LRDNet: a lightweight and efficient network with refined dual attention decorder for real-time semantic segmentation

Mingxi Zhuang [a], Xunyu Zhong [a*], Dongbing Gu [b], Liying Feng [a],

Xungao Zhong [c] & Huosheng Hu [b]

[a] School of Aerospace Engineering, Xiamen University, Xiamen 361005, China
[b] School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK
[c] School of Electrical Engineering and Automation, Xiamen University of Technology, Xiamen 361024, China

Abstract

Most of the current popular semantic segmentation convolutional networks are focus on accuracy and require large amount of computation, which is using complex models. In order to realize real-time performance in practical applications, such as embedded systems and mobile devices, lightweight semantic segmentation has become a new need, where the network model should keep good accuracy in very limited computing budget. In this paper, we propose a lightweight network with the refined dual attention decorder (termed LRDNet) for better balance between computational speed and segmentation accuracy. In the encoding part of LRDNet, we offer an asymmetric module based on the residual network for lightweight and efficiency. In this module, a combination of decomposition convolution and deep convolution is used to improve the efficiency of feature extraction. In the decoding part of LRDNet, we use a refined dual attention mechanism to reduce the complexity of the entire network. Our network attained precise real-time segmentation results on Cityscapes, CamVid datasets. Without additional processing and pretraining, the LRDNet model achieves 70.1 Mean IoU in the Cityscapes test set. With a parameter value below 0.66 M, it can be up to 77 FPS.

Keywords: lightweight semantic segmentation; encoder-decoder; **r**esidual network; dual attention

1. Introduction

Autonomous driving or robot navigation is a complex task that requires perception, planning and execution in a constantly changing environment [1]. Over the last decade, deep learning has attracted the most attention, and it is viewed as an indispensable technology for this kind of tasks. In particular, semantic segmentation can be achieved with convincing results by using deep neural networks, which is important for scene perception and recognition. Semantic segmentation provides valued information about free space on the road for navigation, as well as relevant information such as lane markings and traffic signs for full awareness of the traffic conditions.

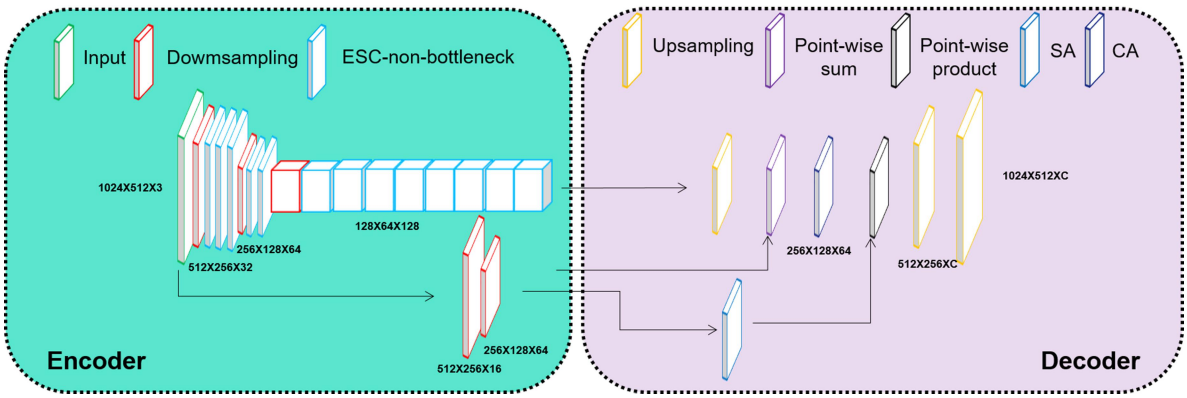

Fig. 1. General asymmetrical architecture of the proposed LRDNet. The encoder employs an FCN type network, while a dual attention is adopted in the decoder. C denotes the number of classes. (Best viewed in color)

Most of today's semantic segmentation research is mainly dedicated to improving the accuracy of the models [2-7], and significant progress along this direction has been made. These papers contribute to the research of semantic segmentation prediction. For example, many well-developed feature extraction modules are proposed based on FCN [3]. The core idea of these methods is to use the convolution layer instead of the fully connected layer in the classification network, and generate segmentation predictions by up sampling the output feature map. Although ordinary convolution is friendly in image segmentation, they have many limitations. All these methods, such as ResNet-101 [6] and VGG-16 [7], have large-scale backbone, in which the complex structure takes up a lot of GPU resources, resulting in slow reasoning speed. Therefore, these networks are not enough to meet the computing power and real-time performance of current mobile platforms.

In order to reduce the computational burden, it is very important to develop a lightweight and efficient semantic segmentation method for the real-time application of low-power GPUs. The use of a small-scale model can improve the inference speed and computational efficiency, and thus can reduce the cost of the equipment. Due to the high redundancy of large-scale deep learning networks, the efficiency of model structure and parameters is limited. Lightweight networks [8-17] are friendly to Edge Computing devices of self-driving cars, and the scalability of their applications is potentially high, such as for mobile robots. Generally, there are two types of lightweight networks: convolution factorization [8-14] and network compression [15-17]. The first one focuses on training small-scale networks directly, which is mainly based on the convolution factorization principle of decomposing standard convolution into group convolution. For example, InceptionNet [8] uses deconvolution as the backbone network to perform effective reasoning. Zhao et al. [9] proposes a cascade network, which combines advanced label guidance to improve performance. The second kind tends to reduce reasoning computation by compressing the pre-train network, including pruning [15], hashing [16] and quantization [17]. In order to further eliminate the redundancy, another method to

reduce CNN depends on the sparse coding theory. In some researches, symmetric encoder-decoder architectures, such as SegNet [2] and ERFNet [10], are used to reduce the number of parameters while maintaining the accuracy. Although some preliminary research work has been done for lightweight architecture network, the accuracy of many real-time semantic segmentation algorithms is not ideal due to the limitation of detail loss. Therefore, it is still an open research problem to pursue the best accuracy in the very limited computing budget.

Our research results in this paper show that the trade-off between the size, speed, and accuracy of a network model can be made by designing various decomposition convolutions. We use 1D decomposition convolution [10,18] and separable convolution [14] with dilation to replace ordinary convolution. As a result, the computational load of the model is greatly reduced, and the efficiency of the model is improved without losing too much accuracy. Recently, various attention mechanisms [19-26] have been successfully applied in many computer vision tasks. Such as SENet [19] and CBAM [20], these papers prove that weighting in space and channel is helpful to improve feature extraction. Inspired by this success, we optimize the network by adopting a refined dual attention mechanism by using high-level feature layers as the input of the channel attention mechanism, and low-level features as the input of the spatial attention mechanism. Compared with SENet [19], we optimize the global operation of 1x1 ordinary convolution to the local operation of 3x1 1D convolution. In this way, we introduce fewer parameters and less computation. This dual attention mechanism is conducive to improving the recognition accuracy.

In this paper, we aim to reduce the loss of detail, improve the inference speed, and achieve better balance between speed and accuracy. Motivated by this objective, an asymmetric and efficient encoder-decoder model is proposed for real-time semantic segmentation tasks, which we call LRDNet, as shown in Fig. 1. Our LRDNet consists of two parts: encoder and decoder networks. We develop an efficient decomposition convolution as a feature extraction network. We use a ResNet's residual module with skipping connection to prevent the network degradation and adopt a channel shuffling operation to enhance the robustness of the network. Through a channel split operation, 1D convolution and dilated separable convolution are combined in the grouping channel, which aims to reduce the computational cost of the model and deal with long-distance and short-distance features. In the decoding part, we combine the advantages of different feature layers to form a refined dual attention mechanism module to enhance the semantic segmentation effect. The contributions in this paper are summarized as follows:

(1) We propose an efficient split convolution with non-bottleneck (ESC-nbt). The combination of 1D convolution and expanded separable convolution can reduce the amounts of parameters, increase the speed of inference, and improve the accuracy of feature extraction for long distance features.

(2) The decoder employs a refined dual attention mechanism, which can reduce the complexity of the model and improve the accuracy of semantic segmentation.

(3) Our LRDNet demonstrates a good trade-off in terms of the parameter size, computational cost and accuracy on the Cityscapes dataset.

The remainder of this work is structured as follows. In Section 2, related work on decomposition convolution, depth separating convolution, dilated convolution, encoder-decoder and attention mechanism is introduced. Following that, a detailed illustration of the proposed LRDNet is presented in Section 3. Furthermore, in Section 4, the LRDNet's performance is tested in detail by performing a number of experiments. Key conclusions and highlights of results achieved are then presented in Section 5.

2. Related work

With the development of DCNN [27-29], more and more semantic segmentation networks based on DCNN have been proposed, and good performance has been demonstrated in various benchmarks. Among them, many methods constructed different convolution structures to reduce the network parameters and improve the utilization of high-level and low-level feature maps. Here, we briefly review the current existing works in this area.

2.1. Decomposition convolution and depth separable convolution

Decomposition convolution [10] is to decompose the ordinary k×k convolution into k×1 and 1×k convolutions. In this way, the amount of computational load can be greatly reduced even if the receptive field is the same, and the number of parameters is also reduced. Compared with ordinary convolution, the number of parameters of decomposition convolution is 2/k of it. For deep separable convolution, the typical network includes MobileNet [11,12]. In general, the work of depth separable convolution [14] is to decompose standard convolution into depth-wise convolution and point-wise convolution. The basic idea is to replace the standard convolution with a decomposed version and split the convolution into two separate layers. The first layer is called deep convolution, and it performs the lightweight filtering by applying a single convolution filter for each input channel. The second layer is a 1×1 convolution, called point-wise convolution [14], which is responsible for constructing new features by computing the linear combination of input channels. The basic principle is to divide the feature map into multiple sub-maps according to the number of channels. For

example, an N×H×W×C feature map is divided into C two-dimensional feature maps of N×H×W, and then convolved. The reduction in the number of parameters is ended with $(N^2-1)/N^2$ compared with a standard convolutional layer.

2.2. Dilated Convolution

In order to refine the high-order feature maps, the expanded convolution [30] introduces a dilated rate, which defines the stride between two adjacent kernel values. The receptive field range of k×k convolution with r = n is identical to the convolution of ((k-1) (n-1) + k) × ((k-1) (n-1) + k), and the number of parameters is constant. For example, for the 3×3 kernel with r = 3 and 7×7 kernel, their receptive fields are the same. Since different dilated rates can receive different proportions of information from high-level feature maps [10,18], they use the convolution-based multi-scale semantic information extractors in semantic segmentation tasks. However, the large dilation could result in local information lose.

2.3. Encoder-Decoder

The significant leap in the overall accuracy of semantic segmentation came after FCN [3] (Fully Convolutional Neural Network) was proposed. FCN uses only the convolutional layer from start to end and the extraction part is generally called an encoder. The subsequent up-sampling or de-convolution part is called a decoder. The encoder-decoder structure is an effective means to solve the problem of resolution degradation, and improves the utilization of high- and low-level feature maps at the same time. By gradually mapping low-level features to high-level features, the boundary and detail information of high-level feature mapping can be restored. FCN uses a skip structure to combine low-level and high-level feature maps. U-Net [31] uses a more efficient skip connection method to reduce the loss of boundary information. RefineNet [32] introduces many refinement blocks to combine low-level and high-level feature maps. Due to the gap in the mapping between the high and low feature layers, the high feature layer has a small amount of spatial information which could easily lose the boundary information, and the low feature layer has less semantic information. In order to mitigate the gap between them, we use a novel fusion module to improve the fusion effect of the high and low feature layers.

2.4. Attention mechanism

Various attention mechanisms have been successfully used in computer vision, especially in the field of semantic segmentation. Normally, there are two attention mechanisms: soft-attention mechanism [19-22] and

self-attention mechanism [23-26]. In the soft-attention mechanism, channel attention and spatial attention are often used for dealing with the task of semantic segmentation. The channel attention mechanism automatically obtains the importance of each feature channel through learning, and uses the obtained importance to enhance the features and suppress the features that are not important to the current task. The spatial attention mechanism is aimed at a single feature layer by weighting the spatial pixels of a feature layer to improve the network's ability to capture remote context information. This paper introduces a dual attention channel mechanism, i.e. channel and spatial attention mechanism, to improve the network's ability to capture both channel and context information. In regard to self-attention mechanism, such as in the Non-local Neural Networks [23], which proposes a non-local information statistics attention mechanism based on capturing the dependencies between long-distance features. For each query point, self-attention mechanisms firstly calculate the paired relationship between the query point and all points to obtain the attention map, and then aggregates the features of all points by weighted sum, so as to obtain the global features related to the query point, and finally the global features are added to the features of each query point respectively to complete the modeling process of remote dependence. Although the non-local method can improve the accuracy to a certain extent, the problem is that the amount of calculation is too large. So, we adopt a lightweight dual attention mechanism.

3. Method

3.1. Residual module with 1D convolution and depth separable convolution

Our module focuses on improving the efficiency of a residual network. Because the residual module has the advantage of preventing the network degradation, it is widely used in neural networks for image processing. The residual module can be formulated as:

$$x_{l+1} = x_l + F(x_l, W_l) \tag{1}$$

Where $F(x_l, W_l)$ is coming from:

$$F(x_l, W_l) = \sigma(W_l x_l) \tag{2}$$

$$\sigma(x) = \max(0, x) \tag{3}$$

In Equ. (1), (2) and (3), the variables variables x_{l+1}, x_l, W_l and σ stand for, respectively, the network output values, input value, the weight and activation function.

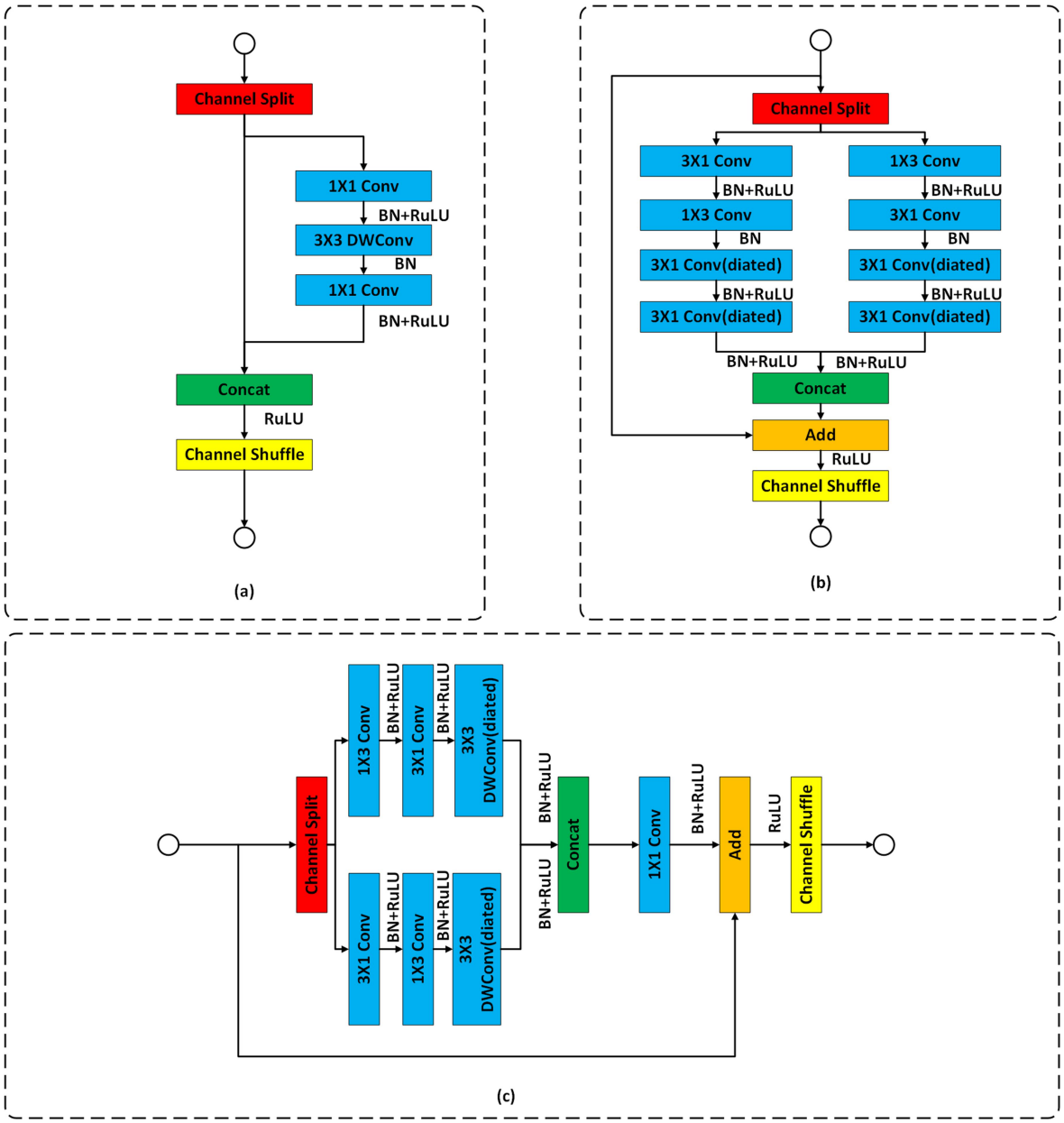

Fig. 2. Comparison of different residual layer modules. From left to right are (a) ShuffleNetV2 [33], (b) SS-nbt of LEDNet [34], (c) our ESC-nbt.

Since the ordinary standard 2D convolution kernel intersects channels through the connection relationship between the input and output, the redundancy caused by its parameters and memory size will affect the real-time performance in feature extraction. MobileNet uses a deep separable convolution, which is composed of

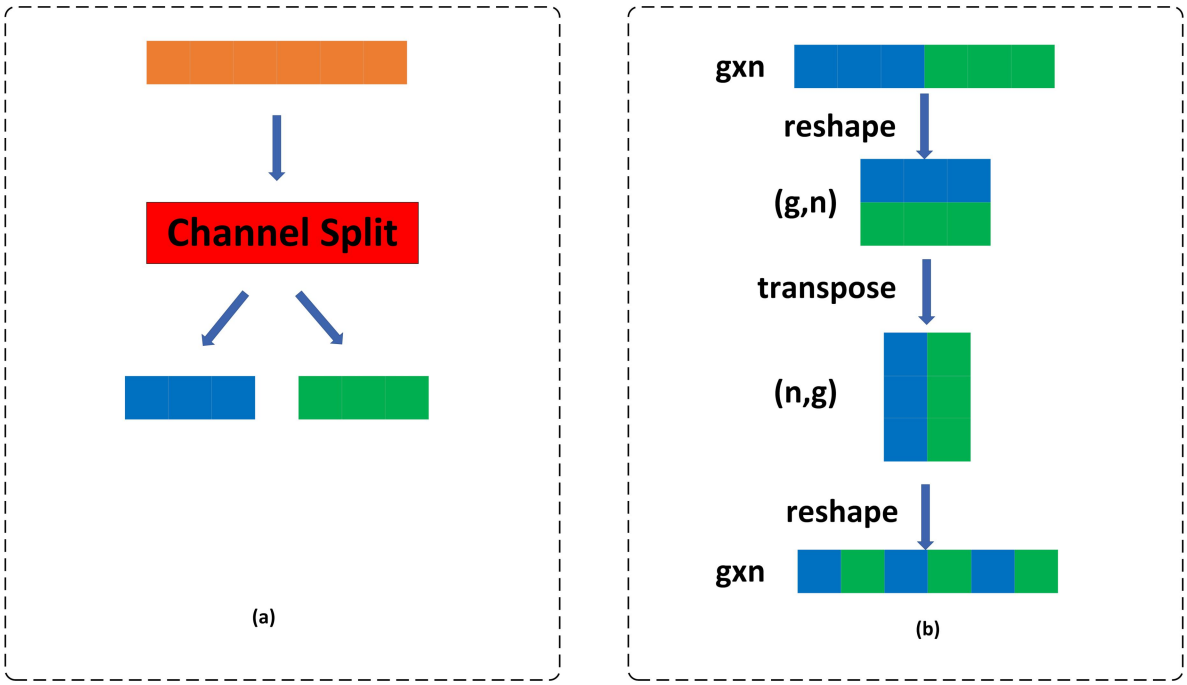

Fig. 3. Overview of the specific working process. From left to right are (a) channel splitting, (b) channel shuffling

1×1 point-wise convolution and deep convolution to learn the relationship between channels and the local relationship of each channel feature map. MobileNet greatly reduces the amount of network parameters and computational costs. At present, there are already some lightweight network modules that have achieved better feature extraction results. As shown in Fig. 2, ShuffleNetV2 [33] adopts the split shuffling strategy and the deep separable convolution to reduce the number of parameters and computational load. The SS-nbt module of LEDNet [34] uses 1D convolution to reduce the number of parameters. We propose a new lightweight and efficient split convolution structure ESC-nbt, as shown in Fig. 2 (c). The proposed ESC-nbt follows the strategy of channel splitting and channel shuffling. In our paper, the purpose of channel splitting, shown in Fig. 3 (a), is to divide the channels of the input feature map into two branches, which replaces the grouped convolution structure by extracting the features of the two branches separately. After channel splitting, it uses 1D convolution and deep convolution with a dilated rate to perform the short-distance and long-distance feature extraction respectively. It replaces the expanded 1D convolution in SS-nbt with an expanded depth convolution to reduce the number of parameters and computational cost. Deep convolution with dilated rate can enlarge receptive field, the receptive field of the $(l+1)$ th convolution layer can be written as:

$$RF_{l+1} = RF_l + \left(k^{'} - 1\right) * S_l \tag{4}$$

Here, the term $k^{'}$ and S_l is defined as:

$$k^{'} = k + (k-1)(d-1) \tag{5}$$

$$S_l = \prod_{i=1}^{l} \mathrm{Stride}_i \tag{6}$$

In Equ. (5) and (6), the variables $k^{'}$, k, d and Stride_i stand for, respectively, the size of dilated convolution, the size of ordinary convolution, the dilation and the step size of the sliding window.

The ESC-nbt also uses point convolution to improve the cross-intersection relationship between channels. Due to channel splitting, which simulates group convolution, will lead to loss of information, it is necessary to exchange channels. Since the information contained in different channels in the same group may be the same, if some channels are exchanged after different groups, then information can be exchanged. This makes the information about each group richer, and naturally more features can be extracted, which is conducive to getting better results. Channel shuffling, shown in Fig. 3 (b), is to reorganize the subsequent feature maps to ensure that information can flow between different groups. It is very easy to implement channel shuffle programmatically: assuming that the input layer is divided into g groups, the total number of channels is g × n, firstly split the channel dimension into two dimensions (g, n), and then transpose the dimensions into (n, g), and finally reshape into a dimension (g × n). Experimental results (Section 4) have shown that our ESC-nbt module can extract the features with high efficiency.

3.2. LRD network structure

As shown in Table 1, our LRDNet is different from LEDNet [34]. Our method uses an asymmetric encoder-decoder mechanism, where the encoder performs the feature extraction on down-sampling of the feature map. The subsequent decoder uses a dual attention mechanism to improve the quality of feature extraction, and uses the similar deconvolution to up-sample the feature map in order to maintain the input resolution. Specially, we also adopt a Res (Refined residual edge) as the input of spatial attention to reduce the boundary loss.

In addition to the ESC-nbt module, we also introduce a down-sampling module in the encoding part, as shown in Fig. 4 (a). This module uses the convolution with a step size of 2 and maximum pooling to reduce the resolution of a feature map, which also helps reduce the computational complexity, and increase the number of feature layer channels to enhance the semantic context information. The down-sampling module can be formulated as:

$$x_{l+1} = h(x_l) + F(x_l, W_l) \tag{7}$$

Table 1. The architecture of LRDNet. "Output Size" denotes the dimension of output feature maps, C is the number of classes.

Stage	Name	Type	Input	Output size
Encoder	d1	**Downsampling Unit**	image	512 X 256 X 32
	m1	**3 X ESC-nbt Unit**	d1	512 X 256 X 32
	d2	**Downsampling Unit**	m1	256 X 128 X 64
	m2	**2 X ESC-nbt Unit**	d2	256 X 128 X 64
	d3	**Downsampling Unit**	m2	128 X 64 X 128
	m3	**ESC-nbt Unit(diated r = 1)**	d3	128 X 64 X 128
	m4	**ESC-nbt Unit(diated r = 3)**	m3	128 X 64 X 128
	m5	**ESC-nbt Unit(diated r = 7)**	m4	128 X 64 X 128
	m6	**ESC-nbt Unit(diated r = 11)**	m5	128 X 64 X 128
	m7	**ESC-nbt Unit(diated r = 2)**	m6	128 X 64 X 128
	m8	**ESC-nbt Unit(diated r = 5)**	m7	128 X 64 X 128
	m9	**ESC-nbt Unit(diated r = 13)**	m8	128 X 64 X 128
	m10	**ESC-nbt Unit(diated r = 17)**	m9	128 X 64 X 128
Res	d4	**Downsampling Unit**	image	512 X 256 X 32
	d5	**Downsampling Unit**	d4	256 X 128 X 64
	sa1	**Spatial attention Unit**	d5	256 X 128 X 1
Decoder	up1	**Upsampling Unit**	m10	256 X 128 X 64
	m11	**up1 + d5**	up1, d5	256 X 128 X 64
	ca1	**Channel attention Unit**	m11	256 X 128 X 64
	sa2	**sa1 X ca1**	sa1, ca1	256 X 128 X 64
	output	**2 X Upsampling Unit**	sa2	1024 X 512 X C

The formula is similar to Equ. (1), but because the number of channels of $F(x_l, W_l)$ and x_l is different, the function h is to increase the number of channels by 1x1 convolution.

In the encoding part, we use the ESC-nbt module with a dilated rate greater than 1 in the low-level feature layer, which allows the network to have a larger receptive field in the process of feature extraction and to ensure the improvement of accuracy. Because the high-level feature map has low resolution with rich semantic information and the low-level feature layer has a high resolution, the boundary with more detailed information is maintained. In order to compensate for the boundary loss and improve the relationship between channels, we introduce a refined dual attention mechanism in the decoding part. The difference between spatial and channel attention is that the former assigns a weight to a spatial point of a feature map, while the

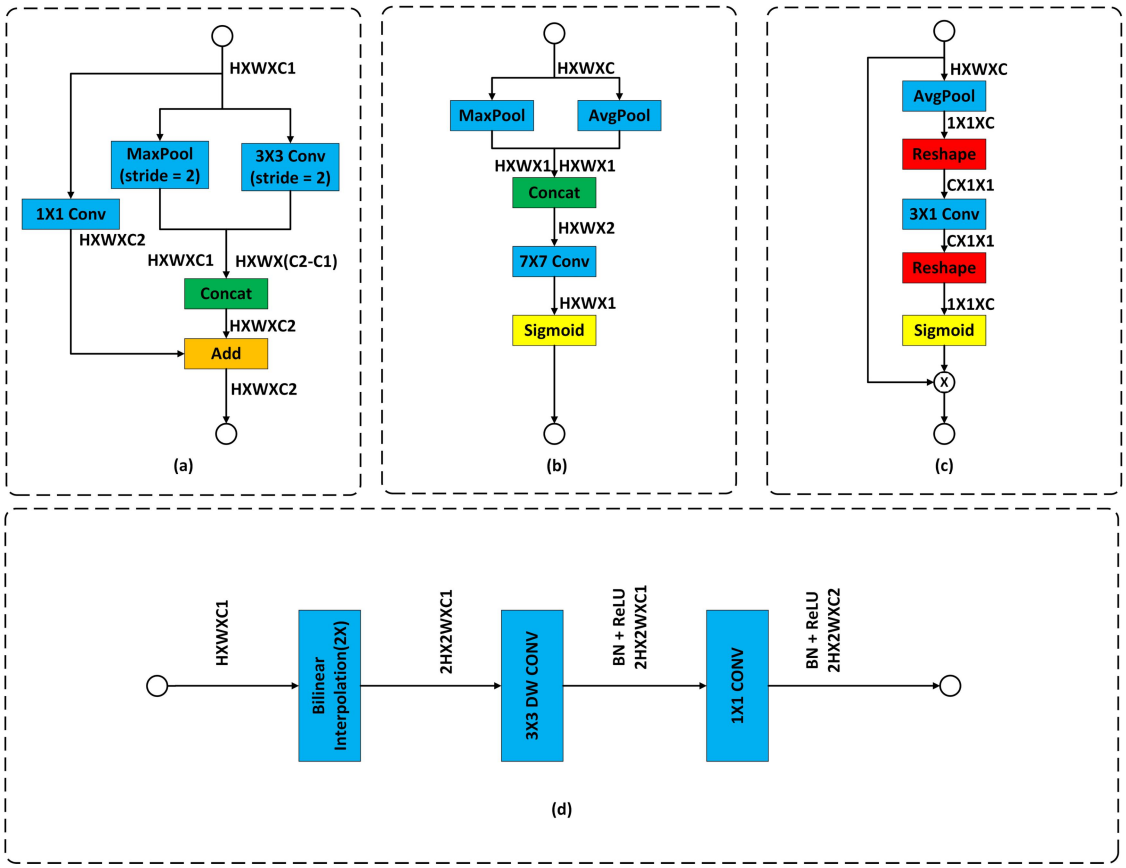

Fig. 4. The module of Decoder. From left to right are (a) Down-sampling, (b) Spatial attention Unit, (c) Channel attention Unit, (d) Up-sampling.

latter assigns different weights for different channels. The output feature map F_{sout} and F_{cout}, which are after spatial attention and channel attention respectively, can be calculated as follows:

$$F_{sout} = F \otimes M(u_{ij}) \tag{8}$$

$$F_{cout} = F \otimes M(u_c) \tag{9}$$

where $M(u_{ij})$ and $M(u_c)$ are coming from:

$$M(u_{ij}) = \frac{1}{C}\sum_{k=1}^{C} u_{ij}(k) + \max_{k\in[1,C]}(u_{ij}(k)) \tag{10}$$

$$M(u_c) = \frac{1}{HXW}\sum_{i=1}^{H}\sum_{j=1}^{W} u_c(i,j) \tag{11}$$

In Equ. (8) and (9), the feature map of input is F ∈ $R^{c\times h\times w}$, $M(u_{ij}) \in R^{1\times h\times w}$ is adopted as the spatial attention mask, and $M(u_c) \in R^{c\times 1\times 1}$ serves as the channel attention mask. And ⊗ denotes element-wise multiplication. In Equ. (10) and (11), the variables u_{ij}, u_c, C, H and W stand for, respectively, the pixel weight, the channel weight, channel number, image height and width.

For spatial attention shown in Fig. 4 (b), we first use a combination of average pooling and maximum pooling, and then reduce the number of channels through a 7×7 convolution to ensure that the number of channels in the input is the same as the output, and finally use the sigmoid function to normalize the result. The spatial attention mechanism, as shown in Fig. 1, is formed by down-sampling the original image twice through a residual edge, which could maintain more refined boundary information. For the channel attention, as shown in Fig. 4 (c), we first adopt the average pooling layer, and then reshape the 1×1×C feature map into C×1×1. The 1D convolution of 3×1 is used to improve the local relationship mapping among channels. Compared with SE [22], it has a lower computational load and does not significantly reduce the accuracy. Then the C×1×1 feature map is reshaped into 1×1×C. Finally, the sigmoid function is also used to normalize the weight value between channels. The input of the channel attention mechanism is the high-level features after feature extractions of the decoder. In addition, as shown in Fig. 4 (d), we use linear interpolation to enlarge the image resolution in the up-sampling module, then use 3×3 depth convolution to filter the image, and finally use 1×1 point convolution to modify the number of channels.

4. Experiments

4.1. Datasets

This paper chooses the widely used CityScapes dataset [35] and CamVid dataset [36]. The CityScapes dataset [35] is composed of high-quality pixel-level annotations of 5000 street scenes of 2048×1024 images, including 19 different types of object categories, and background information. There are 2975, 500 and 1525 images in the training set, validation set and test set, respectively. According to our lightweight method, we use 1024×512 sub-sampled images for testing. The CamVid dataset [36] consists of 367 training images, 101 validating images and 233 testing images with a resolution of 960×720, but we follow the setting as [2,18] using 480×360 resolution for training and testing.

In addition, we choose the DeepScene dataset [37] to verify the real-time performance of our model. The DeepScene dataset consists of 233 training images and 139 validation images of off-road imagery, which are densely labeled with six semantic categories: void, road, vegetation, grass, tree, sky, and obstacle. The resolution is 868×481, but the setting we followed is to use 448×448 images for training and testing.

4.2. Settings

The network structure parameters are the key for improving the performance of the network [38, 39]. An important conclusion obtained from [39] is that the value of learning rate is more sensitive to the stability and accuracy of neural network approaching the optimal state. In particular, a relatively small value of ξ may result in insufficient approximation accuracy. However, a large value of ξ often leads to insufficient robustness of convergence history. In the early stage of algorithm optimization, it will accelerate learning, making the model easier to approach the local or global optimal solution. However, in the later stage, there will be large fluctuations, even the value of the loss function hovers around the minimum value. Therefore, the concept of learning rate decay is introduced, that is, in the initial stage of model training, a larger learning rate will be used for model optimization, and with the increase of iterations, the learning rate will gradually decrease. This ensures that the model will not have too much fluctuation in the later stage of training, so as to be closer to the optimal solution. This is achieved by introducing a natural exponential decay equation:

$$\overline{\xi} = \xi(1 - \frac{\text{iter}}{\text{max\_iter}})^{\lambda} \tag{12}$$

Where iter stands for the current iteration step, max_iter is the decay step, ξ stands for the initial learning rate. λ denotes the decay rate. In addition, we found that the dilation of convolution also has a certain impact on the image feature extraction. Taking dilation without common divisor will not produce grid effect and can better improve the accuracy, so we change the dilation from 1,2,4,6,8,12,14,16 to 1,2,3,5,7,11,13,17.

For a fair comparison, we use Pytorch for training on GTX 1080ti GPU. All the training batch sizes are set to 6 and 1000 iterations of training are performed. The initial learning rate is 5×10^{-4}, "poly" learning rate momentum sampling 0.9, momentum and weight decay are set to 0.9 and 10^{-4}.

4.3. Metrics

(1) mIoU: The ratio of intersection and union of two sets of true and predicted values. It is the most typical comparison metrics in semantic segmentation.
(2) Params: Number of parameters of the CNN, which is involved in memory usage of the device.
(3) Pre-trained: Some methods are pre-trained in ImgeNet to improve accuracy.
(4) Time and Speed: Time (ms) is the inference time spent on GPU (on GTX 1080ti using Pytorch framework), which reflects the real-time performance of the model. Speed (FPS) = 1/Time.
(5) GFLOPS: Giga Floating-point Operations Per Second of a forward step.

Table 2. Comparison with the LEDNet on Cityscapes validation set, including accuracy and parameter size

Model	mIoU	Params(M)
LEDNet	69.6	0.94
Model A of LRDNet	71.5	**0.65**
Model B of LRDNet	**72.0**	0.66

Table 3. Individual category results on the CityScapes test set in terms of class and category mIoU scores. Compared with other approaches.

Method	Roa	Sid	Bui	Wal	Fen	Pol	TLi	TSi	Veg	Ter	Sky	Ped	Rid	Car	Tru	Bus	Tra	Mot	Bic	Cla	Cat
SegNet	96.4	73.2	84.0	28.4	29.0	35.7	39.8	45.1	87.0	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.1	35.8	51.9	57.0	79.1
ENet	96.3	74.2	75.0	32.2	33.2	43.4	34.1	44.0	88.6	61.4	90.6	65.5	38.4	90.6	36.9	50.5	48.1	38.8	55.4	58.3	80.4
ESPNet	97.0	77.5	76.2	35.0	36.1	45.0	35.6	46.3	90.8	63.2	92.6	67.0	40.9	92.3	38.1	52.5	50.1	41.8	57.2	60.3	82.2
CGNet	95.5	78.7	88.1	40.0	43.0	54.1	59.8	63.9	89.6	67.6	92.9	74.9	54.9	90.2	44.1	59.5	25.2	47.3	60.2	64.8	85.7
ERFNet	97.2	80.0	89.5	41.6	45.3	56.4	60.5	64.6	91.4	68.7	94.2	76.1	56.4	92.4	45.7	60.6	27.0	48.7	61.8	66.3	85.2
ICNet	97.1	79.2	89.7	43.2	48.9	61.5	60.4	63.4	91.5	68.3	93.5	74.6	56.1	92.6	51.3	72.7	51.3	53.6	70.5	69.5	86.4
LEDNet	97.1	78.6	90.4	46.5	48.1	60.9	60.4	71.1	91.2	60.0	93.2	74.3	51.8	92.3	61.0	72.4	51.0	43.3	70.2	69.2	86.8
Ours	**97.9**	**81.7**	**90.4**	43.0	**49.1**	58.9	**63.7**	66.8	**92.1**	**69.5**	**94.6**	**77.6**	**58.4**	**93.3**	53.2	62.5	**59.3**	**54.5**	65.9	**70.1**	**87.3**

4.4. Comparison with Other Methods

In order to reflect the advantages of the attention mechanism, we conduct some comparative experiments on the CityScapes validation set. We call the model without the attention mechanism in the LRDNet framework as Model A, and the model with the attention mechanism as Model B. In Table 2, we refer to Comparing Model A and Model B with LEDNet [34]. It can be seen that both of our model has higher accuracy and fewer model parameters, and the addition of a refined dual attention mechanism, i.e. Model B, can achieve even better results.

In the CityScapes test set, in order to demonstrate the advantages of LRDNet, we choose 8 different lightweight networks as benchmarks, including SegNet [2], ENet [18], ERFNet [10], ICNet [9], CGNet [40] And ESPNet [41], LEDNet [34].

In order to quantitatively analyze the segmentation result, we compared our method with other methods in each class. From the comparison results of Table 3, our method reaches a result of 70.1 mIoU in class and 87.3 mIoU in rough classification. We can see that in the above methods, our method obtains the best scores in 13 out of 19 categories, such as road, sidewalk, build, car, people, etc. In our design to ensure real-time performance, we do not obtain the best results in certain classes., such as wall, person, truck.

Table 4. Comparison with the state-of-the-art approaches on the CityScapes test set in terms of segmentation accuracy and implementing efficiency.

Model	Pre-trained	mIoU	Time (ms)	Speed (Fps)	Params (M)	GFLOPS
SegNet [2]	N	57.0	67	15	29.5	286
ENet [18]	N	58.3	**7**	**135**	**0.36**	**3.8**
ERFNet [10]	N	68.0	24	42	2.10	21.0
ESPNet [41]	N	60.3	9	112	0.40	5.2
ICNet [9]	Y	69.5	33	30	7.80	28.3
CGNet [40]	Y	64.8	20	50	0.50	6.0
LEDNet [34]	N	69.2	14	71	0.94	11.5
Our LRDNet	N	**70.1**	13	77	0.66	9.2

As shown in Table 4, our model, in comparison with SegNet, is 5 times faster than SegNet and 45 times smaller in model size, and with the accuracy improved by 23%. Compared to ERFNet, our model is 1.8 times quicker and the number of parameters is 3.2 times fewer, the accuracy is increased by 3.1%. Compared with LEDNet, our model parameters are reduced by 30%, and the accuracy rate is increased by 1.3%. These results lend credibility to our contention that a lightweight network can also achieve excellent results in terms of accuracy and speed.

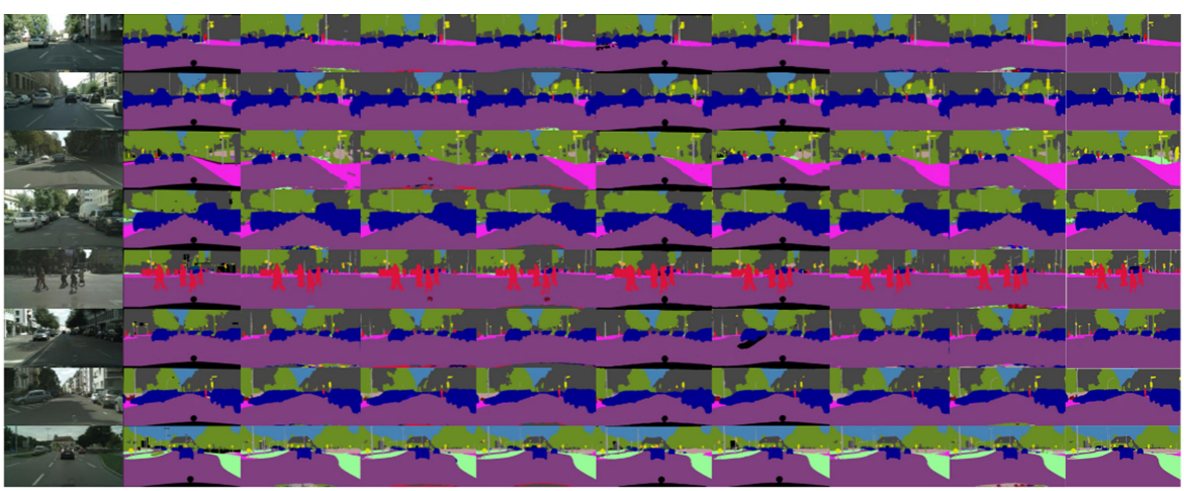

Fig. 5. Visual comparison on CityScapes validation dataset. From left to right are input images, ground truth, segmentation outputs from SegNet [2], ENet [18], ERFNet [10], ESPNet [41], ICNet [9], CGNet [40], LEDNet [34], and our LRDNet. (Best viewed in color)

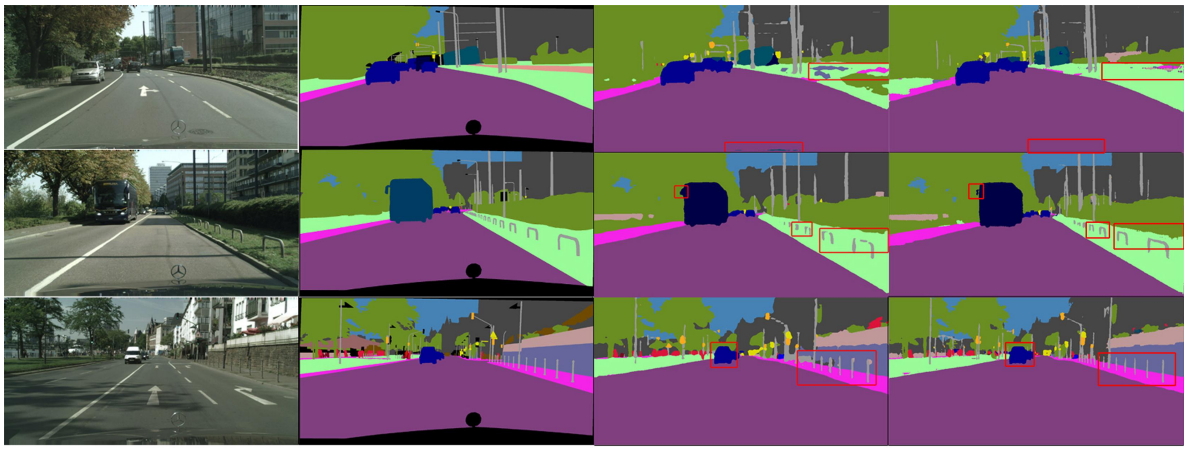

Fig. 6. Detail comparison on CityScapes validation dataset. From left to right are input images, ground truth, segmentation outputs from, LEDNet [34], and our LRDNet. (Best viewed in color)

Fig. 5 and Fig. 6 show the segmentation results on the validation set of Cistyscapes. Compared with other models, as shown in Fig. 5, our model achieves better results in the overall segmentation result. To qualitatively analyze the segmentation result, we show visual results of the LEDNet and LRDNet in Fig. 6. The red box areas in the above pictures show that our LRDNet has lower misclassification on roads and vegetation than LEDNet, and it has more refined edge segmentation results on poles and automobiles.

Table 5. Comparison with the state-of-the-art approaches on the Camvid test set in terms of segmentation accuracy and params.

Model	Pre-trained	Params (M)	mIoU
SegNet [2]	N	29.5	46.4
ENet [18]	N	0.36	51.3
ICNet [9]	Y	7.80	67.1
CGNet [40]	Y	0.50	65.5
LEDNet [34]	N	0.94	66.6
Our LRDNet	N	0.66	**69.7**

In the Camvid test set, we choose 5 different lightweight networks as benchmarks to demonstrate the advantages of LRDNet, including SegNet [2], ENet [18], ICNet [9], CGNet [40] And LEDNet [34]. As shown in Table 5, our method reaches a result of 69.7 mIoU. In comparison with SegNet [2], our model is with the accuracy improved by 50.2%. Compared with recently state-of-the-art LEDNet [34], the accuracy rate of our model is increased by 4.7%. We show some visual results of the LEDNet and LRDNet in Fig. 7. It can be seen that our method has more precise boundary segmentation and more accurate classification.

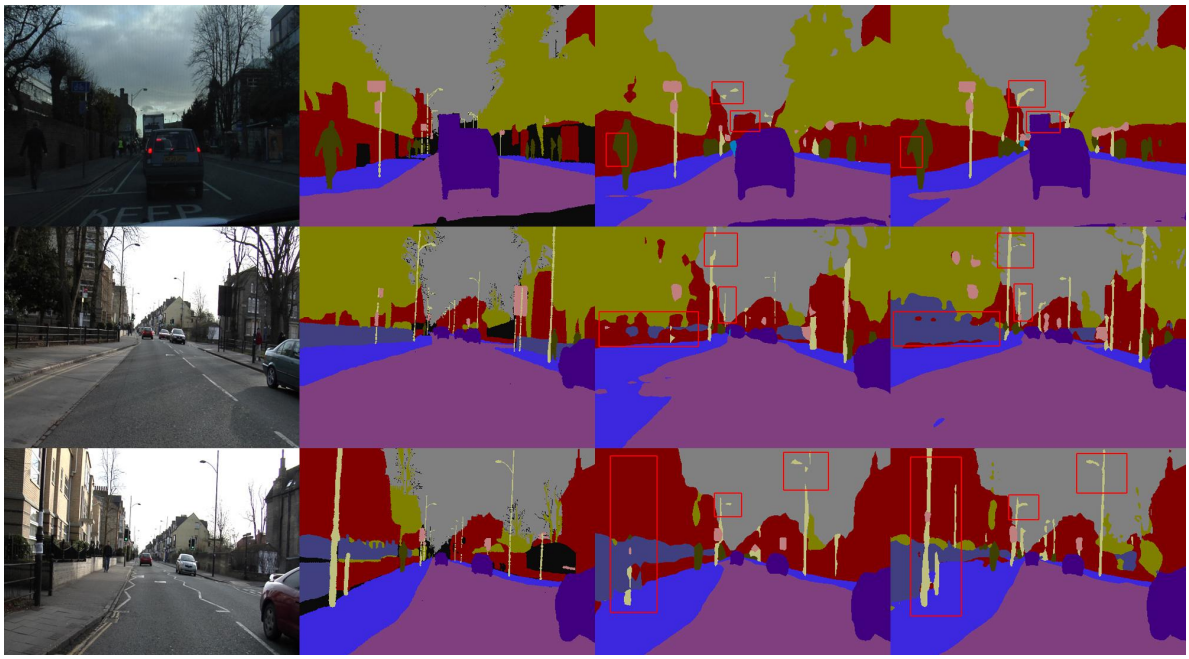

Fig. 7. Detail comparison on Camvid test dataset. From left to right are input images, ground truth, segmentation outputs from, LEDNet [34], and our LRDNet. (Best viewed in color)

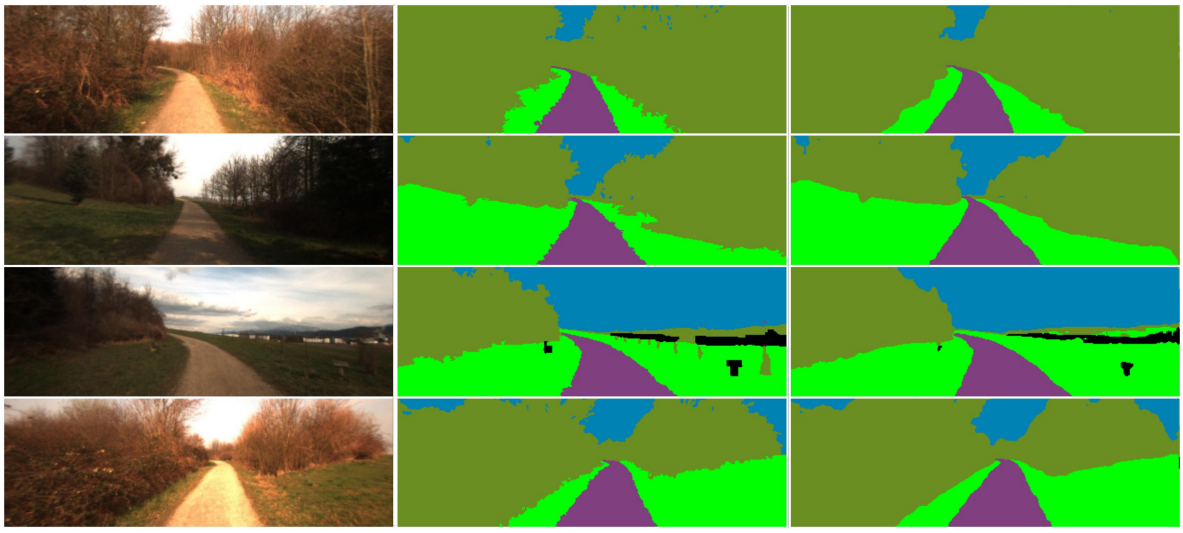

Fig. 8. Visual comparison on DeepScene validation dataset. From left to right are input images, ground truth, segmentation outputs from our LRDNet. (Best viewed in color)

Table 6. Comparison with approaches on the DeepScene validation dataset in terms of segmentation accuracy. The first three rows use a 300 × 300 image size, as in UpNet; the last two uses 448 × 448.

Model	mIoU
Upnet (RGB) [37]	79.86
cnns-fcn [42]	58.51
dark-fcn [42]	60.35
dark-fcn-448 [42]	60.61
LRDNet-488	87.60

In the DeepScene data set, Fig. 8 shows the segmentation outputs from our LRDNet. Regarding to the accuracy, LRDNet is 9.7%, which is higher than Upnet [37]. Compared with the Dark-fcn network [42], the accuracy of our model has increased by 44.5%. As shown in Table 6, we can see that our model achieves the best scores.

From the above experimental results, it can be seen that our model LRDNet has a better trade-off between speed and accuracy.

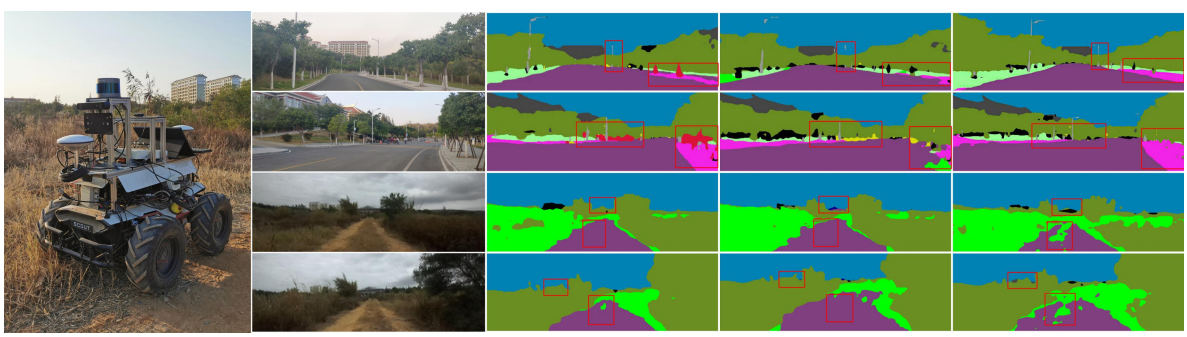

Fig. 9. Some segmentation results of practical scenes, From left to right are field robot, input images which were located in the Xiang'an campus of Xiamen University, segmentation outputs from ICNet, LEDNet, LRDNet, . (Best viewed in color. ■roads, ■sidewalks, ■buildings, ■high vegetation, ■unknown obstacles, ■terrain, ■poles, ■traffic signs, ■grass, ■cars, ■sky, ■people)

4.5. Scene experiment

Although results in Section 4.4 can prove the significant performance of the proposed method in qualitative and quantitative aspects, the practical application results are more ideal for reliable and convincing evaluation. Therefore, we test our proposal in real-world scenarios, as shown in Fig. 9 . In order to be applied in practical scenarios, we combine a city data set and a forest data set to form a new data set. Our semantic

segmentation uses 12 categories in the experiment to distinguish: unknown obstacles, roads, sidewalks, buildings, street lights (poles), traffic signs, people, cars, sky, high vegetation, terrain and grass. In the mixed data set, our model achieves 80.33 mIOU, which is a very good result.

Our experiment is carried out under the ROS system via subscribing to the image topic of a ZED2 camera. With 20Hz subscription frequency, our model can reach a publishing speed of 20Hz, which is able to perform in real-time semantic segmentation tasks. As shown in Fig. 9, we compare the segmentation results of ICNet, LEDNet and our LRDNet in real-world scenes. As can be seen from the red box in the above figure, our method reduces the misclassification of people and sidewalks in urban scenes, and has more refined boundary segmentation on buildings and poles. In the forest scene, our method has a more accurate classification in the low vegetation, which is conducive to unmanned ground vehicle shuttling through the grass (in the field environment, we think that the low vegetation is passable). To sum up, our model has generalization ability. However, the current data sets of urban and forest mixing are still scarce, so the effect of simultaneous application in different scenarios is poor, and this part of the work needs to be further improved.

5. Conclusion

This paper presents an asymmetrical, refined and efficient encoder-decoder model LRDNet for real-time semantic segmentation tasks. To strikes a good trade-off between speed and accuracy, we adopt an ESC-nbt module and a dual attention mechanism. From the results, it can be concluded the following.

1) The proposed efficient split convolution module (termed ESC-nbt), which adopts the combination of decomposition convolution and deep convolution, can indeed provide contributions to the efficient feature extraction.
2) The developed dual attention mechanism and refined residual edges have the capability of reducing boundary loss and improving accuracy with low parameters.
3) The proposed model (LRDNet) has high accuracy and generalization ability, and its real-time performance is verified

The entire network is an end-to-end framework. With regard to semantic segmentation on the Cityscapes and Camvid dataset, our model has a high accuracy and fewer parameters. Field experiments were also executed and the results verify the availability of our method in real application. In the near future, we plan to explore multi-scale feature fusion strategies with low computational cost in the decoding part.

Acknowledgements

This paper is supported by the Equipment Advance Research Funds (NO.61405180205), National Natural

Science Foundation of China (NO. 61703356), Fundamental Research Funds for the Central Universities (NO. 20720190129).

References

[1] B.K. Chen, C. Gong, J. Yang, Importance-Aware Semantic Segmentation for Autonomous Vehicles, Ieee Transactions on Intelligent Transportation Systems, 20 (2019) 137-148.

[2] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, Ieee Transactions on Pattern Analysis and Machine Intelligence, 39 (2017) 2481-2495.

[3] J. Long, E. Shelhamer, T. Darrell, Ieee, Fully Convolutional Networks for Semantic Segmentation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (Ieee, Boston, MA, 2015), pp. 3431-3440..

[4] K.M. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, Ieee Transactions on Pattern Analysis and Machine Intelligence, 42 (2020) 386-397.

[5] F. Yu, V. Koltun, T. Funkhouser, Ieee, Dilated Residual Networks, 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (Ieee, Honolulu, HI, 2017), pp. 636-644.

[6] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in:Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[7] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014)

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Aaai, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, Thirty-First Aaai Conference on Artificial Intelligence, (2017) 4278-4284.

[9] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia, ICNet for Real-Time Semantic Segmentation on High-Resolution Images, Computer Vision - Eccv 2018, Pt Iii, 11207 (2018) 418-434.

[10] E. Romera, J.M. Alvarez, L.M. Bergasa, R. Arroyo, ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation, Ieee Transactions on Intelligent Transportation Systems, 19 (2018) 263-272.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Ieee, MobileNetV2: Inverted Residuals and Linear Bottlenecks, 2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition, 2018), pp. 4510-4520.

[12] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q.V. Le, H. Adam, Ieee, Searching for MobileNetV3, IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, SOUTH KOREA, 2019), pp. 1314-1324.

[13] X. Zhang, X.Y. Zhou, M.X. Lin, R. Sun, Ieee, ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (Ieee, Salt Lake City, UT, 2018), pp. 6848-6856.

[14] F. Chollet, Ieee, Xception: Deep Learning with Depthwise Separable Convolutions, 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (Ieee, Honolulu, HI, 2017), pp. 1800-1807.

[15] S. Han, H. Mao, W.J. Dally, Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization a nd Huffman Coding, ICLR(2015), pp. 3--7.

[16] W. Chen, J.T. Wilson, S. Tyree, K.Q. Weinberger, Y. Chen, Compressing Neural Networks with the Hashing Trick, Compute r Science, (2015) 2285-2294.

[17] J. Wu, L. Cong, Y. Wang, Q. Hu, C. Jian, Quantized Convolutional Neural Networks for Mobile Devices, 2016 IEEE Confe rence on Computer Vision and Pattern Recognition (CVPR)2016).

[18] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation, (2016).

[19] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual Attention Network for Image Classification, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017) 6450-6458.

[20] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional Block Attention Module, Computer Vision - Eccv 2018, Pt Vii, 11211 (2018) 3-19.

[21] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q.J.I.C.C.o.C.V. Hu, P. Recognition, ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks, (2020) 11531-11539.

[22] J. Hu, L. Shen, G. Sun, Ieee, Squeeze-and-Excitation Networks, 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (Ieee, Salt Lake City, UT, 2018), pp. 7132-7141.

[23] X. Wang, R.B. Girshick, A. Gupta, K. He, Non-local Neural Networks, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2018) 7794-7803.

[24] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond, 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), (2019) 1971-1980.

[25] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, H. Shi, W. Liu, CCNet: Criss-Cross Attention for Semantic Segmentation, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), (2019) 603-612.

[26] S. Kundu, S. Sundaresan, AttentionLite: Towards Efficient Self-Attention Models for Vision, ArXiv, abs/2101.05216 (2021).

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, 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, 40 (2018) 834-848.

[28] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking Atrous Convolution for Semantic Image Segmentation, ArXiv, abs/1706.05587 (2017).

[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ArXiv, abs/1802.02611 (2018).

[30] F. Yu, V. Koltun, Multi-Scale Context Aggregation by Dilated Convolutions, (2015).

[31] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention, Pt Iii, 9351 (2015) 234-241.

[32] G.S. Lin, A. Milan, C.H. Shen, I. Reid, Ieee, RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation, 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (Ieee, Honolulu, HI, 2017), pp. 5168-5177.

[33] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design, ECCV2018).

[34] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, L.J. Latecki, Ieee, LEDNET: A LIGHTWEIGHT ENCODER-DECODER NETWORK FOR REAL-TIME SEMANTIC SEGMENTATION, 26th IEEE International Conference on Image Processing (ICIP), Taipei, TAIWAN, 2019), pp. 1860-1864.

[35] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, Ieee, The Cityscapes Dataset for Semantic Urban Scene Understanding, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (Ieee, Seattle, WA, 2016), pp. 3213-3223.

[36] G.J. Brostow, J. Fauqueur, R. Cipolla, Semantic object classes in video: A high-definition ground truth database, Pattern Recognition Letters, 30 (2009) 88-97.

[37] A. Valada, G.L. Oliveira, T. Brox, W. Burgard, Deep Multispectral Semantic Scene Understanding of Forested

Environments Using Multimodal Fusion, International Symposium on Experimental Robotics2016).

[38] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, C.L.P. Chen, Multiobjective Overtaking Maneuver Planning for Autonomous Ground Vehicles, IEEE Transactions on Cybernetics, (2020) 1-15.

[39] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, C.L.P. Chen, Design and Implementation of Deep Neural Network-Based Control for Automatic Parking Maneuver Process, IEEE Transactions on Neural Networks and Learning Systems, (2020) 1-14.

[40] T. Wu, S. Tang, R. Zhang, J. Cao, Y. Zhang, CGNet: A Light-Weight Context Guided Network for Semantic Segmentation, Ieee Transactions on Image Processing, 30 (2021) 1169-1179.

[41] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, H. Hajishirzi, ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation, Computer Vision - Eccv 2018, Pt X, 11214 (2018) 561-580.

[42] D. Maturana, P.W. Chou, M. Uenoyama, S. Scherer, Real-Time Semantic Mapping for Autonomous Off-Road Navigation, 2018).

Mingxi Zhuang received the B.E. degree in Xiamen University, Xiamen, China, in 2019. He is currently pursuing the M.D. degree in the Department of Automation, Xiamen University, Xiamen, China. His research interests include robotics, computer vision, deep learning and image processing.

Xunyu Zhong received the M.E. degree in mechatronics engineering from Harbin Engineering University, Harbin, China, in 2007, and the Ph.D. degree in control theory and control engineering from Harbin Engineering University, in 2009. He is currently an associate Professor with the Department of Automation, Xiamen University, Xiamen, China. He is an academic visitor of the School of Computer Science and Electronic Engineering, University of Essex, U.K., for one year from Sept. 2017. His current research interests include robot motion planning, visual servo and autonomous robots.

Dongbing Gu received the B.Sc. and M.Sc. degrees in control engineering from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in robotics from the University of Essex, Colchester, U.K. He was an Academic Visiting Scholar at the Department of Engineering Science, University of Oxford, Oxford, U.K., from 1996 to 1997. In 2000, he joined the University of Essex as a Lecturer, where he is a Professor with the School of Computer Science and Electronic Engineering. His current research interests include robotics, multi-agent systems, cooperative control, model predictive control, visual SLAM, wireless sensor networks, and machine learning.

Liying Feng received the B.E. degree in Beijing Jiaotong University, Beijing, China, in 2020. She is currently pursuing the M.D. degree in the Department of Automation, Xiamen University, Xiamen, China. Her research interests include robotics, computer vision, deep learning.

Xungao Zhong received the B.E. degree in electronic information engineering from Nanchang University, Nanchang, China, in 2007, the M.S. degree in electromechanical engineering from Guangdong University of Technology, Guangzhou, China, in 2011 and the Ph.D. degree in control theory and control engineering from Xiamen University, in 2014. He is currently an associate Professor with the School of Electrical Engineering and Automation, Xiamen University of Technology at Xiamen, China. His current research interests include machine learning, robotic visual servoing and application. He is selected as distinguished young scientific research talent of Fujian province, China, in 2018.

Huosheng Hu received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993. Currently, he is a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., leading the Robotics Group. He has authored over 500 research articles published in journals, books, and conference proceedings. His research interests include autonomous robots, human–robot interaction, multi-robot collaboration, embedded systems, pervasive computing, sensor integration, intelligent control, cognitive robotics, and networked robots. Prof. Hu is Fellow of the Institute of Engineering and Technology, Fellow of the Institution of Measurement and Control, and a Chartered Engineer in the U.K. He currently serves as Editor-in-Chief for the International Journal of Automation and Computing, Editor-in-Chief of MDPI Robotics Journal, and an Executive Editor for the International Journal of Mechatronics and Automation.