Towards Fully Convolutional Panoptic Segmentation

Yanwei Li
CUHK

Contents

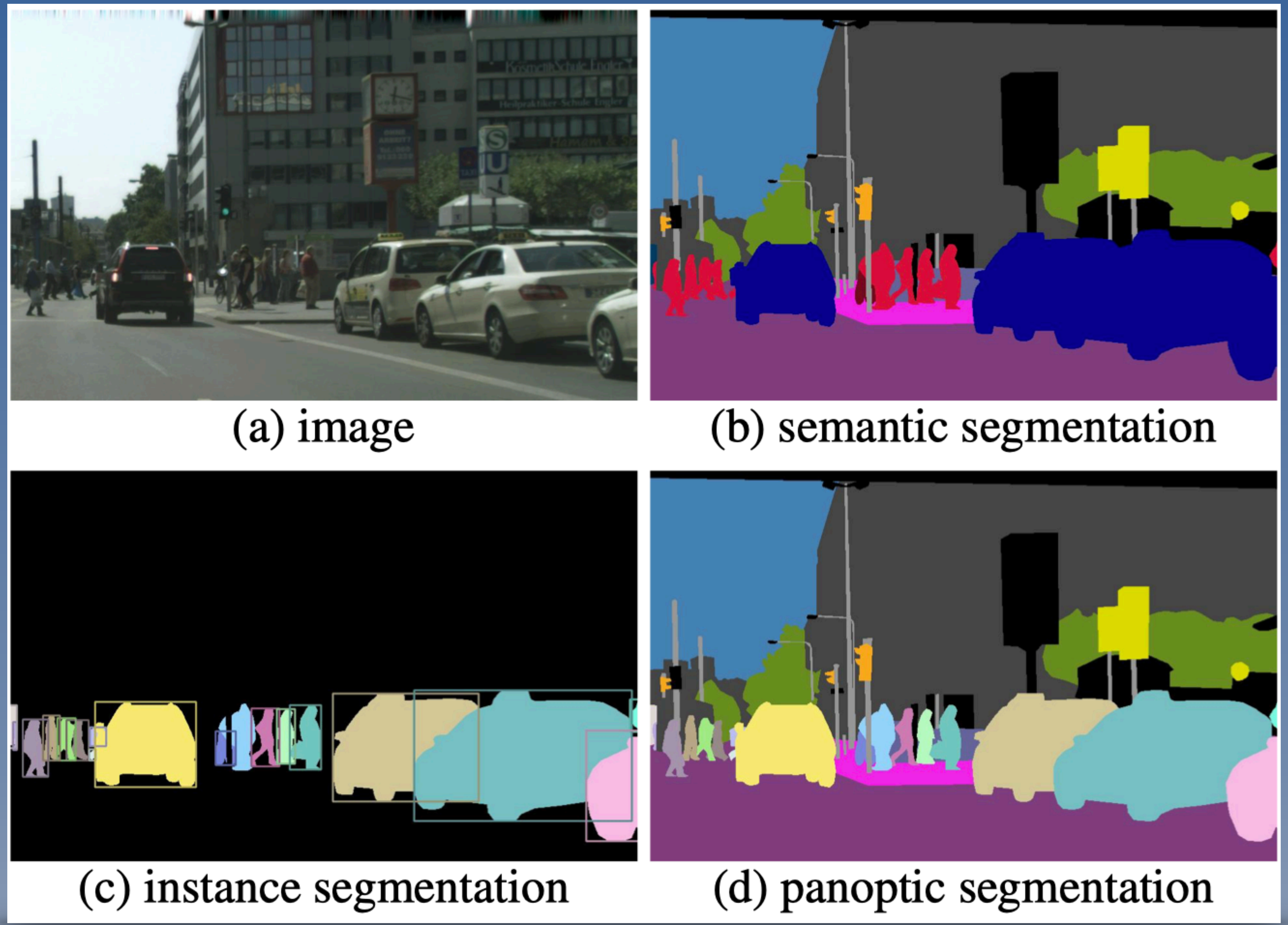
Introduction

Definition of Panoptic Segmentation

Assign each pixel with a semantic label and unique identity to Things and Stuff.

Difficulties in Panoptic Segmentation

- *Conflicting properties of Things and Stuff. Things rely on instance-aware features, while Stuff need semantic-consistent characters.*
- *How to encode things and stuff in a unified representation?*
- *How to model the relationship among things, and between things and stuff?*



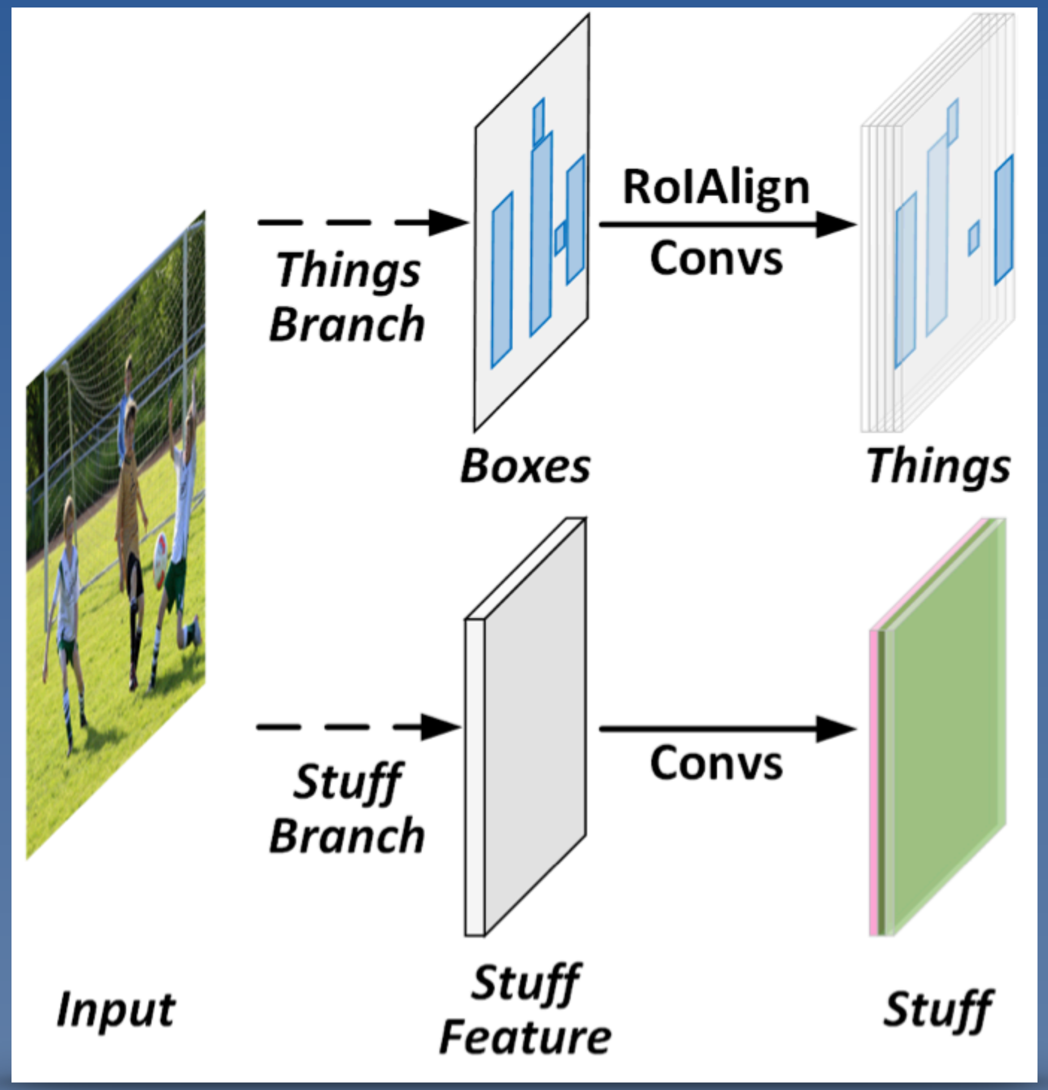
Comparison among tasks. [1]

[1] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Doll´ar. Panoptic segmentation. In *CVPR*, 2019.

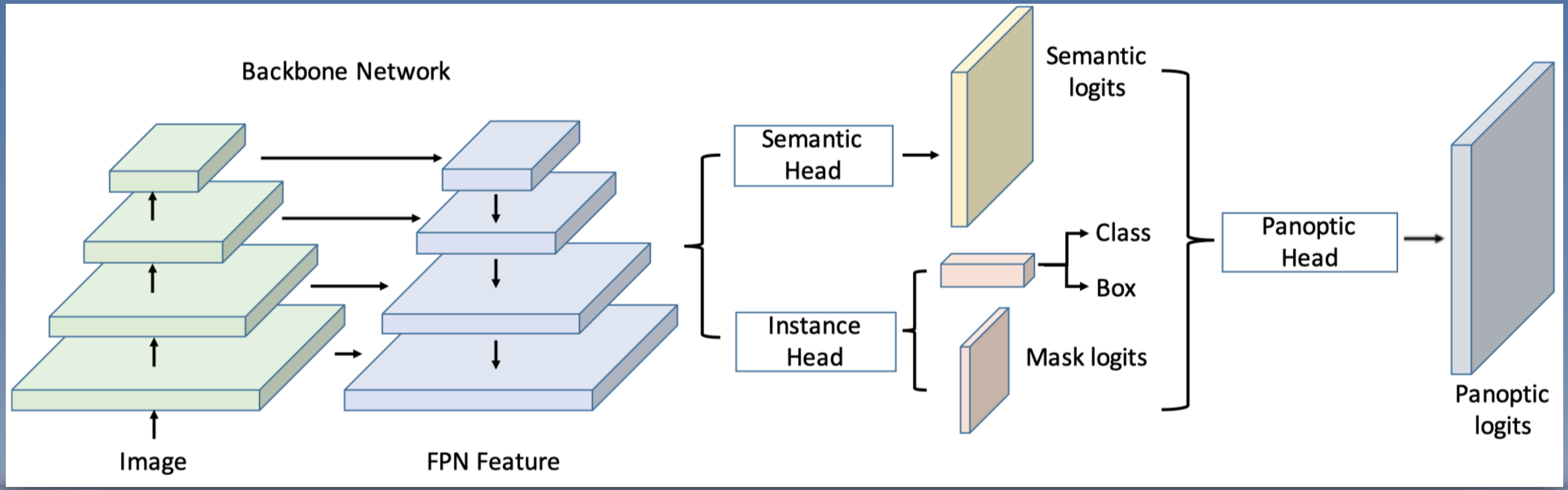
Introduction

Previous methods satisfy demands separately

- *Instance-awareness for things: box-based [2, 3, 4] or box-free [5, 6] branch.*
- *Semantic-consistency for stuff: FCN-based branch.*



Separate representation.



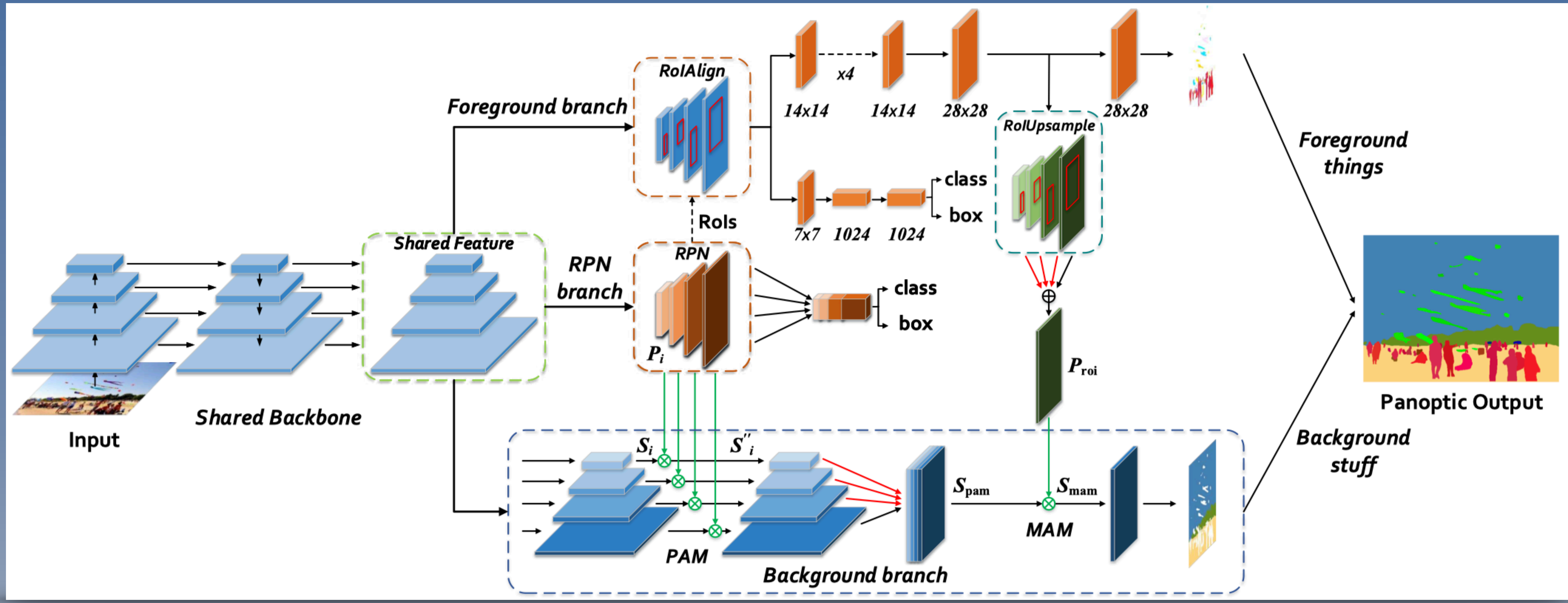
Architecture of UPSNet [3].

[2] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Doll´ar. Panoptic feature pyramid networks. In *CVPR*, 2019.
[3] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.
[4] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *CVPR*, 2019.

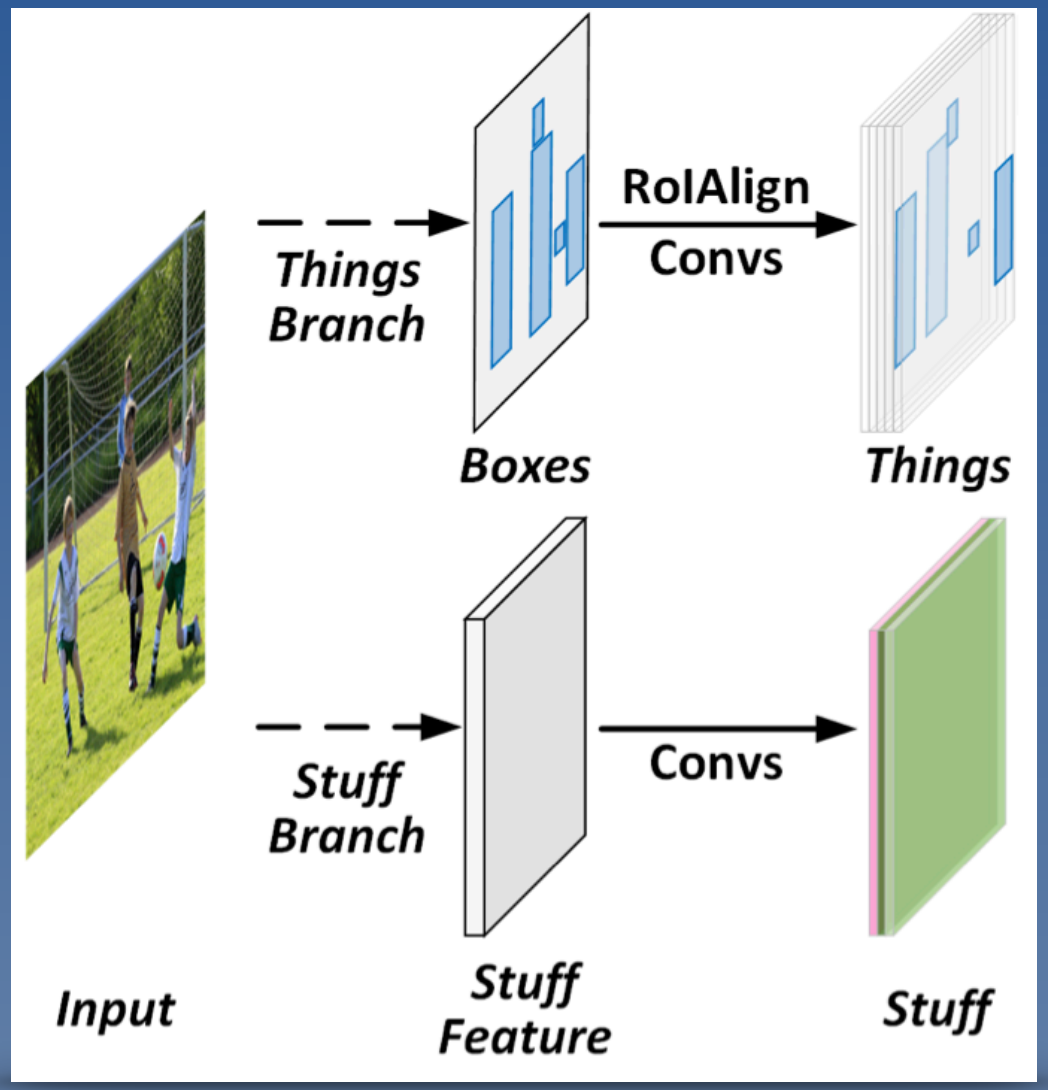
Introduction

Previous methods satisfy demands separately

- *Instance-awareness for things: box-based [2, 3, 4] or box-free [5, 6] branch.*
- *Semantic-consistency for stuff: FCN-based branch.*



Architecture of AUNet [4].



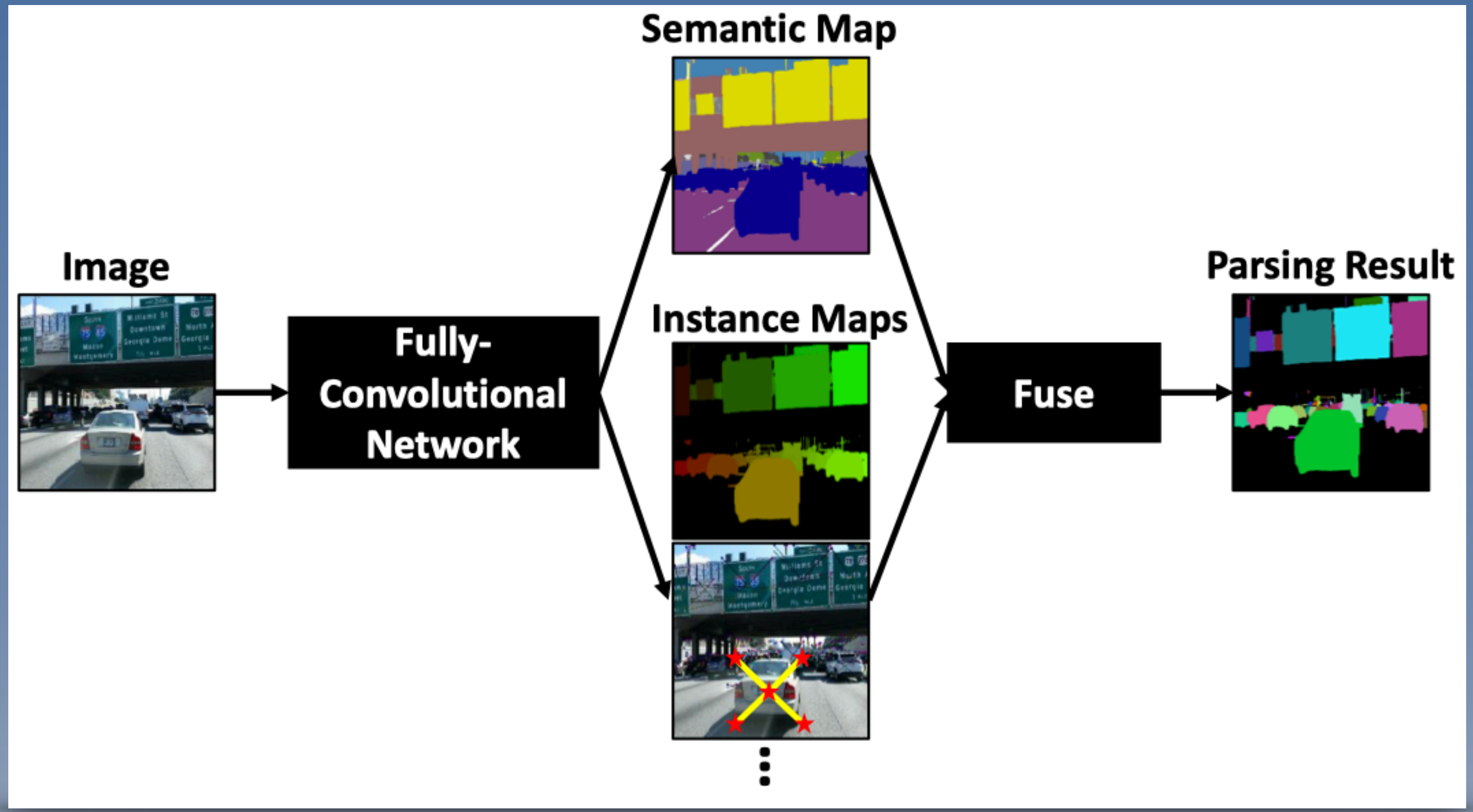
Separate representation.

[2] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Doll´ar. Panoptic feature pyramid networks. In *CVPR*, 2019.
[3] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.
[4] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *CVPR*, 2019.

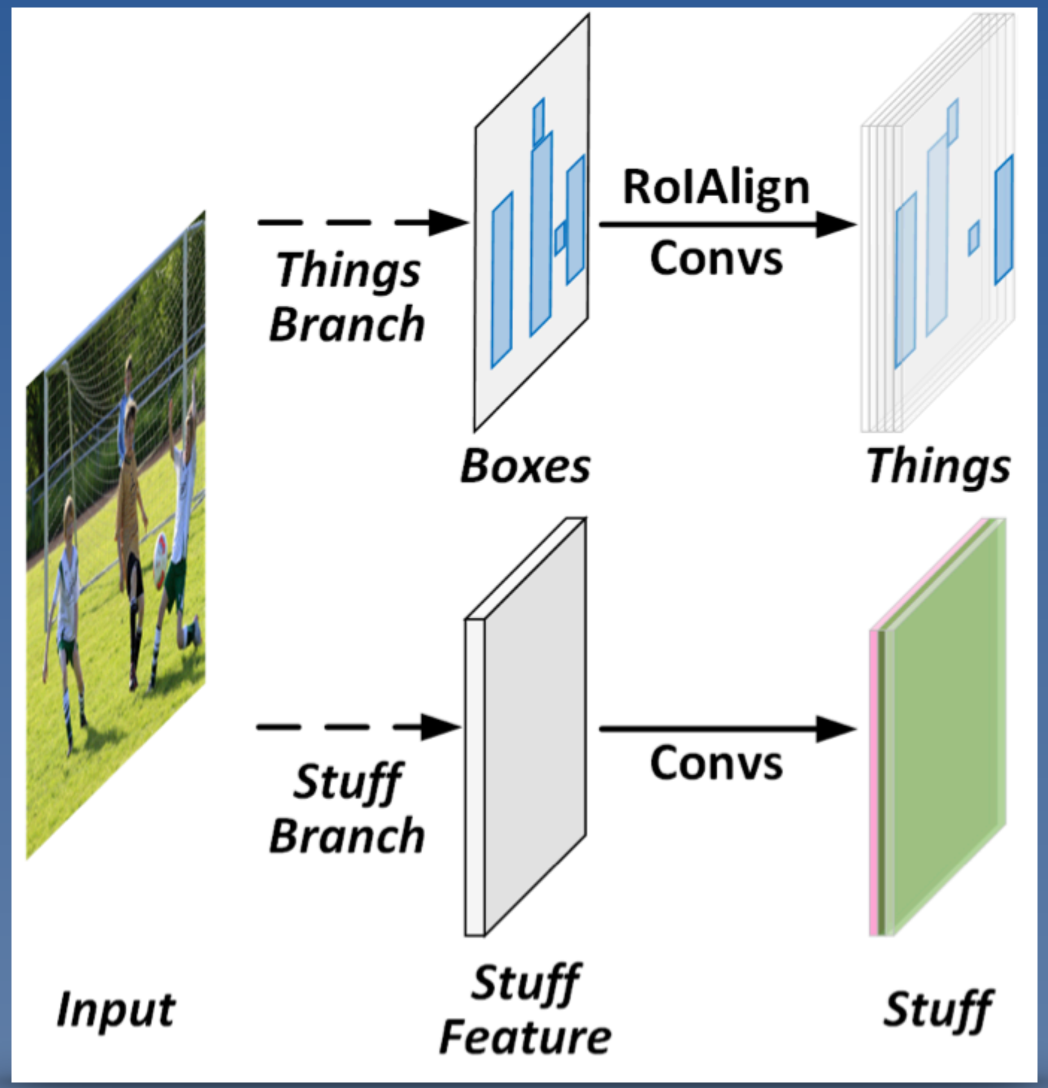
Introduction

Previous methods satisfy demands separately

- *Instance-awareness for things: box-based [2, 3, 4] or box-free [5, 6] branch.*
- *Semantic-consistency for stuff: FCN-based branch.*



Pipeline of DeeperLab [5].



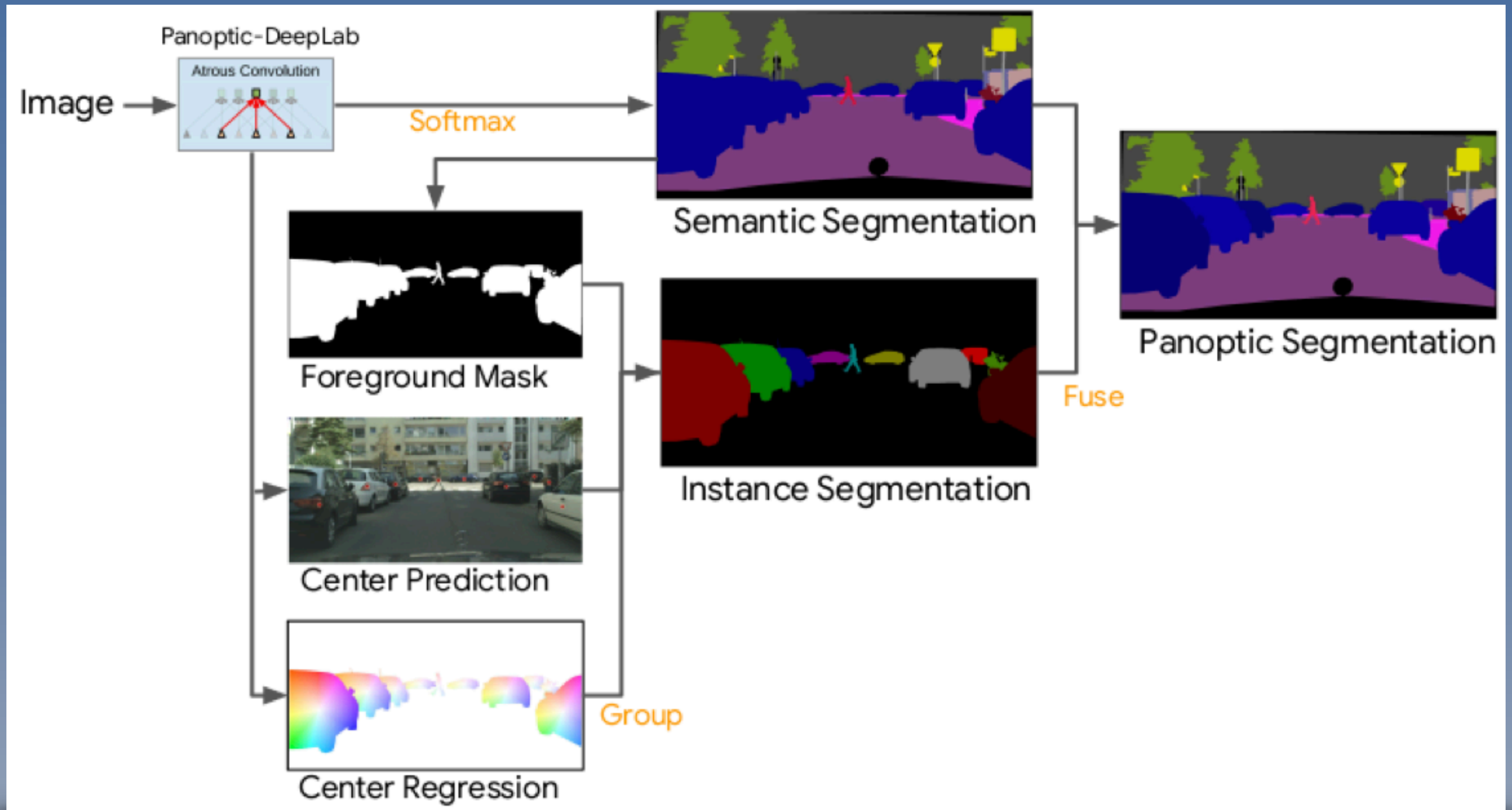
Separate representation.

[5] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv*:1902.05093, 2019.
[6] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020.

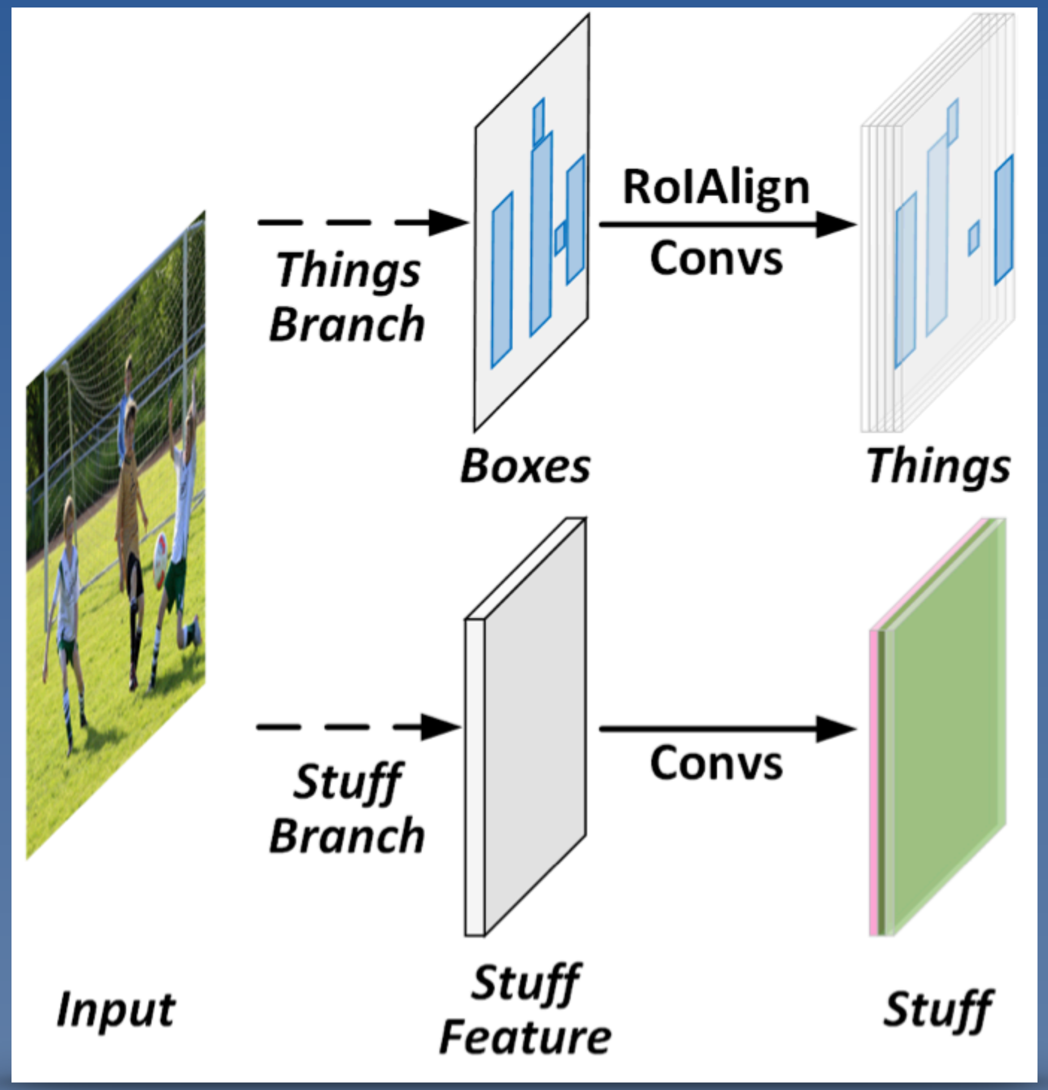
Introduction

Previous methods satisfy demands separately

- *Instance-awareness for things: box-based [2, 3, 4] or box-free [5, 6] branch.*
- *Semantic-consistency for stuff: FCN-based branch.*



Pipeline of Panoptic-DeepLab [6].



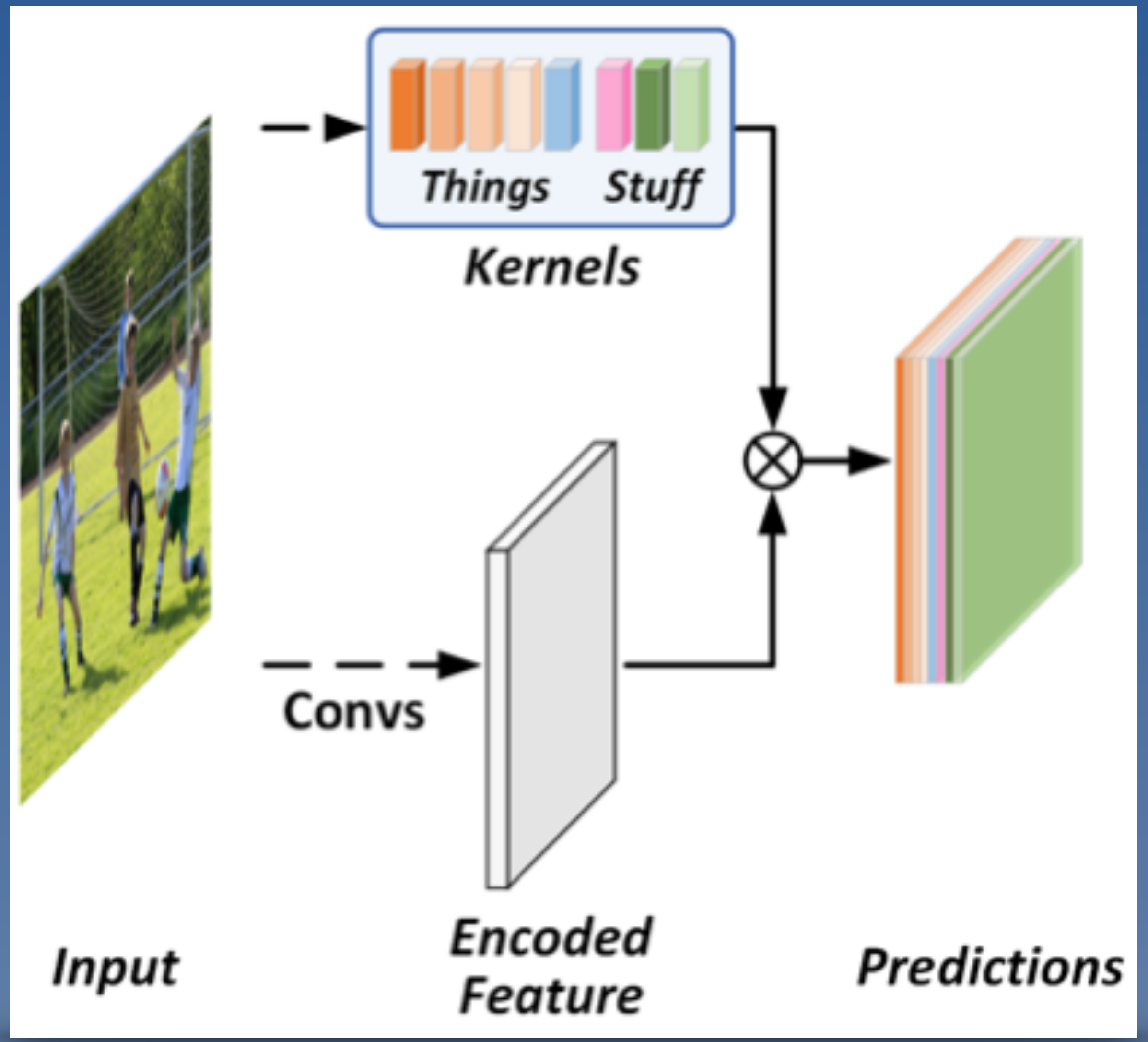
Separate representation.

[5] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv*:1902.05093, 2019.
[6] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020.

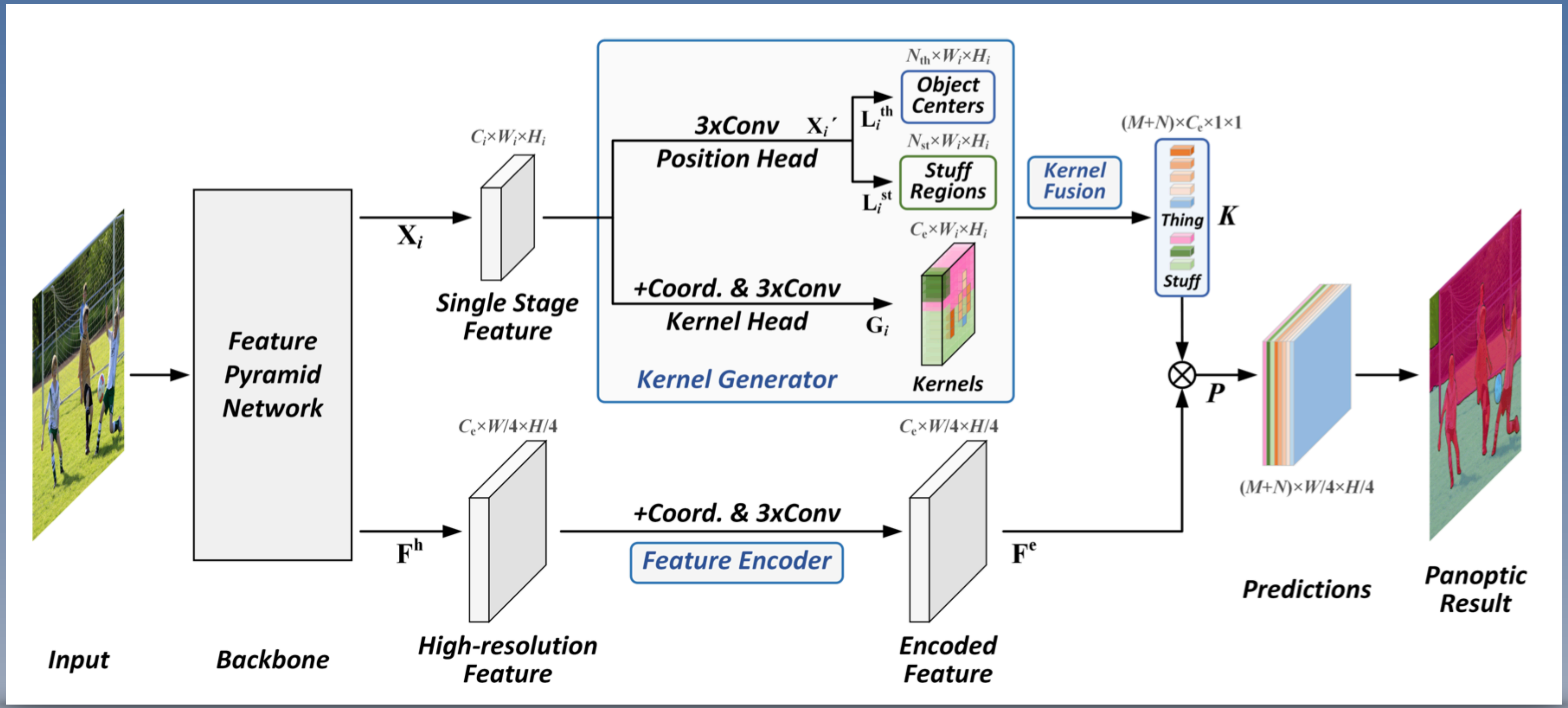
Panoptic FCN

Panoptic FCN represent them uniformly

- *It encodes each instance into a specific kernel and generates the prediction by convolutions directly.*
- *Instance-awareness for things: each thing has unique kernel.*
- *Semantic-consistency for stuff: identical stuff has same kernel.*



Unified representation.



Framework of Panoptic FCN [7].

[7] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, Jiaya Jia. Fully Convolutional Networks for Panoptic Segmentation. In *CVPR*, 2021.

Panoptic FCN

Unified loss function in Panoptic FCN

- *Loss function for position localization*

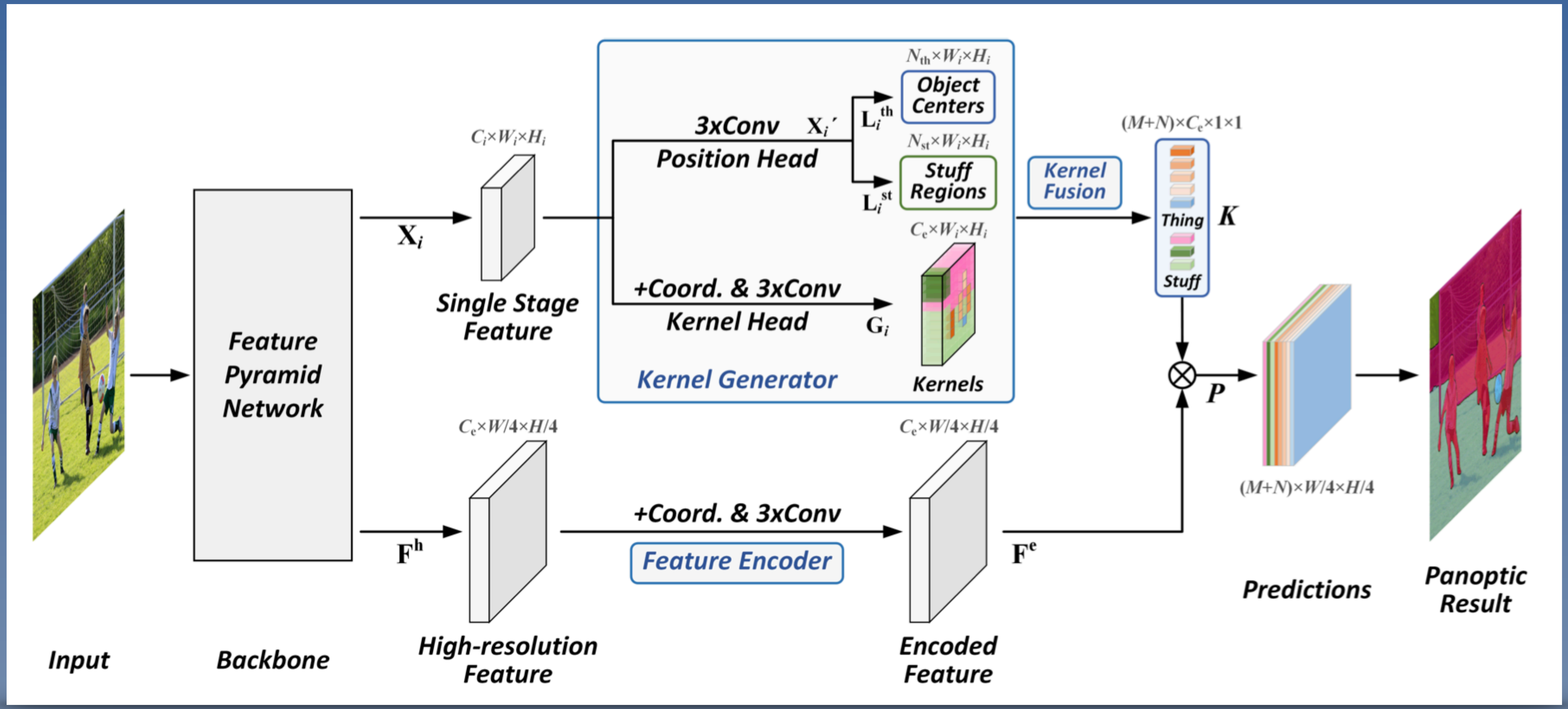
$$\mathcal{L}_{\text{pos}}^{\text{th}} = \sum_i \text{FL}(\mathbf{L}_i^{\text{th}}, \mathbf{Y}_i^{\text{th}})/N_{\text{th}},$$

$$\mathcal{L}_{\text{pos}}^{\text{st}} = \sum_i \text{FL}(\mathbf{L}_i^{\text{st}}, \mathbf{Y}_i^{\text{st}})/W_i H_i,$$

- *Loss function for segmentation*

$$\text{WDice}(\mathbf{P}_j, \mathbf{Y}_j^{\text{seg}}) = \sum_k w_k \text{Dice}(\mathbf{P}_{j,k}, \mathbf{Y}_j^{\text{seg}}),$$

$$\mathcal{L}_{\text{seg}} = \sum_j \text{WDice}(\mathbf{P}_j, \mathbf{Y}_j^{\text{seg}})/(M+N),$$



Framework of Panoptic FCN [7].

[7] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, Jiaya Jia. Fully Convolutional Networks for Panoptic Segmentation. In *CVPR*, 2021.

Results & Analysis

Component-wise Analysis in Panoptic FCN

Ablation studies on kernel generator and feature encoder.

Table 1. Comparisons among different settings of the kernel generator on the COCO *val* set. *deform* and *conv num* respectively denote deformable convolutions for position head and number of convolutions in both heads of the kernel generator.

deform	*conv num*	PQ	PQ^{th}	PQ^{st}	AP	mIoU
✗	1	38.4	43.4	31.0	28.3	39.9
✗	2	38.9	44.1	31.1	28.9	40.1
✗	3	39.2	44.7	31.0	29.6	40.2
✗	4	39.2	44.9	30.8	29.4	39.9
✓	3	**39.9**	**45.0**	**32.4**	**29.9**	**41.2**

Table 3. Comparisons among different similarity thresholds of kernel fusion on the COCO *val* set. *class-aware* denotes only merging kernel weights with the same predicted class c. And *thres* indicates the cosine similarity threshold *thres* for kernel fusion in Sec. 3.2.

class-aware	*thres*	PQ	PQ^{th}	PQ^{st}	AP	mIoU
✓	0.80	39.7	44.3	32.9	29.9	41.7
✓	0.85	40.8	46.1	32.9	31.5	41.7
✓	0.90	**41.3**	46.9	**32.9**	**32.1**	**41.7**
✓	0.95	41.3	**47.0**	32.9	31.1	41.7
✗	0.90	41.2	46.7	32.9	30.9	41.7

Table 2. Comparisons among different positional settings on the COCO *val* set. $coord_w$ and $coord_f$ denote combining coordinates for the kernel head, and feature encoder, respectively.

$coord_w$	$coord_f$	PQ	PQ^{th}	PQ^{st}	AP	mIoU
✗	✗	39.9	45.0	32.4	29.9	41.2
✓	✗	39.9	45.0	32.2	30.0	41.1
✗	✓	40.2	45.3	32.5	30.4	41.6
✓	✓	**41.3**	**46.9**	**32.9**	**32.1**	**41.7**

Table 4. Comparisons among different channel numbers of the feature encoder on the COCO *val* set. *channel num* represents the channel number C_e of the feature encoder.

channel num	PQ	PQ^{th}	PQ^{st}	AP	mIoU
16	39.9	45.0	32.1	30.8	41.3
32	40.8	46.3	32.5	31.7	41.6
64	**41.3**	46.9	**32.9**	32.1	**41.7**
128	41.3	**47.0**	32.6	**32.6**	41.7

Results & Analysis

Component-wise Analysis in Panoptic FCN

Ablation studies on loss function and feature encoder.

Table 5. Comparisons among different feature types for the feature encoder on the COCO *val* set. *feature type* denotes the method to gernerate high-resolution feature $\mathbf{F}^{\mathrm{h}}$ in Sec. 3.3.

feature type	PQ	$\mathrm{PQ}^{\mathrm{th}}$	$\mathrm{PQ}^{\mathrm{st}}$	AP	mIoU
FPN-P2	40.6	46.0	32.4	31.6	41.3
FPN-Summed	40.5	46.0	32.1	31.7	41.1
Semantic FPN [17]	**41.3**	**46.9**	**32.9**	**32.1**	**41.7**

Table 7. Comparisons among different training schedules on the COCO *val* set. 1×, 2×, and 3× *schedule* denote the 90K, 180K, and 270K training iterations in Detectron2 [47], respectively.

schedule	PQ	$\mathrm{PQ}^{\mathrm{th}}$	$\mathrm{PQ}^{\mathrm{st}}$	AP	mIoU
1×	41.3	46.9	32.9	32.1	41.7
2×	43.2	48.8	34.7	34.3	43.4
3×	**43.6**	**49.3**	**35.0**	**34.5**	**43.8**

Table 6. Comparisons among different settings of weighted dice loss on the COCO *val* set. *weighted* and k denote weighted dice loss and the number of sampled points in Sec. 3.4, respectively.

weighted	k	PQ	$\mathrm{PQ}^{\mathrm{th}}$	$\mathrm{PQ}^{\mathrm{st}}$	AP	mIoU
✗	-	40.2	45.5	32.4	31.0	41.3
✓	1	40.0	45.1	32.4	30.9	41.4
✓	3	41.0	46.4	32.7	31.6	41.4
✓	5	41.0	46.5	32.9	32.1	41.7
✓	7	**41.3**	**46.9**	**32.9**	**32.1**	41.7
✓	9	41.3	46.8	32.9	32.1	**41.8**

Table 8. Comparisons among different settings of the feature encoder on the COCO *val* set. *deform* and *channel num* represent deformable convolutions and the channel number C_{e}, respectively.

deform	*channel num*	PQ	$\mathrm{PQ}^{\mathrm{th}}$	$\mathrm{PQ}^{\mathrm{st}}$	AP	mIoU
✗	64	43.6	49.3	35.0	34.5	43.8
✓	256	**44.3**	**50.0**	**35.6**	**35.5**	**44.0**

Results & Analysis

Component-wise Analysis in Panoptic FCN

Ablation studies on loss function and speed-accuracy.

Table 9. Upper-bound analysis on the COCO *val* set. *gt position* and *gt class* denote utilizing the ground-truth position G_i and class C_i in each position head for kernel generation, respectively.

gt position	*gt class*	PQ	PQ^{th}	PQ^{st}	AP	mIoU
✗	✗	43.6	49.3	35.0	34.5	43.8
✓	✗	49.8	52.2	46.1	38.2	54.6
✓	✓	**65.9**	**64.1**	**68.7**	**45.5**	**86.6**
		+22.3	*+14.8*	*+33.7*	*+11.0*	*+42.8*

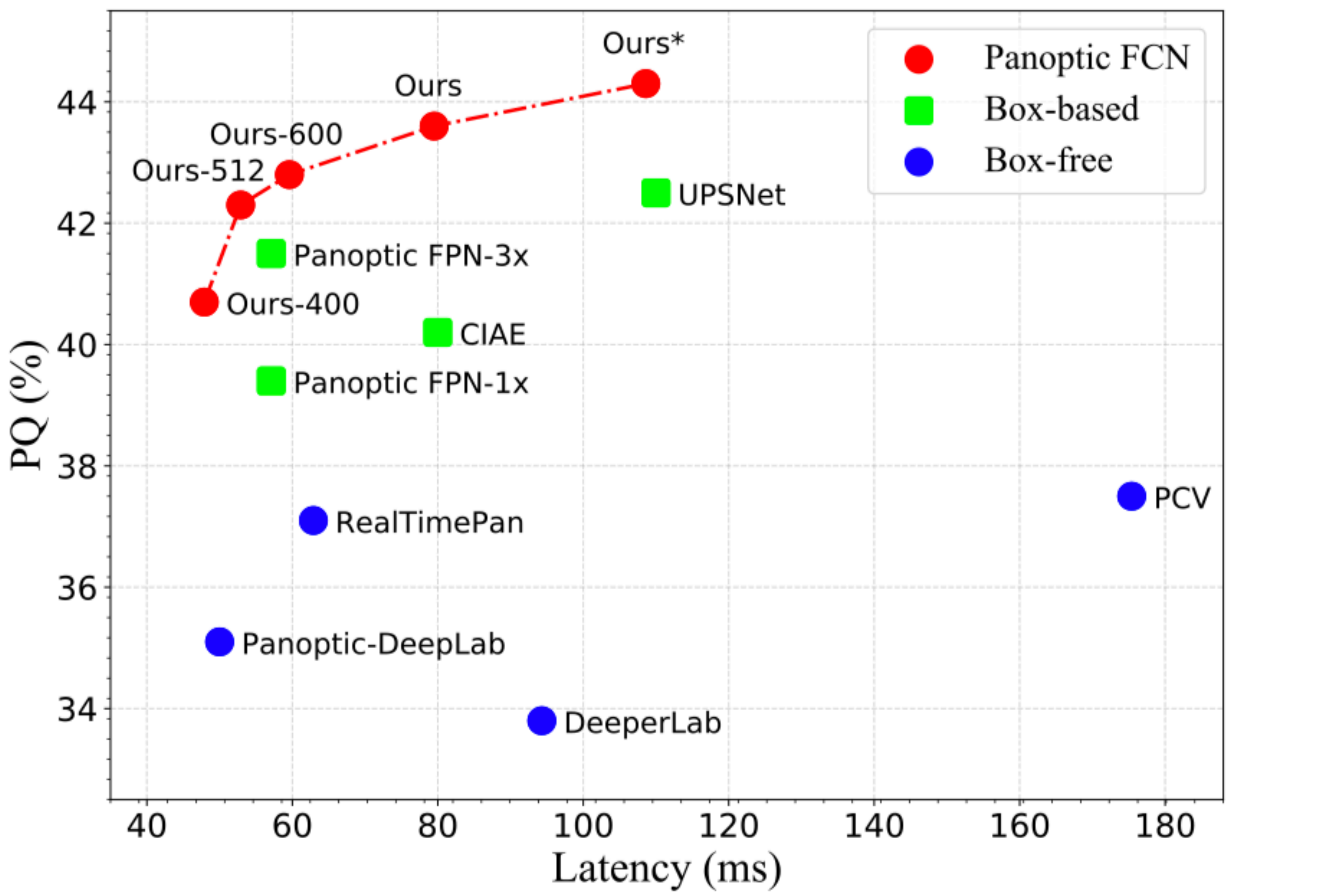


Figure 3. Speed-Accuracy trade-off curve on the COCO *val* set. All the results are compared with Res50 except DeeperLab [49] based on Xception-71 [7]. The latency is measured *end-to-end* from single input to panoptic result. Details are given in Table 10.

Results & Analysis

Results of Panoptic FCN

It surpasses previous box-based and box-free methods with efficiency.

Table 10. Comparisons with previous methods on the COCO *val* set. Panoptic FCN-400, 512, and 600 denotes utilizing smaller input instead of the default setting. All of our results are achieved on the same device with single input and no flipping. FPS is measured *end-to-end* from single input to panoptic result with an average speed over 1,000 images, which could be further improved with more optimizations. The simple enhanced version is marked with *. The model testing by ourselves according to released codes is denoted as †.

Method	Backbone	PQ	SQ	RQ	PQ^{th}	SQ^{th}	RQ^{th}	PQ^{st}	SQ^{st}	RQ^{st}	Device	FPS
box-based												
Panoptic FPN [17]	Res50-FPN	39.0	-	-	45.9	-	-	28.7	-	-	-	-
Panoptic FPN†-1×	Res50-FPN	39.4	77.8	48.3	45.9	80.9	55.3	29.6	73.3	37.7	V100	17.5
Panoptic FPN†-3×	Res50-FPN	41.5	79.1	50.5	48.3	82.2	57.9	31.2	74.4	39.5	V100	17.5
AUNet [24]	Res50-FPN	39.6	-	-	49.1	-	-	25.2	-	-	-	
CIAE [11]	Res50-FPN	40.2	-	-	45.3	-	-	32.3	-	-	2080Ti	12.5
UPSNet† [48]	Res50-FPN	42.5	78.0	52.5	48.6	79.4	**59.6**	33.4	75.9	41.7	V100	9.1
Unifying [23]	Res50-FPN	43.4	79.6	53.0	48.6	-	-	35.5	-	-	-	-
box-free												
DeeperLab [49]	Xception-71	33.8	-	-	-	-	-	-	-	-	V100	10.6
Panoptic-DeepLab [6]	Res50	35.1	-	-	-	-	-	-	-	-	V100	20.0
AdaptIS [40]	Res50	35.9	-	-	40.3	-	-	29.3	-	-	-	-
RealTimePan [14]	Res50-FPN	37.1	-	-	41.0	-	-	31.3	-	-	V100	15.9
PCV [42]	Res50-FPN	37.5	77.7	47.2	40.0	78.4	50.0	33.7	76.5	42.9	1080Ti	5.7
SOLO V2 [45]	Res50-FPN	42.1	-	-	49.6	-	-	30.7	-	-	-	-
Panoptic FCN-400	Res50-FPN	40.7	80.5	49.3	44.9	82.0	54.0	34.3	78.1	42.1	V100	**20.9**
Panoptic FCN-512	Res50-FPN	42.3	80.9	51.2	47.4	82.1	56.9	34.7	79.1	42.7	V100	18.9
Panoptic FCN-600	Res50-FPN	42.8	80.6	51.6	47.9	82.6	57.2	35.1	77.4	43.1	V100	16.8
Panoptic FCN	Res50-FPN	43.6	80.6	52.6	49.3	82.6	58.9	35.0	**77.6**	42.9	V100	12.5
Panoptic FCN*	Res50-FPN	**44.3**	**80.7**	**53.0**	**50.0**	**83.4**	59.3	**35.6**	76.7	**43.5**	V100	9.2

Results & Analysis

Results of Panoptic FCN

It surpasses previous box-based and box-free methods with efficiency.

Table 11. Experiments on the COCO *test-dev* set. All of our results are achieved with single scale input and no flipping. The simple enhanced version and *val* set for training are marked with * and ‡.

Method	Backbone	PQ	PQ^{th}	PQ^{st}
		box-based		
Panoptic FPN [17]	Res101-FPN	40.9	48.3	29.7
CIAE [11]	DCN101-FPN	44.5	49.7	36.8
AUNet [24]	ResNeXt152-FPN	46.5	**55.8**	32.5
UPSNet [48]	DCN101-FPN	46.6	53.2	36.7
Unifying‡ [23]	DCN101-FPN	47.2	53.5	37.7
		box-free		
DeeperLab [49]	Xception-71	34.3	37.5	29.6
SSAP [10]	Res101-FPN	36.9	40.1	32.0
PCV [42]	Res50-FPN	37.7	40.7	33.1
Panoptic-DeepLab [6]	Xception-71	39.7	43.9	33.2
AdaptIS [40]	ResNeXt-101	42.8	53.2	36.7
Axial-DeepLab [43]	Axial-ResNet-L	43.6	48.9	35.6
Panoptic FCN	Res101-FPN	45.5	51.4	36.4
Panoptic FCN	DCN101-FPN	47.0	53.0	37.8
Panoptic FCN*	DCN101-FPN	47.1	53.2	37.8
Panoptic FCN*‡	DCN101-FPN	**47.5**	53.7	**38.2**

Table 12. Experiments on the Cityscape *val* set. All of our results are achieved with single scale input and no flipping. The simple enhanced version is marked with *.

Method	Backbone	PQ	PQ^{th}	PQ^{st}
		box-based		
Panoptic FPN [17]	Res101-FPN	58.1	52.0	62.5
AUNet [24]	Res101-FPN	59.0	54.8	62.1
UPSNet [48]	Res50-FPN	59.3	54.6	62.7
Seamless [36]	Res50-FPN	60.2	55.6	63.6
Unifying [23]	Res50-FPN	61.4	54.7	66.3
		box-free		
PCV [42]	Res50-FPN	54.2	47.8	58.9
DeeperLab [49]	Xception-71	56.5	-	-
SSAP [10]	Res50-FPN	58.4	50.6	-
AdaptIS [40]	Res50	59.0	**55.8**	61.3
Panoptic-DeepLab [6]	Res50	59.7	-	-
Panoptic FCN	Res50-FPN	59.6	52.1	65.1
Panoptic FCN*	Res50-FPN	**61.4**	54.8	**66.6**

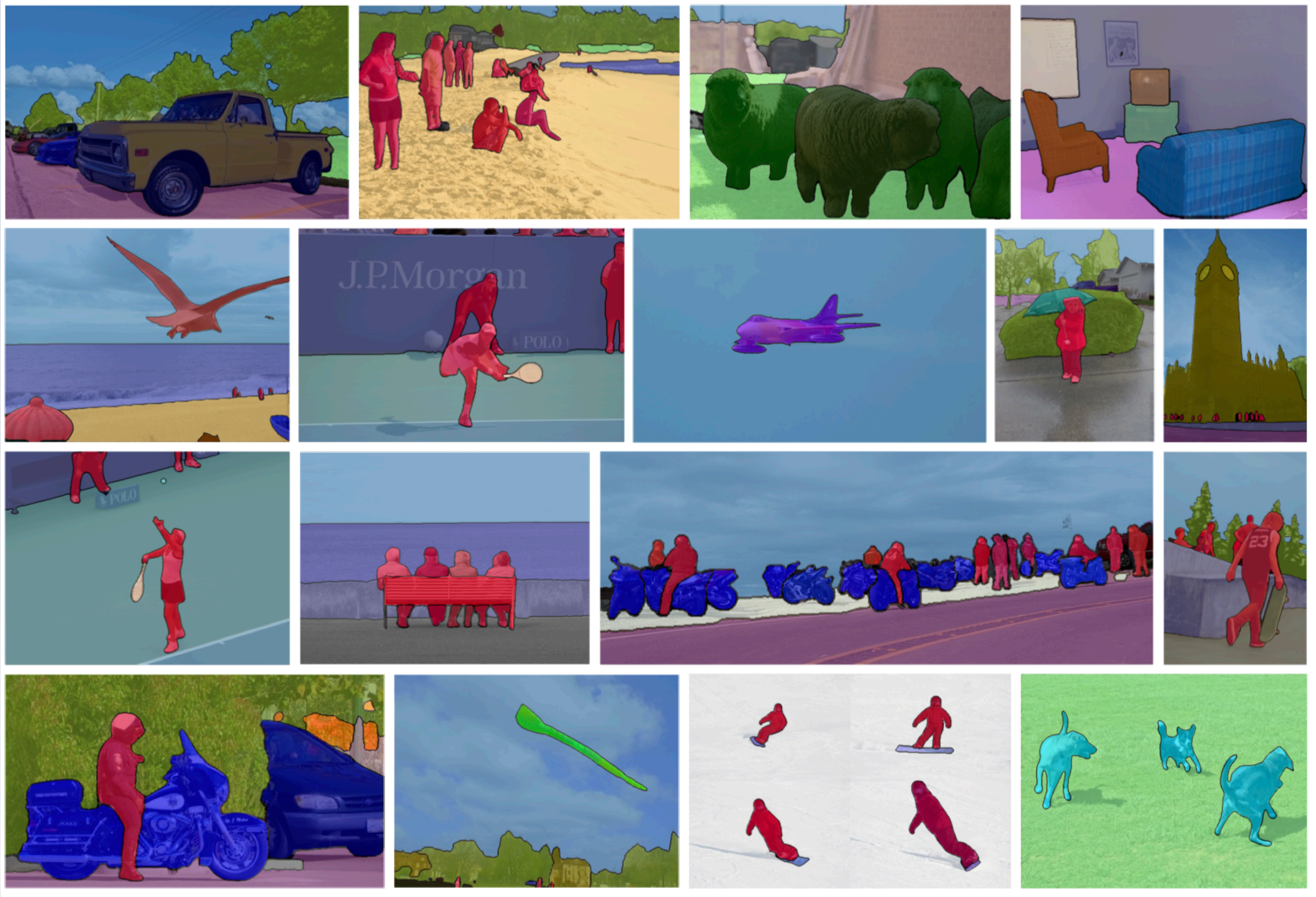
Table 13. Experiments on the Mapillary Vistas *val* set. All of our results are achieved with single scale input and no flipping. The simple enhanced version is marked with *.

Method	Backbone	PQ	PQ^{th}	PQ^{st}
		box-based		
TASCNet [21]	Res50-FPN	32.6	31.1	34.4
Seamless [36]	Res50-FPN	36.2	**33.6**	40.0
		box-free		
DeeperLab [49]	Xception-71	32.0	-	-
AdaptIS [40]	Res50	32.0	26.6	39.1
Panoptic-DeepLab [6]	Res50	33.3	-	-
Panoptic FCN	Res50-FPN	34.8	30.6	40.5
Panoptic FCN*	Res50-FPN	**36.9**	32.9	**42.3**

Results & Analysis

Visualization of Panoptic FCN

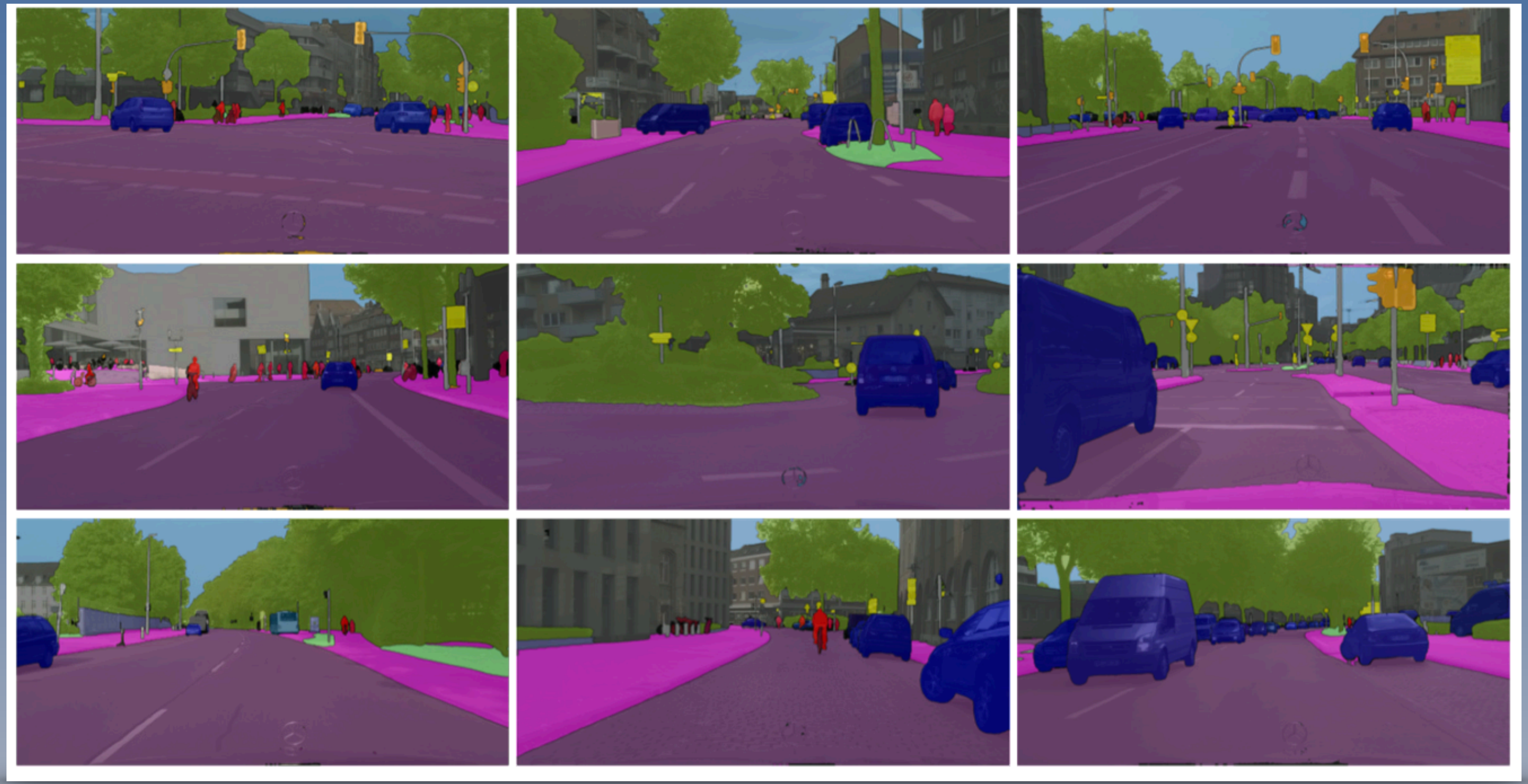
It achieve fine results on common context and traffic-related scenarios.



Results & Analysis

Visualization of Panoptic FCN

It achieve fine results on common context and traffic-related scenarios.



Future Work

More unified localization branch

For example, utilize center to represent Things and Stuff simultaneously.

Simplified panoptic generation

Currently, using argmax for panoptic generation brings 1.4% PQ drop.

More sparse kernel generation

More sparse kernel representation is needed to drop kernel generation.

Thanks

https://github.com/yanwei-li/PanopticFCN

ywli@cse.cuhk.edu.hk

Yanwei Li

CUHK