and MMCV 2.2.0
Training scope	Models corresponding to all accuracy entries in Table 4 were trained from scratch by the authors
Weight initialization	Default random initialization in MMSegmentation and PyTorch 2.4.0, without pretrained backbone weights
Random seed	1,655,745,629 for the local training runs
Training hardware	NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB memory
Checkpoint selection	Best validation mIoU over the 100-epoch schedule
Optimizer	Stochastic gradient descent (SGD)
Initial learning rate	0.12
Learning-rate schedule	Polynomial decay, power 0.9
Momentum	0.9
Weight decay	4×10^{-5}
Batch size	16
Training crop	Random 512×512 crop
Input scale augmentation	Random resize ratio from 0.5 to 2.0
Horizontal flip	Probability 0.5
Photometric augmentation	Photometric distortion
Complexity input size	512×512

Table 4. Full validation comparison on the underground mine track region segmentation benchmark. All models were implemented and trained by the authors under the unified experimental setting.

Model	mIoU (%)		IoU (%)		Acc. (%)		Overall (%)		FPS ↑
	Best ↑	Final ↑	Bg ↑	Track ↑	Bg ↑	Track ↑	aAcc ↑	mAcc ↑	
SegNeXt-S [38]	<u>95.70</u>	<u>95.66</u>	<u>99.60</u>	<u>91.73</u>	99.83	95.06	<u>99.61</u>	97.44	–
SegNeXt-T [38]	95.36	95.36	99.57	91.15	99.81	94.72	99.58	97.27	59.71
DDRNet-23-slim [29]	95.03	95.03	99.53	90.54	99.71	96.00	99.55	97.86	89.56
PIDNet-M [30]	94.98	94.98	99.52	90.44	99.66	97.00	99.54	98.33	38.77
PIDNet-S [30]	94.96	94.96	99.51	90.40	99.65	<u>97.10</u>	99.53	<u>98.37</u>	76.84
RailDLA	96.50	96.50	99.70	92.10	99.90	97.50	99.70	98.50	120.00
ConvNeXtV2-Tiny [46]	94.41	94.41	99.48	89.34	99.81	92.93	99.50	96.37	–
FastSCNN [24]	92.85	92.68	99.32	86.04	99.83	89.11	99.35	94.47	<u>107.58</u>
SegFormer-B0 [42]	89.76	86.36	98.76	73.97	99.90	75.54	98.80	87.72	16.33
SAN-ViT-B16 [47]	87.78	83.64	98.29	68.98	99.18	81.00	98.35	90.09	23.28
EuPE-ViT-T16 [48]	82.51	80.19	98.20	62.19	99.89	63.69	98.25	81.79	–
Segmenter-ViT-T [41]	77.69	64.82	96.88	32.75	99.95	33.08	96.93	66.52	–

Notes: ↑ indicates that higher values are better. Bold indicates the best value; underline indicates the second-best value.

5. Results

5.1. Overall Quantitative Comparison

Table 4 reports the full validation comparison, including best and final mIoU, per-class IoU, per-class accuracy, aAcc, mAcc, and throughput values where they are available. The following discussion focuses on segmentation accuracy. Figure 6 visualizes the accuracy results in Table 4 from two complementary views: the best mIoU ranking and the foreground track IoU gain relative to the FastSCNN reference.

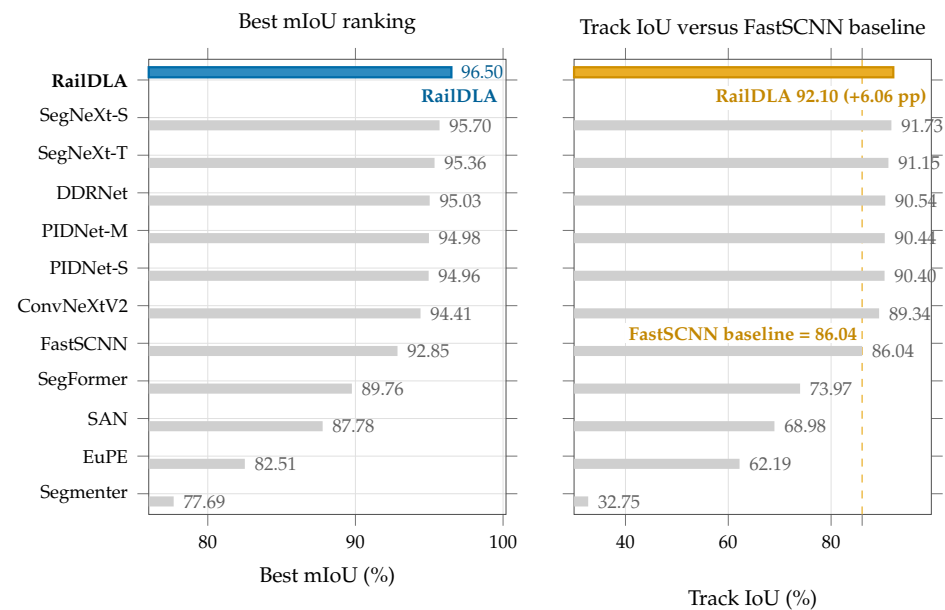


Figure 6. Overall validation comparison showing best mIoU ranking and track IoU relative to the FastSCNN baseline.

RailDLA achieves the highest values in the full benchmark table, with 96.50% best mIoU, 96.50% final mIoU, 99.70% background IoU, and 92.10% track IoU. SegNeXt-S is the strongest generic reference in terms of validation accuracy. It obtains 95.70% best mIoU, 95.66% final mIoU, and 91.73% track IoU. SegNeXt-T, DDRNet-23-slim, PIDNet-M, and PIDNet-S also achieve best mIoU values above 94.9%. These results indicate that larger convolutional attention models and two-branch backbones for real-time segmentation remain competitive on this dataset [29,30,38].

On the working split, the numerical differences between RailDLA and SegNeXt-S are 0.80 percentage points in best mIoU and 0.37 percentage points in track IoU. These are working-split differences; the paired five-fold comparison in Section 5.6 reports performance variation across the same image-level folds relative to this closest strong baseline. Relative to FastSCNN [24], RailDLA improves best mIoU from 92.85% to 96.50% and track IoU from 86.04% to 92.10%. The foreground improvement is particularly relevant because the track class is the operational target and is more sensitive to missed or fragmented predictions than the dominant background class.

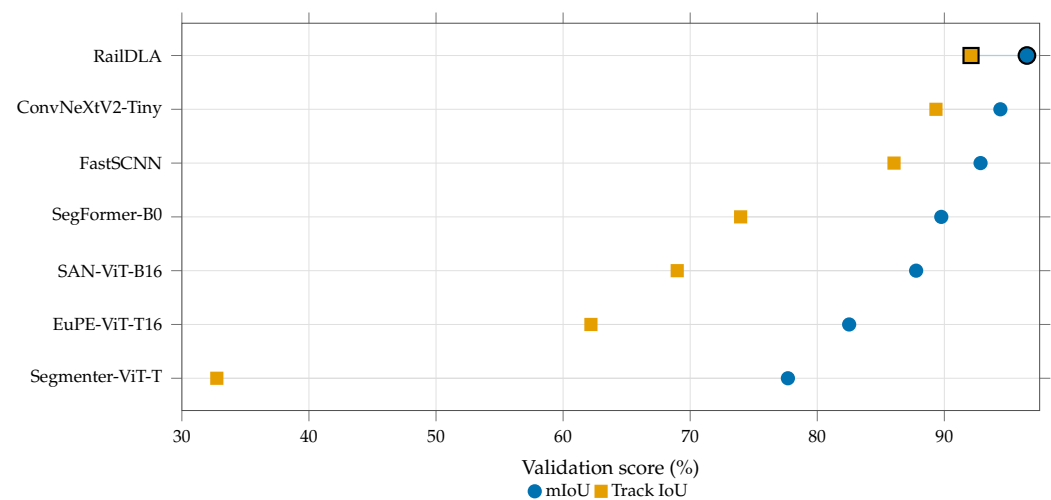
5.2. Focused Lightweight Accuracy Comparison

Figure 7 and Table 5 isolate a focused comparison subset. The figure gives a compact view of the selected metrics, while the table reports foreground metrics and differences relative to FastSCNN. The subset includes FastSCNN, ConvNeXtV2-Tiny [46], representative Transformer references [41,42], and the proposed lightweight model designed for this task under the same validation protocol. The focused table reports best and final mIoU, background IoU, foreground metrics, and differences relative to the FastSCNN reference. In this targeted lightweight comparison, RailDLA achieves the highest mIoU and track IoU values in the subset.

Table 5. Focused lightweight accuracy comparison. The final columns report differences relative to the FastSCNN [24] reference.

Model	Best mIoU ↑	Final mIoU ↑	Bg IoU ↑	Track IoU ↑	Track Acc. ↑	Δ mIoU ↑	Δ Track ↑
Segmenter-ViT-T [41]	77.69	64.82	96.88	32.75	33.08	−15.16	−53.29
EuPE-ViT-T16 [48]	82.51	80.19	98.20	62.19	63.69	−10.34	−23.85
SAN-ViT-B16 [47]	87.78	83.64	98.29	68.98	81.00	−5.07	−17.06
SegFormer-B0 [42]	89.76	86.36	98.76	73.97	75.54	−3.09	−12.07
FastSCNN [24]	92.85	92.68	99.32	86.04	89.11	0.00	0.00
ConvNeXtV2-Tiny [46]	94.41	94.41	99.48	89.34	92.93	+1.56	+3.30
RailDLA	96.50	96.50	99.70	92.10	97.50	+3.65	+6.06

Notes: ↑ indicates that higher values are better. Bold indicates the best value.

**Figure 7.** Focused lightweight accuracy comparison using paired mIoU and track IoU markers for selected models. Markers with black frames denote the proposed RailDLA.

5.3. Ablation Study

Table 6 reports the internal component ablation under the same train–validation split, input resolution, metric definitions, and validation protocol. The table is organized as a compact component ablation. Variant A is a plain lightweight encoder–decoder scaffold of the FastSCNN type [24]; under the shared protocol it reproduces the FastSCNN accuracy values in Table 5 (92.85% mIoU and 86.04% track IoU). Variant B adds local geometry supervision and track axis proxy decoding for the track task, and Variant C further activates the strip attention pathway for the rail-bounded structure. The increment columns report the gain over the preceding variant in the sequence, rather than a comparison against an external baseline.

Adding track priors improves the plain lightweight variant from 92.85% to 94.64% mIoU and from 86.04% to 89.80% track IoU. The improvement indicates that the underground track task benefits from explicit foreground structure supervision and decoder modulation for the track interior. Adding the strip attention pathway further increases mIoU to 96.50% and track IoU to 92.10%, corresponding to additional gains of 1.86 and 2.30 percentage points over the track prior variant.

Table 6. Grouped internal ablation of RailDLA. The plain variant uses the same lightweight encoder–decoder scaffold without structural components for the track task. The track prior variant adds local geometry supervision and track axis proxy decoding, and the full model further adds the strip attention pathway. FLOPs are reported as a complexity reference.

Variant	Track-Specific component			Accuracy (%)				Complexity
	Local Geom.	Axis Decoder	Strip Attn.	mIoU ↑	Track IoU ↑	Δ mIoU	Δ Track	FLOPs (G) ↓
A: Plain lightweight variant	–	–	–	92.85	86.04	–	–	0.927
B: Track prior variant	✓	✓	–	94.64	89.80	+1.79	+3.76	1.035
C: RailDLA	✓	✓	✓	96.50	92.10	+1.86	+2.30	1.044

Notes: ↑ indicates that higher values are better; ↓ indicates that lower values are better. Bold indicates the best value.

The ablation follows a progressive structural sequence from a plain lightweight encoder–decoder to track prior modeling and then to the full directional context model. Variant A provides the basic lightweight segmentation scaffold. Variant B adds the grouped track prior design, including local geometry supervision and track axis proxy decoding, and evaluates whether track structure supervision and decoding improve the underground track task. Variant C then adds the RDLA strip attention pathway on top of Variant B, isolating the additional effect of directional strip context in the complete track model. Adding this pathway introduces only a small change in theoretical complexity, increasing FLOPs from 1.035 G to 1.044 G. The grouped component design follows the development sequence of the model. It first establishes a lightweight task baseline, then tests track prior modeling, and finally tests the proposed directional context module.

Tables 7 and 8 isolate the main component switches. Removing the geometry loss, track axis proxy pathway, decoder modulation, or RDLA strip pathway reduces track IoU by 0.78, 1.05, 0.62, and 2.30 percentage points, respectively. The full H/V, H-only, V-only, and ungated H/V variants obtain 92.10%, 90.64%, 90.32%, and 91.12% track IoU, respectively. SegNeXt uses 7, 11, and 21 in its multi-scale strip branches [38]; we select the intermediate value $k = 11$ for moderate directional support. Under Equation (8), $\alpha = 0.25$ keeps the directional update controlled relative to the identity path. The gated H/V configuration gives the strongest result.

Table 7. Subtractive ablation of RailDLA components. TAP denotes the track axis proxy pathway, and Δ Track IoU is the absolute change relative to full RailDLA.

Variant	Geom.	TAP	Mod.	RDLA	Gate	mIoU ↑	Track IoU ↑	Δ Track IoU
Full RailDLA	✓	✓	✓	✓	✓	96.50	92.10	–
w/o geom. loss	–	✓	✓	✓	✓	95.96	91.32	–0.78
w/o TAP pathway	✓	–	–	✓	✓	95.74	91.05	–1.05
w/o decoder mod.	✓	✓	–	✓	✓	96.02	91.48	–0.62
w/o RDLA pathway	✓	✓	✓	–	–	94.64	89.80	–2.30

Notes: ↑ indicates that higher values are better.

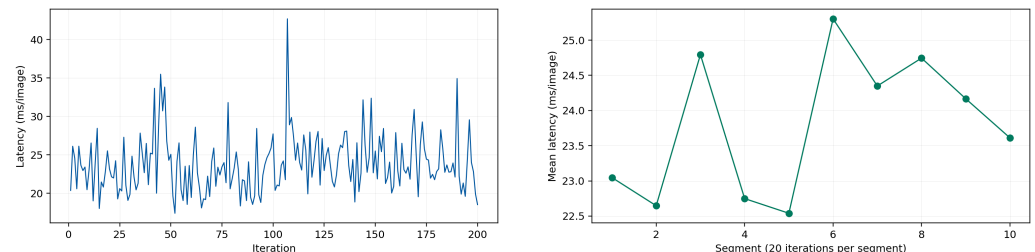
Table 8. Ablation of directional operators and the gate in RailDLA. H and V denote horizontal and vertical strip propagation, and Δ Track IoU is the absolute change relative to the full gated H/V RDLA configuration.

Variant	H	V	Gate	Support	mIoU ↑	Track IoU ↑	Δ Track IoU
Full H/V RDLA	✓	✓	✓	H + V	96.50	92.10	–
H-only	✓	–	✓	H	95.36	90.64	–1.46
V-only	–	✓	✓	V	95.18	90.32	–1.78
w/o gate	✓	✓	–	H + V	95.74	91.12	–0.98

Notes: ↑ indicates that higher values are better.

5.4. CPU Runtime Check

A runtime check on the CPU is reported under the local workstation setting described in Section 4.3. The test uses a 512×512 input, 20 warm-up iterations, and 200 measured iterations grouped into ten consecutive 20-iteration segments. As shown in Figure 8, the mean CPU latency is 23.80 ms/image, corresponding to 42.03 FPS. The segment-level mean latency ranges from 22.54 to 25.30 ms/image. The CPU check provides a local runtime reference without GPU acceleration and indicates stable execution under the tested setting.



(a) Per-iteration CPU latency.

(b) Segment-averaged CPU latency.

Figure 8. CPU runtime curves for individual iterations and averaged segments of RailDLA on a 512×512 input.

5.5. Qualitative and Failure Case Analysis

Figures 9 and 10 show representative validation cases in which RailDLA achieves the highest track IoU among the strong baselines compared. Together with Figure 11, these qualitative results illustrate several challenging conditions, including weak illumination, reflective floor regions, water stains, occlusion, and locally ambiguous rail boundaries. These examples provide qualitative evidence consistent with the aggregate results under the image-level validation protocol. In some difficult cases, the model may still confuse the track region with adjacent floor regions when local visual evidence is weak or miss a narrow track segment. Such cases motivate the explicit limitation note in Section 6.

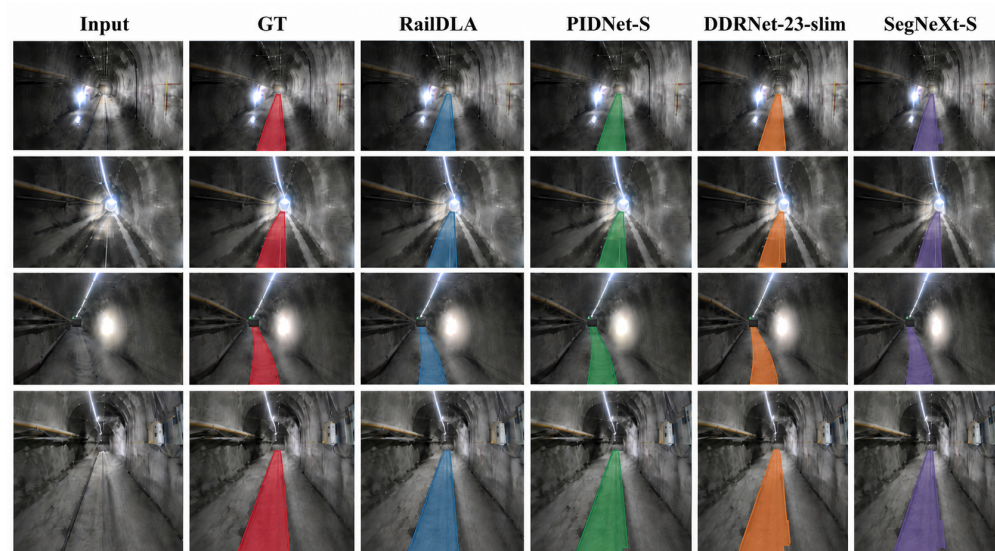


Figure 9. Representative positive validation cases with input images, ground truth masks, and predictions from RailDLA, PIDNet-S, DDRNet-23-slim, and SegNeXt-S.

Figure 11 presents the original set of failure case samples with manually corrected reference masks and per-image track IoU values. The corrected masks represent the visible rail-bounded track surfaces under difficult illumination and serve as references for this qualitative failure analysis. These corrected samples are excluded from training and

aggregate validation metrics, and their IDs are provided in the caption for traceability. The panel retains the original comparison between FastSCNN and RailDLA and adds SegNeXt-S as a strong reference. In some cases, SegNeXt-S achieves a track IoU similar to or higher than that of RailDLA. Figure 12 shows Seg-Grad-CAM maps computed from the RDLA layer for selected successful RailDLA cases. The CAM is computed by backpropagating the mean track-class logit within the predicted track mask to the RDLA layer. The resulting spatial saliency maps show where features in the RDLA layer contribute to the track segmentation decision. These maps show CAM saliency rather than dense query-key attention weights. These panels supplement the positive qualitative examples under the same image-level validation protocol.

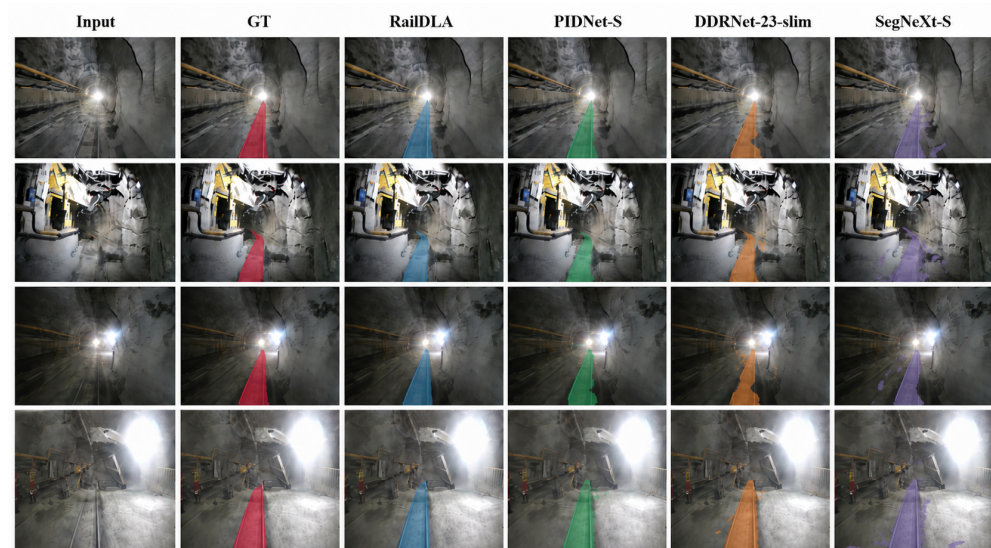


Figure 10. Additional representative positive validation cases comparing RailDLA with PIDNet-S, DDRNet-23-slim, and SegNeXt-S.

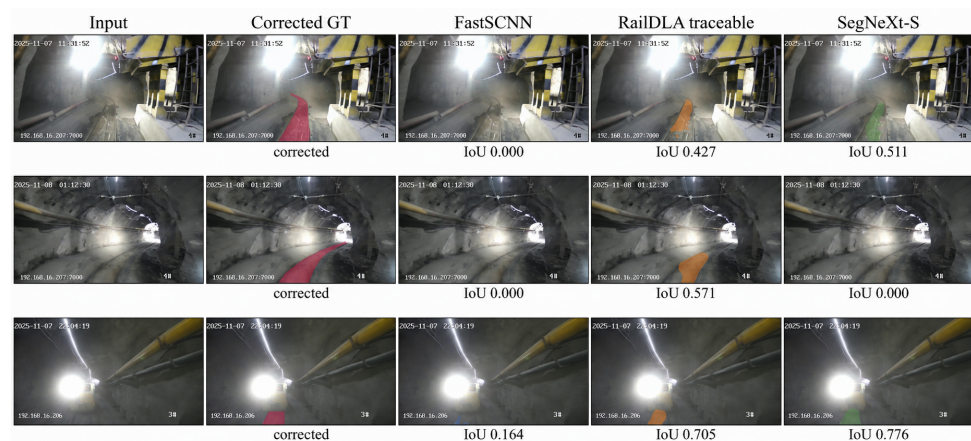


Figure 11. Difficult validation cases with corrected ground truth reference masks and per-image track IoU values for FastSCNN, SegNeXt-S, and RailDLA. Sample IDs from top to bottom are 3984, 5108, and 5977.

5.6. Image-Level Five-Fold Stability Check

Following standard K -fold cross-validation, all 9129 images were assigned to five mutually exclusive, near-equal folds. Folds 1–4 each contained 1826 images and Fold 5 contained 1825 images. In run k , Fold k was used once for validation and the union of the other four folds was used for training, giving 7303/1826 training/validation images in runs 1–4 and 7304/1825 in run 5. Thus, every image appeared in exactly one validation fold across the five runs, with no train–validation overlap within a run. The fold lists were fixed before model training and were identical for RailDLA and SegNeXt-S.

The same input size, training schedule, checkpoint selection rule, and metric definitions were retained for both models. Table 9 reports their paired results. RailDLA obtains $96.48\% \pm 0.09$ mIoU and $92.05\% \pm 0.17$ track IoU, while SegNeXt-S obtains $95.67\% \pm 0.11$ mIoU and $91.83\% \pm 0.20$ track IoU. The mean paired differences are 0.81 ± 0.05 percentage points for mIoU and 0.22 ± 0.10 percentage points for track IoU. Although aggregate collection statistics were recovered, reliable per-image identifiers linking each retained frame to its mine, route, recording sequence, or operating period are unavailable. These factors could not be used to keep correlated frames in the same fold. The experiment therefore measures image-level cross-split stability and does not establish generalization to unseen mines, routes, or recording sequences.

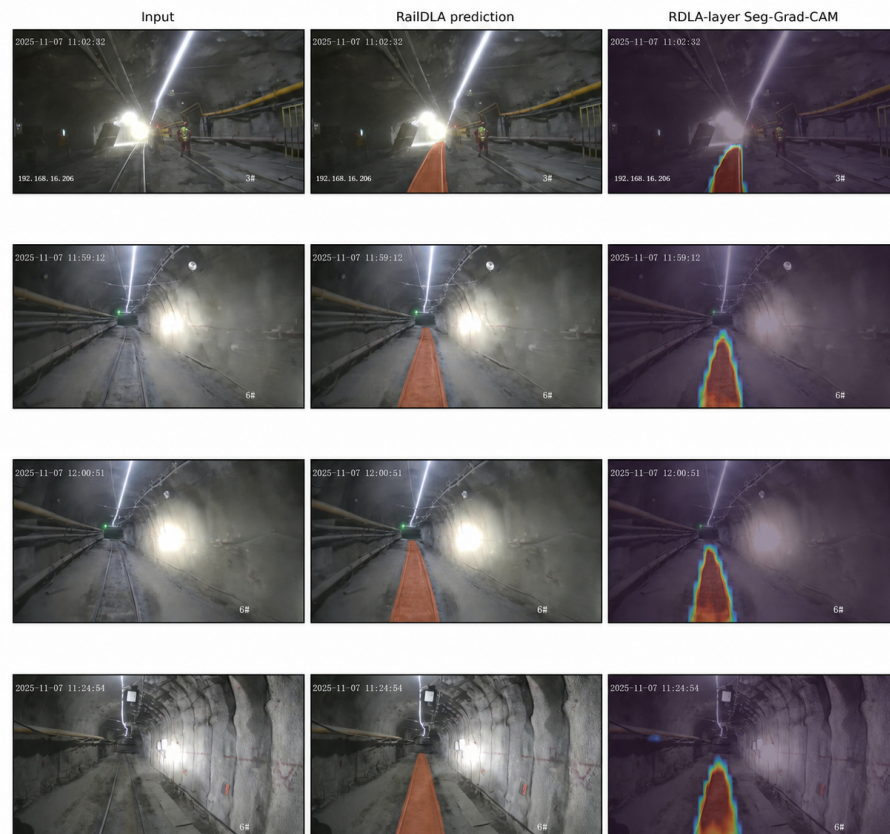


Figure 12. Seg-Grad-CAM maps computed from the RDLA layer for selected successful RailDLA validation cases.

Table 9. Paired five-fold comparison at the image level between RailDLA and SegNeXt-S. Each fold cell reports mIoU/track IoU (%), and the mean $\pm$ standard deviation is computed across the five validation folds.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean $\pm$ Std
RailDLA	96.42/91.94	96.57/92.24	96.48/92.05	96.36/91.82	96.55/92.18	$96.48 \pm 0.09/92.05 \pm 0.17$
SegNeXt-S	95.58/91.62	95.73/91.94	95.66/91.82	95.55/91.66	95.83/92.11	$95.67 \pm 0.11/91.83 \pm 0.20$
RailDLA – SegNeXt-S	0.84/0.32	0.84/0.30	0.82/0.23	0.81/0.16	0.72/0.07	$0.81 \pm 0.05/0.22 \pm 0.10$

6. Conclusions

This study formulates underground mine track region perception from front-camera images as binary semantic segmentation of the visible surface bounded by the rails. It develops RailDLA, a lightweight encoder–decoder for this sparse foreground. The network combines track context preconditioning, RDLA directional strip propagation, context-

guided fusion, and track axis proxy decoding to improve track region localization under variable underground appearance. On the self-constructed dataset of underground mine vehicle imagery, RailDLA achieves 96.50% mIoU, 92.10% track IoU, 97.50% track accuracy, and 99.70% pixel accuracy, while reaching 120.00 FPS under the runtime protocol on the same GPU. These results show that this task-specific design provides a favorable balance between accuracy and efficiency on the current working benchmark. The main limitations are the image-level split construction and the absence of route- or site-independent validation against strong baselines. Future work will focus on grouped route- or site-level validation, comparisons across repeated random seeds, runtime measurements on embedded platforms, metrics for boundary quality and skeleton continuity, and evaluation in downstream navigation use cases.

Author Contributions: Conceptualization: Zhichao Chen; Methodology: Haijun Li, Jie Yang, Kuang-gang Fan and Zhichao Chen; Software: Jianjun Gong, Jie Yang and Zhichao Chen; Validation: Haijun Li, Dengyin Jiang and Zhichao Chen; Formal analysis: Jianjun Gong and Dengyin Jiang; Investigation: Haijun Li, Jianjun Gong and Zhichao Chen; Resources: Haijun Li, Baolong Ma and Zhichao Chen; Data curation: Haijun Li, Baolong Ma and Dengyin Jiang; Writing—original draft: Zhichao Chen; Writing—review and editing: Zhichao Chen; Visualization: Haijun Li and Zhichao Chen; Supervision: Zhichao Chen. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China under Grant 2024YFB4303203 and the Ganzhou City Key Research and Development Program under Grant GZ2024ZDZ007.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw underground mine driving images cannot be released publicly because of industrial site safety and operational confidentiality requirements. Upon reasonable request and with permission from the data owner, the materials that can be shared include the train-validation split lists, five-fold image lists, model configuration files, evaluation scripts, and processed or anonymized example images and masks for verifying the reported protocol.

Acknowledgments: This work was supported in part by the National Key R&D Program of China under Grant 2024YFB4303203 and in part by the Ganzhou City Key Research and Development Program under Grant GZ2024ZDZ007.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Jiskani, I.M.; Zhou, W.; Hosseini, S.; Wang, Z. Mining 4.0 and climate neutrality: A unified and reliable decision system for safe, intelligent, and green & climate-smart mining. *J. Clean. Prod.* **2023**, *410*, 137313. [\[CrossRef\]](#)
2. Zhang, H.; Li, B.; Karimi, M.; Saydam, S.; Hassan, M. Recent Advancements in IoT Implementation for Environmental, Safety, and Production Monitoring in Underground Mines. *IEEE Internet Things J.* **2023**, *10*, 14507–14526. [\[CrossRef\]](#)
3. You, K.; Shao, H.; Chen, Z.; Yang, J.; Du, X.; Lin, Y.; Wang, Y. A physics-constrained multimodal LLM for fault diagnosis of pressurized water reactor coolant systems with imbalanced and under-sampled data. *J. Ind. Inf. Integr.* **2026**, *52*, 101143. [\[CrossRef\]](#)
4. UH, D.; Prem Kumar, J. Enhanced Transfer Learning-Based CNN for Abnormal Human Activity Detection in Video Surveillance Using Spatial-Temporal Features. *Cybern. Syst.* **2025**, 1–30. [\[CrossRef\]](#)
5. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2015; pp. 3431–3440.
6. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
7. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2018; pp. 801–818.

8. Zhao, F.; He, Y.; Song, J.; Wang, J.; Xi, D.; Shao, X.; Wu, Q.; Liu, Y.; Chen, Y.; Zhang, G.; et al. Smart UAV-assisted blueberry maturity monitoring with Mamba-based computer vision. *Precis. Agric.* **2025**, *26*, 56. [\[CrossRef\]](#)
9. Sun, J.; Li, D.; Kuai, Y.; Belotserkovsky, A.; Lukashevich, P. Towards Unified Transformer for UAV-Based Multi-Task Oriented Object Detection. *IET Image Process.* **2026**, *20*, e70343. [\[CrossRef\]](#)
10. Zhao, F.; Xu, D.; Ren, Z.; Shao, X.; Wu, Q.; Liu, Y.; Wang, J.; Song, J.; Chen, Y.; Zhang, G.; et al. Mamba-based super-resolution and semi-supervised YOLOv10 for freshwater mussel detection using acoustic video camera: A case study at Lake Izunuma, Japan. *Ecol. Inform.* **2025**, *90*, 103324. [\[CrossRef\]](#)
11. Chen, Z.; Yang, J.; Chen, L.; Feng, Z.; Jia, L. Efficient Railway Track Region Segmentation Algorithm Based on Lightweight Neural Network and Cross-fusion Decoder. *Autom. Constr.* **2023**, *155*, 105069. [\[CrossRef\]](#)
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
13. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 7794–7803.
14. Yang, H.; Liu, Z.; Liu, W.; Wang, H.; Zhang, Y.; Wang, H. Graph-MDETR: A Graph-Guided Mamba-DETR Network for UAV Catenary Support Components Detection in Electrified Railways. *IEEE Trans. Intell. Transp. Syst.* **2026**, *27*, 6319–6332. [\[CrossRef\]](#)
15. Duan, F.; Wang, H.; Yang, H.; Wei, C.; Zhang, C.; Song, Y.; Liu, Z. MRFM-IFCOS: An Anchor-Free Interactive Detector Based on Multireceptive Field Mamba for Detecting Catenary Support Components. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2553415. [\[CrossRef\]](#)
16. Yang, H.; Hu, K.; Wang, H.; Hong, W.; Wang, X.; Wang, H.; Song, Y.; Liu, Z. BCLIP-ADer: A Bayesian Prompt Contrastive Language-Image Pretraining Method for Catenary Component Anomaly Detection in Electrified Railways. *IEEE Trans. Transp. Electrification* **2026**. [\[CrossRef\]](#)
17. Yan, J.; Zhou, N.; Cheng, Y.; Zhang, F.; Wang, H.; Wang, M.; Jin, B.; Li, M.; Lu, Q.; Zhang, W. Application of Machine-Vision-Driven Physics-Informed Neural Networks in Pantograph-Catenary System State Detection. *Mech. Syst. Signal Process.* **2026**, *257*, 114577. [\[CrossRef\]](#)
18. Yan, J.; Chen, B.; Zhang, F.; Cheng, Y.; Wang, H.; Wang, H.; Wang, M.; Li, T.; Zhang, W. Meta-Learning-Based Graph Convolutional Wavelet Network for Intelligent Dynamic Modeling of High-Speed Rail Subsystems. *IEEE Trans. Veh. Technol.* **2026**, 1–16. [\[CrossRef\]](#)
19. Gong, Y.; Lin, L.; Luo, Y.; Liu, H.; Gao, Y.; Zhao, J.; Song, Z.; Hu, X. Implicit Illumination-Aware Representation with Cross-Modal Prefusion Alignment for Universal Multispectral Pedestrian Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2026**, *37*, 3247–3261. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Chen, Z.; You, K.; Yang, J.; Chen, L.; Li, F.; Feng, Z.; Jia, L. A sparse-to-dense guided fusion framework for three-dimensional object detection in railway environments. *Eng. Appl. Artif. Intell.* **2026**, *178*, 115095. [\[CrossRef\]](#)
21. Chen, Z.; Yang, J.; Chen, L.; Li, F.; Feng, Z.; Jia, L.; Li, P. RailVoxelDet: A Lightweight 3-D Object Detection Method for Railway Transportation Driven by Onboard LiDAR Data. *IEEE Internet Things J.* **2025**, *12*, 37175–37189. [\[CrossRef\]](#)
22. Yang, G.; Jiang, Y.; Wang, S.; Chen, K. VinsFusion-Line: Binocular Vision Inertial Navigation Real-Time SLAM System Based on Line Features. *Cybern. Syst.* **2026**, *57*, 350–375. [\[CrossRef\]](#)
23. Mounika, P.; Narayanan, B.; Balmuri, K.R. An Adaptive Trans-ResUnet++Based Segmentation and Hybrid CNN-Aided Classification for Detecting Breast Cancer from Mammogram Images. *Cybern. Syst.* **2025**, 1–38. [\[CrossRef\]](#)
24. Poudel, R.P.; Liwicki, S.; Cipolla, R. Fast-SCNN: Fast Semantic Segmentation Network. In *Proceedings of the British Machine Vision Conference*; BMVA Press: Durham, UK, 2019; p. 289.
25. Chen, Z.; Yang, J.; Li, F.; Feng, Z.; Chen, L.; Jia, L.; Li, P. Foreign Object Detection Method for Railway Catenary Based on a Scarce Image Generation Model and Lightweight Perception Architecture. *IEEE Trans. Circuits Syst. Video Technol.* **2026**, *36*, 1377–1391. [\[CrossRef\]](#)
26. Shajeena, J.; Govindasamy, B.; Gnanasundaram, M.; Joel, M.R. Mobile-Le Harmonic Fusion Network for Object Recognition and SiamMoT Based Multi-Object Tracking Using Video Surveillance. *Cybern. Syst.* **2026**, *57*, 866–896. [\[CrossRef\]](#)
27. Chen, Z.; Guo, H.; Yang, J.; Jiao, H.; Feng, Z.; Chen, L.; Gao, T. Fast vehicle detection algorithm in traffic scene based on improved SSD. *Measurement* **2022**, *201*, 111655. [\[CrossRef\]](#)
28. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 3051–3068. [\[CrossRef\]](#)
29. Pan, H.; Hong, Y.; Sun, W.; Jia, Y. Deep Dual-Resolution Networks for Real-Time and Accurate Semantic Segmentation of Traffic Scenes. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3448–3460. [\[CrossRef\]](#)
30. Xu, J.; Xiong, Z.; Bhattacharyya, S.P. PIDNet: A Real-time Semantic Segmentation Network Inspired by PID Controllers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2023; pp. 19529–19539.
31. Chen, Z.; Yang, J.; Feng, Z.; Zhu, H. RailFOD23: A dataset for foreign object detection on railroad transmission lines. *Sci. Data* **2024**, *11*, 72. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 7132–7141.

33. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2018; pp. 3–19.
34. Yang, J.; Jiang, Y.; Jiang, D.; Chen, Z. Infrared–Visible Fusion via Cross-Modality Attention and Small-Object Enhancement for Pedestrian Detection. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 477. [[CrossRef](#)]
35. Sreekala, K.; Maniraj, S.P.; Singh, A.; Singh, A.P.; Pyngkodi, M.; Inthiyaz, S. Enhancing Medical Diagnosis through Multimodal Image Fusion: A Novel Approach Using Modified Swin-Based Cross Attention Fusion. *Cybern. Syst.* **2026**, *57*, 765–807. [[CrossRef](#)]
36. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large Kernel Matters: Improve Semantic Segmentation by Global Convolutional Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2017; pp. 4353–4361.
37. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. CCNet: Criss-Cross Attention for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2019; pp. 603–612.
38. Guo, M.H.; Lu, C.Z.; Hou, Q.; Liu, Z.; Cheng, M.M.; Hu, S.M. SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 1140–1156. [[CrossRef](#)]
39. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*; Curran Associates, Inc.: Red Hook, NY, USA, 2021.
40. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.S.; et al. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2021; pp. 6881–6890.
41. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2021; pp. 7262–7272.
42. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
43. Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *Proceedings of the International Conference on Machine Learning*, PMLR; JMLR: New York, NY, USA, 2020; pp. 5156–5165.
44. You, K.; Gu, Y.; Shao, H.; Wang, Y. A liquid-impulse neural network model based on heterogeneous fusion of multimodal information for interpretable rotating machinery fault diagnosis. *Mech. Syst. Signal Process.* **2026**, *246*, 113923. [[CrossRef](#)]
45. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 4510–4520.
46. Woo, S.; Debnath, S.; Hu, R.; Chen, X.; Liu, Z.; Kweon, I.S.; Xie, S. ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2023; pp. 16133–16142.
47. Xu, M.; Zhang, Z.; Wei, F.; Hu, H.; Bai, X. Side Adapter Network for Open-Vocabulary Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2023; pp. 2945–2954.
48. Zhu, C.; Suri, S.; Jose, C.; Oquab, M.; Szafraniec, M.; Wen, W.; Xiong, Y.; Labatut, P.; Bojanowski, P.; Krishnamoorthi, R.; et al. Efficient Universal Perception Encoder. *arXiv* **2026**, arXiv:2603.22387.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.