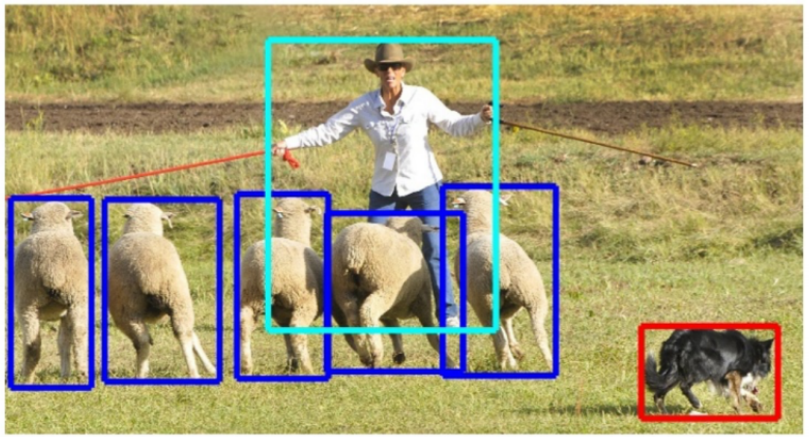
Panoptic Feature Pyramid Networks

Alexander Kirillov **Ross Girshick**
Kaiming He **Piotr Dollár**
Facebook AI Research (FAIR)

Classification, Detection and Segmentation

Classification

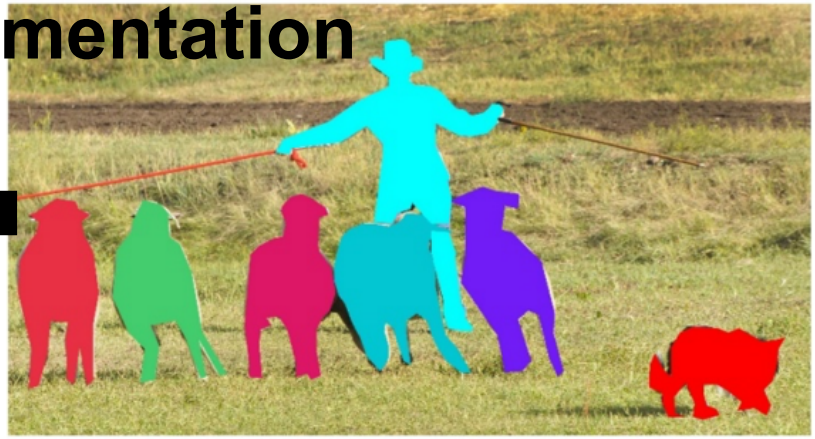


Object detection

Panoptic Segmentation

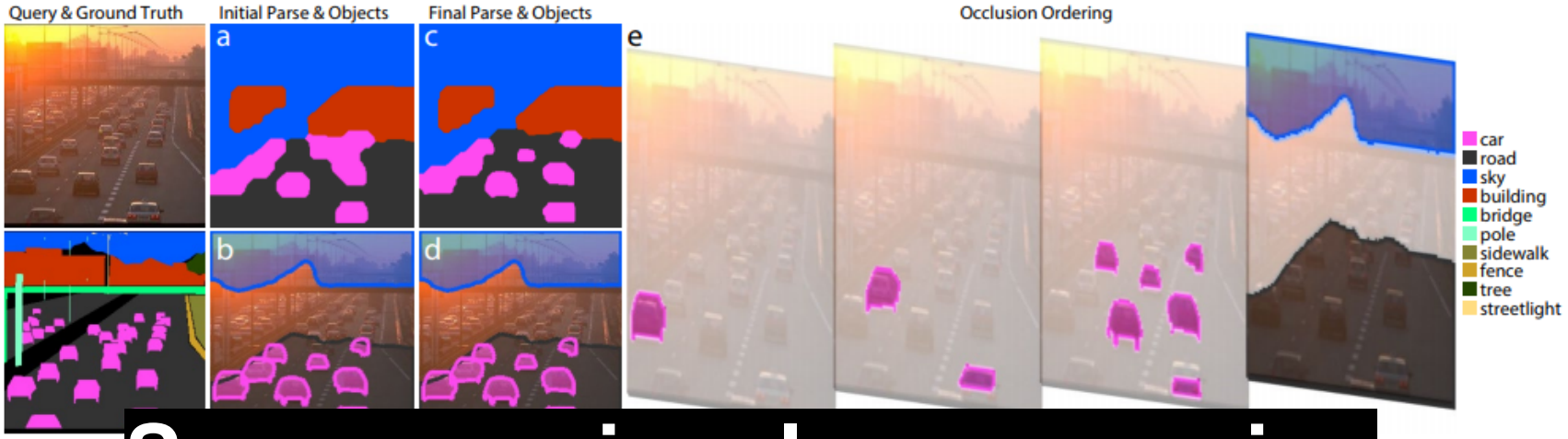
Semantic segmentation

+

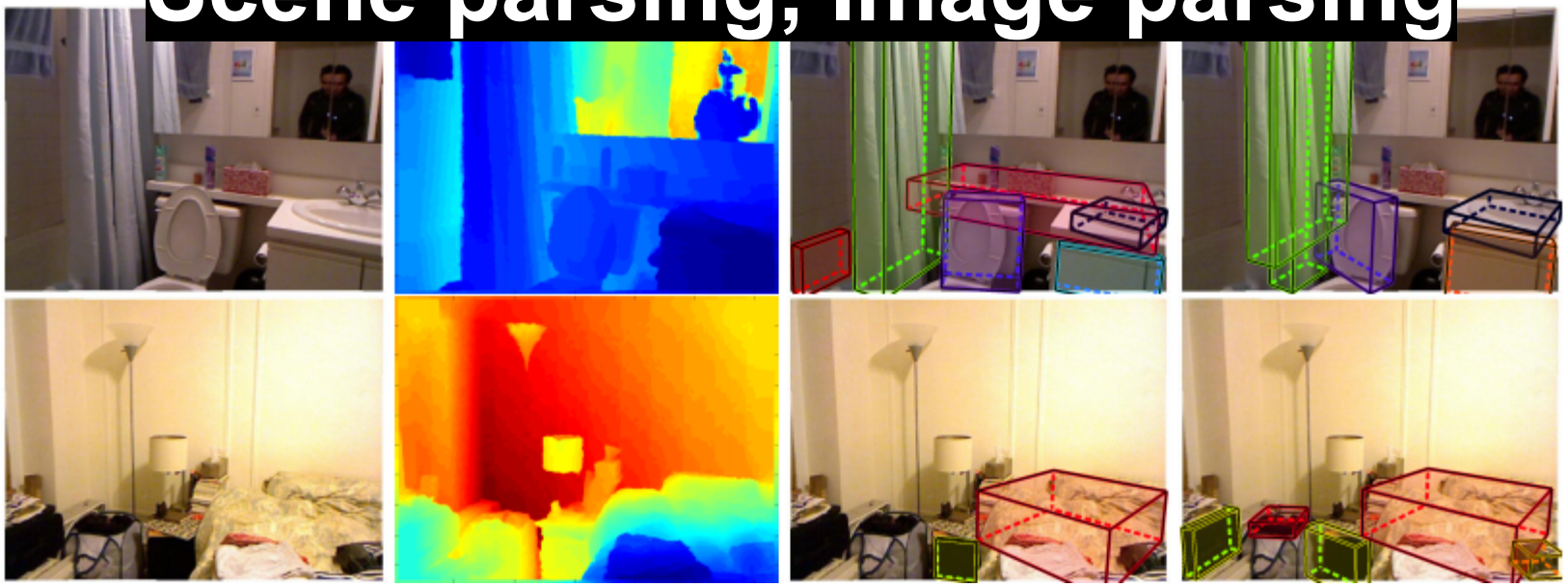


Instance segmentation

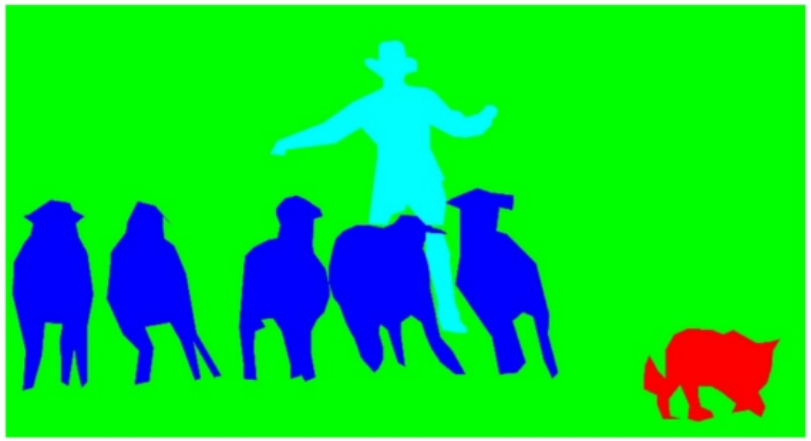
Panoptic Segmentation



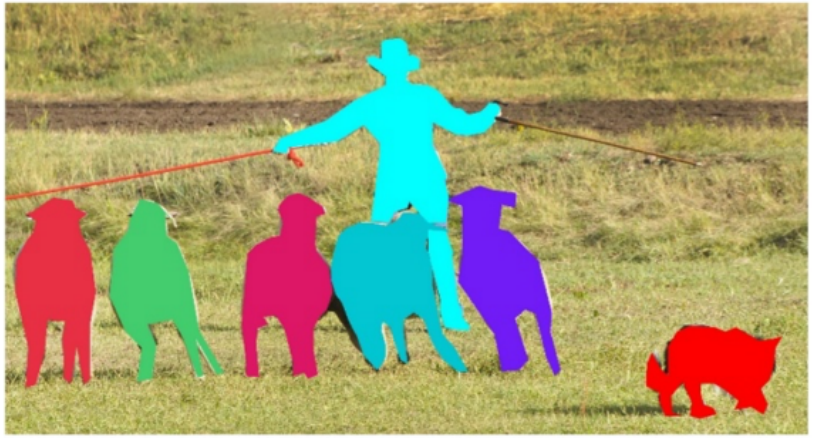
Scene parsing, Image parsing



Panoptic Segmentation



Semantic segmentation



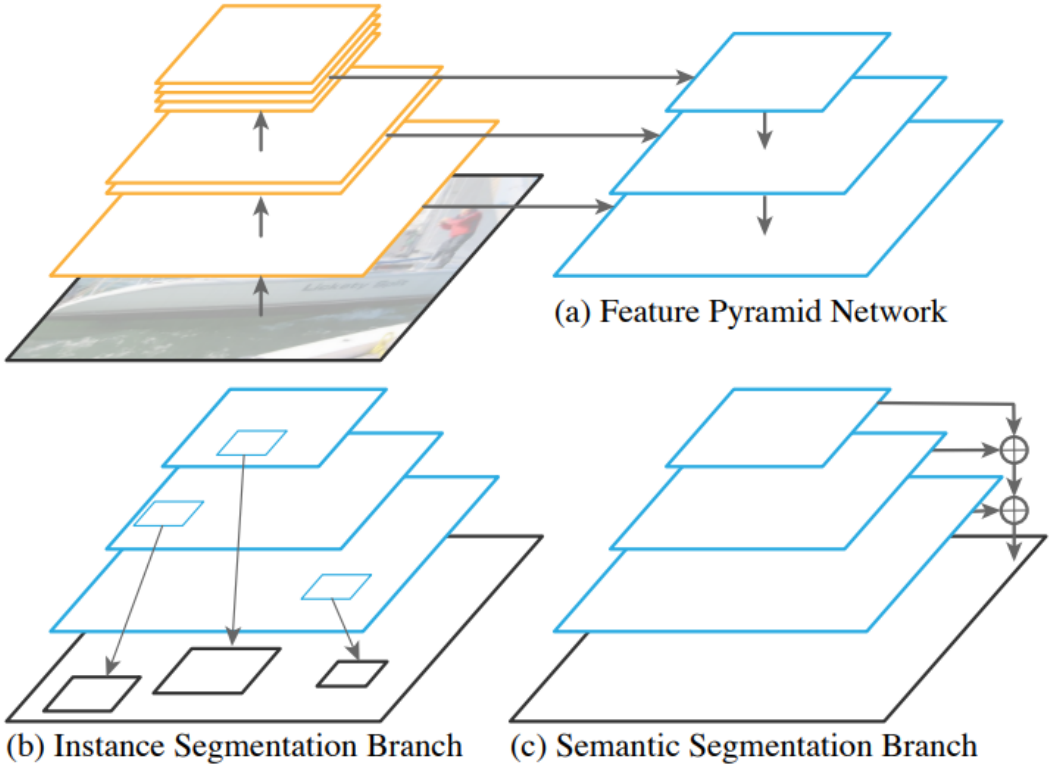
Instance segmentation

Overview of Panoptic FPN

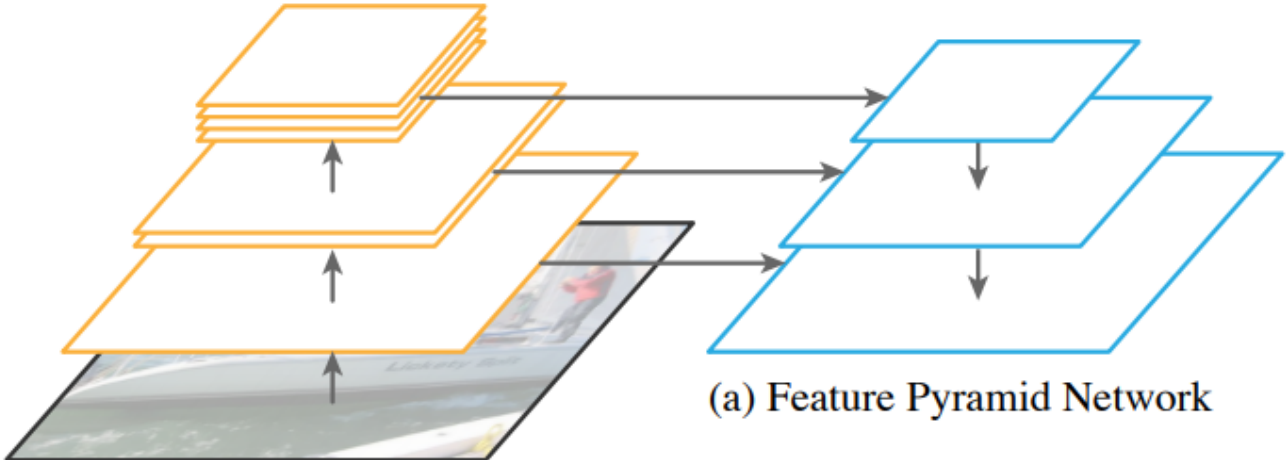
- Goal: to create a single framework for panoptic segmentation
- Endows Mask R-CNN with a shared Feature Pyramid Network (FPN) backbone
- Adds a lightweight dense-prediction branch for semantic segmentation
- Generates robust and accurate results for both instance and semantic segmentation

Architecture

- Panoptic FPN ~ Mask RCNN w/ FPN
  + semantic segmentation branch

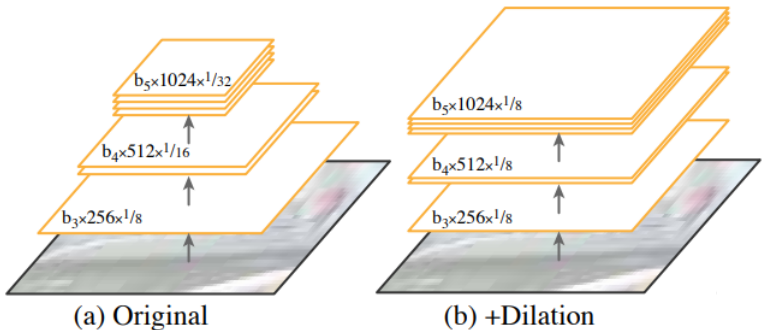


Architecture: FPN



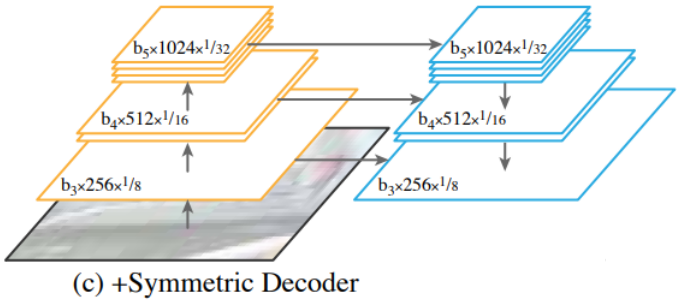
FPN takes a standard network with features at multiple spatial resolutions, and adds a light top-down pathway with lateral connections. FPN generates a pyramid, where each pyramid level has the same channel dimension.

Architecture: FPN

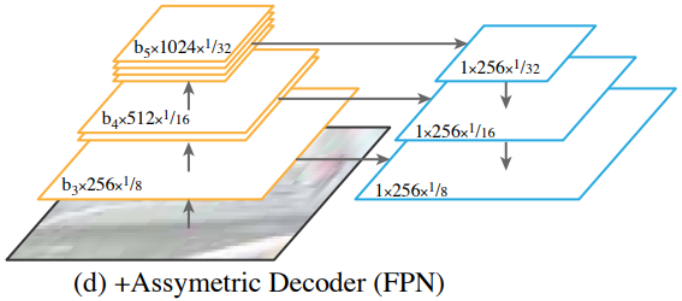


(a) Standard CNN

(b) Dilated convolution (popular for semantic segmentation)

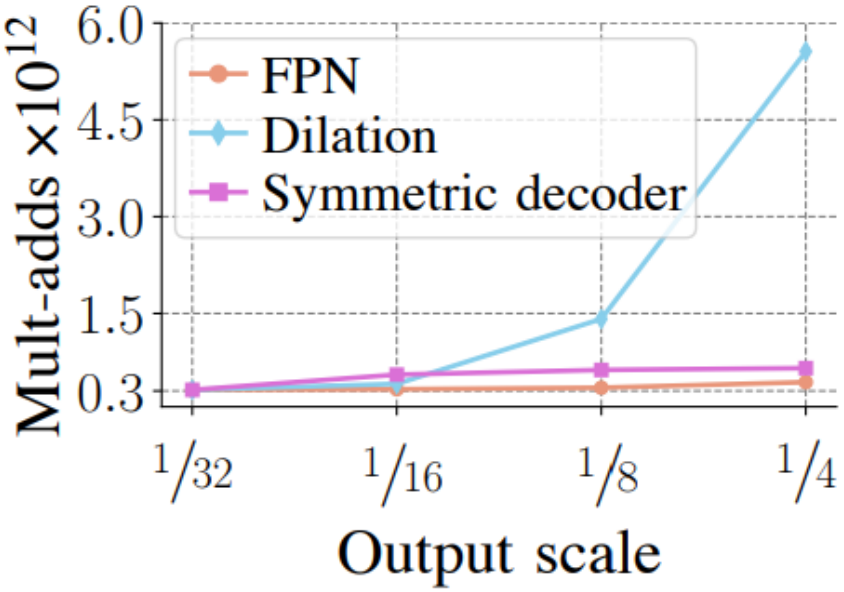


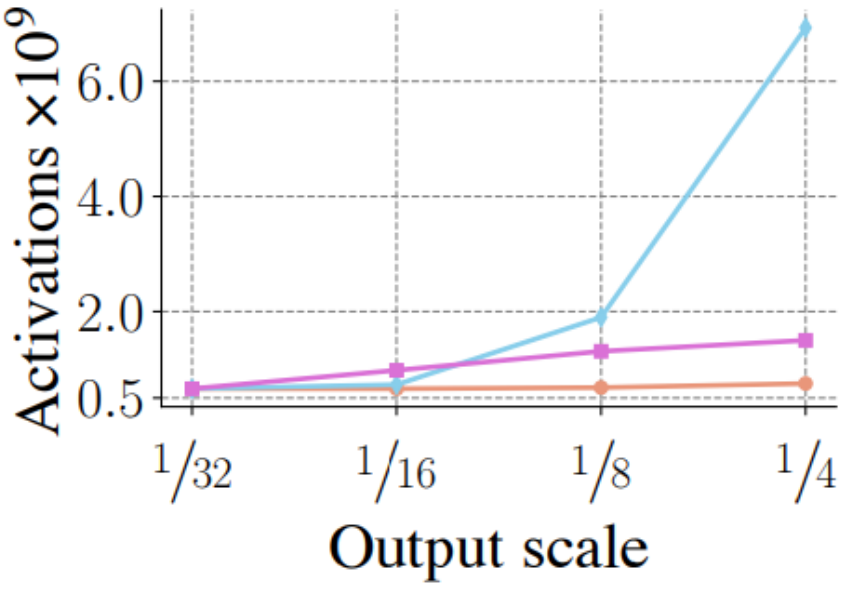
(c) U-Net (symmetric decoder)



(d) FPN (asymmetric decoder, same channel #, one block/stage)

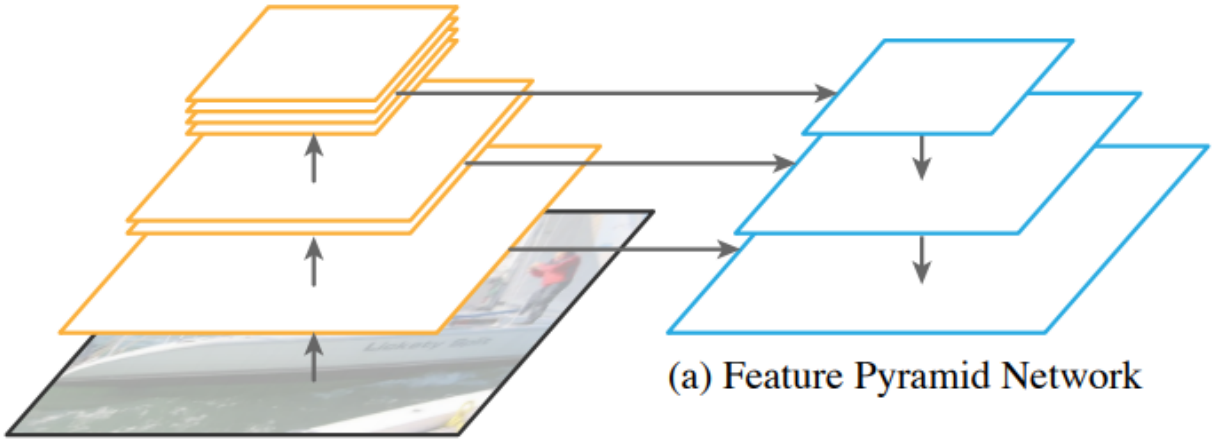
Architecture: FPN

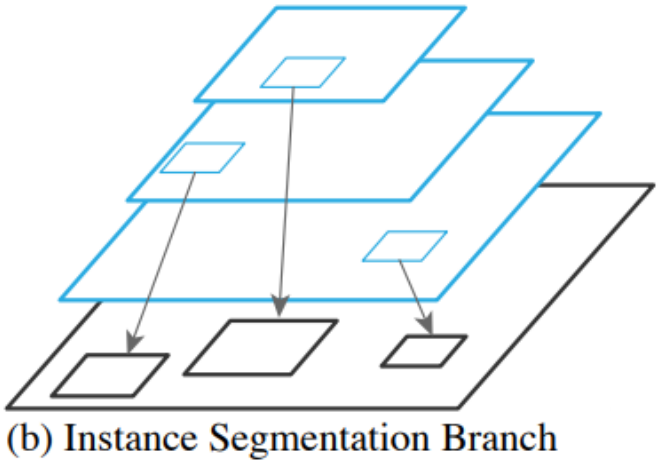




FPN at output scale 1/4 is similar computationally to dilation-16 (1/16 resolution output), but produces a 4× higher resolution output. Increasing resolution to 1/8 via dilation uses a further ~3× more compute and memory.

Architecture: Instance branch





Mask RCNN w/ FPN

Architecture: Semantic branch

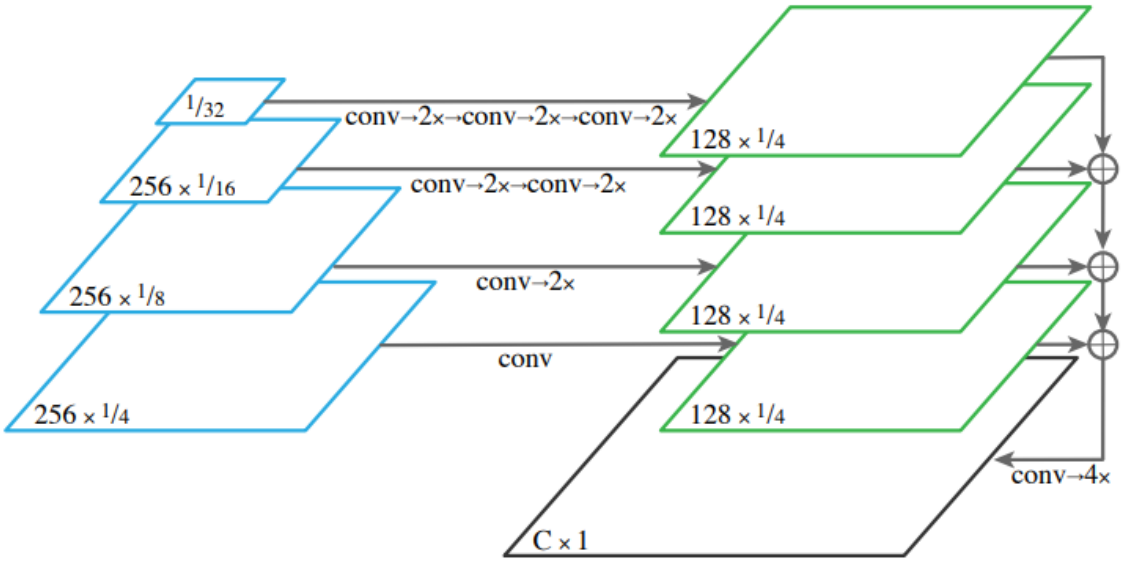


Figure 3: **Semantic segmentation branch.** Each FPN level (left) is upsampled by convolutions and bilinear upsampling until it reaches 1/4 scale (right), theses outputs are then summed and finally transformed into a pixel-wise output.

Joint Training

$$L = \lambda_{\mathrm{i}} \left(L_{\mathrm{c}} + L_{\mathrm{b}} + L_{\mathrm{m}} \right) + \lambda_{\mathrm{s}} L_{\mathrm{s}}$$

Lc: classification loss

Lb : bounding-box loss

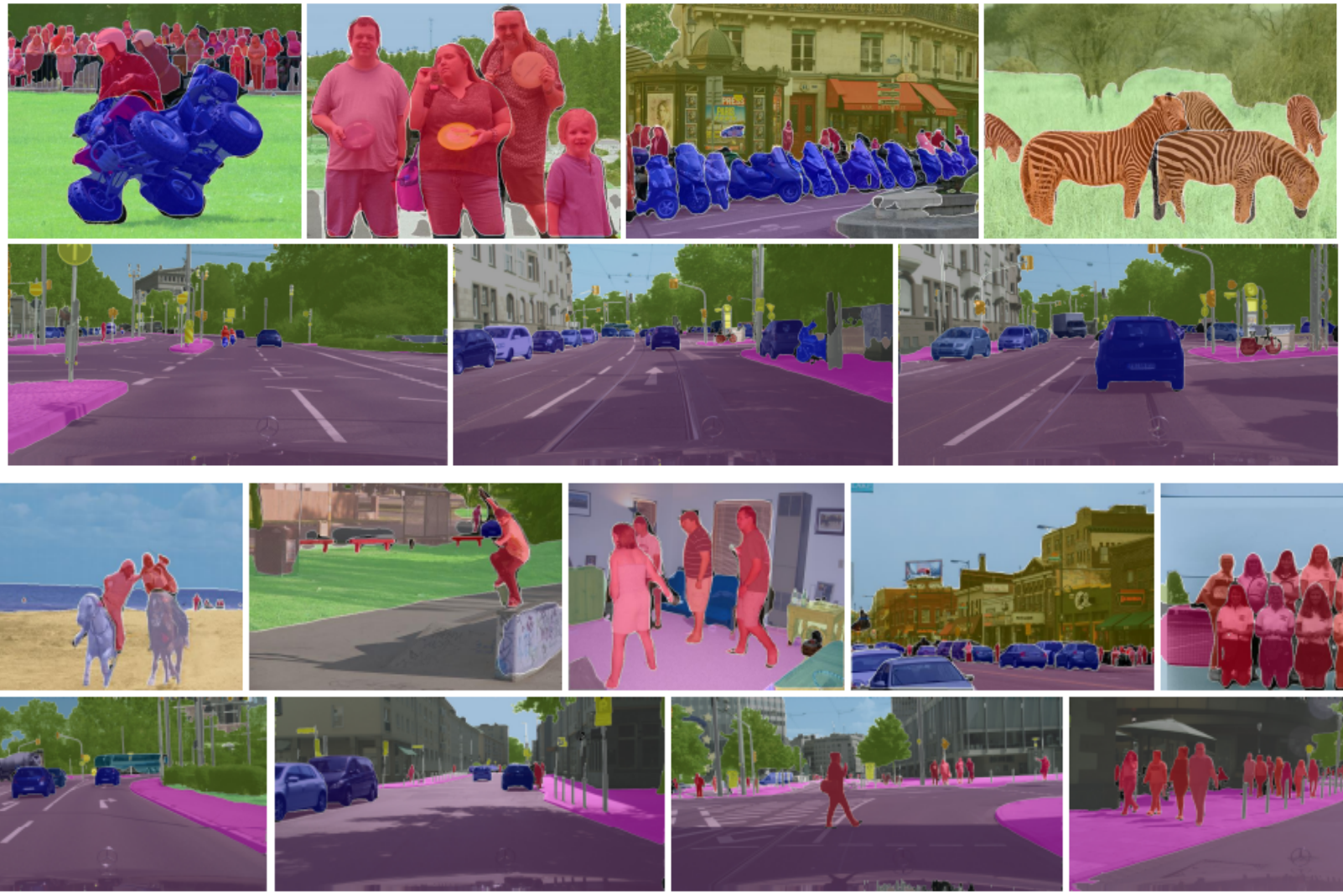
Lm: mask loss

Instance Segmentation

Ls: semantic loss

Semantic Segmentation

Results



Results

	PQ	PQ^{Th}	PQ^{St}
Artemis	16.9	16.8	17.0
LeChen	26.2	31.0	18.9
MPS-TU Eindhoven	27.2	29.6	23.4
MMAP-seg	32.1	38.9	22.0
Panoptic FPN	**40.9**	**48.3**	**29.7**

(a) **Panoptic Segmentation on COCO test-dev.** We submit Panoptic FPN to the COCO test-dev leaderboard (for details on competing entries, please see http://cocodataset.org/#panoptic-leaderboard). We only compare to entires that use a *single network* for the joint task. We do *not* compare to competition-level entires that utilize ensembling (including methods that ensemble separate networks for semantic and instance segmentation). For methods that use *one network* for panoptic segmentation, our approach improves PQ by an ~9 point margin.

	PQ	PQ^{Th}	PQ^{St}	mIoU	AP
DIN [1]	53.8	42.5	62.1	-	28.6
Panoptic FPN	**58.1**	**52.0**	**62.5**	**75.7**	**33.0**

(b) **Panoptic Segmentation on Cityscapes.** For Cityscapes, there is no public leaderboard for panoptic segmentation at this time. Instead, we compare on val to the recent work of Arnab and Torr [1] who develop a novel approach for panoptic segmentation, named DIN. DIN is representative of alternatives to region-based instance segmentation that start with a pixel-wise semantic segmentation and then perform grouping to extract instances (see the related work for a discussion). Panoptic FPN, without any bells and whistles, outperforms DIN by a 4.3 point PQ margin.

Table 4: **Comparisons of ResNet-101 Panoptic FPN to the state of the art.**

Results

	backbone	AP	PQ^{Th}	mIoU	PQ^{St}	PQ
COCO	R50-FPN×2	33.9	46.6	40.2	27.9	39.2
	R50-FPN	33.3	45.9	41.0	28.7	39.0
		-0.6	*-0.7*	*+0.8*	*+0.8*	*-0.2*
Cityscapes	R50-FPN×2	32.2	51.3	74.5	62.4	57.7
	R50-FPN	32.0	51.6	75.0	62.2	57.7
		-0.2	*+0.3*	*+0.5*	*-0.2*	*+0.0*

(a) **Panoptic Segmentation: Panoptic *R50-FPN vs.* R50-FPN×2**. *Using a single FPN network for solving both tasks simultaneously yields comparable accuracy to two independent FPN networks for instance and semantic segmentation, respectively, but with half the compute.*

	backbone	AP	PQ^{Th}	mIoU	PQ^{St}	PQ
COCO	R50-FPN×2	33.9	46.6	40.2	27.9	39.2
	R101-FPN	35.2	47.5	42.1	29.5	40.3
		+1.3	*+0.9*	*+1.9*	*+1.6*	*+1.1*
Cityscapes	R50-FPN×2	32.2	51.3	74.5	62.4	57.7
	R101-FPN	33.0	52.0	75.7	62.5	58.1
		+0.8	*+0.7*	*+1.3*	*+0.1*	*+0.4*

(b) **Panoptic Segmentation: Panoptic *R101-FPN vs.* R50-FPN×2**. *Given a roughly* equal *computational budget, a single FPN network for the panoptic task outperforms two independent FPN networks for instance and semantic segmentation by a healthy margin.*

Table 3: **Panoptic FPN Results.**

Results

	backbone	mIoU	FLOPs	memory
DeeplabV3 [11]	ResNet-101-D8	77.8	1.9	1.9
PSANet101 [57]	ResNet-101-D8	77.9	2.0	2.0
Mapillary [5]	WideResNet-38-D8	79.4	4.3	1.7
DeeplabV3+ [12]	X-71-D16	79.6	0.5	1.9
Semantic FPN	ResNet-101-FPN	77.7	0.5	0.8
Semantic FPN	ResNeXt-101-FPN	79.1	0.8	1.4

(a) **Cityscapes Semantic FPN.** Performance is reported on the *val* set and all methods use only fine Cityscapes annotations for training. The backbone notation includes the dilated resolution 'D' (note that [12] uses both dilation and an encoder-decoder backbone). All top-performing methods other than ours use dilation. FLOPs (multiply-adds $\times 10^{12}$) and memory (# activations $\times 10^{9}$) are approximate but informative. For these larger FPN models we use color and crop augmentation at train time. Our baseline is comparable to state-of-the-art methods in accuracy and efficiency.

	backbone	mIoU	fIoU
Vllab [13]	Stacked Hourglass	12.4	38.8
DeepLab VGG16 [10]	VGG-16	20.2	47.5
Oxford [4]	ResNeXt-101	24.1	50.6
G-RMI [18]	Inception ResNet v2	26.6	51.9
Semantic FPN	ResNeXt-152-FPN	28.8	55.7

(b) **COCO-Stuff 2017 Challenge results.** We submitted an early version of Semantic FPN to the 2017 COCO Stuff Segmentation Challenge held at ECCV (http://cocodataset.org/#stuff-2017). *Our entry won first place* without ensembling, and we outperformed competing methods by at least a 2 point margin on all reported metrics.

Table 1: **Semantic Segmentation using FPN.**

Conclusions

- The method starts with Mask R-CNN with FPN and adds to it a lightweight semantic segmentation branch for dense-pixel prediction.
- Given its effectiveness and conceptual simplicity, this method has the potential to serve as a strong baseline and aid future research in panoptic segmentation.