Learning Multi-modal Class-specific Tokens for Weakly Supervised Dense Object Localization

-Supplementary Material-

A. Implementation details

Classification training and dense localization inference. For the training of the proposed multi-modal token transformer, publicly available pre-trained weights[1] were used to initialize the ViT-base backbone. The decay parameter of the Global Weighted Ranking Pooling (GWRP) was set to 0.996, as suggested in [6]. We selected the model with the best classification performance on the validation set for the dense localization inference. The dense localization inference process is illustrated in Figure 1. For the generation of the transformer attention-driven localization maps, we fused the class-to-patch attention maps from the last six transformer encoding layers. The patch-level pairwise affinity maps were generated by fusing the patch-to-patch attention maps from all twelve transformer encoding layers. Following the prior works [5, 9, 13, 14], we reported the best evaluation results of the dense object localization maps when applying multiple background thresholds.

Segmentation training and inference. As in previous works [4, 7, 11, 12, 14], we used ResNet38-based Deeplab-V1 as the semantic segmentation network. We first trained the ResNet38-based classification network for 15 epochs using the image-level labels of the target segmentation dataset, following the same settings as [2]. We then used the classification weights to initialize the segmentation network. During training, we first randomly scaled the images within the range of (0.7,1.3) and randomly applied horizontal flipping on the scaled images. Moreover, we randomly cropped the processed images to be of size 321×321. The initial learning rate was set as 7×10^{-4}. A polynomial learning rate decay with a power of 0.9 was used. With the stochastic gradient descent (SGD) optimizer, we trained the segmentation model for 30 epochs with a batch size of 4. For inference, we used multi-scale inputs of scales 0.75, 1.0 and 1.5. Results were max-pooled and post-processed by the CRF with the default hyper-parameters suggested in [3]. The same settings were used on both PASCAL VOC 2012 and MS COCO 2014.

B. Additional quantitative results

Following prior works [4, 5, 8–10], we used IRN [1] to post-process our generated class-specific dense localization maps (seeds), generating the pseudo masks (masks) for WSSS. Table 1 shows the evaluation results of the seeds and masks generated by the state-of-the-art methods. The proposed method is seen to achieve the best mIoUs in terms of both seeds and pseudo masks. Table 2 and Table 3 present the detailed segmentation results of per class IoUs on PASCAL VOC 2012 and MS COCO 2014, respectively.

Table 1. Evaluation of the generated multi-label dense localization maps (seed) and their post-processed masks using IRN [1] for WSSS in terms of mIoU (%) on the PASCAL VOC 2012 *train* set.

Method	Cls. backbone	Seed	Mask
IRN (CVPR19) [1]	ResNet50	48.8	66.3
CONTA (NeurIPS20) [15]	ResNet50	48.8	67.9
RIB (NeurIPS21) [8]	ResNet50	56.5	70.6
AdvCAM (CVPR21) [9]	ResNet38	55.6	69.9
CDA (ICCV21) [11]	ResNet38	50.8	67.7
Du *et al.* (CVPR22) [5]	ResNet38	61.5	70.1
W-OoD (CVPR22) [10]	ResNet50	59.1	72.1
Ours	ViT-base	**66.3**	**73.7**

C. Additional qualitative results

Qualitative dense object localization results. Figure 3 and Figure 4 present the object localization heatmaps (where the colors from red to blue indicate the activation scores from high to low) generated by the proposed method on the multi-label PASCAL VOC 2012 and MS COCO 2014 *train* sets, respectively. Compared to the results by the state-of-the-art method, i.e., MCTformer [14], the proposed method is shown to produce more complete object localization maps, even for the challenging cases, such as the multiple TV monitors in a complex background in the last row of Figure 3 and the person in a far corner of the scene in the last row of Figure 4. Figure 5 presents the object localization heatmaps generated by the proposed method on the single-label OpenImages *test* set. Table 5 shows that the proposed method produces accurate dense object localization maps with clear boundaries, where the object regions are generally activated with high scores. These results demonstrate the effectiveness of the proposed method in performing dense object localization on both multi-label and single-label images.

Qualitative segmentation results. Figure 6 and Figure 7 present additional qualitative segmentation results on the PASCAL VOC 2012 and MS COCO 2014 *val* sets, respectively. Using the pseudo labels generated by the proposed method, the trained segmentation models are seen to perform well on both PASCAL VOC and MS COCO in vari-

[1] https://dl.fbaipublicfiles.com/dino/dino_vitbase16_pretrain/dino_vitbase16_pretrain.pth

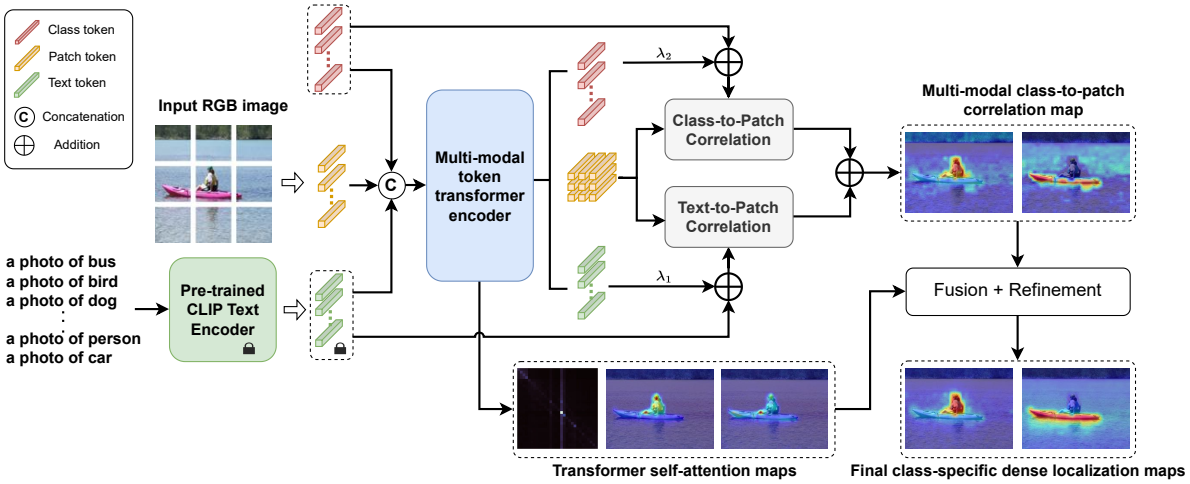

Figure 1. Class-specific dense localization inference of the proposed method.

Table 2. Segmentation performance comparison with the state-of-the-art WSSS methods using only image-level labels in terms of per-class segmentation IoUs (%) on PASCAL VOC 2012. * denotes without post-processing (*i.e.,* multi-scale testing and CRF).

	bkg	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIoU
Results on the *val* set:																						
AdvCAM [9]	90.0	79.8	34.1	82.6	63.3	70.5	**89.4**	76.0	87.3	31.4	81.3	33.1	82.5	80.8	74.0	72.9	50.3	82.3	42.2	74.1	52.9	68.1
Zhang *et al.* [16]	89.9	75.1	32.9	87.8	60.9	69.5	87.7	79.5	89.0	28.0	80.9	34.8	83.4	79.7	74.7	66.9	**56.5**	82.7	44.9	73.1	45.7	67.8
MCTformer [14]	91.9	78.3	39.5	**89.9**	55.9	76.7	81.8	79.0	90.7	32.6	87.1	57.2	87.0	84.6	77.4	79.2	55.1	**89.2**	47.2	70.4	58.8	71.9
W-OoD [10]	91.0	80.1	34.1	88.1	**64.8**	68.3	87.4	**84.4**	**89.8**	30.1	**87.8**	34.7	**87.5**	**85.9**	**79.8**	75.0	56.4	84.5	**47.8**	**80.4**	46.4	70.7
Ours*	91.0	78.5	38.2	85.3	58.8	73.3	84.7	80.5	85.4	26.4	79.8	56.0	79.3	77.6	73.7	78.5	47.8	82.6	44.0	68.4	56.8	68.9
Ours	**92.4**	**84.7**	**42.2**	85.5	64.1	**77.4**	86.6	82.2	88.7	**32.7**	83.8	**59.0**	82.4	80.9	76.1	**81.4**	48.0	88.2	46.4	70.2	**62.5**	**72.2**
Results on the *test* set:																						
AdvCAM [9]	90.1	81.2	33.6	80.4	52.4	66.6	87.1	80.5	87.2	28.9	80.1	38.5	84.0	83.0	79.5	71.9	47.5	80.8	59.1	65.4	49.7	68.0
Zhang *et al.* [16]	90.4	79.8	32.9	85.8	52.9	66.4	**87.2**	**81.4**	87.6	28.2	79.7	50.2	82.9	80.4	78.9	70.6	51.2	83.4	55.4	68.5	44.6	68.5
MCTformer [14]	92.3	84.4	37.2	82.8	60.0	72.8	78.0	79.0	89.4	31.7	**84.5**	**59.1**	**85.3**	**83.8**	79.2	81.0	53.9	85.3	60.5	65.7	57.7	71.6
W-OoD [10]	90.9	83.1	35.6	**89.0**	**61.5**	63.0	86.2	80.8	**89.9**	29.6	79.6	40.1	82.1	81.0	**82.6**	74.0	**60.1**	85.3	58.0	**71.9**	47.0	70.1
Ours*	91.2	82.8	36.6	81.5	54.6	70.2	81.6	79.2	83.9	31.2	78.0	57.8	80.8	81.6	77.6	78.4	50.4	80.9	57.3	58.5	58.7	69.2
Ours	**92.6**	**87.9**	**40.4**	86.2	56.7	**75.3**	82.5	81.1	85.9	**33.8**	82.4	57.8	84.6	83.1	80.0	**81.1**	52.8	**86.2**	**61.5**	61.1	**63.8**	**72.2**

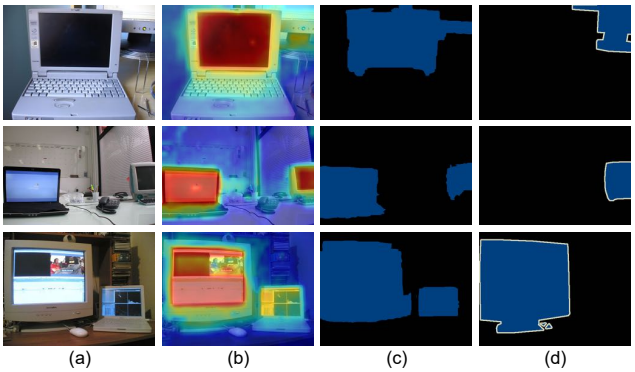

Figure 2. Failure cases. (a) Input. (b) The generated dense localization maps for the class "TV monitor". (c) The corresponding pseudo labels. (d) Ground-truth semantic segmentation labels.

ous challenging indoor and outdoor scenarios, such as small objects and complex backgrounds. This demonstrates the effectiveness and the good generalization ability of the proposed method in WSSS.

Limitation discussion. Our method has a limitation in accurately distinguishing between the target objects and some background objects that have very similar visual features. For example, as shown in Figure 2, the generated dense localization maps for the class "TV monitor" mistakenly include the regions of the laptop given that they are similar in shape and both have screens.

Table 3. Segmentation performance comparison with the state-of-the-art WSSS methods in terms of per class IoU(%) on the MS COCO 2014 *val* set. $*$ denotes without post-processing.

Class	MCTformer [14]	Our*	Ours	Class	MCTformer [14]	Our*	Ours
background	82.4	84.2	**85.3**	wine class	27.0	**35.3**	33.8
person	62.6	71.2	**72.9**	cup	29.0	33.6	**35.8**
bicycle	47.4	**50.8**	49.8	fork	13.9	18.7	**20.0**
car	**47.2**	47.0	43.8	knife	12.0	10.8	**12.6**
motorcycle	63.7	65.1	**66.2**	spoon	6.6	**6.7**	**6.7**
airplane	64.7	64.4	**69.2**	bowl	22.4	**24.0**	23.7
bus	64.5	67.0	**69.1**	banana	63.2	61.4	**64.4**
train	**64.5**	62.3	63.7	apple	44.4	45.4	**50.8**
truck	**44.8**	40.3	43.4	sandwich	39.7	44.3	**47.0**
boat	**42.3**	41.1	**42.3**	orange	63.0	60.6	**64.6**
traffic light	**49.9**	48.8	49.3	broccoli	**51.2**	50.5	50.6
fire hydrant	73.2	72.1	**74.9**	carrot	**40.0**	34.1	38.6
stop sign	76.6	72.6	**77.3**	hot dog	53.0	51.0	**54.0**
parking meter	64.4	62.9	**67.0**	pizza	62.2	63.3	**64.1**
bench	32.8	**35.2**	34.1	donut	55.7	55.4	**59.7**
bird	62.6	62.1	**63.1**	cake	47.9	47.1	**50.6**
cat	**78.2**	74.3	76.2	chair	22.8	**24.5**	**24.5**
dog	68.2	67.3	**70.6**	couch	35.0	37.7	**40.0**
horse	65.8	64.7	**67.1**	potted plant	13.5	**16.8**	13.0
sheep	70.1	68.0	**70.8**	bed	48.6	51.5	**53.7**
cow	68.3	67.0	**71.2**	dining table	12.9	**24.7**	19.2
elephant	81.6	79.9	**82.2**	toilet	63.1	64.6	**66.6**
bear	**80.1**	76.2	79.6	tv	47.9	47.6	**50.8**
zebra	**83.0**	82.0	82.8	laptop	49.5	52.8	**55.4**
giraffe	**76.9**	76.4	76.7	mouse	13.4	11.8	**14.4**
backpack	14.6	16.7	**17.5**	remote	41.9	40.7	**47.1**
umbrella	61.7	60.6	**66.9**	keyboard	49.8	53.3	**57.2**
handbag	4.5	**8.1**	5.8	cellphone	54.1	50.8	**54.9**
tie	25.2	29.3	**31.4**	microwave	38.0	42.7	**46.1**
suitcase	46.8	47.4	**51.4**	oven	29.9	**35.4**	35.3
frisbee	43.8	46.1	**54.1**	toaster	0.0	**4.3**	2.0
skis	12.8	11.6	**13.0**	sink	28.0	29.3	**36.1**
snowboard	**31.4**	26.6	30.3	refrigerator	40.1	50.2	**52.7**
sports ball	9.2	32.3	**36.1**	book	32.2	31.2	**34.8**
kite	26.3	36.2	**47.5**	clock	43.2	46.3	**51.5**
baseball bat	0.9	**7.2**	7.0	vase	22.6	24.4	**25.8**
baseball globe	0.7	8.5	**10.4**	scissors	**32.9**	29.9	30.7
skateboard	7.8	11.7	**15.2**	teddy bear	**61.9**	60.0	61.4
surfboard	46.5	45.2	**51.5**	hair drier	0.0	**1.8**	1.3
tennis racket	1.4	21.2	**26.4**	toothbrush	12.2	16.9	**19.0**
bottle	31.1	36.1	**37.1**	**mIoU**	42.0	43.7	**45.9**

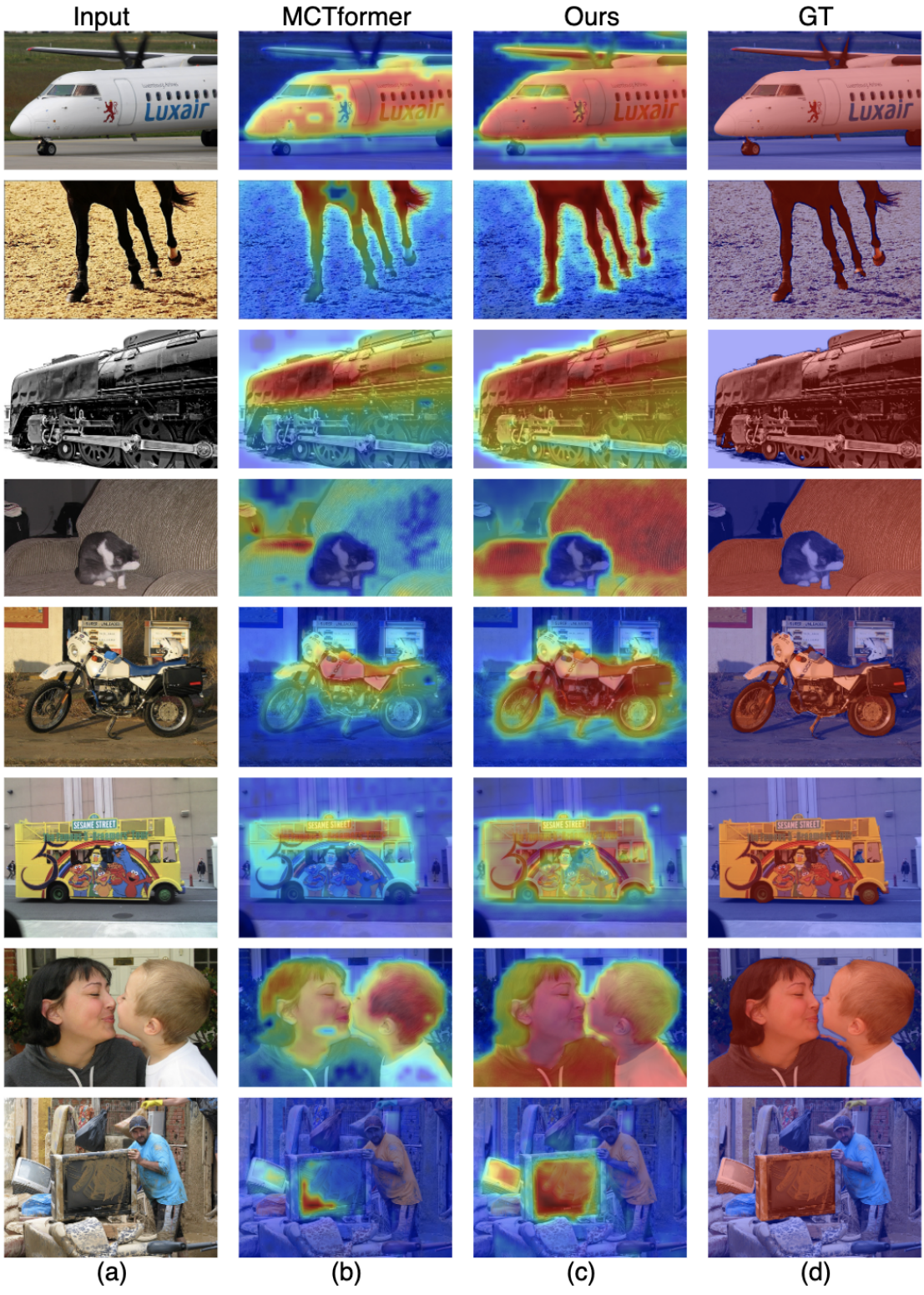

Figure 3. Visualization of the object localization heatmaps on the PASCAL VOC 2012 *train* set. (a) Input. (b) Results of MCTformer [14]. (c) Our results. (d) Ground-truth. (Only the localization heatmaps of the dominant class in each image are presented.)

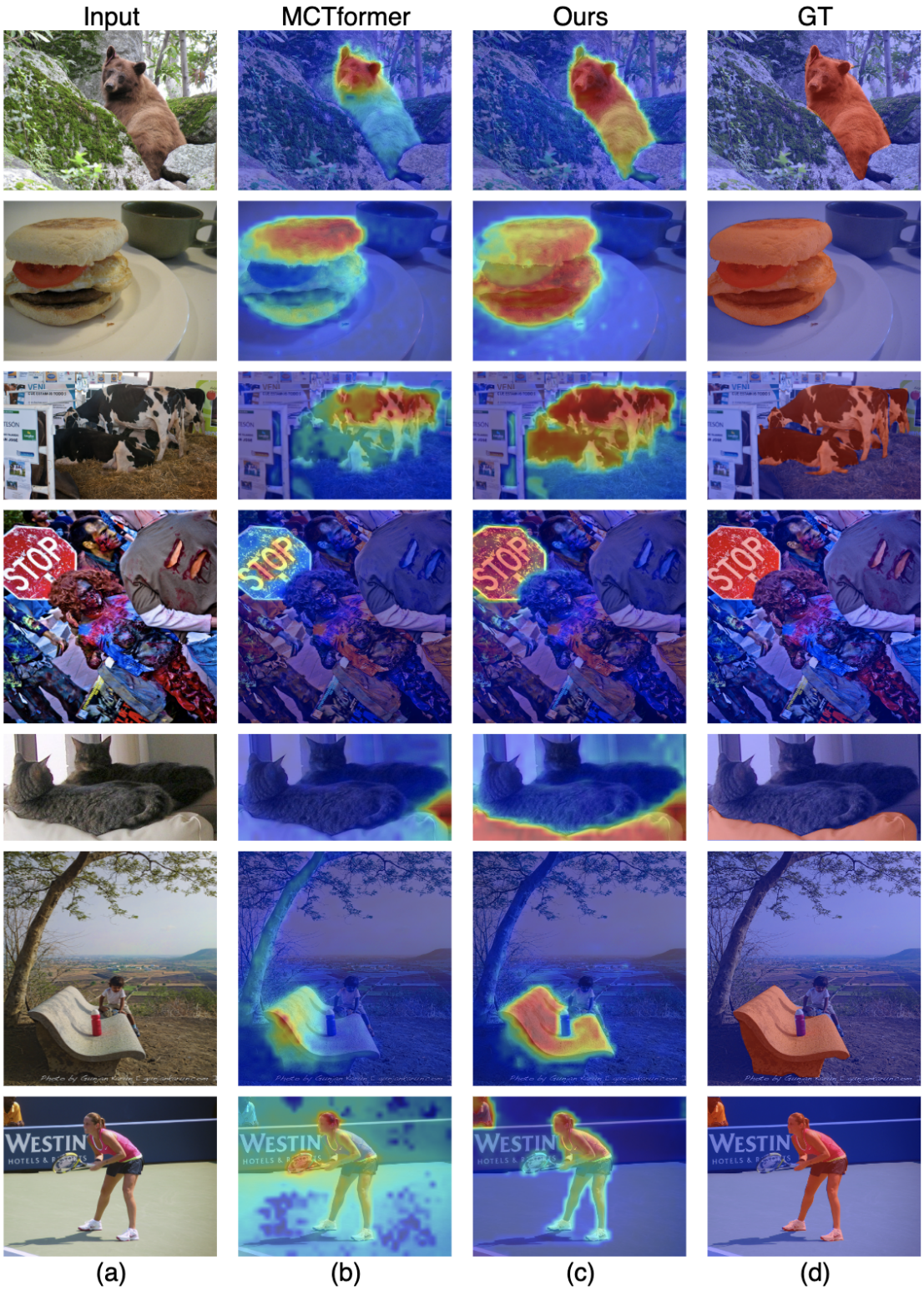

Figure 4. Visualization of the object localization heatmaps on the MS COCO 2014 *train* set. (a) Input. (b) Results of MCTformer [14]. (c) Our results. (d) Ground-truth. (Only the localization heatmaps of the dominant class in each image are presented.)

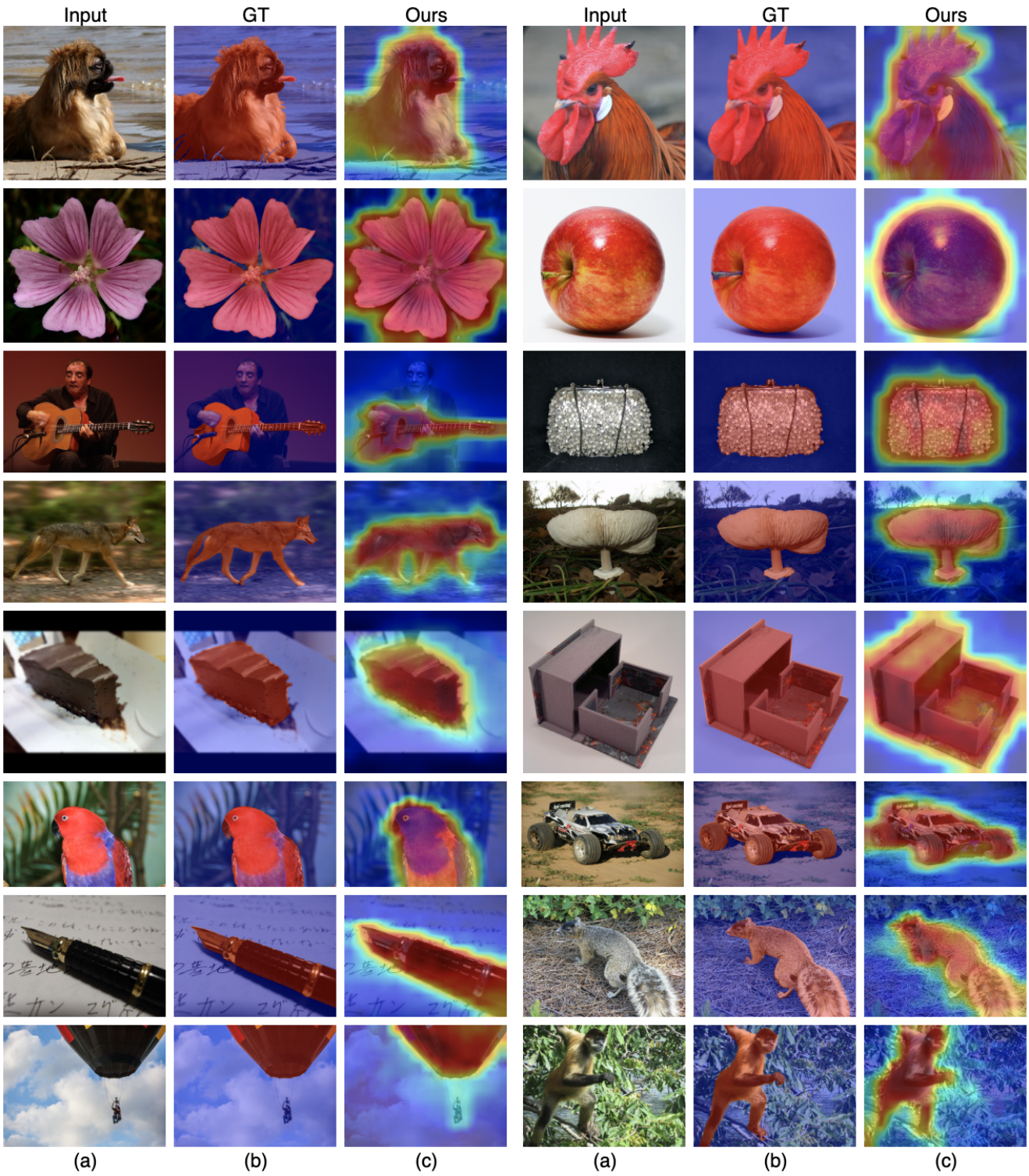

Figure 5. Visualization of the object localization heatmaps on the single-label OpenImages *test* set. (a) Input. (b) Ground-truth. (c) Our results.

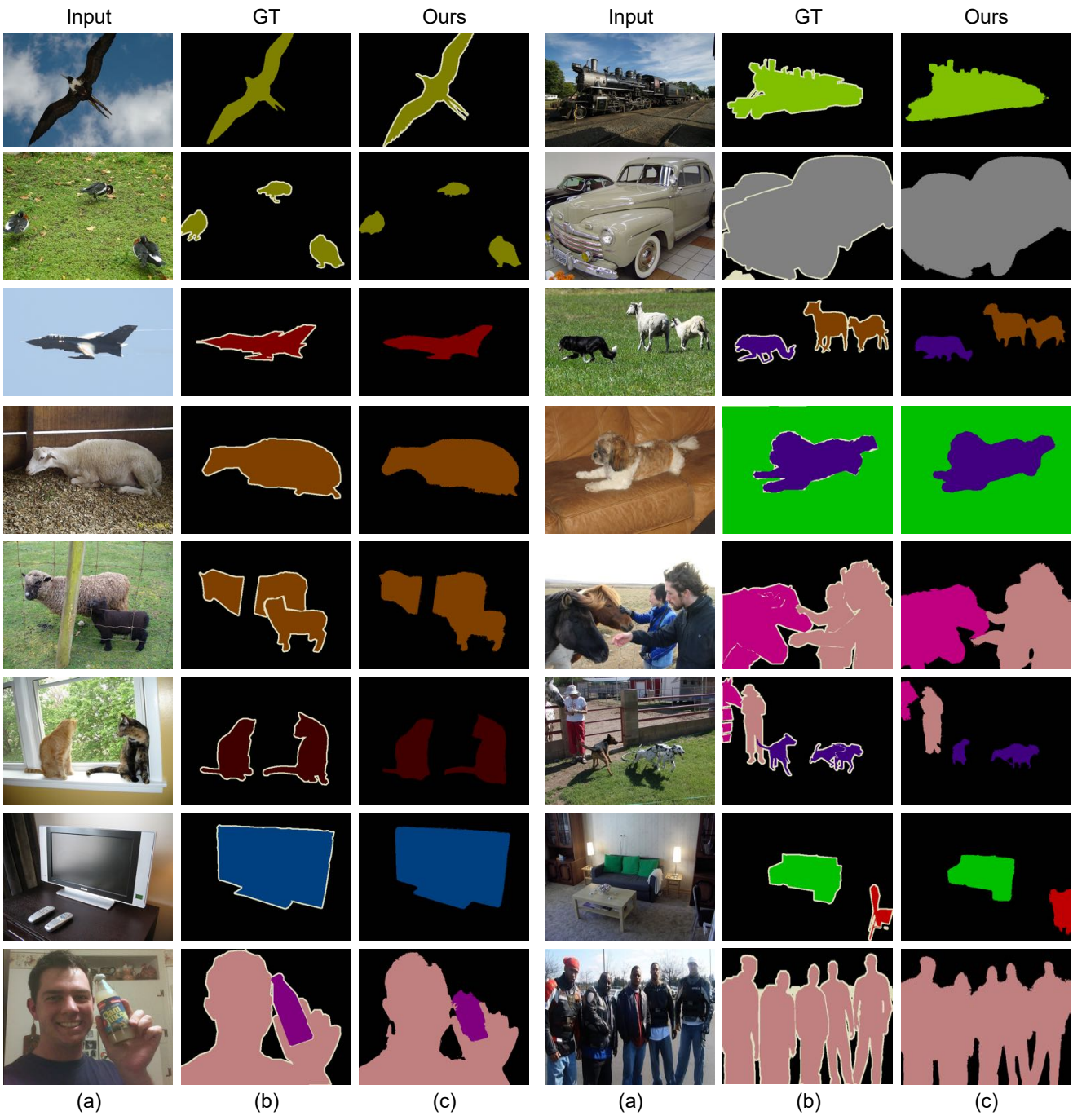

Figure 6. Visualization of the semantic segmentation results on the PASCAL VOC 2012 *val* set. (a) Input. (b) Ground-truth. (c) Our results.

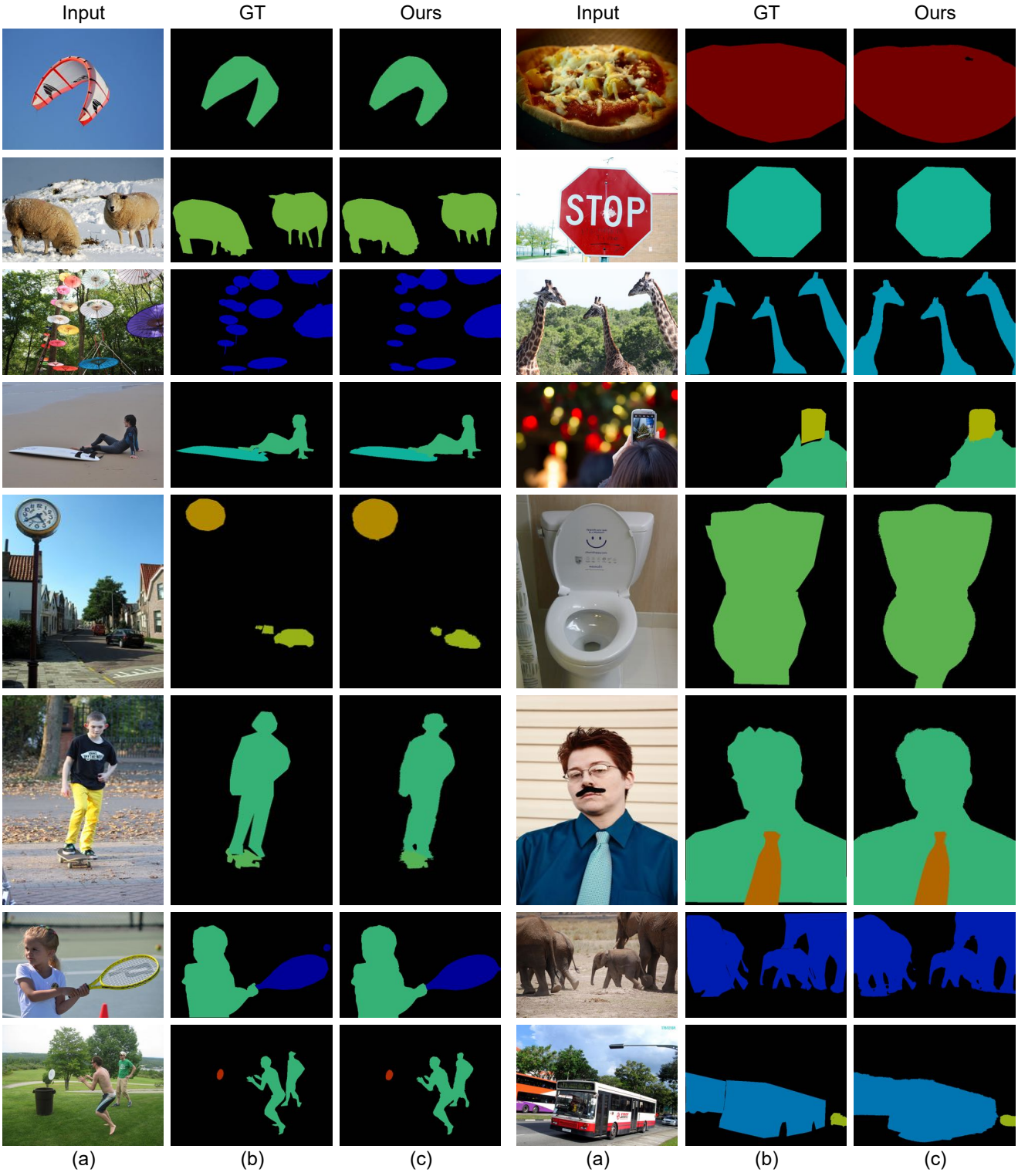

Figure 7. Visualization of the semantic segmentation results on the MS COCO 2014 *val* set. (a) Input. (b) Ground-truth. (c) Our results.

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019. 1

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 1

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 1

[4] Qi Chen, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *CVPR*, 2022. 1

[5] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *CVPR*, 2022. 1

[6] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 1

[7] Hyeokjun Kweon, Sung-Hoon Yoon, Hyeonseong Kim, Daehee Park, and Kuk-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In *ICCV*, 2021. 1

[8] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *NeurIPS*, 2021. 1

[9] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *CVPR*, 2021. 1, 2

[10] Jungbeom Lee, Seong Joon Oh, Sangdoo Yun, Junsuk Choe, Eunji Kim, and Sungroh Yoon. Weakly supervised semantic segmentation using out-of-distribution data. In *CVPR*, 2022. 1, 2

[11] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. In *ICCV*, 2021. 1

[12] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In *ICCV*, 2021. 1

[13] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 1

[14] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 5

[15] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In *NeurIPS*, 2020. 1

[16] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *ICCV*, 2021. 2