VideoCutLER: Surprisingly Simple Unsupervised Video Instance Segmentation

Xudong Wang Ishan Misra Ziyun Zeng Rohit Girdhar Trevor Darrell
UC Berkeley FAIR, Meta AI

Code: https://github.com/facebookresearch/CutLER

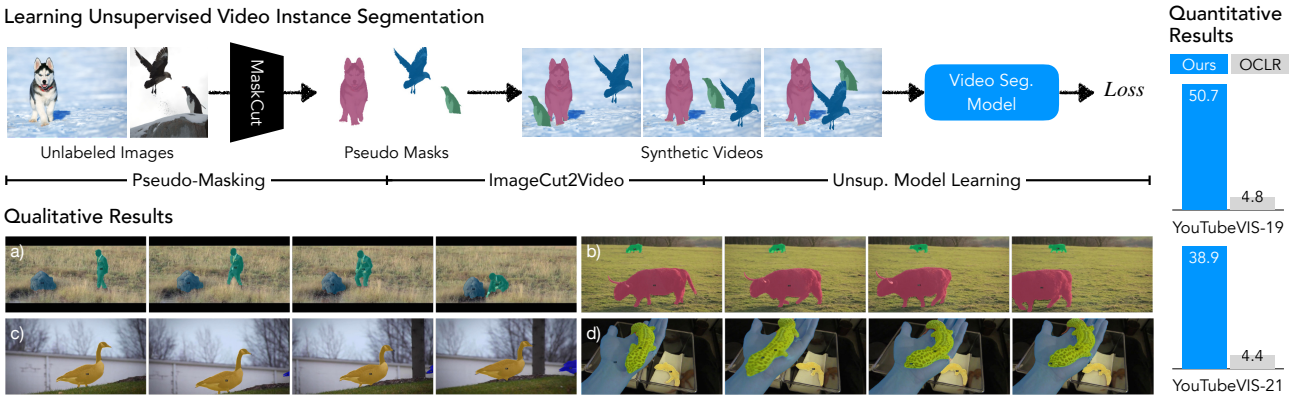

Figure 1. VideoCutLER is a simple unsupervised video instance segmentation method (UnVIS). We show the first competitive unsupervised results on the challenging YouTubeVIS benchmark. Moreover, unlike most prior approaches, we demonstrate that UnVIS models can be learned without relying on natural videos and optical flow estimates. ***Row 1:*** We propose **VideoCutLER**, a simple cut-synthesis-and-learn pipeline that involves three main steps. Firstly, we generate pseudo-masks for multiple objects in an image using MaskCut [35]. Then, we convert a random pair of images in the minibatch into a video with corresponding pseudo mask trajectories using ImageCut2Video. Finally, we train an unsupervised video instance segmentation model using these mask trajectories. ***Row 2:*** Despite being trained only on unlabeled images, at inference time VideoCutLER can be directly applied to unseen videos and can segment and track multiple instances across time (Fig. 1a), even for small objects (Fig. 1b), objects that are absent in specific frames (Fig. 1c), and instances with high overlap (Fig. 1d). ***Column 2:*** Our method surpasses the previous SOTA method OCLR [37] by a factor of 10 in terms of class-agnostic AP_{50}^{video}.

Abstract

Existing approaches to unsupervised video instance segmentation typically rely on motion estimates and experience difficulties tracking small or divergent motions. We present VideoCutLER, a simple method for unsupervised multi-instance video segmentation without using motion-based learning signals like optical flow or training on natural videos. Our key insight is that using high-quality pseudo masks and a simple video synthesis method for model training is surprisingly sufficient to enable the resulting video model to effectively segment and track multiple instances across video frames. We show the first competitive unsupervised learning results on the challenging YouTubeVIS-2019 benchmark, achieving 50.7% AP_{50}^{video}, surpassing the previous state-of-the-art by a large margin. VideoCutLER can also serve as a strong pretrained model for supervised video instance segmentation tasks, exceeding DINO by 15.9% on YouTubeVIS-2019 in terms of AP^{video}.

1. Introduction

Video instance segmentation is vital for various computer vision applications, *e.g.* video surveillance, autonomous driving, and video editing, yet labeled videos are costly to obtain. Hence, there is a pressing need to devise an unsupervised video instance segmentation approach that can comprehend video content comprehensively and operate in general domains without labels.

Prior work in this area typically relies on an optical flow network as an off-the-shelf motion estimator [30, 37, 38]. Although optical flow can be informative in detecting pixel motion between frames, it is not always a reliable technique, particularly in the presence of occlusions, motion blur, complex motion patterns, changes in illuminations, *etc*. As a result, models that heavily rely on optical flow estimations may fail in several common scenarios. For example, stationary or slowly moving objects may have flow estimates similar to the background, causing them to be omitted in the segmentation process (*e.g.*, the parrot with negligible motion is missed in Fig. 2a). Similarly, non-

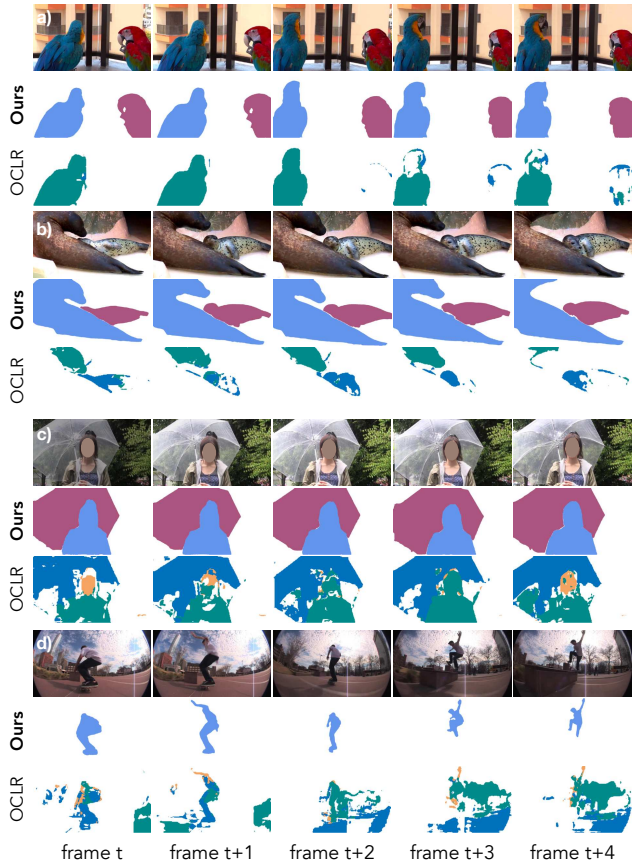

Figure 2. Challenges encountered by the previous state-of-the-art OCLR: Within the framework of OCLR [37], a method that heavily relies on optical flows as model inputs, several distinct failure cases emerge. These include situations where the method struggles to accurately segment both moving and static objects (as demonstrated in Fig. 2a), struggles to effectively track non-rigid objects as a coherent unit (Fig. 2b), encounters difficulties in distinguishing overlapping instances (Fig. 2c), and fails to maintain consistent predictions under varying illumination conditions (Fig. 2d). Nonetheless, many of these challenges can be effectively addressed through the application of our proposed approach, VideoCutLER, without being reliant on the optical estimations used by various prior works [37, 38]. We present qualitative comparisons using the YouTubeVIS dataset [39].

rigid objects with non-consistent motions for several parts have varying optical flows, leading to a failure in segmenting all parts cohesively as a unit if object motion is presumed constant (Fig. 2b). Also, objects with similar motion patterns and high overlap are complex for optic flow methods to accurately distinguish between them, especially in boundary regions (Fig. 2c). Finally, objects with illumination changes across frames can cause optical-flow based models to produce non-consistent and blurred segmentation masks (Fig. 2d). Given the limitations above, we advocate for unsupervised video segmentation models which do not depend on optical flow estimates. We propose a method to train a video segmentation model by generating simple synthetic videos from individual images, without relying on explicit motion estimates or requiring labeled natural videos.

Our method, **VideoCutLER**, is an unsupervised **Video** instance segmentation model that employs a **Cu**t-syn**t**hesis-and-**LEaR**n pipeline (Fig. 1). First, given unlabeled images, we extract pseudo-masks for multiple objects in an image using MaskCut [35], leveraging a self-supervised DINO [4] and a spectral clustering method Normalized Cuts [28] (details in Sec. 3.1). Second, given unlabeled images and their pseudo-masks in a minibatch, we propose ImageCut2Video, a surprisingly simple video synthesis scheme that generates a video from those with corresponding pseudo mask trajectories (details in Sec. 3.2). Finally, those mask trajectories are used to train a video instance segmentation model, aiming to perform object segmentation with temporal consistency across video frames (details in Sec. 3.3). Our model learns to segment and track object instances based on their appearance (feature) similarities across video frames.

Despite being learned from only unlabeled images (and the temporally simple synthetic video sequences we construct from them), VideoCutLER succeeds at multi-instance video segmentation, achieving a new state-of-the-art (SOTA) performance of 50.7% AP_{50}^{video} on YouTubeVIS-2019. This result surpasses the previous SOTA [37] by substantial margins of 45.9% (50.7% vs. 4.8%). This result also considerably narrows the performance gap between supervised and unsupervised learning, reducing it from 29.1% to 11.0% in terms of the AP_{50}^{video}.

Moreover, most prior works on self-supervised representation learning [4, 5, 11, 14, 33] are limited to providing initializations only for the model backbones, with the remaining layers being randomly initialized. In contrast, our pretraining strategy takes a more comprehensive approach that allows all model weights to be pretrained, resulting in a stronger pretrained model better suited for supervised learning. As a result, our method outperforms DINO's [4] AP^{video} on YoutubeVIS-2019 by 15.9%.

Contributions. ***Insights:*** We found that a simple video synthesis method yield surprisingly effective results for training unsupervised multi-instance video segmentation models. This efficacy is achieved without the necessity of explicit motion estimates or the utilization of natural videos (relying solely on unlabeled ImageNet data suffices), a novel aspect that has not been previously demonstrated in the field. ***Methods:*** We propose a simple yet effective cut-synthesize-and-learn pipeline VideoCutLER for learning video instance segmentation models, given unlabeled images. ***Results:*** Our method shows the first successfully results on challenging unsupervised multi-instance video segmentation benchmark YouTubeVIS, outperforming the previous SOTA model's AP_{50}^{video} by a large margin.

	CRW	DINO	OCLR	Ours
Segment multiple objects	✓	✗	✓	✓
Track objects across frames	✓	✗	✓	✓
No need for optical flow	✓	✓	✗	✓
No 1st-frame ground-truth	✗	✗	✓	✓
No human labels at any stage	✗	✗	✓†	✓
Pretrained model for sup. learning	✗	✓	✗	✓

Table 1. We compare previous methods on unsupervised video instance segmentation, including CRW [17], DINO [4], and OCLR [37], with our VideoCutLER in term of key properties. Our VideoCutLER is the only approach that fulfills all these desired properties. †: The optical flow estimator OCLR employs (RAFT [30]) is pretrained on both synthetic data and human-annotated data like KITTI-2015 [18] and HD1K [19].

2. Related Work

Unsupervised video instance segmentation (VIS) requires not only separating and tracking the main moving foreground objects from the background, but also differentiating between different instances, without any human annotations [32]. Previous works [16, 21, 36, 38, 40] on unsupervised video segmentation has primarily centered on ***unsupervised video object segmentation*** (VOS), aiming to detect all moving objects as the foreground and to generate a pixel-level binary segmentation mask, regardless of whether the scene contains a single instance or multiple instances. Despite some works exploring ***unsupervised video instance segmentation*** (VIS), many of these approaches have resorted to either utilizing first frame annotations [4, 17, 22] to propagate label information throughout the video frames or leveraging supervised learning using large amounts of external labeled data [25, 31, 41, 42]. Furthermore, prior studies typically utilized optical flow networks that were pretrained with human supervision using either synthetic data or labeled natural videos [31, 37, 38, 40].

The properties deemed necessary for an unsupervised learning method to excel in video instance segmentation tasks are presented and discussed in Tab. 1. Our proposed method, VideoCutLER, is the only approach that satisfies all these properties, making it an effective and promising solution for unsupervised video instance segmentation.

Unsupervised object discovery aims to automatically discover and segment objects in an image in an unsupervised manner [16, 34–36]. LOST [29] and TokenCut [36] focus on salient object detection and segmentation via leveraging the patch features from a pretrained DINO [4] model. For multi-object discovery, FreeSOLO [34] first generates object pseudo-masks for unlabeled images, then learns an unsupervised instance segmentation model using these pseudo-masks. CutLER [35] presents a straightforward cut-and-learn pipeline for unsupervised detection and segmentation of multiple instances. It has demonstrated promising results on more than eleven different benchmarks, covering a wide range of domains.

In contrast to previous approaches, our unsupervised learning method focuses on simultaneously tracking objects in a video sequence while identifying correspondences between instances across multiple frames.

Self-supervised representation learning generates its own supervision signal by exploiting the implicit patterns or structures present in the input data [3, 4, 14, 15]. Unlike most previous self-supervised learning models, which still require fine-tuning on labeled data to be operative on complex computer vision tasks, such as detection and segmentation, VideoCutLER can tackle these complex, challenging tasks with purely unsupervised learning methods.

3. VideoCutLER

We present VideoCutLER, a simple cut-synthesis-and-learn pipeline consisting of three main steps. First, we generate pseudo-masks for multiple objects in an image using MaskCut (Sec. 3.1). Next, we convert a random pair of images in the minibatch into a synthetic video with corresponding pseudo mask trajectories using ImageCut2Video (Sec. 3.2). Finally, we train an unsupervised video instance segmentation model using these mask trajectories. As the model inputs do not contain explicit motion estimates, it learns to track objects based on their appearance similarity (Sec. 3.3).

3.1. Single-image unsupervised segmentation

We employ the MaskCut method, introduced in the CutLER [35] method. MaskCut is an efficient spectral clustering approach for unsupervised image instance segmentation and object detection and can discover multiple object masks in a single image without human supervision. MaskCut builds upon a self-supervised DINO model [2] with a backbone of ViT [10] and a cut-based clustering method Normalized Cuts (NCut) [28]. MaskCut first generates a patch-wise affinity matrix $W_{ij} = \frac{K_i K_j}{\|K_i\|_2 \|K_j\|_2}$ using the 'key' features K_i for patch i from DINO's last attention layer. Subsequently, the NCut algorithm [28] is employed on the affinity matrix by solving a generalized eigenvalue problem

$$(D - W)x = \lambda D x \tag{1}$$

where D is a diagonal matrix with $d(i) = \sum_j W_{ij}$ and x is the eigenvector that corresponds to the second smallest eigenvalue λ. Then, the foreground masks M^s can be extracted via bi-partitioning x, which segments a single object within the image. To segment more than one object, MaskCut uses an iterative process that masks out the values in the affinity matrix using the extracted foreground mask:

$$W_{ij}^t = \frac{(K_i \prod_{s=1}^t M_{ij}^s)(K_j \prod_{s=1}^t M_{ij}^s)}{\|K_i\|_2 \|K_j\|_2} \tag{2}$$

and repeats the NCut algorithm. We set $t = 3$ by default.

Although MaskCut can effectively locate and segment multiple objects in an image, it operates only on a single image, lacking temporal consistency in the instance segmentation masks produced across video frames.

3.2. ImageCut2Video Synthesis for Training

We propose a learning-based approach to ensuring temporal consistency in video segmentation masks, based on generating synthetic videos from pairs of individual images and MaskCut masks. Surprisingly, we found that an extremely simple synthetic video generation method yields sufficient training data to learn a powerful video segmentation model that can operate on videos with much greater complexity of motion than is present in the training data.

Given unlabeled images in the minibatch and their pseudo-masks, our ImageCut2Video method synthesizes corresponding videos and pseudo-mask trajectories, thereby allowing us to train the model in an unsupervised manner while offering the necessary supervision for simultaneous detection, segmentation, and tracking of objects in videos.

First, given an image and its corresponding pseudo-masks in the mini-batch, we duplicate the image t times and connect its MaskCut pseudo-masks to form the initial trajectories. This synthetic video, however, only contains static foreground objects. To generate additional trajectories with mobile objects, a second image is randomly selected from the mini-batch, and its objects are cropped using its MaskCut pseudo-masks. These objects are then randomly resized, repositioned, and augmented before being pasted onto the first image. The resulting masks are connected along the temporal dimension to generate additional trajectories with mobile objects.

Specifically, given a target image I_1, a random source image I_2 in the mini-batch and its corresponding set of binary pseudo-masks $\{M_2^1, ..., M_2^s\}$, we first apply a transformation function $\mathcal{T}$ to resize and shift these pseudo-masks randomly. This gives us a new set of pseudo-masks $\{\hat{M}_2^1, ..., \hat{M}_2^s\}$, where $\hat{M}_2^s = \mathcal{T}(M_2^s)$. Next, we synthesize a video with t frames by duplicating image I_1 for t times and pasting the augmented masks onto I_1 using:

$$I_1^t = I_1 \times \Pi_{i=1}^{s}(1-\hat{M}_2^i) + I_2 \times (1-\Pi_{i=1}^{s}(1-\hat{M}_2^i)) \quad (3)$$

where $\times$ refers to element-wise multiplication.

3.3. Video Segmentation Model

During training, the synthetic videos produced by ImageCut2Video, comprising both mobile and stationary objects, are used as the inputs to train a video instance segmentation model. The segmentation mask trajectories corresponding to each object in the video serve as 'ground-truth' labels.

We utilize VideoMask2Former [6, 7] with a backbone of ResNet50 [12] as our video instance segmentation (VIS) model. It operates by attending to the 3D spatiotemporal features of our synthetic videos and generating 3D volume predictions of pseudo-mask trajectories using shared queries across frames. The shared queries across frames enable the model to segment and track object instances based on their appearance (feature) similarities, making it a powerful framework for analyzing video sequences.

4. Implementation Details

VideoCutLER. We first employ the MaskCut approach on images preprocessed to a resolution of 480×480 pixels. We then compute a patch-wise cosine similarity matrix using the pretrained ViT-Base/8 DINO [4] model, which serves as input to the MaskCut algorithm for initial segmentation mask generation. We set $t = 3$, which is the maximum number of masks per image. To refine the segmentation masks, we employ a post-processing step using Conditional Random Fields (CRFs) [20], which enforces smoothness constraints and preserves object boundaries, resulting in improved segmentation masks.

Next, we use ImageCut2Video to synthetic videos given images and their pseudo-masks in a mini-batch. We found that synthetic videos with two frames are sufficient to train a video instance segmentation model; therefore, we use $s = 2$ by default. We randomly change the brightness, contrast, and rotation of the masks to create new variations of pseudo-masks. Additionally, we randomly resize the pseudo-masks (scale∈[0.8,1.0]), and shift their positions.

Training and test data. Our model is trained solely on the unlabeled images from ImageNet [8], which comprises approximately 1.3 million images. Without further fine-tuning on any video datasets, we test our model's zero-shot unsupervised video instance segmentation performance on four multi-instance video segmentation benchmarks, including YouTubeVIS-2019 [39], YouTubeVIS-2021 [39], DAVIS2017 [26], and DAVIS2017-Motion [26, 27].

YoutubeVIS-2019 and YouTube-VIS2021 contain 2,883 high-resolution YouTube videos and 3,859 high-resolution YouTube videos, respectively. We evaluate the zero-shot unsupervised learning performance on their training splits in a class-agnostic manner. For DAVIS-2017, we evaluate our model's performance on the 30 videos from its val set.

Training settings. 1) Unsupervised Image Model Pretraining: We first pretrain a Mask2Former [7] model with a backbone of ResNet50 [12] on ImageNet using MaskCut's pseudo-masks. The model is optimized for 160k iterations, with a batch size of 16 and a learning rate of 0.00002. The learning rate is decayed by a factor of 20 at iteration 80,000. To prevent overfitting, a dropout layer with a rate of 0.3 is added after the self-attention layers of transformer decoders. **2) Unsupervised Video Model Learning:** We initialize the VideoMask2Former model [6] with model weights from the previous stage, and then fine-tune

Methods	Training settings			YouTubeVIS-2019							YouTubeVIS-2021						
	flow	videos	sup.	AP_{50}	AP_{75}	AP	AP_S	AP_M	AP_L	AR_{10}	AP_{50}	AP_{75}	AP	AP_S	AP_M	AP_L	AR_{10}
MotionGroup* [38]	✓	✓	✗	1.3	0.1	0.3	0.2	0.3	0.5	1.7	1.1	0.1	0.2	0.1	0.2	0.5	1.5
OCLR* [37]	✓	✓	✗†	4.8	0.4	1.3	0.0	1.2	5.5	11.0	4.4	0.3	1.2	0.1	1.6	7.1	9.6
CutLER‡	✗	✗	✗	37.5	14.6	17.1	3.3	13.9	27.6	30.4	29.2	10.4	12.8	3.1	12.8	27.8	22.6
VideoCutLER	✗	✓*	✗	50.7	24.2	26.0	5.6	20.9	37.9	42.4	38.9	19.0	17.1	5.3	18.3	37.5	31.3
vs. prev. SOTA				**+12.8**	**+9.6**	**+8.9**	**+2.3**	**+7.0**	**+10.3**	**+12.0**	**+9.7**	**+8.6**	**+4.3**	**+2.2**	**+5.5**	**+9.7**	**+8.7**

Table 2. Zero-shot unsupervised multi-instance video segmentation on YouTubeVIS-2019 and YouTubeVIS-2021. We report the instance segmentation metrics (AP and AR) and training settings. *: reproduced MotionGroup [38] and OCLR [37] results with the official code and checkpoints. †: the optical flow estimator OCLR employs (RAFT [30]) is pretrained on both synthetic data [1, 9] and human-annotated data, such as KITTI-2015 [18] and HD1K [19]. ‡: We train a CutLER [35] model with Mask2Former as a detector on ImageNet-1K, following CutLER's official training recipe, and use it as a strong baseline. ❋: VideoCutLER is trained on synthetic videos generated using ImageNet. Sup and flow denote human supervision and optical flow information, respectively. We evaluate results on YouTubeVIS's `train` splits in a class-agnostic manner (note: we never train on YouTubeVIS).

it on the synthetic videos we construct from ImageNet. We train VideoCutLER on 8 A100 GPUs for 80k iterations, using the AdamW optimizer [24]. We set the initial learning rate to 0.000005 and apply a learning rate multiplier of 0.1 to the backbone. A dropout layer with a rate of 0.3 is added after the self-attention layers of transformer decoders.

Evaluation metric AP^{video} and AR^{video}: The evaluation metrics used in YouTubeVIS are Averaged Precision (AP) and Averaged Recall (AR), which are similar to those used in COCO [23]. The evaluation is specifically conducted at 10 intersection-over-union (IoU) thresholds ranging from 50% to 95% with a step of 5% [39]. However, unlike in image instance segmentation, each instance in a video comprises a sequence of masks, so the IoU computation is performed not only in the spatial domain, but also in the temporal domain by summing the intersections at every single frame over the unions at every single frame.

Evaluation metric $\mathcal{J}$ and $\mathcal{F}$: For DAVIS [27], we report results using their official evaluation metrics $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$ and $\mathcal{F}$. The region measure ($\mathcal{J}$) [27] is the intersection-over-union (IoU) score between the algorithm's mask and the ground-truth mask. The boundary measure ($\mathcal{F}$) [27] is the average precision of the boundary of the algorithm's mask. The evaluation metrics are computed separately for each instance, and then the results are averaged over all instances to get the final score. $\mathcal{J}\&\mathcal{F}$ is the mean of $\mathcal{J}$ and $\mathcal{F}$.

5. Experiments

We evaluate the performance of VideoCutLER on several video instance segmentation benchmarks. In Sec. 5.1, we demonstrate that our approach can effectively perform segmentation and tracking of multiple objects in videos, even when trained on unlabeled ImageNet images without any form of supervision. Our experimental results reveal that our method can drastically reduce the performance gap between unsupervised and supervised learning methods for video instance discovery and tracking. Furthermore, Sec. 5.2 demonstrates that fine-tuning VideoCutLER leads to further performance gains in video instance segmentation, surpassing previous works such as DINO in both fully supervised learning and semi-supervised learning tasks. In Sec. 5.3, we conduct an ablation study to examine the impact of key components and their hyperparameters.

5.1. Unsupervised Zero-shot Evaluations

In this section, we evaluate the performance of our method against previous state-of-the-art approaches on various video instance segmentation benchmarks.

Evaluating unsupervised video instance segmentation poses two main challenges. Firstly, as unsupervised learning methods train the model without semantic classes, the class-aware video segmentation setup cannot be used directly for an evaluation. As a result, following previous works, we evaluate video instance segmentation results in a class-agnostic manner. Secondly, video instance segmentation datasets often annotate only a subset of the objects in the video, which makes Average Recall (AR) a valuable metric that does not penalize models for detecting novel objects not labeled in the dataset [35]. Therefore, we report both AR and AP for YouTubeVIS. Regarding DAVIS, we use the official unsupervised learning metrics $\mathcal{J}$, $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$. All these metrics assess the performance of unsupervised video instance segmentation in a class-agnostic manner. Sec. 4 lists more details on evaluation metrics.

Detailed comparisons on YouTubeVIS. Tab. 2 presents a summary of the results for unsupervised zero-shot video instance segmentation on the YouTubeVIS-2019 and YouTubeVIS-2021 datasets. We compare our method's results with the previous state-of-the-art methods OCLR [37] and motion grouping [38]. We reproduce their results using their official code and checkpoints to ensure fairness.

Although OCLR [37] is also trained on synthetic videos, it relies on the off-the-shelf optical flow estimator RAFT [30] to compute optical flows for RGB sequences. It is worth noting that RAFT is pretrained on a combination of synthetic videos [1, 9] and human-annotated videos

Methods	Training settings				DAVIS2017			DAVIS2017-Motion		
	flow	videos	sup.	training data	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$(Mean)	$\mathcal{F}$(Mean)
MotionGroup (sup.) [38]	✓	✓	✗	IN-1K+synthetic	-	-	-	39.5	44.9	34.2
Mask R-CNN (w/ flow)* [13, 37]	✓	✓	✗	IN-1K+synthetic	-	-	-	50.3	50.4	50.2
OCLR (w/ flow)* [37]	✓	✓	✗	IN-1K+synthetic	39.6	38.2	41.1	55.1	54.5	55.7
VideoCutLER	✗	✗	✗	IN-1K	43.6	41.7	45.5	57.3	57.4	57.2
vs. prev. SOTA					**+4.0**	**+3.5**	**+4.4**	**+2.2**	**+2.9**	**+1.5**

Table 3. Zero-shot unsupervised single/few-instance segmentation. VideoCutLER also outperforms the previous state-of-the-arts on DAVIS2017 and DAVIS2017-Motion. *Note: 12 out of 30 videos from DAVIS2017 and 26 out of 30 videos from DAVIS2017-Motion contain only 1 moving instance. Additionally, DAVIS datasets focus solely on the performance of moving prominent objects, even in videos where multiple objects are present.* This disadvantages our model since it can segment both static and moving objects and has not been exposed to any downstream videos during training. *: utilize optical flow predictions from RAFT [30], which is pretrained on external videos. All methods are evaluated in a zero-shot manner, *i.e.* no fine-tuning on target videos.

such as KITTI-2015 [18] and HD1K [19]. Our approach, VideoCutLER, despite not using any optical flow estimations like many previous works on unsupervised video segmentation, achieves over 10× higher AP_{50} and 18× higher AP than OCLR [37] on YouTubeVIS-2019. Additionally, we achieve over 30% higher recall. Furthermore, unlike the previous state-of-the-art method OCLR [37], which exhibits poor performance in segmenting small objects (with 0.0% AP_S), our approach significantly outperforms it. Similar performance gains can be observed on YouTubeVIS-2021. Finally, the performance gains to CutLER [35] demonstrates the effectiveness of VideoCutLER in training unsupervised multi-instance video segmentation models, surpassing CutLER by over 12.8% on YouTubeVIS-2019.

In Fig. 3, we present qualitative visualizations illustrating the zero-shot unsupervised video instance segmentation outcomes of VideoCutLER on YouTubeVIS dataset.

Detailed comparisons on DAVIS. To provide a comprehensive evaluation and comparison with existing unsupervised video instance segmentation approaches, we also assess the performance of our model on the validation sets of DAVIS-2017 and DAVIS2017-Motion [27, 37]. Note that both DAVIS2017 and DAVIS2017-Motion datasets focus only on the performance of instance segmentation on ***prominent moving objects***, even in videos with multiple objects. As a result, only a single or a few objects of interest per video are annotated, which may not reflect the challenges that arise when multiple objects are present.

Although the evaluation of DAVIS is an unfair assessment for us since VideoCutLER is supposed to segment both static and moving objects, whereas DAVIS focuses on moving prominent objects, with only a single or a few moving objects of interest per video annotated. However, Tab. 3 shows that VideoCutLER yields approximately 4% higher $\mathcal{J}$, $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$. The additional results on DAVIS demonstrate that VideoCutLER achieves superior performance not only on static or minimally moving objects but also on dynamic objects, where prior methods relying on optical flow estimates can benefit from additional cues.

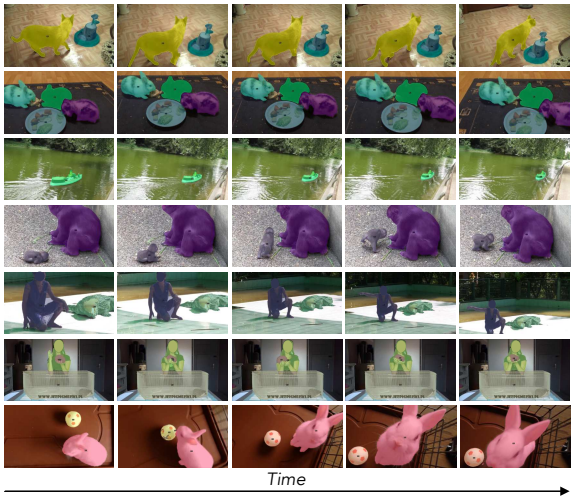

Figure 3. We present **qualitative visualizations** illustrating the zero-shot unsupervised video instance segmentation outcomes of VideoCutLER on YouTubeVIS dataset. It's noteworthy that VideoCutLER is solely pretrained on image dataset ImageNet-1K, and its evaluation is conducted directly on the video dataset YouTubeVIS (no further fine-tuning required). The visual results provided effectively highlight that VideoCutLER is capable of segmenting and tracking multiple instances, delivering consistent tracking results across video frames, and successfully distinguishing between various instances, even when significant overlapping occurs. We show more demo results in appendix.

Comparison of supervised and unsupervised learning in object discovery and tracking abilities is presented in Tab. 4. We train a supervised MaskTrack R-CNN [39] model on the human-annotated training set of YouTubeVIS-2019 dataset, and evaluate it in a class-agnostic manner on the videos that are not shared between YouTubeVIS-2019 and YouTubeVIS-2021 datasets [39]. Tab. 4 shows that our VideoCutLER model significantly narrows the gap between supervised learning and unsupervised learning methods in terms of the averaged precision AP_{50} (gaps: 29.1%→11.0%) and the averaged recall AR_{100} (gaps: 14.9%→3.2%), particularly for the AR_{100}.

Methods	Training settings				YouTubeVIS-2021\YouTubeVIS-2019						
	flow	videos	sup.	training data	AP_{50}	AP_{75}	AP	AP_S	AP_M	AP_L	AR_{100}
Mask2Former [6]	✓	✓	✗	IN-1K+YT2019	48.9	22.2	24.9	-	-	-	-
MaskTrack R-CNN* [39]	✓	✓	✓	IN-1K+YT2019	32.4	13.0	15.0	8.4	24.9	39.0	20.3
MaskTrack R-CNN* [39]	✓	✓	✓	IN-1K+COCO+YT2019	35.8	18.7	18.7	10.5	31.3	46.8	24.5
OCLR* [37]	✓	✓	✗	IN-1K+synthetic	3.3	0.2	1.0	0.3	2.7	7.5	5.4
VideoCutLER	✗	✗	✗	IN-1K	21.4	7.1	9.0	4.9	13.3	29.6	17.1
vs. prev. SOTA					**+18.1**	**+6.9**	**+8.0**	**+4.6**	**+10.6**	**+22.1**	**+11.7**

Table 4. VideoCutLER greatly narrows the **gap between fully-supervised learning and unsupervised learning** for multi-instance video segmentation. Results are evaluated in a class-agnostic manner on the relative complement of the set of videos from YouTubeVIS-2021 and the set of videos from YouTubeVIS-2019. VideoCutLER and Mask2Former use a backbone of ResNet50. *: reproduced results with the official code and checkpoints. IN-1K refers to ImageNet-1K.

Methods	Architecture	YouTubeVIS-2019						YouTubeVIS-2021					
		AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DINO [4]	Mask2Former [6]	23.0	39.0	23.7	6.0	28.0	34.2	24.6	41.4	25.9	8.7	34.0	39.9
VideoCutLER	Mask2Former [6]	38.9	56.7	43.3	22.1	43.1	51.8	33.4	53.8	36.3	15.7	40.9	54.8
vs. prev. SOTA		**+15.9**	**+17.7**	**+19.6**	**+16.1**	**+15.1**	**+17.6**	**+8.8**	**+12.4**	**+10.4**	**+7.0**	**+6.9**	**+14.9**

Table 5. VideoCutLER can serve as a strong pretrained model for the **supervised video instance segmentation** task. The video segmentation model, Mask2Former, is initialized with various pretrained models, *i.e.*, DINO or VideoCutLER, and fine-tuned on the training set with human annotations. We report the instance segmentation metrics and evaluate the model performance on the `val` splits.

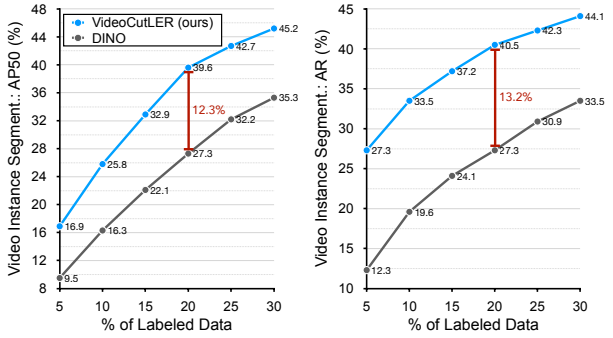

Figure 4. We fine-tune VideoCutLER for **semi-supervised video instance segmentation** on the YouTubeVIS-2019 dataset, using different percentages of labeled training data. We evaluate the performance of our method by reporting the average precision and recall on the validation set of YouTubeVIS-2019. To establish a strong baseline, we use the self-supervised DINO [4] model and initialize the weights of VideoMask2Former with DINO. To ensure a fair comparison, both baselines and VideoCutLER are trained using the same schedule and recipe.

5.2. Label-Efficient and Fully-Supervised Learning

In this section, we investigate VideoCutLER as a pretraining approach for supervised video instance segmentation models, and evaluate its effectiveness in label-efficient and fully-supervised learning scenarios.

Setup. We use VideoMask2Former with a backbone of ResNet50 for all experiments in this section unless otherwise noted. For our experiments on semi-supervised learning, we randomly sample a subset of videos from the training split with different proportions of labeled videos. After pretraining our VideoCutLER model on ImageNet, we fine-tune the model on the YouTubeVIS-2019 [39] dataset with its human annotations. For our experiments on the fully-supervised learning task, we fine-tune the VideoCutLER model on all available labeled data from the training sets of YouTubeVIS. For baselines, we initialize a VideoMask2Former model with a DINO [4] model pre-trained on ImageNet and fine-tuned on labeled videos. Since DINO has shown strong performance in detection and segmentation tasks, it serves as a strong baseline for our experiments.

For semi-supervised learning, both the baselines and our models are trained for $2\times$ schedule, with a learning rate of 0.0001 for all model weights, except for the final classification layers, which use a learning rate of 0.0016. We train the models using a batch size of 16 and 8 GPUs. For fully-supervised learning, we use the $1\times$ schedule and a learning rate of 0.0002 for the final classification layers. We evaluate their performance on the `val` split of the YouTubeVIS-2019, and report results from its official evaluation server.

Data for fully-/semi-supervised VIS. We fine-tune the pretrained VideoCutLER model on all or a subset of the training split of YouTubeVIS-2019. We then evaluate the resulting models on the validation set. To ensure a fair comparison, we use the same amount of human annotations to train our model and baselines. Specifically, we initialize the baselines with the DINO-pretrained model and fine-tune them on the training set of the respective dataset. We evaluate the model performance on their validation sets and report results from its official evaluation server.

Results. Most prior approaches on self-supervised representation learning [4, 5, 11, 14, 33] are limited to providing initializations only for the model backbones, with the remaining layers, such as Mask2Former's decoders,

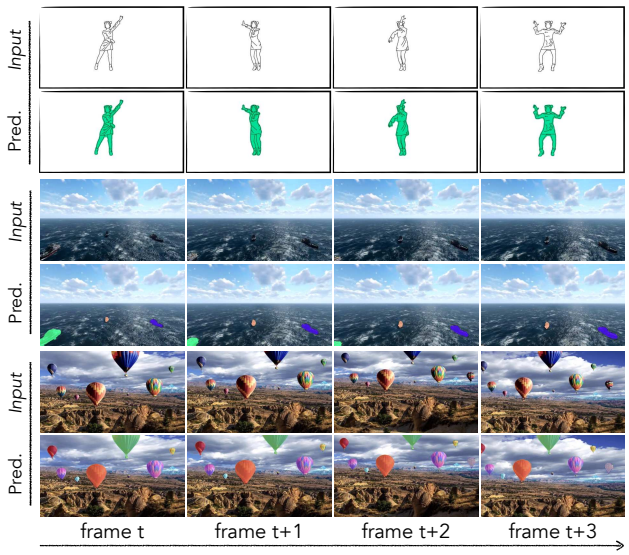

Figure 5. We present qualitative results on videos covering a range of **out-of-domain** sources, *e.g.*, sketches, 3D computer-generated imagery (CGI) and hybrid (CGI + realistic). VideoCutLER can produce high-quality segmentation and tracking results for small objects that are often difficult to distinguish from the background, as well as for object sketches that lack textual information.

being randomly initialized. In contrast, VideoCutLER takes a more comprehensive approach that allows all model weights to be pretrained, resulting in a stronger pretrained model better suited for supervised learning. As a result, as shown in Fig. 4 and Tab. 5, our method outperforms these prior works significantly, offering a strong pretrained model for fully-/semi-supervised learning tasks.

Fig. 4 shows that VideoCutLER consistently outperforms the strong baseline method DINO [4] across all label-efficient learning settings with varying proportions of labeled YouTubeVIS-2019 videos. The most significant performance gains are observed when 20% labeled data is provided, where VideoCutLER exceeds DINO by over 12% AP_{50} and 13.2% AR. As demonstrated in Tab. 5, training the model with all available labeled videos from YouTubeVIS yields considerable performance gains, surpassing DINO by more than 15.9% AP on YouTubeVIS-2019 and 8.8% on YouTubeVIS-2021, respectively.

5.3. Ablation Study

Hyper-parameters and design choices. We present an ablation study on several key hyper-parameters and design choices of VideoCutLER in Tab. 6. First, we analyze the impact of varying the size of video frames used for training VideoCutLER. From Tab. 6a, we observe that the shortest edge length of 240 pixels yields the best performance. Using a larger resolution does not always lead to better results. Next, Tab. 6b shows the effect of the number of frames used

Size →	180	360	480
AP_{50}^{video}	49.9	50.7	50.4

(a) Frame size.

# frames →	CutLER† [35]	2	3	4
AP_{50}^{video}	37.5	49.8	50.7	50.4

(b) # frames. z

Augmentations →	none	+bright	+rotation	+contrast	+crop	all
AP_{50}^{video}	47.8	48.1	48.9	48.3	48.7	50.7

(c) Data augmentations for ImageCut2Video.

Table 6. Ablations for VideoCutLER. We report video instance segmentation result AP_{50}^{video} on YoutubeVIS-2019. **(a)** The impact of varying video frame sizes on training VideoCutLER. **(b)** The effect of the number of frames used for model training. **(c)** The impact of several augmentation methods, including brightness, rotation, contrast, and random cropping, which are used as default during model training. Default settings are highlighted in gray.

for training video instance segmentation models. We found that synthetic videos with three frames are optimal for learning an unsupervised video instance segmentation model. Increasing the number of frames does not result in a further improved performance, aligning with the findings reported in [6]. Furthermore, Tab. 6c investigates the contribution of several augmentation methods, including brightness, rotation, contrast, and random cropping, which are used as default during model training. We found that compared to ImageCut2Video without any data augmentations, adding these augmentations can bring about 3% performance gains.

Generalizability. The results presented in Fig. 5 demonstrate that VideoCutLER can effectively perform video instance segmentation on out-of-domain data sources, *e.g.* sketches, 3D computer-generated imagery, and hybrid videos that combine CGI. These results shows that our model can be applied to a broad range of videos beyond the domains it was initially trained on, *i.e.*, ImageNet.

6. Summary

We presented a simple unsupervised approach to segment multiple instances in a video. Our approach, VideoCutLER, does not require labels, and does not rely on motion-based learning signals like optical flow. In fact, VideoCutLER does not need real videos for training as we synthesize videos using natural images from the ImageNet-1K. Despite being simpler, VideoCutLER outperforms models that use additional learning signals or video data, achieving 10× their performance on benchmarks like YouTubeVIS. Moreover, VideoCutLER is a strong pretrained model for supervised learning. We hope that our approach enables both a wide range of applications in video recognition, as well as its simplicity enables easy future research.

Limitations: while VideoCutLER demonstrates its capability to achieve the state-of-the-art performance without relying on optical flow estimations, potential further improvements may be obtained by leveraging natural videos and integrating joint training with optical flow estimations.

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012. 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2, 3, 4, 7, 8

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 7

[6] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 4, 7, 8

[7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 4

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 5

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2, 7

[12] 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*, pages 770–778, 2016. 4

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2, 3, 7

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 3

[16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 786–802, 2018. 3

[17] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020. 3

[18] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE international conference on computer vision*, pages 3271–3279, 2015. 3, 5, 6

[19] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–28, 2016. 3, 5, 6

[20] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NeurIPS*, 2011. 4

[21] Minhyeok Lee, Suhwan Cho, Seunghoon Lee, Chaewon Park, and Sangyoun Lee. Unsupervised video object segmentation via prototype memory network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5924–5934, 2023. 3

[22] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[25] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. Unovost: Unsupervised offline video object segmentation and tracking. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2000–2009, 2020. 3

[26] 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. 4

[27] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 4, 5, 6

[28] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 2, 3

[29] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021. 3

[30] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 1, 3, 5, 6

[31] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5277–5286, 2019. 3

[32] Wenguan Wang, Tianfei Zhou, Fatih Porikli, David Crandall, and Luc Van Gool. A survey on deep learning technique for video segmentation. *arXiv preprint arXiv:2107.01153*, 2021. 3

[33] Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12586–12595, 2021. 2, 7

[34] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022. 3

[35] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134, 2023. 1, 2, 3, 5, 6, 8

[36] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *arXiv preprint arXiv:2209.00383*, 2022. 3

[37] Junyu Xie, Weidi Xie, and Andrew Zisserman. Segmenting moving objects via an object-centric layered representation. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 5, 6, 7

[38] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177–7188, 2021. 1, 2, 3, 5, 6

[39] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019. 2, 4, 5, 6, 7

[40] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2019. 3

[41] Zhao Yang, Qiang Wang, Song Bai, Weiming Hu, and Philip HS Torr. Video segmentation by detection for the 2019 unsupervised davis challenge. 2019. 3

[42] Tianfei Zhou, Jianwu Li, Shunzhou Wang, Ran Tao, and Jianbing Shen. Matnet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE Transactions on Image Processing*, 29:8326–8338, 2020. 3