Received 6 July 2023, accepted 14 August 2023, date of publication 23 August 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307771

RESEARCH ARTICLE

Unified Network With Detail Guidance for Panoptic Segmentation

QINGWEI SUN[1,2], JIANGANG CHAO[2,3], WANHONG LIN[2,3], ZHENYING XU[2,3], AND WEI CHEN[2,3]

[1]Department of Aerospace Science and Technology, Space Engineering University, Beijing 101416, China
[2]China Astronaut Research and Training Center, Beijing 100094, China
[3]National Key Laboratory of Human Factors Engineering, China Astronaut Research and Training Center, Beijing 100094, China

Corresponding author: Jiangang Chao (acc_cjg@163.com)

This work was supported in part by the Work Enhancement Based on Visual Scene Perception, and in part by the National Key Laboratory Foundation of Human Factors Engineering under Grant SYFD062003.

ABSTRACT Panoptic segmentation has won popularity in image perception for its unique advantages. A generic backbone is utilized to extract image features, either fusing semantic and instance segmentation results or end-to-end. Backbone is able to merge low-level details and high-level semantics. However, in practice, detailed information is weakened after deep convolutions. To address this limitation, we propose a novel unified network consisting of a bilateral feature extraction structure and an aggregation module. Both detail and semantic information extraction are decoupled successfully. Specifically, the bilateral feature extraction structure comprises two main branches. One branch uses a generic backbone to obtain the rich receptive field, while the other uses the guidance of detail ground-truth to extract low-level features. Furthermore, the aggregation module combines the results of two branches to obtain a large receptive field with detailed information. Comparative experiments are performed on COCO and Cityscapes datasets. The results demonstrate that high accuracy is obtained. Among them, 41.3 panoptic quality is achieved on COCO, and 59.9 is achieved on Cityscapes.

INDEX TERMS Panoptic segmentation, unified network, scene perception.

I. INTRODUCTION

Panoptic segmentation is a typical method proposed in [1], in which every pixel obtains a category and an independent instance number. Such a method wins its popularity for unique advantages over semantic segmentation and instance segmentation. Compared to semantic segmentation, panoptic segmentation assigns different instance numbers to each foreground category belonging to *things* (objects with fixed shapes and countable, such as people, tables, cars, etc.). Compared to instance segmentation, it classifies each background region belonging to *stuff* (regions without fixed shape and uncountable, such as grass, beach, woods, etc.). Furthermore, panoptic segmentation comprises box-based and box-free methods according to whether the box is proposed.

The typical box-based method consists of three steps in serial: detection, segmentation, and fusion. Firstly, boxes of *things* are predicted by the box branch. Secondly, the category is obtained for objects in each box. In addition, a separate semantic sub-branch is designed for predicting the pixel category of *stuff*. Overlap not only between different instances but also between instance and semantic segmentation. Finally, the overlap is resolved to obtain precise results [2], [3], [4], [5]. Such methods [6], [7] are generally performed on two-stage instance segmentation networks, such as Mask-RCNN [8]. Meanwhile, another lightweight semantic segmentation head is designed to predict *stuff*. There are also methods [9], [10], [11] based on single-stage object detection networks [12], [13], [14]. And these methods reduce the complexity of the network and speed up the inference.

In contrast, the box-free method [15], [16], [17] generates no box. Therefore, the impact of the box is eliminated,

The associate editor coordinating the review of this manuscript and approving it for publication was Essam A. Rashed.

resulting in direct panoptic segmentation on a larger feature map. Unlike box-based methods, box-free methods use a segmentation-detection step to achieve the goal. Generally, semantic segmentation networks [18], [19] are used to obtain instances of *things* in a clustering-like manner, including Hough-voting [20], Watershed transform [21], [22], or instance center regression [23], [24]. In addition, there is another box-free method that takes *things* and *stuff* as a whole, named the unified network. Panoptic FCN [25] is a classical unified network that encodes each instance into a specific kernel and directly generates the prediction by convolutions. Furthermore, the complex postprocessing is removed, making the network lighter and more accurate.

Either method requires encoding features by a common backbone [26] and decoding features by different branches. Although such a backbone has merged high-level and low-level information, it is still dominated by high-level features and weakens low-level information after deep convolutions. In image segmentation, low-level information is crucial to predict the detail output [27].

To extract low-level information, we propose a novel network based on Panoptic FCN. In parallel with the backbone, we add the *Detail Branch* to capture details with wide channels and shallow layers. And the *Aggregation Module* integrates low-level with high-level information to encode the image better. Moreover, Laplacian convolutions extract the contour of *things* and *stuff* as the detail ground-truth, which optimizes the network to focus on the details. In practice, only eight convolutional layers are used in the *Detail Branch*. Channels rapidly grow from 3 to 256. The *Aggregation Module* fuses two branches without adding too much computation. In particular, Laplacian convolutions are implemented in the training phase, which does not increase the inference's consumption. Experiments are conducted on COCO [28] and Cityscapes [29] datasets. Our network achieves the best panoptic segmentation results, reaching 41.3 panoptic quality on the COCO validation set and 59.9 panoptic quality on the Cityscapes validation set.

In summary, our main contributions lie in the following aspects:

1) We propose a novel unified panoptic segmentation network consisting of a bilateral feature extraction structure and an aggregation module.

2) Both detail and semantic information extraction are decoupled successfully.

3) Our model achieves better results on COCO and Cityscapes datasets without significantly increasing time-consuming.

II. MODEL DESCRIPTION

We propose a novel unified network with detail guidance for panoptic segmentation built on Panoptic FCN. Generally, image segmentation relies on large receptive fields and low-level detail information. In particular, our model adds the *Detail Branch* in parallel with the backbone to encoder image information. Specifically, wide channels and shallow layers are used to extract detailed features. To better optimize the network, we use detail ground-truth to train the *Detail Branch*. Furthermore, the *Aggregation Module* integrates high-level and low-level features, which is conducive to improving the performance of panoptic segmentation. FIGURE 1 shows the top-level structure of the network, and this chapter will introduce each module in detail.

A. MODULES OF PANOPTIC FCN

Our network is based on Panoptic FCN [25], a classic unified panoptic segmentation network. Specifically, Panoptic FCN is a fully convolutional network that uses FPN [30] as the backbone. The *Kernel Generator* aims at generating the kernel weight map with positions for *things* and *stuff*, including position head and kernel head. Unlike other panoptic segmentation models that require non-maximal suppression to eliminate the overlap between instances, Panoptic FCN achieves this goal by fusing kernel weights generated from different stages of the FPN. The Semantic Encoder fuses feature maps from different stages of the FPN to obtain multilevel features, similar to [6]. In particular, CoordConv [31] is used in the *Kernel Generator* and *Feature Encoder* to encode the coordinates where *things* are located.

The ResNet50-based FPN is used in our model to encode high-level features. Such a backbone can extract rich high-level information by stacking many convolutional layers, which have been verified in various networks. Both high-level and low-level features are crucial for segmentation tasks. However, backbones like ResNet [26] and FPN are designed for classification. Although a sizeable perceptive field can be obtained, detailed information is sacrificed. To further improve the segmentation, our model adds the *Detail Branch* in parallel with FPN to extract high-level and detailed information through a bilateral structure.

B. MODULES OF OUR NETWORK

1) DETAIL BRANCH

Similar to [32], our network uses wide channels and shallow layers for details, and the structure is shown in TABLE 1. Conv2d denotes the convolution module, including a convolution layer, a batch normalization layer [33], and a ReLu activation function. Usually, networks with rich channels can encode richer detail information, so we utilize eight convolution modules to grow the channels from 3 to 256. Indeed, the output channel is the same as the *Semantic Encoder*, which facilitates the subsequent information fusion. Specifically, this branch includes fewer layers and uses convolution with stride=2 for downsampling. Meanwhile, residual connections are not used because the network contains fewer layers and does not create degradation. In addition, adding residual structure leads to more parameters and increases the time-consuming.

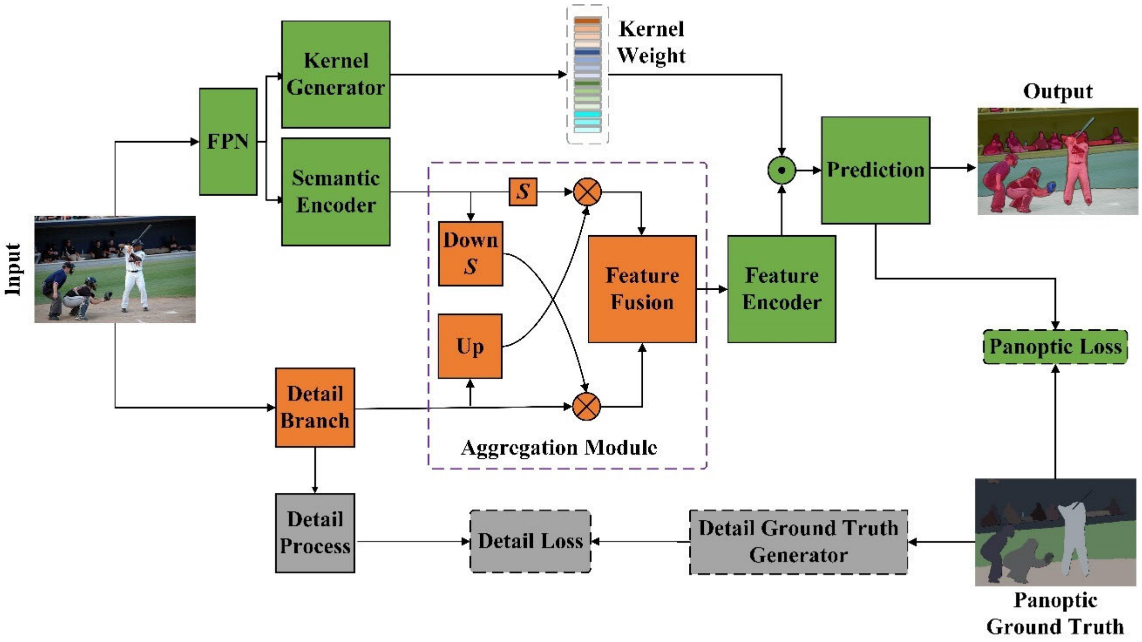


FIGURE 1. Our network is based on Panoptic FCN, where the green modules are the same as Panoptic FCN. The orange modules, including the Detail Branch and the Aggregation Module, are new to the network. The gray modules are training processes guided by the detail ground-truth, discarded in the inference. In the figure, S is the Sigmoid function, Down represents downsampling, Up denotes upsampling, ⊗ means element-wise multiplication, and ⊙ represents convolution operations.

TABLE 1. Structure of the *Detail Branch*. *S* means stage.

Stage	Type	Kernel size	Output channels	Stride
Input				
S_1	Conv2d	3	64	2
	Conv2d	3	64	1
S_2	Conv2d	3	64	2
	Conv2d	3	64	1
	Conv2d	3	64	1
S_3	Conv2d	3	128	2
	Conv2d	3	256	1
	Conv2d	3	256	1

2) AGGREGATION MODULE

Semantic Encoder and *Detail Branch* extract high-level and low-level information, respectively. We must fuse the two kinds of information to get accurate panoptic segmentation results. The downsampling rates of the two branches are different, so we propose the *Aggregation Module* to fuse different information. As described in [32], there are several ways to merge information, but the bidirectional method is more effective. The *Aggregation Module* is shown in FIGURE 2, and the resolution of the *Detail Branch* is 1/2 of the *Semantic Encoder*.

Furthermore, each branch is divided into two subbranches, and the subbranches of the corresponding resolution are multiplied and then added. In this way, the shape of the output tensor is the same as the input of the *Semantic Encoder*. Then, each right branch is connected to the Sigmoid activation function and multiplied with the left branch. It is equivalent to adding a weight to each pixel of the left branch, guiding the *Detail Branch* to obtain information on different scales. Moreover, we upsample the left branch and add the right branch pixel element-wise. Finally, the convolution with a 3 × 3 kernel is used to process the output.

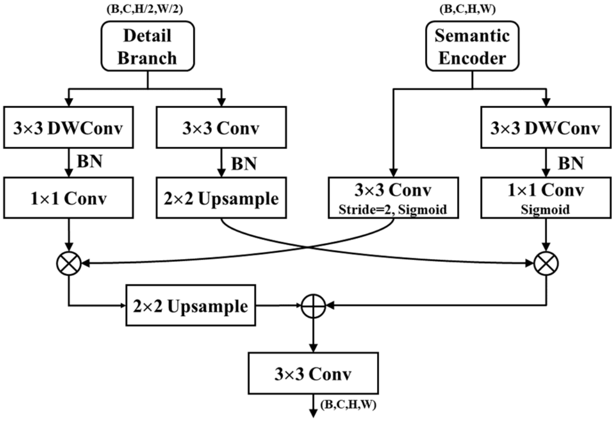


FIGURE 2. Detailed design of the Aggregation Module. DWConv means the depth-wise convolution, Conv donates the convolution, BN represents the batch normalization, Upsample indicates the bilinear interpolation, and Sigmoid is the Sigmoid activation function. Meanwhile, 1 × 1 and 3 × 3 denote the kernel size, (B, C, H, W) means the tensor shape (batch, channel, height, width), ⊗ indicates element-wise multiplication, and ⊕ represents element-wise addition.

3) DETAIL GROUND-TRUTH GENERATOR

To explicitly guide the model to learn detail features, we use 2D convolution with Laplacian kernels to obtain binary detail ground-truth. To obtain details, convolutions with different strides are used to extract detail ground-truth at different res-

TABLE 2. Structure of the detail process.

Stage	Type	Kernel size	Output channels	Stride
Input		-	256	-
Detail Process	Conv2d	3	64	1
	BN	-	64	-
	ReLu	-	64	-
	Conv2d	1	1	1

olutions and then uniformed by upsampling [34]. Our method is different from it. We use 3-dimensional, 5-dimensional, and 7-dimensional Laplacian kernels to generate detail ground-truth with the same resolution, respectively, and achieve better results.

As shown in FIGURE 3, (a) is the method used in [34]. Among these, a 3-dimensional Laplacian kernel is used with strides 1, 2, and 4. The detail ground-truth is obtained with different resolutions by a threshold. (b) is the method used in our model. We get the detail ground-truth with the same resolution by different dimensional Laplacian kernels. Both methods use a convolution of 1 × 1 to fuse features. Our Laplacian kernels are given as follows:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$
$$\begin{bmatrix} -4 & -1 & 0 & -1 & -4 \\ -1 & 2 & 3 & 2 & -1 \\ 0 & 3 & 4 & 3 & 0 \\ -1 & 2 & 3 & 2 & -1 \\ -4 & -1 & 0 & -1 & -4 \end{bmatrix}$$
$$\begin{bmatrix} -10 & -5 & -2 & -1 & -2 & -5 & -10 \\ -5 & 0 & 3 & 4 & 3 & 0 & -5 \\ -2 & 3 & 6 & 7 & 6 & 3 & -2 \\ -1 & 4 & 7 & 8 & 7 & 4 & -1 \\ -2 & 3 & 6 & 7 & 6 & 3 & -2 \\ -5 & 0 & 3 & 4 & 3 & 0 & -5 \\ -10 & -5 & -2 & -1 & -2 & -5 & -10 \end{bmatrix} \tag{1}$$

4) DETAIL PROCESS

The detail ground-truth are binary images. So, the *Detail Branch* feature maps need to be processed for network training. As shown in TABLE 2, the channels of the *Detail Process* are quickly reduced to 1 by a two-layer convolution with the same resolution.

C. LOSS FUNCTION

1) LOSS OF KERNEL GENERATOR

The *Kernel Generator* is mainly used to localize centers of *things* and regions of *stuff*. Our network is optimized using the same loss function, Focal Loss [13], as Panoptic FCN, which is given as follows:

$$\begin{aligned} L_{\text{pos}}^{\text{th}} &= \sum \text{FL}(\text{f}, \text{g})/N \\ L_{\text{pos}}^{\text{st}} &= \sum \text{FL}(\text{f}, \text{g})/WH \\ L_{\text{pos}} &= L_{\text{pos}}^{\text{th}} + L_{\text{pos}}^{\text{st}} \end{aligned} \tag{2}$$

where $L_{\text{pos}}^{\text{th}}$ is responsible for optimizing centers of *things* and $L_{\text{pos}}^{\text{st}}$ is for regions of *stuff*. In addition, FL means Focal Loss. (f, g) denotes the feature map of the *Kernel Generator* and the ground-truth, respectively. N is the number of *things*, and W and H are the width and height of the feature map. And the detailed implementation can be referred to [25].

2) PANOPTIC LOSS

During training, the localization of *things* and *stuff* is mainly optimized by L_{pos}, so Panoptic Loss is primarily used to optimize the segmentation. Following [25], we use the weighted Dice Loss [35], which is formulated as follows:

$$L_{\text{seg}} = w\text{Dice}(\text{p}, \text{g})/(M + N) \tag{3}$$

where w means kernel weight, p is the prediction, and g denotes the ground-truth. M and N represent the number of categories of *stuff* and *things*, respectively.

3) DETAIL LOSS

The detail feature map is a binary map. Detail pixels are much less than non-detail pixels, so detail training is a class-imbalance problem [25]. Following [34] and [36], we use cross-entropy and dice loss for coarse and further optimization, respectively, which is formulated as:

$$L_{\text{detail}} = L_{dice}(\text{d}, g) + L_{ce}(\text{d}, g) \tag{4}$$

L_{dice} denotes dice loss, L_{ce} means cross-entropy loss, d is the Detail Process output, and g represents the detail ground-truth.

Our total loss function is:

$$L = \lambda_{\text{pos}} L_{\text{pos}} + \lambda_{\text{seg}} L_{\text{seg}} + \lambda_{\text{detail}} L_{\text{detail}} \tag{5}$$

where λ is the weight of the corresponding loss, indicating each branch's importance. According to [25] and [34], λ_{pos} is 1, λ_{seg} is 3, and λ_{detail} is 1.

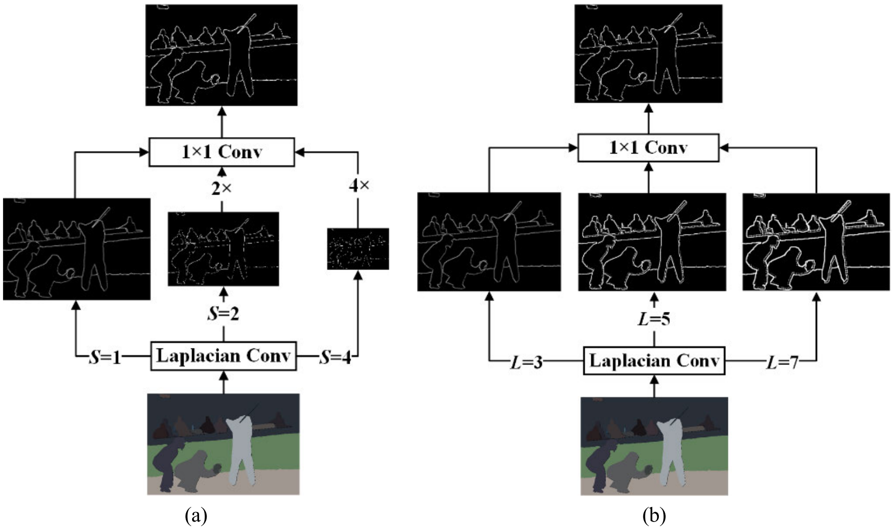
III. ABLATION STUDIES

This section describes the detailed experiment setting first. Then we evaluate the contribution of two key modules, including the *Detail Branch* and the *Detail Ground-truth Generator*, on the COCO and Cityscapes validation sets, respectively.

A. EXPERIMENTAL SETTING

1) DATASETS

The COCO dataset includes 80 *thing* classes and 53 *stuff* classes. The number of its training, testing, and validation images are 118K, 20K, and 5K, respectively, with different resolutions for each image. We randomly flip and rescale the shorter edge from 640 to 800 pixels. The Cityscapes dataset contains street view images captured by in-vehicle cameras. The number of its training, testing, and validation images are 2975, 1525, and 500, respectively, all with a resolution of 1024 × 2048. We randomly scale the input images by 0.5 to 2× and crop them to 512 × 1024 pixels.



(a)

(b)

FIGURE 3. Detail Ground-truth Generator. (a) Method used in [34]. S means stride, Conv denotes convolution, and 1 × 1 indicates kernel size. 2×, 4× represents upsampling of 2× and 4×, respectively. (b) Method used in this paper. L indicates the dimension of the Laplacian kernel.

TABLE 3. Results of COCO. Panoptic FCN results from testing the model on our device using the open-source 1× training strategy [25]. Ours-Detail Branch means adding the Detail Branch and the Aggregation Module to Panoptic FCN. Furthermore, Ours-Multistride Laplacian uses the Laplacian kernel as in [34]. By contrast, Ours-Multidim Laplacian adds three different dimensional Laplacian kernels. Among the table, th means things, and st denotes stuff, respectively. The best results are marked in bold.

Network	PQ	SQ	RQ	PQ_th	SQ_th	RQ_th	PQ_st	SQ_st	RQ_st
Panoptic FCN	41.12	79.75	49.93	46.81	81.80	56.38	32.52	76.66	40.20
Ours-Detail Branch	41.33	79.50	50.07	46.80	81.36	56.15	33.06	76.69	**40.90**
Ours-Multistride Laplacian	**41.36**	**80.11**	50.11	46.83	82.15	56.25	**33.10**	**77.02**	40.83
Ours-Multidim Laplacian	**41.36**	79.98	**50.12**	**47.07**	**82.32**	**56.47**	32.74	76.45	40.53

2) OPTIMIZATION

The network is optimized using stochastic gradient descent (SGD) with weight decay 1e-4 and momentum 0.9. And *WarmupPoly* strategy with power 0.9 is used, i.e., the learning rate is:

$$\mathrm{lr} = \mathrm{base\_lr} \times (1 - \frac{\mathrm{iter}}{\mathrm{max\_iter}})^{\mathrm{power}} \tag{6}$$

COCO's base learning rate *base_lr* is 0.01 with the max iteration *max_iter* 90K, and the batch size is 16. For Cityscapes, we set *base_lr* to 0.02, the max iteration to 65K, and the batch size to 32. *iter* denotes the current number of iterations. ResNet parameters pre-trained on ImageNet are used for initialization.

The experiments are performed on detectron2 [37], using the PyTorch-1.7.1 deployed on Ubuntu 20.04. We conduct training under CUDA 11.0 and CUDNN 8.0.5 on 8 NVIDIA GTX 2080Ti GPUs.

3) METRICS

We select Panoptic Quality (PQ), Semantic Quality (SQ), and Recognition Quality (RQ) to evaluate the results on both datasets.

B. ABLATION STUDIES ON COCO

The validation result on COCO is stable for its large image number. Therefore, we first experiment on COCO.

1) DETAIL BRANCH

The architecture includes only the *Detail Branch* and the *Aggregation Module* without using the detail ground-truth. Results are shown in the second row of TABLE 3. Compared with Panoptic FCN, our network has improved PQ from 41.12 to 41.33, with an increase of 0.51%. This increase is mainly contributed by *stuff*, as the PQ_st is boosted from 32.52 to 33.06. Another notable result is that the RQ is optimized from 49.93 to 50.07, improved by 0.28%. *Stuff* contributes most to this growth, as RQ_st grows by 1.74%. However, it is worth noting that the reduced SQ by 0.31%. The decrease is mainly due to the performance of *things*, as SQ_th reduces by 0.54%. Overall, the results show that adding the *Detail Branch* can benefit *stuff* but not *things* on COCO.

2) DETAIL GROUND-TRUTH GENERATOR

As seen from the third and fourth rows of TABLE 3, adding the detail ground-truth in training is helpful. Specifically, the multidimensional Laplacian kernel helps to improve the

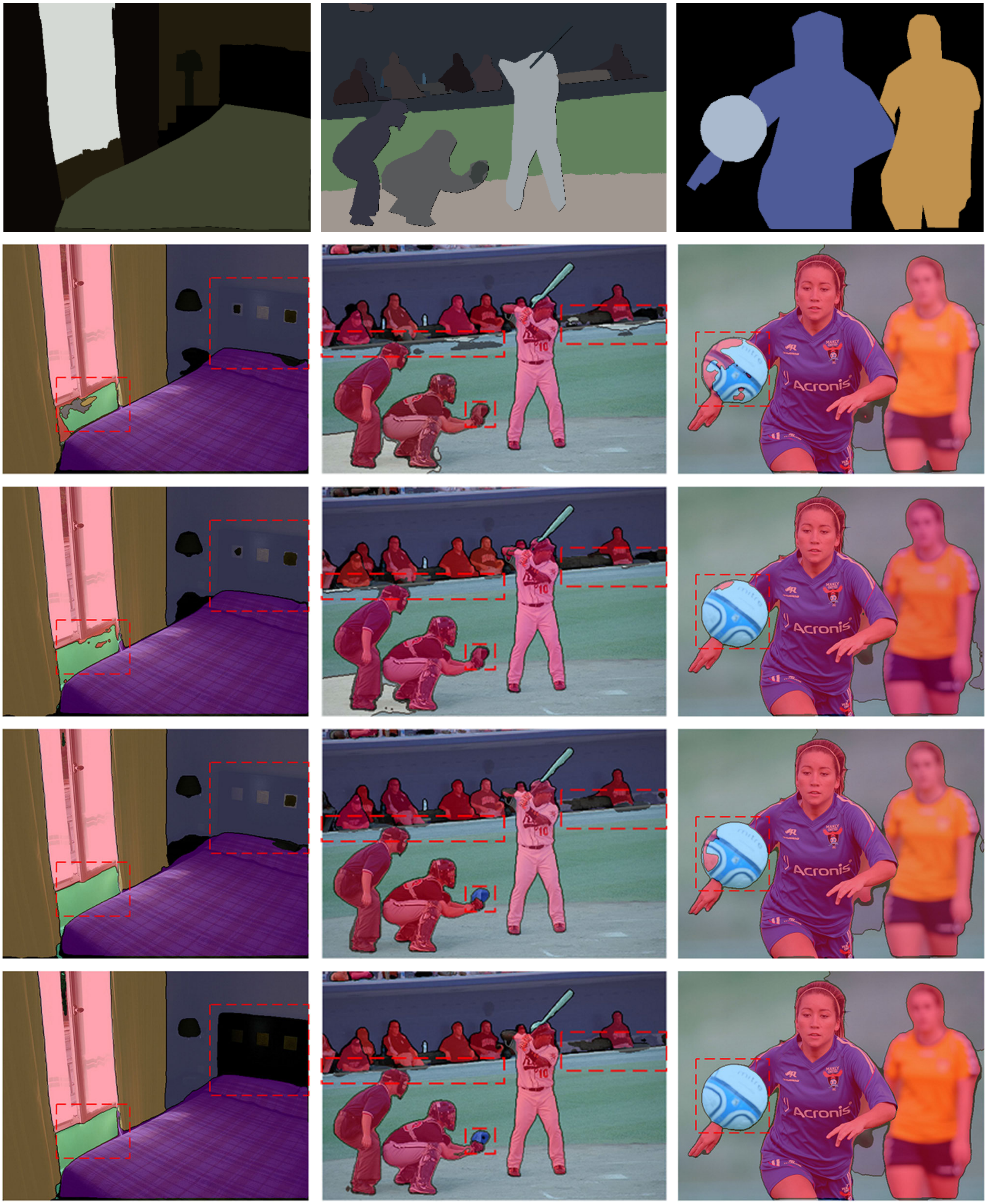


FIGURE 4. Visual comparison of ablation studies. From top to bottom, they are ground-truth, Panoptic FCN, Ours-Detail Branch, Ours-Multistride Laplacian, and Ours-Multidim Laplacian. Significant differences are marked with red rectangular boxes.

results of *things*, where the PQ_th increases by 0.56% compared to Panoptic FCN. Meanwhile, this method optimizes the semantic and instance segmentation of *things*. We note that the SQ_th and RQ_th have been increased by 0.64% and

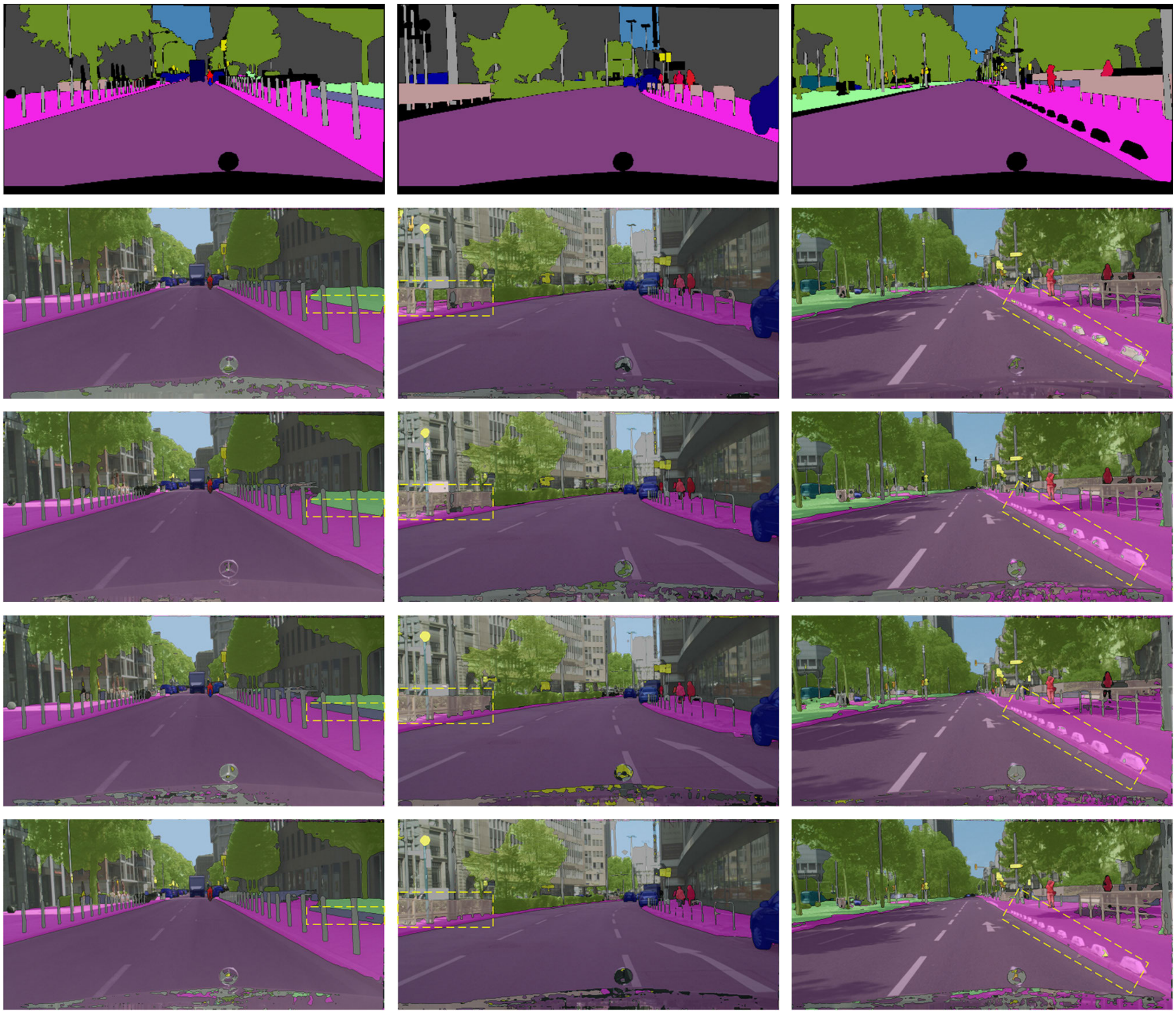


FIGURE 5. Visual comparison of ablation studies. From top to bottom, they are ground-truth, Panoptic FCN, Ours-Detail Branch, Ours-Multistride Laplacian, and Ours-Multidim Laplacian. Significant differences are marked with yellow rectangular boxes.

0.16%, respectively. By contrast, the Laplacian convolutions with different strides are more helpful to *stuff* where the PQ_st is increased by 1.8%. Indeed, we obtain 0.47% and 1.6% increases in SQ_st and RQ_st, respectively. The PQ of both Laplacian kernels is 41.36, with a 0.58% improvement relative to Panoptic FCN.

FIGURE 4 shows a qualitative comparison of some results on COCO, where images with significant differences are marked with red rectangular boxes. It shows that the results are improved after adding different modules. Adding multidimensional Laplacian kernels obtains the best results with higher classification accuracy and better segmentation integrity.

C. ABLATION STUDIES ON CITYSCAPES

The Cityscapes dataset contains rich street-view images that are crucial to autonomous driving. Unlike the COCO dataset, images in Cityscapes include relatively few categories of *things*, with a larger proportion of pixels from *stuff* regions, which is a big challenge for the network.

1) DETAIL BRANCH

As shown in TABLE 4, our network improves PQ by 0.41% from 59.02 to 59.26. The influence mainly comes from *things*, with PQ_th improving by 2.2%. Moreover, we achieve a rise of SQ from 79.63 to 80.16, which comes from SQ_th and SQ_st with improvements of 1.10% and 0.36%, respectively. However, it should be noted that we get a drop in RQ by 0.23% relative to Panoptic FCN. The decrease in RQ is mainly due to the localization of *stuff*, with RQ_st decreasing by 0.43%. The results show that on Cityscapest, the semantic segmentation of *things* and *stuff* could be enhanced by adding the *Detail Branch*. But the localization accuracy of both is slightly reduced.

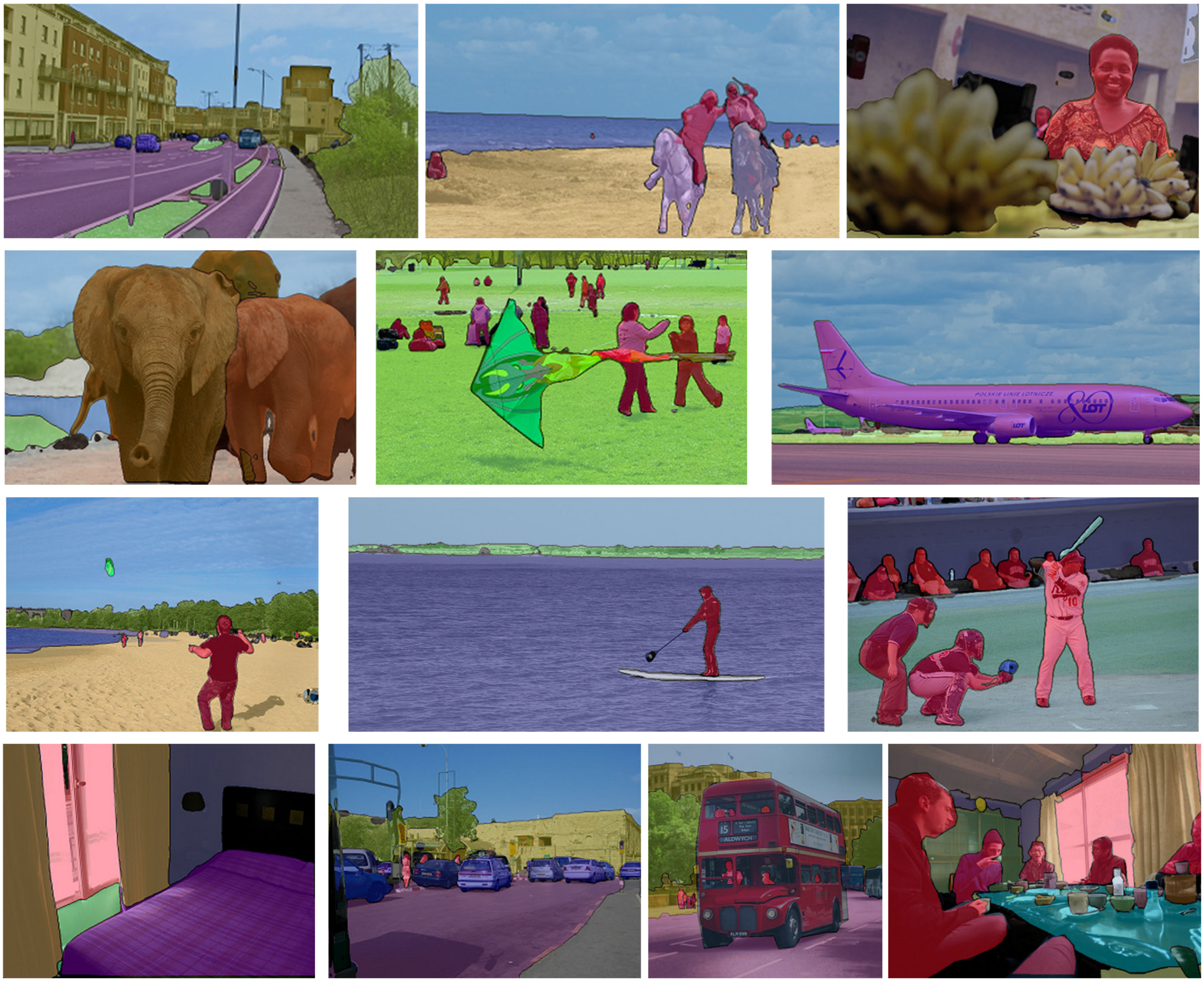


FIGURE 6. Visual results on COCO validation dataset.

TABLE 4. Results of Cityscapes. The Panoptic FCN model trained on Cityscapes is not released, so the data in the table are trained and tested on our device using the same configuration. The rest symbols are the same as in TABLE 3. The best results are marked in bold.

Network	PQ	SQ	RQ	PQ_th	SQ_th	RQ_th	PQ_st	SQ_st	RQ_st
Panoptic FCN	59.02	79.63	72.84	50.60	77.81	64.69	65.15	80.96	78.82
Ours-Detail Branch	59.26	**80.16**	72.67	**51.16**	78.66	64.67	65.15	**81.25**	78.48
Ours-Multistride Laplacian	59.72	80.14	73.35	51.11	**78.73**	64.58	65.98	81.17	79.73
Ours-Multidim Laplacian	**59.89**	80.02	**73.70**	50.99	78.37	**64.81**	**66.38**	81.21	**80.17**

2) DETAIL GROUND-TRUTH GENERATOR

The performance of adding the detail ground-truth improves significantly, similar to that on COCO. We achieve a high PQ from 59.02 to 59.89 with a growth rate of 1.47%. As shown in rows three and four, multidimensional Laplacian kernels help to enhance the result of *stuff* with the PQ_st increases by 1.89% relative to Panoptic FCN. We get 1.71% and 0.31% improvement in SQ_st and RQ_st, respectively. It is indicated that this modification positively influences *stuff*, especially for semantic segmentation. Besides, the Laplacian convolutions with different strides help to improve the result of *things*, and the PQ_th increases by 1.01%. Overall, the best result is using multidimensional Laplacian kernels, the same as COCO.

FIGURE 5 compares some results on Cityscapes. Significant differences are marked with yellow rectangular boxes. With the addition of different modules, the panoptic segmentation quality is improving. On the whole, the best result is adding multidimensional Laplacian kernels.

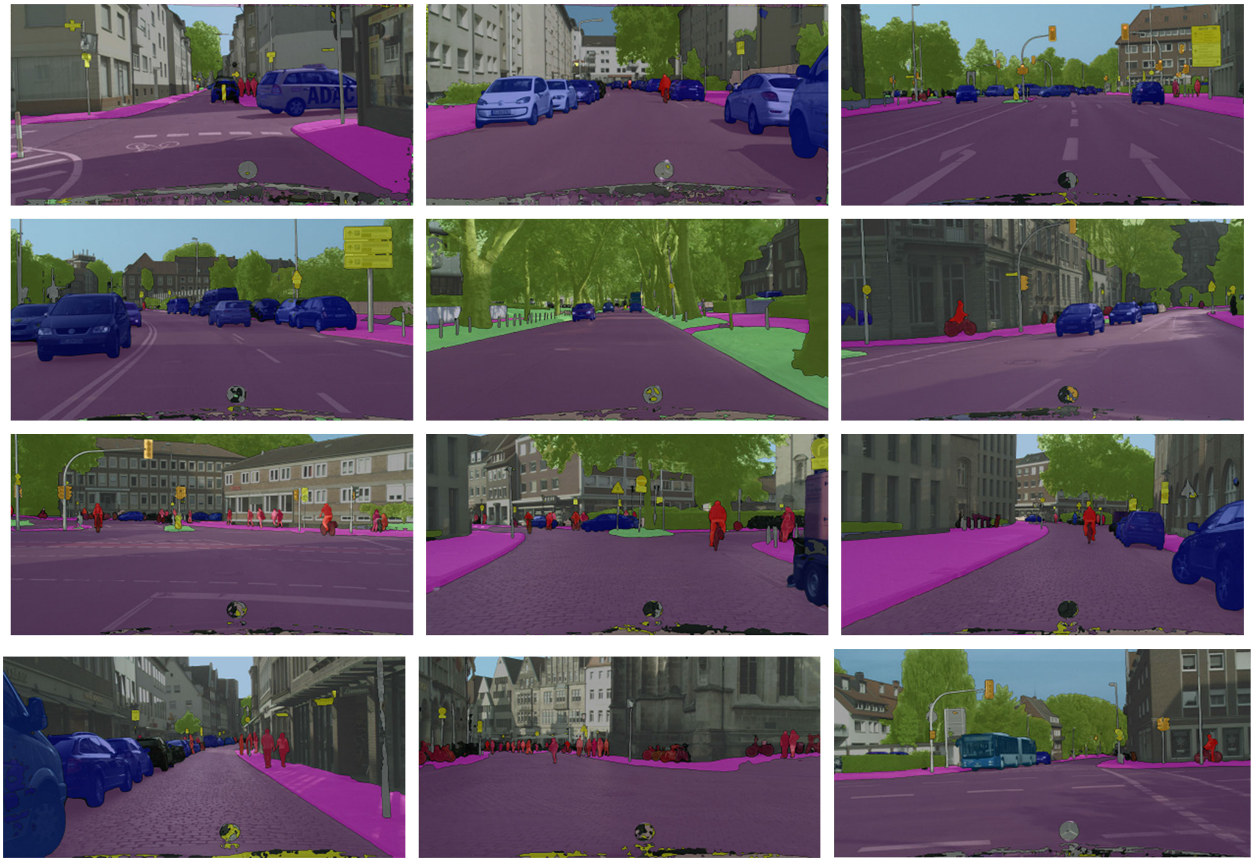


FIGURE 7. Visual results on Cityscapes validation dataset.

TABLE 5. Results on COCO validation dataset. The best results are marked in bold. Ours-512, 600, 800 means utilizing smaller input instead of the default size. And the default is 800, consistent with detectron2 [37].

Network	Backbone	PQ	PQ_th	PQ_st	FPS
		Box-based			
Panoptic FPN[6]	Res50-FPN	39.0	45.9	28.7	-
CIAE[5]	Res50-FPN	40.2	45.3	32.3	12.5
AttentionPS[10]	Res50-FPNlite	33.4	37.8	26.7	**31.2**
		Box-free			
Panoptic-DeepLab[15]	Res50	35.1	-	-	20.0
PCV[17]	Res50-FPN	37.5	40.0	**33.7**	5.7
Panoptic FCN	Res50-FPN	41.1	46.8	32.5	12.0
Ours-512	Res50-FPN	39.2	44.2	31.7	17.8
Ours-600	Res50-FPN	40.5	45.8	32.5	14.7
Ours-800(default)	Res50-FPN	**41.3**	**47.1**	32.7	10.3

PQ is the most important unified metric that evaluates the joint task involving *stuff* and *things* [5]. Both datasets show that the PQ score is improved by adding the *Detail Branch* or the detail ground-truth. It proves the performance of our proposed modules. And the best architecture is the one adding multidimensional Laplacian kernels.

IV. RESULTS AND DISCUSSION

We chose several panoptic segmentation networks for comparison, including box-based and box-free methods. These models use ResNet50 as the backbone, which is consistent with ours and makes the comparison fair. Specifically, our network is the one adding multidimensional Laplacian kernels. In practice, the FPS of our model is measured end-to-end from single input with an average speed of 500 images on an NVIDIA GTX 2080 Ti GPU. We note that the default shortest side of the image is 800, which is the same as detectron2 [37]. In addition, the Panoptic FCN results are tested on our device using the 1× training strategy [25].

A. RESULTS ON COCO

As shown in TABLE 5, our network achieves the highest accuracy. Compared to box-based methods, the accuracy of our model is much higher than the other three methods [5], [6], [10]. Especially the method proposed in [10] achieves

TABLE 6. Results on Cityscapes validation dataset. The best results are marked in bold.

Network	Backbone	PQ	PQ_th	PQ_st	FPS
	Box-based				
Panoptic FPN[6]	Res50-FPN	57.7	51.6	62.2	-
CIAE[5]	Res50-FPN	-	-	-	-
AttentionPS[10]	Res50-FPNlite	59.3	52.8	64.7	10.9
	Box-free				
Panoptic-DeepLab[15]	Res50	59.7	-	-	8.5
PCV[17]	Res50-FPN	54.2	47.8	58.9	5.5
Panoptic FCN	Res50-FPN	59.0	50.6	65.2	4.9
Ours-768	Res50-FPN	59.3	51.3	65.1	7.7
Ours-896	Res50-FPN	60.2	51.8	66.3	6.0
Ours-1024(default)	Res50-FPN	59.9	51.0	66.4	4.8

TABLE 7. Results on Cityscapes validation dataset with fine-tuning.

Network	PQ	SQ	RQ	PQ_th	SQ_th	RQ_th	PQ_st	SQ_st	RQ_st
Panoptic FCN	61.37	80.61	75.01	55.25	80.04	68.72	65.81	81.02	79.58
Ours-Detail Branch	62.00	80.87	75.69	55.68	**80.19**	69.23	**66.59**	81.37	80.38
Ours-Multistride Laplacian	62.10	**80.90**	75.73	**55.96**	80.10	**69.57**	66.56	**81.48**	80.20
Ours-Multidim Laplacian	**62.15**	80.89	**75.80**	55.94	80.16	69.47	66.67	81.42	**80.41**

the best speed but the worst accuracy, whose primary goal is making the model lightweight. Compared to Panoptic-DeepLab [15], the PQ boosts from 35.1 to 41.3, with a growth rate of 17.7%. Moreover, we achieve a 10.1% higher PQ score than PCV [17] and 0.58% higher than Panoptic FCN. We notice the PQ is the highest on both *things* and *stuff*, which are 47.1 and 32.7, respectively. However, the convolutional layers increase due to the *Detail Branch* and the *Aggregation Module*, leading to slower inference speed.

Furthermore, the performance of the model with different input sizes is verified. We set the shortest side of input images to 512, 600, and 800, respectively. It is noted in TABLE 5 that as the input size increases, the panoptic quality becomes better, but the inference speed gradually decreases. Ultimately, we achieve the best accuracy with the default size of 800.

FIGURE 6 shows the visualized results on COCO. Our model deals with common *things*, such as people, cars, animals, etc., more precisely. Different instances in the crowd can be correctly segmented. But it is difficult to segment *stuff* regions accurately, such as woods and meadows, which is the weakness of most panoptic segmentation methods.

B. RESULTS ON CITYSCAPES

As shown in TABLE 6, we obtain the highest panoptic quality accuracy on Cityscapes. Compared to Panoptic FPN, the PQ shows a 3.8% increase from 57.7 to 59.9. Another box-based method, AttentionPS, is the fastest network with a slight decline in accuracy compared to our model. In addition, a 1.47% higher PQ score is obtained compared to Panoptic FCN. Note that our model achieves a much higher PQ on *stuff* but a slightly smaller PQ on *things* than Panoptic FPN.

As performed on COCO, we transform the input size to obtain a surprising result. Indeed, the default image size of Cityscapes is 1024 × 2048. But the best accuracy is achieved by the one with the shortest side of 896 pixels. In practice, we scale images by 0.5 to 2× and crop them to 512 × 1024 pixels in the training stage, which is the strategy adopted in many other works [6], [25]. So, the input size of training is different from the default image size, which we believe is the reason for the result.

In practice, we notice another surprising result. Firstly, we scale images by 0.875 and 1× and crop them to 512 × 1024 pixels. Then, we perform fine-tuning based on the trained model with a mini-iteration of 10K. As shown in TABLE 7, we obtain boost results. Unfortunately, we do not find similar results on COCO. This phenomenon might result from the more similar image size between fine-tuning and inference. We believe this trick will be used in future work.

FIGURE 7 shows the visualized results on Cityscapes. Our model segments vehicles and people accurately. And the background is clear. In particular, it is possible to handle instances both near and far, which is essential for autonomous driving.

As can be seen, our network achieves the best results for both datasets compared to the other models. It is proved that the bilateral network guided by the detail ground-truth improves panoptic segmentation effectively. Indeed, low- level features from shallow layers and deep channels can improve segmentation accuracy. Moreover, Laplacian convolutions with multidimension are more effective than multistride. To sum up, we obtain a significant improvement by adding different modules.

V. CONCLUSION

Panoptic segmentation networks utilize the pre-trained backbone, like ResNet or FPN, to extract image features. Although such backbones fuse low-level features with high-level features, stacking many convolutions leads to the weakness of low-level information. To address this problem, we design a unified network with detail guidance for panoptic segmentation. Our network originates from Panoptic FCN with modifications. Specifically, a *Detail Branch* is compiled with shallow layers and deep channels in parallel on its backbone to enhance low-level features. An *Aggregation Module* is designed to fuse low-level and high-level information. The binary detail ground-truth is obtained using multidimensional Laplacian kernels to guide the network to learn detail features. After experiments on COCO and Cityscapes, the unified network achieves better accuracy compared with other box-based or box-free methods. The performance is indicated by panoptic quality. As a result, we obtain a panoptic quality of 41.3 and 59.9 on COCO and Cityscapes, respectively. The significant result provides essential guidance for many downstream tasks, such as autonomous driving, augmented reality, etc. In the future, we plan to extend our method in the following directions: (i) using a lightweight backbone to fasten the inference; (ii) combining the proposed method with other panoptic segmentation networks to obtain better performance in practice.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their comments which are very helpful to improve this article.

REFERENCES

[1] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, ''Panoptic segmentation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 9396–9405.

[2] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang, ''An end-to-end network for panoptic segmentation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6165–6174.

[3] W. Mao, J. Zhang, K. Yang, and R. Stiefelhagen, ''Panoptic Lintention network: Towards efficient navigational perception for the visually impaired,'' in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Xining, China, Jul. 2021, pp. 857–862.

[4] Y. Yang, H. Li, X. Li, Q. Zhao, J. Wu, and Z. Lin, ''SOGNet: Scene overlap graph network for panoptic segmentation,'' in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, vol. 34, no. 7, 2020, pp. 12637–12644.

[5] N. Gao, Y. Shan, X. Zhao, and K. Huang, ''Learning category- and instance-aware pixel embedding for fast panoptic segmentation,'' *IEEE Trans. Image Process.*, vol. 30, pp. 6013–6023, 2021.

[6] A. Kirillov, R. Girshick, K. He, and P. Dollár, ''Panoptic feature pyramid networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6392–6401.

[7] J. Brünger, M. Gentz, I. Traulsen, and R. Koch, ''Panoptic segmentation of individual pigs for posture recognition,'' *Sensors*, vol. 20, no. 13, p. 3710, Jul. 2020.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[9] D. de Geus, P. Meletis, and G. Dubbelman, ''Fast panoptic segmentation network,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1742–1749, Apr. 2020.

[10] A. Petrovai and S. Nedevschi, ''Fast panoptic segmentation with soft attention embeddings,'' *Sensors*, vol. 22, no. 3, p. 783, Jan. 2022.

[11] M. Weber, J. Luiten, and B. Leibe, ''Single-shot panoptic segmentation,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 8476–8483.

[12] Z. Tian, C. Shen, H. Chen, and T. He, ''FCOS: Fully convolutional one-stage object detection,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 9626–9635.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ''Focal loss for dense object detection,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2999–3007.

[14] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, ''YOLACT: Real-time instance segmentation,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 9156–9165.

[15] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, ''Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 12472–12482.

[16] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen, ''DeeperLab: Single-shot image parser,'' 2019, *arXiv:1902.05093*.

[17] H. Wang, R. Luo, M. Maire, and G. Shakhnarovich, ''Pixel consensus voting for panoptic segmentation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 9461–9470.

[18] 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.

[19] 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. (ECCV)*. Munich, Germany: Springer, 2018, pp. 833–851.

[20] D. H. Ballard, ''Generalizing the Hough transform to detect arbitrary shapes,'' *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.

[21] M. Bai and R. Urtasun, ''Deep watershed transform for instance segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, Hawaii, Jul. 2017, pp. 2858–2866.

[22] L. Vincent and P. Soille, ''Watersheds in digital spaces: An efficient algorithm based on immersion simulations,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[23] R. Cipolla, Y. Gal, and A. Kendall, ''Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7482–7491.

[24] J. Uhrig, E. Rehder, B. Fröhlich, U. Franke, and T. Brox, ''Box2Pix: Single-shot instance segmentation by assigning pixels to object boxes,'' in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Changshu, China, Jun. 2018, pp. 292–299.

[25] Y. Li, H. Zhao, X. Qi, L. Wang, Z. Li, J. Sun, and J. Jia, ''Fully convolutional networks for panoptic segmentation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 214–223.

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

[27] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, ''BiSeNet: Bilateral segmentation network for real-time semantic segmentation,'' in *Proc. Eur. Conf. Comput. Vis.* Munich, Germany: Springer, 2018, pp. 334–349.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, ''Microsoft COCO: Common objects in context,'' in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 740–755.

[29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, ''The cityscapes dataset for semantic urban scene understanding,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3213–3223.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, ''Feature pyramid networks for object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.

[31] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, ''An intriguing failing of convolutional neural networks and the CoordConv solution,'' *presented at the Int. Conf. Neural Inf. Process. Syst.*, Montréal, QC, Canada, Dec. 2018.

[32] 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," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, Nov. 2021.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, 2015, vol. 37, pp. 448–456.

[34] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, "Rethinking BiSeNet for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 9711–9720.

[35] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Stanford, CA, USA, Oct. 2016, pp. 565–571.

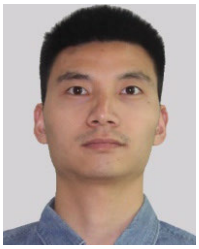
[36] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 570–586.

[37] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. Accessed: Feb. 20, 2023. [Online]. Available: https://github.com/facebookresearch/detectron2

QINGWEI SUN received the master's degree from the National University of Defense Technology, China. He is currently pursuing the Ph.D. degree with the Department of Aerospace Science and Technology, Space Engineering University, China. His current research interests include panoptic segmentation, SLAM, and 3D reconstruction.

JIANGANG CHAO received the M.S. degree from the China Astronaut Research and Training Center and the Ph.D. degree from Xi'an Jiaotong University. He is currently a Professor with the China Astronaut Research and Training Center, responsible for developing astronaut flight simulators. His interests include spacecraft simulation, MR training, VSLAM, and intelligent robot. He is committed to improving the intelligent scene perception of astronaut MR training and has implemented significant projects.

WANHONG LIN received the master's degree from the Institute of Aerospace Medical Engineering, China. He is currently an Associate Professor with the China Astronaut Research and Training Center. His research interests include human and environmental engineering and human–machine interaction technology for complex systems.



ZHENYING XU received the master's degree from the National University of Defense Technology, China. He is currently an Engineer with the China Astronaut Research and Training Center. His research interests include mixed reality and space flight training simulation.

WEI CHEN received the master's degree from the China Astronaut Research and Training Center, China. He is currently an Engineer with the China Astronaut Research and Training Center. His research interests include human and environmental engineering and deep learning.