PiCIE: Unsupervised Semantic Segmentation using Invariance and Equivariance in Clustering – *Supplementary Materials*

Jang Hyun Cho[1] Utkarsh Mall[2] Kavita Bala[2] Bharath Hariharan[2]
[1]University of Texas at Austin [2]Cornell University

S1. More experiment details

Network architectures

For every method, we used the Feature Pyramid Network [11] to effectively encode representations from multiple scales. However, we only use pixel-wise randomly initialized linear (1×1 convolutional) layer for each level of the intermediate feature maps from ResNet-18 [5]. As noted in the main paper, we projected each of the feature maps to 128 dimensions instead of 256 from the FPN. After the linear projection, we directly bilinear-upsampled to $1/4$ scale of image resolution and element-wise summed to get the final $128 \times H \times W$ representation without the last 3×3 convolutional *smoothing* layers ($H = W = 80$ during training with 320×320 images). Note that this is a simplified version of the semantic segmentation branch of Panoptic FPN [8], a simple application of FPN to segmentation task. At the end, the only added parameters from ResNet-18 are 4 1×1 convolutional layers.

IIC For controlled experiments, we changed the network architecture of default IIC from the original shallow VGG-like model to FPN with ResNet-18 as described above. Following the original paper [6], we used auxiliary over-clustering loss: We kept the original $k = 45$ since the difference was minimal between $k \in \{45, 100, 250\}$. Also, the original IIC objective has a hyper-parameter λ which controls the "strictness" of the uniform distribution of clustering constraint. This could potentially alleviate the problem that IIC faces. In Table 1 we tested with $\lambda \in \{1, 1.25, 1.5, 1.75, 2, 3\}$ on COCO-All and $\lambda = 1$ performed the best, hence we kept $\lambda = 1$ in all our experiments. Similarly, we tested different learning rates $\eta \in \{0.1, 0.01, 0.001, 0.0001\}$ and $\eta = 0.0001$ was optimal. Both of these λ and η coincide with those in the original paper.

IIC-res12. We discovered that the shallow version of IIC performs better qualitatively on the (processed) COCO dataset [6]. This is because a shallow network tends to overfit to low-level visual signals such as color and texture due to its narrow receptive field. Since the dataset is pre-processed to reduce images that have too many pixels in the *things* categories, which are often visually more complex, perhaps the shallow IIC can be more effective for solving simple background segmentation compared to deep IIC. Therefore, we tested both versions. Note that the shallow VGG-like network used in the original IIC paper is unable to load ImageNet-pretrained weight, hence we instead used the first two residual layers *res1* and *res2* of ResNet-18 [5] as an alternative. They have nearly the same number of parameters and in the main paper, Table 3, we show that IIC with *res12* achieves similar accuracy on the original COCO-*Stuff* benchmark (27.7 and 27.92) [6]. Similar to IIC, we apply auxiliary over-clustering with $k = 45$.

Modified DC. Since DeepCluster [1] was originally designed for the task of image clustering, we modified the framework to fit the task of segmentation (pixel-wise classification). The network alternates between computing pseudo-labels and training. As mentioned in the main paper, the representation is pixel-level by removing the final pooling layer. This makes storing the feature vectors of the entire dataset infeasible, so we perform mini-batch k-means to first estimate cluster centroids, assign pseudo-labels, and train the network with the pseudo-labels. The same set of transformations as PiCIE is used on each image during training. Note that similar to IIC, image gradient is not concatenated in the input when initialized from ImageNet-weight. We do not apply over-clustering since the model without over-clustering performed the best compared to $k \in \{100, 250, 1000, 2500\}$.

Datasets

For training modified DC and PiCIE, we used simple pre-processing: resizing and center-crop to 320×320. For IIC, we used the original paper's pre-processing with their published code.

Transformations. For photometric transformations, we randomly applied *color jitter*, *gray scale*, and *Gaussian*

λ	1.0	1.25	1.5	1.75	2.0	3.0
Acc	21.8	17.6	16.8	15.6	16.4	16.6
mIoU	6.7	7.0	6.4	6.0	6.5	6.5

Table 1: IIC with different λ.

blur. Random jitter consists of jittering brightness, contrast, saturation, and hue. All jittering transformations are applied with probability $p = 0.8$ and control factors $0.3, 0.3, 0.3, 0.1$, respectively. Random gray scale is applied with probability of $p = 0.2$. Random Gaussian blur is applied with probability of $p = 0.5$ and radius randomly chosen: $\sigma \in [0.1, 2]$. For geometric transformations, we applied *random crop* and *random horizontal flip* with crop factor $r \in [0.5, 1]$ and flipping probability $p = 0.5$. In order to ensure that the same transformations are applied during clustering and training, we first sample transformations during clustering and store them in a list to re-use during training.These hyper-parameters are a standard choice adopted in many other works [4, 2, 3].

Training

Clustering. The cluster centroids are computed with mini-batch k-means with GPUs using the FAISS library [7]. The initial cluster centroids are computed with 50 batches with batch size of 128, then the centroids are updated every 20 iterations. For every other hyperparameters related to clustering, we followed Caron et al. [1]. Since this process is highly optimized, it takes about 20 minutes to prepare the pseudo-labels for training every epoch on the COCO dataset, which makes less than half for training the network in total compared to IIC using the published code.

Training details. We trained every method with 10 epochs when trained with ImageNet weight initialization, and 20 epochs when trained from scratch. For modified DC and PiCIE, we used ADAM optimizer with learning rate $\eta = 1 \times 10^{-3}$, $\beta = (0.9, 0.999)$ and weight decay 0. For IIC, their original hyperparameter setting was better, so we kept their setting ($\eta = 1 \times 10^{-4}$). For the transfer learning and supervised training experiments, we used $\eta = 1 \times 10^{-3}$, $\beta = (0.9, 0.999)$ and weight decay 0, consistent with the setting from the main experiments. For the final objective, we applied weighted cross-entropy loss with per-cluster weight is balanced with the size of each cluster. We simply average the *cross* and *within* losses.

Evaluation metric. For evaluating our model, we followed the evaluation metric from [6] with pixel accuracy after Hungarian-matching [10] the cluster assignments to the ground truth labels. We also report mean IoU to account for false positives and negatives. In Table 2 of the main paper, we compute the accuracy and mIoU from the same model trained on COCO-*All* ($K = 27$), but evaluated by only accounting for the labels in each partition. This can be done efficiently by computing the confusion matrix of the all classes $K = 27$ first and partitioning the matrix accordingly. In Table 3 of the main paper, we closely follow the experiment setting of [6]: the image resolution is 128×128, the images are pre-scaled and constant-padded, and $K = 15$ which means only *stuff* categories are considered for evaluation.

Visualizations

For producing consistent visualizations, we used majority vote for each obtained cluster. That is, we first assigned color values to each ground truth label and for each obtained set of clusters, we assign the color of the majority class. In the main paper, notice that we showed IIC-res12 for COCO and IIC for Cityscapes. We included the version that had better qualitative results. We hypothesize that since COCO was preprocessed to include more *stuff* categories, it is easier for the shallow network which overfits to low-level cues (e.g., color and texture) to segment images well since the majority of *stuff* instances are visually simple. For the nearest neighbor result, we first chose successful and failure results from the large set of randomly selected images (results below), picked a pixel coordinate of interest, and computed the nearest neighbor on the entire validation set of COCO-*All*. Then, we extracted the images that the neighbors belong to and visualized.

S2. More results

In this section, we show more qualitative results randomly chosen for both IIC and IIC-res12, as well as modified DC and PiCIE.

Robustness on Color and Geometric transformations

We show that PiCIE successfully learns photometric invariance and geometric equivariance by evaluating our model with test-time augmentation. We apply the same set of photometric transformations (color jitter, Gaussian blur, and greyscale) and geometric transformations (horizontal flip and random crop) and report the results in Table 2.

S3. Analysis

We discuss a few possible directions for future study. Note that *MDC* stands for modified DeepCluster.

Visual ambiguity. As shown in visualization, visual ambiguity leads to mis-classification of certain classes. Snowy ground is often confused with either sky or water, and grass on a flat ground is confused with ground. The core problem is twofold: First, the classification of the segment masks

Brightness	Contrast	Saturation	Hue	Grayscale	Gaussian blur	Horizontal Flip	Random Crop	Accuracy	mIoU
								48.09	13.84
✓								47.98	13.59
	✓							48.08	13.63
		✓						48.09	13.64
			✓					48.09	13.65
				✓				47.98	13.63
					✓			48.03	13.59
✓	✓	✓	✓	✓	✓			47.42	13.39
						✓		47.61	13.71
							✓	48.08	13.63
						✓	✓	47.60	13.76
✓	✓	✓	✓	✓	✓	✓	✓	46.28	13.16

Table 2: We evaluate PiCIE with test-time augmentation where each transformation follows the same hyper-parameters as training, when applied. The result shows that PiCIE is robust to photometric and geometric transformations during inference.

are done with cluster centroids, which follow the "majority trend." For example, the majority of "ground" instances is not covered by snow, making the confidence low. Second, the visual similarity does not always correlate to the semantic similarity, and such discrepancy leads to confusion. "Snow ground" is often texture-less and mono-colore, similar to "sky" or "water." This is an inherent limitation of unsupervised learning methods.

Co-occurrence. Some foreground classes such as "boat" or "airplane", only occur surrounded by "water" or "sky." Since *stuff* categories have far more pixels, they are often *subsumed* in the co-occurring background classes. We hypothesize that this effect will be mitigated if the dataset had more images of stand-alone "boat" or "airplane." or with an effective way to contrast between the two entities such as using either a generic or a learned boundary detector, which can be a future work.

Boundary precision. Since we do not have any supervision to train for precise boundaries, many foreground instances are segmented with over-confidence. Pixels around boundaries are hard samples to correctly predict. Using a generic edge detector or post-processing through iterative refinement such as CRF [9] may improve the result, which is outside the scope of our project.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 4321, 4322

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 4322

[3] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 4322

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 4322

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4321

[6] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019. 4321, 4322

[7] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 4322

[8] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 4321

[9] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 4323

[10] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4322

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4321

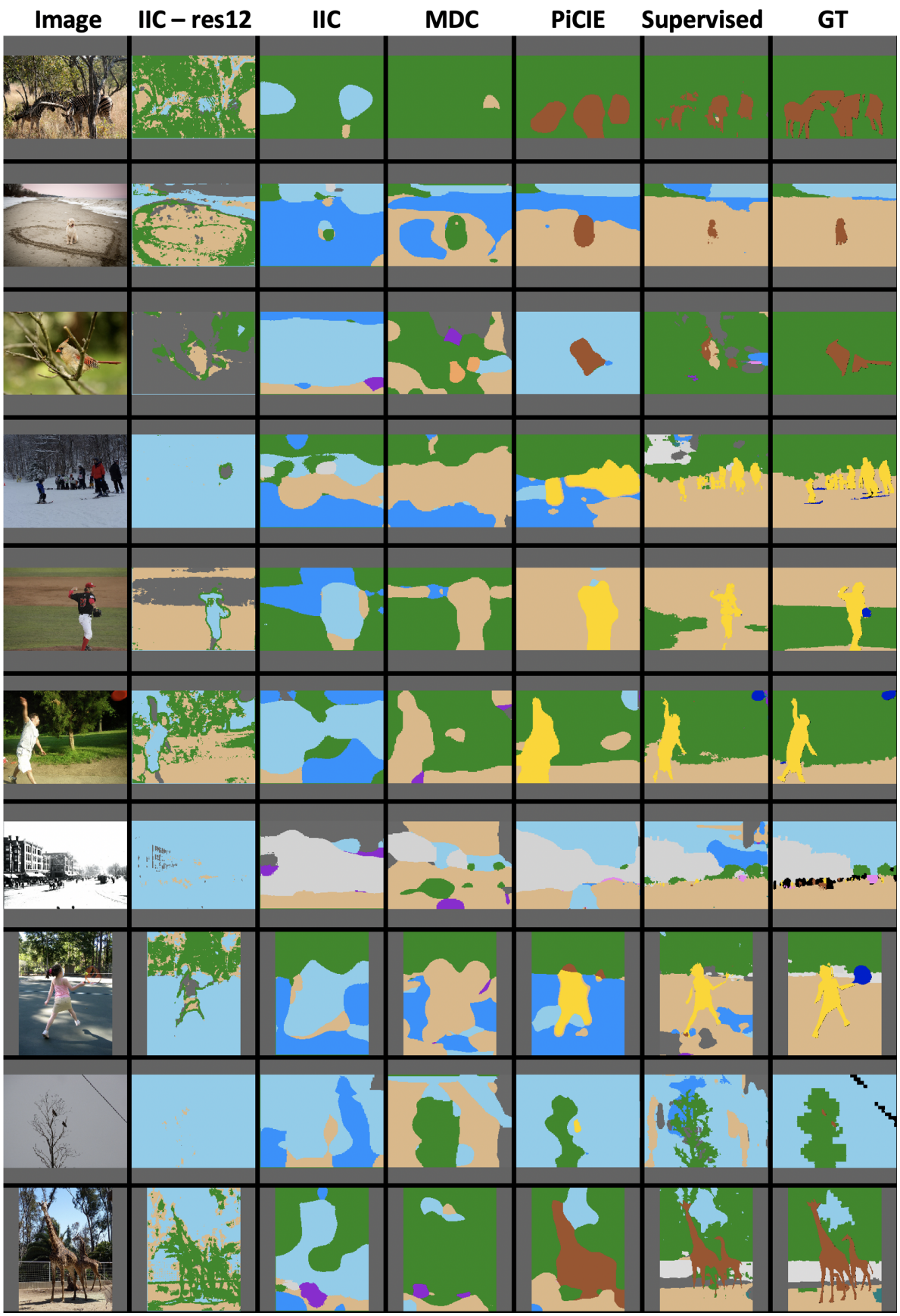
Image
IIC – res12
IIC
MDC
PiCIE
Supervised
GT


Image
IIC – res12
IIC
MDC
PiCIE
Supervised
GT

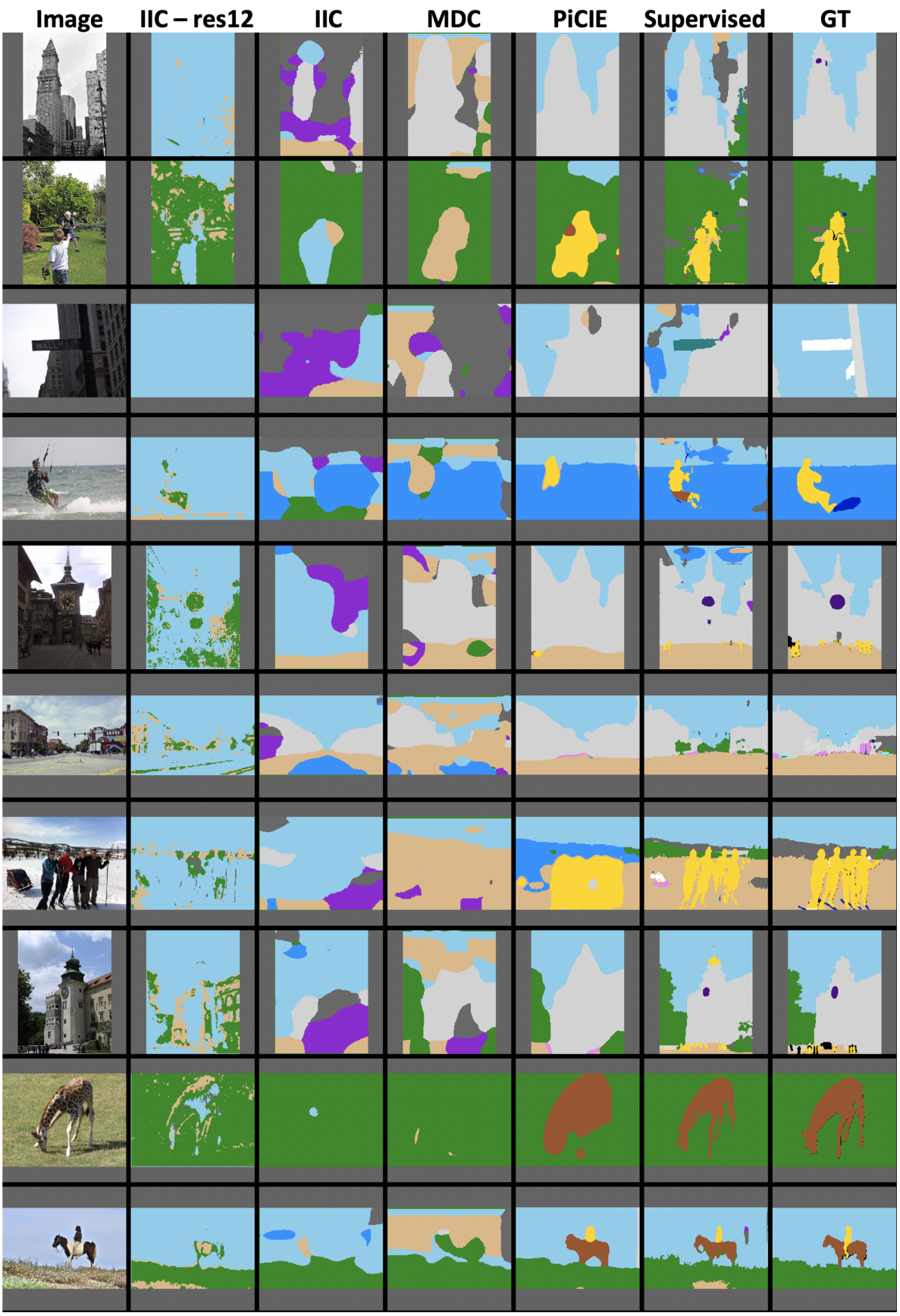
Image
IIC – res12
IIC
MDC
PiCIE
Supervised
GT

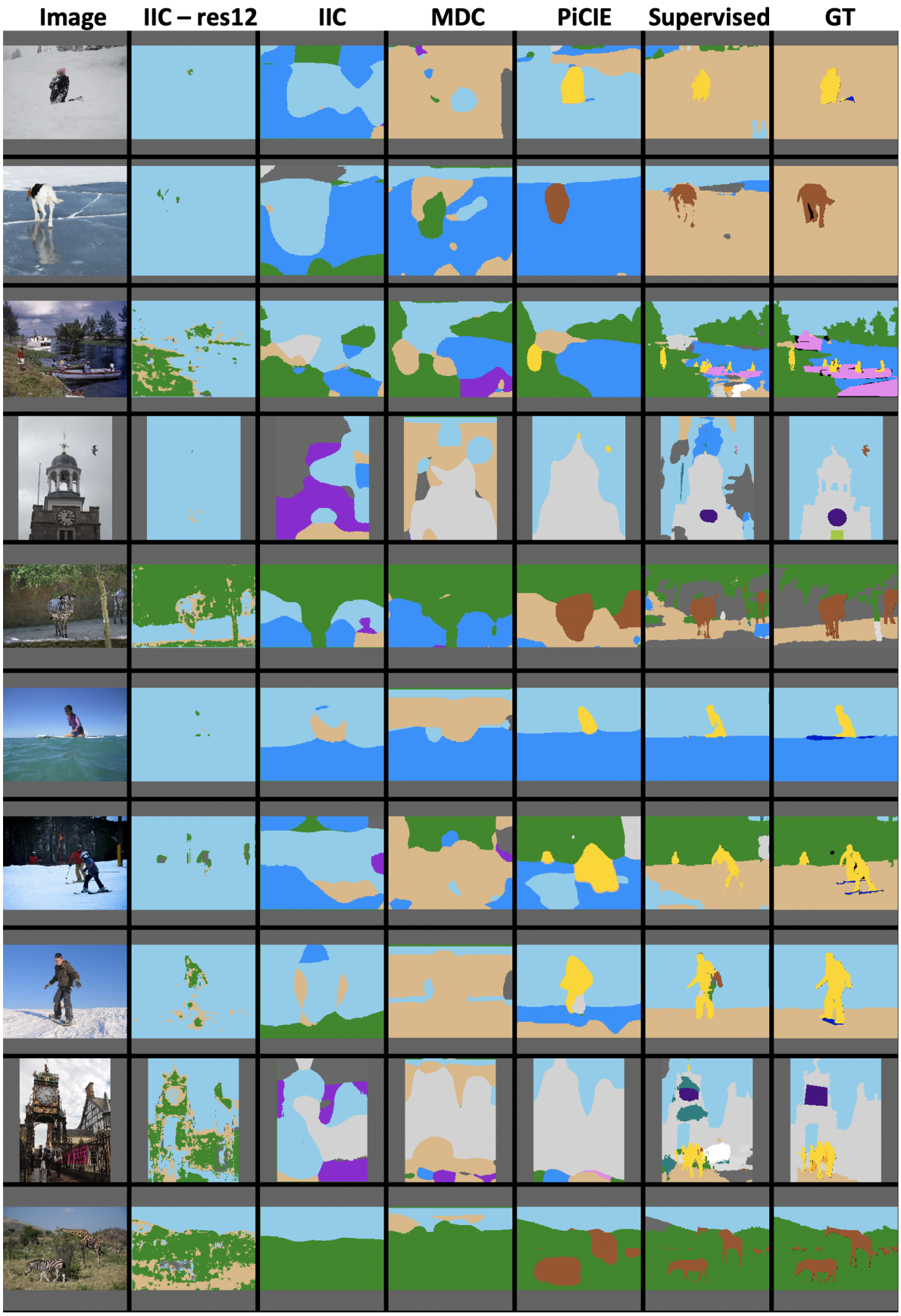
Image
IIC – res12
IIC
MDC
PiCIE
Supervised
GT

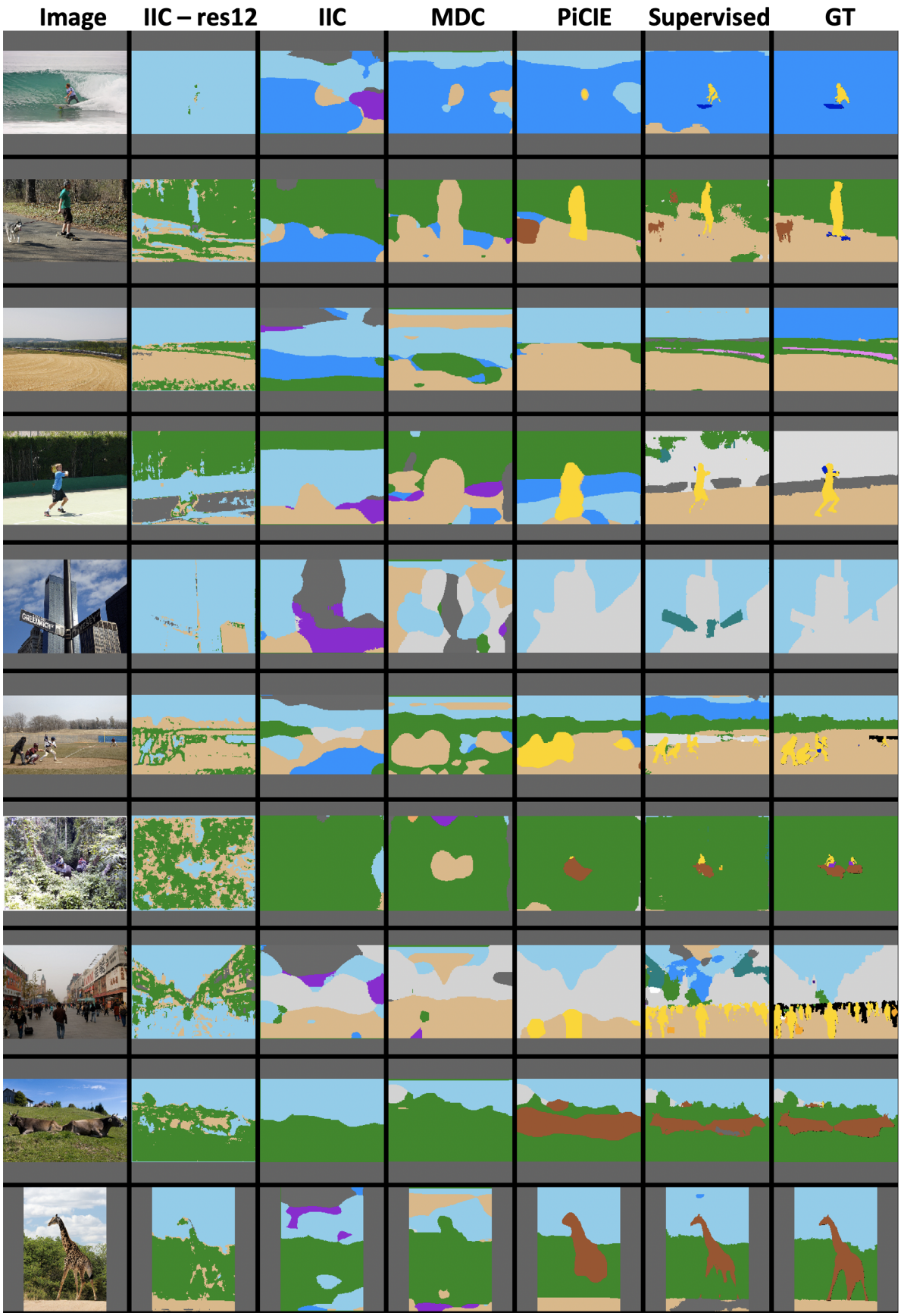
Image
IIC – res12
IIC
MDC
PiCIE
Supervised
GT

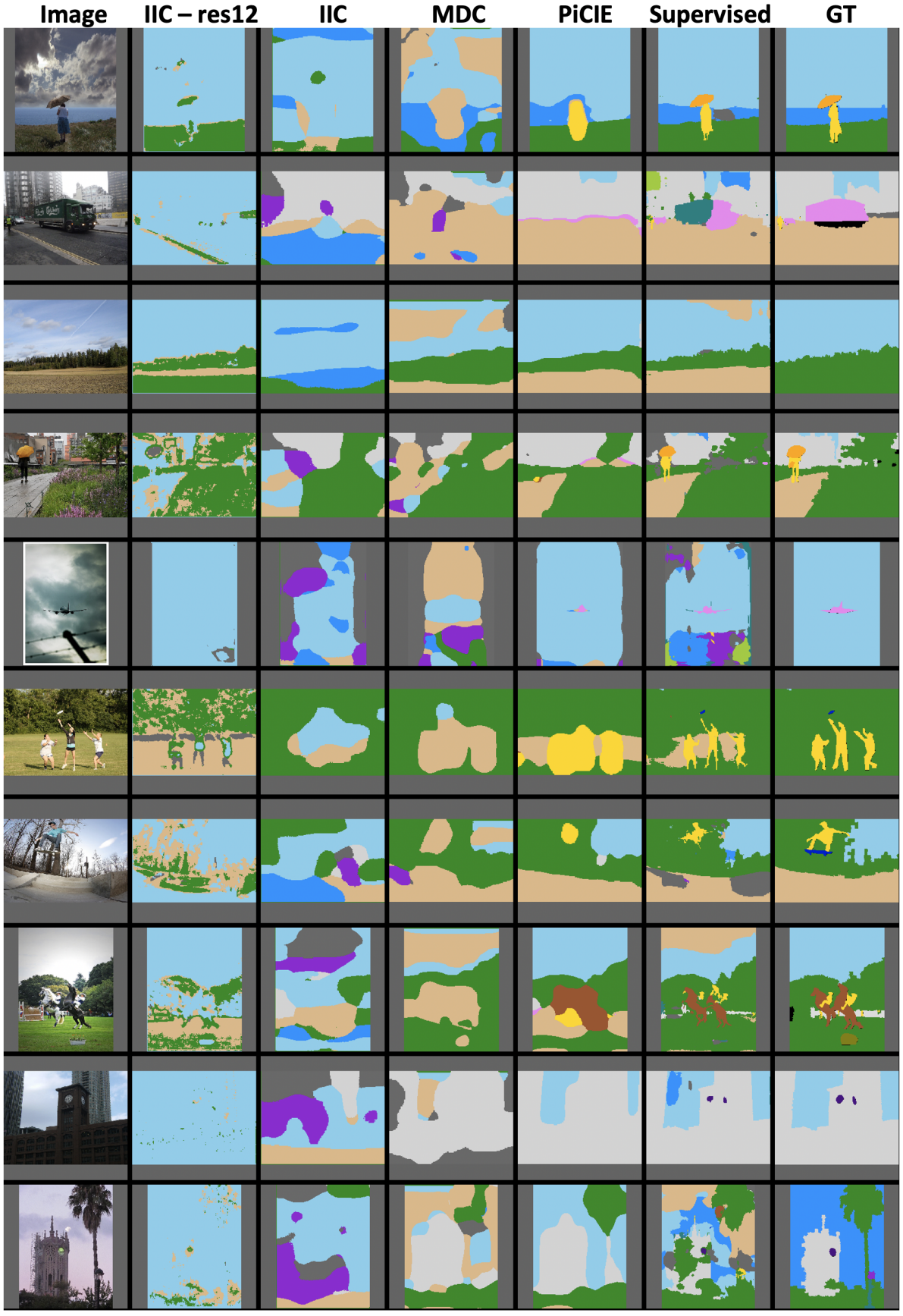
Image
IIC – res12
IIC
MDC
PiCIE
Supervised
GT