Enhanced Lightweight End-to-End Semantic Segmentation for High-Resolution Remote Sensing Images

HE DONG¹, BAOGUO YU¹, WANQING WU¹ AND CHENGLONG HE.¹, (Member, IEEE)
¹The 54th Research Institute of CETC, Shijiazhuang, China

Corresponding author: Baoguo Yu (e-mail: yubg@sina.cn).

This work was supported in part by the Optoelectronic Fusion Bionic Navigation Universal AI Microcore (SXX21684X056).

ABSTRACT Deep learning based methods have shown promising performance in semantic segmentation of high-resolution remote sensing (HRRS) images. However, due to the multi-scale property and complexity of HRRS images, it still faces many challenges in tackling the scale variance problem and obtaining global context information. In this paper, we propose an enhanced lightweight end-to-end semantic segmentation (ELES2) framework for HRRS images, where a superpixel segmentation pooling (SSP) module is embedded with the framework for result refinement, leading to a more accurate end-to-end semantic segmentation. Besides, compensation connections (CC) are applied between encoder blocks to establish long-range dependencies. In addition, a dense dilated convolutional pyramid (DDCP) module is proposed to generate dense features under different scales and capture global context information. Experiments conducted showed that our ELES2 respectively achieves mIoU values of 80.16% and 73.20% on the ISPRS Potsdam and Vaihingen benchmark datasets using only 12.62M parameters and 13.09G FLOPs. Experimental results prove that our method achieves a promising balance between segmentation accuracy and computational efficiency compared with the state-of-the-art semantic segmentation models.

INDEX TERMS high-resolution remote sensing images, semantic segmentation, superpixel pooling

I. INTRODUCTION

Recently, with the continuous development of satellite technology, high-resolution remote sensing (HRRS) images have been widely used in global and regional scale Earth observation and analysis [1]. Semantic segmentation of HRRS images aims to interpret the images by segmenting the images into semantically meaningful objects and assigning each part with a predetermined tag. It can be widely applied in many applications, such as land use survey, natural disaster detection, environmental monitoring, precise vegetation, and urban planning [2].

With the continuous maturity of deep learning technology, the convolutional neural networks (CNNs) based models show stronger classification capabilities than the shallow models (e.g. support vector machine, SVM [3]) in large-scale datasets due to the use of sparse representation, weight sharing, pooling, and etc [4]. These classical CNNs-based models take the original image patches as the input and output a single category probability vector, which is difficult to directly apply to the task of semantic segmentation for pixel-level classification.

Under this circumstance, a fully convolutional network (FCN) model [7] was proposed for semantic segmentation, which adopted a trainable transposed convolutional layer and the layer-by-layer up-sampling strategy. Since then, semantic segmentation methods based on deep learning have been proposed one after another [8]–[11]. UNet [8] and SegNet [9] both adopt an encoder-decoder structure to recover the location information while retaining high-level semantic features. To further improve the accuracy for semantic segmentation, PSPNet [10] and DeeplabV3+ [11] introduced pyramid pooling module and atrous spatial pyramid pooling module respectively, which aim to obtain multiscale features with pooling operations and dilated convolutions.

Semantic Segmentation becomes challenging owing to multiscale objects, several studies aim to extract multiscale...
features and global context information to enhance the segmentation accuracy. In EncNet [12], a context encoding module is proposed to capture the scene-dependent global context as channelwise attention. PSANet [13] introduces the modeling of long-range correlation for each spatial position. In [14], a multiscale design has been introduced to aggregate context information through different branches. TreeUNet [15] adopts a Tree-CNN block to transmit feature maps via concatenating connections and further fuse multiscale representations. In [16], a multiscale pyramid pooling module is introduced to extract multiscale features for semantic segmentation in HRRS images. Whereas in [17], authors attempt to learn and reason about global relationships, thereby capturing long-range multiscale relation. On the other hand, some studies focus on the refinement of semantic segmentation of HRRS images. In [18], the conditional random field (CRF) is combined with a pretrained network to enhance the segmentation results. Besides, a multitask FusionNet [19] is proposed to achieve semantic segmentation and edge detection, and an edge-aware regularization is applied to refine the segmentation prediction. Though the above-mentioned methods can achieve decent segmentation performance, the large amount of parameters and low prediction speed would severely limit their application scenarios.

Besides, several studies aim to solve the multiscale problems by using superpixels. In [20], Mostajabi et al. obtained 14 sub-images based on superpixels from input images. In [21], Kwak et al. applied superpixels to generate the class score, which is used to predict the classes of objects. Or as in [22], superpixel images are used as the input to enhance the segmentation results by exploring the texture and detailed information in superpixels. Or for post-processing, as in [23], superpixel and CRF are combined as the post-processing method to enhance the segmentation results. However, the post-processing will increase the computational cost and reduce the segmentation efficiency.

To solve these problems, we proposed a novel light-weight end-to-end semantic segmentation framework for HRRS images, which can achieve a promising balance between segmentation accuracy and computational efficiency, where a superpixel segmentation pooling (SSP) layer is introduced in our proposed network structure to improve the computational efficiency and makes our network less sensitive to noise and segmentation scale. Besides, we introduced the compensation connections (CC) to make our network achieve semantic segmentation in a more efficient way. Furthermore, a dense dilated convolutional pyramid (DDCP) module is proposed to deal with the phenomenon of different objects with the same spectrum and further solve the multiscale problem.

II. METHODOLOGY

A. THE PROPOSED ELES2 FRAMEWORK

Our proposed ELES2 framework can be applied to any segmentation network based on encoder-decoder structure. In this work, we choose LinkNet [24] as the backbone of our proposed method due to its efficiency and the lightweight structure. LinkNet bypasses the input of each encoder block into the output of its corresponding decoder block. In this way, it can recover spatial information lost in the downsampling stage, which can be further reused by the decoder. By sharing the knowledge learned from the encoder at each layer, the decoder can use fewer parameters.

The overall flowchart of our proposed ELES2 is given in Fig. 1. Compensation connections are applied between encoder blocks of the backbone network, which aims to use the parameters efficiently. Besides, a DDCP module is introduced in ELES2, which can enhance the segmentation performance. Apart from this, an SSP layer is embedded with the model, such that we can achieve more accurate end-to-end segmentation. The details will be introduced as follows.

B. SUPERPIXEL SEGMENTATION POOLING

Although the use of post-processing steps can benefit segmentation results, it will delay the segmentation process
To achieve end-to-end segmentation, we propose an SSP layer that can use the boundary of the superpixel to correct the output feature map, where superpixel refers to an irregular pixel block with a certain visual significance composed of adjacent pixels with similar texture, color, brightness, and other characteristics. In this paper, we choose the simple linear iterative cluster (SLIC) [25] algorithm for superpixel segmentation due to its efficiency and robustness. Compared with other superpixel segmentation methods, SLIC is compatible with different forms of segmentation maps, and is ideal in terms of running speed, compactness of generating superpixels, and contour preservation.

For an input high-resolution remote sensing image \( M \), we can obtain a superpixel segmentation image by applying the SLIC algorithm, which clusters pixels with similar characteristics in the image into superpixels to constrain the unsampled feature maps, thereby improving the accuracy of semantic segmentation. For the superpixel segmentation image \( S \in \mathbb{R}^{H \times W \times C} \), where \( H \) and \( W \) represent the height and width of the input image respectively and \( C \) is the number of channels, we apply matrix transformation to convert the superpixel segmentation image into a weight matrix \( W \in \mathbb{R}^{H \times W} \). The weight matrix \( W \) is a two-dimensional matrix consisting of 0 to \( L \), where \( L \) represents the number of superpixels. After that, the weight matrix if fed into the SSP module to perform a pooling operation on the unsampled feature map. The weight matrix \( W \) is defined as:

\[
    w_{i,j} = \begin{cases} 
        L, & \text{if } w_{i,j} \in M_L, \\
        0, & \text{otherwise},
    \end{cases}
\]

where \( w_{i,j} \) represents the value of the \( i \)-th row and the \( j \)-th column in \( W \), and \( M_L \) is the \( L \)-th superpixel region in \( M \).

As shown in Fig. 2, the superpixel segmentation map performs pooling on the feature map \( F_{\text{out}} \) obtained from the last transposed convolutional layer in the network. Different from common pooling, which averages or maximizes the areas of a fixed size, the SSP layer averages the value of each superpixel area of the feature map. For the output feature map \( F_{\text{out}} \in \mathbb{R}^{H \times W \times N_{\text{class}}} \) and the superpixel segmentation image \( I \in \mathbb{R}^{H \times W} \), \( I \) is first one-hot encoded into \( I' \in \mathbb{R}^{H \times W \times L} \), where \( N_{\text{class}} \) denotes the number of categories. We define \( F_s \) as the superpixel feature map, and the formulation of the SSP operation can be expressed as follows:

\[
    F_s = \sum_{l=1}^{L} \frac{F_{\text{out}} \cdot I'_{(l)}}{\text{sum}(I'_{(l)})}
\]

where \( I'_{(l)} \) represents the \( l \)-th layer of \( I' \), \( \text{sum}(\cdot) \) stands for sum operation, and \( \cdot \) denotes the dot product between the feature maps.

Compared with other post-processing methods, our SSP layer has the following advantages: (1) Very few parameters need to be set; (2) The boundary of the superpixel is closer to the contour of the actual objects; (3) Can be embedded with the model to achieve end-to-end segmentation. (4) Will not affect the prediction speed of the model.

![FIGURE 2. Schematic diagram of the function of the SSP layer, where denotes the superpixel segmentation pooling on the feature map.](Image)

After applying SSP to the three-dimensional output feature map, we can obtain a feature map of the same size. Then we apply the softmax activation function on the obtained feature map, and get the category of each pixel.

### C. COMPENSATION CONNECTIONS

Though using a deeper and wider structure could improve the performance of multilayer networks, it will also increase the number of the parameters and cause the gradient vanishing problem [26]. In [27], DenseNet is proposed to reduce computational cost and address the degradation problem using feature reuse and dense connections. Motivated by this idea, as shown in Fig. 1, we adopt compensation connections between the encoders blocks of our network, which can improve the efficiency of the parameters. Different from DenseNet, we introduce direct connections from any encoder block to all subsequent encoder blocks. Consequently, the \( l \)-th encoder block can receive the feature maps of all preceding encoder blocks, denoted as follows:

\[
    F_{\ell} = H_{\ell}([F_1, F_2, \cdots, F_{\ell-1}])
\]

where \([\cdot]\) is the concatenation operation of the feature maps \( F_1, F_2, \cdots, F_{\ell-1} \) produced in encoder blocks 1, 2, \cdots, \( \ell-1 \), respectively. \( H_{\ell} \) is defined as the function of the \( \ell \)-th encoder block, as illustrated in Fig. 3. As the backbone network used in our work has 4 encoder blocks, \( \ell \) is set as 4.

The compensation connections enable all encoder blocks to receive a direct supervision signal from the preceding encoder blocks, such that the feature maps between network streams can be better reused. Therefore, we can reduce the computational cost and the number of the parameters for semantic segmentation.
The structure of the DDCP module is given in Fig. 4. In order to enlarge the receptive field of feature points, we stack two dilated convolutional layers with different dilation ratios, horizontally and vertically, thereby increasing the accuracy of semantic segmentation. The architecture of the DDCP module is shown in Fig. 4. For example, for a dilated convolutional layer $L_d$ with a dilation rate $d_L$ and a kernel size $k_L$, the equivalent receptive field size $R_d$ can be calculated by:

$$R_d = (d_L - 1) \times (k_L - 1) + k_L$$ (4)

According to [28], stacking convolutional layers with dense connections can provide a larger receptive field, which brings more overall and higher semantic features. Suppose we stack two dilated convolutional layers with dilation rate $d_1$ and $d_2$ respectively, then the total size of the receptive field $R_{total}$ can be reformulated as follows:

$$R_{total} = R_{d_1} + R_{d_2} - 1$$ (5)

where $R_{d_1}$ and $R_{d_2}$ denote the receptive field sizes of the convolutional layers with dilation rate $d_1$ and $d_2$, respectively.

As shown in Fig. 4, if the feature map $F_4$ output by the last encoder block is $32 \times 32$, the corresponding receptive field size of each layer of DDCP from top to bottom are $3, 7, 15$, and $31$, respectively. Such that it can generate multiscale features, which can not only cover a large scale range, but also cover that scale range densely. In addition, 1 $\times$ 1 convolutional layers can be used to align the size of the feature maps from each layer of the DDCP module.

III. EXPERIMENTS AND RESULTS

A. DATASET AND METRICS

We evaluate our method on two state-of-the-art aerial image semantic segmentation benchmarks, i.e., ISPRS 2D Semantic Labeling Challenge for Vaihingen and Potsdam $^1$, consisting of very high-resolution ture orthophotograph (TOP) tiles and corresponding digital surface models (DSMs). The Potsdam dataset is composed of 38 high-resolution aerial images with $6000 \times 6000$ pixels, where 24 images are chosen for training, and the rest 14 images are used for testing. There are four spectral bands in each image (i.e. red (R), green (G), blue (B), near-infrared (NIR)), and one band in each DSM. The data are organized into six categories: impervious surface, building, low vegetation, tree, car, and clutter/background.

The Vaihingen dataset contains 33 Top tiles and the corresponding DSMs, with an average size of $2494 \times 2064$ pixels. 16 images for training and 17 ones to test our model from the Vaihingen dataset, which has the same six categories corresponding to the Potsdam dataset. Note that we do not use DSM in our experiments. Besides, due to the clutter/background category is very different from the other categories, the clutter/background is not included for comparison according to the ISPRS’s rules.

To better evaluate the segmentation accuracy and computational efficiency of the semantic segmentation algorithms, we adopt three segmentation accuracy evaluation indicators (overall accuracy (OA), $F_1$ score and the mean pixel intersection-over-union (mIoU)) and three computational efficiency evaluation indicators (inference time, parameter amount and floating-point operations (FLOPs)) to comprehensively evaluate the performance of the lightweight semantic segmentation model. Each indicator is introduced in detail below.

Semantic segmentation is essentially a pixel-level classification problem to minimize the error of pixel classification. For a binary classification task, it has multiple calculation metrics, including true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These four indicators represent the situation when different samples are predicted to different categories. Overall accuracy (OA) refers to the probability that all samples are predicted to be the correct classes and can be expressed as:

$$OA = \frac{TP + TN}{TP + FP + TN + FN}$$ (6)

$F_1$ score refers to the harmonic average of precision and recall, which allows allows to evaluate the accuracy of each category by combining the precision and the recall metrics. Based on the confusion matrix, the precision rate represents the probability that the actual class in the predicted positive sample is also a positive sample, and its expression is as follows:

$$\text{precision} = \frac{TP}{TP + FP}$$ (7)

$^1$https://www2.isprs.org/commissions/comm2/wg4/benchmark/
The recall rate represents the probability that the actual positive category is predicted to be positive, and the expression is as follows:

\[ \text{recall} = \frac{TP}{TP + FN} \tag{8} \]

The formula for calculating the \( F_1 \) score is as follows:

\[ F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{9} \]

The mean \( F_1 \) score for each category is expressed as follows:

\[ mF_1 = \frac{1}{k} \sum_{i=0}^{k} F_1 \tag{10} \]

where \( k \) represents the total number of categories and \( mF_1 \) is the mean \( F_1 \) score.

Intersection-over-Union (IoU) is one of the most effective accuracy evaluation indicators for semantic segmentation. It represents the ratio of the intersection and union of set A and set B. Its calculation formula is as follows:

\[ \text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{11} \]

The mean pixel intersection-over-union (mIoU) represents the average IoU over all categories, which is expressed as follows:

\[ 
\text{mIoU} = \frac{1}{k} \sum_{i=0}^{k} \text{IoU}_i
\]

\( m\text{IoU} \) is the mean mIoU over all categories, which is expressed as follows:

\[ m\text{IoU} = \frac{1}{k} \sum_{i=0}^{k} \frac{TP}{TP + FP + FN} \tag{12} \]

where \( TP \) is the number of true positives, \( FP \) is the number of false positives, and \( FN \) is the number of false negatives.

We take the prediction time of the model for a single 512 \( \times \) 512 pixel size image as the inference time.

The amount of parameters is an important indicator to measure the computational efficiency of the algorithms. The amount of parameters can intuitively reflect the space complexity of the semantic segmentation algorithm, and correspond to the consumption of computer memory resources at the hardware level. Taking the convolutional layer as an example, the calculation formula of the parameters of the convolutional layer is as follows:

\[ \text{Parameter} = (k_w \times k_h \times C_{in} + 1) \times C_{out} \tag{13} \]

where \( k_w \) and \( k_h \) represent the number of weights of a convolution kernel, 1 means bias, and \( C_{out} \) is the number of convolution kernels in this layer.

Floating-point operations (FLOPs) refers to the total number of multiplication and addition calculations in a neural network, and is used to measure the time complexity of the algorithm. Taking the convolutional layer as an example, the calculation formula of FLOPs is as follows:

\[ \text{FLOPs} = [(C_{in} \times k_w \times k_h) + (C_{in} \times k_w \times k_h - 1) + 1] \times C_{out} \times w \times h \tag{14} \]
where $k_w$ and $k_h$ represent the width and height dimensions of the convolution kernel, $C_{in} \times k_w \times k_h$ denotes the multiplication calculation amount, $C_{in} \times k_w \times k_h - 1$ represents the addition calculation amount, 1 means bias, $C_{out} \times w \times h$ is all the elements in the feature map, and $w$ and $h$ represent the width of the feature map respectively.

### B. IMPLEMENTATION DETAILS

All the experiments are implemented with Pytorch on two NVIDIA GeForce RTX 2080 Ti GPUs, and the optimizer is set as Adam with a 1e-4 learning rate. The Focal loss [37] function is used as a quantitative evaluation, where the focusing parameter $\gamma$ is set to 0.5. Due to the limitation of computational resources, the input data are randomly cropped into the size of $512 \times 512$ pixels. The batch size is fixed as 8, with 60 epochs for the network to converge. Besides, we perform the following data augmentation strategies for model training: (1) Flipping randomly with 0.5 probability; (2) Cropping randomly to the original size of image with $[0.5, 2]$ scale range; (3) Normalizing all channels by subtracting mean and dividing by the standard variance of the dataset. Whereas in the testing phase, only the normalization strategy is applied.

### C. PERFORMANCE COMPARISON

We compare our method with ten state-of-the-art methods, where BiseNet [29], LiteSeg [30], EMANet [31], LEDNet [32], and BiseNet V2 [33] are five lightweight models, PSP-Net [10] and DeepLabV3+ [11] apply atrous convolutions to enlarge the receptive fields, FCN-CRF [34] uses the post-processing to enhance the segmentation results, and CCNet [35] and DANet [36] are two models based on the attention mechanisms.

For the ISPRS Potsdam dataset, the comparison results are shown in Table 1. It can be seen that our ELES2 has achieved the best performance in several classes (i.e., Imp_suf, Low_veg, and Car) and OA, $mF_1$, $mIoU$ metrics. Although lightweight models such as LEDNet and BiseNet V2 have lower computational costs than ELES2, they also lead to lower segmentation performance. Besides, compared to the other three lightweight models BiseNet, LiteSeg, and EMANet, ELES2 performed better. It can also be seen that while achieving the comparable segmentation accuracy as DeepLabV3+, the parameters and FLOPs of DeepLabV3+ ($56.71M$, $244.22G$) are almost five and twenty times that of ELES2($12.62M$, $13.09G$). Hence, our ELES2 can better balance the relationship between segmentation accuracy and computational cost for more efficient image segmentation.

The comparison results for the ISPRS Vaihingen are shown in Table 2. Our ELES2 achieves top-ranking but not optimal segmentation performance. The main reason is that there are few training samples in the ISPRS Vaihingen dataset, which makes shallow networks easier to fall into local overfitting states. However, ELES2 still maintains a high segmentation accuracy for small samples such as cars. DDCP in ELES2 can generate denser and more continuous feature representation horizontally and vertically. With the DDCP module, ELES2 can exploit both global and local context information of the objects under diverse scales in a dense manner, instead of identifying the objects only by their textures, spectra, and shape information. Besides, SSP makes the network less sensitive to the noise and segmentation scale. Hence, our ELES2 can maintain high segmentation accuracy even in the face of small sample conditions.

The visualization results of the lightweight methods on the ISPRS Potsdam and Vaihingen datasets are shown in Fig. 5 and Fig. 6, respectively. It can be seen that our ELES2 has fewer misclassifications compared to other methods, especially the lightweight models. Moreover, the segmentation boundary of ELES2 is more regular and close to the ground truth.

### D. ABLATION STUDIES

To verify the effectiveness of the modules (i.e. CC, DDCP, SSP) in our proposed framework, ablation studies are conducted for the ISPRS Potsdam dataset, where LinkNet is chosen as the baseline. We applied different module collocations on the baseline network. We can see that all the modules can provide performance improvement as compared to the baseline network, where DDCP and CC would bring extra parameters and SSP slightly increases the inference time. With all the three modules, our ELES2 can further improve the performance and achieve the best semantic segmentation results.

Fig. 7 illustrates the visual performance of our proposed
method on the Vaihingen test dataset. It can be seen that our method shows a splendid segmentation performance with few classification errors. Especially, when SSP is applied to the network, the segmentation result is further corrected. This is because the small number of misclassified pixels will be classified into the correct categories, and the segmentation boundary is closer to the actual boundary of the image. After applying SSP, the segmentation result of our method does not need to be further refined by other post-processing methods.

In addition, we also analyzed the influence of the number of superpixels in SSP on the segmentation results of ELES2, and the experimental results are shown in Fig. 8. It can be seen that with the increasing number of superpixels, the segmentation accuracy of ELES2 increases first and then decreases. When the number of superpixels in SSP is set too small, SSP cannot capture enough boundary information, and will misclassify objects of different categories into one category. On the contrary, when the number of superpixels is set too large, the whole object will be divided into several small pieces and classified separately, which will increase the probability of misclassification. When the number of superpixels is 300, ELES2 achieves the optimal segmentation realization. In our experiments, the number of superpixels for ELES2 in both ISPRS Vaihingen and Potsdam datasets is set to 300.

**TABLE 3.** Ablation Studies on the ISPRS Potsdam dataset.

| Baseline | SSP | DDCP | CC | mIoU | Time | Params |
|----------|-----|------|----|------|------|--------|
| ✓        |     |      |    | 75.13 | 0.21 | 11.53  |
| ✓        | ✓   |      |    | 77.49 | 0.77 | 11.53  |
| ✓        | ✓   |      |    | 76.24 | 0.22 | 12.43  |
| ✓        | ✓   |      | ✓  | 75.97 | 0.23 | 11.67  |
| ✓        | ✓   | ✓    |    | 79.51 | 0.78 | 12.43  |
| ✓        | ✓   | ✓    | ✓  | 78.88 | 0.79 | 11.67  |
| ✓        | ✓   | ✓    | ✓  | 77.31 | 0.26 | 12.62  |
| ✓        | ✓   | ✓    | ✓  | **80.16** | 0.81 | 12.62  |

**FIGURE 6.** Visualization of different methods on the ISPRS Vaihingen test set.

**FIGURE 7.** Visual performance on the ISPRS Vaihingen test dataset of the proposed network. (a) original test images. (a) superpixel segmentation images (c) ground truth. (d) segmentation results by ELES2 without SSP. (e) segmentation results by ELES2.

**FIGURE 8.** The impact of different number of superpixels in SSP on segmentation accuracy on ISPRS Potsdam dataset.
IV. CONCLUSION
In this paper, we presented a ELES2 framework to achieve end-to-end efficient semantic segmentation of HRRS images, where a SSP layer is embedded with the framework to solve the scale variance problem and refine the segmentation results. Besides, CC is introduced to enhance the representation capacity of the network, and DDCP is proposed to generate dense and continuous features under different scales. Experiments conducted on Potsdam and Vaigingen dataset demonstrate the efficiency and superiority of the proposed method. ELES2 can achieve promising segmentation accuracy while maintaining high computational efficiency. ELES2 achieves mIoU of 80.16% and 73.20% on the IPSRS Potsdam and Vaigingen datasets, respectively, with only 12.62M parameters and 13.09 FLOPs. Our study confirms that semantic segmentation algorithms for HRRS images are feasible with both segmentation accuracy and computational efficiency. In the future work, we will focus on the study of lightweight semantic segmentation algorithms under the condition of small samples and unsupervised.

REFERENCES
[1] H. A. Mooney, A. Duraiappah, and A. Larigauderie, “Evolution of natural and social science interactions in global change research programs,” National Academy of Sciences, vol. 110, no. Supplement 1, pp. 3665–3672, 2013.
[2] M. Volpi and V. Ferrari, “Semantic segmentation of urban scenes by learning local class interactions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 1–9.
[3] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” Neural Processing Letters, vol. 9, no. 3, pp. 293–300, 1999.
[4] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning, MIT Press Cambridge, 2016, vol. 1, no. 2.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.
[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
[7] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[8] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, 2015, pp. 234–241.
[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.
[10] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890.
[11] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proceedings of the European Conference on Computer Vision, 2018, pp. 801–817.
[12] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, “Context encoding for semantic segmentation,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 7151–7160.
[13] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, “PspNet: Point-wise spatial attention network for scene parsing,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 267–283.
[14] J. Zhang, S. Lin, L. Ding, and L. Bruzzone, “Multi-scale context aggregation for semantic segmentation of remote sensing images,” Remote Sensing, vol. 12, no. 4, p. 701, 2020.
[15] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, “TreeUNet: Adaptive tree convolutional neural networks for subdecimeter aerial image segmentation,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 156, pp. 8–13, 2019.
[16] B. Yu, L. Yang, and F. Chen, “Semantic segmentation for high spatial resolution remote sensing images based on convolution neural network and pyramid pooling module,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 9, pp. 3252–3261, 2018.
[17] L. Mou, Y. Hu, and X. X. Zhu, “Relation matters: Relational context-aware fully convolutional network for semantic segmentation of high-resolution aerial images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 11, pp. 7557–7569, 2020.
[18] T. Hu, W.-h. Li, and X.-x. Qin, “Semantic segmentation of polar images using conditional random field model based on deep features,” in Journal of Physics: Conference Series, vol. 1168, no. 4. IOP Publishing, 2019, p. 024008.
[19] D. Cheng, G. Meng, S. Xiang, and C. Pan, “ FusionNet: Edge aware dual-convolutional networks for semantic segmentation of remote sensing harbor images,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 12, pp. 5769–5783, 2017.
[20] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, “Feedforward semantic segmentation with zoom-out features,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3376–3385.
[21] S. Kwak, S. Hong, and B. Han, “Weakly supervised semantic segmentation using superpixel pooling network,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017.
[22] Z. Xu, T. Ajanthan, and R. I. Hartley, “Deep learning superpixel semantic segmentation with transparent initialization and sparse encoder,” CoRR, 2020.
[23] W. Zhao, Y. Fu, X. Wei, and H. Wang, “An improved image semantic segmentation method based on superpixels and conditional random fields,” Applied Sciences, vol. 8, no. 5, p. 387, 2018.
[24] A. Chaurasia and E. Culurciello, “LinkNet: Exploiting encoder representations for efficient semantic segmentation,” in Proceedings of the IEEE Visual Communications and Image Processing. IEEE, 2017, pp. 1–4.
[25] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Sutrus, “SLIC superpixels compared to state-of-the-art superpixel methods,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274–2282, 2012.
[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[27] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[28] M. Jiang, Z. Li, K. Yu, C. Zhang, Z. Li, and K. Yang, “DenseASPP for semantic segmentation in street scenes,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3684–3692.
[29] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “BiseNet: Bilateral segmentation network for real-time semantic segmentation,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 325–341.
[30] T. Emara, H. E. Abd El Munim, and H. M. Abbas, “LiteSeg: A novel lightweight convnet for semantic segmentation,” in 2019 Digital Image Computing: Techniques and Applications (DICTA). IEEE, 2019, pp. 1–7.
[31] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, “Expectation-maximization attention networks for semantic segmentation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9167–9176.
[32] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, and L. J. Latecki, “LEDNet: A lightweight encoder-decoder network for real-time semantic segmentation,” in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019, pp. 1860–1864.
[33] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, “BiseNet v2: Bilateral network with guided aggregation for real-time semantic segmentation,” International Journal of Computer Vision, vol. 129, no. 11, pp. 3070–3086, 2021.
[34] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and A. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 135, pp. 158–172, 2018.
[35] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, “CCNet: Criss-cross attention for semantic segmentation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 603–612.

[36] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, “Dual attention network for scene segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3146–3154.

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.