(Supplementary Materials)
VISOLO: Grid-Based Space-Time Aggregation for Efficient Online Video Instance Segmentation

Su Ho Han[1], Sukjun Hwang[1], Seoung Wug Oh[2], Yeonchool Park[3],
Hyunwoo Kim[4], Min-Jung Kim[5] and Seon Joo Kim[1]

[1]Yonsei University [2]Adobe Research [3]LG Electronics [4]LG AI Research [5]KAIST

{hansuho123, sj.hwang, seonjookim}@yonsei.ac.kr seoh@adobe.com
yeonchool.park@lge.com hwkim@lgresearch.ai emjay73@kaist.ac.kr

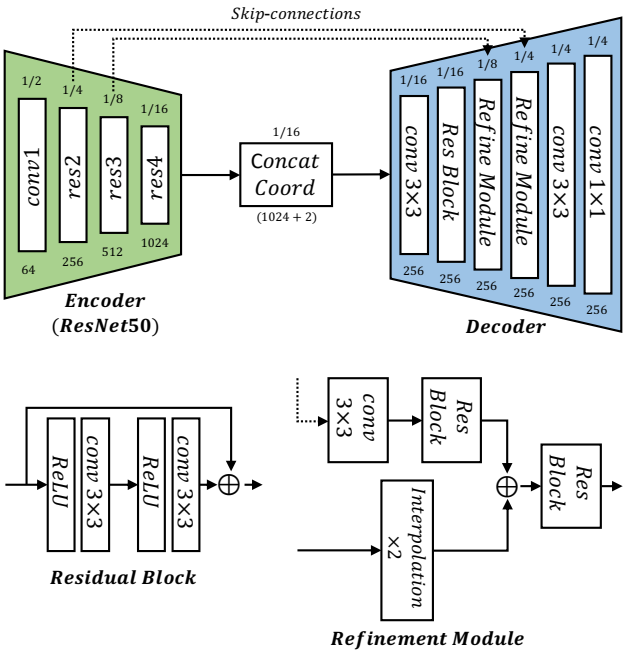

Figure 1. A detailed illustration of decoder of feature branch and encoder. The decoder takes the "res4" feature map from the encoder as the input with normalized coordinates by concatenating two additional input channels. '$\oplus$' denotes element-wise summation.

1. Branch Structure Details

The backbone network for the encoder is ResNet50 [1] and we use the "res4" feature map as the input of the category branch and the mask branch.

1.1. Category Branch

For each grid, the category branch predicts C-dimensional output of the semantic class probabilities, where C is the number of predefined classes. The category branch consists of $8\times$ convolutional layers and takes the resized feature map from the backbone network as the input. Resizing to $S_h \times S_w$ is performed by bilinear interpolation, where S_h and S_w are the number of grids in height and width, respectively. Note that after the first convolutional layer, the resulting feature map (key feature map) is used to obtain grid similarities through memory matching module, also after another three convolutional layers, the resulting feature map (category feature map) is added with the output from the temporal aggregation module. The temporal aggregation provides additional appearance information of the same instance observed in previous frames. These feature maps (key and category feature maps) are used to update the external memory for the memory matching and temporal aggregation modules. Before the final prediction, the object scores are calibrated using the weights from the score reweighting module that also utilizes the information from previous frames. The category branch generates an output tensor $\mathbf{Cat} \in \mathbb{R}^{S_h \times S_w \times C}$.

1.2. Mask Branch

We use the dynamic head from SOLOv2 [6] as the mask branch. The dynamic head consists of two sub- branches: feature branch and kernel branch. Both branches take the feature map from the backbone network as the input with normalized coordinates by concatenating two additional input channels for the spatial information. The feature branch predicts instance-aware feature map $\mathbf{F} \in \mathbb{R}^{H/4 \times W/4 \times E}$ through decoder, where H and W are the height and width of input frames respectively and E is the feature dimension.

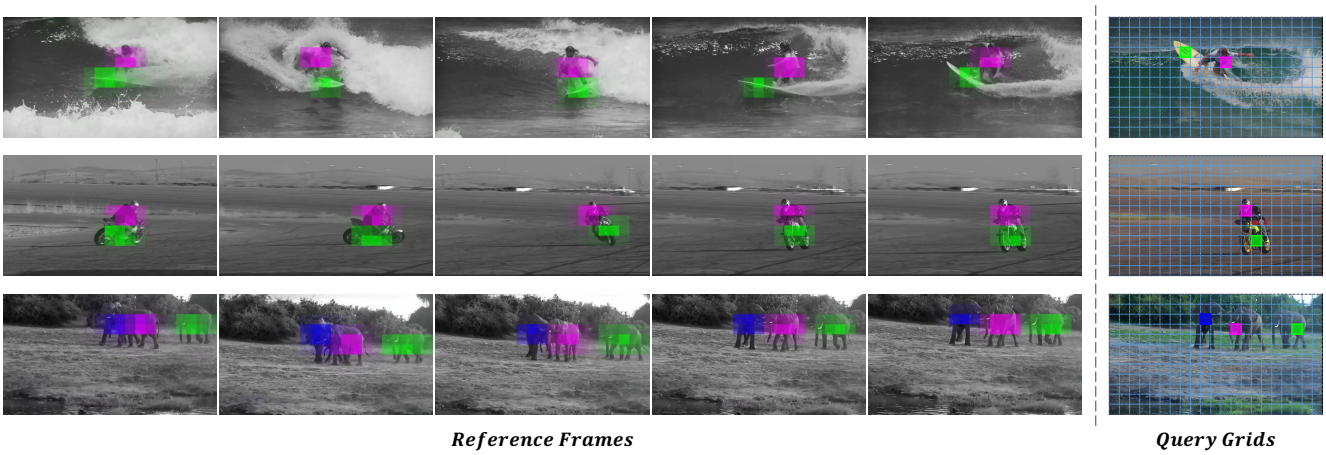

Figure 2. Visualization of our temporal aggregation module operation. We first compute the grid similarities between query grids and all grids of reference frames, and obtain the soft weight by a softmax operation. Then, we visualize the normalized soft weight of the reference frames. The query grids and weights of each grid of reference frames with respect to the query grids are assigned with the different colors.

In the feature branch, we employ the refinement module used in [4] as the building block of the decoder. The refinement module consists of the convolutional layer, residual block [2], and the interpolation operator. The module takes the feature maps of the encoder through the skip-connections. Every convolutional layer in the decoder uses 3×3 filter and produces 256-channel output; the last one uses 1×1 filter. The decoder of the feature branch and the encoder are illustrated in Fig. 1.

The kernel branch predicts 1×1 convolution kernel weights $\mathbf{K} \in \mathbb{R}^{S_h \times S_w \times E}$, conditioned on each grid's location. It consists of $6\times$ convolutional layers that produce 256-channel output, using 3×3 filters except for the last layer (1×1). Note that after the first three convolutional layers, the resulting feature map (mask feature map) is added with the output from the temporal aggregation module in order to provide additional kernel information of the same instance observed in previous frames, also it is used to update the external memory queue for the temporal aggregation module. To obtain the instance mask for grid (i, j), $\mathbf{F}$ is convolved by (i, j) output of the kernel branch, which is a 1×1 convolution kernel. The mask branch estimates the mask on a $1/4$ scale of the input frames.

2. Implementation Details

2.1. Training Details

For the training, we use the image instance segmentation dataset COCO [3], and two video instance segmentation datasets YouTube-VIS 2019 and 2021 [7]. To exploit the static image dataset for video instance segmentation, we transformed static images into 3-frame synthetic videos using random affine transformations.

First, our network is trained on the COCO [3] dataset for pre-training with the batch size 16 for 60 epochs and the initial learning rate of 1e-4, which decays at 40 epochs. After pre-training, we fine-tune the our network on the youtube VIS 2019 and 2021 datasets together with COCO dataset to prevent overfitting. When combining datasets, we use 21 classes of COCO that are related with 40 classes of YouTube-VIS 2019 and 2021, respectively. During fine-tuning, we sampled training data with the following distribution: (YouTube-VIS 2019 (75%), COCO (25%)) and (YouTube-VIS 2021 (75%), COCO (25%)), depending on the test dataset. In fine-tuning, our network trained with batch size 20 for 68 epochs iterations and the initial learning rate of 1e-4, which decays at 30 epochs and 52 epochs. In both training stage, each batch consists of 3 frames.

We use randomly cropped 356x624 patches for the training, and the inference image size is set to 356x624, which is the same size as the training patch. As for the label assignment, to generate the target category probability and the mask, we use the same metric of SOLO [5]. For the target grid similarity $\mathbf{Sim} \in \mathbb{R}^{(S_h \cdot S_w) \times (S_h \cdot S_w)}$, we assign '1' to every grid that contain the center region of the same instances between two frames. Otherwise, we assign '0'.

2.2. NMS

To obtain the final instance segmentation results for each frame, first we filter out the outputs from the category branch and the mask branch with the threshold 0.1. Then, we apply the Matrix NMS introduced in SOLOv2 [6] and use the segmented instances with a score higher than 0.05 as the final results. The binary masks of the instances are produced by applying the threshold of 0.5. The redundant instance masks may still remain after applying the Matrix NMS. If there are masks with $\mathbf{IoU}^*$ greater than 0.5, we further remove the instance mask that has lower classifica-

tion score. $\mathbf{IoU}^*$ is defined as:

$$\mathbf{IoU}^*(i,j) = \frac{\left|\mathbf{M}^i \cap \mathbf{M}^j\right|}{\left|\mathbf{M}^j\right|}, \text{for } score(i) \geq score(j), \quad (1)$$

where i and j are indices of the instance masks, and the score indicates the classification score.

3. Additional Analysis

In Fig. 2, we provide more visualizations for the soft weights that are used to retrieve the information of the reference frames in the temporal aggregation module, i.e. weights of each grid of the reference frames with respect to the query grids. Our temporal aggregation module accurately gathers the appearance information from the reference frames.

4. Video Comparisons

We provide the overall flow of our VISOLO and the comparison of different VIS methods on the YouTube-VIS 2019 dataset [7] at https://youtu.be/2e8DLjCZf40. We compare our method, VISOLO, with two other online methods: MaskTrack R-CNN [7] and CrossVIS [8]. We chose example videos from the validation set of YouTube-VIS 2019 dataset [7]. The video results suggest that VISOLO produces more robust instance tracks than other online methods, even in difficult cases with occlusion and complex motion. Furthermore, since the YouTube-VIS 2019 validation set for evaluating consists of 5 frame intervals, we also provide video results of our VISOLO using all frames of video in the accompanying video (*VISOLO.mp4*)

5. Broader Impact

Our framework is designed for the online video instance segmentation, which targets to classify and generate spatio-temporal pixel masks for all objects in the video in an online manner. Recently, while many online methods are introduced and show promising results, these methods do not make full use of the information of previous frames. In comparison, our VISOLO focuses on maximizing the use of available information from previous frames while maintaining speed. We believe our network can positively impact many VIS applications that require high accuracy and running in real-time, e.g. autonomous navigation of robots and cars. We want to note that for the community to move in the right direction, the studies on VIS should be aware of potential misuses which violates personal privacy.

References

[1] 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 (CVPR)*, June 2016. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016. 2

[3] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2014. 2

[4] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[5] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[6] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2

[7] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 3

[8] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8043–8052, October 2021. 3