Received October 16, 2020, accepted November 19, 2020, date of publication December 29, 2020,
date of current version January 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3042254

Parallel Fully Convolutional Network for Semantic Segmentation

JIAN JI, XIAOCONG LU, MAI LUO, MINGHUI YIN, QIGUANG MIAO, (Senior Member, IEEE), AND XIANGZENG LIU
School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Corresponding authors: Jian Ji (jji@xidian.edu.cn) and Xiangzeng Liu (xzliu@xidian.edu.cn)

This work was supported in part by the National Key Research and Development Project under Grant B019030051, and the Natural Science Foundation of Shannxi Province under Grant 2020JM-195.

ABSTRACT Fully convolutional networks (FCNs) have been widely applied for dense classification tasks such as semantic segmentation. As a large number of works based on FCNs are proposed, various semantic segmentation models have been improved significantly. However, duplicated upsampling and deconvolution operations in the FCNs will lead to information loss in semantic segmentation tasks and to problems such as ignoring the relationship between pixels and pixels and the lack of spatial consistency. In this study, we propose a parallel fully convolutional neural network that integrates holistically-nested edge detection (HED) network to capture image edge information, improving semantic segmentation performance. We carry out comprehensive experiments and achieve a better result on the PASCAL VOC 2012 , PASCAL-Context and Cityscapes, comparing the results with some existing semantic segmentation methods.

INDEX TERMS Fully convolutional networks, edge detection, edge refinement, semantic segmentation.

I. INTRODUCTION

Semantic segmentation is a fundamental problem in image understanding [1] and in video interpretation [2]–[4]. Recently, several deep learning approaches, especially convolutional neural networks (CNN) [5], VGG [6], Residual Net [7], have achieved great success in recognition tasks. However, these methods have severe limitations in dense prediction tasks such as dense depth, normal estimation [8], and semantic segmentation. Therefore, Image semantic segmentation, as the classification of all pixels of images, is proposed in [9] to tackle the above problems.

Proposed in [9] for the first time, FCN is an innovative visual model that can improve the performance of semantic segmentation tasks. By utilizing the FCN method, many subsequent models based on FCNs have yielded significant improvements in [10]–[18]. However, FCN has larger receptive fields and weak edge constraints which result in low segmentation accuracy. [19] suggests that the insufficient contextual fields of FCN perform poorly in ambiguous regions' predictions, while remedies such as adapting downsample would instead degrade the performance on small-size objects. [11] points out that the excessive pooling layers of FCN would reduce feature map sizes, imposing negative effects on the up-sampling. Therefore, the authors propose DenseASPP that can cover large receptive field sizes to tackle the problem. Recently, Chen et al. [10] came up with DeepLab that adopted atrous convolutions to expand receptive fields without reducing the resolution of the feature map and applied the Dense-CRF post-processing operations to refine the coarse FCN semantic segmentation. This method has been widely applied in semantic segmentation task and achieved state-of-the-art performance. Although DeepLab makes significant progress to a degree, it requires much memory to generate high-resolution feature maps and more computational cost to return results, imposing detrimental effects on computational studies of high-resolution prediction. Therefore, balancing valuable information on image extraction, avoiding the overestimation of computational complexity, and preventing excessive memory consumption are the critical issues needed to be addressed. Some popular approaches generate high-resolution predictions by taking advantage of the features maps produced by shallow and middle layers, such as the FCN method in [9], DenseASPP in [11], and multi-label segmentation in [13]. The purpose of these works is to obtain more detailed information from

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

shallow layers and middle layers which contain more mid-level representations and spatial information than deeper layers.

To address the challenges of low segmentation accuracy in FCN and high computational cost in DeepLab, we introduce a simple, yet efficient parallel fully convolutional network (PFCN) for image semantic segmentation. PFCN consists of two convolutional branches: one for coarse semantic segmentation, the other for edge extraction. Then, domain transform is employed to combine the edge information and the coarse semantic segmentation results, to predict a refined final output. Subsequently, we implement an end-to-end process for coarse-to-fine semantic segmentation. Our contributions are as follows:

Firstly, for the semantic segmentation branch in our proposed method, we employ the FCN-32s model as the base model to generate the coarse semantic segmentation result.

Secondly, a parallel image-to-image edge detection model is embedded to extract more edge information for refining the semantic image. For our work, we employ holistically-nested edge detection (HED) to finish this work, which can automatically learn rich image-level features to solve the problem of edge ambiguity in natural images.

Thirdly, domain transform, a new method for high-quality edge-preserving filtering, is proposed to combine the coarse semantic result with the edge information to achieve a semantic segmentation image with a clear edge. Moreover, through a series of analyses, domain transform can be regarded as a recurrent neural network (RNN) that can be integrated into the model, making it plausible to do end-to-end training.

We evaluate the proposed segmentation method on PASCAL VOC 2012 dataset, PASCAL-Context and Cityscapes. The experiments show that the performance of our proposed method is better than the popular semantic segmentation method. In this work, we also validate the performance of edge extraction by using the HED model on BSDS dataset.

A. RELATED WORK

1) SEMANTIC SEGMENTATION

CNN [20] has been the most successful approach for computer vision tasks, including semantic segmentation [21]–[23], and object detection [24], [25] over the past few years. Early methods for semantic segmentation were based on the regional proposal that classified the regional proposals to produce segmentation results. FCNs [9]–[13] achieve success by adopting fully connected layers and deconvolution and allowing end-to-end training. They are proved effective for feature generation on the task of semantic image segmentation, hence becoming the most prevalent method for semantic segmentation. However, FCNs would lower spatial resolution in the deeper layers and lose fine object boundary details because of receptive fields and multiple pooling layers.

For the limitations given above, many scholars have improved CNN models [5], [26], [27] aiming to generate high-resolution predictions. Hence, many subsequent methods based on CNN are proposed. To overcome this problem, FCN [9] introduces a skip connections architecture to fuse the feature maps produced by the shallow and middle layers. SegNet [14] is more efficient than FCN in memory usage, by replicating the max pool index. However, its benchmark score cannot meet the practical needs. DeepLab [10], [12] directly outputs a medium resolution segmentation by utilizing atrous convolution and then takes advantage of fully connected CRFs to refine boundaries. DenseASPP [11] connects multiple parallel atrous convolutional layers in a dense way, which effectively generates multi-scale features with a larger scale range. CRF-RNN [15]formulates mean-field approximate inference for the dense CRF as RNN, forming an end-to-end system. [20], [28], [29] take advantage of contexts and features of hidden layers to accurately learn specified object edges or details. [30] propose an encoder-decoder network called Stacked Deconvolutional Network (SDN), aiming at obtaining more textual information by deconvolutional networks. Likewise, [12] takes advantage of encoder-decoder modules and Xception to strength their performance. And [31] achieves success in dense image prediction via neural architecture search and network level architecture search space. Although we introduce some approaches that utilize features of middle or shallow layers for semantic segmentation, how to effectively use low-level edge features is still a serious problem needed to be solved.

2) EDGE DETECTION

Some works [32]–[39] have obtained remarkable improvement on the edge detection by applying CNNs models, such as N4-Fields [32], DeepEdge [33], CSCNN [34] and DeepContour [35] in recent years. In our work, we mainly employ CNNs to extract features from the low and middle layers, especially edge information of the original images. While Xie and Tu [36] also exploits the features of edge detection in the middle layers of deep networks just for the sake of edge extraction, which is not applied in the higher-level feature learning. Moreover, Bertasius et al. [37] and Kokkinos [38] achieve better improvement in semantic image segmentation tasks by using the edge of objects.

However, edge detection system and semantic segmentation are regarded as two independent works. They only optimize the performance of edge detection rather than the performance of semantic segmentation tasks. In our study, we learn both object boundaries and optimize rough segmentation results by applying edge information.

We propose a parallel fully convolutional neural network consisting of two network branches. The network architecture is different from some existing FCN-based methods. It consists of two convolutional components that can use low-level edge features to refine coarse semantic feature maps. In particular, the network employs the domain transform structure which allows our whole system to train an end-to-end system. Therefore, our segmentation system can obtain excellent performance.

II. BACKGROUND

Before introducing our approach, we briefly review two related approaches, FCN for semantic segmentation and HED network for edge detection respectively.

A. FCN FOR SEMANTIC SEGMENTATION

Recently, FCN model is the first choice for most semantic segmentation systems. FCN model [9] replaces the fully connect layer of CNN with a 1×1 convolution layer to produce feature maps. To reduce parameters and computational cost, they import multiple pooling layers that lead to outputs of 1/32 resolution of the original images into the network framework. In order to overcome this problem, FCN adds an upsampling path (deconvolution or bilinear interpolation) to recover the resolution to the original size. Therefore, it can be trained end-to-end. However, the final predictions in the FCN model are low-resolution predictions. In order to solve this problem, a skip structure is introduced to refine the final predictions, whose idea is to combine itself with features of different pooling layers by the upsampling operation, to generate a more detailed feature score map. In this way, FCN can produce more accurate and detailed semantic segmentations.

In this work, motivated by the success of the FCN model [9] in semantic segmentation tasks, our semantic segmen- tation branch is based on the FCN structure. This branch is trained to produce low-resolution predictions. After that, we can combine the edge extract branch with domain transform to refine the segmentation results.

B. HED FOR EDGE DETECTION

In order to capture the edge information and refine the low-resolution prediction obtained from FCN model, we embed a parallel edge detection branch to extract edge information. The edge detection branch makes use of HED to exploits edge information from the input image. HED is a VGG network model that combines fully convolutional neural networks with deeply-supervised nets to tackle the problem of image-to-image prediction. The HED network removes the fully connected layer and the fifth pooling layer of the VGG, adding a side output layer to produce edge maps and a hybrid layer to fuse the edge maps to refine the final results. Therefore, the learning process of the HED model is not only multi-scale but also multi-level. For the receptive fields of each group will increase in turn, the resolution of the edge map produced by each side-output layer will become smaller. And the result is produced by mixing different results of the side-output layers. The HED method will automatically study rich hierarchical representations, significant for resolving the ambiguity in object edge detection or details. The result of global representation based on edge prediction is an image-to-image and end-to-end training process.

The HED architecture is shown in Figure 1. There are some differences with VGG Net as follows: (1) the network adds a side output layer at the last convolution layer at stages, conv1_2, conv2_2, conv3_3, conv4_3, and conv5_3 respectively; (2) the fifth pooling layer and the full connectivity layer of VGG Net are removed. There are two main reasons for trimming VGG Net. One is that those layers whose strides are 32 will produce too small feature maps because of expecting useful side output with different scales. Besides, the full connectivity layer is computationally intensive, so the memory/time cost in the training and testing process can be significantly reduced by these trimming. Another reason is that the HED network is a multi-scale and multi-level feature learning process, because of the side output layers. HED network model can be divided into five stages, whose receptive fields are 5, 14, 40, 92, and 196 respectively, all nested in the HED model.

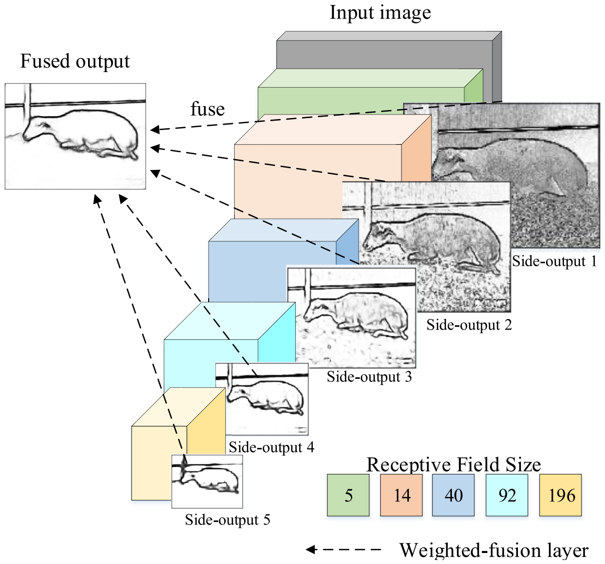


FIGURE 1. Illustration of HED architectures. Multiple side outputs are added after convolutional layers and a fusion layer is added to combine the side output.

The HED network architecture is a multi-stage and multi-level structure which can effectively capture multi-level edge features and the inherent scale of the edge map. This architecture is similar to some previous works, especially the deep supervised network method. It is proved that the hidden layer monitoring can enhance the ability of optimization and generalization in the image semantic segmentation tasks. Moreover, the multiple outputs will give us extra flexibility. Recent works have also shown that fine-tuning of deep neural networks for image semantic segmentation tasks is helpful for the preprocessing of low-level edge detection tasks.

III. PROPOSED MODEL

A. NETWORK ARCHITECTURE

Our proposed method is a novel parallel fully convolutional neural network for semantic segmentation which combines a semantic segmentation network with an edge detection network, achieving a better performance. As shown in Figure 2, the proposed network framework can be divided into three components: (1) a semantic segmentation branch described in part A of Section II, (2) an edge extraction branch described

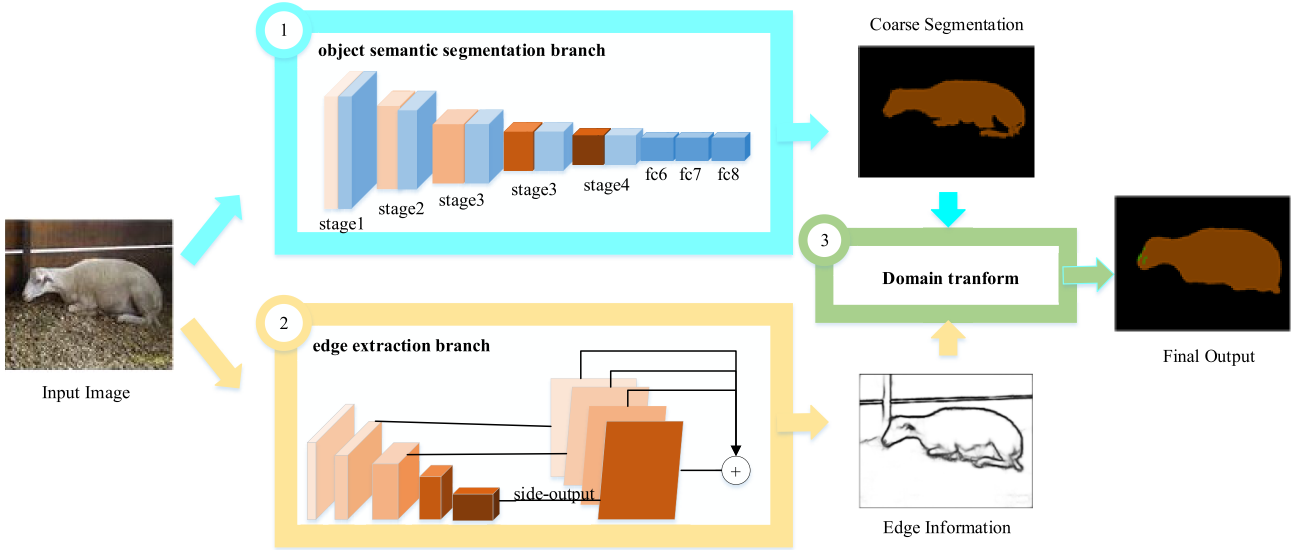


FIGURE 2. Illustration of the proposed model. This model is made up of three parts, including two parallel convolution branches and domain transforms. The two parallel convolutional branches produce coarse semantic segmentation and an edge map, respectively, served as inputs for a domain transform edge-preserving filter for accurate edge recovery in semantic segmentation tasks.

in part B of Section II, and (3) a domain transform structure describes in part B of Section III. The semantic segmentation branch is used to produce coarse semantic segmentation results, while the edge extraction branch is used to capture edge information. These two branches are parallel with each other in our model. In addition, we take advantage of the domain transform structure to refine the semantic segmentation results using the extracted edge information with end-to-end training.

B. DOMAIN TRANSFORM FOR EDGE REFINEMENT

In order to associate these two branches, we employ a domain transform structure to combine the coarse semantic segmentation method with the edge detection in a parallel fashion. Therefore, our model can refine the coarse semantic segmentation guided by the edge information, and also be jointly trained end-to-end.

The domain transform we employ requires two different inputs. One is the original input signal X, which corresponds to the coarse semantic segmentation results in our model. The other is a positive domain transform density signal d, which relates to the edge prediction map. The output of the domain transform is a filtered signal Y, which corresponds to the final refined semantic segmentation. We will introduce the recursive formula of domain transform to explain how the filtered signal Y is obtained.

Let us consider the case of a one-dimensional signal of length X of length N. The output can be computed recursively for $i = 2...N$:

$$y_i = (1 - w_i)x_i + w_i y_{i-1}, \tag{1}$$

where $y_1 = x_1$, and w_i is the weight that lies in the density signal d_i. It is computed as:

$$w_i = exp(-\sqrt{2}d_i/\sigma_s), \tag{2}$$

where σ_s is the standard deviation of filter kernel over the input signal spatial domain. From the Eq. (2), it can be intuitively obtained that the value of the density signal d_i is related to the amount of diffusion/smoothing. This amount controls the contribution of the original input data x_i previous point y_{i-1} when the network computes the filtered output value y_i. The weight w_i acts like a gate, which controls how much information is propagated from pixel $i - 1$ to i. If d_i is very small, the weights are full diffusion, meaning that $w_i = 1$ and $y_i = y_{i-1}$. At the other extreme, if d_i is very large, then $w_i = 0$ and diffusion stops, leading to $y_i = x_i$.

The density signal d is defined as:

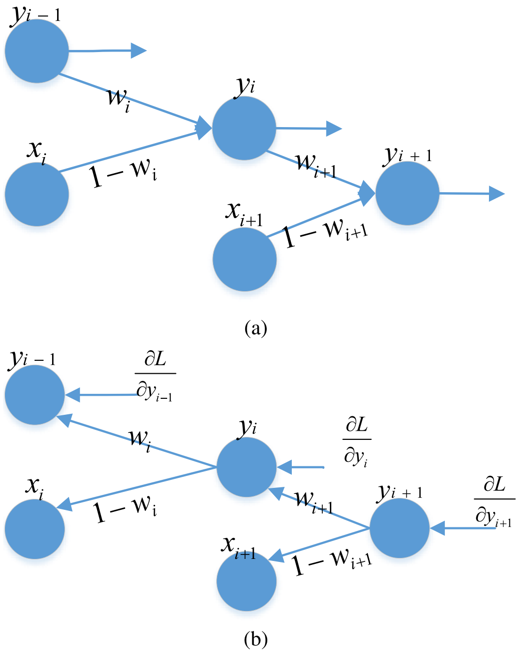
$$d_i = 1 + \vartheta_i \frac{\sigma_s}{\sigma_r}, \tag{3}$$

where ϑ_i corresponds to the edge prediction map and σ_r is the standard deviation of the filter kernel over the edge map's range. Note that the larger the value of ϑ_i is, the more confidence that the pixel x_i gains at the edge.

According to the above formulas, it is known that the current output only depends on the previous outputs. To solve the asymmetry problem, the one-dimensional signal will be filtered two times, one from left to right, and the other from right to left on the previous output passes.

Domain transform works for a 1-D signal, but we apply it in a separable way to transform a 2-D score map generated by our model. In other words, a 1-D filter will be employed sequentially along each signal dimension, performing horizontal transmission along each row (from left to right and right to left), and performing vertical transmission along each column (from top to down and down to top).

In order to transform the domain transform into a trainable filtering, we introduce how the backward propagation process of Eq. (1) of domain transformis, whose computation process of forward propagation is illustrated in Figure 3(a). For each


y_{i-1}
w_i
x_i
$1-w_i$
y_i
w_{i+1}
x_{i+1}
$1-w_{i+1}$
y_{i+1}
$\frac{\partial L}{\partial y_{i-1}}$
$\frac{\partial L}{\partial y_i}$
$\frac{\partial L}{\partial y_{i+1}}$

(a)

(b)

FIGURE 3. Computation process for domain transform recursive filtering: (a) Forward pass that arrows from nodes y_{i+1} denote feeds to subsequent layers. (b) Backward pass, including contributions $\partial L/\partial y_i$ from subsequent layers.

node y_i, it not only directly affects the next node y_{i+1}, but also indirectly affects the subsequent layers through the forward propagation. Thus, the gradient contribution $\partial L/\partial y_i$ of the layer can be obtained from the back propagation. Similar to the process of standard back propagation, the reverse recursion process of Eq. (1) for $i = N, \ldots, 2$ is as shown in Figure 3(b). The back propagation mainly updates the derivatives with respect to y, and also obtain derivatives about x and w:

$$\left(\frac{\partial L}{\partial x_i}\right)' = (1 - w_i)\frac{\partial L}{\partial y_i}, \tag{4}$$

$$\left(\frac{\partial L}{\partial w_i}\right)' = \frac{\partial L}{\partial w_i} + (y_{i-1} - x_i)\frac{\partial L}{\partial y_i}, \tag{5}$$

$$\left(\frac{\partial L}{\partial y_{i-1}}\right)' = \frac{\partial L}{\partial y_{i-1}} + w_i\frac{\partial L}{\partial y_i}, \tag{6}$$

where $\partial L/\partial x_i$ and $\partial L/\partial w_i$ are initialized to 0, and $\partial L/\partial y_{i-1}$ is set to the value obtained from subsequent layers. It is worth noting that the weight w_i is shared in each filtering step (along each row and column) and K iterations.

The forward and backward propagation of the domain transform can be integrated into our model as a trainable layer. In other words, our approach is similar to the recurrent neural network (RNN) that can be added to the whole model. And it can be jointly trained end-to-end with the semantic segmentation branch and edge detection branch. Finally, the coarse semantic segmentation can be refined by edge maps to obtain a refined semantic segmentation result.

IV. EXPERIMENT

In this section, in order to demonstrate the correctness and effectiveness of our approach, we carry out comprehensive experiments on the public dataset and also compare them with some state-of-the-art methods.

A. EXPERIMENT PROTOCOL

1) DATABASES

We train and test our semantic segmentation model on the PASCAL VOC 2012 database. PASCAL VOC 2012 provides a standardized set of excellent datasets for image recognition, classification, and semantic segmentation, containing pixel-wise semantic segmentation images that can classify objects from pixel point to pixel point, involving twenty foreground objects and a background object. The size of the pixel in PASCAL VOC 2012 is different, but the size of the transverse images is about 500 × 375, and the size of the vertical images is about 375 × 500. The deviation will not exceed 100 (in subsequent training, the first step is to resize these images to 500 × 500. Additionally, we evaluate the edge extraction branch HED on a boundary detection dataset, Berkeley Segmentation Dataset and Benchmark 500 (BSDS), which contains 200 training images, 100 validation images, and 200 testing images. Each image has been annotated with a ground real edge.

2) STATE-OF-THE-ART APPROACHES

We compare our proposed approach to several popular state-of-the-art methods, such as DeepLab [10], FCN-8s/32s [9], SegNet [14], PSPNet [22], and U-Net [23].

3) EVALUATION MEASURES

In semantic segmentation tasks, we employ mean IoU (Intersection over Union) to evaluate the performance of our proposed model. We assume that n_{ii} is the number of pixels that should belong to the class i, but are misclassified as the class j. n_{cl} represents available classes and $t_i = \sum_j n_{ij}$ represents the total amount of pixels that belong to the class i. The mean IoU is defined as:

$$\text{mean IoU} = \left(\frac{1}{n_{cl}}\right)\sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}. \tag{7}$$

Besides, the accuracy of edge extraction is assessed via three standard metrics: fixed contour threshold (ODS), per-image best threshold (OIS), and average precision (AP).

4) TRAINING

Our proposed method in training process adopts two-step training. The first step is to train the semantic segmentation branch and the edge detection branch, respectively. Firstly, the setting of parameters is the same as [9] to train when we are in the first step. The parameters are adjusted to getting best performance. Secondly, we train the HED to obtain edge feature maps. The HED is a deep learning model with full use of convolutional neural network and deeply-supervised

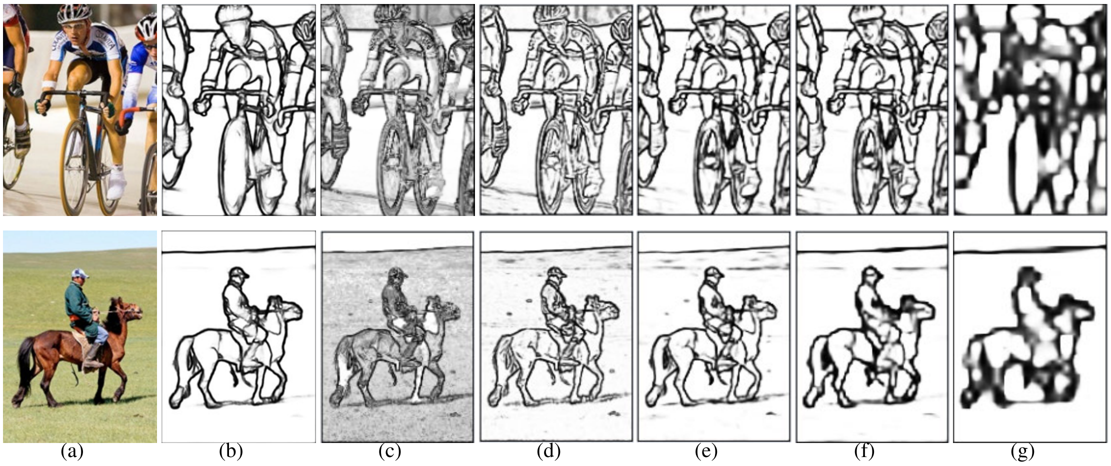


FIGURE 4. Visualizing results on BSDS in each row. We show (a) original image, (b) edge result generated by fusion layer in HED model, (c) (d) (e) (f) (g) are the output of the added five side.

TABLE 1. Results on BSDS Dataset by the HED Model, as well as Some CNN-Based Edge Detection Methods.

POSITION	ODS	OIS	AP
side output 1	0.585	0.621	0.562
side output 2	0.679	0.706	0.658
side output 3	0.724	0.765	0.712
side output 4	0.738	0.763	0.667
side output 5	0.612	0.617	0.419
average 1-4	0.755	0.779	0.789
average 1-5	0.768	0.788	0.800
fusion output	**0.779**	**0.800**	**0.776**

network, which can be trained image-to-image and end-to-end. For the parameters of HED system, we set the mini-batch size 20, set the learning rate 1e-5, set the loss weight 0.2 for each side-output layer, and set the iteration number 10K. The second step is jointly to fine-tune the three components for an end-to-end system, which employs the edge feature maps to refine the coarse semantic segmentation by domain transforms.

B. EXPERIMENTAL RESULT

1) EDGE EXTRACTION RESULTS

We assess the HED model on the BSDS dataset. In order to verify the side outputs from the convolutional layer with the multi-scale feature mapping, we check the results produced by each side outputs in Table 1. In this step, we stress that all the side output predictions can be achieved, resulting in comprehensive analysis in different configurations of combined outputs. We observed a few interesting phenomena in the result: (1) We put in better performance by using multi-scale prediction; (2) All of the side output layers help for improving performance; (3) the side-output layer 1 and layer 5 (the lowest and highest layers in HED model) get a relatively lower performance on edge extraction. We are suspicious that these side outputs make no difference in the average result. But the fact is that, for instance, we can acknowledge the contribution of the side output layer 5 on the average result by the average 1-4 obtaining ODS=0.755 and the average 1-5 obtaining ODS=0.768.

Several visualizing examples are as shown in Figure 4, which illustrates the final edge results and the edge feature maps produced by each side output layer. Note that the higher layers of the side output image are, the less the edge information the output contains. For example, the edge of the side output layer 5 (the highest layer) is blurred. However, the fusion layer's ideal outputs will discard some unnecessary information, such as textures and colors, which can be seen in the side output layer 1. And the outputs also contain rich edge information, compared with the higher side layer, so the fused-outputs preserve the critical for edge refinements' object boundaries.

We carry out an experiment on the HED framework and compare its result with recent CNN-based edge detection models. In this model, we utilize the available side outputs and the fusion layer's outputs to compensate for the loss of average precision by merging all the outputs. This simple operation allows us to achieve better performance. The result are as shown in Table 2, HED achieving a score of 0.827 AP.

2) SEMANTIC SEGMENTATION RESULTS

When training our model, we have attempted different learning rate policies. Different from the fixed learning rate policy and the step learning rate policy, we adopt the "poly" learning rate policy (the learning rate is multiplied by (1 – iter / max_iter)^{power}) to train our model on PASCAL VOC 2012. As shown in Table 3, our experiment is to test the value of batch size and iteration to put in a better performance. We can learn that applying the "poly" learning policy is more effective than "step" or "fixed" learning policy when training our

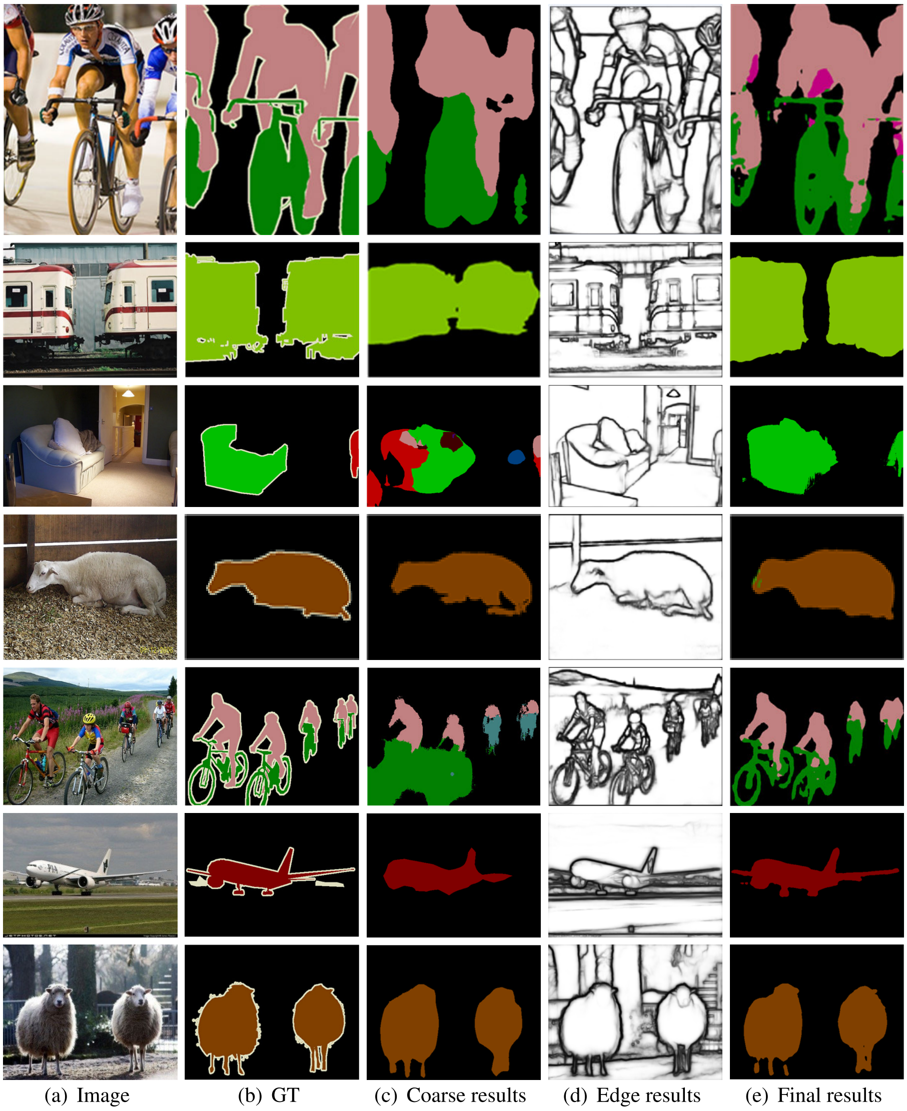


FIGURE 5. The results on the PASCAL VOC 2012 val set.

TABLE 2. Results of Single and Averaged Side Output in HED on the BSDS Dataset. The Single Side Output Contributes to the Fused / Averaged Result and the Fused Result Achieves the Best Performance. The Average1-5 (Averaging of all the Five Layers) Also Produces Better Average Precision.

Methods	ODS	OIS	AP
N4-Fields [32]	0.739	0.766	0.780
Deepedge [33]	0.742	0.762	0.802
DeepContour [35]	0.755	0.763	0.797
HFL [37]	0.752	0.765	0.796
HED [36]	**0.776**	**0.800**	**0.827**

TABLE 3. The Results on PASCAL VOC 2012 val. Set When We Apply Different Learning Hyper Parameters.

Learning policy	Batch size	Iteratorn	mean IoU(%)
fixed	20	10K	60.91
step	20	10K	62.33
poly	20	10K	63.92
poly	20	20K	65.05
poly	10	10K	65.24
poly	10	20K	**66.76**

model. Adjusting the value of the batch size and increasing the iteration number can achieve better performance.

After initializing the parameters, we tested the semantic segmentation performance of our model on the PASCAL VOC 2012 dataset. Table 4 gives the result of our PFCN on the PASCAL VOC 2012 val. set and compares it with some state-of-the-art methods, such as FCN-8s [9], the well-known DeepLab [10], SegNet [14], CRF-RNN [15], PSPNet [22], and U-Net [23]. The inference time of our model on an NVIDIA GTX 1080TI GPU is 241 ms/image, slower than FCN-8s, SegNet, and DeepLab for the reason that our model needs to extract the edge map additionally, but our model outperforms the compared models. We achieve 66.76 mean

TABLE 4. Performance and Inference Time of Some Semantic Segmentation Methods on PASCAL VOC 2012 val. set.

Methods	mean IoU(%)	Infer.Time
FCN-32s [9]	52.42	-
FCN-8s [9]	60.48	156 ms
SegNet [14]	56.96	60 ms
CRF-RNN [15]	62.96	3739 ms
PSPNet [22]	66.34	1250 ms
U-Net [23]	63.17	-
DeepLab(before CRF) [10]	65.25	230 ms
Our method	**66.76**	241 ms

TABLE 5. Segmentation Performance and Comparison on PASCAL-Context.

Methods	mean IoU(%)
DeepLab [40]	37.6
FCN-8s [9]	37.8
CRF-RNN [15]	39.3
ParseNet [21]	40.4
BoxSup [41]	40.5
HO-CRF [42]	41.3
PixelNet [43]	41.4
Piecewise [44]	43.3
Our method	**43.6**

TABLE 6. Segmentation Results and Comparison on Cityscapes.

Methods	IoU$_{class}$(%)	IoU$_{category}$(%)
CRF-RNN [15]	62.5	82.7
FCN-8s [9]	65.3	85.7
SiCNN [45]	66.3	85.0
DPN [46]	66.8	86.0
LRR [47]	69.7	88.2
DeepLab [10]	70.4	86.4
Our method	67.1	**86.5**

IOU score which is better than FCN-8s (about 6.3% improvement), SegNet (about 9.8% improvement), and DeepLab (about 1.5% improvement), interestingly, the accuracy has significantly improved compared to 52.42% of FCN-32s. Meanwhile, it is also much faster and better than PSPNET and CRF-RNN.

Figure 5 gives the semantic segmentation result of our model on the PASCAL VOC 2012. Interestingly, our model puts in a reasonably good performance compared with some recently CNN-based methods. From the coarse semantic segmentation to the refined semantic segmentation, we deliver a noticeable improvement on the boundary of the segmentation, such as the boundary between bicycles and people.

Meantime, we test our method on PASCAL-Context and Cityscapes, as shown in Tables 5 and 6, respectively. The tables illustrate quantitative results of different methods that we achieves mIoU 43.6% on PASCAL-Context, IoU$_{class}$ 67.1% on Cityscapes, and IoU$_{category}$ 86.5% on Cityscapes, better than current methods such as DeepLab [40] and PixelNet [44].

The above experimental results demonstrate that the performance of FCN can be effectively improved by using the edge information of the image. The coarse segmentation results generated by FCN-32s are refined by domain transform under the guidance of edge mapping, as shown in Figure 5. Just like the state-of-the-art works improving their performance by applying different modules, for instance, DeepLabv3+ [12] adopting the decoder structure and multiscale inputs, and Auto-DeepLab [31] adopting multiscale inference and NAS, edge extraction and the domain transform, as detached detachable modules, have improved FCNs to a great extent. And they undoubtly have potentials in other networks.

V. CONCLUSION

In this work, we introduce a parallel fully convolutional network for semantic segmentation. We base our method on FCN and add an additional edge extraction branch to capture the edge information, which is parallel with the semantic segmentation branch. In this way, our approach can generate a more accurate segmentation result by combining coarse semantic segmentation and edge information with domain transform, a traditional edge-preserving filter for graphics applications. Compared to FCN, our method only adds a few additional learnable parameters and obtain satisfactory semantic segmentation accuracy on the PASCAL VOC 2012, PASCAL-Context and Cityscapes.

REFERENCES

[1] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, "Social scene understanding: End-to-End multi-person action localization and collective activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4315–4324.

[2] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6819–6828.

[3] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 221–230.

[4] K. Xu, L. Wen, G. Li, L. Bo, and Q. Huang, "Spatiotemporal CNN for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1379–1388.

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–8.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[8] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[11] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[13] K. Li, W. Tao, X. Liu, and L. Liu, "Iterative image segmentation with feature driven heuristic four-color labeling," *Pattern Recognit.*, vol. 76, pp. 69–79, Apr. 2018.

[14] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.

[16] C. Han, Y. Duan, X. Tao, and J. Lu, "Dense convolutional networks for semantic segmentation," *IEEE Access*, vol. 7, pp. 43369–43382, 2019.

[17] P. Bilinski and V. Prisacariu, "Dense decoder shortcut connections for single-pass semantic segmentation," in *IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 6596–6605.

[18] R. Dong, X. Pan, and F. Li, "Denseu-net-based semantic segmentation of small objects in urban remote sensing images," *IEEE Access*, vol. 7, pp. 65347–65356, 2019.

[19] B. Shuai, T. Liu, and G. Wang, "Improving fully convolution network for semantic segmentation," in *IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1–4.

[20] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 447–456.

[21] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," in *Proc. ICLR*, 2016.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, Oct. 2015, pp. 234–241.

[24] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis.*, Feb. 2017, pp. 2961–2969.

[26] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2018–2025.

[27] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[28] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[29] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7151–7160.

[30] J. Fu, J. Liu, Y. Wang, J. Zhou, C. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *IEEE Trans. Image Process.*, early access, Jan. 25, 2019, doi: 10.1109/TIP.2019.2895460.

[31] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 82–92.

[32] Y. Ganin and V. Lempitsky, "N4-fields: Neural network nearest neighbor fields for image transforms," in *Proc. Asian Conf. Comput. Vis.* Singapore: Springer, Nov. 2014, pp. 536–551.

[33] G. Bertasius, J. Shi, and L. Torresani, "DeepEdge: A multi-scale bifurcated deep network for top-down contour detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4380–4389.

[34] J.-J. Hwang and T.-L. Liu, "Pixel-wise deep learning for contour detection," *Int. Conf. Learn. Represent.*, 2015, pp. 1–8.

[35] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3982–3991.

[36] S. Xie and Z. Tu, "Holistically-nested edge detection," *Int. J. Comput. Vis.*, vol. 125, nos. 1–3, pp. 3–18, Dec. 2017.

[37] G. Bertasius, J. Shi, and L. Torresani, "High-for-Low and Low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 504–512.

[38] I. Kokkinos, "Pushing the boundaries of boundary detection using deep learning," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–12.

[39] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3000–3009.

[40] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[41] J. Dai, K. He, and J. Sun, "BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1635–1643.

[42] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr, "Higher order conditional random fields in deep neural networks," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 524–540.

[43] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan, "Pixelnet: Representation of the pixels, by the pixels, and for the pixels," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017.

[44] G. Lin, C. Shen, A. van den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3194–3203.

[45] I. Krešo, D. Čaušević, J. Krapac, and S. Šegvić, "Convolutional scale invariance for semantic segmentation," in *Proc. Ger. Conf. Pattern Recognit.* Springer, 2016, pp. 64–75.

[46] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1377–1385.

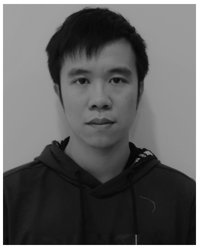
[47] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 519–534.

JIAN JI was born in Xi'an, China, in 1971. She received the B.Sc. degree in computational mathematics from Northwest University, China, in 1993, and the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, China, in 2007.

She is currently a Professor with the School of Computer Science and Technology, Xidian University, China. Her research interests include computational intelligence, pattern recognition, and image analysis.

XIAOCONG LU was born in Guangxi, China, in 1995. He received the B.Sc. degree in computer science and technology from Xiangtan University, China, in 2017. He is currently pursuing the master's degree in computer science and technology from Xidian University, China.

His research interests include semantic segmentation and object detection.



MAI LUO was born in Wuzhou, China, in 1996. He received the B.Sc. degree in materials science and engineering from the Kunming University of Science and Technology, China, in 2018. He is currently pursuing the master's degree in computer science and technology from Xidian University, China.

His research interests include pattern recognition and object detection.

MINGHUI YIN was born in Hebei, China, in 1992. She received the B.Sc. and master's degree in computer science and technology from Xidian University, China, in 2015 and 2018, respectively.

Her research interests include pattern recognition and image semantic segmentation.

QIGUANG MIAO (Senior Member, IEEE) was born in Shandong, China, in 1972. He received the M.Eng. and Ph.D. degrees in computer science from Xidian University, Xi'an, China.

He is currently a Professor with the School of Computer Science and Technology, Xidian University. His research interests include intelligent image processing and multiscale geometric representations for images.

XIANGZENG LIU received the M.S. and Ph.D. degrees in applied mathematics from Northwestern Polytechnical University, Xi'an, China, in 2008 and 2011, respectively.

In 2012, he joined as an Advanced Image Processing Engineer with the Xi'an Microelectronics Technology Institute. He is currently an Associate Professor with the School of Computer Science and Technology, Xidian University, China. His current research interests include image registration, image enhancement, object detection and recognition, computer vision, and machine learning.