Embedding-Interaction based Panoptic Segmentation

Anonymous ICCV submission

Paper ID 8435

A. More Details of Experiments

A.1. Architecture of Model

The encoder of Embedding Interaction based Panoptic Segmentation (EINet) consists of an ImageNet-pretrained neural network. and uses atrous convolution in the semantic branch to extract dense features in the last block of the encoder. Similar to [1], we apply separate ASPP in the decoder of the semantic segmentation branch to capture the remote dependencies of pixels. In contrast, we make instance segmentation branch parameters lighter due to the assumption that embedding interactions enhance each other. Motivated by [1], we also introduce additional low-level features from the encoder to the decoders of the instance branch and semantic branch, but EINet has a different approach in aggregating low-level features and features from the decoder. The details of encoder and decoders in EINet is shown in Fig.1.

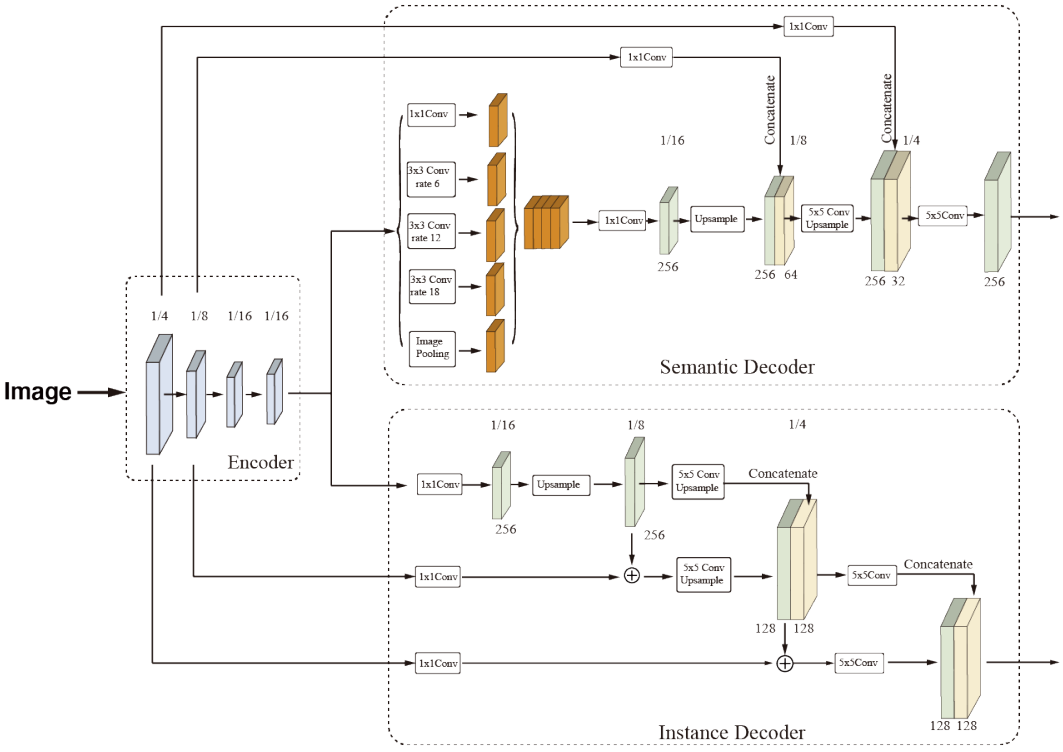

Figure 1. Details of encoder and decoders in EINet. 5×5conv upsample: single 5×5 depthwise-separable convolution[3].

A.2. Initialization of Semantic Center in Dictionary

As defined in the paper, V is the average embedding of the semantic and Q is the dictionary of the center embedding of the semantics. Motivated by BatchNormalization, we use a moving average approach to initialize the semantic center embedding Q with the semantic average embedding V computed from the current input. And we enable semantic center embedding to cover the entire dataset in this way. We use semantic averaging embedding V to initialize the semantic center embedding Q in the dictionary by moving average with the following equation. More concretely, at training time step t, the semantic center embedding Q is initialized as follows:

$$Q_j^{(t)} = Q_j^{(t-1)} \times momentum + V_j^{(t)} \times (1 - momentum), \tag{1}$$

where $j \in J$, and J is semantic category set of current input at training time step t.

$momentum$	PQ(%)	PQ^{Th}(%)	PQ^{St}(%)	AP(%)	mIOU(%)
0.6	62.5	**57.0**	66.5	34.5	79.6
0.8	62.7	55.9	**67.7**	34.4	79.7
0.9	**63.0**	56.9	67.4	**35.2**	**79.9**
0.99	62.8	58.0	66.3	34.8	79.8
0.999	62.2	56.3	66.5	34.2	79.4
0.9999	61.1	53.4	66.7	32.5	78.8

Table 1. Analysis of semantic center momentum on Cityscapes val dataset.

It can be seen from Table 1 that, the best performance (63.0%PQ, 35.2%AP and 79.9%mIOU.) was achieved when value (*e.g.* 0.9) was set to λ to ensure stable convergence.

A.2. Comparison with Different Centroid Heatmap

Object-as-point[7] proposed a class-aware multi-channel scale-adaptive instance representation that allows the kernel-size of the instance heatmap to vary with the size of the instance. Therefore, we tried scale adaptive kernel-size and scale fixed kernel-size.

AdaptiveKernel	MSE	PQ(%)	AP(%)	mIOU(%)
		61.8	33.4	79.1
	✓	62.4	34.5	79.4
✓		62.0	32.8	79.5
✓	✓	**63.0**	**35.2**	**79.9**

Table 2. Analysis of centroid heatmap regression on Cityscapes val dataset. **AdaptiveKernel**: Scale adaptive gaussian radius for centroid heatmap (instead of a fixed gaussian radius). **MSE**: MSE loss for centroid heatmap (instead of Focal loss[5]).

Compared to a centroid heatmap with a fixed Gaussian kernel-size, the use of a heatmap with scale adaptive Gaussian kernel-size improved PQ, AP and mIOU by 0.2%, 0.4% and 0.4% respectively, and in the case of multiple channels, the simple MSE loss can bring 1.0%PQ and 1.4%AP further improvements over Focal-loss [5], as shown in Tab. 2.

A.3. Comparison with Difference Score of Instance

We also evaluated the results of instance segmentation with different confidence scores. When using $Score(Objectness)$, we achieved 29.6% AP. When using only $Score(SemanticClass)$, we achieved 34.7%AP. And when using $Score(Objectness) \times Score(SemanticClass)$ as the score of the instance mask to rank the instance masks, we got the best results. It is worth noting that overlapping predictions of instance masks are not generated in the bottom-up framework, so choosing different confidence scores has no effect on mIoU and PQ.

Instance Score	PQ(%)	AP(%)	mIOU(%)
$Score(objectness)$	63.0	29.6	79.9
$Score(SemanticClass)$	63.0	34.7	79.9
$Score(objectness) \times Score(SemanticClass)$	63.0	**35.2**	79.9

Table 3. Comparison on using different confidence scores to rank instance masks. Note the choice of confidence scores only affects AP.

B. More Qualitative Results

B.1. More Visualization of Embedding with TSNE

We also visualized the semantic embedding[1] of the images in the Cityscapes dataset[2], as shown in Fig.2. As can be seen in Fig.2, the distribution of pixels of different semantic categories changes very significantly after using the interaction mechanism of embedding. The use of instance embedding to enhance the semantic embedding allows embedding of pixels belonging to *stuff* on semantic categories to be further away from embedding of pixels belonging to *thing*, and also allows pixels with the same semantic category to be clustered more compactly.

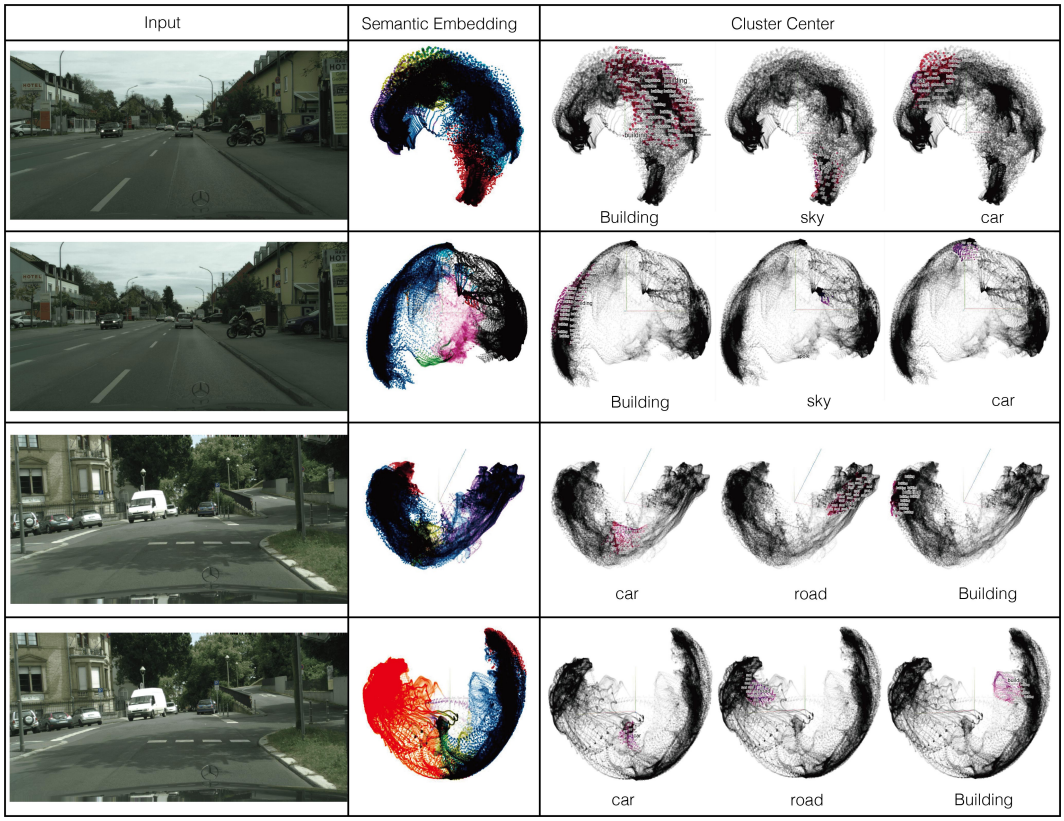

Figure 2. Visualization of Semantic Embedding on Cityscapes dataset. The **odd** rows are the results of Semantic Embedding **without interaction**, and the **even** rows are the results of **interaction-augmented** Semantic Embedding.

B.2. Visualization of Centroid Heatmap on Cityscapes

As shown in Tab.2, we obtained the best results when using the MSE loss to optimize the category-known and scale-adaptive centroid heatmap , and we also visualize the learned scale-adaptive centroid heatmap in Fig.3.

B.3. Comparison on Cityscapes

We provide more visualization results on Cityscapes to compare performance of PanopticDeeplab[1] and EINet with ResNet50 backbone in Fig.4.

B.4. Comparison on COCO

We also provide more visualization results on COCO to compare performance of Detectron2[6, 4], PanopticDeeplab[1] and EINet with ResNet50 backbone in Fig.5,6.

[1] Visualization of semantic embedding is obtained by the TSNE clustering algorithm in the tesorboard:https://projector.tensorflow.org/.

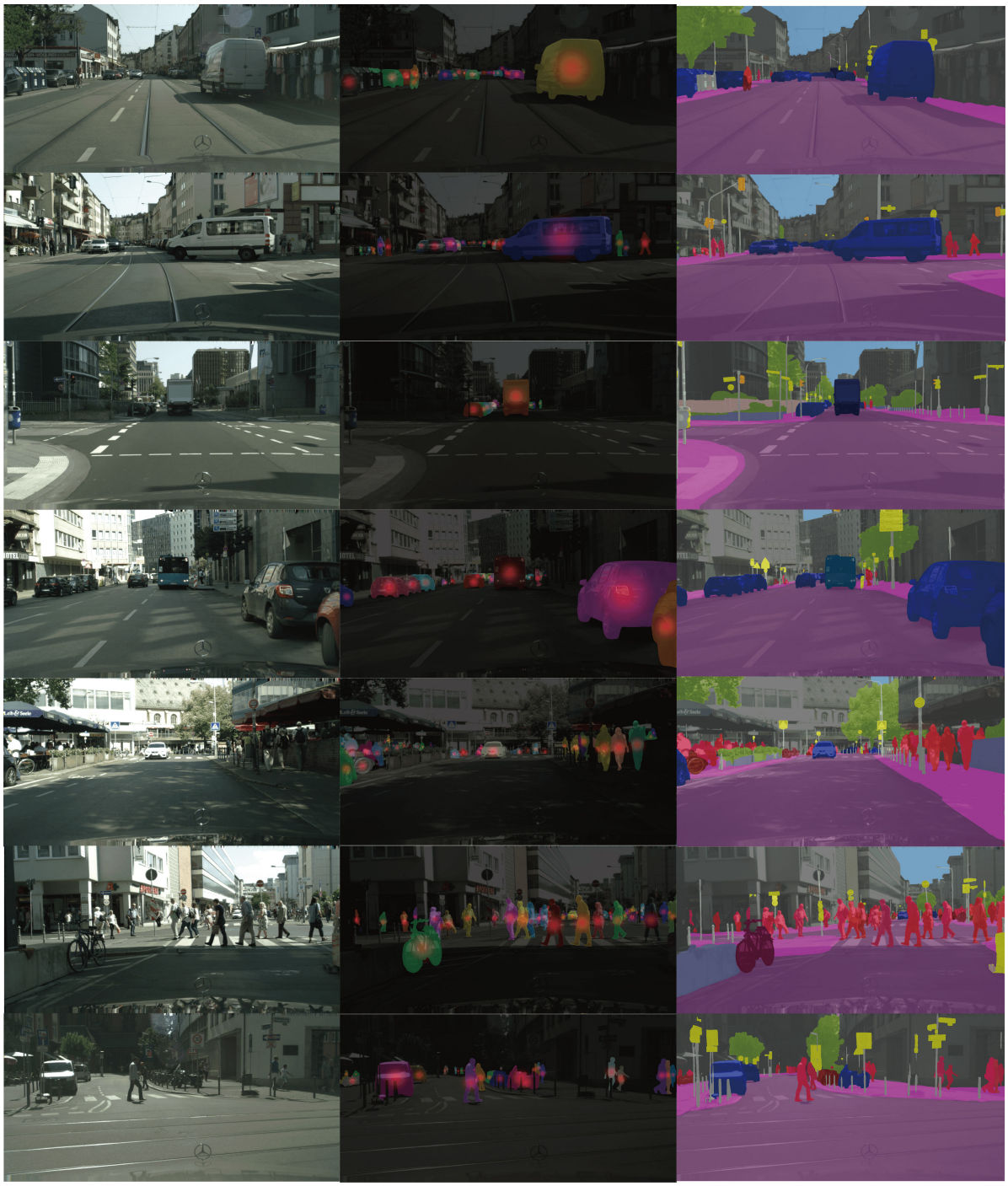

Figure 3. Visualization of panoptic segmentation results of EINet on Cityscapes val dataset with ResNet50. Left to right: input image, instance segmentation results with scale-adaptive centroid heatmap, panoptic segmentation results.

References

[1] 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 *CVPR2020*. 1, 3, 5, 6, 7

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on*

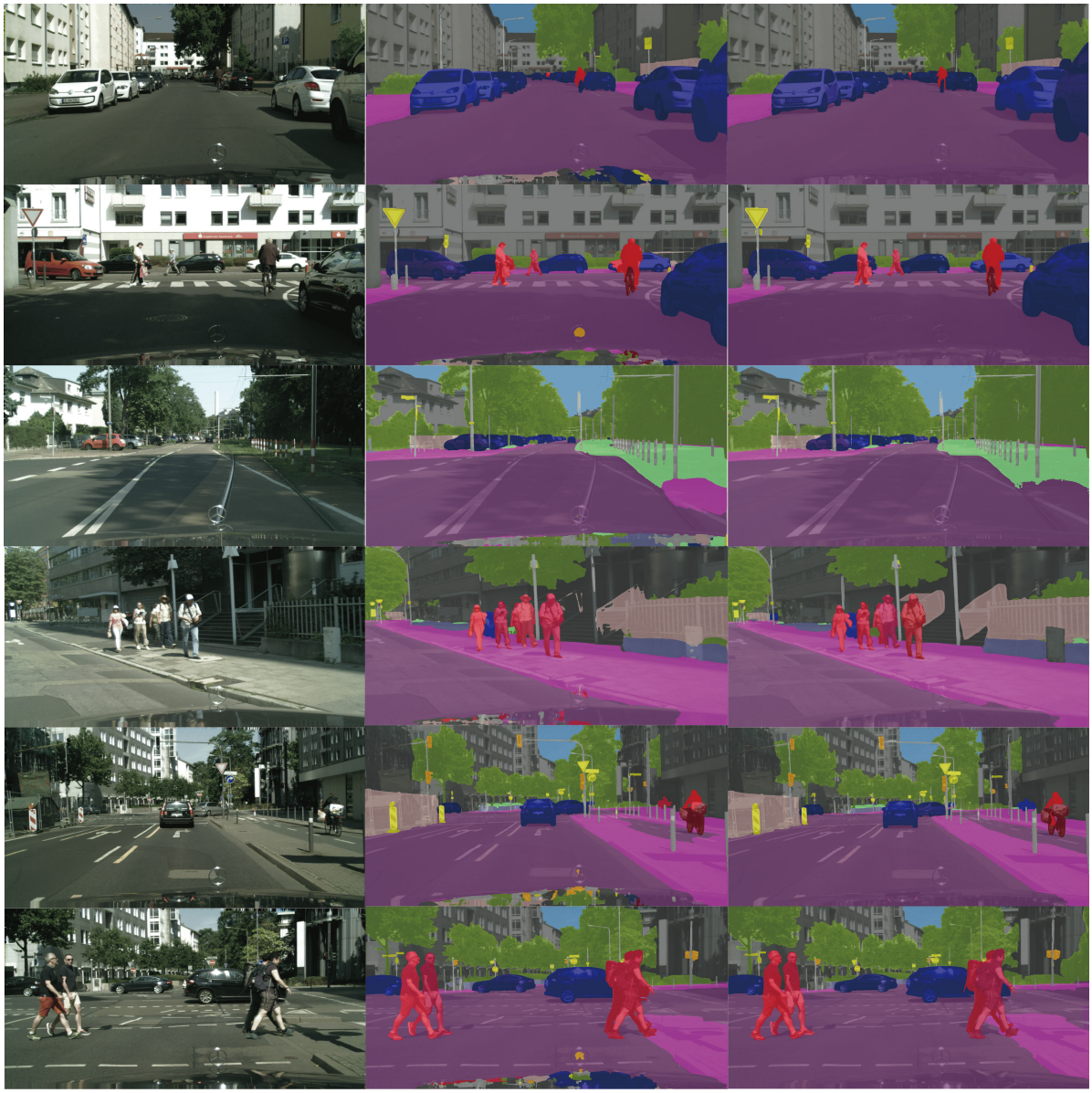

Figure 4. Visualization of bottom-up panoptic segmentation on Cityscapes val dataset with ResNet50. Left to right: input image, results by PanopticDeeplab[1], results by EINet(**ours**).

Computer Vision and Pattern Recognition (CVPR), June 2016. 3

[3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 1

[4] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 6, 7

[5] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[6] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3, 6, 7

[7] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019. 2

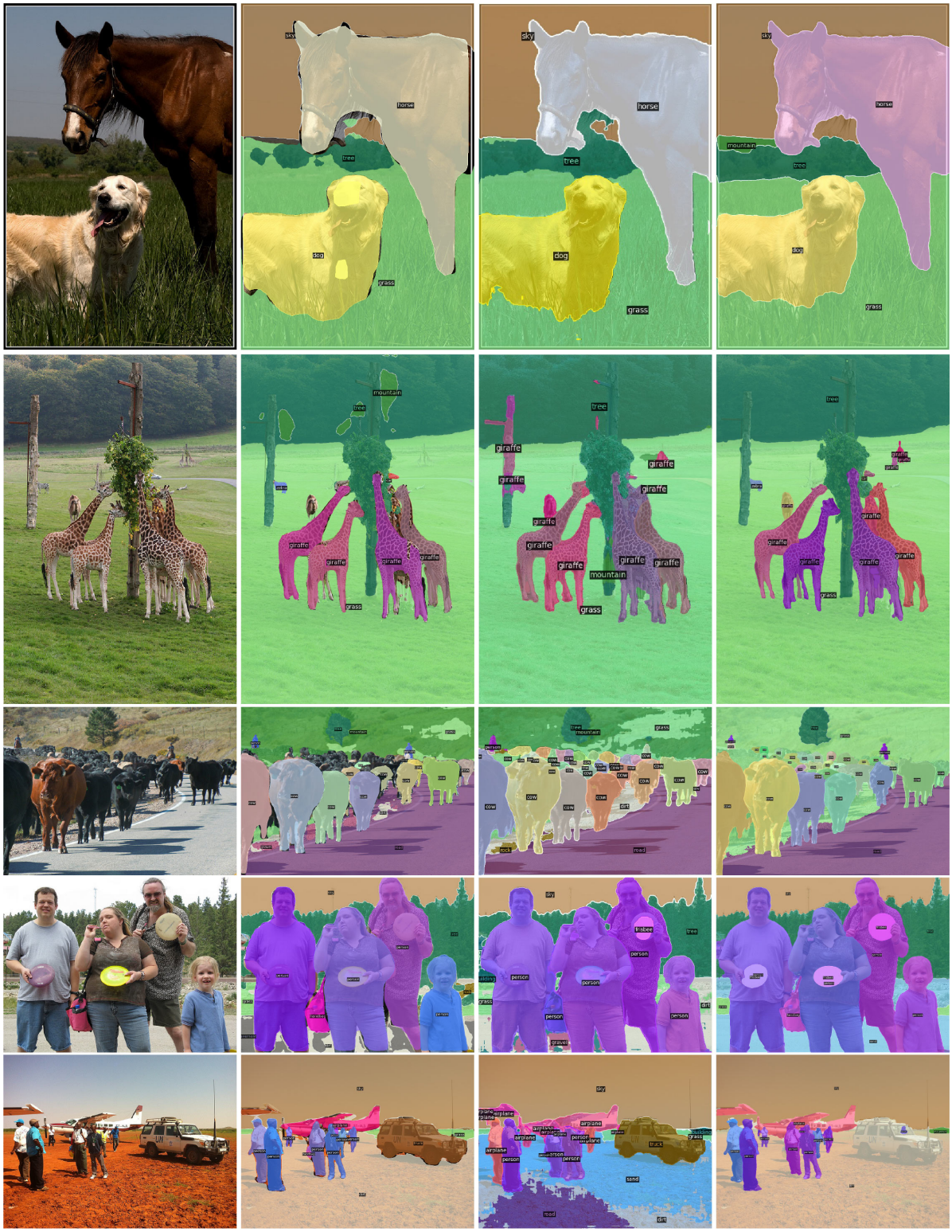

Figure 5. Visualization of panoptic segmentation on COCO with ResNet50. Left to right: input image, results by Detectron2[6, 4], results by PanopticDeeplab[1], ours results through EINet.

Figure 6. Visualization of panoptic segmentation on COCO with ResNet50. Left to right: input image, results by Detectron2[6, 4], results by PanopticDeeplab[1], ours results through EINet.