Human Instance Matting via Mutual Guidance and Multi-Instance Refinement
Supplementary Material

1. Network Structure

From the perspective of functionality, our InstMatt consists of two steps, that is, instance recognition and mask refinement. We adopt MaskRCNN [5] with the backbone ResNet50 [6] as our instance recognition model since MaskRCNN is a conventional and also competitive approach in instance segmentation. We take the publicly released MaskRCNN pre-trained weight from Detectron2 [11] without further finetuning because this model is well-trained on the large-scale COCO [8] dataset containing rich scenarios which can be well generalized to other datasets.

The mask refinement step in InstMatt can be further divided into two modules, i.e., tri-mask guided matting branch and multi-instance refinement. We adopt the network used in MG [12] as our matting branch. The network takes ResNet34 [6] as the backbone and applies three convolution blocks with a stride of 8, 4 and 1 in the decoder respectively to reconstruct the features for tri-matte prediction. During inference stage, after extracting the instance masks, we obtain a tri-matte for each instance. Next, all the tri-mattes are sent to the multi-instance refinement for information synchronization.

Multi-Instance Refinement. Considering the crowded cases with many instances, it is infeasible to perform multi-instance refinement on the whole image without out-of-memory problem when the memory storage is limited. Note that we only have to synchronize information on pixels which have information difference among all the instances and the background, that is, for a pixel p,

$$\sum_{i=1}^{n} \alpha_{p,i,t} + \alpha_{p,b} \neq 1 \tag{1}$$

For other pixels already satisfying the multi-instance alpha constraint, we may not gain much promotion from the information synchronization. Thus, we adopt the patch inference guided by an error map E, which is computed as follows,

$$E = |\frac{1}{n}\sum_{i=1}^{n} \alpha_{i,b} + \sum_{i=1}^{n} \alpha_{i,t} - 1| \tag{2}$$

We take the pixels with an error larger than 0.01 as the centers and correspondingly crop the patches of size 128×128 to perform multi-instance refinement.

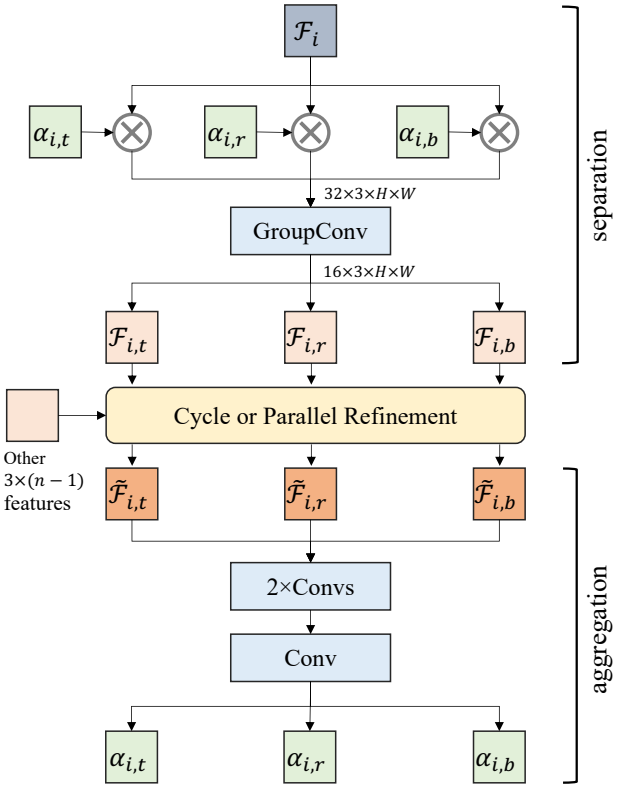

Figure 1. The structure of multi-instance refinement.

In implementation, the multi-instance refinement contains 4 learnable layers as illustrated in Figure 1. The four convolution layers all utilize 3×3 kernel.

Cycle versus Parallel Refinement. Cycle refinement is order-sensitive, which is shown in the example in Figure 2. When adopting order 1 (instance 1, 2), the updated results get worse, while the outliers are perfectly removed when adopting order 2 (instance 2, 1). With user-supplied hints, cycle refinement is able to generate promising refined results. However, the instability makes the cycle refinement strategy inappropriate in non-interactive applications. On the contrary, the parallel refinement produces refined results not relevant to any order and thus shows stable performance.

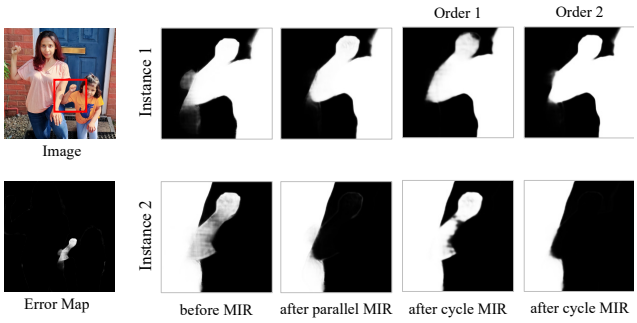

Figure 2. Cycle refinement versus parallel refinement.

Figure 3. An example of partial supervision when adapting segmentation datasets to instance matting task. The segmentation mask annotations are drawn in red while the partial supervised region are highlighted in blue.

2. Training

2.1. Datasets

During training, we use two datasets, one of which is the synthetic training dataset mentioned in the main paper Section 6.1. Since the synthetic training samples have a domain gap with natural images, we also include a natural set containing 41330 images selected from the COCO training set. However, it is non-trivial to adapt the COCO dataset to the instance matting task due to the lack of instance matting annotations. To tackle this issue, we adopt a **partial supervision** strategy to make the samples with segmentation annotations applicable in our task.

Partial Supervision. See the example in Figure 3. The instance segmentation annotations are labeled by polygons, which introduces noise along the boundary region. Thus, we respectively dilate and erode k pixels along the boundary to generate a region shown in blue in Figure 3, which are denoted as the supervised region while other pixels not masked are skipped. Such a partial supervision strategy allows us to make use of segmentation dataset without introducing noise. k is set to 35 in training.

2.2. Augmentation

To enrich the training dataset and avoid overfitting, various augmentation operations are adopted on the training samples. Besides random flip, random zoom, random shearing, as well as random crop, we propose **tri-mask augmentation** to improve the fault tolerance of the model, in particular, robustness against missing instances and imperfect masks.

Missing Instance Tolerance. Let M_r be the mask representing the union of all instances except for the target instance. Sometimes, the instance segmentation model is incapable of detecting all the instances. In this case, we only access a subset of the complete instance set to generate M_r. Therefore, the alpha constraint $\alpha_t + \alpha_r + \alpha_b = 1$ is no longer applicable. To avoid such a dilemma, we relax M_r in the training stage to a subset of reference instances instead of the complete set.

Mask Quality Tolerance. Equation 6 in the main paper mandates that M_b is the complementary set of $M_t \cup M_r$. To avoid overfitting caused by such a strong constraint, we conduct dilation or erosion on the tri-mask after computing M_t, M_r and M_b. In this way, the tri-mask may exhibit various gaps or overlaps among each other, thus introducing some uncertainty in training to better accommodate possible uneven quality of segmentation tasks.

Tri-mask Augmentation. Due to the aforementioned two robustness considerations, we generate tri-masks in three steps, 1) instance mask generation, 2) instance separation, 3) mask perturbation.

In the first step, for an image with n instances, we adopt two ways to generate masks for these instances. For a subset of the n instances, we obtain their masks from the instance segmentation model; for the rest of the instances, we generate their masks from a random truncation on the ground truth alpha matte. A hybrid of the two ways in instance generation increases the diversity of masks.

In the second step, we first randomly pick an instance i as the target instance, then randomly choose a subset from the rest of $n-1$ instances to produce M_r. Finally we obtain M_b by $1-M_t \cup M_r$. Such relaxation operation on M_r make the model ascribe the pixels of those undetected instances to α_r, rather than α_t or α_b.

In the last step, we randomly dilate or erode or perform a hybrid of dilation and erosion on the tri-mask with kernel size in $[1, 30]$. Such perturbation on tri-mask further improves the fault tolerance of our model.

Note that the ground truth tri-matte for the tri-mask are generated without the relaxation or perturbation operation.

Method	IMQ_{mad}	IMQ_{mse}
without tri-mask aug.	67.51	76.54
with tri-mask aug.	69.40	79.74

Table 1. Ablation study on tri-mask augmentation.

Method	IMQ_{mad}	MQ_{mad}	RQ
MaskRCNN [5]	24.22	25.57	94.71
CascadePSP [1]	64.58	68.19	94.71
MaskGuided [12]	57.98	63.70	91.02
InstMatt (Ours)	**70.26**	**73.83**	**95.17**

Table 2. MQ and RQ of MaskRCNN and our InstMatt.

Their generation follows Equation 3–5:

$$\alpha_{i,t} = \alpha_i \tag{3}$$

$$\alpha_{i,r} = \sum_{j=1 \text{ and } j \neq i}^{n} \alpha_j \tag{4}$$

$$\alpha_{i,b} = 1 - \alpha_{i,t} - \alpha_{i,r} \tag{5}$$

Without and with the tri-mask augmentation, the IMQ_{mad} of our InstMatt is 67.51 and 69.40 as tabulated in Table 1, showing a promotion benefiting from the tri-mask augmentation.

2.3. Training Schedule

Our training schedule consists of two steps:

1. Train the matting branch on the synthetic and natural training datasets. The branch is initialized with ImageNet [2] pre-trained weight. We use a batch size of 16 in total on 4 GPU cards. Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ is adopted. The initial learning rate is set to 0.001 and decays at a cosine learning rate [4,9]. The training lasts for 100,000 iterations with a warm-up of the first 5, 000 iterations.

2. After the matting branch is well-trained, we freeze the matting branch and train the multi-instance refinement module. In the second step, we use a batch size of 4 in total on 4 GPU cards. The initial learning rate is set to 0.0001. The training lasts for 25,000 iterations with a warm-up of the first 1, 000 iterations. We keep the other hyper parameters the same as those in the first step.

3. Evaluation and Discussion

3.1. IMQ Metric

Computation. We propose an IMQ metric to provide a comprehensive and unified evaluation of instance matte quality. The computation of IMQ can be divided into two steps, i.e., instance matching and similarity measurement. During matching instances, we first quantify the predicted and ground truth alpha mattes by applying $\alpha > 0$ into binary mask to compute IoU matrix. Here, we use 0 rather than other values as the threshold considering the semi-transparent/transparent objects usually composed of small alpha values. Other threshold will turn the alpha mattes of these objects into an incomplete binary mask which cannot cover the whole objects, thus leading to a wrong instance matching result. Quantification with 0 as threshold makes the IMQ metric applicable for not only human instance alpha mattes but also other semantic classes including transparent objects.

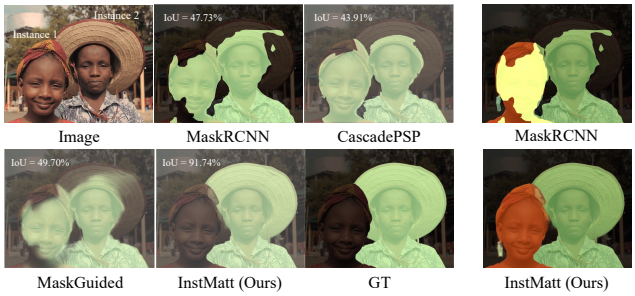

Figure 4. IoU comparison. We show the predicted mask or alpha matte as from multiple methods as well as the ground truth for instance 2 in the left. CascadePSP and MaskGuided are both incapable of turning the wrong mask into a correct one while our InstMatt does. Right shows the results of two instances.

During similarity measurement, we adopt the widely used error functions in conventional matting task to evaluate the instance alpha matte from multiple dimensions. In implementation, we compute $\mathcal{E}(\alpha, \hat{\alpha})$ as follows,

$$\mathcal{E}(\alpha, \hat{\alpha}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathcal{E}(\alpha_p, \hat{\alpha}_p) \tag{6}$$

$$\mathcal{P} = [\alpha > 0] \cup [\hat{\alpha} > 0] \tag{7}$$

We take the average upon the union of quantified α and $\hat{\alpha}$ instead of the whole image to avoid the overwhelming zero values from the large amount of background pixels especially for small instances.

MQ and RQ. As mentioned in Section 5 in the main paper, IMQ can be decomposed of two components, RQ and MQ, measuring the instance recognition quality and the alpha matte quality of TP set respectively. We provide the RQ and MQ in Table 2. Compared to MaskRCNN, CascadePSP significantly promotes the instance matte quality among TP set, however, does not improves the RQ at all, demonstrating that CascadePSP cannot upgrade a low-quality mask which has an IoU below 0.5 with any ground truth instance mask into a high-quality instance mask due to the lack of instance awareness.

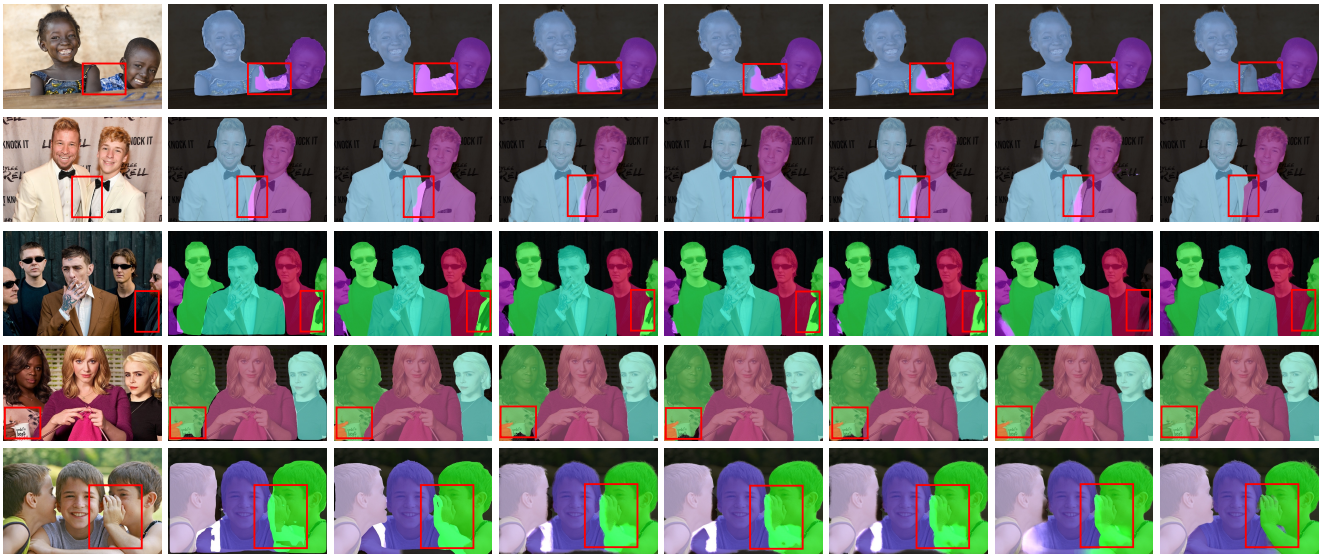

Figure 5. Qualitative comparisons on images containing multiple instances in close proximity.

Figure 6. Qualitative comparisons on images containing overlapping instances with long-range occlusion.

On the contrary, besides refining the instance alpha matte along the boundary and the hairy regions among the TP set, our InstMatt is also capable of recognizing an instance and correspondingly extracting its alpha matte even though only a low-quality mask with misleading instance information is provided, such as the first example in Figure 5 in the main paper and the example in Figure 4.

3.2. Experiment and Comparison

Experiment Setting. We train our method InstMatt and MaskGuided [12] on both the synthetic and natural datasets. For other methods including CascadePSP [1], GCA [7], SIM [10] and FBA [3], we use the released model from their official project website. To generate trimap for the trimap-based matting methods, we respectively dilate and erode the mask predicted from MaskRCNN [5] with a kernel size of 5 and then repeat the dilation and erosion operations for 10 times.

Comparisons on HIM2K. Through our mutual guidance strategy in tandem with the multi-instance refinement module, our InstMatt shows superiority in various challenging cases. We provide more qualitative results for comparisons.

Figure 5 shows the cases containing multiple instances next to each other closely. MaskRCNN is able to distinguish instances but produces overlapping instance masks, which cannot be addressed either by CascadePSP [1] or a naive extension of existing matting models [3, 7, 10, 12]. Our InstMatt, however, can clearly separate the instances

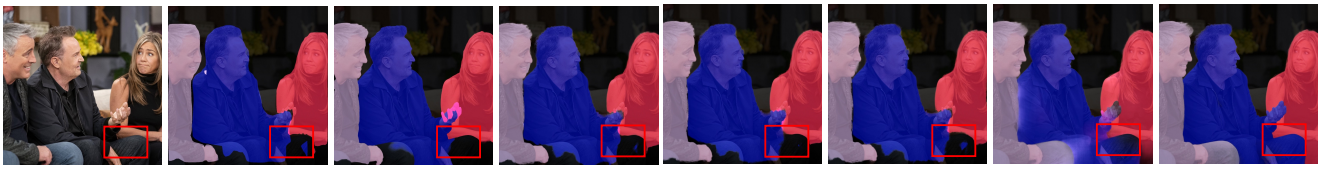

Figure 7. Qualitative comparisons on a case with incomplete instance segmentation masks.

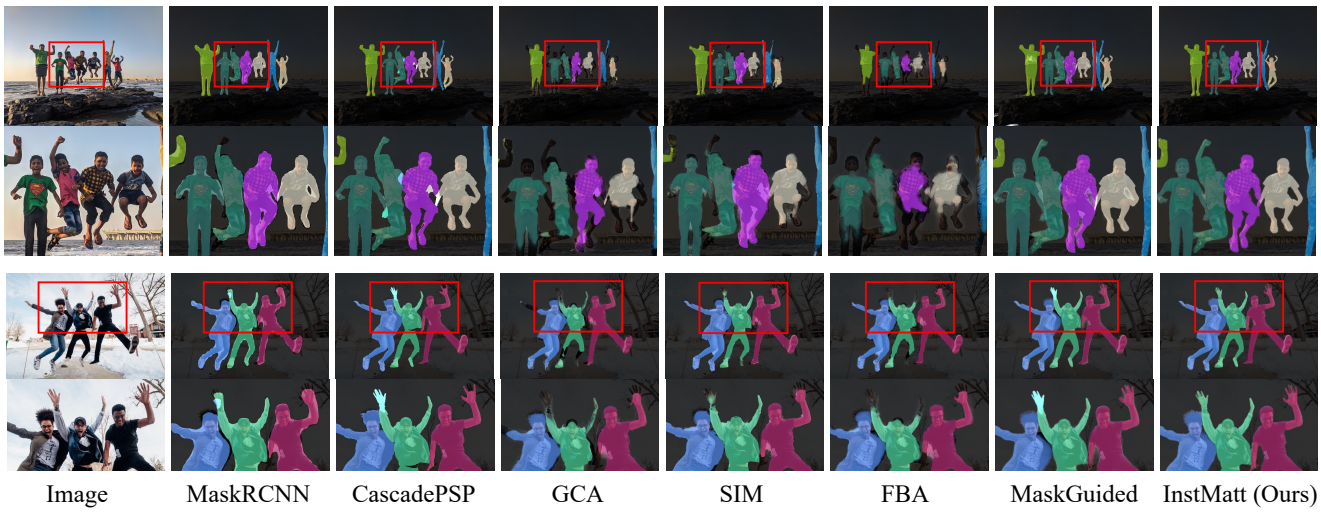

Figure 8. Qualitative comparisons on images with small instances.

and generate non-overlapping instance alpha mattes.

Figure 6 shows the cases with occlusion. Under such cases, a part of one instance, usually a hand, or an arm, appears within the region of another instance and is far away from its own body. It is difficult to solve these cases for recognition tasks due to the limitation of receptive fields and the bottleneck of long-range feature propagation. As shown in Figure 6, both instance segmentation models and matting models fall short of producing satisfactory results in these cases, while our InstMatt still produces promising results. Inter-instance mutual exclusive information guides the model to retrieve the remote pixels sharing the similar appearance with the body region instead of ascribing them to the other instances.

Figure 7 compares the performance on a case with incomplete instance segmentation masks. Although CascadePSP or matting-based models are able to refine the mask along the boundary, they cannot recover a part of missing region due to the lack of instance awareness. Our InstMatt can find the lost region from the background giving the credit to the mutual exclusive guidance between the human instances and the background.

Figure 8 compares the performance on small instances. Compared to other methods, our InstMatt still shows stable performance on the instances of small or tiny scales, demonstrating the generalization ability and the fineness of our results.

Figure 9. An example explaining the limitation of our method.

3.3. Limitation

Under most cases, our method is capable of upgrading a low-quality instance mask into a high-quality alpha matte. However, sometimes the instance segmentation model cannot differentiate two largely overlapping instances as shown in Figure 9. Note that this example is different from the one in Figure 4. In Figure 4, MaskRCNN recognizes two instances although the mask of instance 2 covers a part of instance 1. Differently, in Figure 9, the instance segmentation model regards two left human instances as one and only predict a mask for the two left instances. In this case, our model can only refine the mask but cannot separate them due to the lack of sufficient guidance.

3.4. More Qualitative Results

More qualitative comparisons on images containing complex multiple overlapping or crowded cases are shown in Figure 10.

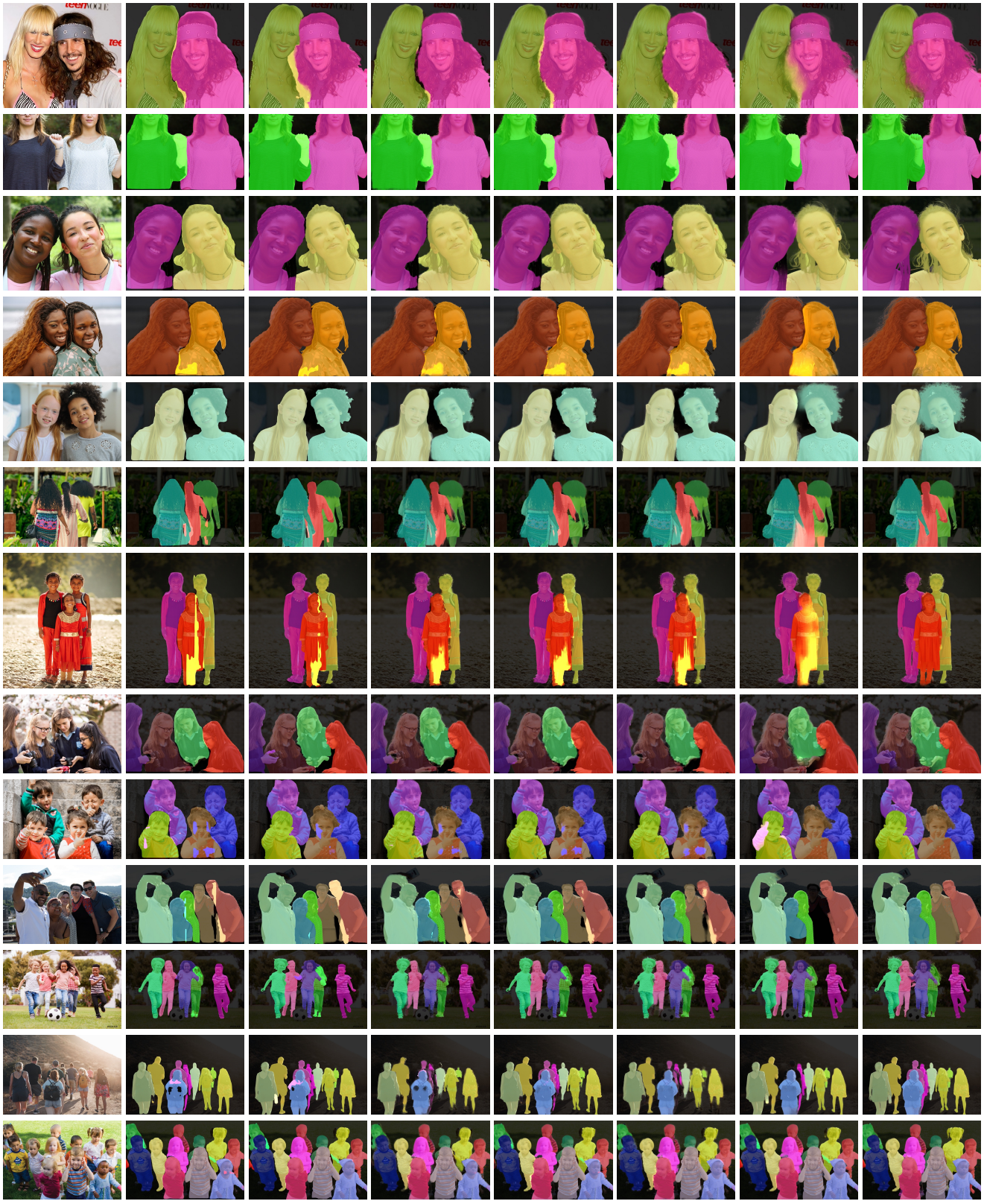

Figure 10. Qualitative comparisons on more complex cases with multiple overlapping or crowded cases.

References

[1] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3, 4

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 3

[3] Marco Forte and François Pitié. F, b, alpha matting. *CoRR*, abs/2003.07711, 2020. 4

[4] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. 3

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision*, 2017. 1, 3, 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 1

[7] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *AAAI Conference on Artificial Intelligence*, 2020. 4

[8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision*, 2014. 1

[9] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 3

[10] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Semantic image matting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 4

[11] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 1

[12] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan L. Yuille. Mask guided matting via progressive refinement network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 3, 4