LOCATE: Self-supervised Object Discovery via Flow-guided Graph-cut and Bootstrapped Self-training (Supplementary)

Silky Singh
silsingh@adobe.com
Shripad Deshmukh
shdeshmu@adobe.com
Mausoom Sarkar
msarkar@adobe.com
Balaji Krishnamurthy
kbalaji@adobe.com

Media and Data Science Research
Adobe

1 Supplementary

1.1 Experimental Setup

Network architecture. Similar to GWM [1], we modify the *PixelDecoder* in MaskFormer's segmentation head by appending the layers *[Conv(3), UpsampleNN(2), Conv(1)]* $\times$ *2* to the output layer to get the output segmentation mask at the same resolution as the input. Also, since we directly obtain object segmentations through the network, we set the number of object queries to 1, which results in a single-channel output. Further, we take $sigmoid(x) = \frac{1}{1+e^{-x}}$ on the output of the network (g_θ) to produce values in the range $[0,1]$. We use a threshold of 0.5 in all our experiments to produce a binary segmentation mask.

Training Setup. All the images are interpolated to a resolution of 256×512 (using bi-cubic interpolation), before passing to the segmentation network while training. At the time of loss computation, we also interpolate the pseudo-ground-truths to 256×512 (using nearest interpolation). We employ the binary cross entropy loss function to optimize the weights of the segmentation network, g_θ. We use AdamW [3] optimizer with a base learning rate of 1.5×10^{-4}, linearly decaying at a rate of 0.01 starting from $1. \times 10^{-6}$ for $1.5k$ iterations. Moreover, we train the network until convergence. Empirically, we found $25k$ iterations to be sufficient. We use a single 80GB $A100$ GPU for training the network with a batch size of 8.

Optical Flow computation in graph-cut. Let's denote the frames of a given video by the sequence, $f_1, f_2, ..., f_N$. For a frame f_i, we compute the optical flow between f_i and f_{i+1} for $1 \leq i < N$. For $i = N$, we take the optical flow between f_N and f_{N-1} in our graph-cut step. The obtained optical flow is a 2-channel tensor indicating displacement of pixels in horizontal and vertical directions. We convert these to 3-channel tensors (in RGB format) using open-source implementations, for e.g., https://github.com/ChristophReich1996/Optical-Flow-Visualization-PyTorch.

1.2 Qualitative Results

Figure 1: **Qualitative results of our flow-guided graph-cut approach on all the video benchmarks - DAVIS16 [6], SegTrackv2 [2] and FBMS59 [5].** Our approach incorporating motion information in traditional graph-cut produces high quality object segmentation masks. Quantitatively, this step alone produces results comparable to current state-of-the-art methods on DAVIS16 and STv2 datasets.

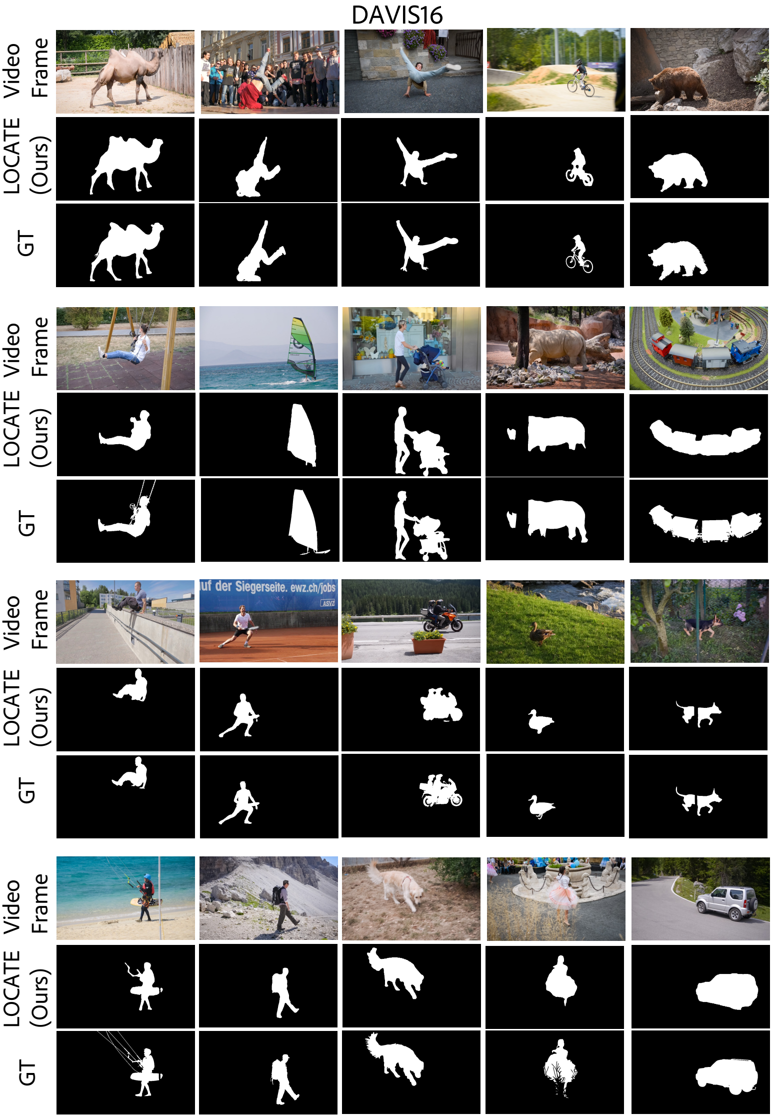

Figure 2: **Qualitative results of our full method (LOCATE) on DAVIS16 [6] benchmark.**

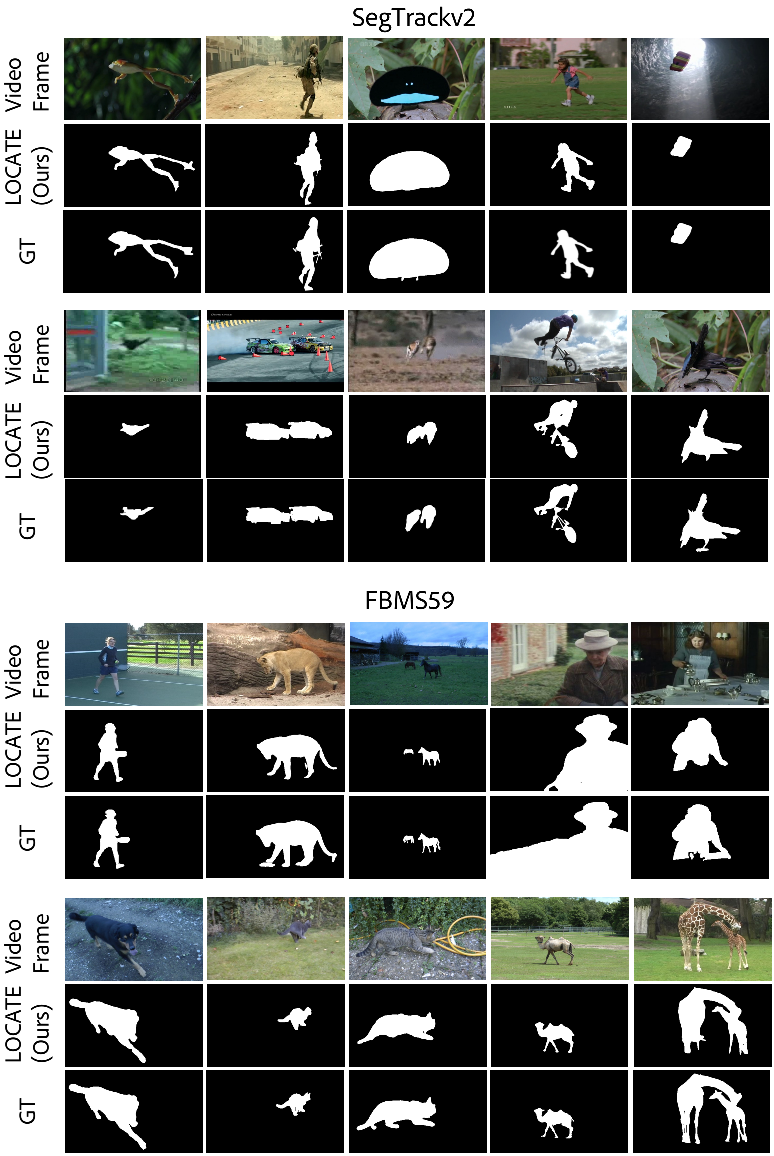

Figure 3: **Qualitative results of our full method (LOCATE) on SegTrackv2 [2] and FBMS59 [5] datasets.**

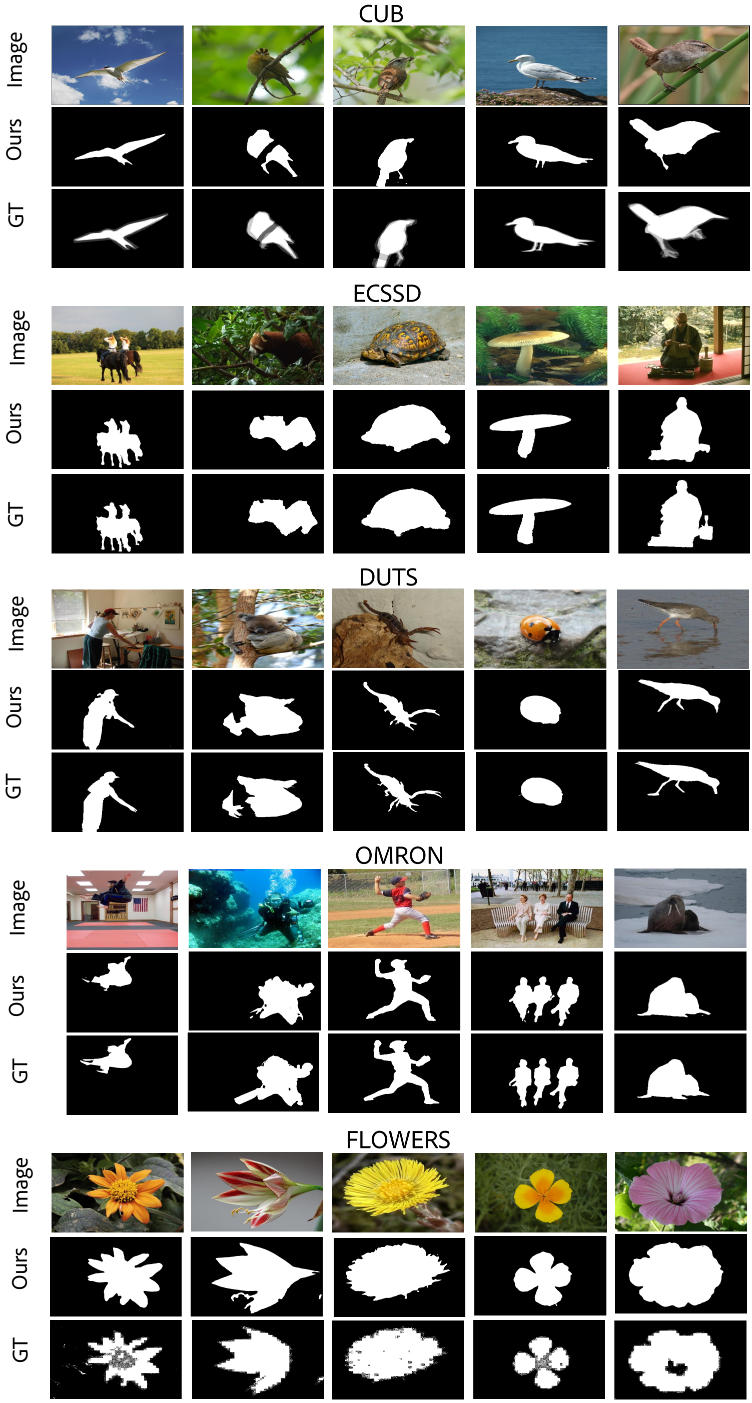

Figure 4: **Qualitative results of our method on image saliency detection (ECSSD [7], DUTS [9], OMRON [10]) and object segmentation (CUB [8], Flowers-102 [2]) benchmarks.**

Figure 5: **Qualitative results of our method (LOCATE) on in-the-wild images.** We asked several users to test our model on random images of their own preference, and collected the results. We show some of the representative examples and their corresponding predicted segmentation masks above. This study reinforces the effectiveness of our model in the wild.

References

[1] Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Guess what moves: Unsupervised video and image segmentation by anticipating motion. *arXiv preprint arXiv:2205.07844*, 2022.

[2] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE international conference on computer vision*, pages 2192–2199, 2013.

[3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[4] Maria-Elena Nilsback and Andrew Zisserman. Delving deeper into the whorl of flower segmentation. *Image and Vision Computing*, 28(6):1049–1062, 2010.

[5] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2013.

[6] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[7] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38 (4):717–729, 2015.

[8] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[9] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 136–145, 2017.

[10] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.