







# VISAGE: Video Instance Segmentation with Appearance-Guided Enhancement

## Supplementary material

Hanjung Kim<sup>1</sup>, Jaehyun Kang<sup>1</sup>, Miran Heo<sup>1</sup>, Sukjun Hwang<sup>2</sup>,  
Seoung Wug Oh<sup>3</sup>, and Seon Joo Kim<sup>1</sup>

<sup>1</sup> Yonsei University

<sup>2</sup> Carnegie Mellon University

<sup>3</sup> Adobe Research

## A Implementation Details

We employ Mask2Former [1] as a query-based detector and initialize our parameters with its official COCO [8] weights. During training, we utilize both COCO pseudo videos, generated through rotation and crop augmentation, and the target video dataset, following previous works [3, 4, 7, 10, 13]. Our batch size is 16 for all datasets, and we sample 4 images from each video. VISAGE is trained for 20,000/60,000/60,000 iterations on YTVIS 2019/2021/OVIS and undergoes decay at 15,000/45,000/45,000 iterations. The initial learning rate is  $1e-4$  and its reduction factor is 0.1. At the inference stage, the memory bank size, denoted as  $W$ , is set to 5, and the resolution of the input frame is resized so that its shortest side is 480 pixels.

## B More Experimental Details

**Simplified Tracker.** Tab. 4 in the main paper demonstrates the effectiveness of our simplified tracker. To substitute our tracker with another, we follow the official code from CTVIS [13] to adopt their tracking style. We select hyperparameter values identical to those used in CTVIS for thresholding the predictions. We have eliminated the heuristic design element, Mask NMS, from the tracker, as shown in the second row of Tab. 4.

**Pseudo Dataset.** To evaluate performance in complex scenarios not easily observed in traditional datasets, we create a pseudo dataset. Using copy-paste augmentation with instances from the COCO [8], we generate 36-frame videos. We manually select instances with high annotation quality from the 21 classes present in both COCO and YouTubeVIS-19 [12]. Randomly choosing 2 or 3 instances, we utilize them to compose each video, with class selection probabilities based on the class distribution in the YouTubeVIS-19 dataset. Background images are randomly chosen from BG-20k [6], and resolutions are randomly selected from cartesian product  $A \times A$  where  $A = \{600, 700, 800, 900\}$ .

Each instance moves along a random Bezier curve, leading to the occurrence of complex scenarios such as occlusion due to simultaneous movements. To ensure consistent depth order when instances overlapped, we maintain a coherent order throughout the video. Additionally, we allow instances to move outside the frame by up to 20% in each direction, creating scenarios where objects exited and re-entered the frame naturally.

The dataset comprises a total of 1000 videos, evenly split into 500 track-type and 500 swap-type videos. In track-type videos, instances move along Bezier curves for all 36 frames. In swap-type videos, instance positions switch at a randomly chosen intermediate frame, returning to their original trajectory after the swap. The pseudo dataset and its generation code are available at GitHub.

**Table 8:** Computation Comparison on YTVIS 2019 validation set.

Method	Metric			YTVIS	YTVIS	OVIS
	FPS	Params	GFLOPs	2019	2021	
MinVIS [5]	25.6	44	3031	47.4	44.2	25.0
CTVIS [13]	13.2	44	3036	55.1	50.1	35.5
VISAGE	23.3	45	3083	55.1	51.6	36.2

## C Comparative Computational Analysis

Our approach, VISAGE, showcases its effective performance through simplicity. To substantiate this, we conduct a comparative analysis of frames per second (FPS) and the number of parameters (Params) with MinVIS [5] and CTVIS [13], both employing the Mask2Former [1] framework and a query-matching approach. All evaluations are carried out on a single A100 GPU, utilizing the YTVIS 2019 validation set equipped with a ResNet-50 backbone.

As shown in Tab. 8, VISAGE exhibits effectiveness at the inference stage. To ensure a fair comparison, each model processes a single frame at a time, and the same frame resolution is used across all methods during evaluation. We measure the FPS on all videos of the YTVIS 2019 validation set. MinVIS, which generates per-frame outputs and associates them using cosine similarity without any heuristic design or memory bank, achieves the highest FPS among all the models. On the other hand, CTVIS, employing some heuristic design in their tracker, has a lower FPS than others. Our VISAGE shows comparable FPS to MinVIS, although its performance is similar to CTVIS.

In Tab. 8, VISAGE shows a slight increase in both parameters and GFLOPs compared to the other two methods. This increase is attributed to VISAGE’s adoption of an additional appearance branch, which results in a larger number of parameters and higher GFLOPs. However, these differences are very marginal.

Notably, when measuring GFLOPs for both VISAGE and MinVIS using the same 10 videos sampled from the YTVIS 2019 validation set, the GFLOPs of VISAGE are only 1.7% higher compared to MinVIS.

VISAGE has similar computational requirements as MinVIS, yet their performances differ significantly. Additionally, VISAGE outperforms CTVIS in accuracy and offers a considerably higher FPS rate. Following these results, our VISAGE, despite its simplicity, not only exhibits competitive performance on various benchmarks but also proves to be highly effective.

## D Additional Experiments

### D.1 Youtube-VIS 2022

We additionally perform experiments on the YouTube-VIS (YTVIS) 2022 dataset, which represents a challenging scenario with longer sequences. The YTVIS 2022 dataset includes 71 extra videos added to the validation set of YTVIS 2021. As shown in Tab. 9, our VISAGE demonstrates comparable performance to both GenVIS [3] and TCOVIS [7].

**Table 9:** Comparisons on the **YouTube-VIS 2022 long videos** sets. Methods are denoted as online or offline, indicated by the text color. **Bold** and underline highlight the highest and second-highest performances, respectively.

Method	Setting	YouTube-VIS 2022				
		AP	AP <sub>50</sub>	AP <sub>75</sub>	AR <sub>1</sub>	AR <sub>10</sub>
MinVIS [5]	online	23.3	47.9	19.3	20.2	28.0
VITA [4]	offline	32.6	53.9	39.3	30.3	42.6
GenVIS [3]	online	<u>37.5</u>	<b>61.6</b>	41.5	32.6	42.2
GenVIS [3]	offline	37.2	58.5	<b>42.9</b>	<u>33.2</u>	40.4
TCOVIS [7]	online	<b>38.6</b>	59.4	<u>41.6</u>	32.8	<b>46.7</b>
<b>VISAGE</b>	online	37.5	<u>60.0</u>	37.1	<b>35.2</b>	44.1

### D.2 Swin-L backbone

In Tab. 10, we evaluate our method using the Swin-L backbone [9] on various benchmarks. VISAGE demonstrates a notable increase in performance compared to the ResNet-50 backbone [2]. When compared with other methods, VISAGE achieves competitive performance across these benchmarks. If VISAGE does not incorporate appearance information by setting  $\alpha$  to 0, overall performance decreases.

Additionally, we conduct further experiments using our pseudo dataset with a stronger backbone. As illustrated in Tab. 11, our VISAGE still outperforms

**Table 10:** Comparisons on the **YouTube-VIS 2019, 2021, and OVIS** validation sets with online methods are presented. Each method is trained using the Swin-L backbone. The best performance is highlighted in **bold**.

Method	YTVIS	YTVIS	OVIS
	2019	2021	
MinVIS [5]	61.6	55.3	39.4
IDOL [11]	64.3	56.1	42.6
GenVIS [3]	64.0	59.6	45.2
DVIS [14]	63.9	58.7	<b>47.1</b>
TCOVIS [7]	64.1	<b>61.3</b>	46.7
CTVIS [13]	<b>65.6</b>	61.2	46.9
VISAGE (w/o app)	63.1	58.6	41.5
VISAGE	64.2	59.6	46.5

**Table 11:** Comparisons on **Pseudo dataset** using Swin-L backbone. **Bold** denote the highest accuracy.

Method	AP	AP <sub>50</sub>	AP <sub>75</sub>	AR <sub>1</sub>	AR <sub>10</sub>	
	Track	GenVIS [3]	71.9	83.7	76.2	74.1
CTVIS [13]		75.0	87.6	81.5	78.6	82.1
VISAGE (w/o app)		73.8	87.0	78.2	76.1	80.3
VISAGE		<b>76.6</b>	<b>88.2</b>	<b>81.8</b>	<b>78.9</b>	<b>83.0</b>
Swap	GenVIS [3]	47.2	64.8	48.5	59.4	67.0
	CTVIS [13]	60.5	79.7	64.3	66.4	69.9
	VISAGE (w/o app)	54.3	75.0	55.5	58.4	63.2
	VISAGE	<b>64.5</b>	<b>82.0</b>	<b>68.1</b>	<b>68.0</b>	<b>73.2</b>

other methods in scenarios where appearance information is crucial. On the track dataset, all methods exhibit marginal differences from each other. Conversely, on the swap dataset, VISAGE significantly outperforms the other methods. Moreover, the performance degradation observed when appearance information is removed from our method shows the effectiveness of appearance cues.

### D.3 Ablation Studies

**Window Size.** We analyze the effect of the window size  $W$  of our memory bank on the YTVIS 2019 validation set. As demonstrated in Tab. 12, using a memory bank improves performance compared to not using one ( $W = 1$ ). Furthermore, a longer window size contributes to this improvement. Given that setting the window size to both 5 and 10 yields the same performance, we set the default window size of  $W = 5$ .

**Table 12:** Ablation study of the window size  $W$ . **Table 13:** Results of Ablation Experiments on  $\alpha$ .

$W$	AP	AP <sub>50</sub>	AP <sub>75</sub>	AR <sub>1</sub>	AR <sub>10</sub>	Dataset	0.00	0.25	0.50	0.75	1.00
2	53.2	75.2	58.7	50.1	62.1	YT21	50.2	50.2	50.8	51.6	23.2
3	54.6	76.8	60.3	50.3	62.7	YT22	32.1	34.4	36.7	37.5	18.0
5	55.1	78.1	60.6	51.0	62.3	Track	65.2	65.5	65.8	65.6	29.2
10	54.0	76.4	59.9	50.2	61.6	Swap	51.8	58.5	63.0	66.1	28.0

**Table 14:** Feature analysis. All experiments are repeated 5 times.

Dataset	Backbone	Transformer Encoder	Per-pixel Embedding	
R50	YT19	$0.96 \pm 0.003$	$0.90 \pm 0.001$	$0.84 \pm 0.001$
	YT21	$0.94 \pm 0.001$	$0.87 \pm 0.002$	$0.79 \pm 0.002$
	OVIS	$0.91 \pm 0.002$	$0.79 \pm 0.002$	$0.66 \pm 0.002$
SwinL	YT19	$0.99 \pm 0.000$	$0.89 \pm 0.001$	$0.85 \pm 0.001$
	YT21	$0.98 \pm 0.000$	$0.85 \pm 0.002$	$0.79 \pm 0.001$
	OVIS	$0.96 \pm 0.001$	$0.76 \pm 0.003$	$0.67 \pm 0.004$

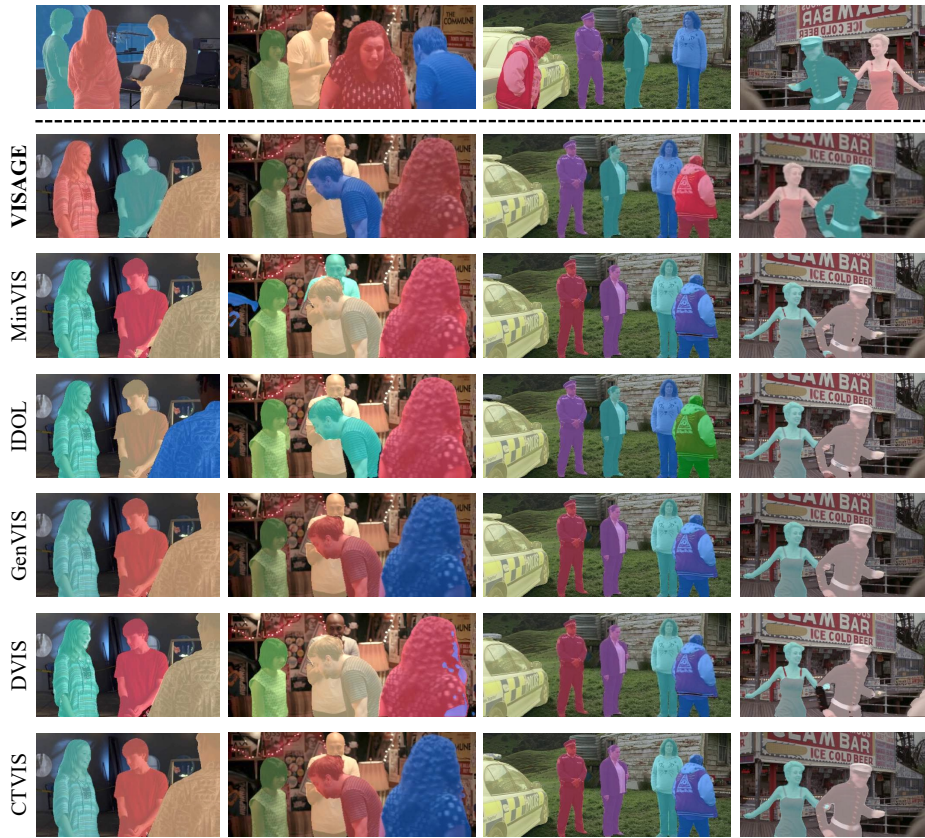
**Additional Analysis of Appearance Weight  $\alpha$ .** Building on the analysis in Tab. 5, we provide additional analysis on the appearance weight  $\alpha$  using the remaining datasets. As shown in Tab. 13, setting  $\alpha$  to 0.75 yields the best performance across most datasets. This result underscores the importance of appearance-guidance.

#### D.4 Analysis on the Model Design.

We generate the appearance query from the backbone feature. However, other features may also contain appearance information. To address this, we conduct experiments to identify which features best capture appearance information, as shown in Tab. 14.

We extract object features from various types: backbone, transformer encoder, and per-pixel embedding (mask feature) using ground-truth masks. For the same object, we measure the cosine similarity between two features extracted from different timesteps. This experiment is conducted on all objects for each dataset using Mask2Former COCO pre-trained weights.

Object features from the backbone show higher cosine similarity compared to those from the transformer encoder and mask feature. We hypothesize that self-attention layers aggregate the features, thereby diminishing their appearance expressiveness. This aligns with the results shown in Tab. 2.



**Fig. 7: Additional qualitative results** present a side-by-side comparison of our method, VISAGE, with previous methods [3, 5, 11, 13, 14] on various scenarios. Each row showcases the predicted results from different methods, with VISAGE distinctly achieving accurate predictions. Consistency across evaluations is maintained by employing the same backbone and benchmark-trained weights for all methods, as illustrated in this collection of images.

## E Additional Qualitative Results

We compare our method, VISAGE, with previous methods [3, 5, 11, 13, 14] across various scenarios. We expand upon the qualitative results presented in the main paper, specifically Fig. 1 and Fig. 2, by including additional comparisons with other methods as depicted in Fig. 7. Notably, only VISAGE accurately predicts the results. All methods evaluated in each video utilize the same backbone and benchmark-trained weights.

Additionally, we compare our method with previous state-of-the-art methods [3, 13] on benchmark videos and demonstrate our generalization ability on real-world videos, as demonstrated in our submitted *demo\_video*. The video

illustrates that our VISAGE exhibits robust performance in scenarios with heavy intersections. Furthermore, VISAGE shows effective performance in situations involving shot changes or dynamic movements, as evidenced by our real-world video samples sourced from YouTube.

## References

1. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
3. Heo, M., Hwang, S., Hyun, J., Kim, H., Oh, S.W., Lee, J.Y., Kim, S.J.: A generalized framework for video instance segmentation. In: CVPR (2023)
4. Heo, M., Hwang, S., Oh, S.W., Lee, J.Y., Kim, S.J.: Vita: Video instance segmentation via object token association. In: NeurIPS (2022)
5. Huang, D.A., Yu, Z., Anandkumar, A.: Minvis: A minimal video instance segmentation framework without video-based training. In: NeurIPS (2022)
6. Li, J., Zhang, J., Maybank, S.J., Tao, D.: Bridging composite and real: Towards end-to-end deep image matting. IJCV (2022)
7. Li, J., Yu, B., Rao, Y., Zhou, J., Lu, J.: Tcovis: Temporally consistent online video instance segmentation. In: ICCV (2023)
8. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
9. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
10. Wu, J., Jiang, Y., Zhang, W., Bai, X., Bai, S.: Seqformer: a frustratingly simple model for video instance segmentation. In: ECCV (2022)
11. Wu, J., Liu, Q., Jiang, Y., Bai, S., Yuille, A., Bai, X.: In defense of online models for video instance segmentation. In: ECCV (2022)
12. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
13. Ying, K., Zhong, Q., Mao, W., Wang, Z., Chen, H., Wu, L.Y., Liu, Y., Fan, C., Zhuge, Y., Shen, C.: Ctvis: Consistent training for online video instance segmentation. In: ICCV (2023)
14. Zhang, T., Tian, X., Wu, Y., Ji, S., Wang, X., Zhang, Y., Wan, P.: Dvis: Decoupled video instance segmentation framework. In: ICCV (2023)