LiteViLNet: Lightweight Vision-LiDAR Fusion Network for Efficient Road Segmentation

Daojie Peng, Bingtao Wang, Fulong Ma, Liang Zhang, Jun Ma†

***Abstract*— Road segmentation is a fundamental perception task for autonomous driving and intelligent robotic systems, requiring both high accuracy and real-time inference, especially for deployment on resource-constrained edge devices. Existing multi-modal road segmentation methods often rely on heavy transformer-based encoders to achieve state-of-the-art performance, but their enormous computational cost prohibits real-time deployment on embedded platforms. To address this dilemma, we propose LiteViLNet, a lightweight multi-modal network that fuses RGB texture information and LiDAR geometric information for efficient road segmentation. Specifically, we design a dual-stream lightweight encoder and depth-wise separable convolutions to extract hierarchical features from both modalities with minimal parameters. We further propose a Multi-Scale Feature Fusion Module (MSFM) to facilitate cross-modal interaction at different levels, and a large-kernel-bridge module to capture long-range dependencies with linear complexity. Extensive experiments on the KITTI Road dataset and real-world applications demonstrate that LiteViLNet achieves a promising balance between accuracy and efficiency. Notably, with only 14.04M parameters, our model attains a 96.36% MaxF score, ranking the best among all CNN-based methods and being comparable to larger transformer-based models, and runs at 163.79 FPS in model-only inference on RTX 4060 Ti (22.18 FPS on Jetson Orin NX). It outperforms numerous heavy-weight methods in inference speed while maintaining highly competitive accuracy, fully validating the potential of LiteViLNet for real-time embedded deployment in autonomous driving and intelligent robotics.**

I. Introduction

Drivable area segmentation, which aims to identify the free road surface from the surrounding environment, is a critical component for autonomous vehicles and intelligent robotics systems [1]. This task serves as the foundation for trajectory planning, obstacle avoidance, and navigation. Recent work on road segmentation is extensive, ranging from single-modal approaches [2], [3] to multi-modal fusion architectures [4], [5], [6], [7], and from fully self-supervised to annotation-free solutions [8], [9]—all of which have made notable strides. However, deploying high-performance segmentation models on edge devices remains challenging due to the strict constraints on computation, memory, and power consumption. This challenge has driven extensive research on lightweight perception algorithms for edge deployment in recent years [10], [11].

† Corresponding author: jun.ma@ust.hk

Daojie Peng, Fulong Ma and Jun Ma are with The Hong Kong University of Science and Technology (Guangzhou) (e-mail: {fmaaf, dpeng108}@connect.hkust-gz.edu.cn, jun.ma@ust.hk.)

Bingtao Wang and Liang Zhang are with The Shandong University, wangbt@mail.sdu.edu.cn, 201299800013@sdu.edu.cn

Traditional single-modal road segmentation methods, which only use RGB images, often suffer from poor robustness under challenging conditions such as low light, shadows, or texture-less surfaces [12]. To overcome these limitations, multi-modal fusion has become a popular trend. By combining RGB images with LiDAR data, which provides accurate 3D geometric information, these methods can significantly improve the segmentation accuracy. For instance, SNE-RoadSegV2 [13] and RoadFormer [14] leverage transformer architectures to fuse RGB and normal/depth features, achieving impressive performance on the KITTI Road benchmark. Additionally, some methods [5], [15] pre-process LiDAR point cloud to generate Altitude Difference Image (ADI), which are then used as model inputs. This approach preserves the advantages of LiDAR point cloud while accelerating processing speed. However, these methods usually rely on heavy backbones like Swin Transformer [16], resulting in models with hundreds of millions of parameters. Such large models are computationally expensive and cannot run in real-time on embedded platforms, which are commonly used in practical applications.

On the other hand, lightweight segmentation methods aim to reduce the model size and computational cost. Recent works such as TwinLiteNet+ [17] and the Knowledge Generation and Distillation (KGD) framework [18] have attempted to build efficient models for edge deployment, achieving promising results. Methods like LRDNet [19] and USNet [20] have also made early attempts in this direction. However, most of these lightweight designs either sacrifice too much accuracy or fail to fully exploit the complementary information from multi-modal data. For example, simple concatenation or addition of features from different modalities cannot effectively model the complex cross-modal interactions, limiting the final performance. Similarly, SDFNet [21] and LCIRE-Net [22] propose lightweight dual-stream networks, but they still struggle with the trade-off between cross-modal interaction capability and computational efficiency. Therefore, it remains an open problem how to design an efficient multi-modal network that can achieve high accuracy while satisfying the real-time requirements of edge deployment.

To this end, we propose LiteViLNet, a Lightweight Vision-LiDAR Network for efficient multi-modal road segmentation. The overall architecture is illustrated in Fig. 1. Our key insight is to maintain the high representational capacity of multi-modal fusion while drastically reducing the computational overhead through carefully designed lightweight modules. Specifically, we first build a dual-stream encoder where

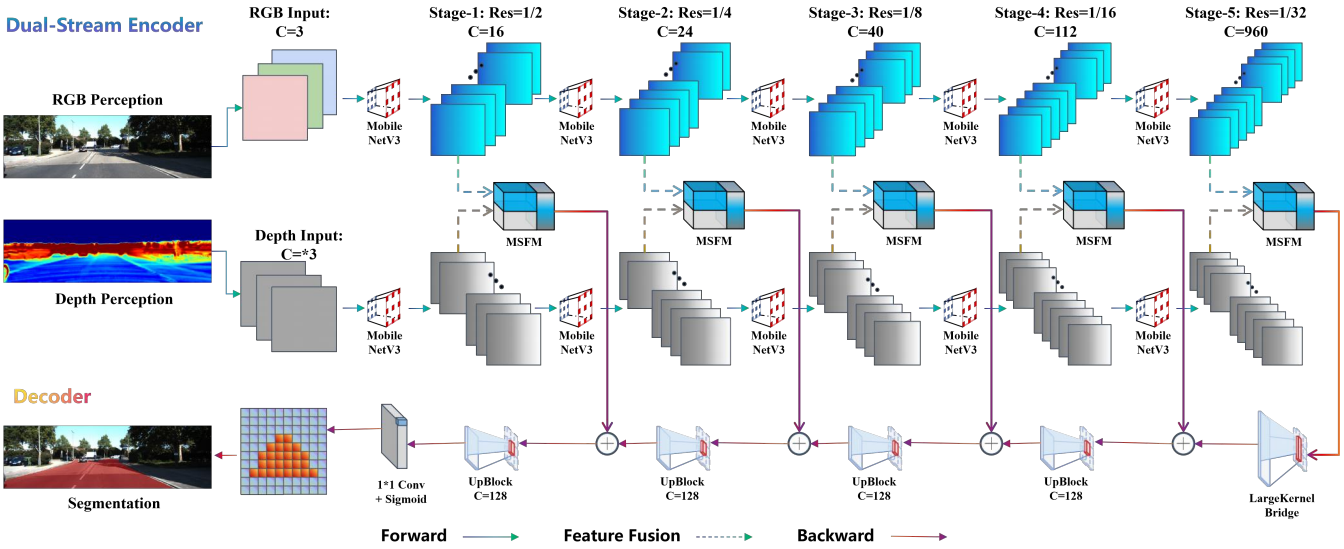

Fig. 1: **Overall Architecture of LiteViLNet.** The network consists of a dual-stream lightweight encoder, a multi-scale feature fusion module, a large-kernel-bridge module, and a decoder with deep supervision.

the RGB stream uses a pre-trained MobileNetV3-Large [23] backbone, and the LiDAR stream uses a tiny encoder based on depth-wise separable convolutions. This allows us to extract multi-scale features from both modalities with very few parameters. Then, we propose a Multi-Scale Feature Fusion Module (MSFM) to perform cross-modal attention interaction at each scale, enabling the network to adaptively fuse the texture and geometric information. To capture global context without the quadratic cost of self-attention, we design a large-kernel-bridge module that uses large kernel depth-wise convolution to enlarge the receptive field with linear complexity. Finally, we adopt a deep supervision strategy to facilitate the training of the multi-scale network.

Our main contributions are summarized as follows:

- We propose LiteViLNet, a lightweight multi-modal network specifically designed for efficient road segmentation. It achieves an excellent balance between accuracy and inference efficiency, which is critical for real-time deployment on resource-constrained edge devices and robotic platforms.
- We design a suite of efficient, synergistic modules to fully exploit complementary RGB and LiDAR information with minimal computational and parameter overhead, including a dual-stream lightweight encoder for modality-specific feature extraction, a multi-scale cross-modal fusion module (MSFM) for effective cross-modal interaction, and a large-kernel semantic enhancement module (large-kernel-bridge) for improved semantic representation.
- We propose an ADI generation scheme compatible with RGB-D cameras, which can replace LiDAR to reduce deployment costs and further enhance the practical application value of our model in real-world scenarios.
- We conduct extensive experiments on the KITTI Road dataset and real-world robotic deployments (including Kuafu Delivery Vehicle, Unitree-B2, and Unitree-G1). The results demonstrate that LiteViLNet achieves 96.36% MaxF with only 14.04M parameters and 163.79 FPS model-only inference speed on RTX 4060 Ti (22.18 FPS on Jetson Orin NX), significantly outperforming existing lightweight methods in speed while maintaining competitive accuracy.

II. Related Works

A. Road Segmentation

Road segmentation has been extensively studied over the past decades. Early works primarily rely on hand-crafted features, such as color and texture [24]. With the rise of deep learning, Fully Convolutional Networks (FCNs) [25] and encoder-decoder architectures like U-Net [26] have become the standard paradigm. Recent works have further improved the performance by introducing more powerful backbones. For example, SNE-RoadSeg [4] introduces surface normal estimation to help road segmentation. Later, SNE-RoadSegV2 [13] extended this work by using Swin Transformer as the backbone, achieving state-of-the-art results. RoadFormer [14] further proposes a transformer-based fusion framework to model the cross-modal interactions. However, these methods usually require heavy computation, making them unsuitable for real-time applications.

B. Multi-Modal Fusion for Segmentation

Multi-modal fusion has been proven effective for improving segmentation performance. By combining RGB images with other modalities such as depth, LiDAR, or thermal images, the network can leverage complementary information to handle challenging scenarios. Early fusion methods simply concatenate the input channels [27]. Later works propose more sophisticated fusion strategies. CMX [12] proposes a cross-modal feature calibration module for RGB-X segmentation. Cross-view transformers [28] have also

been proposed to model the interactions between different modalities. However, most of these advanced fusion methods are designed for high-performance servers and introduce significant computational overhead, which is not affordable for edge devices. In contrast, our work focuses on lightweight fusion mechanisms that can run efficiently on embedded platforms.

C. Lightweight Semantic Segmentation

To enable deployment on edge devices, many lightweight segmentation networks have been proposed. MobileNet [29] and ShuffleNet [30] introduce depth-wise separable convolutions to reduce the computation cost. Based on these backbones, methods like ESPNet [31] and BiSeNet [32] have been proposed for real-time semantic segmentation. For the specific task of road segmentation, LRDNet [19] proposes a lightweight network for LiDAR-assisted road detection. USNet [20] designs an uncertainty-aware symmetric network for fast RGB-D road segmentation. However, these methods either use simple fusion strategies or only focus on single-modal input. Our work differs from them by proposing a dedicated lightweight multi-modal fusion framework that can effectively combine RGB and LiDAR information with minimal parameters.

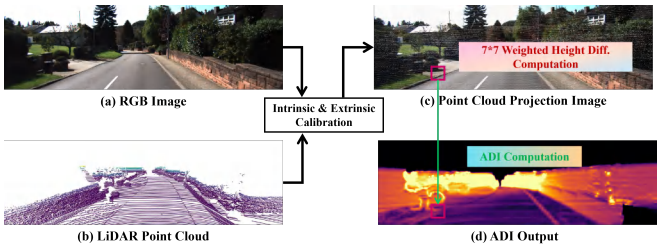

Fig. 2: **Illustration of the ADI Generation Pipeline.** This process converts the raw 3D LiDAR point cloud into a 2D geometric feature map, which encodes the local height difference between the ground plane and obstacles to provide strong geometric cues for road segmentation.

III. Method

In this section, we present the details of the proposed LiteViLNet framework. The overall architecture is illustrated in Fig. 1.

A. Input Representation

Given an RGB image $I \in \mathbb{R}^{H\times W\times 3}$ and a 3D LiDAR point cloud $\mathbf{P} = \{(x_i, y_i, z_i)\}$, we first generate the Altitude Difference Image (ADI) [5] $\mathbf{A} \in \mathbb{R}^{H\times W}$ as input for the LiDAR stream. The process consists of two main steps as illustrated in Fig. 2: **1)** Point Cloud Projection: Using the camera-LiDAR calibration parameters, we project each 3D LiDAR point onto the 2D image plane via the projection matrix $\mathbf{K}[\mathbf{R}|\mathbf{t}]$, obtaining a sparse set of projected points (u, v) with corresponding height $z_{u,v}$. **2)** Weighted Altitude Difference Calculation: For each valid projected point (u, v), we compute the weighted height difference within a $K \times K$ neighborhood $\mathcal{N}(u, v)$:

$$V_{u,v} = \frac{1}{M} \sum_{(u',v')\in\mathcal{N}(u,v)} \frac{|z_{u,v} - z_{u',v'}|}{\sqrt{(u'-u)^2 + (v'-v)^2}} \quad (1)$$

where $\mathcal{N}(u, v)$ is the set of valid projected points in the neighborhood, M is the number of valid neighbors, and z denotes the height value of each point.

All values $V_{u,v}$ are normalized to the range $[0, 255]$ to form a single-channel grayscale image, which encodes the geometric height difference between road and non-road regions. Before feeding into the encoder, the ADI is replicated across three channels to obtain $\mathbf{A}' \in \mathbb{R}^{H\times W\times 3}$, matching the input interface of the RGB encoder.

B. Dual-Stream Lightweight Encoder

We use a dual-stream architecture to extract features from the RGB and LiDAR modalities separately.

- **RGB Stream**: We adopt MobileNetV3-Large [23] as the backbone for the RGB stream. This network is pre-trained on ImageNet and provides an excellent trade-off between accuracy and efficiency. It consists of five stages, generating a feature pyramid $F_l^{rgb}{}_{l=0}^{4}$ with resolutions of $1/2, 1/4, 1/8, 1/16, 1/32$ of the input size, respectively.
- **LiDAR Stream**: For the LiDAR stream, we design a tiny encoder using only depth-wise separable convolutions (DSConv). This encoder has exactly the same structure and channel configuration as the RGB stream, ensuring that the features from both modalities are aligned at each scale. The total parameters of the LiDAR encoder are only about 0.12M, which introduces negligible overhead to the whole model.

C. Multi-Scale Feature Fusion Module

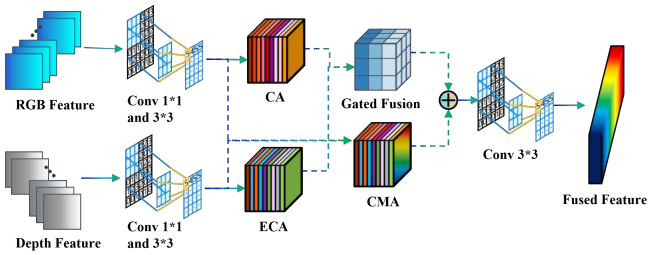

Fig. 3: **Overall Architecture of the Proposed MSFM.** It sequentially conducts channel dimension compression, intra-modal feature enhancement via ECA and coordinate attention, bidirectional cross-modal attention interaction, and adaptive gated feature fusion to effectively integrate complementary RGB texture and LiDAR geometric information at individual feature scales.

To effectively fuse the features from the two modalities, we propose the Multi-Scale Feature Fusion Module (MSFM). As shown in Fig. 3, we apply this module at each of the five scales to perform deep cross-modal interaction between RGB

texture features and LiDAR geometric features. For each scale l, given the RGB feature $F_l^{rgb} \in \mathbb{R}^{B\times C_l\times H_l\times W_l}$ and LiDAR feature $F_l^{lidar} \in \mathbb{R}^{B\times C_l\times H_l\times W_l}$, the fusion process is defined as follows:

1) **Dimension Reduction**: We first use a 1×1 convolution to reduce the channel dimension of both features from C_l to $C_l/2$, reducing the computational cost of the subsequent attention calculation:

$$F_l^{rgb'} = \text{Conv}_{1\times1}\left(F_l^{rgb}\right),$$
$$F_l^{lidar'} = \text{Conv}_{1\times1}\left(F_l^{lidar}\right)$$

2) **Intra-modal Enhancement**: We apply Efficient Channel Attention (ECA) [33] to the dimension-reduced RGB features to enhance the channel-wise dependencies, and Coordinate Attention (CA) [34] to the LiDAR features to capture the spatial position information. ECA captures local cross-channel interactions via 1D adaptive convolution, avoiding the information loss caused by dimension reduction in SE modules; CA decomposes channel attention into horizontal and vertical positional encoding, enabling the LiDAR features to capture precise spatial awareness while remaining lightweight:

$$\hat{F}_l^{rgb} = \text{ECA}\left(F_l^{rgb'}\right),$$
$$\hat{F}_l^{lidar} = \text{CA}\left(F_l^{lidar'}\right)$$

3) **Cross-modal Attention (CMA)**: We perform bidirectional cross-modal attention which allows each modality to attend to the informative regions of the other modality, and the interaction features are obtained by element-wise addition of the two directional attention outputs. Let $d = C_l/2$ be the channel dimension after projection:

$$F_l^{cross} = \text{softmax}\left(\frac{Q_{rgb}K_{lidar}^\top}{\sqrt{d}}\right)V_{lidar} + \text{softmax}\left(\frac{Q_{lidar}K_{rgb}^\top}{\sqrt{d}}\right)V_{rgb}$$

where $Q_{rgb}, K_{rgb}, V_{rgb}$ are linear projections of $\hat{F}_l^{rgb}$, and $Q_{lidar}, K_{lidar}, V_{lidar}$ are projections of $\hat{F}_l^{lidar}$. Softmax is normalized along the spatial dimension.

4) **Gated Fusion**: Finally, we use a learnable gate mechanism to dynamically fuse the cross-modal interaction features with the original features. The gate weights are learned based on the concatenation of the input features, allowing the network to adaptively adjust the fusion ratio according to the input content. We then recover the channel dimension to C_l via a 1×1 convolution to generate the final fused feature:

$$F_l^{fused} = \text{Conv}_{1\times1}\left(G_F\right)$$
$$G_F = g_l \odot F_l^{cross} + (1-g_l)\odot\left(\hat{F}_l^{rgb} + \hat{F}_l^{lidar}\right)$$

where $g_l = \sigma\left(\text{Conv}_{1\times1}\left(\left[\hat{F}_l^{rgb}; \hat{F}_l^{lidar}; F_l^{cross}\right]\right)\right)$ is the gate weight, $\sigma(\cdot)$ is the Sigmoid function, $[\cdot;\cdot;\cdot]$ denotes channel-wise concatenation, and $\odot$ is element-wise multiplication.

This process produces the fused feature F_l^{fused} for each scale, which contains both the texture information from RGB and the geometric information from LiDAR.

D. Large-Kernel-Bridge Semantic Enhancement

After fusion, the deepest feature F_4^{fused} has the highest semantic level but the smallest spatial resolution (only 1/32 of the input size). To capture the global context without using expensive self-attention, we propose the large-kernel-bridge module. Inspired by recent works on large kernel convolutions [35], we use a 7×7 depth-wise separable convolution to enlarge the receptive field with linear complexity, which is significantly cheaper than the transformer-based bridge which has $O(N^2)$ complexity.

The module adopts a bottleneck structure to control the parameters, and the forward process is defined as:

$$F_4^{enh} = F_4^{fused} + \text{Conv}_{1\times1}^{up}\left(\text{Drop}\left(D_F\right)\right)$$
$$D_F = \text{GELU}\left(\text{DWConv}_{7\times7}\left(\text{Conv}_{1\times1}^{down}\left(F_4^{fused}\right)\right)\right)$$

where $\text{Conv}_{1\times1}^{down}$ reduces the channel dimension from 960 to 128, $\text{DWConv}_{7\times7}$ is the 7×7 depth-wise convolution, $\text{GELU}(\cdot)$ is the Gaussian Error Linear Unit activation, $\text{Drop}(\cdot)$ is the Dropout2d regularization with a dropout rate of $p = 0.2$, and $\text{Conv}_{1\times1}^{up}$ recovers the channel dimension back to 960. The residual connection ensures stable gradient propagation.

This module only adds 0.26M parameters in total, which introduces negligible overhead to the whole network. Compared with the standard Transformer bridge which requires 24.85M parameters, our large-kernel-bridge achieves better performance with far fewer parameters and computational cost.

E. Decoder and Deep Supervision

We use a U-Net style decoder to recover the spatial resolution. The decoder takes the enhanced deepest feature as input and gradually upsamples it, concatenating with the fused features from the encoder via skip connections. The decoder consists of one bottleneck layer and four UpBlocks, with channel configuration [128, 64, 32, 16, 16].

The bottleneck layer first reduces the channel dimension of F_4^{enh} from 960 to 128 via a 1×1 convolution. Then, each UpBlock performs upsampling and feature fusion sequentially, defined as:

$$D_l = \text{DoubleConv}\left(\left[\text{Up}\left(D_{l+1}\right); \text{Conv}_{1\times1}\left(F_l^{fused}\right)\right]\right)$$

where $\text{Up}(\cdot)$ is the bilinear upsampling to match the spatial resolution of the skip connection feature, $\text{Conv}_{1\times1}$ adjusts the channel dimension of the skip connection feature to match the current decoder layer, $[\cdot;\cdot]$ is channel-wise concatenation, and DoubleConv consists of two cascaded 3×3 convolution-BatchNorm-ReLU blocks.

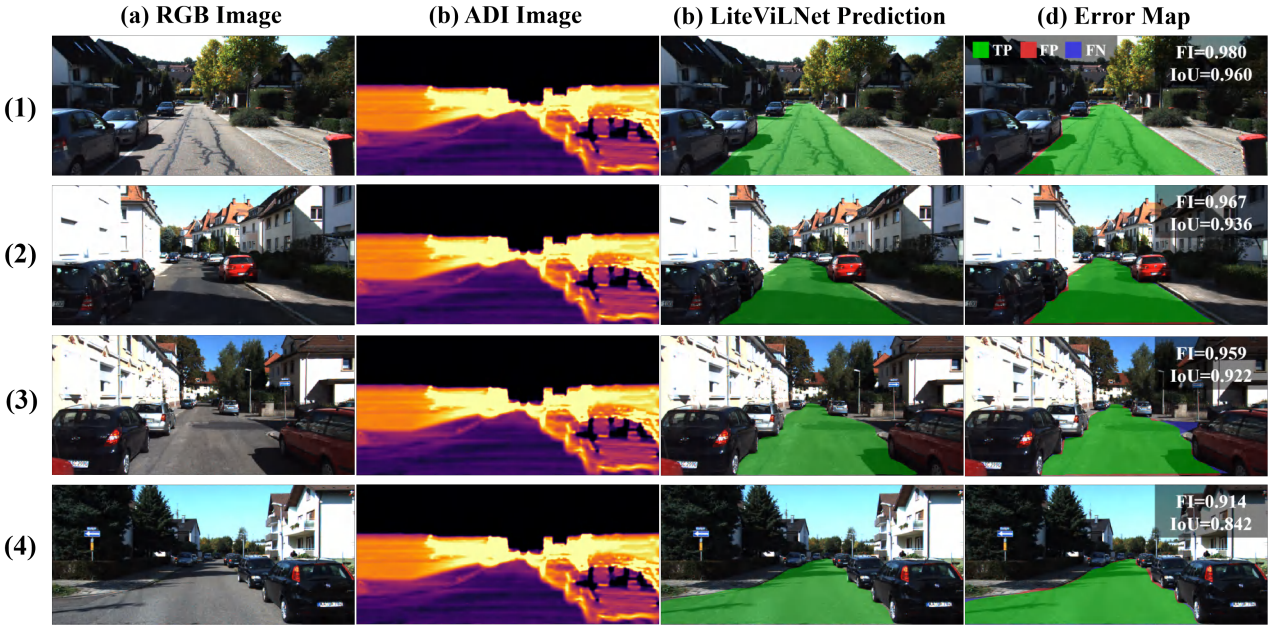

Fig. 4: **Qualitative Segmentation Results on the KITTI Road Validation Set.** Each row shows (a) the input RGB image, (b) the corresponding Altitude Difference Image (ADI) derived from LiDAR depth data, (c) the segmentation prediction of LiteViLNet, and (d) the error map visualizing true positives (TP, green), false positives (FP, red), and false negatives (FN, blue). Quantitative metrics including F1-score and IoU are reported for each example. Our method can accurately segment the road area in various scenarios, with errors mainly concentrated on the boundaries.

To facilitate the training of the deep network, we adopt a deep supervision strategy. We add auxiliary segmentation heads to the intermediate layers of the decoder. These heads provide additional gradient signals during training, helping the network to learn better multi-scale features. They are removed during inference, so they do not introduce any overhead.

The total loss function is a combination of the main loss and the auxiliary losses:

$$\mathcal{L}_{total} = \mathcal{L}_{main}(\hat{Y}, Y) + \sum_{k=1}^{3} w_k \cdot \mathcal{L}_{aux}(\hat{Y}_k^{aux}, Y^{\downarrow_k})$$

where Y is the ground-truth label, $Y^{\downarrow_k}$ is the downsampled ground-truth corresponding to the auxiliary prediction resolution, and $w_k = [0.5, 0.3, 0.2]$ are the decreasing weights for the auxiliary losses from deep to shallow layers.

The main loss $\mathcal{L}_{main}$ is a combination of three components:

$$\mathcal{L}_{main} = \mathcal{L}_{BCE} + \mathcal{L}_{Lovász} + 0.5 \cdot \mathcal{L}_{Focal}$$

where $\mathcal{L}_{BCE}$ is the binary cross-entropy loss providing basic pixel-level classification supervision; $\mathcal{L}_{Lov'asz}$ is the Lovász-Hinge loss [36], which converts the discrete IoU metric into a continuous differentiable surrogate loss to directly optimize the segmentation quality; $\mathcal{L}_{Focal}$ is the Focal loss [37] with $\alpha = 0.25$ and $\gamma = 2$, which down-weights the easy samples and forces the network to focus on hard regions such as road boundaries. The auxiliary losses $\mathcal{L}_{aux}$ adopt the same combination form as the main loss.

IV. Experiments

A. Experimental Setup

Dataset: We evaluate our method on the KITTI Road dataset [38], which is the most widely used benchmark for road segmentation. It contains 289 images with synchronized RGB and LiDAR data. We follow the standard split, using 231 images for training and 58 for validation.

Metrics: We use the standard metrics for KITTI Road, including Maximum F1-measure (MaxF), Precision (PRE), Recall (REC), and Intersection over Union (IoU). We also report the number of parameters and the inference speed (FPS).

Implementation Details: We train the model using the AdamW optimizer with an initial learning rate of $2e^{-4}$ using 1 A100 GPU. We use a cosine annealing learning rate scheduler with 150 epochs. The input size is 1248×384. Experiments are conducted on an NVIDIA RTX 4060 Ti GPU as well as Jetson Orin NX. Unless otherwise specified, FPS is measured as FP16 model-only inference with batch size 1 at 384×1248, excluding preprocessing and post-processing.

B. Comparison with State-of-the-Art Methods

We compare our method with state-of-the-art road segmentation methods. The results are shown in Table I. The accuracy metrics are taken from the corresponding papers or official reports, while Params and FPS are measured locally when reproducible under the stated protocol. From the results, we can observe that:

TABLE I: Performance Comparison on the KITTI Road Validation Set. FPS-1: RTX 4060 Ti Linux FP16 model-only benchmark. FPS-2: Jetson Orin NX FP16 model-only benchmark. Both FPS columns use 384×1248 inputs with batch size 1 and exclude preprocessing. All numeric FPS entries are measured with checkpoint-loaded models when local implementations and weights are available. † denotes SNE-RoadSeg local Params/FPS measured with an official-checkpoint-compatible wrapper because the released checkpoint keys differ from the current public source.

Method	Encoder	Input	MaxF (%) ↑	PRE (%) ↑	REC (%) ↑	Params (M) ↓	FPS-1 ↑	FPS-2 ↑
Transformer-based Methods								
SNE-RoadSegV2 [13]	Swin-B×2	RGB+Normal	97.08	96.83	97.34	205.80	-	-
RoadFormer [14]	Swin-T	RGB+Depth	97.02	96.61	97.43	206.80	-	-
CNN-based Methods								
USNet [20]	ResNet-18	RGB+Depth	96.11	95.86	96.37	30.74	104.43	6.26
LRDNet [19]	VGG-19	RGB+LiDAR	96.18	95.94	**96.42**	28.57	9.28	1.73
PLARD [5]	ResNet-101	RGB+LiDAR	95.95	96.25	95.65	76.93	17.10	3.52
SNE-RoadSeg [4]	RN-50×2†	RGB+Normal	96.03	96.22	95.83	132.06†	13.33†	2.70†
LiteViLNet (Ours)	MobileNetV3-L	RGB+LiDAR	**96.36**	**96.79**	95.85	**14.04**	**163.79**	**22.18**

- Our method achieves the best efficiency. With only 14.04M parameters, it runs at 163.79 FPS model-only inference on RTX 4060 Ti and 22.18 FPS on Jetson Orin NX under PyTorch FP16. With TensorRT FP16 deployment, the Jetson Orin NX speed can further increase to 68.73 FPS. For example, under the RTX 4060 Ti PyTorch model-only setting, LiteViLNet is 1.57 times faster than USNet and more than 9.5 times faster than PLARD.
- In terms of accuracy, our LiteViLNet achieves the best performance among all CNN-based methods, and is only slightly inferior to the larger Transformer-based state-of-the-art approaches with a negligible performance gap. This demonstrates that our method can obtain comparable accuracy to heavy-weight models while using significantly fewer parameters and achieving much higher inference efficiency.
- Our method achieves a very high precision of 96.79%, which is comparable to the best methods. This means that our model has a very low false positive rate, which is crucial for safety-critical applications.

The qualitative results are shown in Fig. 4. Our method consistently generates smooth and accurate road boundaries across diverse urban scenarios, with errors primarily localized to challenging edge cases such as occlusions and shadowed regions.

C. Ablation Study

To verify the effectiveness of each component in LiteViLNet, we conduct an ablation study. The results are shown in Table II from which we can draw the following conclusions:

1) The LiDAR encoder is the most cost-effective component. Adding it only increases the parameters by 0.12M but improves the MaxF by 0.51%, demonstrating the effectiveness of the geometric information.
2) The large-kernel-bridge module brings a significant improvement of 0.38% with only 0.26M parameters, verifying the effectiveness of the large kernel design for global context modeling.
3) Deep supervision further improves the performance by 0.12% without adding any inference parameters, which helps the training of the multi-scale network.
4) Although MSFM alone may cause overfitting on the small dataset, it works synergistically with the other modules. When combined with Bridge and DeepSup, it enables the full model to achieve the best performance.

TABLE II: Ablation Study Results of LiteViLNet.

LiDAR	MSFM	Bridge	DeepSup	MaxF (%)	Params (M)
-	-	-	-	95.51	3.31
✓	-	-	-	96.02	3.43
✓	✓	-	-	95.77	13.78
✓	✓	✓	-	96.15	14.04
✓	✓	✓	✓	96.27	14.04

D. Real-World Experiments

To validate the practical deployment capability of LiteViLNet on embedded platforms and embodied intelligent systems, we conduct real-world experiments on three representative robotic platforms: a Kuafu delivery vehicle, the Unitree-B2 quadruped robot and a Unitree-G1 humanoid robot, as illustrated in Fig. 5. All experiments are performed on the NVIDIA Jetson Orin NX computing platform, which is a widely used edge-computing module for mobile robots. For sensing, we employ the Orbbec Gemini 336L RGB-D camera, which provides synchronized RGB frames and dense depth maps that serve as a substitute for LiDAR point clouds in real-world scenarios.

The input resolution is configured to 1280×800, and the camera is running at 15 FPS to balance perception quality and system latency. This 15 FPS denotes the real-world camera and system operating rate, whereas Table I reports model-only inference speed under the benchmark protocol. The depth stream is converted into an ADI using the same pipeline described in Section III-A, providing geometric cues for road segmentation without requiring mechanical LiDAR hardware.

To verify the usability of our segmentation results, we implement a simple yet effective lane-centering control al-

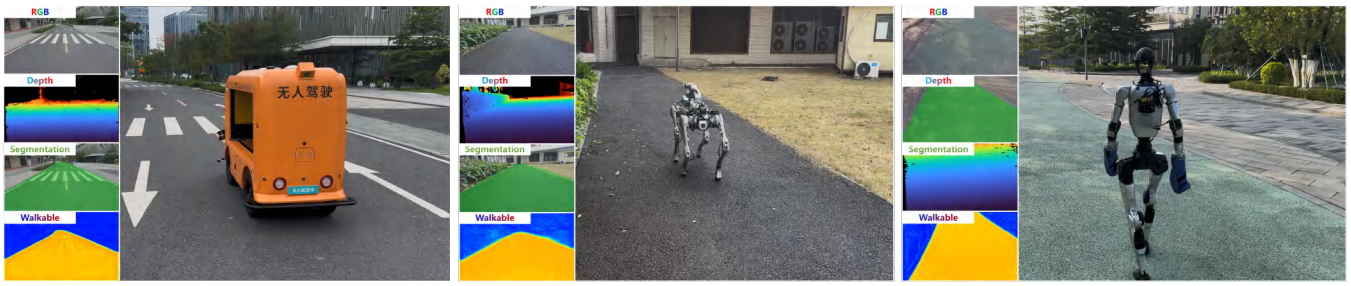

Fig. 5: **Real-world Deployment on Different Robots. LEFT: Kuafu Delivery Vehicle, MIDDLE: Unitree-B2, RIGHT: Unitree-G1.** Left column of each case shows the first-person perception pipeline of LiteViLNet: RGB image, depth map, drivable area segmentation mask, and walkable confidence heatmap. Right column shows the robot navigating autonomously using our lightweight perception system.

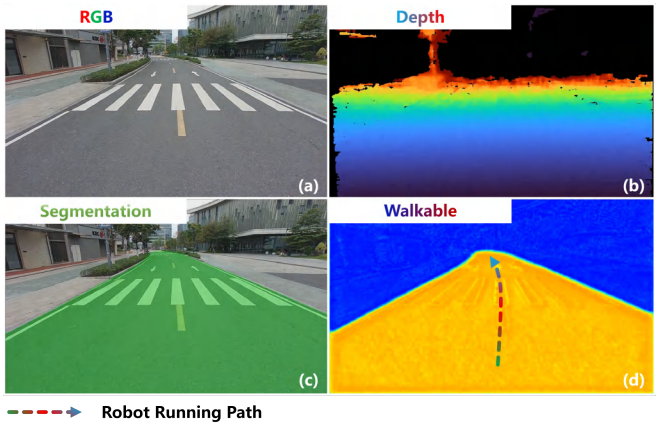

Fig. 6: **First-person Perception Pipeline of LiteViLNet on the Kuafu Delivery Vehicle.** The panels show: **(a)** raw RGB image from the Orbbec Gemini 336L camera, **(b)** corresponding depth map, **(c)** drivable area segmentation mask, and **(d)** walkable confidence heatmap overlaid with the planned robot trajectory. The bottom legend indicates the robot running path, demonstrating that LiteViLNet provides stable and accurate road perception for lane-centering navigation in real-world scenarios.

gorithm. As shown in Fig. 6, the system computes the road center line from the binary drivable-area mask output by LiteViLNet and generates velocity commands to keep the robot stably moving along the middle of the detected road.

Experimental results demonstrate that our lightweight LiteViLNet runs stably at the full camera frame rate on Jetson Orin NX, with low memory usage and consistent segmentation performance under real-world lighting, terrain, and occlusion conditions. The robot can smoothly navigate along the road without drifting or collision, confirming that the proposed method satisfies the accuracy, efficiency, and robustness requirements for real-world embodied AI deployment.

V. Conclusion

In this paper, we propose LiteViLNet, a lightweight Vision-LiDAR Network for efficient multi-modal road segmentation in autonomous driving. To resolve the performance-computation conflict in existing methods, we design three core components: a dual-stream lightweight encoder, MSFM, and large-kernel-bridge. The dual-stream encoder reduces overhead while extracting RGB/LiDAR features; MSFM enables cross-modal interaction; large-kernel-bridge enhances semantics without excessive computation.

KITTI Road experiments confirm LiteViLNet's superiority: 96.36% MaxF with 14.04M parameters, running at 163.79 FPS (RTX 4060 Ti) and 22.18 FPS (Jetson Orin NX). It outperforms state-of-the-art lightweight methods in efficiency with competitive accuracy, resolving the computational bottleneck and verifying its suitability for edge deployment—further validated by its real-world deployment on different robots, including Kuafu Delivery Vehicle, Unitree-B2, and Unitree-G1.

In summary, LiteViLNet addresses the accuracy-efficiency balance challenge, providing a practical solution for real-time edge deployment. Future work will extend it to challenging scenarios and explore self-supervised learning to enhance robustness and generalization.

References

[1] S. Mozaffari, O. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2022.

[2] T. Sun et al., "Rod: Rgb-only fast and efficient off-road freespace detection," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2025, pp. 9787–9793.

[3] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2019, pp. 4213–4220.

[4] R. Fan, H. Wang, P. Cai, and M. Liu, "Sne-roadseg: Incorporating surface normal information into semantic segmentation for accurate freespace detection," in *European Conference on Computer Vision*, Springer, 2020, pp. 340–356.

[5] Z. Chen, J. Zhang, and D. Tao, "Progressive LiDAR adaptation for road detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 693–702, 2019.

[6] C. Min et al., "Orfd: A dataset and benchmark for off-road freespace detection," in *2022 international conference on robotics and automation (ICRA)*, IEEE, 2022, pp. 2532–2538.

[7] G. Zhao, F. Ma, W. Qi, Y. Liu, M. Liu, and J. Ma, "Curbnet: Curb detection framework based on lidar point cloud segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[8] F. Ma, Y. Liu, S. Wang, J. Wu, W. Qi, and M. Liu, "Self-supervised drivable area segmentation using lidar's depth information for autonomous driving," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 41–48.

[9] F. Ma, D. Peng, and J. Ma, "Annotation-free detection of drivable areas and curbs leveraging lidar point cloud maps," *arXiv preprint arXiv:2603.27553*, 2026.

[10] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "Pidnet: A real-time semantic segmentation network inspired by pid controllers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 529–19 539.

[11] D. Peng, J. Cao, Q. Zhang, and J. Ma, "Lovon: Legged open-vocabulary object navigator," *arXiv preprint arXiv:2507.06747*, 2025.

[12] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14 679–14 694, 2023.

[13] Y. Feng et al., "Sne-roadsegv2: Advancing heterogeneous feature fusion and fallibility awareness for freespace detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–9, 2025.

[14] J. Li, Y. Zhang, P. Yun, G. Zhou, Q. Chen, and R. Fan, "Roadformer: Duplex transformer for rgb-normal semantic road scene parsing," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 7, pp. 5163–5172, 2024.

[15] F. Ma, P. Hou, Y. Liu, Y. Liu, M. Liu, and J. Ma, "Annotation-free curb detection leveraging altitude difference image," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2025, pp. 762–768.

[16] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9992–10 002.

[17] Q.-H. Che, D.-T. Le, M.-Q. Pham, V.-T. Nguyen, and D.-K. Lam, "Twinlitenet+: An enhanced multi-task segmentation model for autonomous driving," *Computers and Electrical Engineering*, vol. 128, p. 110 694, 2025.

[18] M. Li, J. Wang, and H. Chen, "Knowledge generation and distillation for road segmentation in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[19] A. A. Khan, J. Shao, Y. Rao, L. She, and H. T. Shen, "Lrdnet: Lightweight lidar aided cascaded feature pools for free road space detection," *IEEE Transactions on Multimedia*, vol. 27, pp. 652–664, 2025.

[20] Y. Chang, F. Xue, F. Sheng, W. Liang, and A. Ming, "Fast road segmentation via uncertainty-aware symmetric network," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 11 124–11 130.

[21] Y. Cao and H. Qu, "Sdfnet for real-time semantic segmentation on urban road images," *IAENG International Journal of Computer Science*, vol. 52, no. 12, pp. 4815–4821, 2025.

[22] Y. Duan et al., "Lcire-net: Lightweight cross-modal information interaction for road feature extraction from remote sensing images and gps trajectory/lidar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–18, 2025.

[23] A. Howard et al., "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[24] J. M. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 184–193, 2011.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.

[27] F. Wulff, B. Schaufele, O. Sawade, D. Becker, B. Henke, and I. Radusch, "Early fusion of camera and lidar for robust road detection based on u-net fcn," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 1426–1431.

[28] B. Zhou and P. Krahenbuhl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 750–13 759.

[29] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[30] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 122–138.

[31] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 561–580.

[32] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 334–349.

[33] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 531–11 539.

[34] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 708–13 717.

[35] W. Wang et al., "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 408–14 419.

[36] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421.

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2999–3007.

[38] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.