ParallelNet: A Depth-Guided Parallel Convolutional Network for Scene Segmentation

Shiyu Liu and Haofeng Zhang(✉)

School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China
zhanghf@njust.edu.cn

Abstract. In the past few years, deep convolutional neural networks (CNN) have shown great superiority and also been the first choice in semantic segmentation. However, the pooling layers in the CNN cause the increasing loss (mainly positioning structure details) which is not favourable for segmentation. Moreover, the vast majority of previous studies only utilize the color or textural information of the image, without considering the depth information which is helpful for segmentation. In this paper, we propose a novel and effective end-to-end network for semantic segmentation namely Depth-guided Parallel Convolutional Network (ParallelNet). Compared to previous work, the contribution of our ParallelNet is that we have taken advantages of the mutual benefit and strong correlations between depth information and semantic information, which are combined to guide scene semantic segmentation. Besides, we utilise a new method to obtain the depth information of the image by calculating the correlation distance with $\mathcal{L}_1$-norm between left and right feature maps, thus, we just need to input the RGB images instead of RGB images and encoded 3D images in some conventional methods. Furthermore, we apply the concept of our ParallelNet to the current popular networks by exploiting the guidance of the depth information and transfer their learned representations with fine-tuning. The extensive experiments on the popular dataset Cityscape exhibit that our ParallelNet outperforms the original methods.

1 Introduction

Recently, semantic pixel-wise segmentation has been one of active research topics, since it has a very wide range of applications, including autonomous driving, three-dimensional reconstruction etc. Image semantic segmentation can be regarded as the cornerstone of image understanding technology, which plays a significant role in autonomous systems. Early approaches [24,28] which mostly dependent on low-level vision cues of image pixels have quickly been substituted by popular machine learning algorithms [21,23,26]. Especially, when deep convolution neural networks (CNN) were applied to object classification [15,25,27] with a great success, more and more researchers start to exploit CNN features to

X. Geng and B.-H. Kang (Eds.): PRICAI 2018, LNAI 11012, pp. 588–603, 2018.
https://doi.org/10.1007/978-3-319-97304-3_45

solve other structured prediction problems [10,14]. To the end, a series of CNN based semantic segmentation methods have been proposed, and the accuracy of image semantic segmentation is repeatedly refreshed.

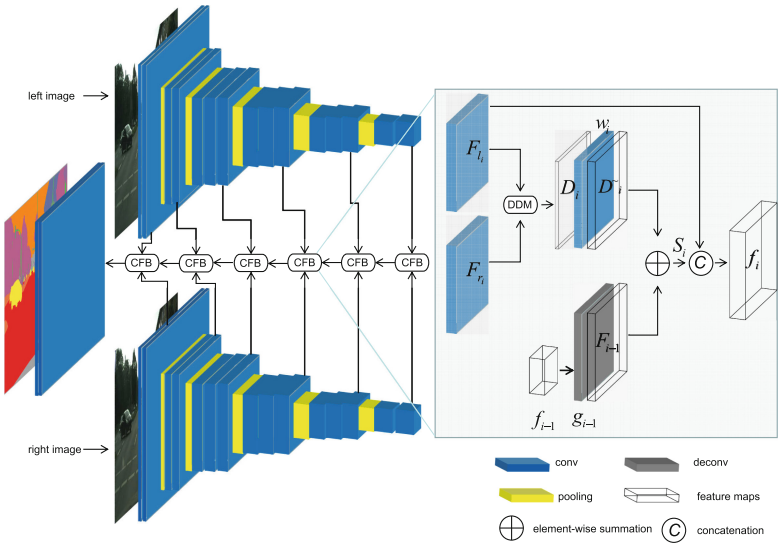

Fig. 1. An illustration of the framework of our proposed network, which contains two-parallel VGG branches to extract RGB features. The block namely CFB is exploited to calculate depth information, which is then combined with the RGB features to obtain the final results.

Most semantic segmentation methods based on CNN come from a common ancestor: Fully Convolutional Networks (FCN) [19]. FCN extended the well-known classification networks to dense pixel-wise labelling through convolution-alization. Its results, though very inspiring, is rough. This is mainly because the pooling layers reduce the spatial resolution of the feature maps, which results in the loss of positioning structure detail information. The increasingly lossy image feature representation is unfavourable for segmentation in which structural information is vital. In order to solve this limitation, various approaches have been proposed. One [1,22] is the encoder-decoder architecture, another [3,5,6] exploits dilated convolution. In addition, some studies [2,4,31] apply conditional random fields as post-processing to obtain detailed prediction.

These above methods are all committed to capturing and storing boundary information. And these networks do have superior performance in delineating boundaries. However, the effect is less than satisfactory when the networks distinguish the certain indeterminate areas or the categories with geometric distinction. This is primarily because most of the current semantic networks only extract color or textual features of images, which only contains 2D information, while some 3D geometric information may be lost in RGB-only features and

there will still be uncertainty for recognizing objects. Therefore, incorporating 3D scene information into 2D information is helpful for scene semantic segmentation as the 3D scene information can provides additional structural information to compensate the lossy structural representation in 2D images.

Depth as a type of 3D scene information is important in realistic scenarios. Depth information and semantic information have strong correlations and mutually beneficial: objects or pixels nearby with same depth have great opportunity to have the same semantic meanings. Besides, the Depth information provides rich and relatively accurate position information, which plays as an auxiliary guiding role in semantic segmentation.

Recently, some approaches have exploited multi-model feature fusion for semantic segmentation [9,12,29], most of these methods use RGB-D images as inputs, and apply two CNN branches to extract RGB and Depth features respectively, and then simply fuse the features from two branches. However, the learned feature representations using raw depth images with CNN is not rich. In order to effectively exploit the pre-trained network with fine-tuning to learn stronger features, the depthmap is encoded with three channels: horizontal disparity, height above ground, and angle with gravity (HHA [11]), which are computed from the original disparity map, and the disparity map should be calculated in advance, while the stereo images pairs are relatively more accessible.

Besides, inspired by Binocular Stereo Vision [13,17,20], which is based on the parallax principle that exploits two images obtained from different views and obtains the three-dimensional geometric information of the object by calculating the positional deviation between corresponding points of the image, we present a novel method to obtain the depth features of the image by calculating the correlation distance with $\mathcal{L}_1$-norm between image feature pairs, thus, we just need to input the RGB images instead of RGB images and HHA in those conventional methods.

In order to continue to exploit the learned rich representations of CNN pre-trained on RGB images with fine-tuning directly, as well as take the benefit from depth to segmentation, we design a Depth-guided Parallel Convolutional Network (ParallelNet) to incorporate depth features calculated from the RGB image pairs in the network into RGB features to improve segmentation accuracy. The framework of our ParallelNet is illustrated in Fig. 1. We utilize the Depth information of the image, which is combined with semantic information to guide scene semantic segmentation. Our network contains two-parallel VGG branches for extracting RGB features of the left and right image respectively. These two VGG branches share weights. Then several cascaded depth and RGB features fusion blocks (CFB) are exploited to obtain the final results. The CFB is crucial to our network. It consists of three vital blocks: the depth determination module (DDM), element-wise summation and concatenation. The DDM is essential for calculating the depth feature information. This block is inspired by the Binocular Stereo Vision with two inputs. The inputs are RGB features of the left and right image of a certain level. Then, perform the element-wise summation on the output of the DDM and the previously refined feature computed from the last

CFB, followed by concatenation which connects the summed results with the current level of the RGB feature. The CFB adaptively trains the RGB features of the image pair to effectively fuse the complementary features in depth and RGB modalities, while combing the high-level and low-level feature to finer the results. In this architecture, discriminative RGB and depth features in different level can be availably trained and fused, while retaining the advantage of skip architecture. Since the depth information is calculated inside the network, we can train the network end-to-end. It should be noted that, although we input image pairs, the left network branch is the main branch and the right branch is just an auxiliary branch to help obtain the depth information, the segmentation ground-truth is the left image in the supervised training process. We apply the concept of our ParallelNet to the current popular networks, and the extensive experiments on the popular dataset Cityscape [7] show that our parallel network can improve the performance over the original methods.

To sum up, the contribution of this paper mainly has the following three points:

(1) We propose a novel ParallelNet with RGB image pairs as input that exploits the advantage of the mutual benefit and strong correlations between depth and semantic information, which are combined to guide scene semantic segmentation.
(2) Inspired by Binocular Stereo Vision, we present an innovative module namely DDM which enables efficiently obtaining the depth information from the RGB images inputs instead of RGB images and HHA inputs in some previous methods.
(3) The experiments on the popular dataset Cityscape show that our method can improve the performance over the convolutional methods especially on these categories such as fences, Pole which have clear depth distinction.

2 Related Work

Since a great success in object classification employing deep CNN [15,25,27], a majority of studies on semantic segmentation have exploited deep CNN. Fully Convolutional Networks (FCN) [19] is the common ancestor of most current semantic segmentation methods. The advantage of FCN is that it employs the existing CNN as powerful visual models to learn hierarchies of features. FCN extended the well-known classification networks (AlexNet [15], GoogleNet [27], and the VGG [25]) into fully convolutional networks by replacing the last fully connected layers with convolutional layers, and produced feature maps instead of classification scores.

Although the results achieved by FCN is very encouraging, there are still some drawbacks. The first limitation is the low resolution of the feature maps due to the max pooling and sub-sampling, which leads to the results coarse. In order to solve this limitation, some approaches have been proposed. One [1,22] is an

encoder-decoder architecture. Another [3,5,6] exploits astrous convolution, also named dilated convolution, which enlarges the receptive field in a exponential expansion way with no loss of resolution. Moreover, some researches [2,4,31] combine conditional random fields (CRF) into deep CNNs as post-processing to improve the segmentation accuracy.

Another limitation is the ensemble of multi-scale feature. FCN [19] exploited a skip architecture to combine what and where to obtain the finer prediction. To fully utilize global and local image-level features, Liu et al. [18] certified that global average pooling with FCN is efficient. Lin et al. [16] presented RefineNet that modified higher-level features by exploiting lower-level features via residual connections and achieved great increase. PSPNet [30] utilized the ability of global context information by integrating the contexts of different regions to generate good quality segmentation results.

Recently, some approaches utilizing depth information for segmentation have been studied. They extended the RGB based Convolutional networks to RGB-D situation. Early fusion method [8] was just concatenating depth into RGB channels as four-channel input. Later fusion method [19] added the two predictions computed by the two modalities. The architecture proposed by Wang et al. [29] is a network for deconvolution of multiple modalities. However, its training process included two stages, it can't be a end-to-end network. Moreover, in order to exploit the pre-trained network with fine-tuning to extract richer features, the depthmap should be encoded to a 3D image called HHA. In contrast, our proposed end-to-end architecture exploits the learned rich representations of CNN pre-trained on RGB images with fine-tuning directly and, as well as takes the benefit from depth to segmentation with the RGB images input.

3 Methodology

Our ParrelleNet benefits from the strong correlation and complementarity of depth and semantic information. We apply the concept of ParrelleNet to current popular networks FCN [19], Deeplab [5], PSPNet [30]. We mainly take FCN as basic network as an example to introduce our ParallelNet in detail. The other two variants are also introduced.

Our proposed ParallelNet's framework based on FCN is showed in Fig. 1. Our network contains two-parallel VGG branches to extract RGB features of different levels from bottom to up on the left and the right image respectively. The two VGG branches share weights. Following that, we employ several cascaded depth and RGB features fusion blocks (CFB) to get the final prediction with the skip architecture. CFB is the key to our network. The details of these modules will be elaborated in this section.

3.1 The Depth Determination Module (DDM)

An illustration of the depth determination module (DDM) is shown in Fig. 2. Given RGB features of the left and right images at certain level from the CNN,

first we fix the left RGB feature maps, then we use a novel $right - shift - n$, $s.t. 0 < n < m$ operation to represent that the right feature maps are moved parallel n pixels to the right to match the corresponding points in the left feature maps. m is the depth level we set in advance. Then do the Correlation Distance Calculation (CDC) by $\mathcal{L}_1$-norm between the left and the new right feature maps obtained after $right-shift-n$, which generates a depth feature map. It is worth noting that the process of obtaining the first depth feature map don't perform the $right - shift - n$, but perform the CDC directly on the left and right RGB feature maps. Repeat the above processes for m times to finally receive m depth feature maps. Last, concatenate all the depth feature maps to get the depth information.

Specifically, let $h \times w \times g$ represents the spatial size of the given RGB feature maps and let $\mathcal{F}_l$, $\mathcal{F}_r$ denote the left and right RGB feature maps respectively. $\mathcal{F}_l$, $\mathcal{F}_r$ are both $h \times w$ matrix, $\boldsymbol{l}_{(x,y)}$ is a g-dimensional vector of the (x, y) position of the left RGB feature maps. Every element in the $\boldsymbol{l}_{(x,y)}$ is the feature value $l_{(x,y)_i}$ of the i^{th} $s.t. 1 \leq i \leq g$ RGB feature map at (x, y) position. So does $\boldsymbol{r}_{(x,y)}$. In this case,

$$\boldsymbol{l}_{(x,y)} = [l_{(x,y)_1}, l_{(x,y)_2}, \ldots, l_{(x,y)_i}, \ldots, l_{(x,y)_g}] \tag{1}$$

$$\mathcal{F}_l = \begin{pmatrix} \boldsymbol{l}_{(1,1)} & \cdots & \boldsymbol{l}_{(1,w-n)} & \boldsymbol{l}_{(1,w-n+1)} & \cdots & \boldsymbol{l}_{(1,w)} \\ \vdots & \boldsymbol{l}_{(x,y)} & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{l}_{(h,1)} & \cdots & \boldsymbol{l}_{(h,w-n)} & \boldsymbol{l}_{(h,w-n+1)} & \cdots & \boldsymbol{l}_{(h,w)} \end{pmatrix} \tag{2}$$

Next, we utilize the above formulas to introduce two important parts of our DDM:

(1) $right - shift - n$: We do this $right - shift - n$ on the right RGB feature maps, keeping the left RGB feature maps unchanged. Connect the last n columns of the original matrix $\mathcal{F}_r$ to the left of the remainder. We let ${\mathcal{F}^{\sim}}_{r_n}$ denote the new right RGB feature maps after $right - shift - n$.

$${\mathcal{F}^{\sim}}_{r_n} = \begin{pmatrix} \boldsymbol{r}_{(1,w-n+1)} & \cdots & \boldsymbol{r}_{(1,w)} & \boldsymbol{r}_{(1,1)} & \cdots & \boldsymbol{r}_{(1,w-n)} \\ \vdots & \vdots & \vdots & \vdots & \boldsymbol{r}_{(x,y)} & \vdots \\ \boldsymbol{r}_{(h,w-n+1)} & \cdots & \boldsymbol{r}_{(h,w)} & \boldsymbol{r}_{(h,1)} & \cdots & \boldsymbol{r}_{(h,w-n)} \end{pmatrix} \tag{3}$$

(2) Correlation Distance Calculation (CDC): We do the CDC on the $\mathcal{F}_l$ and ${\mathcal{F}^{\sim}}_{r_n}$ by $\mathcal{L}_1$-norm. Assuming that the $\boldsymbol{l}_{(x_1,y_1)}$ and the $\boldsymbol{r}_{(x_1,y_1)}$ are the vectors of the two feature maps in (x_1, y_1) position, their correlation distance can be calculated as follows:

$$d_{(x_1,y_1)} = \|\boldsymbol{l}_{(x_1,y_1)} - \boldsymbol{r}_{(x_1,y_1)}\|_1 = \sum_{i=1}^{g} |l_{(x_1,y_1)_i} - r_{(x_1,y_1)_i}| \tag{4}$$

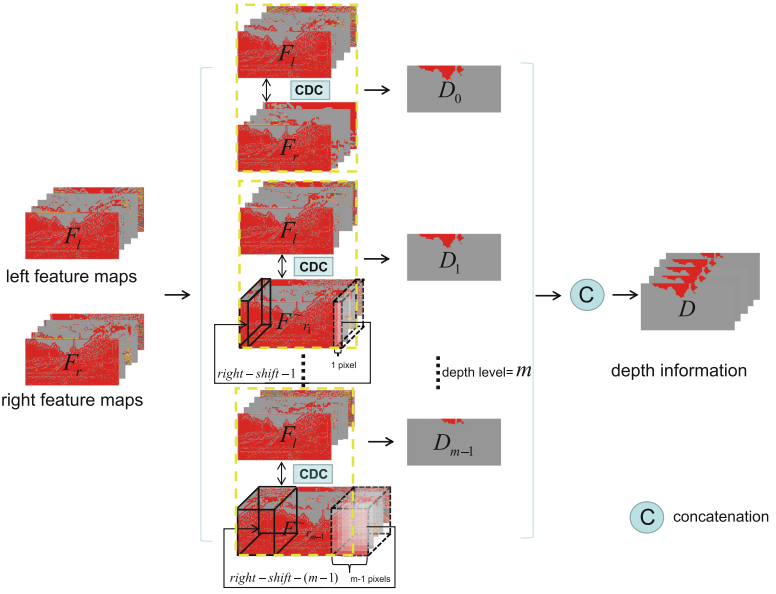

Fig. 2. A detailed illustration of the depth determination module. For the left and right feature maps extracted from the CNN, do the $right - shift - n$ operation, followed by the Correlation Distance Calculation (CDC) to obtain one feature map. Repeat the two operations for m times. Finally concatenate all the feature maps to get the depth features.

Based on this, do $\mathcal{L}_1$-norm on every corresponding position vector between $\mathcal{F}_l$ and $\mathcal{F}^{\sim}{}_{r_n}$, we can get the n^{th} depth feature map D_n as follows:

$$D_n = \begin{pmatrix} \|\boldsymbol{l}_{(1,1)} - \boldsymbol{r}_{(1,w-n+1)}\|_1 & \cdots & \|\boldsymbol{l}_{(1,w)} - \boldsymbol{r}_{(1,w-n)}\|_1 \\ \vdots & \vdots & \vdots \\ \|\boldsymbol{l}_{(h,1)} - \boldsymbol{r}_{(h,w-n+1)}\|_1 & \cdots & \|\boldsymbol{l}_{(h,w)} - \boldsymbol{r}_{(h,w-n)}\|_1 \end{pmatrix} \tag{5}$$

3.2 Cascaded Depth and RGB Features Fusion Block (CFB)

Our proposed efficient cascaded depth and RGB features fusion block (CFB) is able to fuse the two complementary modalities features and combine coarse higher-level features with fine lower-level to generate higher-resolution semantic feature maps with skip architecture.

As shown in Fig. 1, the i^{th} CFB has three inputs (except the first one): the refined feature maps f_{i-1} obtained from the previous CFB, the left and right RGB feature maps $\mathcal{F}_{l_i}$, $\mathcal{F}_{r_i}$. $\mathcal{F}_{l_i}$ and $\mathcal{F}_{r_i}$ are fed into the DDM to get the primary depth feature maps $\mathcal{D}_i$, then the $\mathcal{D}_i$ is passed through a 3×3 convolution layer

ω_i with the number of channels equals to the channels of $\mathcal{F}_{l_i}$ (Assuming equal to c), therefore, we obtain the new depth feature maps $\mathcal{D}^{\sim}{}_i$ with c channels. For the remaining input f_{i-1}, we get feature maps $\mathcal{F}_{i-1}$ of the same resolution as $\mathcal{D}^{\sim}{}_i$ by feeding the f_{i-1} into a deconvolution layer of stride 2 with c channels. Following that we perform element-wise summation on $\mathcal{F}_{i-1}$ and $\mathcal{D}^{\sim}{}_i$, the results are denoted as S_i. Later we concatenate $\mathcal{F}_{l_i}$ into S_i as f_i. The output of CFB block is f_i. The entire process can be expressed as follows:

$$f_i = T\{\mathcal{F}_{l_i}, \omega_i * D(\mathcal{F}_{l_i}, \mathcal{F}_{r_i}) + g_{i-1} * f_{i-1}\} \tag{6}$$

where the first $*$ represents convolution, the second $*$ denotes deconvolution. The $+$ represents element-wise summation. And $D(\cdot, \cdot)$ indicates the DDM operation, $T(\cdot, \cdot)$ indicates concatenation.

Last, the f_i are passed through two consecutive convolution layers, followed by a 1×1 convolutional layer with channel dimension 19 to predict the scores for each class at each location of the final high-resolution feature map.

In CFB, $\mathcal{D}_i$ obtained from DDM firstly passed through one 3×3 convolutional layer, which non-linearly transform the primary depth feature $\mathcal{D}_i$ to obtain rich and effective depth features. The output of previous CFB f_{i-1} provides the high-level information which includes both depth information and semantic information. For the purpose of getting finer prediction f_i, the left RGB features $\mathcal{F}_{l_i}$, the right RGB features $\mathcal{F}_{r_i}$, and the previous features f_{i-1} are embedded with ω_i, $D(\cdot)$ and g_{i-1}. We can also think that the $\mathcal{F}_{l_i}$ and $\mathcal{F}_{r_i}$ are used to provide residual low-level information including depth and RGB information between f_{i-1} and f_i. It is worth noting that f_i is the key value of CFB. The specific reasons are as follows: First, f_i not only fuses depth and RGB features at current layer, but also combines with the features at the previous layer. Second, the f_{i-1} provides coarse high-level feature maps from the deeper layers. Last, the supervision training on f_i can enable the network to continuously utilize the complementarity and mutual benefit of the two modalities to learn to transform and fuse the depth and RGB features by learning the parameters ω_i, g_{i-1} to achieve better performance than before.

3.3 Other Network Variants

ParallelNet-Deeplab Network is shown in Fig. 3. As we know, the conv1, conv2, conv3 in Deeplab have reduced resolution of image. So we feed the output of the last block of conv1, conv2 and ASPP to CFB with skip architecture to get final prediction. PSPNet and Deeplab are the same architecture except the combination method of the Pyramid Pooling Module output. Deeplab is sum-fusion, while PSPNet is concatenate-fusion. Based on this, the structure of ParallelNet-PSPNet is the same as ParallelNet-Deeplab.

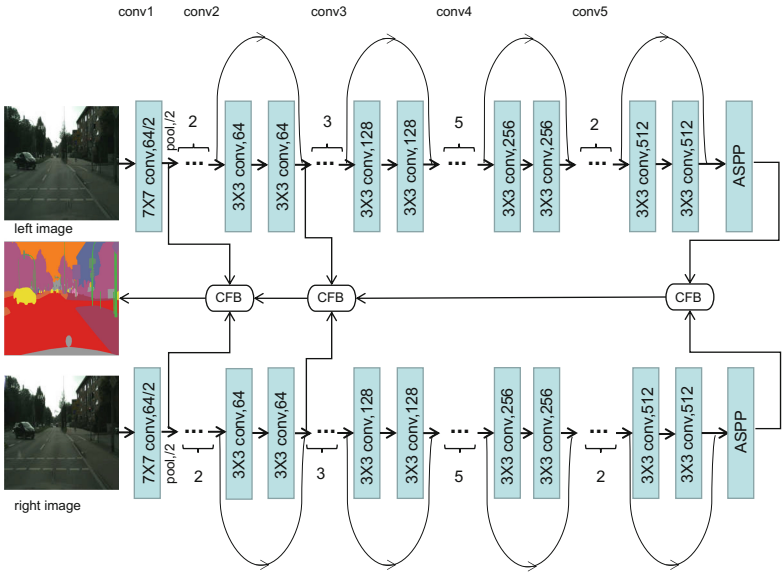

Fig. 3. The illustration of ParallelNet-Deeplab. We feed the output of the last module of the conv1, conv2 and ASPP to three CFBs with skip architecture to get the final results.

4 Experiments

Dataset. In this section, we evaluate our approach through a series of experiments on Cityscape Dataset for semantic segmentation. Cityscape Dataset is a large-scale dataset for autopilot-related aspects, focusing on pixels-wise scene semantic segmentation and instance annotation. The data scene includes different scenes from 50 different cities (mainly in Germany), with high quality pixel-level annotations of 5000 frames in addition to a larger set of 20000 weakly annotated frames. What's more, the dataset provides corresponding right images, which meets the requirements of the input image pairs required by our network. We use the left and right 8-bit images with 5000 frames (pixel-level) for our experiments. The image data is divided into 34 categories which contain both stuff and objects. And the 5000-frame fine-labelled (pixel-level) data are partitioned training, verification and test set. There are 2975 images for training, 500 images for verification and 1525 images for testing. It is noted that there are only 19 classes included in our experiments assessment.

Evaluation Metrics. We mainly employed three widely used metrics to evaluate our experimental results: the pixel-wise accuracy (Pixel Acc), the mean of class-wise intersection over union (Mean IoU) and the instance-level intersection-over-union (iIoU).

Implementation Details. Our experiments are implemented on the public platform Tensorflow. We apply the concept of ParallelNet to FCN, Deepalb and

PSPNet. We exploit the "exponential decay" learning rate method so that a better solution can be quickly obtained and the model can be more stable later in the training process. We set the $learning-rate$, $decay-steps$, $decay-rate$ to 0.1, 1000, 0.96 respectively. We train our network by Adam optimizer. Specially, for the three networks, the weights in the bottom-up RGB feature extraction (convolutional network) are initialized by employing the pre-trained net, while the weights in CFB are initialized with Xavier initialization, and zero-initializes the bias. Then we fine-tune all layers with back-propagation. Moreover, dropout is performed on each network to prevent overfitting. The input image pairs are randomly cropped during the training. And we set *batchsize* to 4 due to our limited memory of GPU. During the testing, we cropped five patches (the four corner and the center patches) followed by averaging the predictions to make the final results.

4.1 Comprehensive Experiments

We compare our ParallelNet applied on FCN, Deeplab, PSPNet namely ParallelNet-FCN, ParallelNet-Deeplab, ParallelNet-PSPNet with the original networks. The results are shown in Table 1. It can be clearly seen that our ParallelNet outperforms the corresponding original network. For our ParallelNet-FCN, the results of Pixel acc, mIoU and iIoU are 93.63%, 63.68% and 43.82% respectively with our settings (depth level is 128, Num of CFBs is 5), and it improves the accuracy of FCN by 0.68%, 1.77% and 0.67% for Pixel acc, mIoU and iIoU respectively. For our ParallelNet-Deeplab, it increases the results of Deeplab by 0.79%, 1.16% and 0.55%. And for our ParallelNet-PSPNet, it enhance the accuracy of PSPNet by 0.82%, 1.22%, 0.56%. The results indicates combining depth with RGB features can help achieve better semantic segmentation.

Table 1. Comparison of our ParallelNet with original network. Ours outperforms the original network.

Method	Pixel Acc. (%)	mIoU (%)	iIoU (%)
FCN	92.95	61.91	43.15
PaallelNet-FCN	93.63	63.68	43.82
Deeplab	86.03	39.16	32.37
ParallelNet-Deeplab	86.82	40.32	32.92
PSPNet	88.65	44.74	35.11
ParallelNet-PSPNet	89.47	45.96	35.67

Class-wise accuracies of ParallelNet compared with the corresponding original networks are illustrated in Tables 2, 3 and 4. As it can be seen, the results of our ParallelNet have been improved in most categories by incorporating the

Table 2. Comparison of Class-wise semantic segmentation accuracy between FCN and ParallelNet-FCN (the bold fonts in the tables indicate the superiority of our results).

Method	Road(%)	swalk(%)	build(%)	wall(%)	fence(%)	pole(%)	tlight(%)
FCN	95.5	75.11	86.89	31.10	42.55	49.45	56.20
PallelNet-FCN	**96.36**	75.44	**89.46**	**34.79**	**47.64**	**55.17**	56.39
Method	sign(%)	Veg.(%)	terrain(%)	sky(%)	person(%)	rider(%)	car(%)
FCN	65.47	88.71	56.20	89.09	72.81	44.22	89.88
PallelNet-FCN	**67.37**	**90.5**	55.83	88.97	**76.96**	**47.92**	**90.4**
Method	trunk(%)	bus(%)	train(%)	mbike(%)	bike(%)	mIoU(%)	iIoU(%)
FCN	31.86	48.96	43.83	38.19	70.19	61.91	43.15
PallelNet-FCN	32.39	49.23	**44.52**	**39.83**	**70.83**	**63.68**	**43.82**

depth into RGB features especially in those categories with clear depth and geometric distinction, for example, wall, pole, fences, person, while in some classes which have little geometric distinction such as sky, terrain, our methods have shown no superiority.

Table 3. Comparison of Class-wise semantic segmentation accuracy (IoU) between Deeplab and ParallelNet-Deeplab (the bold fonts in the tables indicate the superiority of our results).

Method	Road(%)	swalk(%)	build(%)	wall(%)	fence(%)	pole(%)	tlight(%)
Deeplab	90.69	64.65	68.43	17.11	19.52	30.20	1.15
ParallelNet-Deeplab	**91.62**	65.21	**70.49**	**20.13**	**23.46**	**33.98**	1.36
Method	sign(%)	Veg.(%)	terrain(%)	sky(%)	person(%)	rider(%)	car(%)
DEeplab	32.35	82.42	45.40	44.39	51.56	7.72	80.64
ParallelNetDeeplab	**33.70**	**83.55**	45.60	44.17	**53.67**	**8.78**	**81.34**
Method	trunk(%)	bus(%)	train(%)	mbike(%)	bike(%)	mIoU(%)	iIoU(%)
Deeplab	5.27	24.33	17.05	12.45	48.61	39.16	32.37
ParallelNet-DEeplab	**6.09**	**25.03**	17.07	12.96	47.78	**40.32**	**32.93**

4.2 Ablation Studies

We conduct ablative experiments for ParallelNet-FCN by setting different depth levels and cascade numbers of CFB. The results are shown in Tables 5 and 6 respectively.

Table 5 shows the effect of different depth levels on the semantic segmentation. It is noted that we set the number of CFB to 5. We find that the pixel accuracy and the mIoU first increase and then decrease as the depth levels increase. We think that when the set depth level is relatively small, that is to say, for a certain pixel on the left image, the corresponding target pixel on the right image may not be matched. With the set depth level increases, the most pixels in the left can be matched correctly with the corresponding target pixels on the right images. In this case, the extracted depth information can play a positive role in guiding semantic segmentation. However, when the depth level increases continually, there may be additional pixels which matches the certain pixel in the

Table 4. Comparison of Class-wise semantic segmentation accuracy(IoU) between PSPNet and ParallelNet-PSPNet (the bold fonts in the tables indicate the superiority of our results).

Method	Road(%)	swalk(%)	build(%)	wall(%)	fence(%)	pole(%)	tlight(%)
PSPNet	92.56	67.84	77.22	21.32	25.02	38.47	3.00
PallelNet-PSPNet	**93.89**	**68.19**	**78.11**	**24.09**	**28.95**	**43.10**	2.90
Method	sign(%)	Veg.(%)	terrain(%)	sky(%)	person(%)	rider(%)	car(%)
PSPNet	27.16	85.30	50.09	75.43	58.30	11.62	83.40
PallelNet-PSPNet	**29.87**	**86.46**	48.55	75.71	**60.40**	**12.10**	84.00
Method	trunk(%)	bus(%)	train(%)	mbike(%)	bike(%)	mIoU(%)	iIoU(%)
PSPNet	22.45	38.35	7.48	15.20	49.92	44.74	35.11
PallelNet-PSPNet	22.55	37.69	**8.25**	**16.35**	**50.99**	**45.96**	**35.66**

left. So the number of disturbing pixels in the right image may increase and the errors correspondingly increase. At this time, the depth information will have a negative effect on semantic segmentation due to the existence of the errors, which causes both pixel accuracy and mIoU to decrease. We find that the depth level is set to 128 for the best performance.

Table 5. The results of different depth levels on semantic segmentation. And setting the depth level to 128 will achieve the best performance

Method	Pixel Acc. (%)	mIoU (%)	iIoU (%)
32	91.35	56.67	41.19
64	92.64	59.43	41.81
128	93.63	63.68	43.82
192	92.47	61.08	42.07

Table 6 shows the effect of diverse numbers of CFB on semantic segmentation. We have set the depth level to 128. From the bottom to the top of the network, we record the CFB at the top of the network as the first CFB and the CFB at the bottom of the network as the sixth CFB. From Table 6, we find that multiple CFBs which utilize the skip structure has improved performance, the pixel accuracy and the mIoU grow fast as the number of CFB increases from 2 to 5. The network using 5 CFBs achieve 63.97% mean IU and 93.63% pixel accuracy. And our cascaded improvements have met diminishing returns both with respect to the IU metric and also in terms of pixel-wise accuracy when the number of CFB increases from 5 to 6.

4.3 Qualitative Results

We show some qualitative results of ours proposed method compared with FCN network which only employ the RGB information on semantic segmentation in

Table 6. Refining ParallelNet by cascading different numbers of CFB improves semantic segmentation.

Num of CFB	Pixel Acc. (%)	mIoU (%)	iIoU (%)
6	93.79	63.59	43.88
5	93.63	63.68	43.82
4	92.34	60.57	42.07
3	91.87	58.41	42.38
2	91.3	55.88	40.69

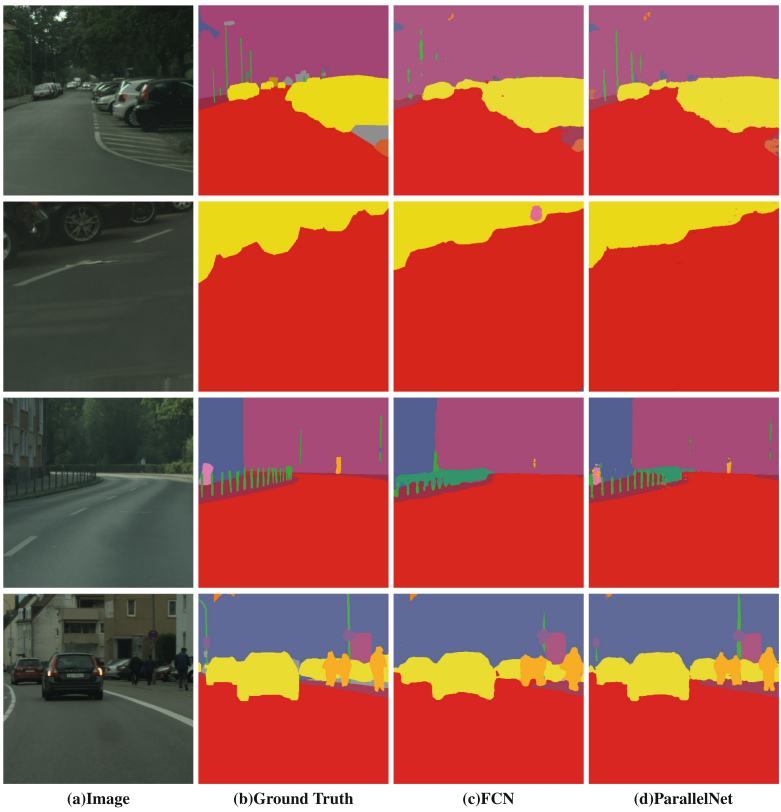

Fig. 4. Qualitative results of our ParallelNet-FCN compared with FCN. From the left to right for each example:image, Ground truth, the resluts of FCN and our ParallelNet. Note that our network shows significant improvement in these categories which have clear depth distinction, e.g., (a) the Pole which has clear depth distinction. (b) eliminating noise points with the help of depth to segmentation, (c) the fence which has geometric distinction, (d) the pedestrian which has obvious depth characteristic. Best viewed in color.

Fig. 3. We obtain the semantic segmentation results of FCN by running the available source code. We compare the results with our ParallelNet which employs image pairs as inputs and combines the depth information with the RGB information for segmentation (Fig. 4). We can see that our network shows significant improvement in fences, Pole, pedestrians categories which have clear depth distinction that may be lost in RGB-only features and our network helps eliminating noise points.

5 Conclusion

We present a novel ParallelNet for effectively segmenting images by taking benefit of the relevance and complementarity between depth and RGB modalities on the RGB images inputs. Our effective CFB with skip architecture can availably fuse the discriminative RGB and depth features in different level and combine the higher-level and lower-level features to get finer prediction. Our experiments demonstrate that our proposed ParallelNet outperforms the original network which only utilize the RGB features. In the future we plan to extend our proposed method to object detection and classification tasks to obtain more competitive results.

Acknowledgements. This work was supported in part by the Key Research and Development Plan of Jiangsu Province (BE2015162) and the Major Special Project of Core Electronic Devices, High-end Generic Chips and Basic Software (2015ZX01041101).

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for scene segmentation. PAMI **PP**(99), 2481–2495 (2017)
2. Chandra, S., Kokkinos, I.: Fast, exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs (2014). arXiv preprint arXiv:1412.7062
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. Comput. Sci. **4**, 357–361 (2014)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs (2016). arXiv preprint arXiv:1606.00915
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
7. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding (2016)

8. Couprie, C., Farabet, C., Najman, L., Lecun, Y.: Indoor semantic segmentation using depth information. Eprint Arxiv (2013)
9. Deng, Z., Todorovic, S., Jan Latecki, L.: Semantic segmentation of RGBD images with mutex constraints. In: ICPR, pp. 1733–1741 (2015)
10. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE T-PAMI **35**(8), 1915–1929 (2013)
11. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from RGB-D images for object detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 345–360. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_23
12. Hazirbas, C., Ma, L., Domokos, C., Cremers, D.: FuseNet: incorporating depth into semantic segmentation via fusion-based CNN architecture. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (eds.) ACCV 2016. LNCS, vol. 10111, pp. 213–228. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54181-5_14
13. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR, vol. 2, pp. 807–814. IEEE (2005)
14. Khan, S.H., Bennamoun, M., Sohel, F., Togneri, R.: Geometry driven semantic labeling of indoor scenes. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 679–694. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_44
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
16. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2017)
17. Liu, S., Zhao, L., Li, J.: The Applications and Summary of Three Dimensional Reconstruction Based on Stereo Vision (2012)
18. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: looking wider to see better. In: ICLR Workshop (2016)
19. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR, pp. 3431–3440 (2015)
20. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: CVPR, pp. 5695–5703 (2016)
21. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. PAMI **36**(11), 2227–2240 (2014)
22. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
23. Sánchez, A.V.D.: Advanced support vector machines and kernel methods. Neurocomputing **55**(1–2), 5–20 (2003)
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22**(8), 888–905 (2000)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
26. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**(3), 199–222 (2004)
27. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR, pp. 1–9 (2015)
28. Tang, M., Gorelick, L., Veksler, O., Boykov, Y.: Grabcut in one cut. In: ICCV, pp. 1769–1776. IEEE (2013)

29. Wang, J., Wang, Z., Tao, D., See, S., Wang, G.: Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 664–679. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_40
30. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR, pp. 2881–2890 (2017)
31. Zheng, S., et al.: Conditional random fields as recurrent neural networks, pp. 1529–1537 (2015)