

CSRNet: Cascaded Selective Resolution Network for Real-time Semantic Segmentation

Jing-Jing Xiong^{a,*}, Lai-Man Po^a, Wing-Yin Yu^a, Chang Zhou^a, Peng-Fei Xian^a, Wei-Feng Ou^a

^a*Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China*

Abstract

Real-time semantic segmentation has received considerable attention due to growing demands in many practical applications, such as autonomous vehicles, robotics, etc. Existing real-time segmentation approaches often utilize feature fusion to improve segmentation accuracy. However, they fail to fully consider the feature information at different resolutions and the receptive fields of the networks are relatively limited, thereby compromising the performance. To tackle this problem, we propose a light Cascaded Selective Resolution Network (CSRNet) to improve the performance of real-time segmentation through multiple context information embedding and enhanced feature aggregation. The proposed network builds a three-stage segmentation system, which integrates feature information from low resolution to high resolution and achieves feature refinement progressively. CSRNet contains two critical modules: the Shorted Pyramid Fusion Module (SPFM) and the Selective Resolution Module (SRM). The SPFM is a computationally efficient module to incorporate the global context information and significantly enlarge the receptive field at each stage. The SRM is designed to fuse multi-resolution feature maps with various receptive fields, which assigns soft channel attentions across the feature maps and helps to remedy the problem caused by multi-scale objects. Comprehensive experiments on two well-known datasets demonstrate that the proposed CSRNet outperforms the main-stream efficient semantic segmentation approaches by accuracy and can be performed in real-time reference.

Keywords: Semantic segmentation; Attention mechanism; Real-time inference; Deep neural networks.

*Corresponding author

URL: jingxiong9-c@my.cityu.edu.hk (Jing-Jing Xiong), eelmpo@cityu.edu.hk (Lai-Man Po), wingyinyu8-c@my.cityu.edu.hk (Wing-Yin Yu), chanzhou3-c@my.cityu.edu.hk (Chang Zhou), xian.pf@my.cityu.edu.hk (Peng-Fei Xian), weifengou2-c@my.cityu.edu.hk (Wei-Feng Ou)

1. Introduction

Semantic segmentation is a process of assigning dense semantic labels to each pixel in an image. Currently, semantic segmentation has gained remarkable improvement with deep learning, especially convolutional neural networks (CNN). Researchers are continually working on improving the segmentation accuracy and advance state-of-the-art performances (Yuan et al., 2019; Wang et al., 2020b; Tao et al., 2020; Yu et al., 2020b; Zhang et al., 2020; Fu et al., 2020). General semantic segmentation networks care less about inference speed and computation cost. Hence, they usually have large-scale backbones such as HRNet (Wang et al., 2020b), ResNet-101 (He et al., 2016), and Xception (Chollet, 2017). With complex structures and deep layers, the network is able to extract comprehensive features from larger spatial ranges and benefit feature representation. However, heavy backbones often have high computational complexity and low inference speed. In recent years, real-time semantic segmentation has received considerable attention due to growing demands in autonomous vehicles, robotics, etc. These applications put a strict requirement on inference efficiency with low latency to make real-time implementations (≥ 30 Frames Per Second) and expect competitive segmentation performance, which is not suitable to employ heavy backbones. Therefore, how to balance efficient inference speed and high segmentation accuracy becomes a challenging task.

Existing real-time semantic segmentation methods employ three strategies to accelerate the whole model: (1) Input resolution restriction. A down-sampling ratio γ (usually 0.5 or 0.75) is utilized for the original image to obtain a smaller input resolution. Under the same network architecture, smaller inputs will consume fewer computing resources, resulting in higher inference speed. However, the down-sampling process may interfere with the detection of small objects and lose spatial information around boundaries, which degrade segmentation performance. (2) Channel pruning. As the backbone network goes from shallow to deep, the resolution of feature maps is getting smaller and smaller, and the number of feature channels is increasing. Pruning the channel numbers can decrease computational cost and improve inference speed, yet it will weaken feature extraction ability. (3) Shallow model architectures. Real-time semantic segmentation methods focus on achieving a decent trade-off between efficiency and accuracy, which means the backbone network should be a lightweight structure. The main drawback of lightweight architectures is that the extracted features cannot achieve sufficient receptive fields. Generally, current approaches leverage fusion of the above strategy to increase inference speed for real-time semantic segmentation.

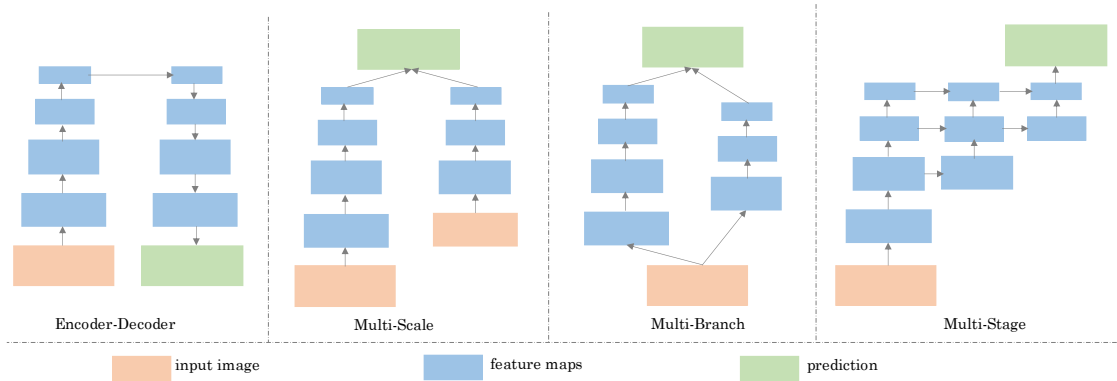


Figure 1: Typical network structures for real-time semantic segmentation. From left to right: encoder-decoder network, multi-scale network, multi-branch network, multi-stage network.

In addition to inference speed, segmentation accuracy is another criterion to measure the quality of real-time semantic segmentation algorithms. Designing an organized network structure is one of the keys to achieving high segmentation accuracy. Popular real-time segmentation structures include encoder-decoder networks (Paszke et al., 2016; Badrinarayanan et al., 2017; Orsic et al., 2019; Hu et al., 2020; Arani et al., 2021), multi-scale networks (Zhao et al., 2018; Mazzini, 2018; Orsic et al., 2019; Yu et al., 2020b; Li et al., 2021), multi-branch networks (Yu et al., 2018a; Poudel et al., 2019; Yu et al., 2020a), and multi-stage networks (Li et al., 2019b; Zhuang et al., 2019), which are visualized in Figure 1. Note that some networks may develop a combination of the above structures. Encoder-decoder networks have skip connections to recover high-resolution features from low-resolution features. Multi-scale networks take advantage of multi-resolution inputs to extract more spatial features from larger scale input and collect more semantic features from smaller scale input. Apart from multi-scale networks, multi-branch networks design multiple branches to achieve this goal, where a deeper branch is employed to capture the global context and a shallow branch is developed to learn spatial details. Lastly, multi-stage networks aggregate discriminative features by cascading structures and implement skip-connection to recover image details in the decoder module. The network architecture proposed in this paper is a multi-stage network. With the help of the multi-stage structure, our network can strengthen the feature representation ability progressively.

Since real-time semantic segmentation approaches exploit lightweight networks which have limited layers and insufficient receptive fields, it mainly faces three problems. First, the down-sampling operation leads to poor performances in segmenting some small structures like thin “poles”, small “traffic lights”, and “traffic signs”. An example of segmenting thin poles is shown

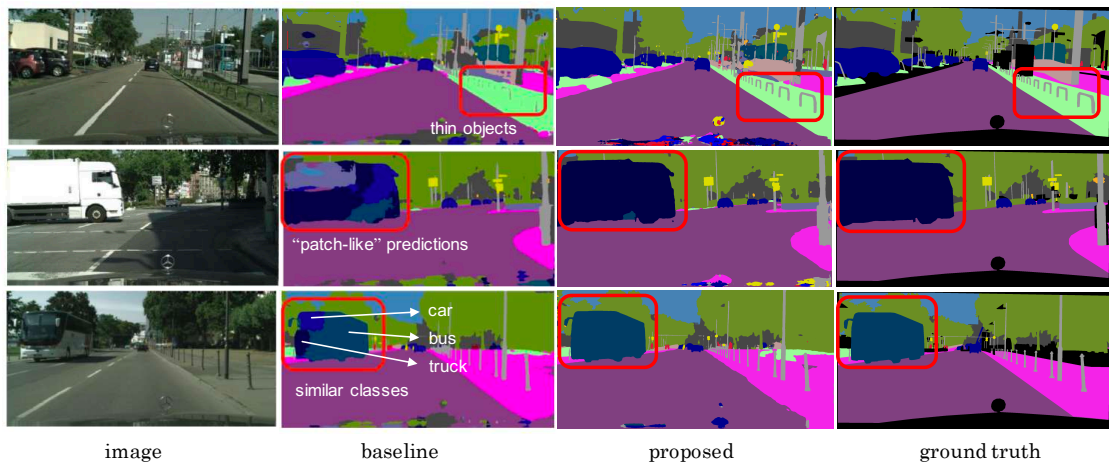


Figure 2: Typical weaknesses for real-time semantic segmentation approaches: thin objects segmentation, dominant objects segmentation and similar classes segmentation.

in the top row of Figure 2. Second, there may exist “patch-like” discontinuous predictions in the interior of the dominant objects (e.g., nearby buses, trucks, or trains close to the camera). The category of the dominant objects may be misclassified into other classes, as shown in the middle row of Figure 2. Third, due to the shallow network, the feature extraction ability is weaker than that of the deep structure without a doubt. Some classes with similar shape and appearance are confusing in classification for a neural network, e.g., “bus” and “truck”. Therefore, it is easy to misclassify the “bus” pixel into the “truck” pixel and vice versa, as shown in the bottom row of Figure 2. More details can be found in Section 4.5.

To tackle these issues and improve the performance for real-time segmentation, we propose a Cascaded Selective Resolution Network (CSRNet) by using progressive aggregation of features at different resolutions. In the feature aggregation process, CSRNet continues to incorporate global semantics and preserve edge details, thereby alleviating the problems of segmenting thin objects and similar classes. Specifically, the proposed CSRNet consists of two critical modules: (1) The Shorted Pyramid Fusion Module (SPFM) is employed to embed the global context information and enlarge the receptive field in each stage at the corresponding resolution of feature maps. The multiple adoption of SPFM enables rich context information embedding in the proposed network. (2) The Selective Resolution Module (SRM) based on the attention mechanism, is designed to fuse feature information of adjacent resolutions. SRM assigns soft channel attentions across the feature maps to handle the problem caused by multi-scale objects. More attention will be assigned to the lower-resolution feature map due to its sizeable receptive field if there is a dominant object

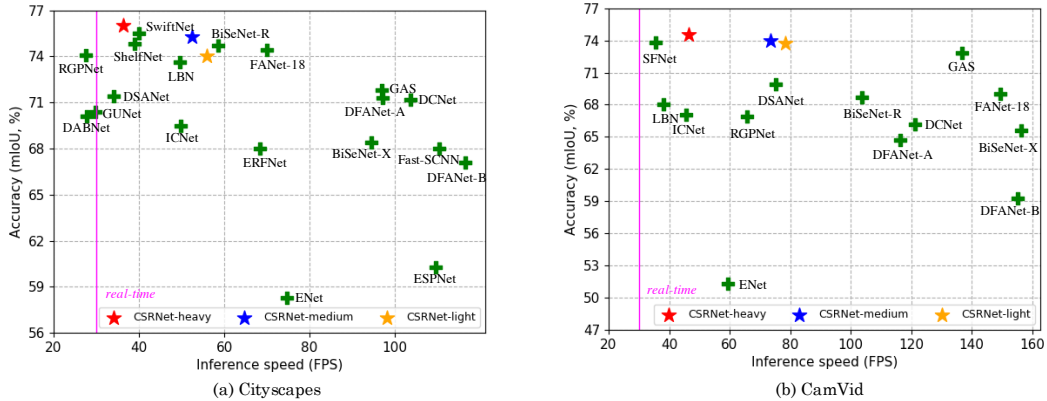


Figure 3: A comparison of speed-accuracy trade-off on Cityscapes and Camvid testing set. The pentagrams indicate our methods while green crosses indicate other methods. Methods on the right side of the purple vertical line represent real-time implementations. More details can be found in Table 1 and Table 2.

inside the input image. A comparison of speed-accuracy trade-off with the previous real-time methods on testing set of Cityscapes (Cordts et al., 2016) and Camvid (Brostow et al., 2008) is shown in Figure 3. The proposed method outperforms the previous efficient segmentation approaches by accuracy and can be run in real-time. The main contributions of our work are as follows:

- **Multiple-stage segmentation network.** We propose a multiple-stage segmentation network CSRNet to refine the feature maps progressively and improve the feature representation ability.
- **Selective resolution module.** We propose a novel selective resolution module (SRM), which exploits the attention mechanism to better aggregate multi-resolution feature maps with various receptive fields.
- **Remedy the problem caused by multi-scale objects.** The proposed network can alleviate the problem caused by multi-scale objects, showing the superiority in segmenting the dominant objects.
- **Experiments for real-time semantic segmentation.** The proposed network strikes a balance between segmentation accuracy and inference speed on two public challenging datasets: Cityscapes and CamVid.

2. Related work

This section will introduce the works related to our approach, such as real-time semantic segmentation, self-attention mechanism, and context information embedding.

2.1. Real-time Semantic Segmentation

Compared with the general semantic segmentation tasks, the main challenge of real-time semantic segmentation is to strike a balance between segmentation accuracy and inference speed. Popular approaches like input resolution restriction, channel pruning, or shallow model architectures can accelerate the model by reducing the computation burden. However, all of these strategies sacrifice the segmentation accuracy to some extent, and the loss of spatial details caused by decreasing the input resolution is irreparable. The most typical way to remedy the decline of accuracy is to insert feature aggregation modules. Shallow layers in the neural networks tend to learn rich spatial and edge features from the local area, while deep layers can learn significant abstract features like context information. Designing a feature fusion module is an intuitive solution to fuse the features from different layers and combine various feature information for better feature representation capabilities (Zhao et al., 2018; Yu et al., 2018a; Li et al., 2019b; Zhuang et al., 2019; Arani et al., 2021). DFANet (Li et al., 2019b) introduces the sub-network aggregation and sub-stage aggregation to achieve multi-scale feature propagation. BiSeNet (Yu et al., 2018a) extracts spatial features and categorical semantic features separately via two independent paths and develops a feature fusion module to fuse features from different paths.

Some works adopt multi-scale input to enhance segmentation quality. ICNet (Zhao et al., 2018) takes cascaded image inputs (i.e., low-, medium-, and high-resolution images), which exploits the efficiency of processing low-resolution images via a full segmentation backbone. Meanwhile, it can achieve high inference speed by feeding medium and high-resolution images into extremely light backbones. SwiftNet (Orsic et al., 2019) employs an image pyramid as input to improve segmentation accuracy. The lowest resolution of the image pyramid can enlarge the receptive field and regularize the encoder of the proposed model. Apart from multi-scale input, designing a proper loss function is another way to promote training performance. SwiftNet V2 (Oršić & Šegvić, 2021) increases the penalty for boundary pixels by proposing the boundary-aware loss to emphasize pixels near semantic boundaries. BiSeNet V2 (Yu et al., 2020a) inserts the auxiliary segmentation head to different semantic branches and trains the network with the auxiliary loss.

2.2. Self-attention Mechanism

The self-attention mechanism was first adopted to show the global dependencies of different words in a specific sentence in natural language processing (NLP) (Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017). In segmentation task, the attention module, which is generally adopted on the higher layer of a neural network, enables the network to perform better by refining feature maps via the feature denoising process (Xie et al., 2019). However, the traditional non-local block is unsuitable for real-time semantic segmentation due to its heavy computation cost when processing a high-resolution feature map. Many later methods have been proposed to reduce the computation burden of non-local blocks (Chen et al., 2018; Yuan et al., 2019; Huang et al., 2019b; Li et al., 2019d; He et al., 2019; Huang et al., 2019a; Cao et al., 2019). The authors in (Huang et al., 2019b; He et al., 2019; Huang et al., 2019a; Cao et al., 2019) implemented consecutive sparse attention to replace the dense attention in the non-local networks, while the authors in (Chen et al., 2018; Yuan et al., 2019; Li et al., 2019d) employed compact descriptors to reduce the computational complexity. In CCNet (Huang et al., 2019b), a criss-cross attention module was designed to generate a sparse attention map and exploits the recurrent criss-cross attention module to obtain richer and denser context information. APCNet (He et al., 2019) constructs multiple adaptive context modules, and each module transforms the feature map into a set of subregion representations. The context vector for each local position is calculated by leveraging the local affinity coefficients in each subregion. A2Net (Chen et al., 2018) first gathers key features from the entire feature space into global descriptors and adaptively distributes them to each position. OCRNet (Yuan et al., 2019) obtains the object region representation and then computes the relation between each pixel position and each object region.

In addition to the methods that model the relationships among positions at different spatial locations, channel attention performs dynamic channel-wise feature recalibration by modeling interdependencies between different channels. Channel attention is more computation-friendly than spatial attention. The spatial attention assigns the same attention weights on the channel dimension, while the channel attention assigns the same attention weights on the spatial dimension (Hu et al., 2018; Li et al., 2019c; Wang et al., 2020c). SENet (Hu et al., 2018) exploits global average pooling to generate channel-wise statistics and employs a sigmoid activation as a gating operation to determine the feature importance. Based on SENet, the authors in (Wang et al., 2020c) proposed ECA-Net, which avoids dimensionality reduction in the SE module and develops a local cross-channel interaction strategy to guarantee efficiency. SKNet (Li et al., 2019c) is inspired by SENet and Inception (Szegedy et al., 2015) as a dynamic selection mechanism of

receptive field size. In SK convolutions, multiple features are calculated from a single feature map with various kernel sizes to obtain different receptive fields. Soft attention across different channels is computed to select the generated multiple features adaptively. Besides, some works (Woo et al., 2018; Fu et al., 2019) combine spatial attention and channel attention simultaneously to achieve better segmentation performances.

2.3. Context Information Embedding

Convolutional neural networks are inherently limited to local receptive fields due to the fixed geometric structures. Two common ways to enlarge the receptive field include global average pooling and dilated convolution (Trembl et al., 2016; Chen et al., 2017b,a; Mehta et al., 2018, 2019; Dong et al., 2020). DFNet (Yu et al., 2018b) introduces the global context with global average pooling at the end of the backbone network. DeepLab (Chen et al., 2017a) employs dilated convolution to control the receptive field by adjusting the rate. Besides, to mitigate the inconsistency problem of segmenting the objects with multiple scales inside a single image, constructing a feature pyramid to capture diverse context information is an effective way. Several works have further integrated feature pyramid and multi-scale feature ensemble, such as pyramid scene parsing (PSP) (Zhao et al., 2017) and atrous spatial pyramid pooling (ASPP) (Chen et al., 2017b,a). PSPNet (Zhao et al., 2017) explores important global contextual information along with sub-region context, which fuses features under several pyramid scales by employing different scales of average pooling layers. ASPP generates multiple paths with various dilated convolutions to extract rich context information. DeepLab V3 (Chen et al., 2017b) incorporates the global context information by adding the image-level features through a global average pooling to the ASPP module.

3. Methodology

In this section, we will introduce the backbone network and the whole architecture of CSRNet, and then present two key modules in the proposed CSRNet: the SPFM and the SRM. Finally, we will explain how to obtain the medium and light network versions from the original network architecture.

3.1. Backbone Network

ResNet-18 (He et al., 2016), light Xception (Chollet, 2017), MobileNet (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019) are the popular lightweight backbones in the real-time

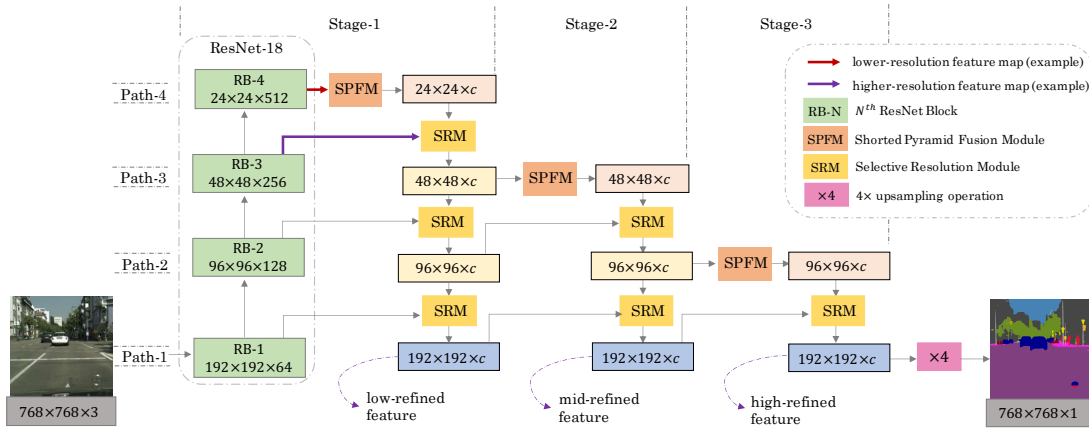


Figure 4: An overview of the architecture of the proposed CSRNet. It contains 4 paths in the horizontal direction and 3 stages in the vertical direction.

semantic segmentation. Researchers may also create a moderate depth structure that has low complexity and promote efficient training from scratch (Romera et al., 2017; Poudel et al., 2019; Li et al., 2019a; Lin et al., 2020; Elhassan et al., 2021). Xception and MobileNet series utilize depth-wise separable convolutions to reduce network parameters. From the computational cost perspective, the 3×3 depth separable convolution is computationally 8 to 9 times less than the conventional 3×3 convolution. MobileNet V2 (Sandler et al., 2018) introduces two blocks to improve the performance: Linear Bottleneck and Inverted Residual Blocks. MobileNet series is particularly suitable for embedded device designs like mobile applications. However, depth-wise separable convolutions are not directly supported in GPU firmware (the cuDNN library), and the latest cuDNN makes special optimizations for standard 3×3 convolution. Therefore, MobileNetV2 tends to be slower than ResNet-18 on GPUs’ performance. Consequently, we choose ResNet-18 as the backbone of our network since we run on GPU platform.

3.2. Network Architecture

The proposed network has 4 paths in the horizontal direction and 3 stages in the vertical direction, as shown in Figure 4. The resolutions of the feature maps in each path are the same. Apart from the backbone, each stage in this structure aggregates the feature information from the lower-resolution feature map to the higher-resolution feature map through fusing features in adjacent paths. Stage-1 has a U-shape structure with a ResNet-18 backbone. Given an input $\mathbf{X} \in \mathbb{R}^{H \times W \times N}$ where it is normally a RGB image consisting of dimension height H , width W and channel $N = 3$, the spatial resolutions of four feature maps generated by ResNet Blocks

(RB-1 to RB-4) are $4 \times$, $8 \times$, $16 \times$, $32 \times$ down-sampling of the input spatial resolution, and the channel numbers increase to 64, 128, 256 and 512, respectively. For real-time inference, the channel pruning is employed to reduce the channel numbers of the first three features in Stage-1. The number of output channels becomes 128 after the channel pruning process. The $32 \times$ feature map outputted by RB-4 is fed into the SPFM to extract global context information. This process helps refine the spatial information by adding the global context to the corresponding feature map.

Next, the feature maps in adjacent paths are aggregated through the SRM. The SRM takes the lower-resolution feature map and the higher-resolution feature map as input and assigns soft attentions across feature channels. Three consecutive SRMs combine the high-level features and low-level details and output a $4 \times$ down-sampling feature map, called the low-refined feature map. The process of Stage-2 and Stage-3 are similar to Stage-1, but we feed $16 \times$ and $8 \times$ feature maps outputted by previous stages into the SPFM, respectively. In this way, we can obtain the mid-refined feature map and high-refined feature map. The multi-stage process can be regarded as a feature refinement process, where the feature outputted by each stage becomes more and more representative and determinate. We will show the effectiveness of feature refinement in Section 4. Finally, the high-refined feature map goes through a 1×1 convolutional layer and a SoftMax operation to generate the segmentation prediction. This prediction is unsampled by a factor of 4 to obtain the final segmentation map.

3.3. Shorted Pyramid Fusion Module (SPFM)

The SPFM is designed to efficiently enlarge the receptive field at each stage. SPFM operates on the high-level feature map in the corresponding path at each stage, providing effective global contextual prior. In Stage-1, Stage-2, and Stage-3, SPFM is applied on $32 \times$, $16 \times$, $8 \times$ down-sampling feature maps, respectively. The structure of SPFM is similar to Pyramid Pooling Module (PPM) in PSPNet (Zhao et al., 2017), and we modify the architecture of PPM to reduce the computational burden and meet the requirements of real-time inference. Before the pyramid pooling operation with several different pyramid scales, the feature map (e.g., size: $a \times b \times c_o$) is passed to a 1×1 convolutional layer followed by batch normalization and ReLU activation function to transform the dimensionality to a fixed smaller number (size: $a \times b \times c$, $c \leq c_o$). $c = 128$ is a typical setting in our experiments. Then three feature maps with varying levels of details are produced by pooling operations over coarse spatial grids 2×2 , 4×4 , and 8×8 , following by convolution, upsampling and concatenation layers to form the feature representation

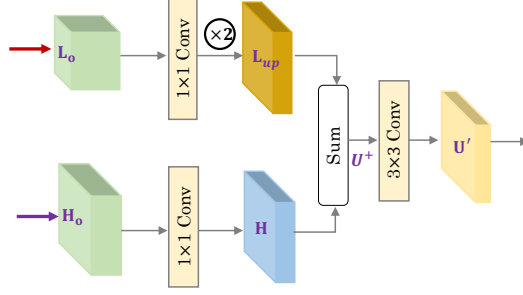


Figure 5: An example of a simple way to fuse the information of different feature maps with diverse spatial resolutions.

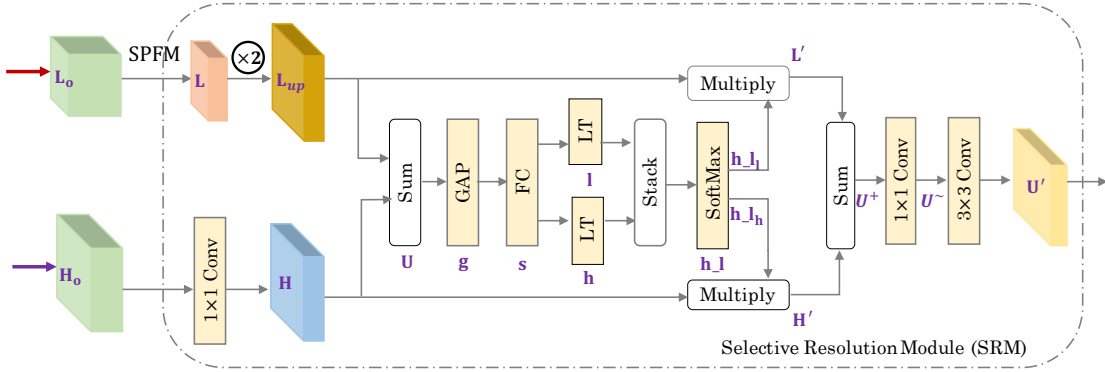


Figure 6: The details of SRM, taking the fusing process of the outputs of RB-3 and RB-4 in Stage-1 as an example. The SRM integrates the attention mechanism and aggregates the feature maps with different resolution and receptive fields. SPFM: Shorted Pyramid Fusion Module. GAP: Global Average Pooling. $\times 2$: 2 times upsampling using bilinear interpolation. FC: Fully Convolutional layer followed by a batch-normalization layer and a ReLU layer. LT: Linear Transformation. 1×1 or 3×3 Conv: 1×1 or 3×3 convolutional layer followed by a batch-normalization layer and a ReLU layer.

(size: $a \times b \times 2c$). Finally, the feature representation is fed into a 1×1 convolutional layer for channel reduction and obtain the final feature map (size: $a \times b \times c$). Compared with the PSPNet that operates PPM on the feature map of the last convolutional layer and receives the prediction directly, our SPFM works on three feature maps of different resolutions in the corresponding path. The global contextual priors of Stage-1 to Stage-3 are from different resolutions of feature maps. The low-refined feature map contains global information from the $32 \times$ feature map. Likewise, the mid-refined feature map and the high-refined feature map contain the global information from the $16 \times$, $8 \times$ feature map, respectively. Finally, the cascaded structure enables the network to propagate global information in different stages and embed more comprehensive context information.

3.4. Selective Resolution Module (SRM)

Feature aggregation is widely adopted in current real-time semantic segmentation approaches, yet few researchers explore how to aggregate two specific feature maps with different resolutions. The simplest way to merge the information of different feature maps with various spatial resolutions is to bilinearly upsample the lower-resolution feature map to the higher-resolution maps and do the element-wise summation. This process can be formulated as:

$$\mathbf{U}^+ = \mathbf{H} + \mathbf{L}_{up}. \quad (1)$$

If the channel numbers of two feature maps are not equal, a 1×1 convolution can be applied to match the channel number. After summation operation, a 1×1 or 3×3 convolution followed by batch normalization and ReLU activation is adopted to integrate the feature map further, as shown in Figure 5. This method is extremely efficient since only an upsampling and several convolution operations are included in the feature aggregation process. However, merely adding two feature maps damages the receptive information contained in the original feature maps.

The original lower-resolution feature map tends to have a larger receptive field and can extract more contextual information. Moreover, the receptive field of the feature map after SPFM is significantly enlarged. The higher-resolution feature map has a shallow receptive field, but it can extract more low-level information such as edges and texture from the local area. Instead of directly adding the feature maps, can we construct a module to determine the importance of the feature maps with different receptive fields when fusing these feature maps? Attention is a possible answer. We can learn soft channel attentions across feature maps with different resolutions, i.e., if there is a dominant object inside the input image, more attention desires to be assigned to the lower-resolution feature map due to its sizeable receptive field and vice versa.

Inspired by SKNet (Li et al., 2019c), we propose a feature aggregation module SRM, which exploits the attention mechanism to better aggregate feature maps with various resolutions and receptive fields. The structure of SRM is shown in Figure 6, taking the fusing process of the outputs (\mathbf{H}_o and \mathbf{L}_o) of RB-3 and RB-4 in Stage-1 as an example. Given an image input $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$, after 1×1 convolution and SPFM, $\mathbf{H} \in \mathbb{R}^{\frac{1}{16}H \times \frac{1}{16}W \times c}$ and $\mathbf{L} \in \mathbb{R}^{\frac{1}{32}H \times \frac{1}{32}W \times c}$ are fused through the SRM. In Figure 6, $H = W = 768$ and $c = 128$. First, \mathbf{L} is upsampled to $\mathbf{L}_{up} = \mathcal{F}_{up}(\mathbf{L})$, where \mathcal{F}_{up} is the bilinear interpolation, to match the dimensional size of \mathbf{H} for element-wise summation:

$$\mathbf{U} = \mathbf{H} + \mathbf{L}_{up}. \quad (2)$$

After combining the feature information with the simple summation, we seek dual-channel descriptors to capture channel-wise dependencies corresponding to the two feature maps. Therefore, we extract a global descriptor from the summed feature map and split it into two descriptors. We employ a global average pooling for each channel of \mathbf{U} independently to squeeze the global spatial information and obtain a descriptor $\mathbf{g} \in \mathbb{R}^c$. Formally, \mathbf{g} is generated by averaging \mathbf{U} through spatial dimensions $h \times w$, where $h = \frac{1}{16}H$ and $w = \frac{1}{16}W$. The t^{th} element of \mathbf{g} can be calculated by:

$$\mathbf{g}_t = \mathcal{F}_{gap}(\mathbf{U}_t) = \frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w \mathbf{U}_t(i, j). \quad (3)$$

To ensure the capability of the global descriptor, we employ a fully connected (FC) layer without channel dimensionality reduction to learn a more effective descriptor \mathbf{s} :

$$\mathbf{s} = \mathcal{F}_{fc}(\mathbf{g}) = \delta(\mathcal{B}(\mathcal{W}_1 \mathbf{g})), \quad (4)$$

where δ is the ReLU non-linearity, \mathcal{B} is the Batch Normalization (Ioffe & Szegedy, 2015), $\mathcal{W}_1 \in \mathbb{R}^{c \times c}$ contains the parameters of the fully connected layer. Then the channel descriptors \mathbf{h} and \mathbf{l} are obtained by applying two linear transformations to the incoming \mathbf{s} . Afterwards, to normalize \mathbf{h} and \mathbf{l} across the channel dimension, \mathbf{h} and \mathbf{l} are stacked, and a *SoftMax* operator is adopted to generate soft attention across channels $\mathbf{h} \cdot \mathbf{l}$:

$$\mathbf{h} = \mathcal{W}_2 \mathbf{s} + b_2, \mathbf{l} = \mathcal{W}_3 \mathbf{s} + b_3, \quad (5)$$

$$\mathbf{h} \cdot \mathbf{l} = \sigma(\mathcal{F}_{stack}(\mathbf{h}, \mathbf{l})), \quad (6)$$

where $\mathcal{W}_2 \in \mathbb{R}^{c \times c}$, $\mathcal{W}_3 \in \mathbb{R}^{c \times c}$, $b_2 \in \mathbb{R}^c$, $b_3 \in \mathbb{R}^c$ are the weights and bias of the two linear transformations. σ denotes SoftMax operation. We separate two normalized soft attention vectors $\mathbf{h} \cdot \mathbf{l}_h$ and $\mathbf{h} \cdot \mathbf{l}_l$ from $\mathbf{h} \cdot \mathbf{l}$. Next, the values of the two descriptors are multiplied to the higher-resolution feature map \mathbf{H} and lower-resolution feature map \mathbf{L} . Since the higher-resolution feature map and the lower-resolution feature map are separately outputted from different residual blocks, a simple weighted summation can discriminate the importance of the feature maps with various receptive fields when fusing, but it cannot integrate the information inside the feature maps effectively. To tackle this problem, two convolutional layers are employed to further aggregate the channel and spatial information of the merged feature map at the end of SRM. A $1 \times$

1 convolutional layer followed by a batch-normalization layer and a ReLU layer is utilized to aggregate information in channel dimension, and a 3×3 convolution followed by a batch-normalization layer and a ReLU layer further blends the features.

$$\mathbf{H}' = \mathbf{h}_h \cdot \mathbf{H}, \mathbf{L}' = \mathbf{h}_l \cdot \mathbf{L}_{up}, \quad (7)$$

$$\mathbf{U}^+ = \mathbf{H}' + \mathbf{L}' = \mathbf{h}_h \cdot \mathbf{H} + \mathbf{h}_l \cdot \mathbf{L}_{up}, \quad (8)$$

$$\mathbf{U}^\sim = \delta(\mathcal{B}(\mathcal{W}_4(\mathbf{H}' + \mathbf{L}'))), \quad (9)$$

$$\mathbf{U}' = \delta(\mathcal{B}(\mathcal{W}_5 \mathbf{U}^\sim)), \quad (10)$$

where \mathbf{h}_h and \mathbf{h}_l are from \mathbf{h} , denoting the soft attention vector for \mathbf{H} and \mathbf{L} . $\mathcal{W}_4 \in \mathbb{R}^{1 \times 1 \times c \times c}$ and $\mathcal{W}_5 \in \mathbb{R}^{3 \times 3 \times c \times c}$ are the weights for the 1×1 convolutional layer and 3×3 convolutional layer, respectively.

The main differences between SRM and Selective Kernel (SK) Unit lie in three aspects. (1) The input of an SK unit is a single feature map, followed by two convolutions with varying sizes of kernels to generate two feature maps with diverse receptive fields. In contrast, the inputs of SRM are two different feature maps which have various receptive fields already. Moreover, the receptive field of the feature map in the lower-resolution path after SPFM is significantly enlarged. The difference of receptive field of SRM inputs is larger than that of SK Unit inputs. (2) In the SK Unit, a compact feature is created from the global descriptor by a simple FC layer, with the shrinking of channel number for better efficiency. SRM does not reduce the channel number due to the observation found by (Wang et al., 2020c) that dimensionality reduction destroys the direct correspondence between the channel and its weight, while avoiding dimensionality reduction helps learn more effective channel attention. Besides, since we have applied channel reduction in previous steps, the dimensionality c in the FC layer is small. Keeping dimensionality unchanged still has a low computational capacity. (3) Both the SK module and SRM have two convolutional layers, but the intentions of convolution are absolutely different. The two convolutional operations at the beginning of SK module aim to produce two feature maps with different kernel sizes, while the two convolutional operations at the end of SRM further aggregate the feature information of the merged feature map. The convolution is significant to SRM, the effectiveness of which will be validated in the ablation study.

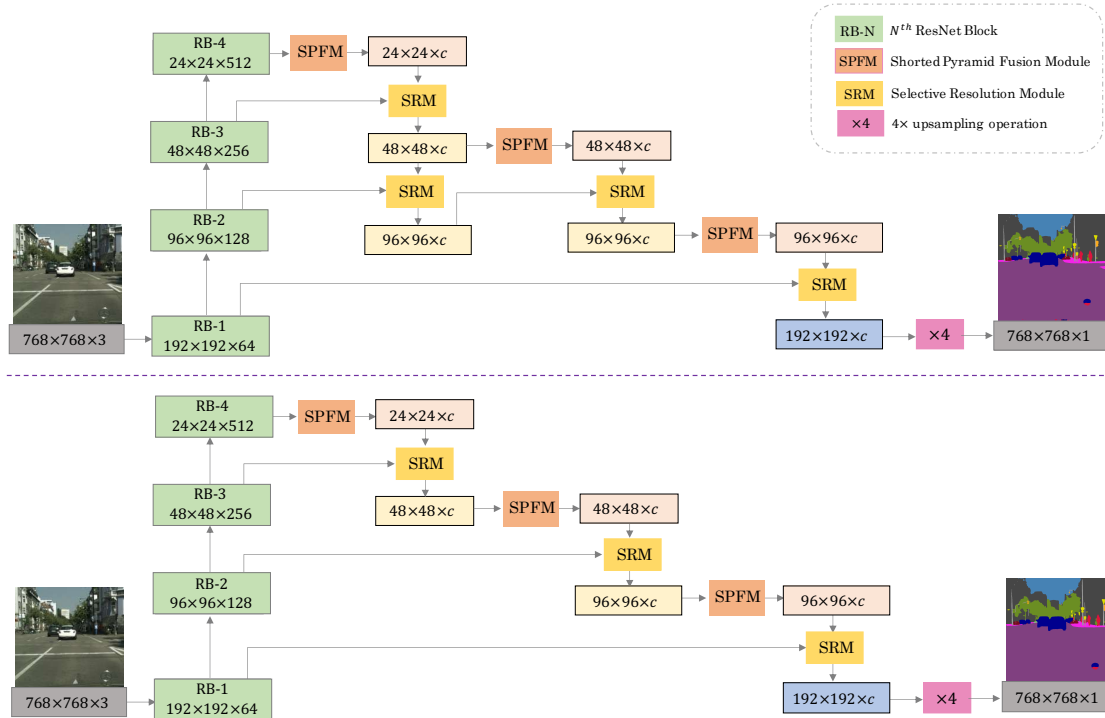


Figure 7: The architecture of medium and light network. The upper image shows the architecture of CSRNet-medium and the bottom image shows the architecture of CSRNet-light.

3.5. Medium and Light Network

From the network architecture, it is evident that Path-1 occupies a heavier computational capacity than Path-2, and Path-2 occupies a heavier computation capacity than Path-3, etc. Calculations on the high-resolution feature map usually require more computation resources than calculations on the low-resolution feature. The spatial input size is downsampled by factors of 4, 8, 16, and 32 in Path-1 to Path-4. Since the calculation on the feature map of $4 \times$ down-sampling resolution is very time-consuming, we can remove the 1×1 convolution and SRM of Stage-1 and Stage-2 in Path-1 and only keep the last 1×1 convolutional layer and SRM. The simplified network is called CSRNet-medium (the upper image in Figure 7). Similarly, the 1×1 convolution and SRM of Stage-1 in Path-2 can also be deleted, leading to the simplest network CSRNet-light (the bottom image in Figure 7). The medium and light version of networks avoid heavy computation on large resolution feature maps to further boost inference speed. The original network is called CSRNet-heavy. Note that the all versions of the proposed network are capable of real-time semantic segmentation. CSRNet-heavy extracts the global information from

high-level feature maps using SPFM, and aggregates the global semantics to the lowest-level feature map which contains rich spatial information. The multiple adoption of SPFM enable rich context information embedding in different stages and significantly enlarge the receptive field. Both CSRNet-medium and CSRNet-light reserve the design of SPFM in different paths and discard the SRM on larger resolution phases, which can decrease computational complexity and avoid dramatic drop in accuracy.

4. Experiments and Results

In this section, we introduce the basic information of the datasets and the implementation details of CSRNet. Ablation studies of different components in three versions of the network are conducted. We also provide a comprehensive explanation of experimental results. For quantitative evaluation, the standard metric of the mean intersection over union (mIoU) is adopted as the criteria of segmentation accuracy, and the Time and Frames Per Second (FPS) are adopted as the criteria of inference speed.

4.1. Datasets

Cityscapes Dataset: The cityscapes dataset (Cordts et al., 2016) focuses on the semantic understanding of urban street scenes. In this dataset, images were collected from a car perspective in 50 cities with good or medium weather conditions. This dataset includes 5000 images with fine annotations and 20,000 images with coarse annotations. We merely utilize the fine annotations to train and validate our model. The annotations contain 30 classes, 19 of which are chosen as the labels of semantic segmentation task. The resolution of images in this dataset is 1024×2048 . We follow the standard setting to partition the images into three sets: training, validation, and testing with 2975, 500, and 1525 images, respectively.

CamVid Dataset: Cambridge-driving Labeled Video Database (Brostow et al., 2008) is an early road scene dataset captured from the perspective of a driving automobile, containing 701 images with resolution 960×720 . The images and annotation quality of this database are lower than that of Cityscapes, which makes this dataset challenging. Images in this dataset are extracted from three different video sequences. We adopt the same setting in (Badrinarayanan et al., 2017) by grouping the 32 original classes into 11 classes through combining the same category of classes. The images are split into training, validation, and testing set with 367, 101, and 233 images, respectively. Following the pioneering work (Orsic et al., 2019), we train our model on the combined training and validation subsets and test the accuracy on the testing set.

4.2. Implementation Details

Training. We train all of our models using the Adam (Kingma & Ba, 2014) optimizer with the initial learning rate set to $4e-4$ in Cityscapes and $8e-4$ in CamVid. For all datasets, we adopt 8 batch size. We decay the learning rate with cosine annealing strategy (Ilya & Hutter, 2016) to the minimum value of $1e-6$ in the last epoch. The weight decay is set to $1e-4$. We update the pre-trained ResNet-18 backbone parameters with 4 times smaller learning rate and apply 4 times smaller weight decay than the randomly initialized parameters. We employ random horizontal flip, random scale, and random crop the image to a fixed size for training. The random scales contain 0.75, 1.0, 1.25, 1.5, 1.75, 2.0. We adopt the cropped resolution of 768×768 for Cityscapes and 960×720 for CamVid. We employ the conventional cross-entropy loss to train our network, with 250 epochs on Cityscapes and 400 epochs on CamVid. We do not adopt other evaluation tricks, like sliding-window evaluation and multi-scale testing.

Setup. We conduct all experiments based on PyTorch 1.1.0 framework with one NVIDIA GeForce GTX 1080 Ti under CUDA 10.2 and cuDNN 7.6.5.

4.3. Performances

This subsection compares our networks with other state-of-the-art methods on two benchmarks: Cityscapes and CamVid.

4.3.1. Accuracy and Inference Speed Comparison

We present the segmentation accuracy and inference speed of the proposed network on Cityscapes and CamVid dataset. Table 1 reports the comparison results of our method and state-of-the-art methods on the Cityscapes testing set. In this table, mIoU (val) and mIoU (test) indicate the mean IoU of Cityscapes validation set and testing set. The scaling factors of computing the normalized FPS are: 1 for GTX 1080 Ti, 0.44 for Tesla K40, 0.79 for Tesla K80, 0.61 For Titan X, 1.12 for Titan XP, and 1.37 for RTX 2080Ti. As shown in this table, our method achieves 75.3% mean IoU with a speed of 52.5 FPS under CSRNet-medium network and yields 76.0% mean IoU with 36.3 FPS under CSRNet-heavy network. Although CSRNet-heavy obtains a subtle higher mean IoU accuracy, it sacrifices nearly 16 FPS compared to the FPS achieved by CSRNet-medium. CSRNet-medium can run at high speed while accomplishing high-quality segmentation, which shows a better trade-off between efficiency and accuracy. Table 2 reports the comparison results of our method and state-of-the-art method on the CamVid testing set. As shown in this table, our method reaches 73.9% mean IoU with 73.5 FPS under CSRNet-medium

Table 1: Comparison of segmentation performance with several real-time state-of-the-art semantic segmentation methods on validation set and testing set of Cityscapes dataset. Table ordered by published time. “-” indicates that the corresponding result is not provided by the method.

Method	GPU	Backbone	mIoU (val)	mIoU (test)	Time (ms)	FPS	FPS norm
ENet (Paszke et al., 2016)	Titan X	None	-	58.3	13.0	76.9	74.7
ERFNet (Romera et al., 2017)	Titan X	None	70.0	68.0	24.0	41.7	68.4
ESPNet (Mehta et al., 2018)	Titan X	ESPNet	-	60.3	8.9	112.9	109.6
GUNet (Mazzini, 2018)	Titan Xp	DRN-D-22	69.6	70.4	30.0	33.3	29.7
BiSeNet-X (Yu et al., 2018a)	Titan Xp	Xception-39	69.0	68.4	9.5	105.8	94.5
BiSeNet-R (Yu et al., 2018a)	Titan Xp	ResNet-18	74.8	74.7	15.3	65.5	58.5
ICNet (Zhao et al., 2018)	Titan X	PSPNet-50	-	69.5	33	30.3	49.7
ESPNet V2 (Mehta et al., 2019)	Titan X	ESPNet V2	66.4	66.2	-	-	-
Fast-SCNN (Poudel et al., 2019)	Titan Xp	None	68.6	68.0	8.1	123.5	110.3
DABNet (Li et al., 2019a)	GTX 1080Ti	None	-	70.1	36.1	27.7	27.7
DFANet-A (Li et al., 2019b)	Titan X	Xception A	-	71.3	10	100	97.1
DFANet-B (Li et al., 2019b)	Titan X	Xception B	-	67.1	8.3	120	116.5
SwiftNet (Orsic et al., 2019)	GTX 1080Ti	ResNet-18	75.4	75.5	25.1	39.9	39.9
ShelfNet (Zhuang et al., 2019)	GTX 1080Ti	ResNet-18	-	74.8	25.6	39	39
FANet-18 (Hu et al., 2020)	Titan X	ResNet-18	75.0	74.4	13.9	72	69.9
LBN (Dong et al., 2020)	Titan X	LBN-AA	-	73.6	19.6	51	49.5
GAS (Lin et al., 2020)	Titan Xp	GCN-Guided	-	71.8	9.2	108.4	96.8
RGPNet (Arani et al., 2021)	GTX 2080Ti	ResNet-18	-	74.1	26.5	37.8	27.6
Lite-HRNet (Yu et al., 2021)	V100	Lite-HRNet-18	73.8	72.8	-	-	-
DCNet (Li et al., 2021)	GTX 2080Ti	ResNet-18	-	71.2	7.0	142	103.6
DSANet (Elhassan et al., 2021)	GTX 1080Ti	None	79.8	71.4	29.3	34.1	34.1
CSRNet-light	GTX 1080Ti	ResNet-18	76.1	74.0	17.9	56	56
CSRNet-medium	GTX 1080Ti	ResNet-18	76.6	75.3	19.0	52.5	52.5
CSRNet-heavy	GTX 1080Ti	ResNet-18	77.3	76.0	27.5	36.3	52.5

network and delivers 74.5% mean IoU with 46.4 FPS under CSRNet-heavy network. The segmentation accuracies on both datasets have achieved state-of-the-art performances and satisfied the real-time inference requirements.

4.3.2. Performances on Cityscapes Validation Set

Some visual examples of CSRNet-heavy, CSRNet-medium, CSRNet-light on Cityscapes validation set are shown in Figure 8. From this figure, we find that the segmentation qualities of the proposed networks are quite well, and the segmentation predictions become more and more convincing as the network becomes heavier. CSRNet-heavy has reached the most accurate results

Table 2: Comparison of segmentation performance with several state-of-the-art real-time semantic segmentation methods on testing set of CamVid dataset. Table ordered by published time. “-” indicates that the corresponding result is not provided by the method.

Method	GPU	Backbone	mIoU (test)	Time (ms)	FPS	FPS norm
ENet (Paszke et al., 2016)	Titan X	None	51.3	16.3	61.2	59.4
ICNet (Zhao et al., 2018)	Titan	PSPNet-50	67.1	36.0	27.8	45.6
BiSeNet-X (Yu et al., 2018a)	Titan Xp	Xception-39	65.6	5.7	175	156.3
BiSeNet-R (Yu et al., 2018a)	Titan Xp	ResNet-18	68.7	8.6	116.3	103.8
DFANet-A (Li et al., 2019b)	Titan X	Xception A	64.7	8.3	120	116.5
DFANet-B (Li et al., 2019b)	Titan X	Xception B	59.3	6.3	160	155.3
SwiftNet (Orsic et al., 2019)	GTX 1080Ti	ResNet-18	73.9	-	-	-
FANet-18 (Hu et al., 2020)	Titan X	ResNet-18	69	6.5	154	149.5
LBN (Dong et al., 2020)	Titan X	LBN-AA	68.0	25.4	39.3	38.2
GAS (Lin et al., 2020)	Titan Xp	GCN-Guided	72.8	6.5	153.1	136.7
SFNet (Li et al., 2020)	GTX 1080Ti	ResNet-18	73.8	28.2	35.5	35.5
RGPNet (Arani et al., 2021)	GTX 2080Ti	ResNet-18	66.9	11.1	90.2	65.8
DCNet (Li et al., 2021)	GTX 2080Ti	ResNet-18	66.2	6.0	166	121.2
DSANet (Elhassan et al., 2021)	GTX 1080Ti	None	69.9	13.2	75.3	75.3
CSRNet-light	GTX 1080Ti	ResNet-18	73.7	12.8	78.4	78.4
CSRNet-medium	GTX 1080Ti	ResNet-18	74.0	13.6	73.5	73.5
CSRNet-heavy	GTX 1080Ti	ResNet-18	74.5	21.6	46.4	46.4

about the majority of objects regardless of their sizes, such as the “traffic sign” in the first row, the “truck” in the second row, the “fence” in the third row, the “pole” in the fourth row, and the “person” in the last row.

4.3.3. Performances on Segmenting Thin, Dominant or Similar Objects

Comparison experiments of the segmentation performances on thin, dominant or similar objects with baseline, DANet (Fu et al., 2019), SwiftNet (Orsic et al., 2019), ShelfNet (Zhuang et al., 2019) are conducted. DANet is a state-of-the-art semantic segmentation method that models the semantic interdependencies in spatial and channel dimensions. SwiftNet and ShelfNet are methods for real-time semantic segmentation. The visualization results are shown in Figure 3.13. The first to the third column are duplicated from DANet, which are the original input images from the Cityscapes validation set, segmentation results of baseline, and segmentation results of DANet, respectively. The fourth to the seventh column are the segmentation results

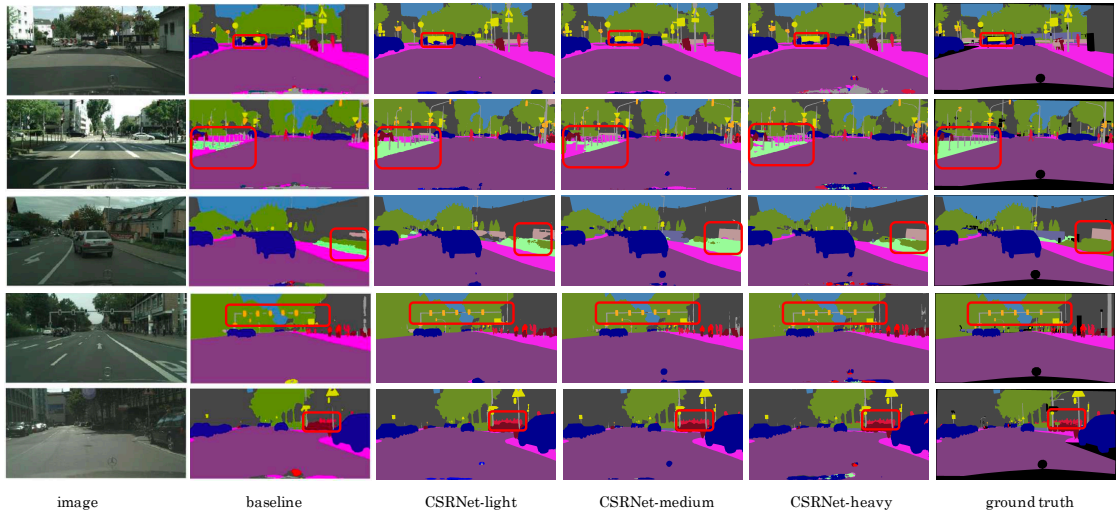


Figure 8: Visualization examples on the Cityscapes validation set. Best viewed in color. Please zoom in for more details.

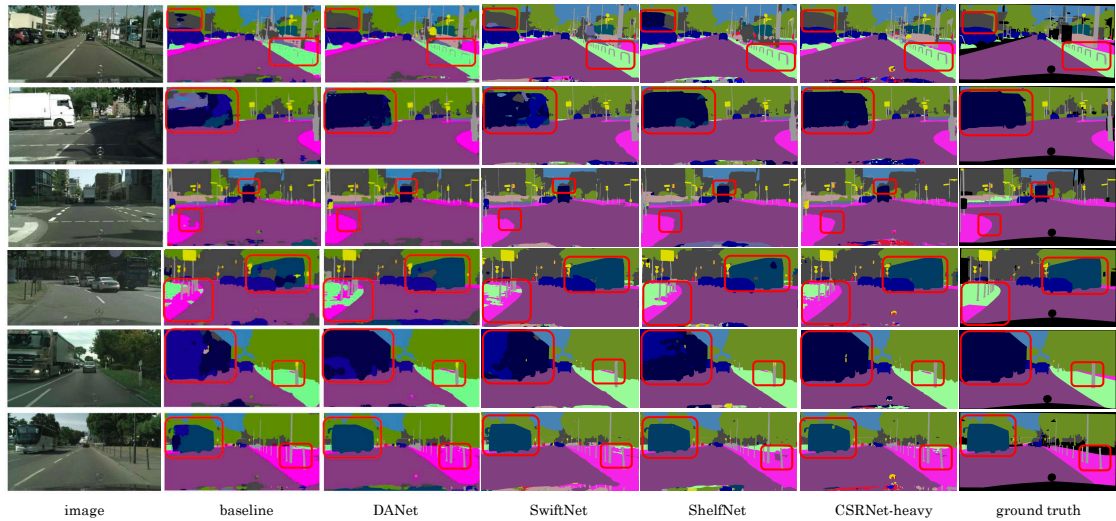


Figure 9: Performances on segmenting thin, dominant or similar objects on the Cityscapes validation set. Best viewed in color. Please zoom in for more details.

of SwiftNet, ShelfNet, the proposed CSRNet-heavy and the ground truth.

As shown in Figure 9, it is obvious that DANet, SwiftNet and ShelfNet achieve better performances on dominant objects than the baseline method. However, misclassification situations still exist in the interior of some dominant objects, such as the “bus” in the fourth row and the “truck” in the fifth row. Meanwhile, the segmentation of small objects can be further improved,

like the “pole” in the first row. The prediction for pole pixels is discontinuous and thinner compared with the ground truth. Moreover, the previous methods also misclassify some truck pixels into car pixel. On the contrary, CSRNet-heavy has alleviated these problems with larger receptive fields and more comprehensive feature maps. It achieves high semantic consistency in dominant objects, and the object boundaries are more clear and precise than other methods. CSRNet has a cascaded network structure to progressively refine the feature map. In the feature aggregation process, CSRNet incorporates global semantics and preserves edge details.

4.4. Ablation Study

This subsection introduces the ablation experiments to prove the effectiveness of different components in the proposed network. The following experiments were conducted with ResNet-18 as the backbone and trained on the Cityscapes training set and evaluated on the Cityscapes validation set.

4.4.1. Ablation Study of Simplified Networks

As discussed in Section 3, we remove some modules in the heavy network and propose two simple networks to boost the inference time. The accuracy comparisons of mean IoU, running Time and FPS are reported in Table 3 (**#3**, **#9**, **#13**). mIoU (val) indicates the mean IoU of Cityscapes validation set. CSRNet-heavy achieves 77.33 mIoU, which outperforms the medium and light network version by 0.78% and 1.22%. Although the segmentation accuracies of light networks are slightly lower than the heavy network, they achieve a better balance between accuracy and speed. CSRNet-medium and CSRNet-light achieve a speed of 52.5 and 56 FPS, respectively, which are 1.55 - 1.65 times FPS of the heavy network.

4.4.2. Ablation Study of Cascaded Stages

The proposed heavy network builds three similar stages, where each stage starts the feature aggregation process from different path. The output features from Stage-1 to Stage-3 are called low-refined feature map, mid-refined feature map, and high-refined feature map, respectively, with the same spatial resolution and channel numbers. Based on the structure of CSRNet-heavy, we construct two sub-structure networks to quantitatively evaluate the feature refinement process. One design ends at Stage-1 and outputs the low-refined feature map as the final feature map, following a 1×1 convolution to make the final prediction (**#1**). The other network ends at Stage-2 and outputs the mid-refined feature map as the final feature map, following a 1×1 convolution to make the prediction (**#2**). **#3** is the original heavy network. As can be seen from

Table 3: Ablation Study of CSRNet-heavy, CSRNet-medium and CSRNet-light on Cityscapes dataset. Numbers in bold represent the best results.

Model architecture	Experiment number	Discription	mIoU (val)	Time (ms)	Frame (FPS)
CSRNet-heavy	#1	Heavy-low-refined	75.95	16.6	60.2
	#2	Heavy-mid-refined	76.63	23.3	42.9
	#3	Heavy-high-refined (full model)	77.33	27.5	36.3
	#4	Heavy-high-refined-without SPFM	75.67	25.7	38.9
	#5	Heavy-low-refined-without attention	75.71	15.3	65.4
	#6	Heavy-mid-refined-without attention	76.06	20.9	47.9
	#7	Heavy-high-refined-without attention	76.17	24.1	41.5
	#8	Heavy-without 1×1 Conv in SRM	76.32	24.9	40.1
CSRNet-medium	#9	Medium (full model)	76.55	19.0	52.5
	#10	Medium-without SPFM	75.32	17.4	57.6
	#11	Medium-without attention	75.86	17.5	57.0
	#12	Medium-without 1×1 Conv in SRM	76.05	18.0	55.7
CSRNet-light	#13	Light (full model)	76.11	17.9	56.0
	#14	Light-without SPFM	74.89	16.2	61.8
	#15	Light-without attention	75.54	16.6	60.2
	#16	Light-without 1×1 Conv in SRM	75.59	17.0	58.7

Experiment **#1**, **#2**, and **#3** in Table 3, employing more stages result in higher segmentation accuracy (75.95 mIoU vs. 76.63 mIoU vs. 77.33 mIoU), but it will increase the inference time (60.2 FPS vs. 42.9 FPS vs. 36.3 FPS).

4.4.3. Ablation Study of SPFM

The SPFM is responsible for incorporating the global context information and enlarging the receptive field. To clarify the effect of SPFM on the whole network, we replace SPFM with a 1×1 convolution followed by a batch-normalization layer and a ReLU layer. In Table 3, **#4**, **#10**, **#14** are the counterparts of **#3**, **#9**, **#13**, respectively. The SPFM improves the mean IoU in all experiments by approximately 1.3% without sacrificing too much inference speed, proving that SPFM is an effective and efficient module in the proposed network.

4.4.4. Ablation Study of SRM

The SRM is designed to aggregate the feature maps with different resolutions and receptive fields. Compared with upsampling the lower feature map and summing the feature maps directly, the SRM exploits the attention mechanism, assigning different weights to the feature maps to adaptively select the feature receptive fields. In Table 3, **#5**, **#6**, **#7**, **#11**, **#15** are the counterparts of **#1**, **#2**, **#3**, **#9**, **#13** when we replace the attention mechanism in SRM with a simple addition operation (which is equivalent to $\mathbf{H}' = \mathbf{H}$ and $\mathbf{L}' = \mathcal{F}_{up}(\mathbf{L})$ in Figure 6). As

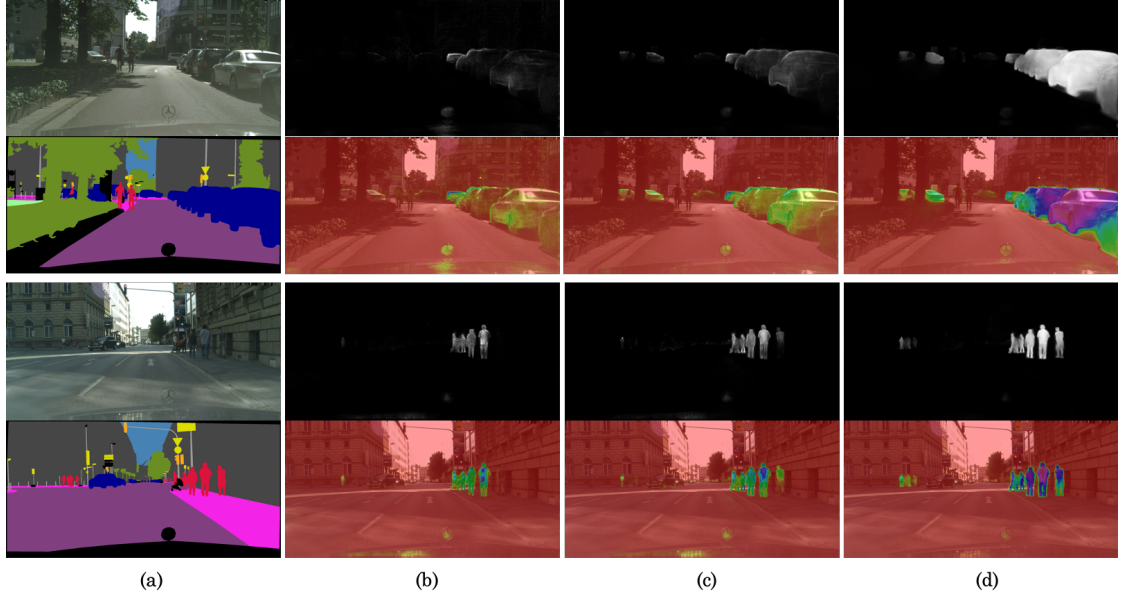


Figure 10: Visualization of feature refinement process on Cityscapes dataset. (a) The input images and the ground truth. (b) The Score-CAM result of the low-refined feature map. (c) The Score-CAM result of the mid-refined feature map. (d) The Score-CAM result of the high-refined feature map.

shown in Table 3, SRM brings additional calculations, resulting in a slight FPS drop. However, it leads to segmentation accuracy gain in all experiments, especially for **#3** vs. **#7**, showing an increase of 0.89% when we leverage SRM in CSRheavy-high-refined network. Moreover, adding the cascaded stages only brings a 0.46% mIoU increase with the addition operation (**#5** \rightarrow **#7**), while the accuracy increment for SRM when applying cascaded stages is 1.11% mIoU (**#1** \rightarrow **#3**).

4.4.5. Ablation Study of the Convolutional Layer in SRM

As discussed in Section 3, at the end of SRM, it employs an 1×1 convolution and a 3×3 to blend the features. The 1×1 convolution is important to SRM since it aggregates channel information after merging the two feature maps into one. In Table 3, **#8**, **#12**, **#16** are the counterparts of **#3**, **#9**, **#13**, respectively where we delete the 1×1 convolution in SRM and keep other parts intact. From this table, we observe that removing the 1×1 convolution causes decreased accuracy in all experiments.

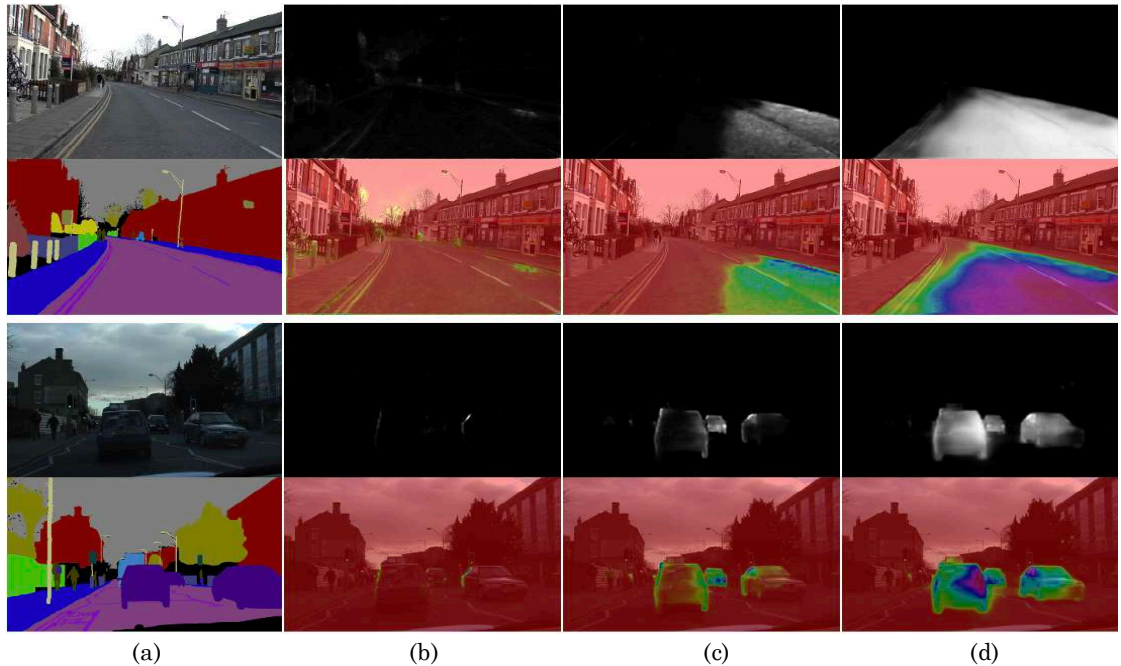


Figure 11: Visualization of feature refinement process on CamVid dataset. (a) The input images and the ground truth. (b) The Score-CAM result of the low-refined feature map. (c) The Score-CAM result of the mid-refined feature map. (d) The Score-CAM result of the high-refined feature map.

4.5. Visualization of the Feature Refinement Process

The class activation map (CAM) of a particular class represents the distinguished regions utilized by CNN to identify the class. This section exploits the Score-CAM method proposed in (Wang et al., 2020a) to generate the class activation map. The normalized activation map in Score-CAM works as a mask on the original input image, then a forward-passing score on the target class is obtained as the weight of the corresponding activation map. The Score-CAM is generated by a linear combination of score-based weights and activation maps. We employ Score-CAM at different convolutional layers in CSRNet-heavy for the target class “car” and “person” of two examples in the Cityscapes validation set. The visualization examples are shown in Figure 10, demonstrating how localization regions change qualitatively when performing Score-CAM to the outputted feature maps in Stage-1 to Stage-3. We find that the high-refined feature map obtains the best-looking visualizations, and localizations get progressively worse at the early stages. Some small objects cannot be localized by the early-stage feature maps, e.g., the cars under the trees on the left side of the road in the first row, and people under the left building in the third row. Besides, the boundary in the last column is clearer than the first two columns.

The high-refined feature map captures more semantic concepts together with more spatial details. Similarly, the CAMs at different convolutional layers for the target class “road” and “car” of two examples in the CamVid testing set are shown in Figure 11.

5. Conclusion

To fully investigate the feature fusion process of multi-resolution features with diverse receptive field, we propose a novel three-stage network CSRNet in this paper. The proposed network exploits the attention mechanism to aggregate feature maps with diverse spatial resolutions and receptive fields. Meanwhile, multiple adoption of global semantics embedding significantly enlarges the receptive field. Each stage aggregates feature information from the lower resolution to the higher resolution, and continues to incorporate context information and preserve edge details. The multi-stage process can be regarded as a feature refinement process, and the information flow between different stages is flexible. The output feature maps of each stage have the same spatial size and dimension, while the feature maps become more and more discriminative and recognizable. Extensive experiments demonstrate that the proposed CSRNet effectively improves the performance for real-time segmentation. We hope this work will foster further research in real-time semantic segmentation. In future work, we will extend the proposed network to efficient video semantic segmentation and exploit the spatial consistency clues in videos to ensure the speed and accuracy of segmentation.

References

- Arani, E., Marzban, S., Pata, A., & Zonooz, B. (2021). Rgpnet: A real-time general purpose semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3009–3018).
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39, 2481–2495.
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, .

- Brostow, G. J., Shotton, J., Fauqueur, J., & Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision* (pp. 44–57). Springer.
- Cao, Y., Xu, J., Lin, S., Wei, F., & Hu, H. (2019). Gnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (pp. 0–0).
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*, 834–848.
- Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017b). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, .
- Chen, Y., Kalantidis, Y., Li, J., Yan, S., & Feng, J. (2018). A2-nets: Double attention networks. *arXiv preprint arXiv:1810.11579*, .
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213–3223).
- Dong, G., Yan, Y., Shen, C., & Wang, H. (2020). Real-time high-performance semantic image segmentation of urban street scenes. *IEEE Transactions on Intelligent Transportation Systems*, .
- Elhassan, M. A., Huang, C., Yang, C., & Munea, T. L. (2021). Dsanet: Dilated spatial attention for real-time semantic segmentation in urban street scenes. *Expert Systems with Applications*, (p. 115090).
- Fu, J., Liu, J., Jiang, J., Li, Y., Bao, Y., & Lu, H. (2020). Scene segmentation with dual relation-aware attention network. *IEEE Transactions on Neural Networks and Learning Systems*, .

- Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3146–3154).
- He, J., Deng, Z., Zhou, L., Wang, Y., & Qiao, Y. (2019). Adaptive pyramid context network for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7519–7528).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V. et al. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1314–1324).
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, .
- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).
- Hu, P., Perazzi, F., Heilbron, F. C., Wang, O., Lin, Z., Saenko, K., & Sclaroff, S. (2020). Real-time semantic segmentation with fast attention. *IEEE Robotics and Automation Letters*, *6*, 263–270.
- Huang, L., Yuan, Y., Guo, J., Zhang, C., Chen, X., & Wang, J. (2019a). Interlaced sparse self-attention for semantic segmentation. *arXiv preprint arXiv:1907.12273*, .
- Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., & Liu, W. (2019b). Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 603–612).
- Ilya, L., & Hutter, F. (2016). Sgdr: stochastic gradient descent with restarts. *Learning*, *10*.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.

- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, .
- Li, G., Yun, I., Kim, J., & Kim, J. (2019a). Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. *arXiv preprint arXiv:1907.11357*, .
- Li, H., Xiong, P., Fan, H., & Sun, J. (2019b). Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9522–9531).
- Li, X., Wang, W., Hu, X., & Yang, J. (2019c). Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 510–519).
- Li, X., You, A., Zhu, Z., Zhao, H., Yang, M., Yang, K., Tan, S., & Tong, Y. (2020). Semantic flow for fast and accurate scene parsing. In *European Conference on Computer Vision* (pp. 775–793). Springer.
- Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., & Liu, H. (2019d). Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9167–9176).
- Li, Y., Liu, Y., & Sun, Q. (2021). Real-time semantic segmentation via region and pixel context network. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 7043–7049). IEEE.
- Lin, P., Sun, P., Cheng, G., Xie, S., Li, X., & Shi, J. (2020). Graph-guided architecture search for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4203–4212).
- Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, .
- Mazzini, D. (2018). Guided upsampling network for real-time semantic segmentation. *arXiv preprint arXiv:1807.07466*, .
- Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., & Hajishirzi, H. (2018). Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the european conference on computer vision (ECCV)* (pp. 552–568).

- Mehta, S., Rastegari, M., Shapiro, L., & Hajishirzi, H. (2019). Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9190–9200).
- Orsic, M., Kreso, I., Bevandic, P., & Segvic, S. (2019). In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12607–12616).
- Oršić, M., & Šegvić, S. (2021). Efficient semantic segmentation with pyramidal fusion. *Pattern Recognition*, *110*, 107611.
- Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, .
- Poudel, R. P., Liwicki, S., & Cipolla, R. (2019). Fast-scnn: fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, .
- Romera, E., Alvarez, J. M., Bergasa, L. M., & Arroyo, R. (2017). Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, *19*, 263–272.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).
- Tao, A., Sapra, K., & Catanzaro, B. (2020). Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, .
- Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schubert, P., Mayr, A., Heusel, M., Hofmarcher, M., Widrich, M. et al. (2016). Speeding up semantic segmentation for autonomous driving. In *MLITS, NIPS Workshop*. volume 2.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*, .

- Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., & Hu, X. (2020a). Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 24–25).
- Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X. et al. (2020b). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, .
- Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020c). Eca-net: efficient channel attention for deep convolutional neural networks, 2020 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. *IEEE*.
- Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–19).
- Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., & He, K. (2019). Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 501–509).
- Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., & Sang, N. (2020a). Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *arXiv preprint arXiv:2004.02147*, .
- Yu, C., Wang, J., Gao, C., Yu, G., Shen, C., & Sang, N. (2020b). Context prior for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12416–12425).
- Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018a). Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 325–341).
- Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018b). Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1857–1866).
- Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., & Wang, J. (2021). Lite-hrnet: A lightweight high-resolution network. *arXiv preprint arXiv:2104.06403*, .

- Yuan, Y., Chen, X., & Wang, J. (2019). Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, .
- Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R. et al. (2020). Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, .
- Zhao, H., Qi, X., Shen, X., Shi, J., & Jia, J. (2018). Icnets for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 405–420).
- Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2881–2890).
- Zhuang, J., Yang, J., Gu, L., & Dvornik, N. (2019). Shelfnet for fast semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (pp. 0–0).