Exploiting Low-level Representations for Ultra-Fast Road Segmentation

Huan Zhou, Feng Xue, Yucong Li, Shi Gong, Yiqun Li, Yu Zhou

Abstract—Achieving real-time and accuracy on embedded platforms has always been the pursuit of road segmentation methods. To this end, they have proposed many lightweight networks. However, they ignore the fact that roads are "stuff" (background or environmental elements) rather than "things" (specific identifiable objects), which inspires us to explore the feasibility of representing roads with low-level instead of high-level features. Surprisingly, we find that the primary stage of mainstream network models is sufficient to represent most pixels of the road for segmentation. Motivated by this, we propose a <u>L</u>ow-level <u>F</u>eature <u>D</u>ominated <u>Road</u> <u>S</u>egmentation network (LFD-RoadSeg). Specifically, LFD-RoadSeg employs a bilateral structure. The spatial detail branch is firstly designed to extract low-level feature representation for the road by the first stage of ResNet-18. To suppress texture-less regions mistaken as the road in the low-level feature, the context semantic branch is then designed to extract the context feature in a fast manner. To this end, in the second branch, we asymmetrically downsample the input image and design an aggregation module to achieve comparable receptive fields to the third stage of ResNet-18 but with less time consumption. Finally, to segment the road from the low-level feature, a selective fusion module is proposed to calculate pixel-wise attention between the low-level representation and context feature, and suppress the non-road low-level response by this attention. On KITTI-Road, LFD-RoadSeg achieves a maximum F1-measure (MaxF) of 95.21% and an average precision of 93.71%, while reaching 238 FPS on a single TITAN Xp and 54 FPS on a Jetson TX2, all with a compact model size of just 936k parameters. The source code is available at https://github.com/zhouhuan-hust/LFD-RoadSeg.

Index Terms—Road segmentation, real-time, low-level representation, selective fusion

I. INTRODUCTION

VISUAL road segmentation has become the fundamental scene understanding approach for autonomous driving and robots [1]–[10]. Although it was originally introduced more than 15 years ago, the improvement of embedded platforms and deep networks has enabled the deployment of road segmentation on autonomous driving systems only in recent years. Due to scarce computing resources, embedded platforms
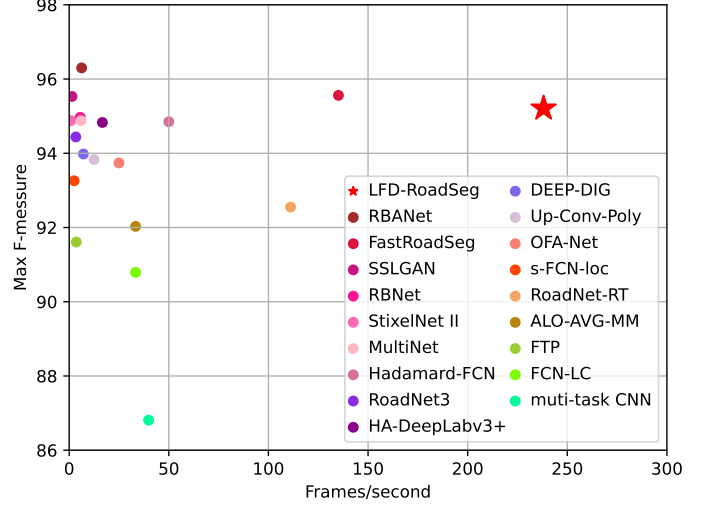
Fig. 1. Accuracy (MaxF) vs. efficiency (FPS) for various monocular road segmentation algorithms on KITTI-Road benchmark.

require models to be low-latency and lightweight, which is the goal that the recent lightweight road segmentation networks have been pursuing. The efficiency of these methods is shown in Fig. 1. Oliveira *et al.* [11] proposed a lightweight FCN-like network that achieves 12 FPS on an NVIDIA TITAN X GPU. Oeljeklaus *et al.* [12] appended two decoders of object detection and road segmentation after the inception-v2 network to realize a fast multi-task CNN, achieving a speed of 187.9 milliseconds (ms) per image on an NVIDIA Jetson TX2. Bai *et al.* [13] designed a lightweight segmentation network with a bilateral structure, namely, RoadNet-RT, achieving a speed of 9 ms per image on a GTX 1080 GPU. Gong *et al.* [14] proposed a fast encoder-decoder network that further increases the speed to 135 FPS on a TITAN Xp GPU while achieving MaxF over 95%. Overall, it is not too much to be faster for the road segmentation models on embedded systems.

Although these approaches vary in network topology and training process, they overlook a crucial characteristic of the road: *roads are 'stuff', namely background or environmental elements in an image, rather than 'things', which refer to specific identifiable objects*. Therefore, the classification of road pixels depends much less on semantic information than that of objects with semantic categories. To make a deep exploration, we implement four networks that respectively utilize the 1st, 2nd, 3rd and 4th stages' feature maps of ResNet-18 to segment the road, and their performances are shown in Table I. Apparently, the model using the 1st stage feature is obviously inferior in precision, while it obtains a recall rate as

TABLE I
Comparison of the models using different stages of ResNet-18 for road segmentation. The highlighted row compares the 1st and 3rd stages, and the bolded values show the best results. The input image resolution is 375×1240.

Feat used	MaxF	AP	PRE	REC	Params	Time(ms)
1st Stage	90.86	86.30	88.29	93.58	**157,634**	**1.80**
1st vs. 3rd	-5.35	-7.81	-8.27	-3.28	-94.33%	-45.29%
2nd Stage	95.53	93.10	95.25	95.81	683,330	2.61
3rd Stage	**96.21**	**94.11**	**96.56**	**95.86**	2,783,298	3.29
4th Stage	95.08	93.74	94.87	95.29	11,177,538	4.58

high as 93.58% and saves 94.33% parameters compared to the model using the 3rd stage. This phenomenon demonstrates that the model using the 1st stage feature finds most pixels of the road, but it suffers from false detection of several areas similar to the road surface, which is consistent with the example result in Fig. 2 (a). Subsequently, we delve deeper into the characteristics of these mis-detected pixels. To this end, we group all pixels in the prediction into three sets, i.e., the true positives (TP), the false positives (FP), and other pixels (OP). Then, the RGB gradient variance for each set is computed as an indicator of the texture intensity. Fig. 2 (c) shows the gradient variance of TP, FP, and OP. Intuitively, TP and FP typically have lower texture intensity than OP, and FP has an even lower texture intensity than TP, meaning that FP is generally situated in areas with weak texture. With all the above in mind, through extracting the context for correlating each weak texture area in a fast manner, the false detection can be eliminated effectively, while the advantages of minimal parameter and time cost can also be retained.
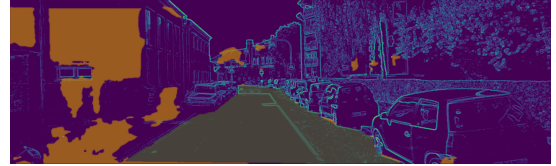
Based on the insights from the experiment above, we propose a low-level feature dominated road segmentation network (LFD-RoadSeg) that follows the bilateral structure. For the spatial detail branch, we employ the primary stage of lightweight backbone networks to extract the low-level road representation, ensuring high resolution and low latency. To eliminate non-road response in the low-level features, we design a context semantic branch in a fast manner to capture context as a supplement. For this context semantic branch, we first propose asymmetric downsampling to enable contextual features to have a large horizontal receptive that has been proven to be crucial for street scenes. Then, we design a lightweight aggregation module to capture the context with comparable receptive fields to the ResNet-18s 3rd stage that is proven effective in road segmentation (Table I). Finally, we design a selective fusion module to segment road regions from low-level road representations. This module leverages context-based spatial attention to suppress non-road responses in low-level features. The KITTI-Road, Cityscapes and CamVid datasets are employed to evaluate our method. In the experiments, LFD-RoadSeg achieves excellent effectiveness and the fastest speed so far, which can be observed in Fig. 1.

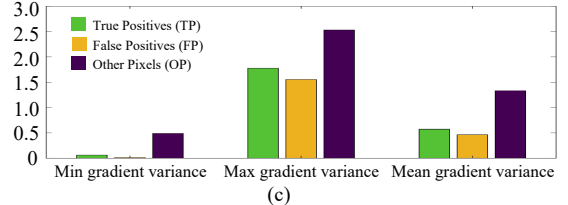In summary, the contributions of this paper are as follows:

- We reveal the "stuff" characteristic of roads overlooked by the previous road segmentation methods, and find that the primary stage of mainstream networks is adequate to extract road features. This motivates us to represent the

(a)

(b)

(c)

Fig. 2. (a) shows an example image, where yellow indicates FP, and green for TP. (b) shows the gradient of this image. (c) shows the minimal, maximal, and mean gradient variance of TP, FP and OP in the validation set of the KITTI-Road dataset.

road by low-level features.

- We propose LFD-RoadSeg. It novelly leverages low-level road representation as the basis for segmentation and employs the proposed asymmetric downsampling and aggregation modules to accelerate context extraction.
- LFD-RoadSeg boosts speed to 238 FPS (twice the previous fastest method) on a single TITAN Xp and 54 FPS on a Jetson TX2 with only 936k parameters, but still gains a decent MaxF of 95.21% on KITTI-Road. Thus, our method advances the practicability of road segmentation.

II. RELATED WORK

In this section, we briefly review the monocular road segmentation and the bilateral network for semantic segmentation, which are closely related to our approach.

A. Monocular Road Segmentation

In the early years of researching monocular road segmentation, the community [15]–[22] mainly focused on designing low-level features, such as color, edge, texture, etc., to represent and classify the road at pixel level or patch level. Later, several works tried to introduce global information to improve the reliability of road representation. Vitor *et al.* [23] designed the global probabilistic model to aggregate multiple descriptors to represent the road. Mario *et al.* [24] employed conditional random fields (CRF) to model dependencies across the whole image. Although these approaches usually achieve a road region with crisp boundary, they perform poorly in complex scenes that contain illumination change or tree shade, due to the poor generalization of the handcrafted feature. In addition, these works are not GPU-accelerated generally and thus far from real-world applications.
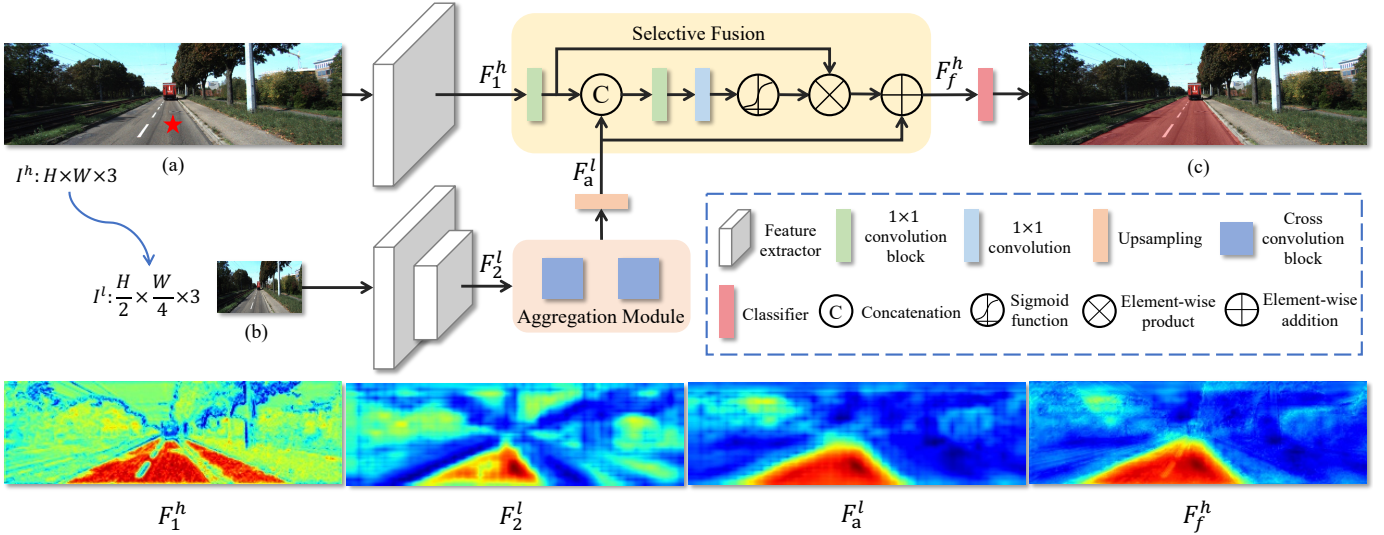
Fig. 3. The network architecture of LFD-RoadSeg. (a) is the input image $I^h \in \mathbb{R}^{H \times W \times 3}$ of spatial detail branch. (b) is the input image $I^l \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{4} \times 3}$ of context semantic branch. (c) is the visualization of the road prediction result, where the area covered by red is the road. The red star in the input image refers to the query point. The feature cosine similarity heat maps represent the correlation of the features between the query point and all others.

In recent years, monocular road segmentation has been greatly boosted by the development of neural networks. Several early approaches [25], [26] attempted to classify pixels or patches of the road by using neural networks as the classifier. The later methods [11], [14], [27], [28] followed the pipeline of deep semantic segmentation [29]–[31]. Teichmann *et al.* [28] designed a unified architecture to conduct road segmentation, object detection, and scene classification simultaneously. Sun *et al.* [27] focused on the attention mechanism to recover detailed information around the road boundary. Compared to the handcrafted methods, these methods achieve superior generalization, which makes them perform accurately in many road scenes. However, they still have a non-negligible computational burden on a mobile platform.

B. Methodologies for Embedded Systems

With the increasing application of road segmentation in the autonomous driving system, road segmentation for embedded systems has received increasing attention. Oeljeklaus *et al.* [12] proposed a fast multitask CNN for perceiving objects and the road, which achieves a speed of 5.32 FPS (375×1240) on the Jetson TX2 [32] embedded platform. Bai *et al.* [13] designed a road segmentation network RoadNet-RT optimized for FPGA, which achieves a speed of 111 FPS (280×960) on a GTX 1080 GPU. With the same setting of the experiment, Gong *et al.* [14] proposed a fast encoder-decoder network to speed up road segmentation to 31 FPS (187×620) on the Jetson TX2 embedded platform, while achieving a MaxF above 95%, much higher than the works mentioned above. However, they neglect that roads are "stuff", meaning that road segmentation relies more on low-level features. This inspires us to propose LFD-RoadSeg which achieves faster speed, lighter weight and better trade-off than the previous methods.

C. Bilateral Network for Semantic Segmentation

The bilateral networks [33] extract the spatial details and categorical semantics separately. And it is popular in the community of semantic segmentation due to its faster speed than other structures. Dong *et al.* [34] designed a lightweight baseline network with atrous convolution and a distinctive atrous spatial pyramid pooling for semantic extraction. BiSeNetv2 [35] proposed a bilateral guided aggregation layer to enhance the mutual connections of the two branches. CABiNet [36] designed a context branch with lightweight versions of global aggregation and local distribution blocks. Different from the above methods, which use two independent branches, Fast-SCNN [37], EACNet [38] and DDRNet [39] all utilized a shared trunk and two parallel branches with different resolutions. ContextNet [40] reduced the input image's resolution for the semantic branch to a quarter of the original image to accelerate the inference process.

D. Motivation

LFD-RoadSeg is motivated by the fact that roads are "stuff". According to [41], "stuff" is the background and environmental elements in the image, rather than "things" (such as person and car) that rely on semantic features in classification. Thus, the same "stuff" area has a similar texture, leading us to believe that road pixels can be classified by using low-level features. To this end, we compare the differences of various stages in road segmentation. Consequently, we find that the first stage of ResNet is adequate for representing road, as evidenced by the high recall rate of 93.58% in Table I and the green area in Fig. 2. However, the first stage of ResNet suffers from the false positives, which is indicated by the low precision and yellow area in Fig. 2. This inspires us to design a lightweight context semantic branch to quickly extract correlation between each area, which gives the indicators for reducing the non-road response in the low-level feature.

III. APPROACH

In this section, we first describe the specific network architecture in Sec. III-A in detail, including the spatial detail branch, the context semantic branch and the selective fusion module. Then the loss function dedicated to hard negative sample mining we used is introduced in Sec. III-B.

A. The Proposed Network Architecture

Inspired by the comparisons in Table I, we represent the road by low-level features to achieve ultra-fast road segmentation. As illustrated in Fig. 3, the overall structure of our network includes two branches and a feature selective fusion module. And regarding the two branches, one is a high-resolution spatial detail branch, and the other is a low-resolution context semantic branch.

1) Spatial Detail Branch: The spatial detail branch is to capture most pixels of the road from the input RGB image $I^h \in \mathbb{R}^{H \times W \times 3}$ with only a small amount of convolutions. Although our method can be applied to various backbone networks, for representation simplification, we describe the following network structure based on ResNet-18 by default. To be specific, let F_i^h be the output feature maps of the ith stage of a ResNet-18 network when taking I^h as input. And the spatial detail branch employs the 1st stage of ResNet-18 as the backbone and outputs the feature $F_1^h \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 64}$. According to Table I, the extremely light structure of this branch encodes most of the road pixels and preserves high resolution with low computational cost.

2) Context Semantic Branch: The context semantic branch aims to capture contextual information in a fast manner for suppressing the texture-less non-road region in F_1^h. Specifically, this branch employs two designs to achieve an extremely fast speed of context extraction. Firstly, since the correlation between horizontal areas is crucial for road segmentation [14], the input RGB image I^h is downsampled by a factor of 4 horizontally and by a factor of 2 vertically to obtain a low-resolution RGB image $I^l \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{4} \times 3}$. The resizing operation achieves a larger horizontal receptive field and a decent inference time. Secondly, since the 3rd stage of ResNet-18 outperforms others in Table I, the context semantic branch requires the same receptive field as the 3rd stage. To this end, this branch utilizes the first two stages of ResNet-18 to obtain the feature $F_2^l \in \mathbb{R}^{\frac{W}{16} \times \frac{W}{32} \times 128}$ from I^l, and then appends a newly designed aggregation module to achieve a similar receptive field as the 3rd stage, but at a faster speed than the first three stages of ResNet-18.

Next, we elaborate on the structure of the newly designed aggregation module and discuss the advantages of the above design in terms of computation and parameter amount.

Aggregation Module. The 3rd stage of the original ResNet-18 has parameters up to 2.1M (see Table I, 2,783,298 - 683,330 = 2,099,968), and the aggregation module implements similar feature extraction capabilities with a lighter structure. It consists of two cross convolution blocks, and the structure of each cross convolution block is shown in Fig. 4. In each cross convolution block, assuming that F_I denotes the input feature, we use a 1×5 row convolution and a 5×1 column
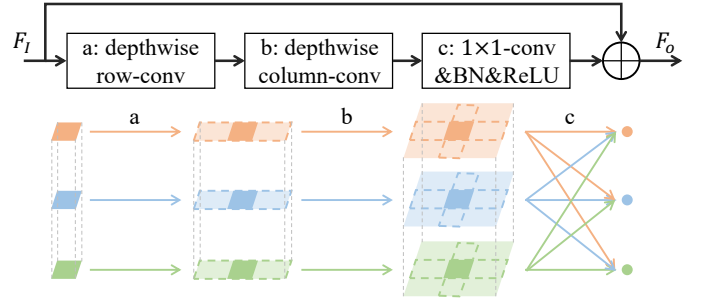

Fig. 4. Structure of the cross convolution block. $\oplus$ denotes the element-wise sum operation.

TABLE II
MODEL COMPLEXITY AND INFERENCE TIME COMPARISON. THE INPUT IMAGE RESOLUTION OF THE FIRST ROW IS 375×1240, AND THE RESOLUTION OF THE SECOND ROW IS 187×310.

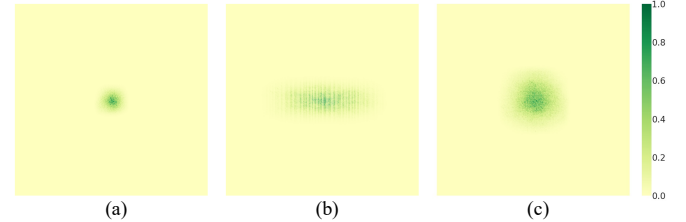|	MACs[G]	Params[M]	Time(ms)
Three stages of ResNet-18	13.23	2.78	3.24
Context semantic branch	1.21(-91%)	0.72(-74%)	1.90(-41%)

Fig. 5. Visualization of the model effective receptive field. (a) is the effective receptive field for the first three stages of ResNet-18. (b) is the effective receptive field for the context semantic branch. (c) is the effective receptive field for all four stages of ResNet-18.

convolution to simulate a large-kernel convolution for context feature extraction on F_I. Both the row and column convolution are depth-separable and aggregate context information in each channel respectively. Then, a 1×1 convolution is employed to fuse all channels. Note that, to gain a larger horizontal receptive field, the row convolution in the first cross convolution block has a dilation of 2. Finally, we add the fused feature and the input feature F_I element-wisely to reserve the detail contained in the input feature, and the output feature F_O can be formulated as:

$$F_O = B_{1 \times 1}(C(R(F_I))) + F_I \tag{1}$$

where R denotes the depthwise row convolution (row-conv for short), C denotes the depthwise column convolution (column-conv for short), and $B_{1 \times 1}$ denotes the 1×1 convolution block which contains a 1×1 convolution (1×1-conv for short), a batch normalization and a ReLU activation. By using two cross convolution blocks connected in series, we extract the context feature from the feature F_2^l, which is followed by an upsampling operation to align the feature resolution to F_1^h, namely $\frac{H}{4} \times \frac{W}{4}$. And we obtain a high-resolution context feature $F_a^l \in \mathbb{R}^{\frac{W}{4} \times \frac{W}{4} \times 128}$.

Discussion. In the section above, two designs are given to make the context semantic branch have similar discriminative

power as the first three stages of ResNet-18 but with fewer parameters and computations. One is to reduce the input image size, and the other is the aggregation module that replaces ResNet-18's 3rd stage. Compared to the original first three stages of ResNet-18, the context semantic branch reduces the MACs (multiply-accumulate operations) by 91%, parameters by 74%, and inference time by 41%, as shown in Table II. Furthermore, we visualize the effective receptive field of ResNet-18 and the context semantic branch in Fig. 5. It can be seen that the context semantic branch has a comparable receptive field to ResNet-18's 3rd stage in the vertical direction. Note that, due to the asymmetric downsampling and the dilation row-conv, our method has a larger receptive field in the horizontal direction, which is good for extracting correlation between different areas (e.g., drivable road, sidewalk, rail track, etc.) that are in the horizontal direction of each other [14].

3) Selective Fusion: The high-resolution road representation F_1^h obtained by the spatial detail branch suffers from weakly textured non-road regions that are easily misclassified. In this section, we propose a selective fusion module to remove these non-road regions by the context feature F_a^l obtained from the context semantic branch. Specifically, the structure of the selective fusion module is shown in Fig. 3. Firstly, we use a 1×1 convolution block to adjust the number of channels of F_1^h to be the same as that of F_a^l, namely 128. Then we concatenate the output feature of both branches, namely F_1^h and F_a^l, and calculate the pixel-wise attention between them by a 1×1 convolution block, a 1×1 convolution and a sigmoid function, which can be formulated as:

$$F_{attention} = S(P(B_{1\times1}(B_{1\times1}(F_1^h) \copyright F_a^l))) \quad (2)$$

where $F_{attention} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 1}$ is the spatial attention weight. S denotes sigmoid function, P denotes 1×1 convolution, and $\copyright$ denotes concatenation. Finally, the spatial attention weight $F_{attention}$ is employed to select the road area from the low-level representation F_1^h, and the context feature also serves as the complementary expression of the road:

$$F_f^h = F_{attention} \otimes B_{1\times1}(F_1^h) \oplus F_a^l \quad (3)$$

where $\otimes$ denotes the element-wise product operation, $\oplus$ denotes the element-wise sum operation, and the attention weights $F_{attention}$ are applied to all channels of the high-resolution branch feature. Finally, the fused feature is fed into a pixel-wise classifier, namely a block containing a 1×1 convolution, a batch normalization, a ReLU activation and a 1×1 convolution, to segment the road. As shown in Fig. 3, after the selective fusion, the feature response of the sidewalk on the right side of the road in F_1^h is suppressed, and the road features are enhanced.

Differences between RoadNet-RT and LFD-RoadSeg: Although the overall structure of LFD-RoadSeg is roughly similar to RoadNet-RT [13], there are still three differences between them. Firstly, we determine the network depth and receptive field of the two branches by experiments, rather than solely relying on intuition. Secondly, we propose the aggregation module to more effectively capture the contextual information, instead of using ASPP and global attention.

Thirdly, during fusing the two branches' features, we calculate the spatial attention to express the relation between the low-level feature and the context feature, and utilize it to suppress texture-less non-road regions in low-level features. Therefore, our fusion process is different from RoadNet-RT which performs channel attention fusion on the two branches.

B. Loss Function

Since road segmentation is a binary classification problem, binary cross-entropy loss is generally used as the loss function of the classifier to supervise the final output. Let N denote the number of pixels, i and j are the pixel index in the image, $y \in \{0,1\}$ denotes the ground truth and p is the predicted confidence, the binary cross-entropy loss is formulated as:

$$\mathbf{L}_{bce} = -\frac{1}{N}\sum_{i,j} y^{i,j}log(p^{i,j}) + (1-y^{i,j})log(1-p^{i,j}) \quad (4)$$

As we know, the texture-less road and some texture-less areas such as vegetation, sky and buildings are easier to distinguish than highly textured areas. And the drivable road boundaries, sidewalks, and abnormal road areas such as overexposure and shadow are prone to be misclassified. Therefore, in the training process, we utilize a strategy similar to OHEM [42] to mine difficult-to-segment pixels. Given a confidence threshold λ_b in a batch, we only perform gradient backpropagation for pixels whose predicted confidence p is less than the threshold λ_b. Let $\mathbb{1}(\cdot)$ denote the indicator function, and the entire loss function of the network $\mathbf{L}_{main}$ is formulated as:

$$\mathbf{L}_{main} = \frac{1}{N}\sum_{i,j} \mathbb{1}(p^{i,j} < \lambda_b)\mathbf{L}_{bce}^{i,j} \quad (5)$$

IV. EXPERIMENTS

In this section, we first describe the datasets and evaluation metrics in Sec. IV-A. Then the detail of the network training is given in Sec. IV-B. Sec. IV-C reports the quantitative and qualitative results of our method. Finally, the ablation study of each component and the discussion on the input image size of the context semantic branch are given in Sec. IV-D and Sec. IV-E, respectively.

A. Dataset and Evaluation

1) Dataset: Three datasets are employed to evaluate the effectiveness of our method. KITTI-Road [54] is a real-world road segmentation dataset containing 289 training images and 290 testing images. It has three categories of road scenes, Urban Unmarked (UU), Urban Marked (UM) and Urban Multiple Marked (UMM). URBAN is a combination of the three above. The resolution of KITTI-Road training images ranges from 370 × 1224 to 375 × 1242. For the convenience of training, we unify them as 375 × 1240 by padding operation. The evaluation is done by the official online evaluation server. Following [14], in the ablation experiments, we use 5-fold cross-validation on the training images, and the experimental results are expressed as (mean ± standard deviation). Cityscapes [55] is a real-world dataset of urban

TABLE III
COMPARISON WITH PRIOR REPRESENTATIVE ROAD SEGMENTATION WORKS ON KITTI-ROAD DATASET
("-" MEANS IT IS NOT MENTIONED IN THE OFFICIAL DATABASE AND THE ORIGINAL PAPER)

Method	Sensor	Input shape	MaxF(%)↑	AP(%)↑	PRE(%)↑	REC(%)↑	FPR(%)↓	FNR(%)↓	Time(ms)↓	Device
RBANet [27]	Cam.	360×720	96.30	89.72	95.14	97.50	2.75	2.50	160	TITAN Xp
FastRoadSeg [14]	Cam.	375×1240	95.56	93.89	95.53	95.59	2.47	4.41	7.4	TITAN Xp
SSLGAN [43]	Cam.	375×1242	95.53	90.35	95.84	95.24	2.28	4.76	700	TITAN X
RBNet [44]	Cam.	300×900	94.97	91.49	94.94	95.01	2.79	4.99	180	Tesla K20c
StixelNet II [45]	Cam.	370×800	94.88	87.75	92.97	96.87	4.04	3.13	1200	Quadro M6000
MultiNet [28]	Cam.	384×1248	94.88	93.71	94.84	94.91	2.85	5.09	170	GTX 1080
Hadamard-FCN [46]	Cam.	375×1242	94.85	91.48	94.81	94.89	2.86	5.11	20	TITAN X
HA-DeepLabv3+ [47]	Cam.	-	94.83	93.24	94.77	94.89	2.88	5.11	60	-
RoadNet3 [48]	Cam.	160×600	94.44	93.45	94.69	94.18	2.91	5.82	300	GTX 950M
DEEP-DIG [49]	Cam.	-	93.98	93.65	94.26	93.69	3.14	6.31	140	TITAN X
Up-Conv-Poly [50]	Cam.	500×500	93.83	90.47	94.00	93.67	3.29	6.33	80	TITAN X
OFA-Net [51]	Cam.	-	93.74	85.37	90.36	97.38	5.72	2.62	40	-
s-FCN-loc [52]	Cam.	500×500	93.26	-	94.16	92.39	3.16	7.61	400	Tesla K80
RoadNet-RT [13]	Cam.	280×960	92.55	93.21	92.94	92.16	3.86	7.84	9	GTX 1080
ALO-AVG-MM [53]	Cam.	192×624	92.03	85.64	90.65	93.45	5.31	6.55	30	GTX 1080
LFD-RoadSeg	Cam.	375×1240	**95.21**	**93.71**	**95.35**	**95.08**	**2.56**	**4.92**	**4.2**	TITAN Xp

TABLE IV
MODEL COMPLEXITY COMPARISON

Method	MACs [G]	Parameters [M]
FastRoadSeg [14]	18.323	11.334
LFD-RoadSeg	**8.392**	**0.936**

street scenarios, including 2975 images for training and 500 images for validation. All images are at 1024×2048 resolution. CamVid [56] is also a real-world dataset for driving scenarios, which contains 367 training images, 101 validation images, and 233 test images with a resolution of 720×960. Cityscapes and CamVid are classical semantic segmentation datasets with multiple category annotations. When applying them to the road segmentation task, we set the label of the road to 1 and the other categories to 0.

2) Evaluation Metrics: For the KITTI-Road dataset, we evaluate the performance using six official metrics, namely maximum F1-measure (MaxF), average precision (AP), precision (PRE), recall (REC), false positive rate (FPR) and false negative rate (FNR). Among them, MaxF is the main accuracy evaluation metric because it comprehensively considers precision and recall. It is worth noting that the metrics are computed in the Birds Eye View (BEV) for the KITTI-Road dataset as a common practice. We also evaluate the parameters, MACs, and inference time of our network. With 375×1240 as the input resolution, we compute the average time of 1000 forwards on a single GPU as our inference time. For Cityscapes and CamVid datasets, MaxF, PRE, REC, and mIoU (mean intersection over union) are employed to evaluate the performance in the image space.

B. Training Details

LFD-RoadSeg is implemented in PyTorch on Intel(R) Xeon(R) CPUs and is deployed on a single NVIDIA TITAN Xp GPU. The feature extractors of the two branches, namely part of the ResNet-18 [57], are loaded with the model parameters which are pre-trained on ImageNet [58] as initial weights. Other parameters of the LFD-RoadSeg are randomly initialized. Stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay of 10^{-4} is used to optimize our network.

The initial learning rate is set to 0.01 and the cosine annealing learning rate decay strategy is adopted in the training process. The final learning rate is set to 10^{-5}. In order to prevent the model from overfitting, we apply some data augmentations on the training images such as random cropping, random horizontal flipping, random brightness adjusting with the range of [0.9, 1.1], and random scaling with the range of [0.5, 2.0]. The OHEM loss threshold λ_b is set to 0.7 in the experiment. For KITTI-Road, we employ a two-step training scheme. First, we randomly crop out 320×500 patches from the original images as the inputs I^h and second, the training is resumed by using the full images as I^h. The batch size for the first training step is set to 16 and for the second training step is set to 6. We train LFD-RoadSeg for 150 epochs in these two steps. For Cityscapes, 800×800 patches are randomly cropped out for training. The batch size is 4 and the maximum number of epochs is 250. For CamVid, the input image size is 720×960. The batch size is 4 and the maximum number of epochs is 150.

C. Final Results and Comparison with Prior Works

KITTI-Road dataset. Table III reports the segmentation results and the corresponding time cost comparison between LFD-RoadSeg and the prior representative road segmentation works on the KITTI-Road leaderboard. Note that, we cannot compare all methods on the same platform as most of them did not release the code. Therefore, we use the "Device" and "Time" information provided by the official KITTI benchmark database and each paper to comprehensively evaluate the computational efficiency of each method. From Table III, we can see that LFD-RoadSeg is the fastest and outperforms many monocular-based methods [28], [44]–[48] in MaxF and achieves a high AP of 93.71%. In addition, RBANet [27] and SSLGAN [43] fail to meet real-time requirements. Compared with the current second-fastest method FastRoadSeg [14], LFD-RoadSeg only reduces MaxF by 0.35% (95.56%-95.21% = 0.35%), but the speed is increased by 43.2% (1-4.2/7.4 = 43.2%). Furthermore, Table IV shows the complexity of FastRoadSeg [14] and LFD-RoadSeg. LFD-RoadSeg has more than 12 times fewer parameters than FastRoadSeg and reduces the MACs (multiply-accumulate operations) by 54.2% (1-8.392/18.323 = 54.2%). Compared with RoadNet-RT [13],

Benchmark	MaxF	AP	PRE	REC	FPR	FNR
UM_ROAD	94.58%	93.42%	95.20%	93.98%	2.16%	6.02%
UMM_ROAD	96.59%	95.40%	96.29%	96.90%	4.11%	3.10%
UU_ROAD	93.49%	92.19%	93.46%	93.52%	2.13%	6.48%
URBAN_ROAD	95.21%	93.71%	95.35%	95.08%	2.56%	4..92%

Methods	MaxF(%)↑	PRE(%)↑	REC(%)↑	mIoU(%)↑
Zohourian *et al.* [59]	92.44	89.08	96.76	-
FCN [30]	94.68	93.69	95.70	-
†FCN [30]	94.75	93.65	95.77	93.96
s-FCN-loc [52]	95.36	94.63	96.11	-
SegNet [60]	95.81	94.55	97.11	-
†SegNet [60]	95.92	94.73	96.78	94.33
†U-Net [61]	96.26	94.89	97.19	95.21
FastRoadSeg [14]	96.48	95.84	97.12	95.74
†FASSD-Net [62]	97.47	97.51	98.07	96.34
RBANet [27]	98.00	97.87	98.13	-
LFD-RoadSeg	**96.01**	**94.28**	**97.80**	**96.68**

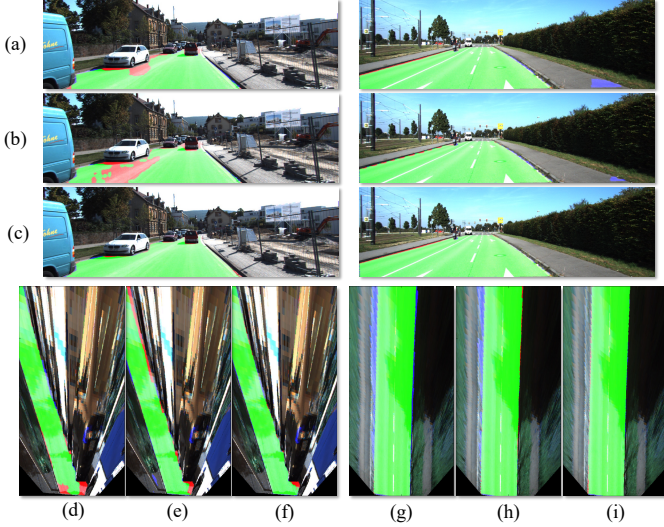| Methods	MaxF(%)↑	PRE(%)↑	REC(%)↑	mIoU(%)↑
RoadNet-RT [13]	92.98	94.70	91.91	-
SegNet [60]	93.95	93.07	94.86	-
†SegNet [60]	94.89	94.27	95.91	94.13
Yadav *et al.* [63]	94.14	93.31	94.99	-
†U-Net [61]	96.45	95.73	96.11	95.65
RBANet [27]	96.72	97.14	96.30	-
FastRoadSeg [14]	97.02	96.79	97.24	96.11
†FASSD-Net [62]	97.38	97.69	98.81	97.64
LFD-RoadSeg	**97.02**	**96.80**	**97.25**	**95.70**

Fig. 6. Qualitative comparison with other real-time monocular road segmentation methods. (a)-(c) are in the camera's perspective view and (d)-(i) are in the birds eye view. (a) (d) (g): ALO-AVG-MM [53], (b) (e) (h): RoadNet-RT [13], (c) (f) (i): LFD-RoadSeg. Red marks false negatives, blue marks false positives, and green marks true positives.

which is also a bilateral network, LFD-RoadSeg increases the MaxF from 92.55% to 95.21%, which is due to the more reasonable network depth and more efficient fusion process. The detailed performance provided by the KITTI online test server is shown in Table V. Fig. 6 illustrates the qualitative comparison with other real-time monocular methods, ALO-AVG-MM [53] and RoadNet-RT [13]. We show the camera's perspective view and birds eye view respectively. Compared to other methods, LFD-RoadSeg reduces both the number of false positive and false negative pixels.

Cityscapes dataset. Table VI presents the performance of several methods on the Cityscapes dataset with metrics such as MaxF, PRE, REC, and mIoU. In terms of the MaxF metric, despite LFD-RoadSeg fails to surpass the best method, i.e., RBANet [27], it outperforms conventional benchmark methods like FCN [27], and it is only 0.47% lower than FastRoadSeg [14] on MaxF. Not that, the accuracy gap between our method and the best method lies in precision (3.59% lower), while our recall is not much different from that of the best method (only 0.33% lower). The reason is that the Cityscapes dataset contains a large number of specific categories of objects, that is, "things" defined in panoptic segmentation. Using low-level features to express these objects during training would slightly degrade the discriminative ability of the model.

CamVid dataset. Table VII provides the quantitative comparisons on the CamVid dataset. Notably, our method achieves an impressive MaxF score of 97.02%, the same as Fas-

tRoadSeg [14] and even outperforming FastRoadSeg [14] in precision and recall. Compared to RBANet [27], our method achieves 0.3% higher MaxF than RBANet [27] which is a non-lightweight method. Compared to RoadNet-RT [13] that also utilizes a bilateral architecture, our method obtains significantly higher accuracy in MaxF (4.04% higher), PRE (5.20% higher), and REC (1.04% higher) metrics. In contrast, RoadNet-RT [13] requires 9 ms for inference at a resolution of 280×960 on a GTX 1080 GPU, whereas LFD-RoadSeg achieves inference in just 4.2 ms at 375×1240 resolution on a TITAN Xp GPU, showcasing superior speed performance.

Overall, the performance of our method is close to that of the State-of-the-art method on the KITTI-Road and CamVid datasets. This is due to the fact that in urban street scenes, low-level features are sufficient to represent most road areas.

D. Ablation Study of Each Component

We further examine the effectiveness of each component in the proposed LFD-RoadSeg. From Table VIII, we can see that the performance of LFD-RoadSeg is better than using the spatial detail branch only and the context semantic branch only. The reason is that the low-level features are not effective in distinguishing different long-range areas due to their limited local receptive field, while the context feature lacks spatial detail due to the low resolution. As shown in Table VIII, when using the context semantic branch only, the aggregation module helps to improve the MaxF from 94.69% to 94.88%. When using both branches, the aggregation module helps to improve the MaxF from 95.76% to 96.28%. The two

TABLE VIII
EFFECTIVENESS OF EACH COMPONENT ON KITTI-ROAD CROSS-VALIDATION. (SDB: SPATIAL DETAIL BRANCH; CSB: CONTEXT SEMANTIC BRANCH.

SDB	CSB	Aggregation Module	©	⊗	⊕	Selective Fusion	Classifier	MaxF(%)↑	PRE(%)↑	REC(%)↑	Parameters
✓							✓	91.13±0.18	88.90±0.46	93.49±0.40	183106
	✓						✓	94.69±0.28	94.51±0.52	94.88±0.43	700098
	✓	✓					✓	94.88±0.17	94.80±0.56	94.96±0.43	736706
✓	✓					✓	✓	95.76±0.25	95.78±0.45	95.75±0.42	899459
✓	✓	✓	✓				✓	95.96±0.21	96.03±0.35	95.89±0.21	919170
✓	✓	✓		✓			✓	95.77±0.16	96.14±0.30	95.40±0.26	902786
✓	✓	✓			✓		✓	95.92±0.10	96.21±0.30	95.62±0.30	902786
✓	✓	✓				✓	✓	**96.28±0.14**	**96.56±0.11**	**96.00±0.21**	936067

TABLE IX
PERFORMANCE COMPARISON OF THE CONTEXT SEMANTIC BRANCH WITH DIFFERENT INPUT IMAGE SIZES ON KITTI-ROAD CROSS-VALIDATION

Height	Width	MaxF(%)↑	MACs [G]	Time (ms)
1/2	1/2	96.32±0.20	9.598	4.6
1/2	**1/4**	**96.28±0.14**	**8.392**	**4.2**
1/4	1/2	95.86±0.18	8.400	4.2
1/4	1/4	96.01±0.23	7.792	4.1

TABLE X
PERFORMANCE COMPARISON OF THE CONTEXT SEMANTIC BRANCH WITH DIFFERENT INPUT IMAGE SIZES ON CITYSCAPES AND CAMVID DATASETS

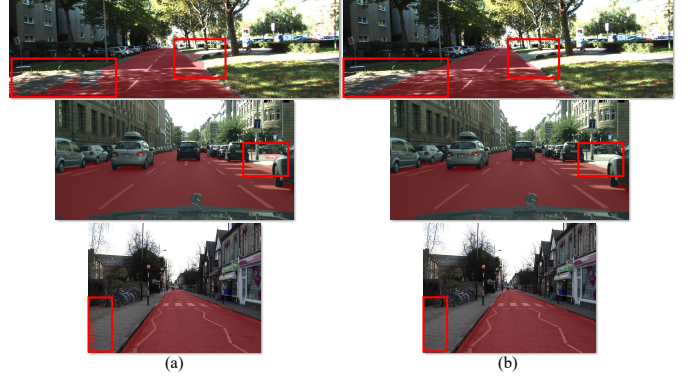| Datasets	Height	Width	MaxF(%)↑	PRE(%)↑	REC(%)↑
Cityscapes	1/2	1/2	95.83	94.23	97.50
	1/2	**1/4**	**96.01**	**94.28**	**97.80**
	1/4	1/2	95.58	94.12	97.08
	1/4	1/4	95.87	94.53	97.26
CamVid	1/2	1/2	96.20	95.75	96.65
	1/2	**1/4**	**97.02**	**96.80**	**97.25**
	1/4	1/2	95.80	94.82	96.80
	1/4	1/4	96.41	96.03	96.79

Fig. 7. The visualization of the road prediction result, where the area covered by red is the road. (a): The input image size of the context semantic branch is 1/4 the height of the original image and 1/2 the width of the original image. (b): The input image size of the context semantic branch is 1/2 the height of the original image and 1/4 the width of the original image.

comparisons indicate that the aggregation module is suitable for capturing the context information with a large receptive field. Furthermore, we verify the effectiveness of selection fusion, and compare it with three fusion processes, namely, direct concatenation, direct element-wise product and direct element-wise addition. As we can see from the last four rows of Table VIII, the selection fusion achieves the best MaxF, PRE, and REC with only a few additional parameters. Finally, when we train the network with all components, the proposed LFD-RoadSeg achieves the best performance on the KITTI-Road dataset.

E. Discussion on Input Resolution

The input image I^l for the context semantic branch is obtained by asymmetrically downsampling the original image I^h, which has two goals. The first goal is to reduce the computational burden as much as possible and speed up the training and testing processes. The second goal is to gain a larger horizontal receptive field so that LFD-RoadSeg captures longer-range dependencies in the horizontal direction. The results using different aspect ratios are shown in Table IX. Compared to our asymmetric input size setting, when both height and width are reduced to half of the original size, the MaxF only increases from 96.28% to 96.32% but at the cost of

0.4 ms longer inference time. And when both height and width are shrunk by a quarter, the model increases the MaxF from 96.01% to 96.28% and only sacrifices 0.1 ms in inference time. The two comparisons illustrate that our experimental setting has a better trade-off than others. Furthermore, when inverting the downsampling rate of height and width, that is, the network has a larger vertical receptive field, the MaxF drops dramatically and is even worse than the model using a smaller resolution.

We then discuss the asymmetric downsampling on the Cityscapes and CamVid datasets in Table X. Our experimental setting (the bolded row in Table X) yields the highest MaxF scores, while the opposite setting results in the lowest MaxF scores. The MaxF gaps between the two reach 0.43% on Cityscapes and 1.22% on CamVid. It verifies that the significance of horizontal receptive fields to deep networks is not limited to the KITTI-Road dataset, but is ubiquitous in various street scenes. Fig. 7 further displays the visual comparisons between our setting and the opposite setting. The first to third rows are from the KITTI-Road, Cityscapes, and CamVid datasets. Observably, the model with the opposite setting obtains many false positives on the texture-less area outside the road as it has a narrow receptive field, which is highlighted by the red rectangular boxes. However, our model with the proposed asymmetric downsampling setting eliminates most false positives.

Methods	MaxF(%)↑	AP(%)↑	Params	Time (ms)
ResNet-18	95.08	93.74	11,177,538	4.58
Ours (ResNet-18)	96.39	94.19	936,067	4.24
MobileNetV2	95.55	93.90	1,884,162	9.05
Ours (MobileNetV2)	96.06	94.16	161,395	7.49
MobileViT-XXS	95.15	93.85	947,554	22.75
Ours (MobileViT-XXS)	95.98	94.11	253,779	8.20

F. Results and Comparison for Different Backbones

Table XI provides a performance comparison of lightweight backbones with various structures on the KITTI-Road dataset, including ResNet-18 [57], MobileNetV2 [64] and MobileViT-XXS [65]. We measure them based on MaxF, AP, parameters and inference time.

Specifically, for ours (ResNet-18), the spatial detail branch utilizes the first stage of ResNet-18 as its backbone, while the context semantic branch employs the first two stages of ResNet-18 as its backbone. For ours (MobileNetV2) and ours (MobileViT-XXS), due to the lack of official stages for MobileNetV2 and MobileViT, we use layers that extract features at 1/4 the original resolution as the spatial detail branch, and layers that extract features at 1/8 the original resolution as the context semantic branch.

From Table XI, observably, ours (ResNet-18) achieves the highest MaxF, reaching 96.39%, and the highest AP, reaching 94.19%. Compared with the original backbone network (i.e., ResNet-18, MobileNetV2, and MobileViT-XXS), all variants equipped with the proposed structure achieve higher accuracy (MaxF improvement of 1.31%, 0.51%, 0.83%, respectively), lower parameters (reduced by 99.99%, 91.43%, 73.22% respectively) and less inference time (reduced by 7.42%, 17.24%, 63.95% respectively). This experiment proves that the proposed structure is not only suitable for classic residual networks, but also advanced transformer networks. Even on the transformer model, our structure contributes to a significant speed improvement.

G. Deployment and Speed Comparison

Following the works of [14] [66] [67], we deployed the proposed model on the Jetson TX2 [32]. The Jetson TX2 is equipped with a quad-core ARM A57 processor, a dual-core Denver2 processor, a 256-core NVIDIA Pascal architecture GPU, and 8GB of 128-bit LPDDR4 memory. This configuration makes it suitable for use in robots, drones, and other intelligent edge devices. During the inference process, the input image is resized to 187×620, aligning with FastRoadSeg [14]. Table XII provides details on the speed and power consumption in both maximum processing efficiency (Max-N) and maximum energy efficiency (Max-Q) modes on the Jetson TX2 and offers a comparison with FastRoadSeg [14] on the same platform. Note that all methods do not use any acceleration techniques. Observably, LFD-RoadSeg is 74% faster than FastRoadSeg with 2W lower power consumption. When using less than 7W of power, its speed is still 35% faster than that of FastRoadSeg [14].

Method	Runtime	Frame Rate	Power Consumption
FastRoadSeg [14]	32.2 ms	31 FPS	14.8 W(3.1 W idle)
LFD-RoadSeg (Max-N)	18.5 ms	54 FPS	12.8 W(3.1 W idle)
LFD-RoadSeg (Max-Q)	14.4 ms	42 FPS	6.9 W(2.3 W idle)

V. CONCLUSION

In this study, considering that roads are part of the environmental background rather than specific objects, we propose a Low-level Feature Dominated Road Segmentation network (LFD-RoadSeg) to achieve accurate and efficient road segmentation. It follows a bilateral structure. The spatial detail branch extracts low-level road representation. The context semantic branch quickly captures the context having large horizontal receptive fields with the help of asymmetric downsampling and lightweight aggregation modules. In addition, the selective fusion module leverages the context to suppress the non-road response in the low-level feature. Comprehensive experiments on three datasets indicate that our method achieves similar accuracy as mainstream methods but at a speed of 238 FPS on a single TITAN Xp, 54 FPS on a Jetson TX2 and costs the parameter amount of only 936k.

VI. LIMITATIONS AND FUTURE WORK

While our research has made significant strides in real-time road segmentation, there are several limitations to acknowledge. Our model performs well under typical road conditions. However, it may face challenges in extreme weather conditions, such as heavy rain, snow, or fog. Future work should focus on enhancing the model's robustness under adverse conditions. In addition, due to using low-level features, our model cannot be generalized to the scenes totally different from the training set, such as training models on street scenes and testing models on off-road scenes. In the future, it will be possible to enhance the model's robustness and generalization by utilizing techniques such as training on multiple datasets, data augmentation, and transfer learning, thereby improving its practicability.

REFERENCES

[1] F. Xue, J. Cao, Y. Zhou, F. Sheng, Y. Wang, and A. Ming, "Boundary-induced and scene-aggregated network for monocular depth prediction," *Pattern Recognition(PR)*, vol. 115, p. 107901, 2021.

[2] F. Xue, A. Ming, M. Zhou, and Y. Zhou, "A novel multi-layer framework for tiny obstacle discovery," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2939–2945.

[3] F. Xue, A. Ming, and Y. Zhou, "Tiny obstacle discovery by occlusion-aware multilayer regression," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 9373–9386, 2020.

[4] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "Tanet: Robust 3d object detection from point clouds with triple attention," in *AAAI Conference on Artificial Intelligence(AAAI)*, 2020.

[5] R. Lu, F. Xue, M. Zhou, A. Ming, and Y. Zhou, "Occlusion-shared and feature-separated network for occlusion relationship reasoning," in *IEEE/CVF International Conference on Computer Vision(ICCV)*, 2019, pp. 10 343–10 352.

[6] Y. Gao, X. Li, J. Zhang, Y. Zhou, D. Jin, J. Wang, S. Zhu, and X. Bai, "Video text tracking with a spatio-temporal complementary model," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 9321–9331, 2021.

[7] F. Xue, Y. Chang, T. Wang, Y. Zhou, and A. Ming, "Indoor Obstacle Discovery on Reflective Ground via Monocular Camera," *International Journal of Computer Vision*, oct 2023.

[8] A. Kherraki, M. Maqbool, and R. El Ouazzani, "Efficient lightweight residual network for real-time road semantic segmentation," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, p. 394, 2023.

[9] H. Xu, H. He, Y. Zhang, L. Ma, and J. Li, "A comparative study of loss functions for road segmentation in remotely sensed road datasets," *International Journal of Applied Earth Observation and Geoinformation*, vol. 116, p. 103159, 2023.

[10] Y. Zhou, R. Lu, F. Xue, and Y. Gao, "Occlusion relationship reasoning with a feature separation and interaction network," *Visual Intelligence*, vol. 1, no. 1, p. 23, 2023.

[11] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[12] M. Oeljeklaus, F. Hoffmann, and T. Bertram, "A fast multi-task cnn for spatial understanding of traffic scenes," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[13] L. Bai, Y. Lyu, and X. Huang, "Roadnet-rt: High throughput cnn architecture and soc design for real-time road segmentation," *IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS)*, vol. 68, no. 2, pp. 704–714, 2020.

[14] S. Gong, H. Zhou, F. Xue, C. Fang, Y. Li, and Y. Zhou, "Fastroadseg: Fast monocular road segmentation network," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 23, no. 11, pp. 21505–21514, 2022.

[15] J. M. Alvarez, M. Salzmann, and N. Barnes, "Learning appearance models for road detection," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013.

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

[17] D. A. Chacra and J. Zelek, "Road segmentation in street view images using texture information," in *13th Conference on Computer and Robot Vision (CRV)*, 2016.

[18] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *British Machine Vision Conference (BMVC)*, 2009.

[19] J. Ma, A. Ming, Z. Huang, X. Wang, and Y. Zhou, "Object-level proposals," in *IEEE International Conference on Computer Vision(ICCV)*, 2017, pp. 4921–4929.

[20] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki, "Similarity fusion for visual tracking," *International Journal of Computer Vision(IJCV)*, vol. 118, no. 3, pp. 337–363, 2016.

[21] Y. Zhou, X. Bai, W. Liu, and L. Latecki, "Fusion with diffusion for robust visual tracking," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[22] Y. Zhou., Y. Yang., Y. Meng., X. Bai, W. Liu, and L. J. Latecki., "Online multiple person detection and tracking from mobile robot in cluttered indoor environments with depth camera," *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, vol. 28, no. 1, pp. 1455001.1–1455001.28, 2014.

[23] G. Bernardes Vitor, A. C. Victorino, and J. V. Ferreira, "A probabilistic distribution approach for the classification of urban roads in complex environments," in *IEEE Workshop on International Conference on Robotics and Automation (ICRA Workshop)*, 2014.

[24] M. Passani, J. J. Yebes, and L. M. Bergasa, "Crf-based semantic labeling in miniaturized road scenes," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[25] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *European Conference on Computer Vision (ECCV)*, 2012.

[26] C. C. T. Mendes, V. Frémont, and D. F. Wolf, "Vision-based road detection using contextual blocks," *arXiv preprint arXiv:1509.01122*, 2015.

[27] J. Sun, S. Kim, S. Lee, Y. Kim, and S. Ko, "Reverse and boundary attention network for road segmentation," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCV Workshop)*, 2019.

[28] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1013–1020.

[29] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 4, pp. 834–848, 2018.

[30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] "Jetson tx2 module." [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2

[33] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 325–341.

[34] G. Dong, Y. Yan, C. Shen, and H. Wang, "Real-time high-performance semantic image segmentation of urban street scenes," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 22, no. 6, pp. 3258–3274, 2021.

[35] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision (IJCV)*, vol. 129, no. 11, pp. 3051–3068, 2021.

[36] S. Kumaar, Y. Lyu, F. Nex, and M. Y. Yang, "Cabinet: efficient context aggregation network for low-latency semantic segmentation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13517–13524.

[37] R. P. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," *arXiv preprint arXiv:1902.04502*, 2019.

[38] Y. Li, X. Li, C. Xiao, H. Li, and W. Zhang, "Eacnet: Enhanced asymmetric convolution for real-time semantic segmentation," *IEEE Signal Processing Letters (SPL)*, vol. 28, pp. 234–238, 2021.

[39] Y. Hong, H. Pan, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes," *arXiv preprint arXiv:2101.06085*, 2021.

[40] R. P. Poudel, U. Bonde, S. Liwicki, and C. Zach, "Contextnet: Exploring context and detail for semantic segmentation in real-time," *arXiv preprint arXiv:1805.04554*, 2018.

[41] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] X. Han, J. Lu, C. Zhao, S. You, and H. Li, "Semisupervised and weakly supervised road detection based on generative adversarial networks," *IEEE Signal Processing Letters (SPL)*, vol. 25, no. 4, pp. 551–555, 2018.

[44] Z. Chen and Z. Chen, "Rbnet: A deep neural network for unified road and road boundary detection," in *International Conference on Neural Information Processing (ICONIP)*, 2017.

[45] N. Garnett, S. Silberstein, S. Oron, E. Fetaya, U. Verner, A. Ayash, V. Goldner, R. Cohen, K. Horn, and D. Levi, "Real-time category-based and general obstacle detection for autonomous driving," in *IEEE International Conference on Computer Vision Workshop (ICCV Workshop)*, 2017, pp. 198–205.

[46] M. Oeljeklaus, *An Integrated Approach for Traffic Scene Understanding from Monocular Cameras*. VDI Verlag, 2021.

[47] R. Fan, H. Wang, P. Cai, J. Wu, M. J. Bocus, L. Qiao, and M. Liu, "Learning collision-free space detection from stereo images: Homography matrix brings better data augmentation," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 225–233, 2021.

[48] Y. Lyu, L. Bai, and X. Huang, "Road segmentation using cnn and distributed lstm," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019.

[49] J. Munoz-Bulnes, C. Fernandez, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection," in *20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 366–371.

[50] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[51] S. Zhang, Z. Zhang, L. Sun, and W. Qin, "One for all: a mutual enhancement method for object detection and semantic segmentation," *Applied Sciences*, vol. 10, no. 1, p. 13, 2019.

[52] Q. Wang, J. Gao, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection,"

IEEE Transactions on Intelligent Transportation Systems (TITS), vol. 19, no. 1, pp. 230–241, 2018.

[53] F. A. L. Reis, R. Almeida, E. Kijak, S. Malinowski, S. J. F. Guimares, and Z. K. G. do Patrocnio, "Combining convolutional side-outputs for road image segmentation," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2019.

[54] J. Fritsch, T. Khnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[55] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[56] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letter (PRL)*, vol. 30, no. 2, pp. 88–97, 2009.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[58] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[59] F. Zohourian, B. Antic, J. Siegemund, M. Meuter, and J. Pauli, "Superpixel-based road segmentation for real-time systems using CNN," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2018.

[60] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 12, pp. 2481–2495, 2017.

[61] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351, 2015, pp. 234–241.

[62] L. Rosas-Arias, G. Benitez-Garcia, J. Portillo-Portillo, J. Olivares-Mercado, G. Sanchez-Perez, and K. Yanai, "Fassd-net: Fast and accurate real-time semantic segmentation for embedded systems," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 2021.

[63] S. Yadav, S. Patra, C. Arora, and S. Banerjee, "Deep cnn with color lines model for unmarked road segmentation," in *IEEE International Conference on Image Processing (ICIP)*, 2017.

[64] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[65] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[66] C. Chen, G. Yao, L. Liu, Q. Pei, H. Song, and S. Dustdar, "A cooperative vehicle-infrastructure system for road hazards detection with edge intelligence," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[67] C. Chen, C. Wang, B. Liu, C. He, L. Cong, and S. Wan, "Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2023.