Supplementary File to "RANet: Ranking Attention Network for Fast Video Object Segmentation"

Ziqin Wang[1,3], Jun Xu[2,4], Li Liu[2], Fan Zhu[2], Ling Shao[2]
[1]The University of Sydney, Sydney, Australia
[2]Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE
[3]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
[4]Media Computing Lab, College of Computer Science, Nankai University, Tianjin, China

Project page: https://github.com/Storife/RANet

1. Content

In this supplementary file, we provide more details of the proposed *Ranking Attention Network* (RANet), and additional quantitative and qualitative results to support our experiments in the main paper. Specifically,

- in §2, we present more implementation details of the proposed RANet, including its overall structure (encoder and decoder) and data processing strategy;
- in §3, we present qualitative results to show the effects of the proposed *Ranking Attention Module* (RAM);
- in §4, we provide more qualitative results to show the influence of training strategies, including static images pre-train (IP) and video fine-tuning (VF), as described in §3.5 of the main paper, and also online learning;
- in §5, we present qualitative comparison of different methods on DAVIS$_{16}$/$_{17}$ datasets;
- in §6, we give two failure examples to show the potential limitation of the proposed RANet.

2. More Implementation Details of RANet

Here, we present more details of the proposed RANet.

Encoder. The encoder is built on a basic ResNet-101 [3], and the features extracted from the last three blocks are employed for correlation calculation. We process these features via convolutional layers (with the kernel size of 1×1) to decrease their channel size by four-fold. The features are resized into the same size and concatenated. Then, the concatenated features are passed through a convolutional layer for feature merging. The combined features are of size $W = 54$ and $H = 30$, where W and H are the weight and height of the current frame, respectively. To reduce computational costs and to increases robustness for feature matching, we add a pooling layer after the template stream in the encoder. Hence, the template features are of size $W_0 = 27$ and $H_0 = 15$, where W_0 and H_0 are the weight and height of the template frame, respectively.

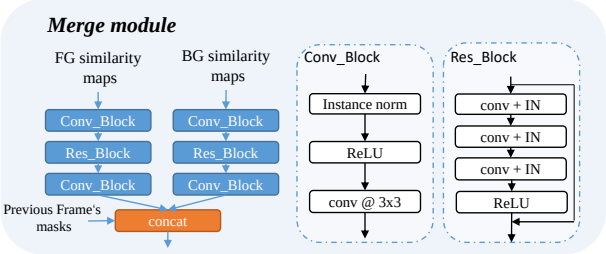

Figure 1: **Structure of the merge module**. It consists of two branches with shared parameters.

Decoder. As shown in Figs. 1 and 2, the decoder consists of a merge module and a pyramid network. For each object, the RAM module generates two sets of similarity maps for the foreground (FG) and background (BG), respectively. And the merge module aims to integrate the sets of FG and BG maps, as well as the previous frame's mask. The FG stream and the BG stream in the merge module (Fig. 1) share the same parameters, and each stream contains a Res-block and two Conv-blocks. The detailed structures of Res-block and Conv-block are shown in the right part of Fig. 1. Next, the features of the two streams are concatenated and fed into the pyramid network for refinement. A pyramid structure network is used in our decoder, along with a multi-scale skip-connection, which allows the network to utilize rich features from different layers for refinement. However, it would be computationally expensive if all the features were fed into the decoder. To this end, we reduce the channel size of the multi-scale features using convolutional layers, before they are fed into the decoder. As shown in Fig. 2, the decoder is a three-level pyramid. Each level has a Multi-scale block, a Res-block and two Conv-block as shown in Fig. 1. The Multi-scale block has three branches of convolutional layers with dilated sizes of 1, 6, and 9. All the branches are merged through element-wise summation.

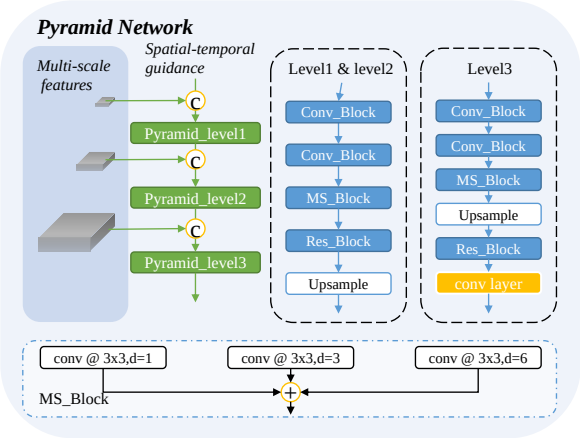

Figure 2: **Structure of the pyramid network**. The multi-scale features are extracted from the backbone network.

3. Qualitative Results for RAM

In Fig. 3, we compare the results of different variants to RANet: *w/ RAM* (original RANet), *w/o Ranking*, and *Maximun*, as described in §4.3 of the main paper. One can see that RANet performs better with the RAM module.

4. Qualitative Results for Training Strategies

Now we present qualitative results to show the effects of different training strategies. In Fig. 4, we show the results of the proposed RANet only pre-trained on static images (*IP*), RANet pre-trained on static images and fine-tuned on training videos (IP+VF), and the original RANet boosted by online learning (OL) techniques. It can be seen that IP, VF, and OL all help improve the VOS performance of the proposed RANet. The corresponding quantitative results are provided in Tables. 5 and 6 of the main paper.

5. Qualitative Results of Different Methods on DAVIS$_{16/17}$ Datasets.

Here, we provide qualitative comparison of different VOS methods on DAVIS$_{16/17}$ datasets. The results are listed in Figs. 5-8. It can be seen that the proposed RANet achieves more accurate performance on VOS than the other competing methods, i.e., OSVOS [1], SiamMask [6], and FAVOS [2], on both single-object and multi-object tasks.

6. Failure Case

We show some failure cases of RANet in Fig. 9. Since the similarity maps are measured on pixel-level, it is difficult to distinguish similar instances those are spatially close. In this case, both spatial and temporal guidance are not feasible to locate the objects.

References

[1] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixe, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, July 2017. 2, 5, 6, 7, 8

[2] Jingchun Cheng, Yi Hsuan Tsai, Wei Chih Hung, Shengjin Wang, and Ming Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 2, 7, 8

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[4] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 4, 5, 6

[5] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 7, 8

[6] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*. IEEE, 2019. 2, 5, 6

Figure 3: **Results of different variants to RANet on sequence *motocross-jump* from DAVIS$_{16}$ dataset**. *Maximun* and *w/o Ranking* are two baseline variants of *w/ RAM* (original RANet) described in §4.3 of the main paper.

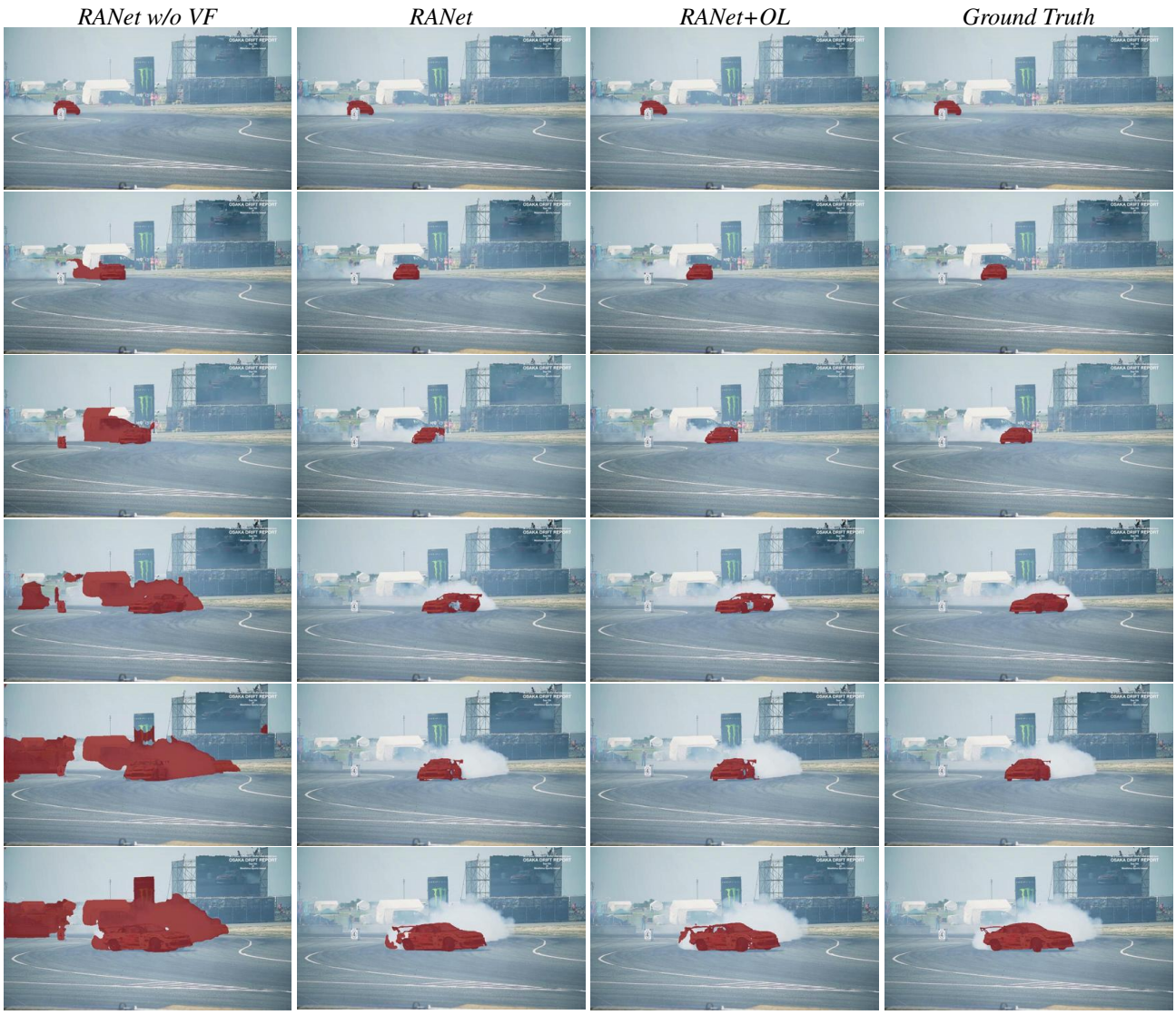

Figure 4: **Comparison results of RANet with different training strategies, on sequence *drift-chicane* from DAVIS$_{16}$ dataset [4]**. From left to right, the results are from RANet trained only on static images (*RANet w/o VF*), RANet trained on static images and fine-tuned on videos (*RANet*), RANet boosted by online learning (*RANet+OL*), and *Ground Truth*.

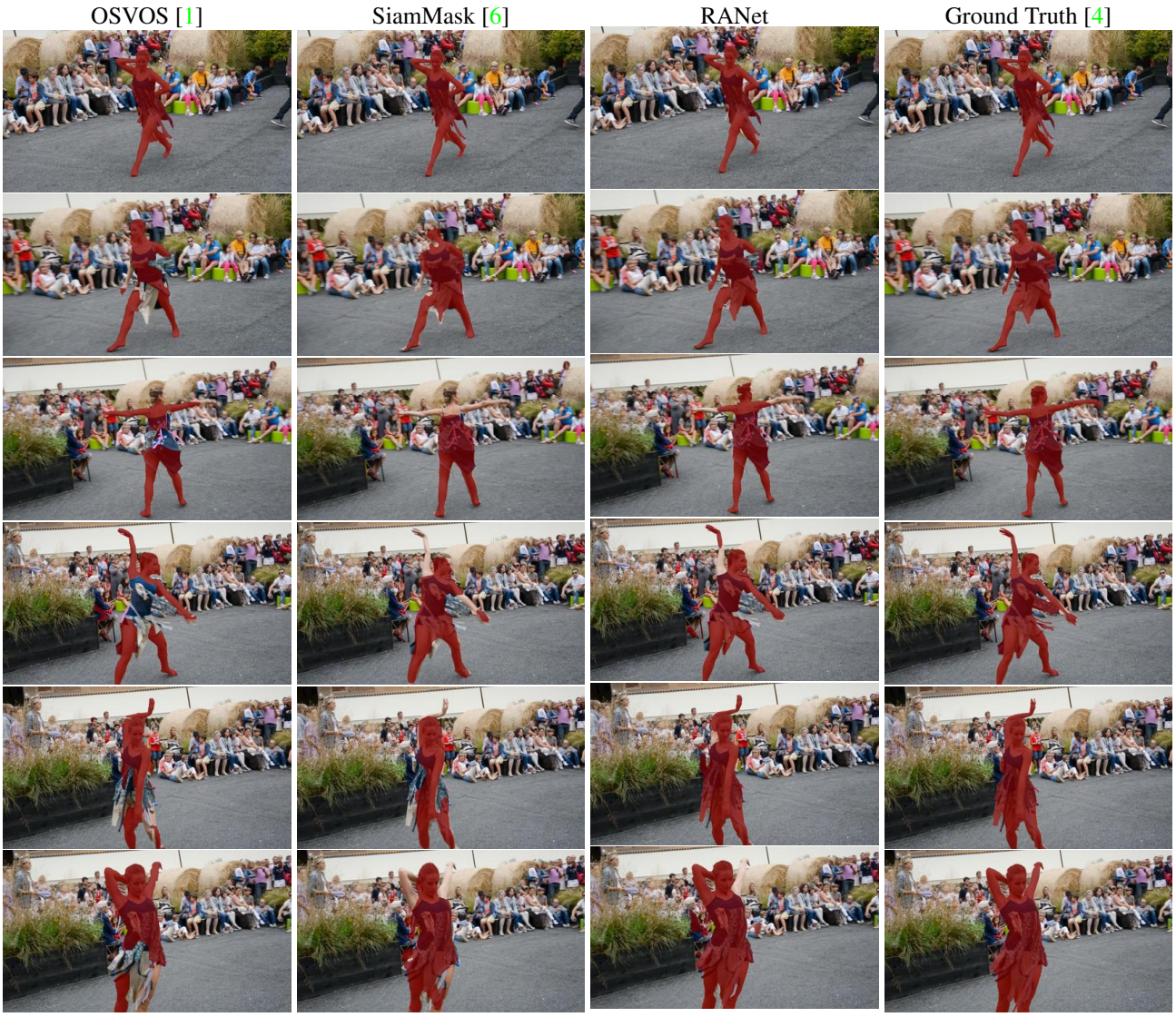

Figure 5: **Comparison of different methods on sequence *dance-twirl* from DAVIS$_{16}$ dataset [4].** From left to right: OSVOS, SiamMask, the proposed RANet, and the "Ground Truth".

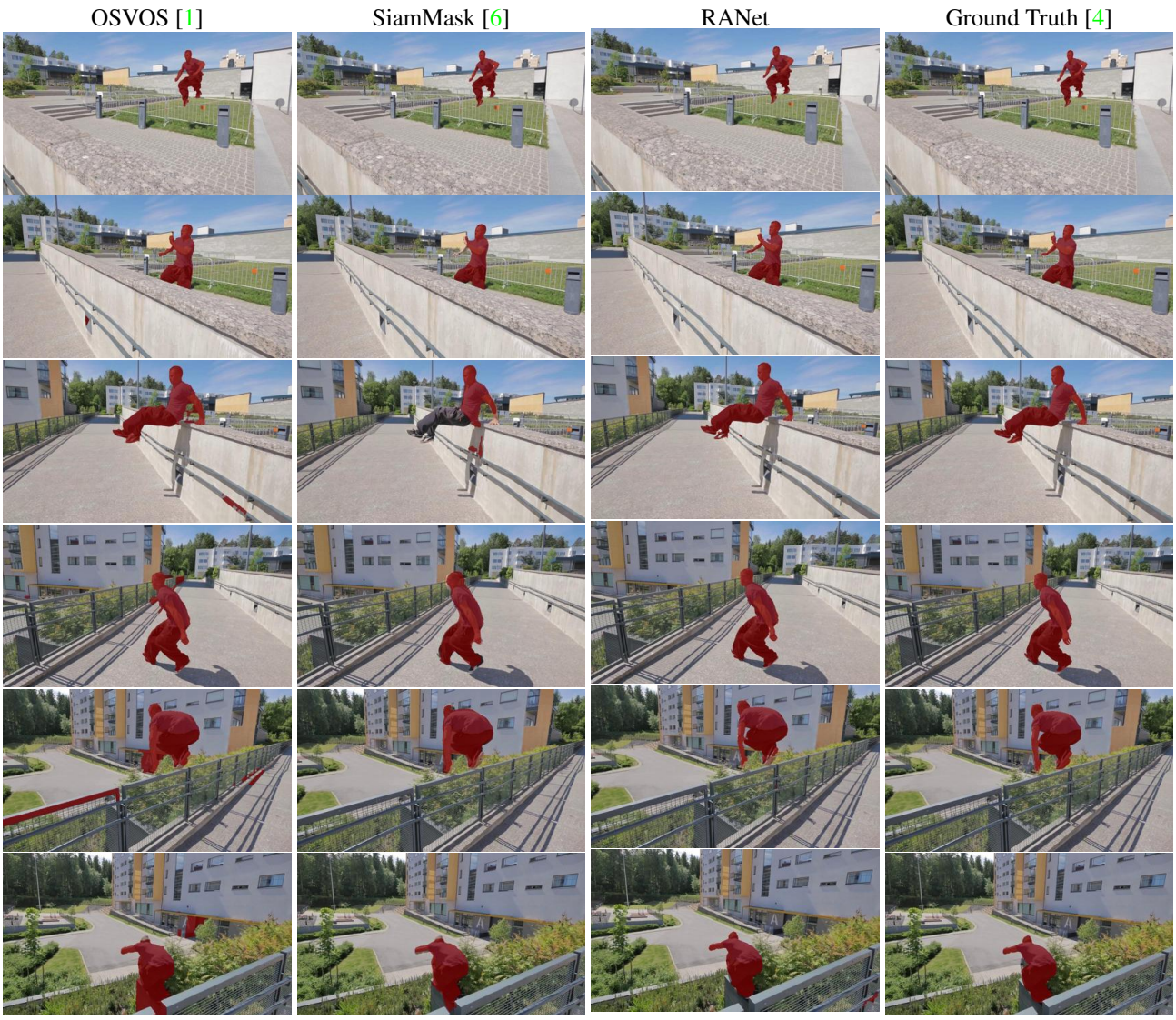

Figure 6: **Comparison of different methods on sequence *parkour* from DAVIS$_{16}$ dataset [4]**. From left to right: OSVOS, SiamMask, the proposed RANet, and the "Ground Truth".

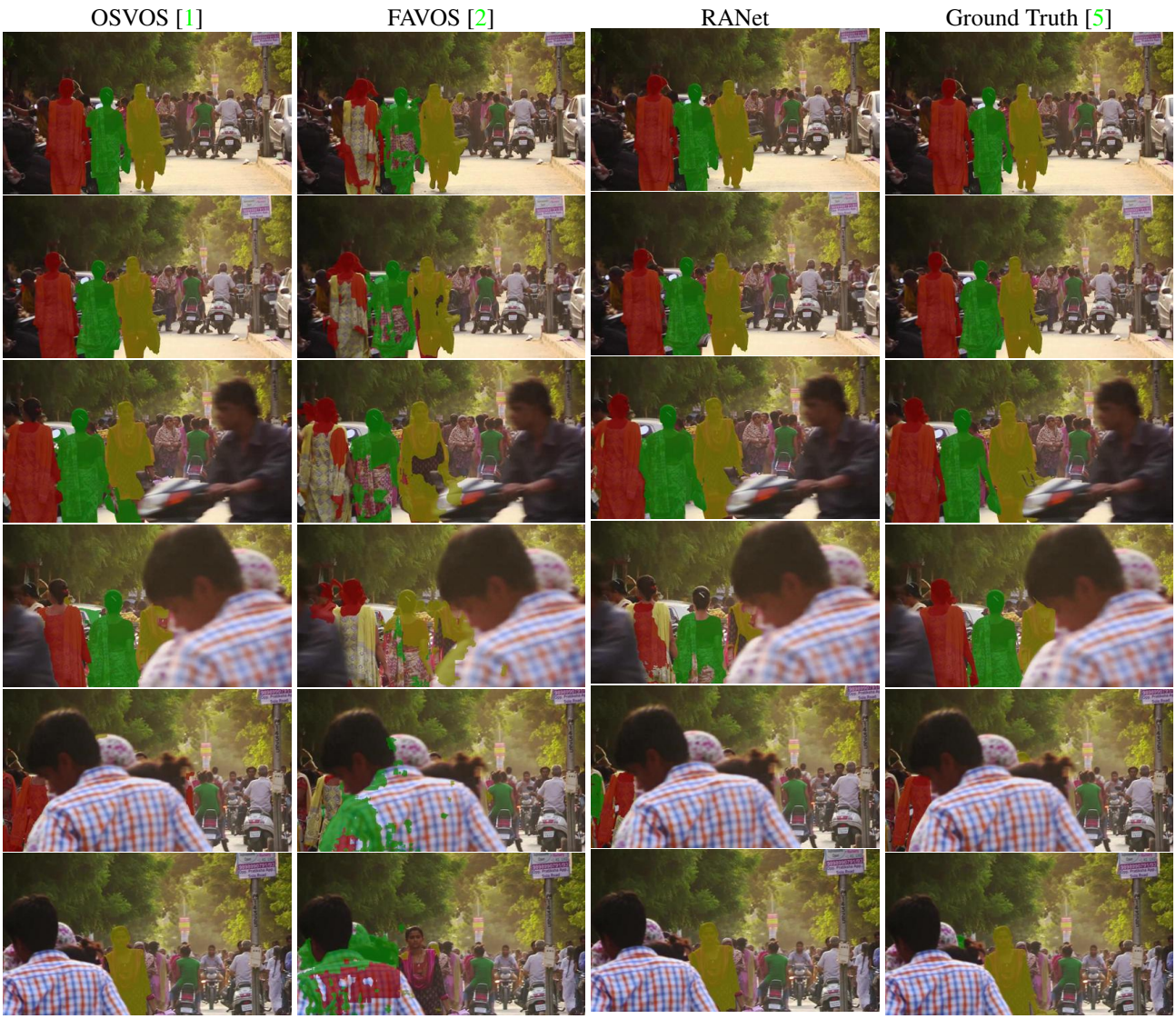

Figure 7: **Comparison of different methods on sequence *india* from DAVIS$_{17}$ dataset [5]**. From left to right: OSVOS, FAVOS, the proposed RANet, and the "Ground Truth".

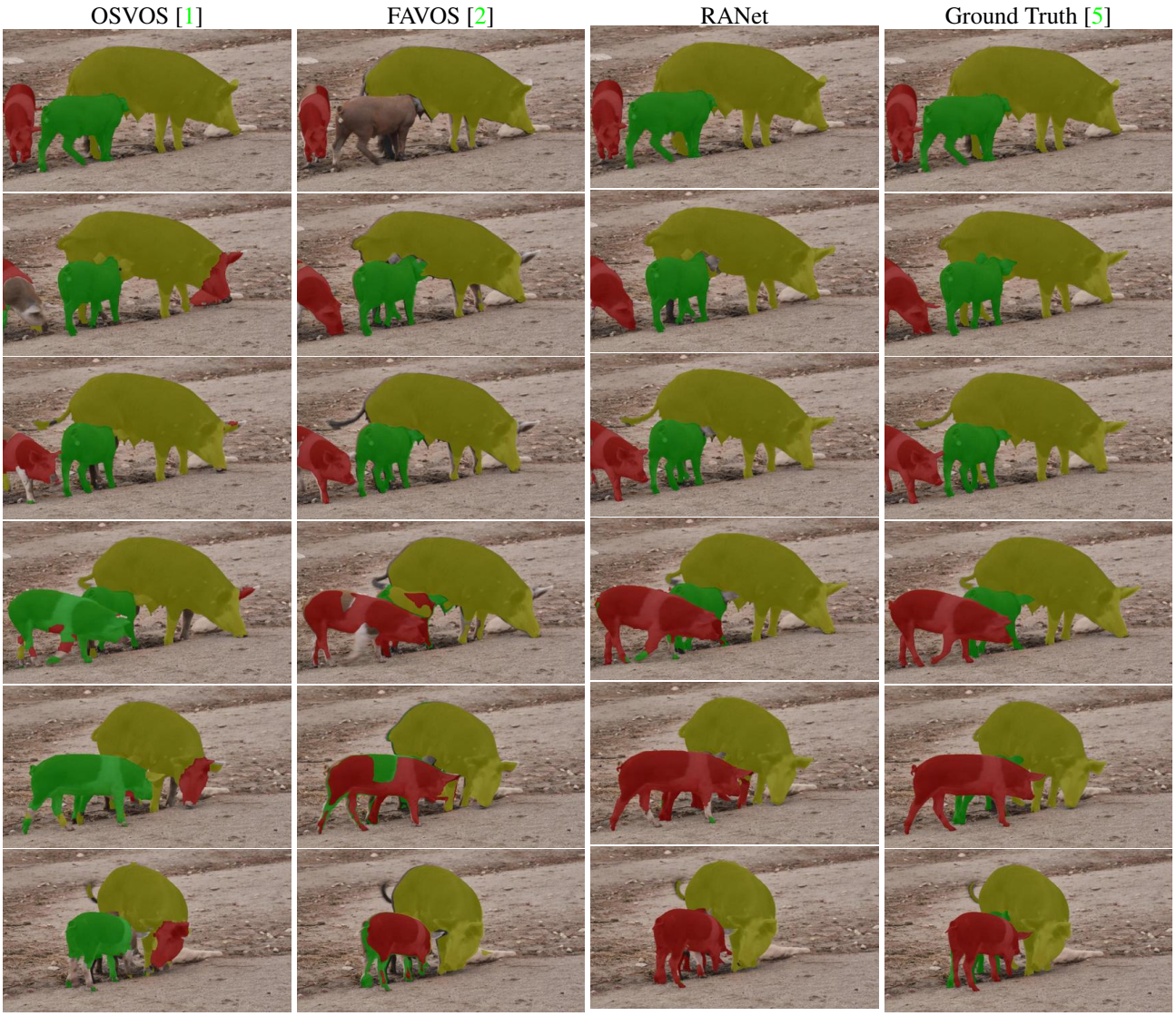

Figure 8: **Comparison of different methods on sequence *pigs* from DAVIS$_{17}$ dataset [5]**. From left to right: OSVOS, FAVOS, the proposed RANet, and the "Ground Truth".

Figure 9: **Failure cases** of our RANet. Top: template frames and masks; Bottom: current frames and predictions.