Lucid Data Dreaming for Video Object Segmentation

Anna Khoreva[1] Rodrigo Benenson[2] Eddy Ilg[3] Thomas Brox[3] Bernt Schiele[1]
[1]Max Planck Institute for Informatics [2]Google [3]University of Freiburg

Abstract

Convolutional networks reach top quality in pixel-level video object segmentation but require a large amount of training data ($1k \sim 10k$) to deliver such results. In [6] we propose a new training strategy which achieves state-of-the-art results while using $20\times \sim 100\times$ less annotated data than competing methods. Our approach is suitable for both single and multiple object segmentation. Instead of using large training sets hoping to generalize across domains, we generate in-domain training data using the provided annotation on the first frame of each video to synthesize ("lucid dream") plausible future frames. In-domain per-video training data allows us to train high quality appearance- and motion-based models, as well as tune the post-processing stage. Our results indicate that using a larger training set is not automatically better, and that for the video object segmentation task a smaller training set that is closer to the target domain is more effective.

1. Introduction

Top performing results in video object segmentation are currently obtained using convolutional networks (convnets) [1, 7]. Like most deep learning techniques, convnets for video segmentation benefit from large amounts of training data. Current state-of-the-art methods rely, for instance, on pixel accurate foreground/background annotations of $\sim 2k$ video frames [1] or $\sim 10k$ images [7]. Labelling videos at the pixel level is a laborious task (compared e.g. to drawing bounding boxes for detection), and creating a large training set requires significant annotation effort.

In this work we aim to reduce the necessity for such large volumes of training data. We show that for video object segmentation having a larger training set is not automatically better and that improved results can be obtained by using $20\times \sim 100\times$ less training data than previous approaches [1, 7]. The main insight of our work is that for video object segmentation using few training frames ($1\sim 100$) in the target domain is more useful than using large training volumes across domains ($1k\sim 10k$).

To ensure a sufficient amount of training data close to the target domain, we develop a new technique for synthesizing training data particularly tailored for the pixel-level video object segmentation scenario. We call this data generation strategy "*lucid dreaming*", where the first frame and its annotation mask are used to generate plausible future frames of the videos. The goal is to produce a large training set of reasonably realistic images which capture the expected appearance variations in future video frames, and thus is, by design, close to the target domain.

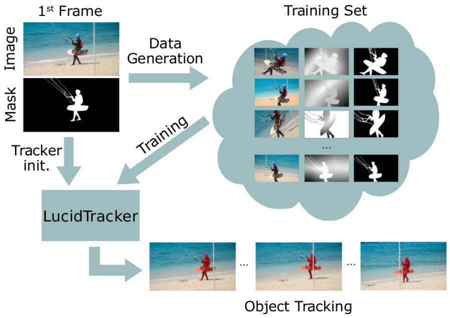


Figure 1: Starting from scarce annotations we synthesize in-domain data to train a specialized object segmenter.

Our approach is suitable for both single and multiple object segmentation in videos. Enabled by the proposed data generation strategy and the efficient use of optical flow, we are able to achieve high quality results while using only ~ 100 individual annotated training frames. For more details of the proposed approach please see [6].

2. LucidTracker architecture

We model video object segmentation as a mask refinement task based on appearance and motion cues. From frame $t-1$ to frame t the estimated mask M_{t-1} is propagated to frame t, and the new mask M_t is computed as a function of the previous mask, the new image $\mathcal{I}_t$, and the optical flow $\mathcal{F}_t$, i.e. $M_t = f\left(\mathcal{I}_t, \mathcal{F}_t, M_{t-1}\right)$. Since objects have a tendency to move smoothly through space in time, there are little changes from frame to frame and mask M_{t-1} can be seen as a rough estimate of M_t. Thus we require our trained convnet to learn to refine rough masks into accurate masks. Fusing the complementary image $\mathcal{I}_t$ and motion flow $\mathcal{F}_t$ enables to exploits the information inherent to video and enables the model to segment well both static and moving objects. To adapt the model f per video the annotated first frame $\mathcal{I}_0$, M_0 is used for finetuning.

RGB image $\mathcal{I}$. Typically a semantic labeller generates labels based on the input image (e.g. $M = g\left(\mathcal{I}\right)$). We use an augmented semantic labeller with an input layer modified to accept 4 channels (RGB + previous mask) so as to generate outputs based on the previous mask estimate, e.g. $M_t = f_{\mathcal{I}}\left(\mathcal{I}_t, M_{t-1}\right)$. Our approach is general and can leverage any semantic labelling architecture. We select the DeepLabv2 architecture with VGG base network [3].

Optical flow $\mathcal{F}$. We use flow in two complementary ways. First, to obtain a better initial estimate of M_t we warp

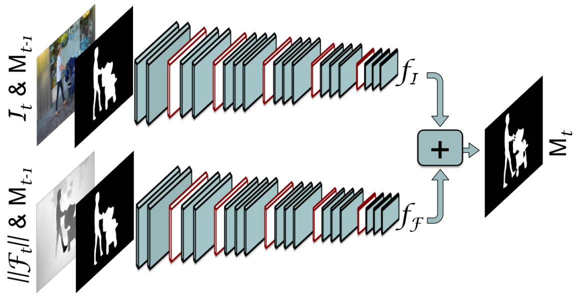


Figure 2: Two stream architecture of LucidTracker.

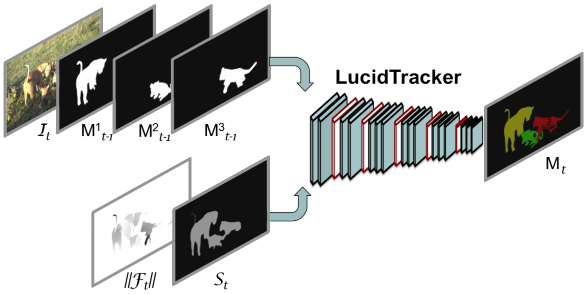


Figure 3: Extension of LucidTracker to multiple objects.

M_{t-1} using the flow $\mathcal{F}_t$: $M_t = f_\mathcal{I}\left(\mathcal{I}_t,\, w(M_{t-1},\, \mathcal{F}_t)\right)$; we call this "mask warping". Second, we use flow as a direct source of information about the mask M_t. When the object is moving relative to background, the flow magnitude $\|\mathcal{F}_t\|$ provides a very reasonable estimate of the mask M_t. We thus consider using a convnet specifically for mask estimation from flow: $M_t = f_\mathcal{F}\left(\|\mathcal{F}_t\|,\, w(M_{t-1},\, \mathcal{F}_t)\right)$, and merge it with the image-only version by naive averaging

$$M_t = 0.5 \cdot f_\mathcal{I}\left(\mathcal{I}_t, \ldots\right) + 0.5 \cdot f_\mathcal{F}\left(\|\mathcal{F}_t\|, \ldots\right). \qquad (1)$$

We use the state-of-the-art optical flow estimation method FlowNet2.0 [4], which itself is a convnet that computes $\mathcal{F}_t = h\left(\mathcal{I}_{t-1},\, \mathcal{I}_t\right)$. For the optical flow magnitude computation we subtract the median motion for each frame, average the magnitude of the forward and backward flow and scale the values per-frame to $[0; 255]$, bringing it to the same range as RGB channels.

In our experiments $f_\mathcal{I}$ and $f_\mathcal{F}$ are trained independently. Our two streams architecture is illustrated in Figure 2. We also explored expanding our network to accept 5 input channels (RGB + previous mask + flow magnitude) in one stream: $M_t = f_{\mathcal{I}+\mathcal{F}}\left(\mathcal{I}_t,\, \|\mathcal{F}_t\|,\, w(M_{t-1},\, \mathcal{F}_t)\right)$, but did not observe much difference in the performance compared to naive averaging. One stream network is more affordable to train and allows to easily add extra input channels, e.g. providing additionally semantic information about objects.

Multiple objects. The proposed framework can easily be extended to segmenting multiple objects simultaneously. Instead of having one additional input channel for the previous frame mask we provide the mask for each object instance in a separate channel, expanding the network to accept $3 + N$ input channels (RGB + N object masks): $M_t = f_\mathcal{I}\left(\mathcal{I}_t,\, w(M^1_{t-1}, \|\mathcal{F}_t\|, ..., w(M^N_{t-1},\, \mathcal{F}_t)\right)$, where N is the number of objects annotated on the first frame.

For the multiple object segmentation task we employ a one-stream architecture for the experiments, using optical flow $\mathcal{F}$ and semantic segmentation $\mathcal{S}$ as additional input channels: $M_t{=}f_{\mathcal{I}+\mathcal{F}+\mathcal{S}}\left(\mathcal{I}_t,\, \|\mathcal{F}_t\|,\, \mathcal{S}_t,\, w(M^1_{t-1},\, \mathcal{F}_t), ..., w(M^N_{t-1},\, \mathcal{F}_t)\right)$. This allows to leverage the appearance model with semantic priors and motion information. See Figure 3 for an illustration. In our preliminary results using a single architecture provides better results than segmenting multiple objects separately, one at a time; and avoids the need to design a merging strategy amongst overlapping tracks.

Semantic labels $\mathcal{S}$. To compute semantic labelling $\mathcal{S}_t = h\left(\mathcal{I}_t\right)$ we use PSPNet [15], trained on Pascal VOC12. Note that we only use $\mathcal{S}_t$ for the multi-object segmentation challenge, discussed in §5. In the same way as for the optical flow we scale $\mathcal{S}_t$ to bring all the channels to the same range.

We additionally experiment with ensembles of different variants, that allows to make the system more robust. For our main results on the multiple object segmentation task we consider an ensemble of four models: $M_t{=}0.25 \cdot (f_{\mathcal{I}+\mathcal{F}+\mathcal{S}} + f_{\mathcal{I}+\mathcal{F}} + f_{\mathcal{I}+\mathcal{S}} + f_\mathcal{I})$, where we merge the outputs of the models by naive averaging.

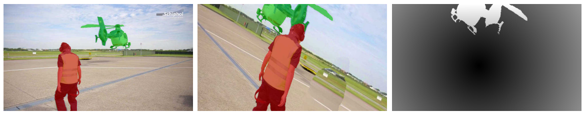
Temporal coherency. To improve the temporal coherency of the proposed framework we introduce an additional step into the system. Before providing as input the previous frame mask warped with the optical flow $w(M_{t-1},\, \mathcal{F}_t)$, we look at frame $t-2$ to remove inconsistencies between the predicted masks M_{t-1} and M_{t-2}. In particular, we split the mask M_{t-1} into connected components and remove all components from M_{t-1} which do not overlap with M_{t-2}. This way we remove possibly spurious blobs generated by our model in M_{t-1}. Afterwards we warp the "pruned" mask $\widetilde{M}_{t-1}$ with the optical flow and use $w(\widetilde{M}_{t-1},\, \mathcal{F}_t)$ as an input to the network. This step is applied only during inference, it mitigates error propagation issues, as well as helps to generate more temporally coherent results.

Post-processing. As a final stage, we refine per-frame t the generated mask M_t using DenseCRF [8]. This adjusts small image details that the network might not be able to handle. It is known by practitioners that DenseCRF is quite sensitive to its parameters and can easily worsen results. We use our lucid dreams to handle per-dataset CRF-tuning.

We refer to our full $f_{\mathcal{I}+\mathcal{F}}$ system as `LucidTracker`. The usage of S_t or model ensemble will be explicitly stated. Our `LucidTracker` obtains best results when first pre-trained on ImageNet, then trained per-dataset using all data from first frame annotations, and finally fine-tuned per-video for each evaluated video. For more details please see [6].

3. Lucid data dreaming

To train the function f one would think of using ground truth data for M_{t-1} and M_t, however such data is expensive



Original image $\mathcal{I}_0$ and mask M_0 | Generated image $\mathcal{I}_\tau$ and mask M_τ | Generated flow magnitude $\|\mathcal{F}_\tau\|$

Figure 4: Lucid data dreaming examples.

to annotate and rare. [1] thus trains on a set of 30 videos ($\sim 2k$ frames) and requires the model to transfer across multiple tests sets. [7] side-steps the need for consecutive frames by generating synthetic masks M_{t-1} from a saliency dataset of $\sim 10k$ images with their corresponding mask M_t. We propose a new data generation strategy to reach better results using only $\sim$100 individual training frames. To ensure our training data is in-domain, we propose to generate it by synthesizing samples from the provided first frame annotation in each target video. This is akin to "lucid dreaming" as we intentionally "dream" the desired data by creating sample images that are plausible hypothetical future frames of the video. See Figure 4 for examples.

Synthesis process. The target domain is the set of future frames of the given video. Traditional data augmentation via small image perturbation is insufficient to cover the expect variations across time, thus a task specific strategy is needed. Across the video the object might change in illumination, deform, translate, be occluded, show different point of views, and evolve on top of a dynamic background. All of these aspects should be captured when synthesizing future frames. We achieve this by cutting-out the foreground object, in-painting the background, perturbing both foreground and background, and finally recomposing the scene. This process is applied twice with randomly sampled transformation parameters, resulting in a pair of frames ($\mathcal{I}_{\tau-1}$, $\mathcal{I}_\tau$) with known pixel-level ground-truth mask annotations ($M_{\tau-1}$, M_τ), optical flow $\mathcal{F}_\tau$, and occlusion regions. The object position in $\mathcal{I}_\tau$ is uniformly sampled, but the changes between $\mathcal{I}_{\tau-1}$, $\mathcal{I}_\tau$ are kept small to mimic the usual evolution between consecutive frames. The same strategy for data synthesis can be employed for multiple object segmentation task. For more details of the synthesis process we refer the reader to [6]. Lucid data dreaming implementation is available at https://www.mpi-inf.mpg.de/lucid-data-dreaming.

4. Single object segmentation results

We present here results for the single object segmentation task and evaluate our method on three datasets: DAVIS$_{16}$ [11], YouTubeObjects [13], and SegTrack$_{v2}$ [9]. To measure the accuracy we use the mean intersection-over-union overlap (mIoU) between the ground truth and the predicted segmentation, averaged across all sequences.

Table 1 compares our results to previous work. Our full system, LucidTracker, provides the best video segmentation quality across three datasets while being trained on each dataset using only one frame per video (50 frames for DAVIS$_{16}$, 126 for YouTubeObjects, 24 for SegTrack$_{v2}$), which is 20× ∼100× less than the top competing methods. Compared to flow propagation methods such as BVS and ObjFlow, we obtain better results by building per-video a stronger appearance model of the tracked object (embodied in the fine-tuned model). Compared to convnet learning methods such as OSVOS and MaskTrack, we require significantly less training data, yet obtain better results.

Method	# training	Flow	Dataset, mIoU		
	images	$\mathcal{F}$	DAVIS$_{16}$	YoutbObjs	SegTrck$_{v2}$
BVS [10]	0	✗	66.5	59.7	58.4
ObjFlow [14]	0	✓	71.1	70.1	67.5
OSVOS [1]	~2.3k	✗	*79.8*	72.5	65.4
MaskTrack [7]	~11k	✓	80.3	72.6	70.3
LucidTracker	**24~126**	✓	**91.2**	**77.3**	**78.0**

Table 1: Comparison across three datasets. Numbers in italic are reported on subsets of DAVIS$_{16}$.

Method	J&F
LucidTracker (5)	67.8
ILC_RIL (4)	69.5
dawnsix	69.7
lixx (2)	73.8
jono (1)	**74.7**

(a) DAVIS 2018 challenge.

Method	J&F
voigtlaender (5)	57.7
haamooon (4)	61.5
vantam299 (3)	63.8
LucidTracker (2)	67.8
lixx (1)	**69.9**

(b) DAVIS 2017 challenge.

Table 2: DAVIS challenge results, test-challenge set.

5. Multiple object segmentation results

We present here an empirical evaluation for multiple object segmentation task and evaluate our method on DAVIS$_{17}$ [12]. To measure the accuracy of multiple object segmentation we use the region (J) and boundary (F) measures [12]. The average of J and F measures (J&F) is used as overall performance score.

Tables 2a and 2b presents the results of the 2017 and 2018 DAVIS Challenge competitions [12, 2]. Our main results are obtained via an ensemble of four different models ($f_{\mathcal{I}}$, $f_{\mathcal{I}+\mathcal{F}}$, $f_{\mathcal{I}+\mathcal{S}}$, $f_{\mathcal{I}+\mathcal{F}+\mathcal{S}}$). The proposed system, LucidTracker shows competitive performance, holding the second place in the 2017 competition and the fifth place the 2018 competition. The full system is trained using the standard ImageNet pre-training, Pascal VOC12 semantic annotations for the $\mathcal{S}_t$ input ($\sim 10k$ annotated images), and one annotated frame per test video, 30 frames total on each test set. As shown in Table 3, even without $\mathcal{S}_t$ LucidTracker obtains competitive results. For comparison 2017 entry lixx uses a deeper convnet model (ImageNet pretrained ResNet), trains it over external data ($\sim 120k$ pixel-level annotated images from COCO and Pascal VOC12 for pre-training, and akin to [1] fine-tuning on the DAVIS$_{17}$ train and val sets, $\sim 10k$ annotated frames), and extends it with a box-level object detector (trained over COCO and Pascal VOC12, $\sim 500k$ bounding boxes) and a box-level object re-identification model trained over $\sim 60k$ box annotations. We argue that our system reaches competitive

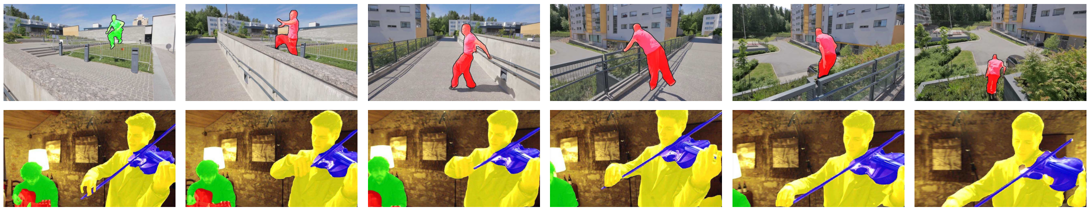


Figure 5: LucidTracker qualitative results. Frames sampled along the video duration (e.g. 50%: video middle point).

Variant	$\mathcal{I}$	$\mathcal{F}$	$\mathcal{S}$	ensmbl	CRF tuning	temp. coherency	DAVIS$_{17}$, test-dev		
							J&F	J	F
Ensemble	✓	✓	✓	✓	✓	✓	**66.6**	**63.4**	**69.9**
	✓	✓	✓	✓	✓	✗	65.2	61.5	69.0
	✓	✓	✓	✓	✗	✗	64.7	60.5	68.9
	✓	✓	✗	✓	✗	✗	64.2	60.1	68.3
$\mathcal{I}+\mathcal{F}+\mathcal{S}$	✓	✓	✓	✗	✗	✗	62.0	57.7	62.2
$\mathcal{I}+\mathcal{F}$	✓	✓	✗	✗	✗	✗	61.3	56.8	65.8
$\mathcal{I}+\mathcal{S}$	✓	✗	✓	✗	✗	✗	61.1	56.9	65.3
$\mathcal{I}$	✓	✗	✗	✗	✗	✗	59.8	63.1	63.9

Table 3: Ablation study on DAVIS$_{17}$, test-dev set.

results with a significantly lower amount of training data.

Ablation study. Table 3 explores how different ingredients contribute to our results. We see that adding extra information to the system, either optical flow magnitude or semantic segmentation, or both, does provide $1\sim2$ percent point improvement. Combining in ensemble four different models allows to enhance the results even further (62.0 vs. 64.7 J&F). Excluding the models which use semantic information from the ensemble results only in a minor drop in the performance (64.2 vs. 64.7 J&F). This shows that even without the semantic segmentation signal $\mathcal{S}_t$ our ensemble result is competitive. Our lucid dreams enable automatic CRF-tuning, allowing further improvement (64.7 → 65.2 J&F). Employing the temporal coherency step during inference brings an additional gain (65.2→66.6 J&F).

6. Conclusion

We have described a convnet-based approach for object segmentation in videos [6]. In contrast to previous work, we show that top results for single and multiple object segmentation can be achieved without requiring external training datasets (neither annotated images nor videos). Even more, our experiments indicate that it is not always beneficial to use additional training data, synthesizing training samples close to the test domain is more effective than adding more training samples from related domains.

Showing that training a video object segmentation convnet can be done with only few (~100) training samples changes the mindset regarding how much general knowledge about objects is required to approach this problem [7, 5], and more broadly how much training data is required to train large convnets depending on the task at hand. We hope our results will fuel the ongoing evolution of convnet techniques for video object segmentation.

References

[1] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. V. Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 3

[2] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018. 3

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 1

[4] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 2

[5] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *arXiv:1701.05384*, 2017. 4

[6] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for multiple object tracking. *arXiv:1703.09554*, 2017. 1, 2, 3, 4

[7] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *arXiv:1612.02646*, 2016. 1, 3, 4

[8] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*. 2011. 2

[9] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 3

[10] N. Maerki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 3

[11] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 3

[12] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 3

[13] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 3

[14] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 3

[15] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2